ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ
ΣΧΟΛΗ ΟΙΚΟΝΟΜΙΑΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ
Π.Μ.Σ. «ΕΠΙΣΤΗΜΗ ΥΠΟΛΟΓΙΣΤΩΝ»

Υλοποίηση Ασφαλούς Κοινωνικού Δικτύου Με Πολυμεσικό Περιεχόμενο

Διπλωματική Εργασία

Συγγραφείς:
Γαλάτης Αθανάσιος, Α.Μ: 2022201902003, email: pcst19003@uop.gr
Κωνσταντόπουλος Αθανάσιος, Α.Μ: 2022201902009, email: pcst19009@uop.gr

Επιβλέπων:
Βασιλάκης Κωνσταντίνος, email: costas@uop.gr

Ιανουάριος 2021

Πίνακας περιεχομένων

Κατάλογος συντομογραφιών

ACK	ACKnowledgement
AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
CDN(s)	Content Distribution Network(s)
CMS	Content Management System
CNN	Convolutional Neural Network
CSP	Content Security Policy
DBA	Database Administrator
DDNS	Dynamic Domain Name System
(D)DoS	(Distributed) Denial of Service
DOM	Document Object Model
DVCS	Distributed Version Control System
GDPR	General Data Protection Regulation
HTML	HyperText Markup Language
HTTP(S)	Hypertext Transfer Protocol (Secure)
IDS	Intrusion Detection System
IIS	Internet Information Services
IoT	Internet of Things
IP	Internet Protocol
JS	JavaScript
JSP	Jakarta Server Pages
MAC	Media Access Control
MD5	Message Digest Algorithm version Five
ML	Machine Learning
MLP	Multilayer Perceptron
MTCNN	Multi-task Cascaded Convolutional Neural Networks
NIST	National Institute of Standards and Technology
NSA	National Security Agency
OS	Operating system
OWASP	Open Web Application Security Project

PHP	PHP: Hypertext Preprocessor
QR	Quick Response
RDBMS	Relational Database Management System
SHA	Secure Hash Algorithm
SMS	Short Message Service
SMTP	Simple Mail Transfer Protocol
SQL	Structured Query Language
SQLi	SQL Injection
SSD	Single Shot multibox Detector
SSL	Secure Sockets Layer
SYN	SYNchronization
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UI / GUI	User Interface / Graphical User Interface
URL	Uniform Resource Locator
XAMPP	Cross-platform, Apache, MySQL, PHP and Perl
XHTML	Extensible HyperText Markup Language
XSS	Cross Site Scripting
YOLO	You Only Look Once algorithm
ΕΛ/ΛΑΚ	Ελεύθερο Λογισμικό / Λογισμικό Ανοικτού Κώδικα
ΣΝΔ	Συνελικτικά Νευρωνικά Δίκτυα
ΥΚΔ	Υπηρεσίες Κοινωνικής Δικτύωσης

Ευρετήριο εικόνων

Περίληψη

Η ανάπτυξη και η ραγδαία αύξηση της δημοτικότητας των Υπηρεσιών Κοινωνικής Δικτύωσης (ΥΚΔ) έχει δημιουργήσει έναν καινοτόμο κόσμο επικοινωνίας και συνεργασίας στον οποίο ενσωματώνονται διαρκώς νέα χαρακτηριστικά και δυνατότητες. Μέσα σε αυτόν τον κόσμο οι υπηρεσίες αυτές όχι μόνο δεν χάνουν την δυναμική τους αλλά καταλαμβάνουν όλο και σπουδαιότερο μερίδιο στην αγορά, με ένα μεγάλο (και αυξανόμενο) μερίδιο του πληθυσμού να κάνει χρήση των λειτουργιών που αυτές προσφέρουν μέσω κινητών συσκευών, οι οποίες εντάσσονται στον ευρύτερο χώρο των συσκευών του διαδικτύου των πραγμάτων (Internet of Things – IoT). Οι εφαρμογές κοινωνικών δικτύων επιτρέπουν στους χρήστες να ανταλλάσσουν πληροφορίες σχετικά με διάφορα ενδιαφέροντα, όπως επαγγελματικές δραστηριότητες, χόμπι, κ.ά. Αρκετές εμπορικές πλατφόρμες κοινωνικής δικτύωσης που δημιουργήθηκαν πρόσφατα έγιναν εξαιρετικά δημοφιλείς στη διεθνή σκηνή. Εκτός από τα προφανή πλεονεκτήματα που έχουν όσον αφορά την ταχεία ανταλλαγή πληροφοριών σε επαγγελματικό και ιδιωτικό επίπεδο, οι πλατφόρμες εγείρουν πολλά ζητήματα σχετικά με την προστασία της ιδιωτικής ζωής και της ασφάλειας των δεδομένων των χρηστών τους.

Ο στόχος αυτής της διπλωματικής εργασίας είναι ο σχεδιασμός και η υλοποίηση ενός κοινωνικού δικτύου “από το μηδέν” με ένα ευρύ σύνολο δυνατοτήτων καθώς και ο εντοπισμός και η αντιμετώπιση προβλημάτων απορρήτου και ασφάλειας σε αυτό το εγχείρημα. Η ασφάλεια και η ιδιωτικότητα δυστυχώς δεν βρίσκονται στον κεντρικό πυρήνα των σύγχρονων κοινωνικών δικτύων, με αποτέλεσμα το απόρρητο των πληροφοριών όπως και η ασφάλεια των χρηστών να βρίσκονται σε διαρκή κίνδυνο. Λαμβάνοντας αυτά υπ’ όψιν, ο στόχος μας είναι να σχεδιάσουμε και να υλοποιήσουμε ένα σχετικά πλήρες σε λειτουργικότητα και καινοτόμο κοινωνικό δίκτυο, στο οποίο παράλληλα θα διασφαλίζεται ότι τα δεδομένα των χρηστών θα είναι ασφαλή μέσα στο κοινωνικό δίκτυο και θα παραμένουν στην κυριότητά τους, διεπόμενα από τις αρχές της ασφάλειας και του απορρήτου των πληροφοριών, όπως αυτές αποτυπώνονται στα νομοθετικά και κανονιστικά πλαίσια (π.χ. GDPR) (1). Τέλος, θα επιχειρήσουμε να μελετήσουμε μελλοντικά προβλήματα που μπορεί να δημιουργηθούν από τη διαρκώς διευρυνόμενη χρήση συσκευών IoT για πρόσβαση στην εφαρμογή, μία διάσταση που δεν έχει μελετηθεί επαρκώς έως τώρα. Εξετάζονται νέοι μηχανισμοί για την επίλυση κάποιων κλασικών προβλημάτων ασφάλειας των πληροφοριών που αποθηκεύονται στις πλατφόρμες κοινωνικών δικτύων. Προκειμένου να διασφαλιστεί η ασφάλεια και το απόρρητο των δεδομένων των χρηστών από πιθανές παραβιάσεις, αναλύουμε και προτείνουμε μεθόδους για την επίτευξη αυτών των στόχων. Ακολουθούμε πολυάριθμες προσεγγίσεις, τόσο κλασικές όπως τεχνικές επιπέδου δεδομένων (sanitization και validation στα δεδομένα που εισάγονται και κατάλληλη κρυπτογράφηση των δεδομένων τόσο κατά τη μετάδοση (at transit) όσο και κατά την αποθήκευση (at rest)) ή τεχνικές ελέγχου πρόσβασης (π.χ. καταγραφή της διεύθυνσης IP της συσκευής που εκτελεί την πρόσβαση και φιλτράρισμα μέσω αυτής, έλεγχος με βάση τη χώρα πρόσβασης και άλλων χαρακτηριστικών, IP bans για διευθύνσεις ή συσκευές που αναγνωρίζονται ως κακόβουλες), αλλά και πιο πρωτοποριακές όπως έλεγχος για συσκευές IoT και ειδική μεταχείριση τους. Επίσης στόχος είναι να αξιοποιήσουμε τις σχέσεις εμπιστοσύνης που αναπτύσσονται μεταξύ των μελών των κοινωνικών δικτύων και αποτελούν μέρος του μοντέλου των δικτύων αυτών, όπως συμβαίνει και με τις σχέσεις μεταξύ ανθρώπων και

στην πραγματική ζωή, έτσι ώστε να δημιουργηθούν κατά το δυνατόν πιο αξιόπιστοι μηχανισμοί διατήρησης της ιδιωτικότητας των δεδομένων που θα λειτουργούν στο πλαίσιο της διαδικτυακής εφαρμογής. Συνδυάζοντας αυτές τις αρχές σχεδιασμού, δημιουργήθηκε το *Pik-Pok*, ένα κοινωνικό δίκτυο το οποίο παρέχει λειτουργίες όπως η δημιουργία φιλίας, διαφόρων ειδών αναζητήσεις, ανάρτηση κοινοποιήσεων, κοινή χρήση εικόνων, συνομιλία με φίλους, διαχείριση προφίλ χρήστη κ.ά.

Λέξεις κλειδιά: κοινωνικό δίκτυο, υλοποίηση, ασφάλεια, ιδιωτικότητα, διαδίκτυο των πραγμάτων

Abstract

The rapid growth and popularity of Social Networking Services, has created an innovative world of communication and collaboration. In this world, services like these not only do not lose their momentum, but they also occupy an increasingly important market share. Moreover, a large (if not the largest) portion of the population logs in from mobile and IoT devices, which are highly promising to increase the level of comfort and efficiency for the users, but to be able to implement such a world in an evergrowing fashion requires high security, privacy, authentication, and recovery from attacks. In this regard, it is imperative to make the required changes in the architecture of these applications for achieving secure IoT social media environments. Social networking applications allow users to exchange information about various interests, such as professional activities, hobbies, etc. Several commercial social networking platforms that have recently come to light have become extremely popular on the international stage. In addition to the obvious advantages they have over the rapid exchange of information on a professional and private level, the platforms raise many issues related to the protection of the privacy and security of their users' data.

The aim of this dissertation is the design and implementation (almost "from scratch") of a social network with numerous possibilities as well as the identification of privacy and security problems that may occur. As we all know security (and privacy) is not the number one priority of modern social networks and user's data safety is at constant risk. Our goal is both to design and also implement a "complete" and innovative social network and to ensure that users' data will not fall into the wrong hands and that all actions within the social network will be occupied by the principles of security and confidentiality of information. Finally, we will make an effort to study future problems that may arise from the use of IoT devices in the application, an aspect which according to our study, has not been explored so far. New mechanisms are being explored to solve some of the classic security problems of information stored on social networking platforms. In order to ensure the security and privacy of users' data from possible security breaches, we analyze and propose methods to achieve these goals. We follow numerous approaches, both classic ones such as sanitization and validation of the data entered or IP address loging, IP-based and country of origin-based filtering, bans for addresses or devices that are determined to be malicious, proper data encryption and more innovative such as special consideration for IoT devices. The aim is also to take advantage of the relationships of trust that are developed between members and are part of social networks as in real life, in order to address the problem of creating the most reliable privacy mechanisms as part of the web application. Combining these design principles, Pik-Pok was created, a social network that provides functions such as making friends, various types of searches, posting notifications, and sharing images via a user interface, chatting with friends, managing user profiles and more.

Keywords: social network, implementation, security, privacy, Internet of Things

Ευχαριστίες

Ως ένδειξη ευγνωμοσύνης για τη πολύτιμη βοήθειά του, θα θέλαμε να εκφράσουμε την ειλικρινή ευχαριστία μας προς τον επιβλέποντα καθηγητή μας, κ. Βασιλάκη Κωνσταντίνο που μας εμπιστεύτηκε και μας ανέθεσε αυτό το θέμα, αλλά και προς τον καθηγητή μας κ. Τσελίκα Νικόλαο καθώς από το μάθημά του στο μεταπτυχιακό γεννήθηκε στο μυαλό μας η ιδέα για αυτή την διπλωματική εργασία. Θα θέλαμε επίσης να ευχαριστήσουμε θερμά τις οικογένειές μας για τη ψυχολογική και οικονομική υποστήριξη που μας παρείχαν καθ' όλη την διάρκεια των μεταπτυχιακών μας σπουδών. Ιδιαίτερη μνεία πρέπει να γίνει στους φίλους μας για τη στήριξη και την υπομονή τους. Τέλος να ευχαριστήσουμε τους συναδέλφους οι οποίοι δημιουργούν ελεύθερο (open source) λογισμικό καθώς αν και το μεγαλύτερο μέρος του κώδικα γράφτηκε από εμάς, για λειτουργίες όπως εισαγωγή emoticons, barcodes, λειτουργίας αναγνώρισης προσώπων κ.ά., τα ελεύθερα λογισμικά που χρησιμοποιήσαμε μας εξοικονόμησαν αρκετό χρόνο.

1 Εισαγωγή

Στο κεφάλαιο αυτό παρουσιάζονται γενικές πληροφορίες για τα κοινωνικά δίκτυα, τις υπηρεσίες που προσφέρουν και την ιστορία τους. Ακόμα παρατίθεται προγενέστερη βιβλιογραφία που αναφέρεται στις Υπηρεσίες Κοινωνικής Δικτύωσης. Στην πορεία περιγράφουμε τα σχέδια και το όραμά μας σε αυτή την διπλωματική και τι είδους σύστημα επιθυμούσαμε να υλοποιήσουμε, το οποίο ήταν μεγάλο κομμάτι της εργασίας μας. Τέλος γίνεται μία παρουσίαση της δομής της εργασίας μας, έτσι ώστε το έργο να γίνει πιο κατανοητό προς τον αναγνώστη.

1.1 Τι είναι ένα κοινωνικό δίκτυο;

Για την πληρέστερη κατανόηση των λειτουργικών αναγκών που υπάρχουν στα κοινωνικά δίκτυα και των ζητημάτων της ασφάλειας και της ιδιωτικότητας που μπορούν να προκύψουν σε αυτά, αρχικά προσπαθούμε να οριοθετήσουμε την έννοια των κοινωνικών δικτύων. Ανατρέχοντας στη βιβλιογραφία μπορούμε να εντοπίσουμε αρκετούς ορισμούς για την έννοια του κοινωνικού δικτύου και στη συνέχεια παραθέτουμε μερικούς από τους πιο σημαίνοντες εξ αυτών:

- Ο Barnes, (1954) χρησιμοποίησε τον όρο «κοινωνικά δίκτυα» για να δηλώσει τρόπους και μορφές κοινωνικών συνδέσμων. Με τον όρο αυτόν συνδύασε έννοιες που χρησιμοποιούσε ο απλός κόσμος με αυτές που μεταχειρίζονταν οι κοινωνιολόγοι. (2)
- Οι Walker et al, (1967) όρισαν ως κοινωνικό δίκτυο το άθροισμα των προσωπικών επαφών μέσω των οποίων το άτομο διατηρεί την κοινωνική του ταυτότητα, λαμβάνει συναισθηματική υποστήριξη, υλική ενίσχυση και συμμετοχή στις υπηρεσίες. Μέσω αυτών των επαφών έχει πρόσβαση σε πληροφορίες και δημιουργεί νέες κοινωνικές επαφές. Επομένως κοινωνικό δίκτυο μπορεί να ονομαστεί οποιοδήποτε δίκτυο εμπεριέχει μια κοινωνική δομή από ανθρώπους, οι οποίοι συνδέονται μεταξύ τους με διάφορους δεσμούς όπως φιλία, κοινά ενδιαφέροντα, συναδελφικότητα, συγγένεια κ.τ.λ. (3)
- Ο Χτούρης, (2004) ορίζει ως κοινωνικά δίκτυα τα «πολυδιάστατα συστήματα επικοινωνίας και διαμόρφωσης της ανθρώπινης πρακτικής και της κοινωνικής Ταυτότητας». (4)
- Τέλος, σύμφωνα με τους Amichai-Hamburger (2005 και 2009), το διαδίκτυο είναι μια κοινωνική αρένα όπου οι άνθρωποι μπορούν να βρεθούν με άλλους και να αλληλεπιδράσουν μεταξύ τους, ενώ καθημερινά δημιουργούνται πολλά εργαλεία όπως ιστολόγια (blogs), φανταστικά περιβάλλοντα και υπηρεσίες κοινωνικής δικτύωσης (5) (6). Μολονότι στις εργασίες αυτές δεν παρατίθεται ρητώς ένας ορισμός κοινωνικού δικτύου, ο παραπάνω ορισμός απορρέει από το συνολικό κείμενο των αναφερόμενων εργασιών.

Λαμβάνοντας υπ' όψιν τους παραπάνω ορισμούς καθώς και τις συνηθέστερες λειτουργικότητες και πρακτικές που εφαρμόζονται στα πιο διαδεδομένα συστήματα κοινωνικής δικτύωσης, στην παρούσα εργασία ως κοινωνικό δίκτυο θεωρείται ένα σύστημα επικοινωνίας, όμοιο με μία πραγματική κοινωνία (ή μία μικρογραφία αυτής), μέσα από το οποίο το κάθε άτομο δημιουργεί και διατηρεί μία ψηφιακή κοινωνική

ταυτότητα, η οποία επηρεάζει και επηρεάζεται σε μεγάλο βαθμό από την ταυτότητά του και τις δραστηριότητές του στην πραγματική ζωή, αλλά είναι εν τέλει μία διαφορετική ταυτότητα. Στον ψηφιακό κόσμο ενός κοινωνικού δικτύου, το άτομο, μπορεί να δείχνει μόνο τις πτυχές της ζωής του που αυτό επιθυμεί (συμπεριλαμβάνοντας τις μύχιες σκέψεις και επιθυμίες του) και να προβάλλει μία διαφορετική «προσωπικότητα» από αυτή που πραγματικά έχει ή δείχνει στην καθημερινή ζωή του. Μέσα από αυτόν τον κόσμο λαμβάνει συναισθηματική υποστήριξη, υλική ενίσχυση και συμμετοχή σε υπηρεσίες ή/και εμπλέκεται σε διαλόγους και αντιπαραθέσεις. Διαθέτει πρόσβαση σε συνεχή ροή πληροφοριών και μπορεί να δημιουργήσει νέες κοινωνικές επαφές. Συνοψίζοντας, στο πλαίσιο της παρούσας εργασίας ως κοινωνικό δίκτυο μπορεί να θεωρηθεί οποιοδήποτε σύστημα διαθέτει μια κοινωνική δομή από ανθρώπους, οι οποίοι αλληλοεπιδρούν άμεσα ή έμμεσα μεταξύ τους με οποιονδήποτε τρόπο. Δηλαδή είτε συνδέονται άμεσα μεταξύ τους με διάφορους δεσμούς όπως φιλία, κοινά ενδιαφέροντα, συναδελφικότητα, συγγένεια κ.τ.λ., είτε προβάλλουν μία ψηφιακή εικόνα έχοντας έναν υποκειμενικό σκοπό (όπως π.χ. την αποδοχή τους από άλλους ανθρώπους), χωρίς απαραίτητα να έχουν αναπτύξει δεσμούς μαζί τους. Τότε επιδιώκουν την έμμεση αλληλεπίδραση (π.χ. αποδοχή - like από άτομα που δεν γνωρίζουν). Οπότε πρακτικά η θεώρησή μας συμβαδίζει με τον ορισμό των Walker et al. και τον επεκτείνει, λαμβάνοντας υπ' όψιν ότι τα κοινωνικά δίκτυα έχουν μεταβληθεί αρκετά από το 1967.

1.2 Τα κοινωνικά δίκτυα και η ιστορία τους

Οι πρώτες κοινωνικές δικτυακές υπηρεσίες εμφανίστηκαν σχεδόν αμέσως μόλις η τεχνολογία ήταν ικανή να υποστηρίξει την λειτουργία τους. Η βασική ιδέα επικεντρώνεται στη χρήση μεμονωμένων υπολογιστών που συνδέονται ηλεκτρονικά μεταξύ τους ώστε να μπορούν να αποτελέσουν τη βάση της ψηφιακής κοινωνικής αλληλεπίδρασης και δικτύωσης. Αρχικά εμφανίστηκε το ηλεκτρονικό ταχυδρομείο (e-mail) και τα προγράμματα συνομιλίας στις αρχές της δεκαετίας του 1970. Ωστόσο, οι πρώτες κοινότητες δικτύωσης δεν είχαν ακόμα γίνει ορατές μέχρι το 1979, όταν εμφανίστηκε το USENET, ένα ανοικτό σύστημα που στόχευε στην επικοινωνία με email και ήταν αφιερωμένο στις ομάδες ειδήσεων. Στη συνέχεια παρατίθεται μια συνοπτική παρουσίαση κάποιων από τα κυριότερα κοινωνικά δίκτυα των τελευταίων ετών:

- Το **USENET** (7) (8) άρχισε ως ένα σύστημα μηνυμάτων μεταξύ του Πανεπιστημίου του Duke και του Πανεπιστημίου της Βόρειας Καρολίνας, αλλά επεκτάθηκε γρήγορα και σε άλλα αμερικανικά πανεπιστήμια και κυβερνητικές υπηρεσίες. Δημιουργήθηκε το 1979, από δύο φοιτητές, αμέσως μετά την κυκλοφορία του λογισμικού V7 Unix, χρησιμοποιώντας το πρωτόκολλο UNIX-to-UNIX copy protocol (UUCP) (9). Στην ουσία, το USENET επέτρεπε στους χρήστες να δημοσιεύουν και να λαμβάνουν μηνύματα μέσα σε θεματικές ενότητες γνωστές και ως *ομάδες πληροφόρησης*. Παρόμοια λειτουργία με το USENET, την εποχή εκείνη, είχαν το **ARPANET** (10), το **LISTSERV** καθώς και το φόρουμ συζητήσεων **BBS**. Αυτά τα συστήματα έδωσαν μια πρώτη γεύση σχετικά με την ικανότητα που άρχιζαν να αποκτούν οι χρήστες, να αλληλοεπιδρούν δηλαδή σε ένα online κοινωνικό δίκτυο, παρόλο που το καθένα τους ήταν ουσιαστικά κλειστό σύστημα. Το 1993, με την κυκλοφορία της εφαρμογής πλοήγησης Mosaic, τα συστήματα ενώθηκαν σε μια πιο εύχρηστη γραφική διεπαφή.
- Οι πρώτοι κοινωνικοί ιστοχώροι δικτύωσης άρχισαν υπό μορφή γενικευμένων online κοινοτήτων όπως τα **The WELL** 1985, **Theglobe.com** 1994, **Geocities**

1994 και **Tripod** 1995. Αυτές οι πρώτες κοινότητες εστίασαν στο να φέρουν τους ανθρώπους κοντά και να αλληλοεπιδράσουν ο ένας με τον άλλον μέσω των χώρων συνομιλιών και το μοίρασμα προσωπικών πληροφοριών και ιδεών γύρω από οποιαδήποτε θέμα μέσω εργαλείων σύνταξης και δημοσίευσης προσωπικών σελίδων που θεωρείται απαρχή του blogging.

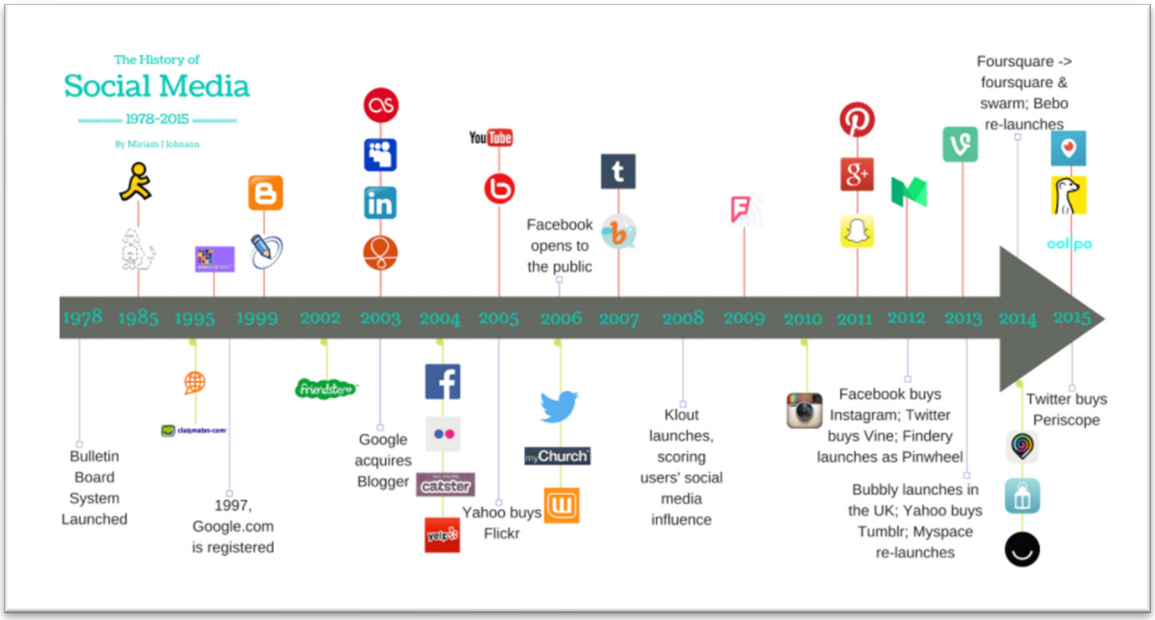
- Οι πρώτες ιστοσελίδες κοινωνικής δικτύωσης εισέρχονται στο διαδίκτυο τη δεκαετία του 1990. Συγκεκριμένα, το **Classmates.com** εμφανίστηκε το 1995 ενώ το **SixDegrees.com** το 1997. Το Classmates.com εστίαζε στους δεσμούς με τους πρώην συμμαθητές και το SixDegrees.com σε άμεσους δεσμούς. Γενικότερος στόχος τους ήταν η σύνδεση ανθρώπων μεταξύ τους, με κύριο εργαλείο τις διευθύνσεις ηλεκτρονικού ταχυδρομείου των χρηστών. Οι κύριες υπηρεσίες που παρείχαν ήταν η δημιουργία profile, η αποστολή μηνυμάτων σε διαδικτυακούς φίλους, και η αναζήτηση μελών με κοινά ενδιαφέροντα. Αυτά τα χαρακτηριστικά υπήρχαν μεμονωμένα σε διάφορα sites, ωστόσο το SixDegrees.com ήταν αυτό που τα συνδύασε όλα μαζί για πρώτη φορά. Ωστόσο, η ιστοσελίδα δεν απέφερε τα αναμενόμενα κέρδη και τελικά έκλεισε.
- Το 1999 εμφανίστηκαν 2 διαφορετικά πρότυπα κοινωνικής δικτύωσης, εκ των οποίων το ένα βασίζονταν στις σχέσεις φιλίας (friendship-based) και το άλλο στις σχέσεις εμπιστοσύνης (trust-based). Μέχρι το 2001 στα πλαίσια αυτών των μοντέλων αναπτύχθηκαν καινοτομίες που έδιναν στους χρήστες όχι μόνο τη δυνατότητα να βλέπουν ποιος είναι φίλος με ποιον, αλλά τους επέτρεπε να έχουν καλύτερο έλεγχο στους δεσμούς τους και την επικοινωνία τους με άλλους χρήστες. Την ίδια ακριβώς χρονιά προτάθηκε μια ακόμα επιπλέον ιδέα κοινωνικών υπηρεσιών δικτύωσης σχετικά με την ενίσχυση των επιχειρησιακών δικτύων. Έτσι, προωθήθηκε το **Ryze.com**.
- Μεταξύ 2002 και 2004, δημιουργήθηκαν τρία ακόμα sites κοινωνικών δικτύων τα οποία και κατάφεραν να είναι τα πιο δημοφιλή της κατηγορίας τους σε όλο τον κόσμο. Το πρώτο από αυτά ήταν το **Friendster**, δεύτερο το **MySpace**, και τέλος το **Bebo**. Μέχρι το 2005 το MySpace έγινε τόσο δημοφιλές που ξεπέρασε σε αριθμό επισκέψεων ακόμα και τη μηχανή αναζήτησης της Google.
- Το **LinkedIn** ξεκίνησε το 2003 και αποτελεί ένα διασυνδεδεμένο δίκτυο για επαγγελματίες διαφόρων κλάδων και ειδικοτήτων από όλο τον κόσμο. Υπάρχει δυνατότητα αναζήτησης, εισαγωγής και συνεργασίας με καταρτισμένους επαγγελματίες. Επίσης στο LinkedIn απεικονίζονται οργανισμοί και εταιρείες και οι συνδέσεις μεταξύ επαγγελματιών και οργανισμών/εταιρειών.
- Το 2004 ήρθε στο φως το **Facebook**, που εντάχθηκε δυναμικά στον χώρο των συστημάτων κοινωνικής δικτύωσης και η ανάπτυξη του ακολούθησε γοργούς ρυθμούς. Το 2006 ήταν η χρόνια που το Facebook σταμάτησε να απευθύνεται μόνο στην κοινότητα των αμερικανικών κολεγίων, και άρχισε να χρησιμοποιείται από ανθρώπους σε όλο τον κόσμο. Αυτό που το έκανε τόσο αγαπητό ήταν το γεγονός ότι αναπτύχθηκε μία πληθώρα εφαρμογών καθιστώντας το ευχάριστο στη χρήση και το γεγονός ότι δεν υπήρχε κανένα γεωγραφικό όριο στην επικοινωνία μεταξύ των χρηστών. Η κοινωνική δικτύωση άρχισε να χρησιμοποιείται ευρέως στον τομέα των επιχειρήσεων περίπου τον Μάρτιο του 2005 όταν η **Yahoo** προώθησε το «Yahoo! 360». Τον Ιούλιο του 2005 η εταιρία News Corporation αγόρασε το MySpace, και ακολούθησε η ITV αγοράζοντας το Friends Reunited στα τέλη του 2005. Στην ίδια περίπου περίοδο αρκετά κοινωνικά δίκτυα άρχισαν να υποστηρίζουν διαφορετικές γλώσσες ώστε να χρησιμοποιούνται και σε άλλες χώρες.

- Επιπλέον, καθώς η χρήση των κοινωνικών δικτύων επεκτάθηκε και αυξήθηκε ο βαθμός δημιουργίας του περιεχομένου αποκλειστικά από το χρήστη, ιστοχώροι που αφορούσαν στο διαμοιρασμό εικόνας και ήχου, άρχισαν να εφαρμόζουν τα χαρακτηριστικά γνωρίσματα των κοινωνικών υπηρεσιών δικτύωσης και να γίνονται και οι ίδιοι αυτοί ιστοχώροι κοινωνικών υπηρεσιών δικτύωσης. Τα παραδείγματα περιλαμβάνουν το **Flickr** για διαμοιρασμό φωτογραφίας, το **Last.FM** για διαμοιρασμός τραγουδιών, και το **YouTube** το 2005 για διαμοιρασμό βίντεο.
- Το **Twitter**, εμφανίστηκε το 2006 και είναι ένα είδος κοινωνικής δικτυακής υπηρεσίας που κύριο χαρακτηριστικό του είναι η δυνατότητα που παρέχεται στον χρήστη να αναφέρει τι κάνει κάθε στιγμή. Με άλλα λόγια, η βασική ιδέα είναι να προσφέρει έναν τρόπο στους χρήστες για να παρέχουν πιο λεπτομερείς αναφορές κατάστασης στους φίλους, την οικογένεια και τις επαφές τους. Αυτές οι ενημερώσεις μπορούν να αφορούν δραστηριότητες και σκέψεις, πραγματοποιούνται δε με ποικίλα μέσα. Τα μέσα αυτά περιλαμβάνουν τη διεπαφή χρήστη του ιστοχώρου, τα στιγμιαία μηνύματα, καθώς και τα SMS από κινητό τηλέφωνο.
- Το **Instagram,** έκανε την εμφάνισή του τον Οκτώβριο του 2010 και είναι μια δωρεάν εφαρμογή κοινωνικής δικτύωσης που δίνει την δυνατότητα επεξεργασίας και κοινοποίησης φωτογραφιών και βίντεο στο διαδίκτυο. Οι χρήστες μπορούν να μοιράζονται φωτογραφίες και βίντεο με τους ακολούθους τους (followers) ή με επιλεγμένη ομάδα φίλων, να σχολιάζουν και να δηλώνουν ότι μια δημοσίευση τους αρέσει. Η δημοφιλής εφαρμογή δημιουργήθηκε από δύο απόφοιτους του Πανεπιστημίου του Στάντφορντ, τους Κέβιν Σίστρομ και Μάικ Κρίγκερ και ξεκίνησε τον Οκτώβριο του 2010. Μόλις δύο μήνες αργότερα, τον Δεκέμβριο του 2010, ο αριθμός των εγγεγραμμένων χρηστών έφτασε το 1.000.000. Σήμερα η εφαρμογή μετράει 20 δισεκατομμύρια φωτογραφίες από όλο τον κόσμο και 1 δισεκατομμύριο ενεργούς χρήστες. Το όνομα της προέρχεται από τον συνδυασμό της λέξης **Instant** (στιγμιαίο) και **telegram** (τηλεγράφημα). Το 2012 η εφαρμογή αγοράστηκε από το Facebook, προς ένα 1 δισεκατομμύριο δολάρια Η.Π.Α.
- Το **TikTok** είναι η τελευταία τάση στα κοινωνικά δίκτυα. Πρόκειται για μία εφαρμογή για iOS και Android, η οποία επιτρέπει την δημιουργία και κοινοποίηση μικρών κωμικών και μη βίντεο. Η εφαρμογή δημιουργήθηκε το 2017 από την ByteDance, για αγορές εκτός της Κίνας. Ήδη από τον Σεπτέμβριο του 2016 η ByteDance έχει ξεκινήσει την εφαρμογή **Douyin** (κινέζικα: 抖音) για τους χρήστες στην Κίνα. Το TikTok και το Douyin είναι αδελφές εφαρμογές, αλλά τρέχουν σε διαφορετικούς διακομιστές προκειμένου η δεύτερη να είναι συμβατή με τους κανονισμούς λογοκρισίας στην Κίνα. Η εφαρμογή επιτρέπει στους χρήστες να δημιουργήσουν μουσική και βίντεο σύντομης διαρκείας. Η εφαρμογή χρησιμοποιείται από εκατομμύρια άτομα στην Ασία, τις Ηνωμένες Πολιτείες και άλλες χώρες του κόσμου. Το TikTok δεν είναι διαθέσιμο στην Κίνα και οι διακομιστές της εφαρμογής TikTok είναι εγκατεστημένοι σε χώρες όπου η εφαρμογή είναι διαθέσιμη. Το 2018, η εφαρμογή απέκτησε σημαντική δημοτικότητα. Για τον Οκτώβριο του 2018, στις ΗΠΑ η εφαρμογή TikTok είχε τον μεγαλύτερο αριθμό λήψεων από οποιαδήποτε άλλη εφαρμογή. Το 2019 είναι διαθέσιμο σε πάνω από 150 αγορές και 75 γλώσσες. Τον Φεβρουάριο του 2019, οι εγκαταστάσεις των TikTok και Douyin ξεπέρασαν το ένα δισεκατομμύριο σε παγκόσμιο επίπεδο, εξαιρουμένων των εγκαταστάσεων της εφαρμογής σε

συσκευές Android στην Κίνα. Το 2020 οι δύο πλατφόρμες (TikTok και Douyin) ξεπέρασαν τις 2 δισεκατομμύρια λήψεις εν μέσω πανδημίας (covid-19). (11)

- Τέλος, μια ακόμα προσθήκη στον λαμπερό κόσμο των κοινωνικών δικτύων αφορά τις εικονικές κοινότητες, «virtual worlds». Πρόκειται για υπολογιστικά προσομοιωμένα περιβάλλοντα μέσα στα οποία κινούνται τρισδιάστατα avatars, δηλ. εικονικοί χαρακτήρες οι οποίοι ελέγχονται από ανθρώπους. Ένας από τους πιο γνωστούς εικονικούς κόσμους αυτή την στιγμή είναι η εφαρμογή **Second Life**, που εισήχθη το 2003. Το Second Life είναι ένας εικονικός κόσμος όπου οι άνθρωποι αλληλεπιδρούν και κοινωνικοποιούνται μέσω τρισδιάστατων avatars. Η εικονική Γη αποτελείται από περιοχές στις οποίες οι χρήστες έχουν πρόσβαση μέσω των avatars τους. Τα avatars ζουν παράλληλα σε αυτό τον κόσμο, όπου εξερευνούν, συναντούν άλλους χρήστες, επικοινωνούν, παίζουν, κάνουν εμπόριο και ανταλλαγές κ.λπ. (12) Ένα άλλο κοινωνικό δίκτυο – εικονικός κόσμος που ανήκει σε αυτή την κατηγορία είναι το δίκτυο **IMVU** (https://secure.imvu.com/welcome/ftux/), το οποίο δίνει έμφαση στα avatars, ενώ υπάρχουν και αρκετά μικρότερα δίκτυα, τα οποία έχουν κάνει τα τελευταία χρόνια την εμφάνιση τους.

Στις προηγούμενες παραγράφους παρουσιάστηκε μόνο ένα μικρό αλλά αντιπροσωπευτικό δείγμα των συστημάτων κοινωνικής δικτύωσης που έχουν δημιουργηθεί έως σήμερα, καθώς η εξέλιξη του Διαδικτύου έχει οδηγήσει στην ύπαρξη πολυάριθμων ιστοσελίδων καθώς και μέσων κοινωνικής δικτύωσης. Στη συνέχεια παρουσιάζεται μια εικόνα που περιλαμβάνει ένα χρονολογικό διάγραμμα των μέσων κοινωνικής δικτύωσης από το 1978 έως το 2018. (13)



Εικόνα 1. Η εξέλιξη των κοινωνικών δικτύων

1.3 Σκοπός- Παρουσίαση του προβλήματος

Σε αυτό το σημείο θα θέλαμε να αναφέρουμε πως γεννήθηκε στο μυαλό μας η ιδέα της δημιουργίας του συγκεκριμένου κοινωνικού δικτύου και πως εξελίχθηκε στην πορεία, στην ανάληψη της παρούσας διπλωματικής εργασίας. Αρχικά, η ιδέα ξεκίνησε από μία εργασία στην οποία στόχος ήταν να δημιουργήσουμε έναν μικρό ιστοχώρο, όπου οι χρήστες έχουν κάποιες από τις δυνατότητες ενός κοινωνικού δικτύου, όπως τη δυνατότητα να δημοσιεύουν νέο περιεχόμενο, να δηλώνουν την αρέσκεια τους ή να σχολιάζουν σε μία υπάρχουσα δημοσίευση. Ο αρχικός συνεπώς στόχος μας ήταν να δημιουργήσουμε μία μικρογραφία ενός κοινωνικού δικτύου. Καθώς όμως προχωρούσαμε σε αυτήν την υλοποίηση, δημιουργήθηκαν πολλές νέες ιδέες για την επέκταση αυτής της εφαρμογής, ιδέες οι οποίες σκεφτήκαμε ότι θα μπορούσαν αν υλοποιηθούν να προσδώσουν νέο ενδιαφέρον στο εγχείρημά μας. Ταυτόχρονα παρατηρήσαμε ότι στα υπάρχοντα κοινωνικά δίκτυα, τόσο η ασφάλεια όσο και η ιδιωτικότητα των χρηστών παραμελούνται πλήρως και αν και έχουν γίνει κάποιες προσπάθειες, δεν έχουν δοθεί ακόμη σαφείς λύσεις σε αυτά τα ζητήματα. Οπότε δεδομένου ότι το κομμάτι της ασφάλειας ανήκει στα ερευνητικά μας ενδιαφέροντα, θελήσαμε ταυτόχρονα να μελετήσουμε και τα προβλήματα ασφαλείας που μπορεί να υπάρξουν και να υλοποιήσουμε υπάρχουσες ή καινοτόμες λύσεις για την αποφυγή αυτών των ζητημάτων και γενικότερα να αναλύσουμε πως τέτοιες λύσεις μπορούν να ενσωματωθούν σε ένα κοινωνικό δίκτυο. Σίγουρα ο δρόμος μας δεν ήταν πάντα εύκολος, καθώς το να δημιουργηθεί από το μηδέν ένα σύστημα υψηλού επιπέδου, με τις καινοτόμες προσθήκες που επιθυμούσαμε, όπως συνειδητοποιήσαμε είχε αρκετές δυσκολίες, όμως πήραμε πολυάριθμες γνώσεις στην πορεία προς την επίτευξη αυτού του στόχου. Ταυτόχρονα έγινε μεγάλη μελέτη της ασφάλειας και των αλγορίθμων κρυπτογράφησης που μας χρειάστηκαν και μελετήθηκαν θέματα που μπορεί να δημιουργηθούν στο μέλλον από IoT συσκευές, μία μελέτη αρκετά καινοτόμα, καθώς όπως είδαμε δεν έχει γίνει κάτι για τις ευπάθειες που μπορεί να δημιουργηθούν στο μέλλον από κινητές συσκευές ή ολόκληρα δίκτυα αυτού του τύπου.

1.4 Δομή της εργασίας

Η δομή της εργασίας είναι κλιμακωτή. Αρχικά, στο κεφάλαιο 2 προβάλλονται οι κυριότερες λειτουργίες της εφαρμογής και ο σχεδιασμός της διεπαφής για αυτήν. Γίνεται αναφορά σε όλες τις λειτουργίες που ήταν σημαντικές και εμπεριέχονται στην εφαρμογή. Στο Κεφάλαιο 3 δίνονται κάποιες βασικές πληροφορίες για τις τεχνολογίες, τα εργαλεία και τις βιβλιοθήκες που χρησιμοποιήθηκαν. Συγκεκριμένα, εξηγούμε τα βασικότερα δομικά στοιχεία που χρησίμευσαν για να οικοδομήσουμε αυτό το κοινωνικό δίκτυο, ώστε να μπορέσει κάποιος να κατανοήσει πως δημιουργήθηκε και λειτουργεί η υπάρχουσα εφαρμογή. Στο Κεφάλαιο 4, περνάμε στην υλοποίηση της εφαρμογής, όπου παρουσιάζουμε τις βασικές οντότητες που έχουν αναδειχθεί και αναλύουμε το σχήμα αποθήκευσής τους στη βάση δεδομένων. Σε αυτό το σημείο προσπαθήσαμε να κάνουμε μια λεπτομερής αναφορά στον τρόπο αποθήκευσης των δεδομένων μας και στους πίνακες της βάσης δεδομένων. Αμέσως μετά, στα κεφάλαια 5 και 6 γίνεται μια λεπτομερής παρουσίαση του κομματιού της ασφάλειας, καθώς αυτή αποτελεί ένα από τα σημαντικότερα ζητήματα της εφαρμογής μας. Γίνεται ανάλυση των κινδύνων και των μέτρων που πάρθηκαν για την αντιμετώπισή τους. Τέλος στο κεφάλαιο 7 περιγράφονται πιθανές βελτιώσεις και επεκτάσεις, που μπορεί να λάβει η εφαρμογή στο μέλλον. Επίσης, αναγράφονται τα συμπεράσματα, που προέκυψαν από την ανάπτυξη της εφαρμογής.

2 Λειτουργικότητα και σχεδιασμός διεπαφής της εφαρμογής

Οι χρήστες της εφαρμογής κατατάσσονται σε ένα πλήθος κατηγοριών, ανάλογα με τον ρόλο που έχουν σε σχέση με το σύστημα κοινωνικής δικτύωσης. Κάθε κατηγορία έχει στη διάθεσή της ένα εξειδικευμένο σύνολο από λειτουργικότητες, που είναι κατάλληλες για τη συγκεκριμένη κατηγορία. Οι ρόλοι που έχουν αναδειχθεί έχουν ως εξής:

- **Επισκέπτης χρήστης**, που χρησιμοποιεί το σύστημα χωρίς να συνδεθεί. Σε γενικές γραμμές, οι επισκέπτες χρήστες έχουν τη δυνατότητα προβολής των φωτογραφιών, των σχολίων (comments) και της δημοφιλίας (likes) αυτών, ενώ παράλληλα είναι δυνατή για κάθε επισκέπτη η δημιουργία λογαριασμού στην εφαρμογή ώστε να μετατραπεί σε εγγεγραμμένο χρήστη.
- **Εγγεγραμμένος χρήστης**, ο οποίος έχει συνδεθεί στο σύστημα. Ένας εγγεγραμμένος χρήστης έχει όλες τις λειτουργικότητες των επισκεπτών-χρηστών και περαιτέρω είναι σε θέση να ανεβάσει φωτογραφίες, να κάνει σχόλια σε όποια φωτογραφία επιθυμεί και να μπορεί να εκδηλώσει την προτίμησή του (like) για μία ή περισσότερες φωτογραφίες. Ταυτόχρονα μπορεί μεταξύ άλλων να κάνει φιλίες με άλλα μέλη του κοινωνικού δικτύου, να συνομιλεί (chatting) μαζί τους, να αποθηκεύει τις δημοσιεύσεις που του αρέσουν και πολλά άλλα.

Οι λειτουργίες για την κάθε κατηγορία χρήστη παρατίθενται αναλυτικά παρακάτω:

1) Επισκέπτης Χρήστης (visitor): Ο επισκέπτης χρήστης, ο οποίος δεν έχει συνδεθεί στην εφαρμογή, μπορεί μεταξύ άλλων:

1. Να έχει πρόσβαση και να προβάλλει όλες τις φωτογραφίες που έχουν ήδη καταχωρηθεί στο κοινωνικό δίκτυο. Εξ ορισμού ο επισκέπτης της κεντρικής σελίδας της υπηρεσίας βλέπει τις ήδη καταχωρημένες φωτογραφίες με φθίνουσα χρονολογική σειρά (από την πιο πρόσφατη προς τη λιγότερο πρόσφατα καταχωρημένη).
2. Να βλέπει από ποιον χρήστη έχει καταχωρηθεί η κάθε φωτογραφία και πότε.
3. Να βλέπει πόσα likes έχει λάβει η κάθε φωτογραφία
4. Να προβάλλει τις φωτογραφίες με φθίνουσα σειρά δημοφιλίας, από την περισσότερο προς τη λιγότερο δημοφιλή (δηλ. οι φωτογραφίες με τα περισσότερα likes πρώτες), επιλέγοντας “Most Popular” από το μενού της εφαρμογής.
5. Να προβάλλει προφίλ άλλων χρηστών. Εκεί έχει την δυνατότητα να δει πληροφορίες για αυτούς και τις δημοσιεύσεις τους (αν είναι δημόσιες), αλλά δεν μπορεί να τους κάνει φίλους, να επικοινωνήσει μαζί τους ή να προβεί σε ενέργειες like και comment (οι λειτουργίες αυτές είναι διαθέσιμες μόνο στους εγγεγραμμένους χρήστες).
6. Να επικοινωνήσει με τους δημιουργούς μέσω φόρμας επικοινωνίας της εφαρμογής για τυχόν ερωτήσεις και προβλήματα.
7. Να αναζητήσει φωτογραφίες βάσει των tags που έχουν προστεθεί στις καταχωρημένες φωτογραφίες του κοινωνικού δικτύου. Η αναζήτηση

λειτουργεί ταιριάζοντας το προς αναζήτηση tag με οποιοδήποτε μέρος των καταχωρημένων tags. Π.χ., κάνοντας αναζήτηση για “summer” επιστρέφονται όλες οι φωτογραφίες, στων οποίων τα tags περιέχεται η λέξη “summer”, π.χ. “summertime”, “I_love_summer”, “mysummer”, “summer-2020”, “corona-virus-summer” κ.τ.λ.

8. Φυσικά έχει την δυνατότητα να δημιουργήσει νέο λογαριασμό (Sign Up) ή να συνδεθεί με κάποιον υπάρχοντα λογαριασμό στην εφαρμογή (Login).

Τονίζεται ότι κατά τη χρήση σαν επισκέπτης χρήστης, δεν είναι διαθέσιμες αρκετές από τις λειτουργίες της εφαρμογής όπως ανέβασμα νέας δημοσίευσης, σχολιασμός, έκφραση αρέσκειας (like), δημιουργία φιλιών, συνομιλίας κ.ά.

2) Εγγεγραμμένος Χρήστης (registered user): Ο εγγεγραμμένος χρήστης, πέραν των παραπάνω λειτουργιών, αφού έχει συνδεθεί στον λογαριασμό του, μπορεί επιπλέον:

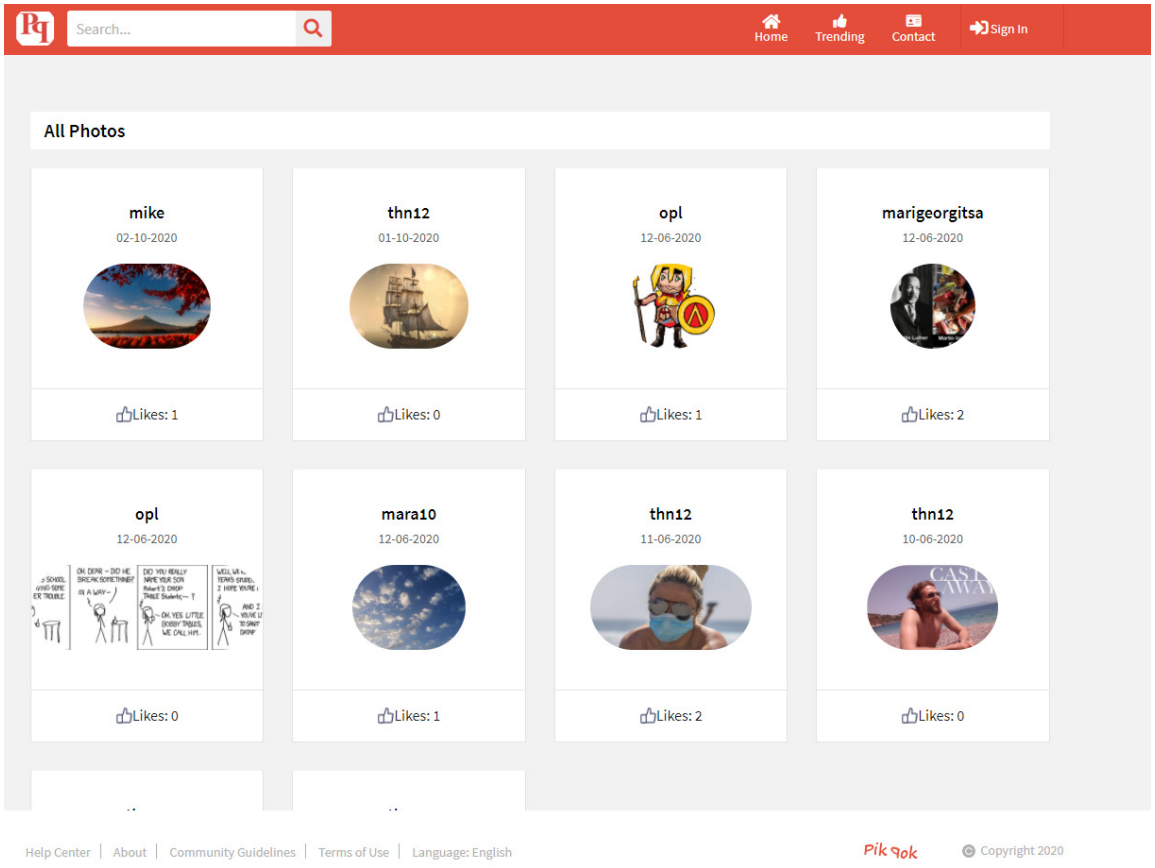
1. Να εκφράζει την προτίμησή του (δηλαδή να κάνει “like”) σε μία υπάρχουσα φωτογραφία του κοινωνικού δικτύου.
2. Να σχολιάζει μία υπάρχουσα φωτογραφία του κοινωνικού δικτύου (δηλαδή να μπορεί να προσθέσει κάποιο comment σε μία υπάρχουσα φωτογραφία).
3. Μέσω της κεντρικής σελίδας (My-profile) του λογαριασμού του:
    1. να καταχωρήσει (να κάνει upload) κάποια φωτογραφία στο Πικ-Ποκ.
    2. να προσθέσει μία ή περισσότερες ετικέτες (tags) στη φωτογραφία που κάνει upload, με σκοπό να υπάρχει στο τελικό μενού και δυνατότητα αναζήτησης φωτογραφίας, βάσει tag
    3. να προβάλλει όλες τις φωτογραφίες που έχει καταχωρήσει στο παρελθόν στο Πικ-Ποκ.
    4. να διαγράψει κάποια από τις φωτογραφίες που έχει καταχωρήσει στο παρελθόν στο Πικ-Ποκ.
    5. να δει τις δημοσιεύσεις που έχει αποθηκεύσει, τις δημοσιεύσεις του με τα περισσότερα likes καθώς και τις 20 τελευταίες δημοσιεύσεις που έχει κάνει στο κοινωνικό δίκτυο.
4. Να κάνει νέες φιλίες. Στο πλαίσιο αυτό μπορεί να αναζητήσει νέα άτομα με βάση κριτήρια όπως το ονοματεπώνυμο τους. Η αναζήτηση λειτουργεί με παρόμοιο τρόπο όπως και η αναζήτηση μέσω tags, ταιριάζοντας δηλαδή το προς αναζήτηση ονοματεπώνυμο π.χ. με οποιοδήποτε μέρος των καταχωρημένων στη βάση στοιχείων. Π.χ., κάνοντας αναζήτηση για “geor” επιστρέφονται όλες οι φωτογραφίες, στων οποίων τα tags περιέχεται η λέξη “geor”, π.χ. “Georgopoulos Hlias”, “Georgopoulou Maria”, “Georgia Kanellou”, κ.τ.λ.
5. Να συνομιλήσει με άλλους χρήστες (chatting). Για λόγους ασφαλείας που θα εξηγηθούν σε επόμενο μέρος της εργασίας, ο χρήστης μπορεί να συνομιλήσει μόνο με χρήστες οι οποίοι, έχουν δεχτεί το αίτημα φιλίας του και συνεπώς είναι φίλοι του.
6. Υπάρχει μία λειτουργία αποθήκευσης της δημοσίευσης στις αγαπημένες φωτογραφίες του χρήστη, σαν «σελιδοδείκτης», έτσι ώστε να υπάρχει δυνατότητα άμεσης εύρεσης και επαναπροβολής μίας δημοσίευσης μέσω του κεντρικού μενού στην προσωπική του σελίδα (MyProfile).

7. Δυνατότητα διαμοιρασμού. Ο χρήστης μπορεί να διαμοιραστεί τη φωτογραφία με άλλους χρήστες. Για τη διευκόλυνση της αλληλεπίδρασης των χρηστών, είναι δυνατό να παρέχονται πολλαπλοί τρόποι πραγματοποίησης της λειτουργίας, ενδεικτικά (α) αντιγράφοντας τον σύνδεσμο που βρίσκεται σε αυτή, πατώντας ένα κουμπί «Αντιγραφή» και (β) σκανάροντας τον κωδικό QR που στην ουσία περιέχει τον σύνδεσμο της εκάστοτε δημοσίευσης. Αυτή η λειτουργία διευκολύνει τον διαμοιρασμό δημοσιεύσεων από κινητές συσκευές.

Στις προηγούμενες παραγράφους έγινε μία περιληπτική αναφορά στις σημαντικότερες λειτουργίες της εφαρμογής. Να τονιστεί βέβαια ότι με την πάροδο των εκδόσεων, ενδέχεται κάποιες λειτουργίες να αναβαθμιστούν, να διαγραφούν αν δεν είναι χρήσιμες, ή να προστεθούν και νέες, αλλά οι αναφερθείσες αποτελούν το βασικό σύνολο λειτουργιών του κοινωνικού μας δικτύου και έτσι παρατίθενται ρητώς καθώς πρέπει να αναφερθούν. Στη συνέχεια του κεφαλαίου γίνεται και μία πληρέστερη ανάλυση αυτών των λειτουργιών, καθώς και μία παρουσίαση του σχεδιασμού της διεπαφής.

2.1.1 *Αρχική Οθόνη μη συνδεδεμένου χρήστη*

Όταν ο χρήστης επισκεφτεί το σύνδεσμο του PikPok, η πρώτη σελίδα που εμφανίζεται είναι αυτή που φαίνεται στην Εικόνα 2. Ο χρήστης έχει κάποιες δυνατότητες στο κοινωνικό δίκτυο ακόμη και αν δεν έχει δημιουργήσει λογαριασμό. Μπορεί να δει εικόνες που έχουν δημοσιευτεί από άλλους χρήστες και να προβάλει προφίλ άλλων χρηστών αλλά δεν έχει την δυνατότητα να προβεί σε ενέργειες δημοσίευσης νέου περιεχομένου, σχολιασμού, like ή ανταλλαγής μηνυμάτων. Φυσικά δεν μπορεί και να αλλάξει στοιχεία στο προφίλ του ή να κάνει φίλους αφού δεν διαθέτει κάποιο λογαριασμό.



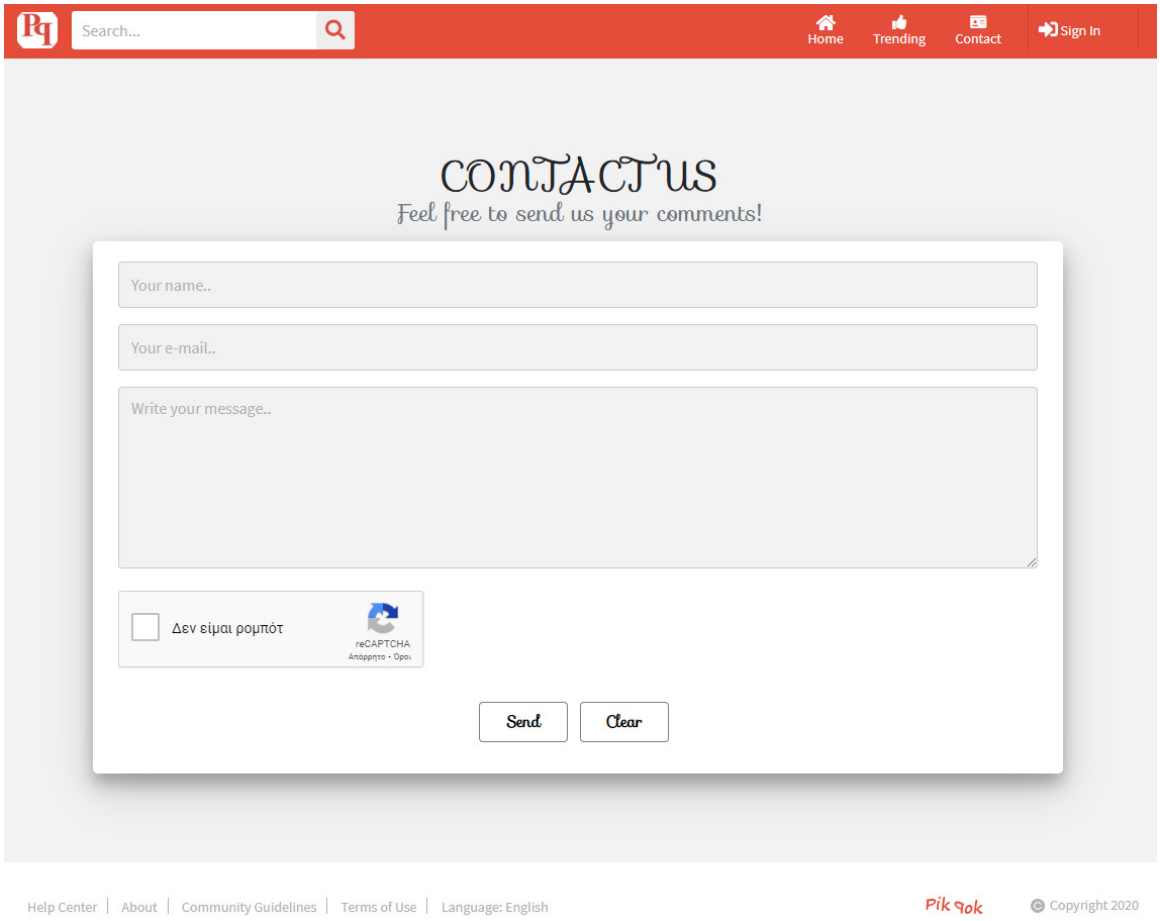
Εικόνα 2. Αρχική οθόνη επισκέπτη χρήστη

Αναλυτικά, οι λειτουργίες που μπορεί να αξιοποιήσει ένας χρήστης όταν χρησιμοποιεί το κοινωνικό δίκτυο χωρίς να διαθέτει λογαριασμό έχουν ως εξής:

Αρχική Σελίδα	**Περιγραφή**
	Στην αρχική οθόνη χωρίς ο χρήστης να διαθέτει λογαριασμό μπορεί να δει στην κατηγορία All Photos, όλες τις φωτογραφίες που έχουν κοινοποιηθεί στην εφαρμογή Pik Pok από υπάρχοντες χρήστες. Οι φωτογραφίες εμφανίζονται ταξινομημένες με την ημερομηνία που έχουν δημοσιευθεί (από την πιο πρόσφατη έως τη λιγότερο πρόσφατη). Στη κάθε φωτογραφία αναφέρεται το όνομα του χρήστη που την ανέβασε, η ημερομηνία καθώς και το πλήθος των likes που έχει.
	Στη καρτέλα *Popular* ο χρήστης βλέπει και πάλι όλες τις φωτογραφίες που έχουν δημοσιευθεί, με τη μόνη διαφορά ότι σε αυτήν τη σελίδα είναι ταξινομημένες βάσει των likes (φθίνουσα σειρά).
	Στη καρτέλα *Contact* είναι διαθέσιμη μια φόρμα επικοινωνίας με τους δημιουργούς της εφαρμογής για τυχόν ερωτήσεις και προβλήματα. Η φόρμα έχει προστασία έναντι αυτοματοποιημένων αποστολών μηνυμάτων (bots).

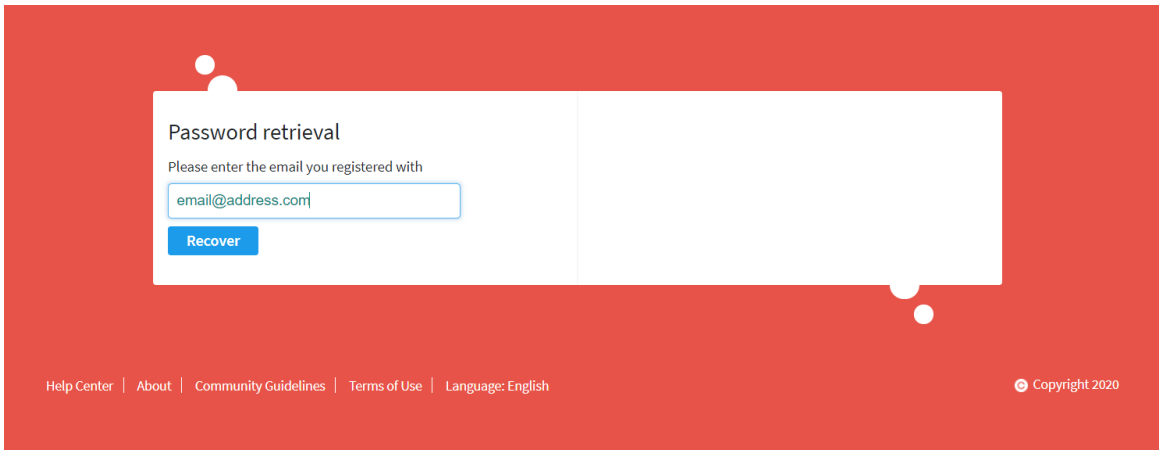
	Μέσω της καρτέλας Sign in, ένας χρήστης που διαθέτει ήδη λογαριασμό μπορεί να συνδεθεί εισάγοντας τα προσωπικά του στοιχεία. Διατίθεται και η λειτουργία *Forgot Password* για την περίπτωση που ο χρήστης ξεχάσει ή χάσει το κωδικό του. Σε αυτήν την περίπτωση του αποστέλλεται ένας νέος τυχαίος κωδικός στην ήδη δηλωμένη διεύθυνσή του. Στην ίδια καρτέλα υπάρχει και η επιλογή Sign up στην οποία μπορεί να δημιουργηθεί ένας καινούριος λογαριασμός χρήστη. Τα πεδία First Name, Surname, Date Of Birth, Gender, email, username, Password, Repeat Password και το κουμπί Yes, I understand and agree to the pik pok Terms & Conditions είναι απαραίτητα (required fields) ώστε να προχωρήσει η εφαρμογή στη δημιουργία νέου χρήστη. Επίσης γίνεται έλεγχος για το αν έχει ήδη δημιουργηθεί λογαριασμός για το ίδιο email ή το ίδιο username, ενώ παράλληλα έχει ενσωματωθεί και προστασία έναντι των αυτοματοποιημένων υποβολών. Εσωτερικά, ο κωδικός που καταχωρείται στη βάση κρυπτογραφείται με τον αλγόριθμο sha256, για περισσότερη ασφάλεια και γίνονται και οι απαραίτητοι έλεγχοι πιθανών επικίνδυνων χαρακτήρων σε όλα τα fields ώστε να εμποδίζονται τυχόν sql injection και άλλου τέτοιου είδους επιθέσεις.
	Ο χρήστης έχει τη δυνατότητα αναζήτησης φωτογραφιών βάσει tags.

Όπως αναφέρθηκε ανωτέρω, στη καρτέλα *Contact* διατίθεται μια φόρμα επικοινωνίας με τους δημιουργούς της εφαρμογής για τυχόν ερωτήσεις και προβλήματα. Η φόρμα επιτρέπει στον χρήστη να εισάγει το όνομά του, το email του και το μήνυμα που θέλει να στείλει. Για λόγους ασφαλείας, θα πρέπει ο χρήστης οπωσδήποτε να πατήσει στο κουμπί - reCaptcha στο οποίο αναγράφεται η πρόταση “I’m not a robot”, έτσι ώστε να γίνει ο κατάλληλος έλεγχος για το αν ο χρήστης είναι άνθρωπος. Ο μηχανισμός αυτός είναι αρκετά δημοφιλής πλέον και εύκολος στην χρήση, οπότε ταυτόχρονα δεν επιδρά αρνητικά στην εμπειρία που έχει ο χρήστης στο κοινωνικό δίκτυο. Ταυτόχρονα, γίνονται έλεγχοι και sanitization σε όλα τα πεδία της φόρμας. Θα αναφερθούμε περισσότερο στο κομμάτι της ασφάλειας στα επόμενα κεφάλαια.

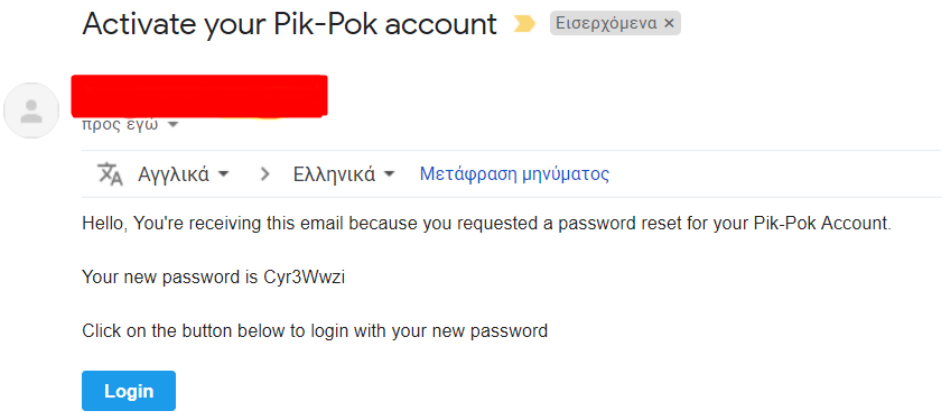


Εικόνα 3. Λειτουργία Contact

Μέσω της καρτέλας Sign in, μπορεί ένας χρήστης που διαθέτει ήδη λογαριασμό να συνδεθεί, εισάγοντας τα προσωπικά του στοιχεία. Διατίθεται και η λειτουργία Forgot Password για την περίπτωση που ο χρήστης έχει απωλέσει τον κωδικό του. Σε αυτήν την περίπτωση του αποστέλλεται ένας νέος τυχαίος κωδικός (8 ψηφίων) στο ισχύον email του χρήστη.



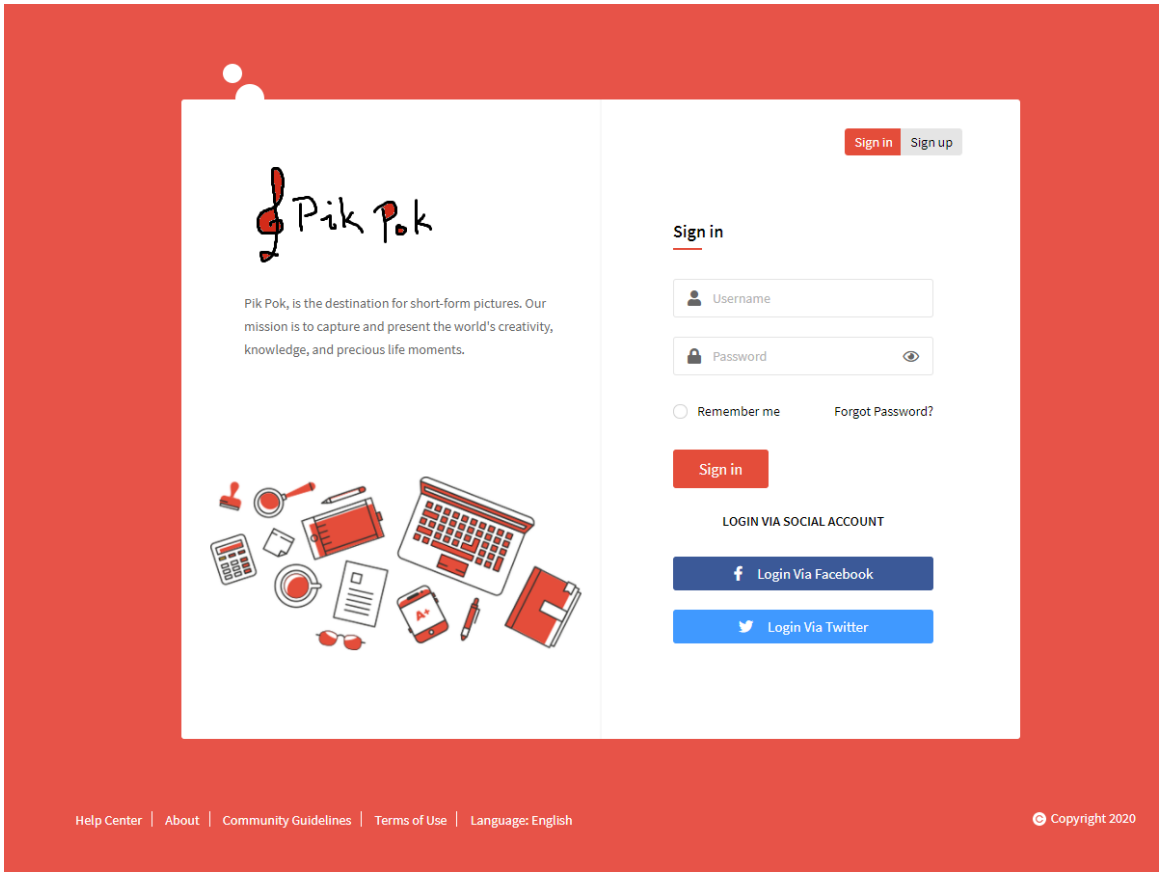
Εικόνα 4. Οθόνη Υπενθύμισης Κωδικού



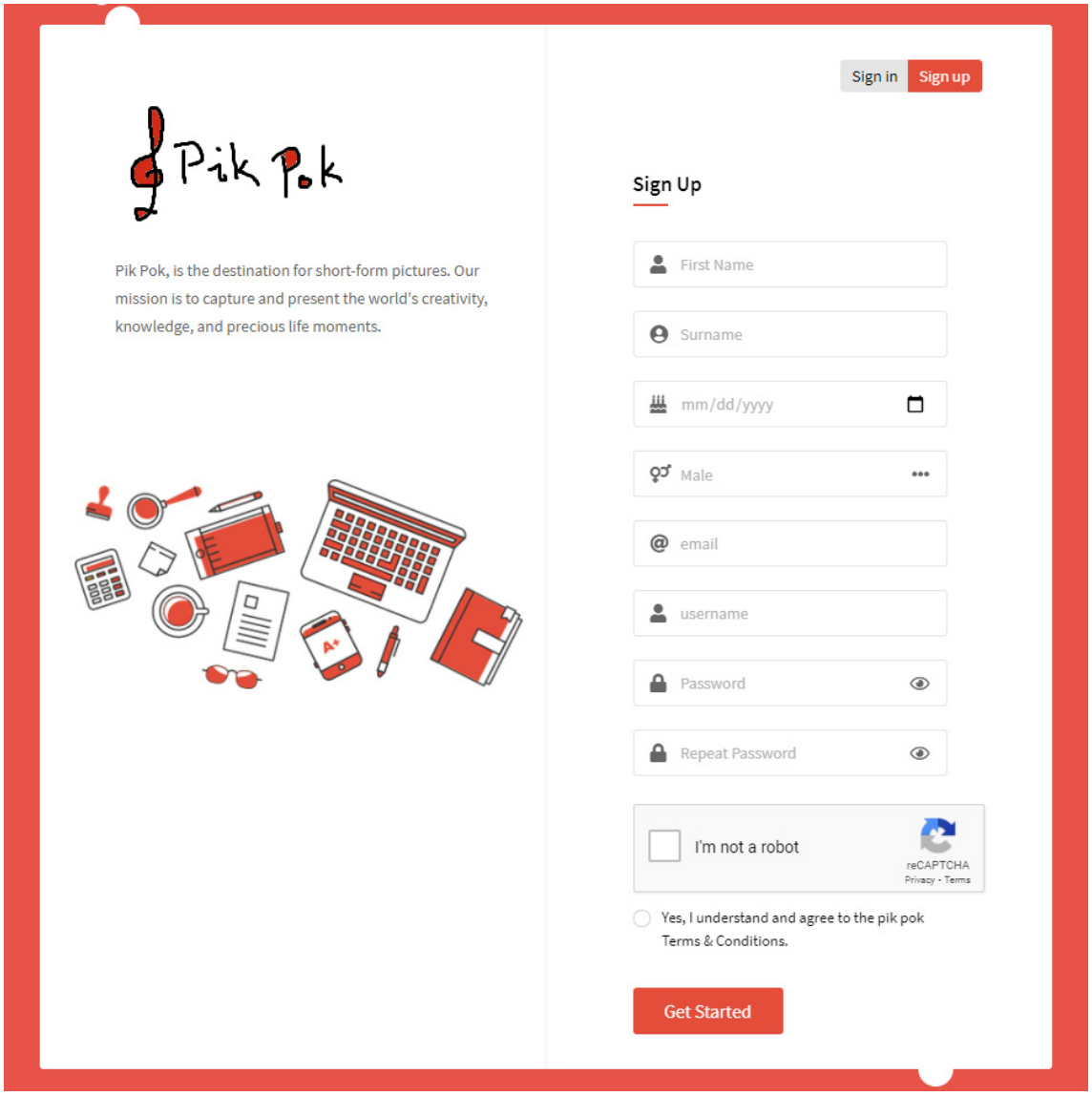
Εικόνα 5. Email αλλαγής κωδικού

Στην ίδια καρτέλα υπάρχει και η επιλογή Sign up στην οποία μπορεί να δημιουργηθεί ένας καινούριος λογαριασμός χρήστη. Τα πεδία First Name, Surname, Date Of Birth, Gender, email, username, Password, Repeat Password και το κουμπί Yes, I understand and agree to the pik pok Terms & Conditions είναι απαραίτητα (required fields) ώστε να προχωρήσει η εφαρμογή στη δημιουργία νέου χρήστη. Επίσης γίνεται έλεγχος για το αν υπάρχει ήδη λογαριασμός με το ίδιο email ή το ίδιο username και υπάρχει και έλεγχος για αποφυγή αυτοματοποιημένων υποβολών. Ο κωδικός που καταχωρείται στη βάση κρυπτογραφείται με τον αλγόριθμο SHA256, για περισσότερη ασφάλεια και γίνονται και οι απαραίτητοι έλεγχοι πιθανώς επικίνδυνων χαρακτήρων σε όλα τα πεδία, ώστε να εμποδίζονται τυχόν SQL Injection και άλλες επιθέσεις παρόμοιου είδους. Ταυτόχρονα γίνονται και οι κατάλληλοι έλεγχοι για το αν το email του χρήστη είναι στην κατάλληλη μορφή (something@email.com). Επίσης ελέγχεται με χρήση regular expression, αν ο κωδικός του χρήστη είναι ισχυρός. Ο κωδικός θα πρέπει να έχει τουλάχιστον μικρά και κεφαλαία γράμματα, αριθμούς και ελάχιστο μήκος 7 χαρακτήρων για να είναι έστω επιτρεπτός κωδικός, χωρίς βέβαια αυτοί οι έλεγχοι να καλύπτουν πλήρως το ασφαλές του κωδικού. Προτείνεται ειδικότερα η χρήση ειδικών χαρακτήρων, εναλλαγή πεζών-κεφαλαίων γραμμάτων και μήκος πάνω από 12 χαρακτήρες. Τέλος, γίνεται έλεγχος για την ηλικία του χρήστη, ο οποίος θα πρέπει να είναι απαραιτήτως μεγαλύτερος των 16 ετών για να γίνει μέρος του κοινωνικού δικτύου, έλεγχος ο οποίος όπως παρατηρήσαμε παραλείπεται σε άλλες μεγάλες εφαρμογές κοινωνικής δικτύωσης, κάτι που μπορεί να προκαλέσει σοβαρά ζητήματα ιδιωτικότητας, νομοθεσίας αλλά και ηθικής. Αυτός ο έλεγχος επίσης δεν καλύπτει πλήρως όλες τις περιπτώσεις, π.χ. αν ο χρήστης δηλώσει ψευδή ημερομηνία γέννησης. Μία προσέγγιση, που εν μέρει λύνει αυτό το πρόβλημα είναι η χρήση του μηχανισμού αναγνώρισης προσώπων που έχουμε υλοποιήσει, για την αναγνώριση ηλικιών στις φωτογραφίες που δημοσιεύονται. Όταν αυτή η ηλικία είναι κατά σημαντική πιθανότητα μικρότερη των 16, θα μπορούσαν να διερευνώνται σε μεγαλύτερο βάθος αυτές οι περιπτώσεις και ενδεχομένως να υπήρξε και μία διαδικασία απαγόρευσης εισόδου αυτών των χρηστών στην εφαρμογή. Επιπροσθέτως, μία έρευνα (14) ισχυριζόταν ότι μπορούν να εξαχθούν αποτελέσματα για την ηλικία ενός χρήστη μόνο αναλύοντας κάποια βιομετρικά χαρακτηριστικά. Συγκεκριμένα σύμφωνα με τους ερευνητές αναλύοντας τα αυτιά των χρηστών, μπορούμε να εξάγουμε αρκετά ακριβή συμπεράσματα για την ηλικία των χρηστών αυτών, καθώς δημοφιλή ανθρώπινα χαρακτηριστικά όπως το πρόσωπο και τα δακτυλικά αποτυπώματα μπορούν να τροποποιηθούν ή να αλλάξουν με

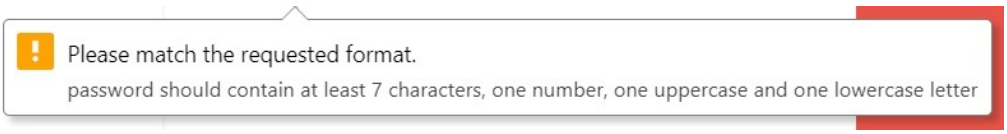
το χρόνο, σύμφωνα με τους ερευνητές. Ωστόσο, το αυτί έχει μια σταθερή δομή που δεν αλλάζει με το χρόνο και έχει μοναδικά χαρακτηριστικά που ικανοποιούν τις απαιτήσεις ενός βιομετρικού χαρακτηριστικού. Αυτή η έρευνα πρότεινε ένα μοντέλο για τον προσδιορισμό της ανθρώπινης ηλικίας και την κατηγοριοποίηση της σε δύο ομάδες, την ομάδα άνω των 18 και την ομάδα κάτω των 18 ετών. Η κατηγοριοποίηση αυτή έγινε ερευνώντας μόνο το σχήμα του αυτιού, χρησιμοποιώντας ένα συνελικτικό νευρωνικό δίκτυο (convolutional neural network - CNN), ενώ για να γίνει το μοντέλο πιο ακριβές εφαρμόζονται και οι κατάλληλες συναρτήσεις ενεργοποίησης. Το προτεινόμενο μοντέλο έφτασε σε εξαιρετική ακρίβεια 98,75%, χρησιμοποιώντας για την εκπαίδευση δεδομένα από τις βάσεις - datasets AMI (15), AWE (15) καθώς και προσωπικές φωτογραφίες παιδιών. Λόγω της υψηλής ακρίβειας του εντοπισμού των ηλικιακών ομάδων, το μοντέλο μπορεί σύμφωνα με τους ερευνητές να εφαρμοστεί σε πλατφόρμες κοινωνικών μέσων για έλεγχο της ταυτότητας των πληροφοριών του χρήστη. (14)



Εικόνα 6. Εικόνα 5: Λειτουργία σύνδεσης χρήστη (Sign in)

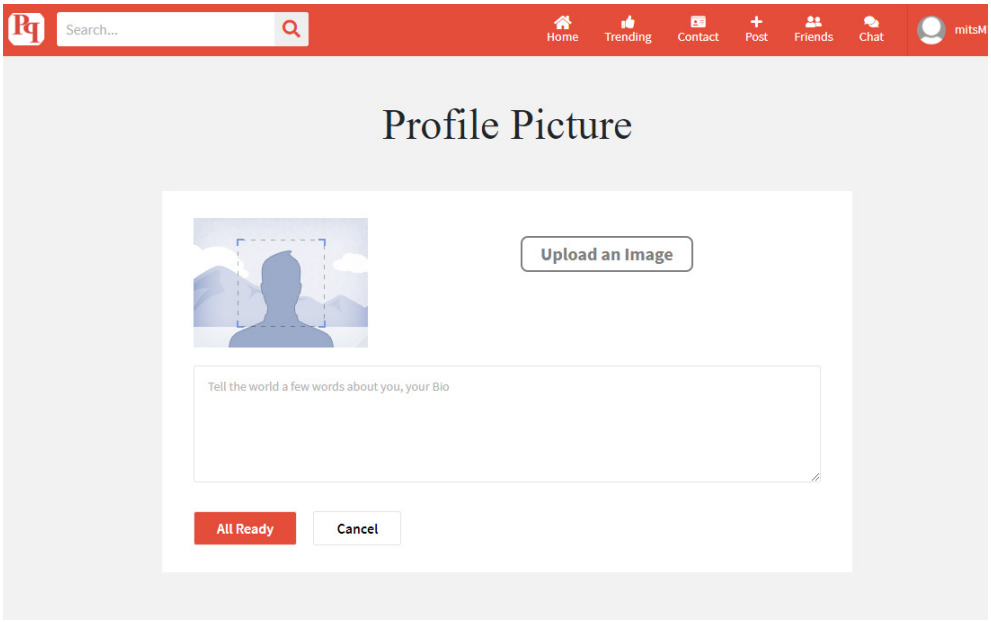


Εικόνα 7. Λειτουργία εγγραφής νέου χρήστη (Sign up)

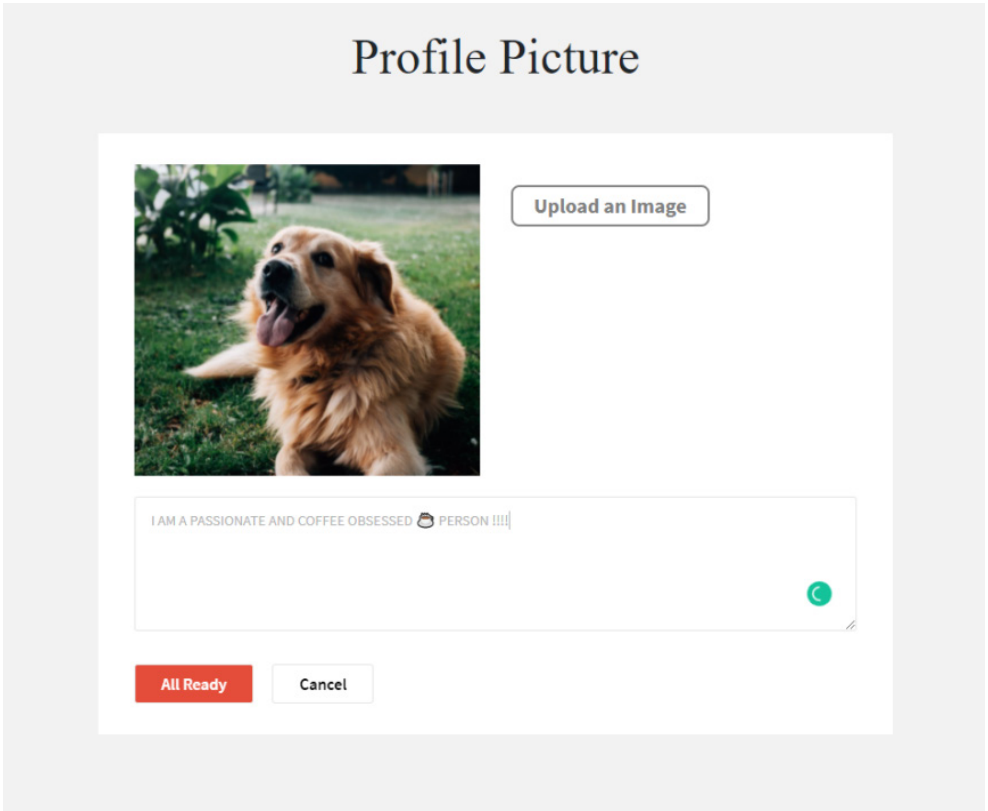


Εικόνα 8. Ενδεικτικό μήνυμα ειδοποίησης για έλλειψη ασφάλειας κωδικού

Στο επόμενο βήμα μετά την εγγραφή, ο χρήστης οδηγείται σε μία σελίδα, όπου μπορεί να συμπληρώσει το προφίλ του, ανεβάζοντας την προσωπική του φωτογραφία και ένα μικρό κείμενο για αυτόν (short bio). Αν και παροτρύνουμε τους χρήστες να ολοκληρώσουν την διαδικασία, αυτά τα στοιχεία δεν είναι υποχρεωτικά και ο χρήστης μπορεί να παραλείψει ("skip") αυτό το στάδιο, αν δεν επιθυμεί να εισάγει ακόμη τα συγκεκριμένα στοιχεία. Αν παραλείψει συνεπώς αυτήν τη φάση, το avatar του συνεχίζει να έχει την προκαθορισμένη (default) φωτογραφία και δεν διαθέτει κάποιο σύντομο βιογραφικό κείμενο.



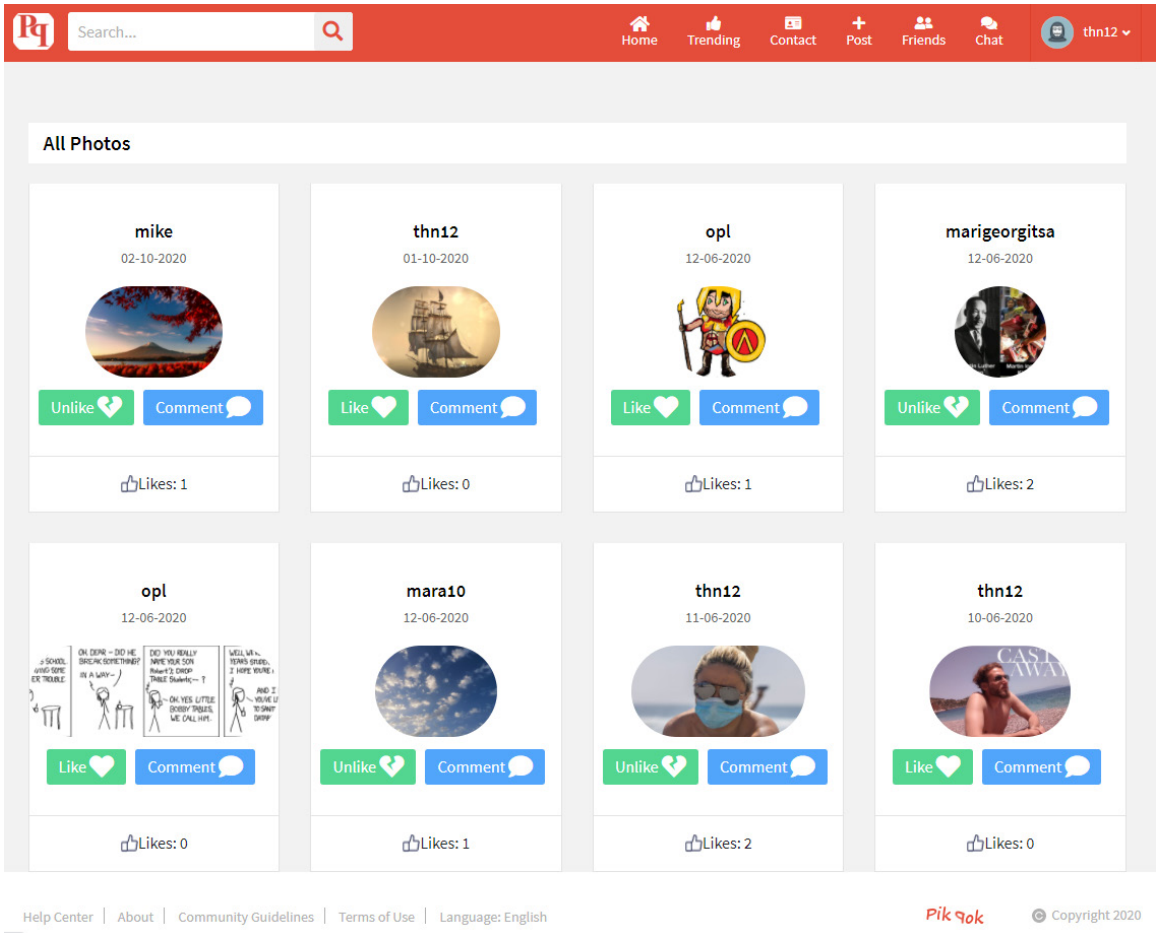
Εικόνα 9. Οθόνη εισαγωγής φωτογραφίας και σύντομου βιογραφικού



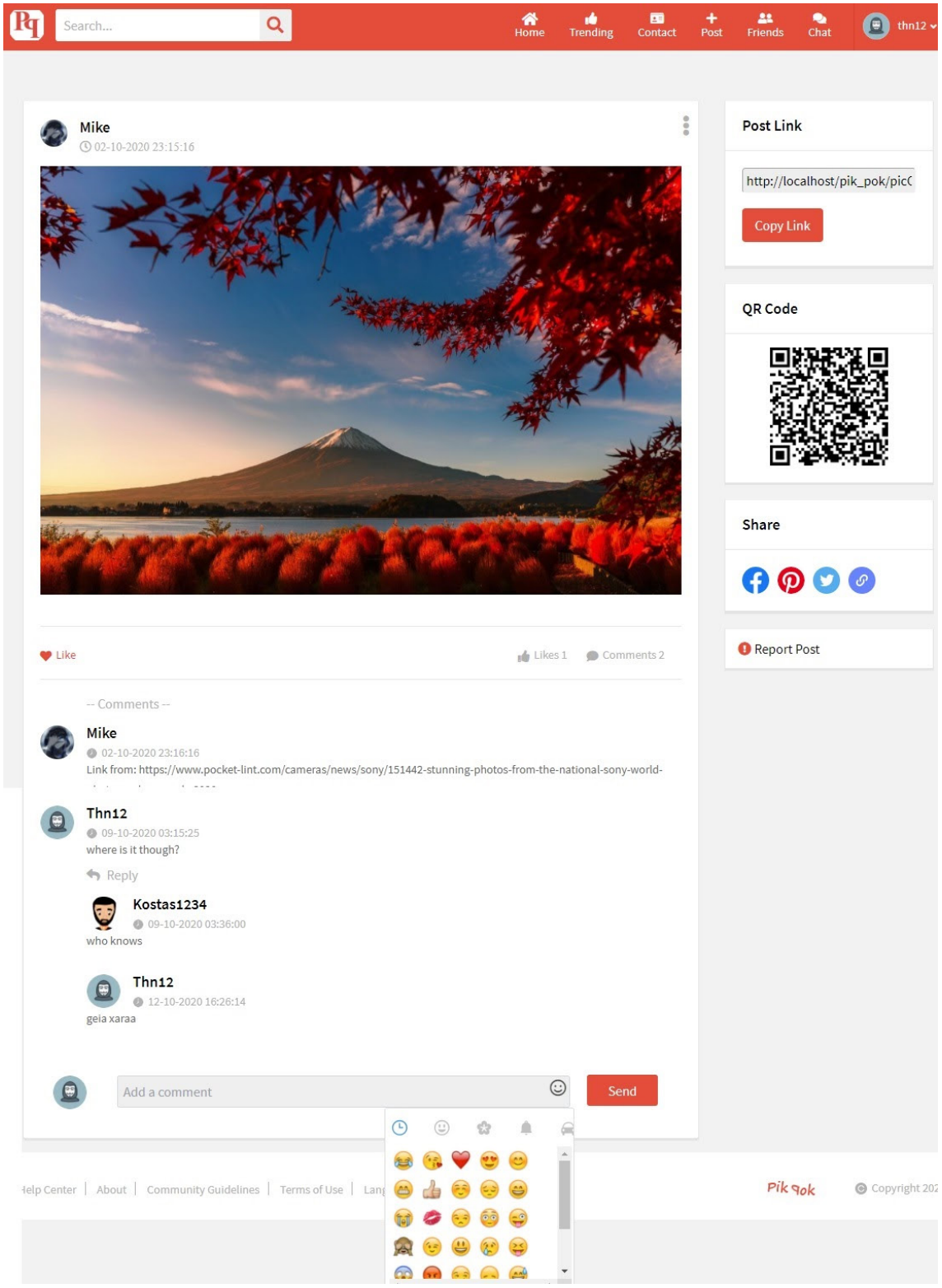
Εικόνα 10. Μετά από την εισαγωγή φωτογραφίας

2.1.2 Αρχική οθόνη συνδεδεμένου χρήστη

Εάν ο χρήστης διαθέτει λογαριασμό στο Pik Pok και πραγματοποιήσει σύνδεση στην εφαρμογή μέσω του κουμπιού “SignIn”, ανακατευθύνεται αυτόματα στην αρχική οθόνη, όπου μπορεί να δει όλες τις φωτογραφίες που έβλεπε και ως μη συνδεδεμένος χρήστης, με τη μόνη διαφορά ότι τώρα είναι διαθέσιμες οι λειτουργίες Like και Unlike (η λειτουργία “unlike” επιτρέπει στον χρήστη να αναιρέσει την έκφραση αρέσκειας, αν τυχόν αλλάξει γνώμη σχετικά με το like του σε κάποια φωτογραφία). Επίσης μπορεί να εισάγει και κάποιο σχόλιο χρησιμοποιώντας τη λειτουργία «Comment», η οποία τον οδηγεί σε μία σελίδα η οποία περιέχει τη συγκεκριμένη φωτογραφία σε πλήρες μέγεθος, τα comments και τα likes που έχει αυτή η φωτογραφία και έναν χώρο στο κάτω μέρος, όπου ο χρήστης μπορεί να προσθέσει το δικό του σχόλιο και τα δικά του emoticons. Σε αυτή τη σελίδα υπάρχει και λειτουργία αποθήκευσης της δημοσίευσης στις αγαπημένες του φωτογραφίες, έτσι ώστε να υπάρχει δυνατότητα άμεσης εύρεσης και επαναπροβολής μέσω του κεντρικού μενού στην προσωπική του σελίδα (MyProfile). Μπορεί επίσης να διαμοιραστεί τη φωτογραφία, είτε αντιγράφοντας τον σύνδεσμο που βρίσκεται αυτή ή ακόμα και σκανάροντας τον QR code που στην ουσία περιέχει τον σύνδεσμο της εκάστοτε δημοσίευσης.



Εικόνα 11. Οθόνη συνδεδεμένου χρήστη (με δυνατότητες Like/Unlike,Comment κ.λπ.).



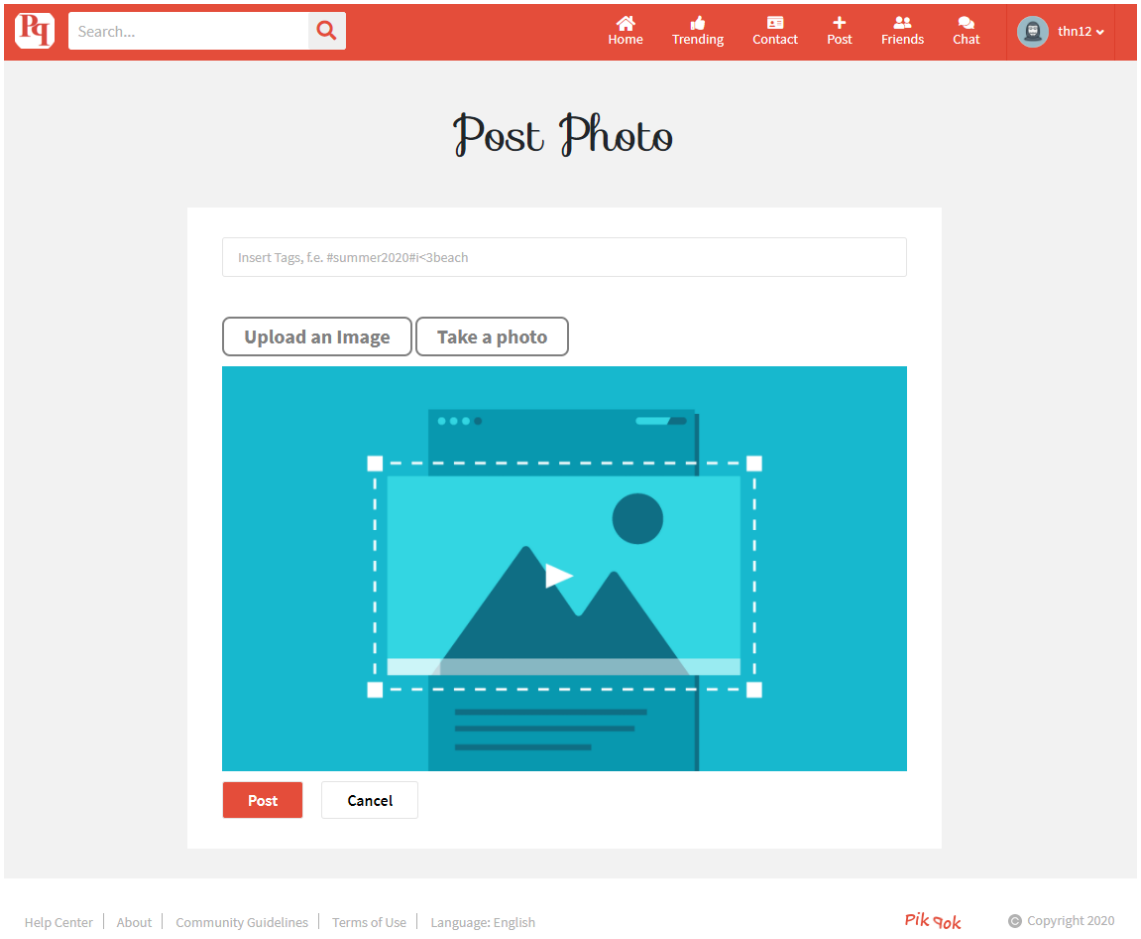
Εικόνα 12. Λειτουργία Σχολιασμού (Comment)

2.1.3 Δημοφιλέστερες δημοσιεύσεις (Most Popular)

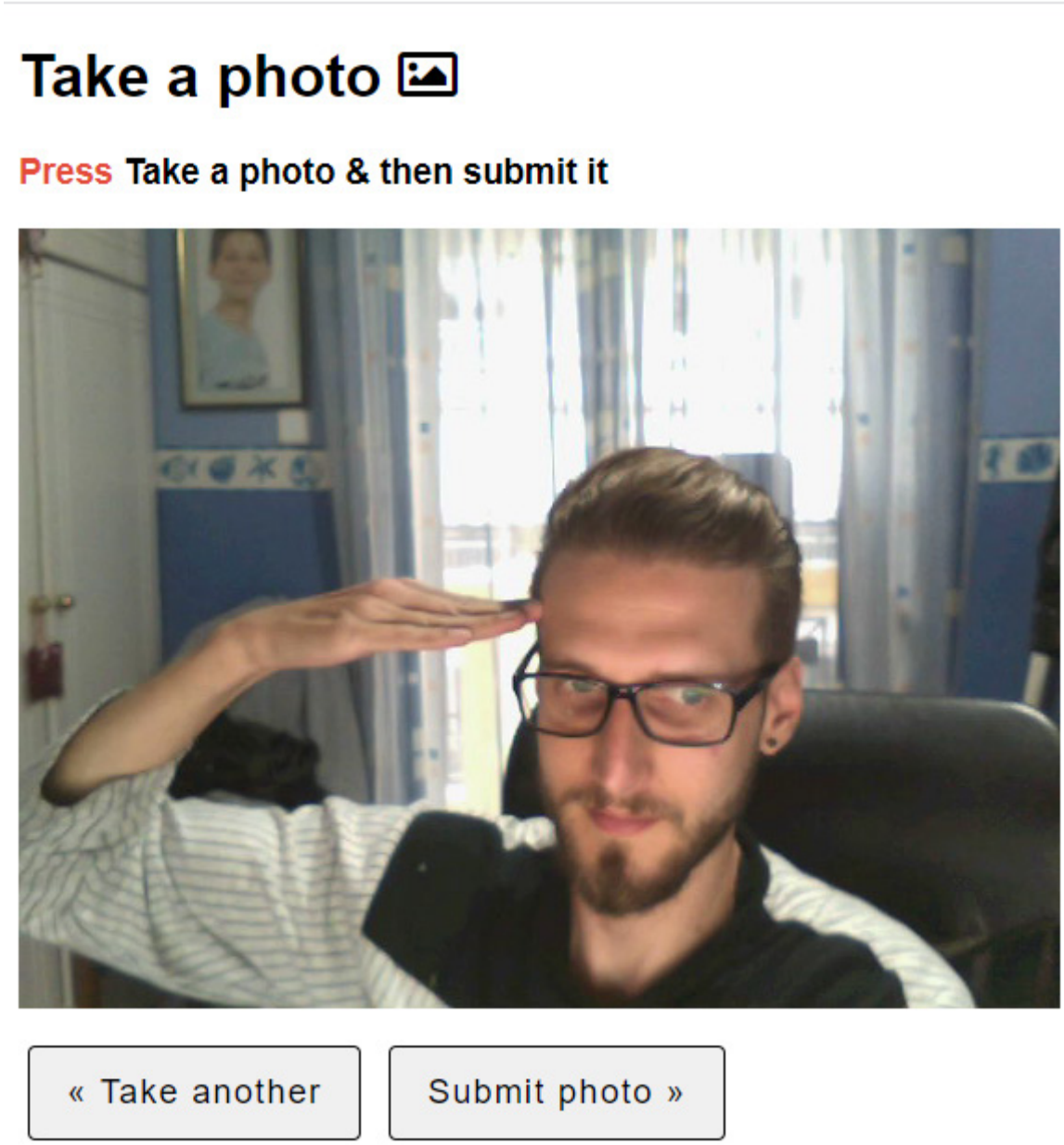
Στη καρτέλα Popular, ο χρήστης βλέπει και πάλι όλες τις φωτογραφίες που έχουν δημοσιευθεί, ταξινομημένες βάσει των likes, σε φθίνουσα σειρά. Μέσω χρήσης Ajax αλλάζει η ταξινόμηση σε πραγματικό χρόνο αν πατηθεί Like ή Unlike.

2.1.4 Δημοσίευση νέας φωτογραφίας

Ο χρήστης έχει τη δυνατότητα να ανεβάσει μια φωτογραφία της επιλογής του από τα τοπικά του αρχεία ή και ακόμα να χρησιμοποιήσει την κάμερα ώστε να βγάλει μια νέα φωτογραφία. Αν επιθυμεί, μπορεί να προσθέσει μία ή περισσότερες ετικέτες (tags) στην εικόνα που ανεβάζει, με σκοπό να δώσει πληροφορίες για το περιεχόμενο της. Οι πληροφορίες αυτές αξιοποιούνται στη συνέχεια στη λειτουργία της αναζήτησης βάσει ετικετών. Η λειτουργία αυτή είναι προσβάσιμη και μέσω της σελίδας του προφίλ χρήστη (My Profile).



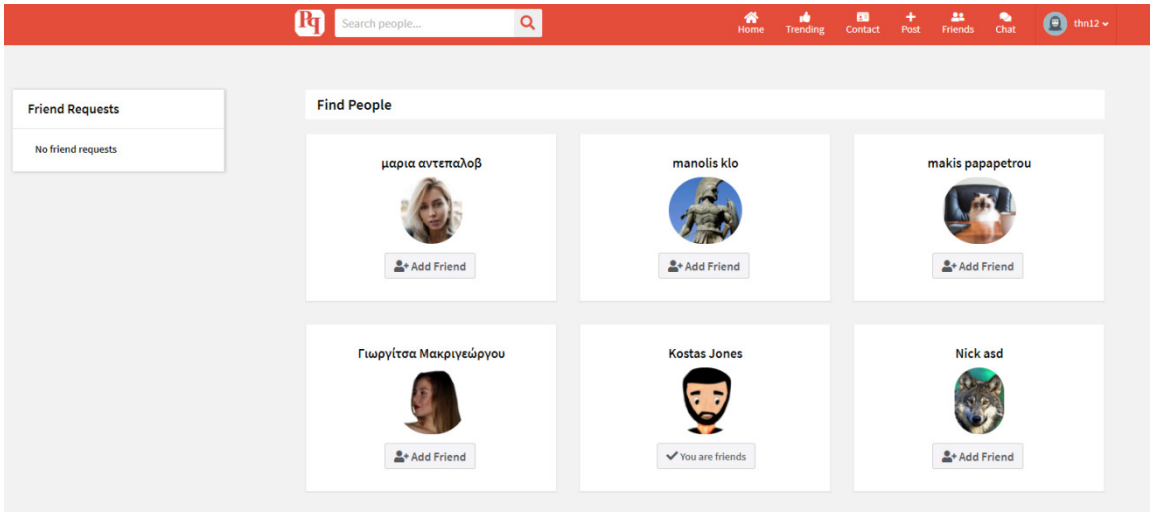
Εικόνα 13. Λειτουργία Δημοσίευσης (Post Photo)



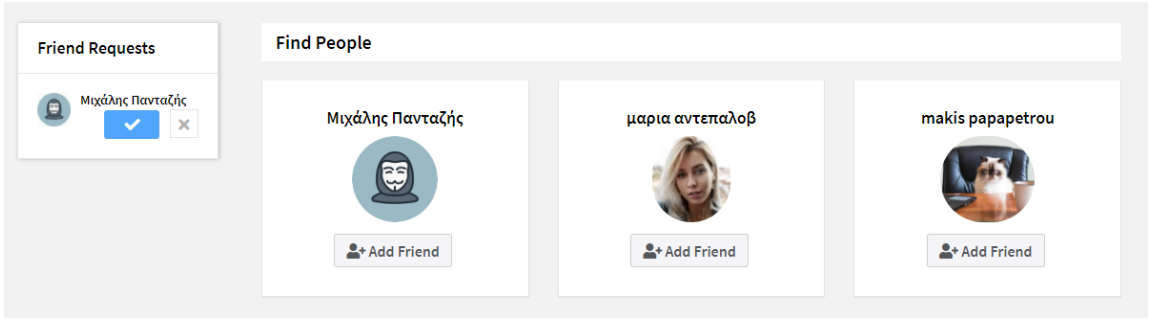
Εικόνα 14. Λειτουργία λήψης φωτογραφίας (Take a photo)

2.1.5 Σύναψη Φιλιών

Πέρα από τη λειτουργία ανάρτησης φωτογραφίας είναι διαθέσιμη η καρτέλα *friends* η οποία εμφανίζει υπάρχοντες χρήστες και παρέχει πρόσβαση στις λειτουργίες διαχείρισης κοινωνικών σχέσεων εντός του δικτύου. Πιο αναλυτικά, παρέχει πρόσβαση στη δυνατότητα αποστολής αιτήματος φιλίας σε χρήστες καθώς και προσθήκης χρηστών στους φίλους μας. Στην αριστερή πλευρά της καρτέλας παρουσιάζονται και τυχόν εκκρεμή αιτήματα φιλίας που έχουμε λάβει. Παρέχεται επίσης η δυνατότητα αναζήτησης χρηστών με βάση κριτήρια όπως το ονοματεπώνυμο τους. Η αναζήτηση λειτουργεί με παρόμοιο τρόπο όπως και η αναζήτηση μέσω ετικετών (tags), ταιριάζοντας δηλαδή το προς αναζήτηση ονοματεπώνυμο π.χ. με οποιοδήποτε μέρος των καταχωρημένων στη βάση στοιχείων. Για παράδειγμα, κάνοντας αναζήτηση για “geor” επιστρέφονται όλοι οι χρήστες, στων οποίων τα στοιχεία περιέχεται η συμβολοσειρά “geor”, π.χ. “Georgopoulos Hlias”, “Georgopoulou Maria”, “Georgia Kanellou”, κ.τ.λ.



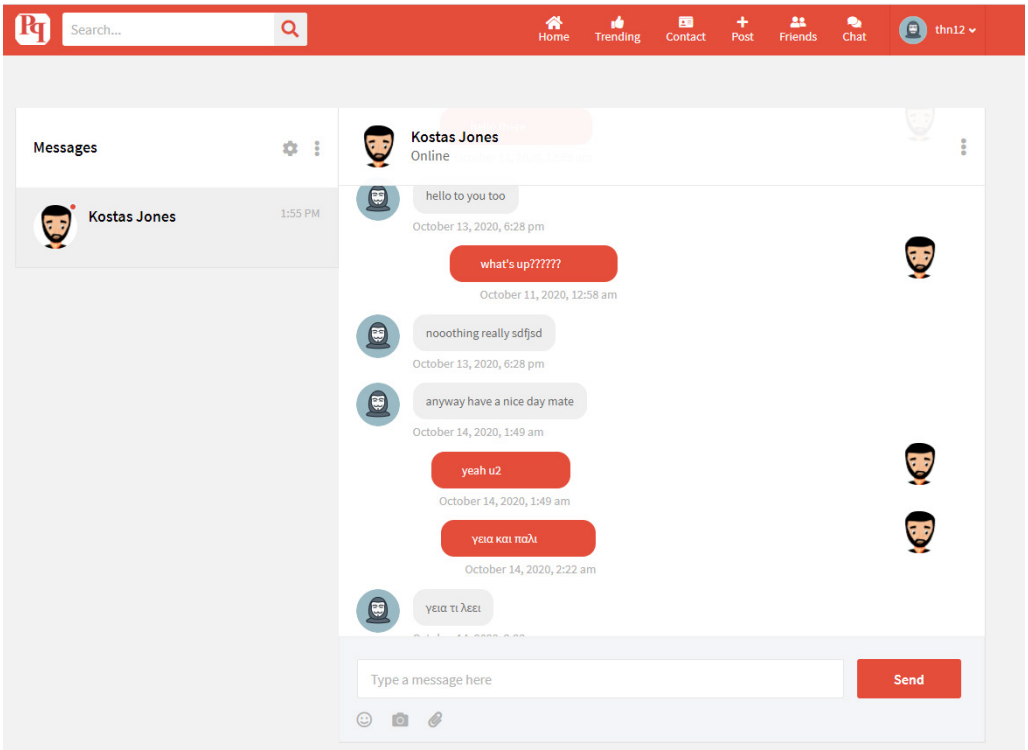
Εικόνα 15. Λειτουργία Σύναψης Φιλιών (Friend Requests)



Εικόνα 16. Δυνατότητα Αποδοχής/Απόρριψης αιτημάτων φιλίας

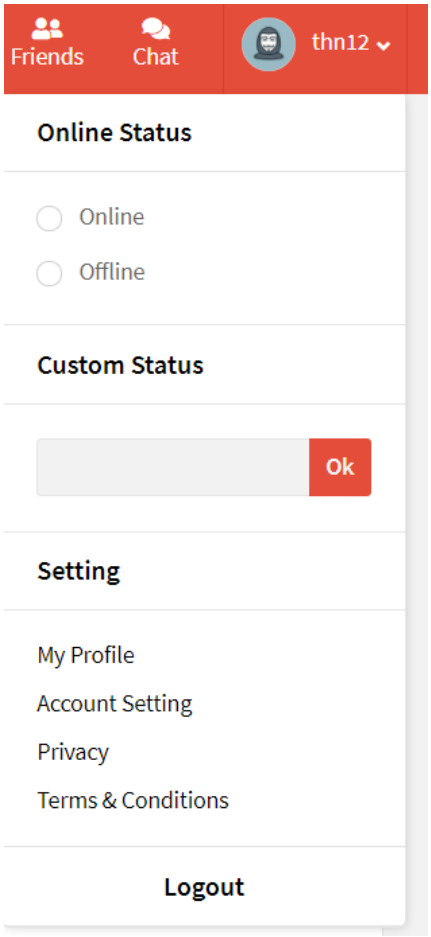
2.1.6 Συνομιλία με άλλους χρήστες

Μία πρόσθετη λειτουργία στο πλαίσιο της κοινωνικής δικτύωσης είναι η άμεση συνομιλία (chat). Η λειτουργία αυτή υλοποιείται σε μία ξεχωριστή καρτέλα, όπου μπορούμε να ανταλλάξουμε μηνύματα με κάποιον χρήστη που ήδη περιλαμβάνεται στους φίλους μας. Ο περιορισμός της άμεσης συνομιλίας στον κύκλο των φίλων μας έχει εισαχθεί για λόγους ιδιωτικότητας και ασφάλειας, καθώς η παροχή δυνατότητας επικοινωνίας με κάθε χρήστη του κοινωνικού δικτύου χωρίς κανέναν περιορισμό, θα ενείχε κινδύνους για την ιδιωτικότητα και την ασφάλεια των χρηστών, αφού αυτοί θα αναγκάζονταν να δέχονται μηνύματα από άτομα που δεν γνωρίζουν, τα οποία θα μπορούσαν να είναι απρεπή, προσβλητικά, υπερβολικά σε πλήθος/συχνότητα (και άρα ενοχλητικά) κ.λπ. Θα ήταν επίσης δυνατόν μέσω της μη ελεγχόμενης επικοινωνίας με άλλους χρήστες να γίνουν οι χρήστες στόχοι τακτικών όπως social engineering, με αποτέλεσμα τη δημιουργία σοβαρών προβλημάτων ασφαλείας, όπως κλοπή λογαριασμών, κλοπή ταυτότητας άλλου μέλους (identity theft) από την οποία προκύπτουν εν συνεχεία άλλα προβλήματα (π.χ impersonation), persuasion, bribery κ.ο.κ. Τα ζητήματα αυτά αναλύονται εκτενέστερα σε επόμενο εδάφιο της εργασίας.



Εικόνα 17: Λειτουργία συνομιλίας (Chat)

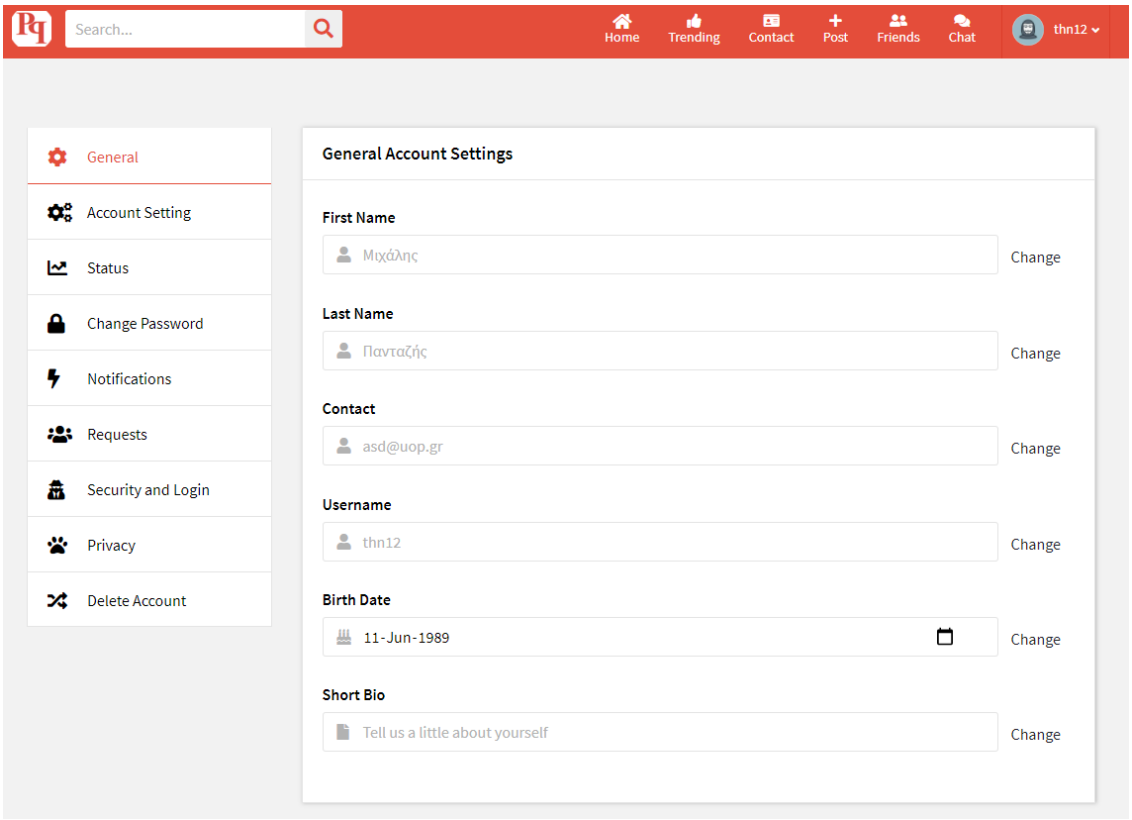
Δίπλα από το Chat εμφανίζεται το Avatar (εικόνα προφίλ) και το όνομα του χρήστη καθώς και ένα στοιχείο διεπαφής για πρόσβαση σε περισσότερες επιλογές (βελάκι με φορά προς τα κάτω). Αν πατηθεί το όνομα του χρήστη, τότε παρουσιάζεται το προφίλ του. Αν πατηθεί το κάτω βελάκι ή οποιοδήποτε άλλο σημείο του στοιχείου διεπαφής, τότε εμφανίζεται ένα μενού, σε ένα αναδυόμενο παράθυρο, από το οποίο ο χρήστης μπορεί να επιλέξει να ανακατευθυνθεί είτε στο προφίλ του (My Profile), είτε κατευθείαν στις ρυθμίσεις του λογαριασμού του (Settings), να αποσυνδεθεί από το κοινωνικό δίκτυο κ.ά.



Εικόνα 18: Μενού γρήγορης πλοήγησης χρήστη

2.1.7 Ρυθμίσεις εφαρμογής

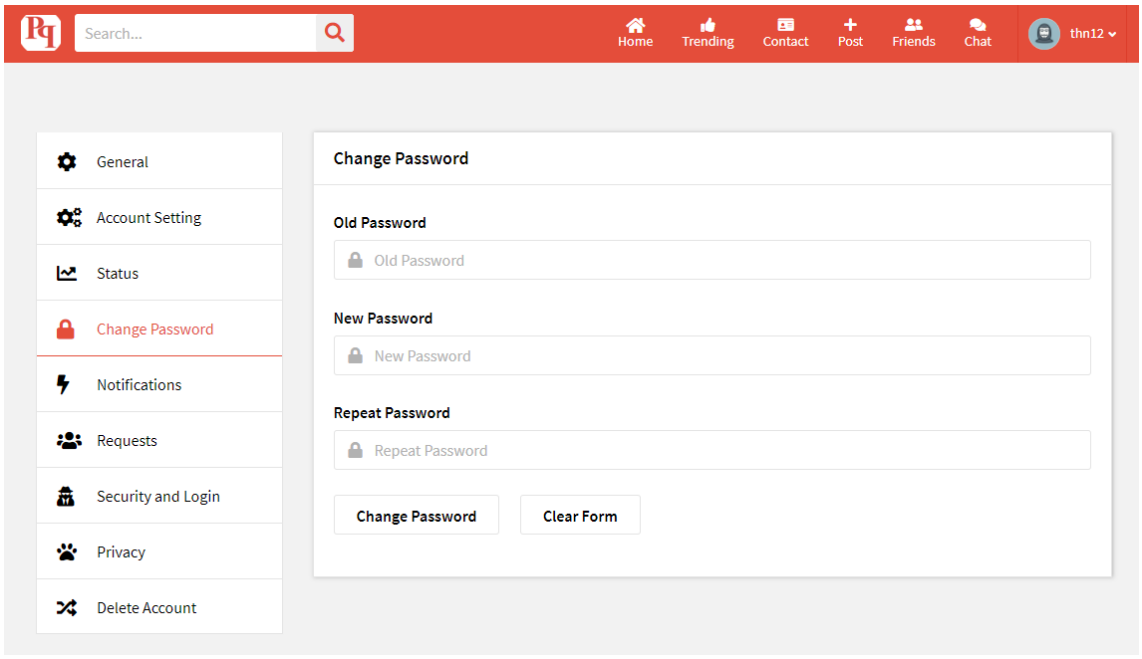
Στις ρυθμίσεις του λογαριασμού (Account Settings), είναι διαθέσιμες αρκετές επιλογές αλλαγής στοιχείων και προβολής δεδομένων για τον χρήστη. Στη πρώτη καρτέλα βρίσκονται οι γενικές ρυθμίσεις του λογαριασμού και εμφανίζονται τα πεδία First Name, Last Name, Contact, Username, Birth Date. Οι πληροφορίες που εμφανίζονται σε αυτά τα πεδία ανακτώνται κατευθείαν από τη βάση και πρόκειται για τα στοιχεία που έχει δηλώσει ο χρήστης κατά την εγγραφή του στην εφαρμογή. Αν επιθυμεί να αλλάξει κάποιο πεδίο, το επιλέγει και πληκτρολογεί την καινούρια τιμή. Κατόπιν, ο χρήστης έχει τη δυνατότητα να αποθηκεύσει τις αλλαγές. Στη περίπτωση που ο χρήστης εισάγει τα ίδια στοιχεία (χωρίς κάποια αλλαγή), εμφανίζεται ένα μήνυμα που τον ειδοποιεί ότι αυτά τα στοιχεία υπάρχουν ήδη (ιδιαίτερος έλεγχος πραγματοποιείται για το username και το email, για τα οποία ισχύουν οι ίδιοι περιορισμοί που ισχύουν και στην λειτουργία εγγραφής νέου χρήστη).



Εικόνα 19. Οθόνη Ρυθμίσεων (General Account Settings)

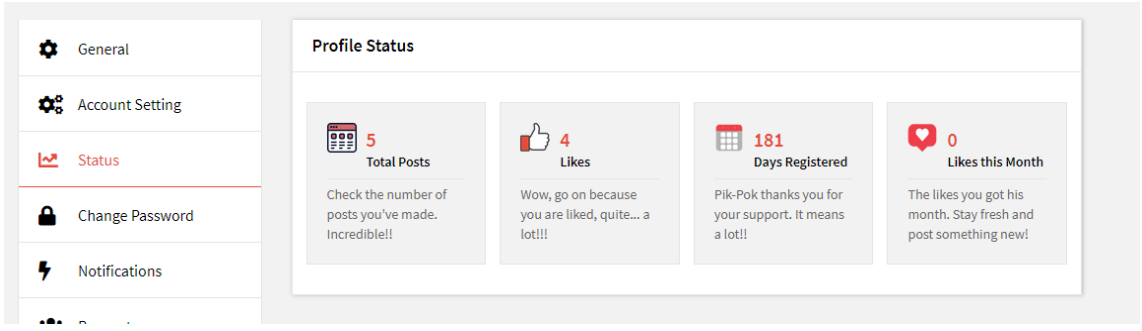
Μία ακόμη χρήσιμη λειτουργία είναι αυτή της αλλαγής κωδικού. Στη σχετική καρτέλα της διεπαφής περιέχονται τα πεδία Old Password, New Password και Repeat Password, τα οποία είναι απαραίτητα (required fields), ώστε να προχωρήσει ο χρήστης στην αλλαγή. Αφού γίνουν οι απαραίτητοι έλεγχοι ώστε να διαπιστωθεί ότι ο παλιός κωδικός ταιριάζει με τον αποθηκευμένο και ότι ο νέος κωδικός έχει επαναληφθεί σωστά (repeat), τότε με το κουμπί *Change Password* αλλάζει ο κωδικός. Πριν την καταχώρηση του κωδικού εφαρμόζονται όλοι οι έλεγχοι για την ασφάλεια του κωδικού (ελάχιστο μήκος, αλφάβητο κ.τ.λ.) και η αλλαγή πραγματοποιείται μόνο εάν το σύνολο των ελέγχων είναι επιτυχείς. Τέλος υπάρχει και το κουμπί clear form που δίνει τη δυνατότητα καθαρισμού όλων των πεδίων, στη περίπτωση που ο χρήστης έχει κάνει κάποιο λάθος και θέλει να τα

πληκτρολογήσει ξανά όλα τα στοιχεία από την αρχή. Γίνονται τέλος κατάλληλοι έλεγχοι ασφαλείας στα πεδία (sanitization με τα κατάλληλα φίλτρα κ.λπ.).



Εικόνα 20. Λειτουργία αλλαγής κωδικού

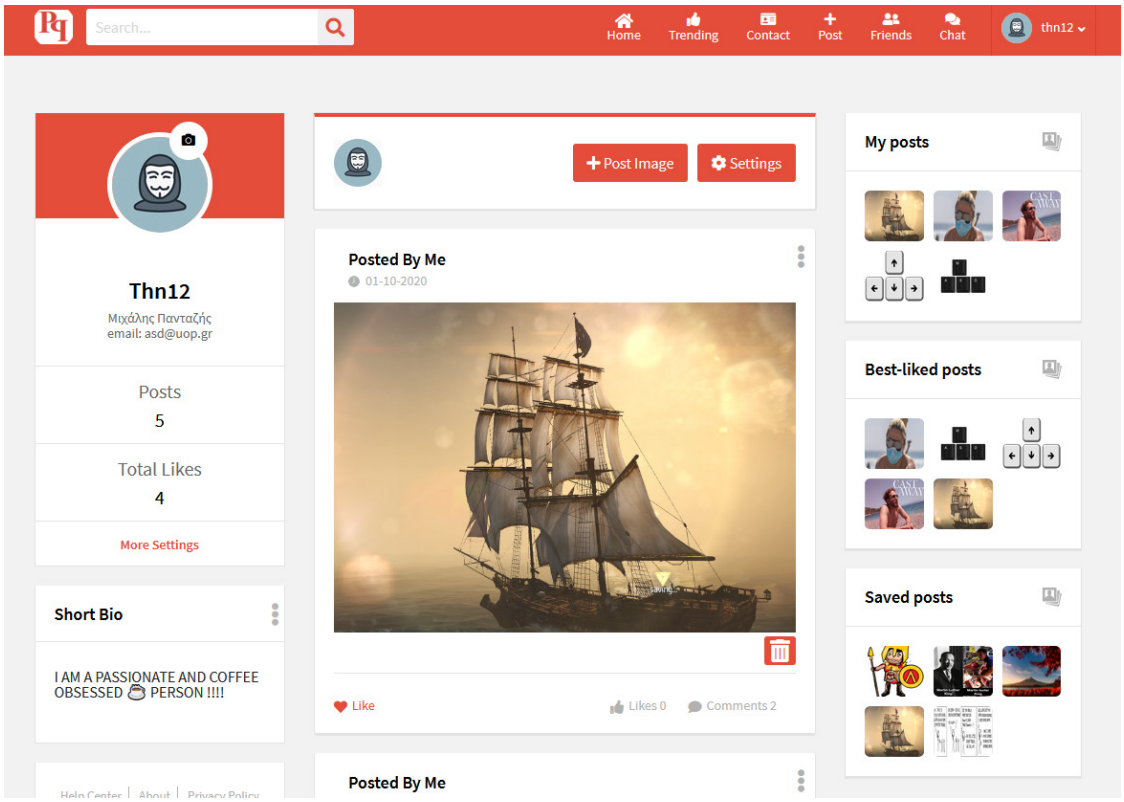
Στην καρτέλα **Status**, ο χρήστης μπορεί να δει στατιστικά στοιχεία (αριθμητικά κυρίως) για τον λογαριασμό του, όπως οι συνολικές δημοσιεύσεις που έχει κάνει, τα likes που έχει λάβει μέχρι τώρα σε όλες τις δημοσιεύσεις, τα likes κατά τον τελευταίο μήνα και το πλήθος ημερών που έχουν περάσει από την εγγραφή του. Υπάρχει η δυνατότητα επέκτασης, με συμπλήρωση πρόσθετων πληροφοριών και στατιστικών στο μέλλον. Όλα τα δεδομένα ανακτώνται από τη βάση δεδομένων, μέσω κατάλληλων ερωτήσεων.



Εικόνα 21. Καρτέλα Status

2.1.8 Προσωπική σελίδα χρήστη (MyProfile)

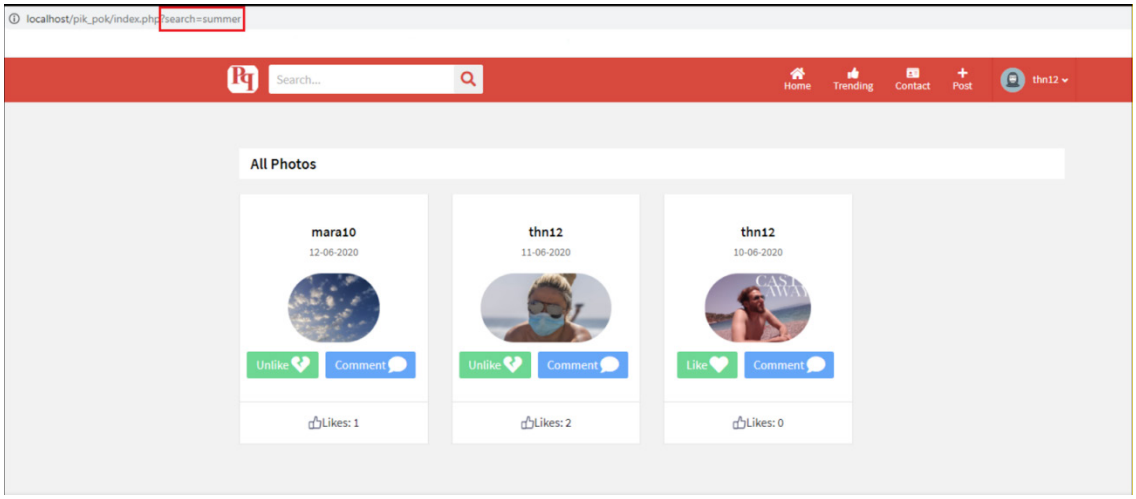
Αν ο χρήστης πατήσει ακριβώς δίπλα από το Avatar του, πάνω στο username του, τότε θα οδηγηθεί σε μία νέα σελίδα. Πρόκειται για την προσωπική σελίδα προφίλ του χρήστη. Σε αυτή τη σελίδα μπορεί να δει διάφορα στοιχεία για το προφίλ του. Πάνω αριστερά υπάρχει το Avatar του. Σε περίπτωση που πατήσει το κουμπί που βρίσκεται ακριβώς δίπλα, (η φωτογραφική μηχανή, η οποία παρουσιάζεται πάνω δεξιά), μπορεί να αλλάξει το Avatar του, ανεβάζοντας μία νέα φωτογραφία της επιλογής του. Στην σελίδα αυτή, ο χρήστης μπορεί να δει και άλλα φυσικά στοιχεία του, όπως το username, το όνομα του, το email του, πόσα posts έχει κάνει και τα συνολικά Likes που έχει λάβει από όλες τις δημοσιεύσεις του. Μπορεί επίσης και από εδώ να κάνει ένα νέο post και να μπει στις ρυθμίσεις του λογαριασμού του. Καθώς περιηγείται προς τα κάτω, βλέπει όλες τις δημοσιεύσεις που έχει ανεβάσει, ταξινομημένες βάσει ημερομηνίας και του παρέχεται και η δυνατότητα να διαγράψει όποιο post επιθυμεί, πατώντας στο εικονίδιο που έχει σχήμα κάδου και βρίσκεται κάτω δεξιά από τη φωτογραφία. Στο δεξί μέρος της Οθόνης (στο δεξιά sidebar), εμφανίζονται όλες οι φωτογραφίες που έχουν δημοσιευθεί από το χρήστη σε μορφή μικρογραφίας καθώς και τα post με τα περισσότερα Likes και τα αποθηκευμένα Posts (Ο χρήστης έχει τη δυνατότητα να αποθηκεύσει τα Post που του αρέσουν περισσότερο εφόσον αυτός επιθυμεί). Αν πατηθεί μία συγκεκριμένη φωτογραφία, ανοίγει σε μεγέθυνση και δίνεται η δυνατότητα για Comment και Like.



Εικόνα 22. Προσωπική σελίδα συνδεδεμένου χρήστη (MyProfile)

2.1.9 Λειτουργία αναζήτησης μέσω tags

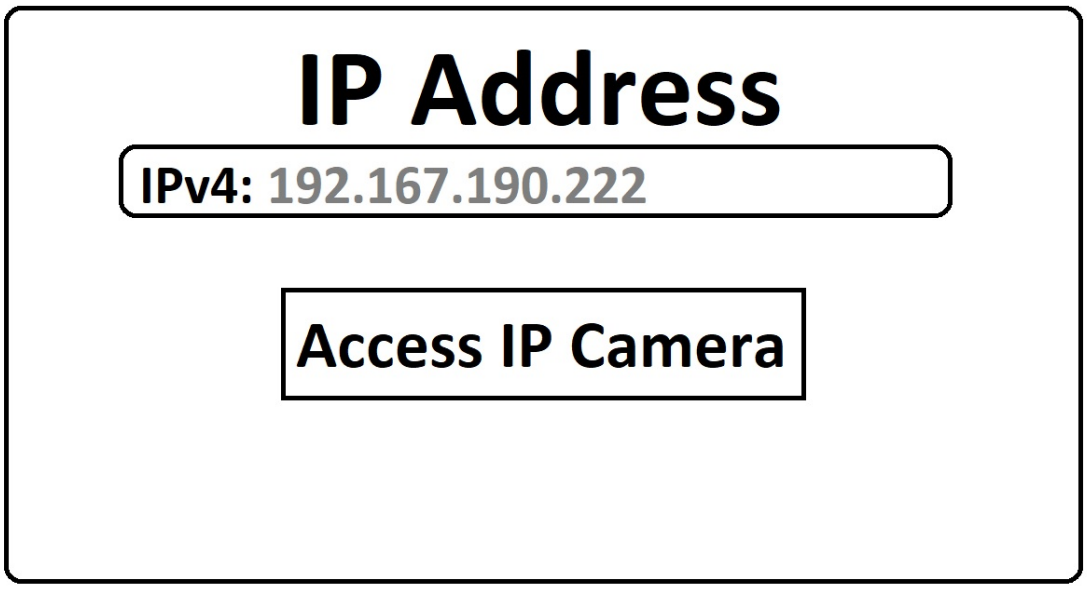
Σε κάθε καρτέλα υπάρχει η μπάρα αναζήτησης μέσω της οποίας ο χρήστης μπορεί να αναζητήσει μία ή περισσότερες φωτογραφίας βάσει των tags που οι φωτογραφίες αυτές έχουν. Αν π.χ. ένας χρήστης έχει δημοσιεύσει μία φωτογραφία και έχει βάλει σαν tag τη λέξη summer, τότε αν γίνει αναζήτηση βάσει tags και εισαχθεί ο όρος αναζήτησης «summer», τα αποτελέσματα της αναζήτησης θα περιλαμβάνουν τόσο τη συγκεκριμένη φωτογραφία, καθώς και οποιαδήποτε φωτογραφία περιέχει το tag «summer». Η αναζήτηση λειτουργεί ταιριάζοντας το προς αναζήτηση tag με οποιοδήποτε μέρος των καταχωρημένων tags: για παράδειγμα, κάνοντας αναζήτηση για “summer” επιστρέφονται όλες οι φωτογραφίες, στων οποίων τα tags περιέχεται η λέξη “summer”. Επίσης, αν ο χρήστης εισάγει ως όρο αναζήτησης μέρος του όρου, όπως π.χ. “sum” ή “mer” ή “m” πάλι επιστρέφονται οι φωτογραφίες που έχουν το tag summer, καθώς η προς αναζήτηση συμβολοσειρά περιέχεται στο tag των φωτογραφιών.



Εικόνα 23. Λειτουργία αναζήτησης μέσω tags

2.1.10 Λειτουργία αυτόματου ανεβάσματος από IP κάμερα

Μια πρόταση η οποία επί του παρόντος έχει αποτυπωθεί μόνο ως απαίτηση, διότι εκτιμήθηκε ότι προκύπτουν αρκετά θέματα ασφάλειας και ιδιωτικότητας (τα οποία θα αναλυθούν στην ενότητα 6) είναι να καταχωρούνται στο σύστημα αυτόματα φωτογραφίες ή/και βίντεο που αντλούνται αυτόματα από μία κάμερα IP. Από τη στιγμή που το σύστημα θα ρυθμιστεί κατάλληλα για επικοινωνία με την κάμερα (καταχώρηση της διεύθυνσης IP στην οποία η κάμερα λειτουργεί και τυχόν άλλες απαραίτητες παραμέτρους) η λειτουργία θα μοιάζει με την λειτουργία “take a photo” που περιγράφηκε προηγουμένως με την διαφορά ότι οι φωτογραφίες θα μπορούν να παίρνονται αυτόματα ανά τακτά χρονικά διαστήματα. Αργότερα θα μελετήσουμε τις επιπτώσεις που ενδέχεται να έχει αυτό στην ασφάλεια και την ιδιωτικότητα.



Εικόνα 24. Ενδεικτικό παράθυρο σύνδεσης στην Camera IP

3 Τεχνολογίες και εργαλεία που χρησιμοποιήθηκαν στην υλοποίηση της εφαρμογής

3.1 Εργαλεία που χρησιμοποιήθηκαν

Στη συνέχεια παρατίθενται οι τεχνολογίες και τα εργαλεία που κρίθηκαν απαραίτητα για την υλοποίηση του κοινωνικού δικτύου. Σε αυτά περιλαμβάνονται ένας επαγγελματικός code editor, ένας τοπικός web server και ένα σύστημα διαχείρισης βάσεων δεδομένων. Πιο συγκεκριμένα, ο code editor που χρησιμοποιήθηκε ονομάζεται Sublime Text Editor και αποτέλεσε βασικό εργαλείο για την ανάπτυξη του κώδικα του κοινωνικού δικτύου. Ο server που χρησιμοποιήθηκε ονομάζεται Apache και εγκαταστάθηκε ως κομμάτι της πλατφόρμας λογισμικού ανοιχτού κώδικα XAMPP. Ενώ ένα από τα σημαντικότερα εργαλεία το οποίο κρίθηκε απαραίτητο για τον ολοκληρωμένο και επαγγελματικό έλεγχο των εκδόσεων της υλοποίησης μας καθώς και για την αποτελεσματική συνεργασία μας στην συγγραφή του κώδικα ήταν το σύστημα διαχείρισης κώδικα Github.

3.1.1 XAMPP

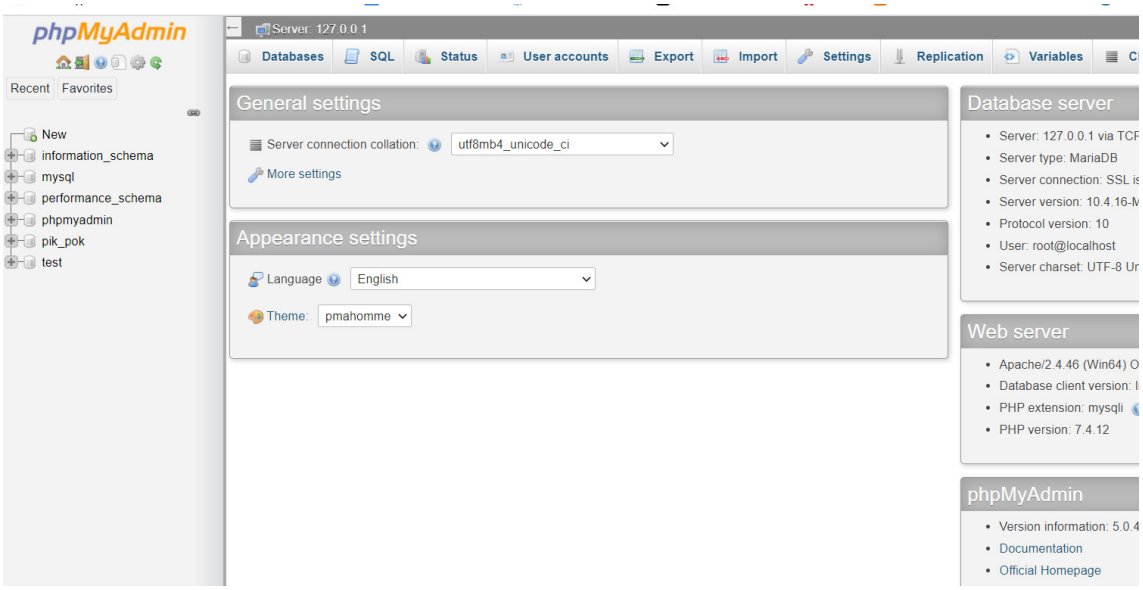
Το XAMPP είναι ένα πακέτο για δημιουργία και λειτουργία εφαρμογών διαδικτύου. Η ονομασία XAMPP είναι για ακρωνύμιο και αναφέρεται στα παρακάτω αρχικά:

- X (ανεξάρτητο πλατφόρμας - “cross-platform”)
- εξυπηρετητής ιστού Apache HTTP
- σύστημα διαχείρισης βάσεων δεδομένων MySQL
- Γλώσσες προγραμματισμού PHP και Perl

Το XAMPP είναι ένα δωρεάν, ελεύθερο λογισμικό το οποίο περιέχει έναν διακομιστή ιστοσελίδων (web server), ο οποίος μπορεί να εξυπηρετεί τόσο στατικές σελίδες HTML όσο και δυναμικές ιστοσελίδες τεχνολογίας PHP/MySQL ή/και Perl/MySQL. Αναπτύχθηκε από την Apache Friends και το λογισμικό αποτελείται κυρίως από τον διακομιστή Apache HTTP, τη βάση δεδομένων MariaDB (που είναι ο κλώνος ελεύθερου λογισμικού του συστήματος διαχείρισης βάσεων δεδομένων MySQL) και διερμηνείς για σενάρια γραμμένα στις γλώσσες προγραμματισμού PHP και Perl. Είναι ανεξάρτητο πλατφόρμας και τρέχει σε Microsoft Windows, Linux, και Mac OS X και χρησιμοποιείται ως πλατφόρμα για την σχεδίαση και ανάπτυξη ιστοσελίδων με τις τεχνολογίες όπως PHP. Στις τελευταίες εκδόσεις του XAMPP έχει εισαχθεί ως ενσωματωμένο συστατικό (build-in) και ο web server tomcat, και έτσι ο χρήστης έχει τη δυνατότητα να χρησιμοποιήσει σελίδες JSP καθώς και Servlets και να δημιουργήσει διαδικτυακές εφαρμογές βασισμένες σε Java, αφού ρυθμιστεί κατάλληλα το περιβάλλον. Όταν το XAMPP εγκατασταθεί στον τοπικό υπολογιστή, οι εφαρμογές ιστού μπορούν να ανακτούν τις στατικές ή δυναμικές σελίδες μέσω της διεύθυνσης http://localhost/. (16) Φυσικά, εάν η διεύθυνση του συστήματος είναι καταχωρημένη στο DNS είναι δυνατή η χρήση URL όπου αντί του localhost εισάγεται η διεύθυνση DNS του συστήματος.

3.1.2 PhpMyAdmin

Το phpMyAdmin είναι ένα ελεύθερο λογισμικό ανοικτού κώδικα για διαχείριση βάσεων δεδομένων MySQL, γραμμένο σε γλώσσα PHP. Στο πλαίσιο της εργασίας, το PhpMyadmin χρησιμοποιήθηκε για τη διαχείριση της βάσης δεδομένων και των πινάκων της με εύκολο τρόπο, μέσα από ένα γραφικό GUI. Ταυτόχρονα διευκόλυνε σημαντικά τη διαχείριση, καθώς υπήρχε η δυνατότητα προβολής των εγγραφών των πινάκων με εύκολο τρόπο, καθώς και για δημιουργία, διαγραφή ή τροποποίηση υπαρχόντων πεδίων και πινάκων. Το γραφικό περιβάλλον του phpMyAdmin φαίνεται στην παρακάτω εικόνα. (17)



Εικόνα 25. Περιβάλλον του εργαλείου phpMyAdmin

3.1.3 Github

Ένα από τα βασικά χαρακτηριστικά της ανάπτυξης λογισμικού είναι ότι γράφουμε τον πηγαίο κώδικα σε στάδια. Ξεκινάμε αρχικά με απλό κώδικα και αφότου πειραματιστούμε, προσθέτουμε, διορθώνουμε και γράφουμε περισσότερο κώδικα. Έτσι κατά την ανάπτυξη της εφαρμογής κρίθηκε σκόπιμο να υπάρχει ένα σύστημα που να καταγράφει τις διαδοχικές εκδόσεις του κώδικα αυτού. Ταυτόχρονα υπήρχε η ανάγκη συνεργασίας μεταξύ των μελών της ομάδας ώστε να εργάζεται ο καθένας σε διαφορετικό κομμάτι της υλοποίησης, χωρίς να δημιουργείται κάποιο πρόβλημα. Τα προβλήματα έγιναν πολύ εντονότερα όταν το λογισμικό έγινε μεγαλύτερο σε μέγεθος. Τότε, έγινε επιτακτική η ανάγκη να έχουμε ένα σύστημα για τη διαχείριση των εκδόσεων της υλοποίησής μας.

Ένα από τα συστήματα διαχείρισης εκδόσεων λογισμικού είναι το Git, που γράφτηκε αρχικά από τον Linus Torvalds (ο οποίος είναι επίσης δημιουργός του Linux) και στηρίζει την ανάπτυξη του πυρήνα Linux, του GNOME και πολλών άλλων έργων Ελεύθερου Λογισμικού / Λογισμικού Ανοικτού Κώδικα (ΕΛ/ΛΑΚ). Το Git είναι ένα κατανεμημένο σύστημα διαχείρισης εκδόσεων λογισμικού (Distributed Version Control System, DVCS),

και διαθέτει πλήθος λειτουργιών και χαρακτηριστικών. Κάποια από τα χαρακτηριστικά του Git είναι:

- Εύκολο branching (18), δηλ. δημιουργία αντιγράφων στοιχείων του λογισμικού, προκειμένου να αναπτυχθούν, να εξελιχθούν και να δοκιμαστούν ανεξάρτητα από τα «κύρια αντίγραφα». Δύο βασικές χρήσεις του branching είναι η εξέλιξη δυνατοτήτων του λογισμικού ή η προσθήκη νέων λειτουργιών, μέσα στο πλαίσιο ενός ελεγχόμενου περιβάλλοντος.
- Εύκολο merging (19), δηλ. ενσωμάτωσης των αντιγράφων που δημιουργήθηκαν κατά τη διαδικασία του branching στην κύρια ροή ανάπτυξης και εξέλιξης του λογισμικού. Το merging συνήθως λαμβάνει χώρα όταν οι λειτουργίες ή τα χαρακτηριστικά που αναπτύσσονται στο branch θεωρηθούν επαρκώς ώριμα.
- Έλεγχος εγκυρότητας των δεδομένων (μέσω της χρήσης της συνάρτησης κερματισμού SHA1)
- Παρακολουθεί το περιεχόμενο συνολικά και όχι ανά αρχείο
- Πολύ γρήγορο

Οι κύριες επιλογές που είχαμε ήταν τα λογισμικά Gitlab και Github. Το καθένα από τα δύο είχε τα δικά του θετικά και αρνητικά. Έμεις καταλήξαμε στο GitHub, καθώς είχαμε εκτενέστερη εμπειρία σε αυτό και κάλυπτε πλήρως τις απαιτήσεις του έργου μας. Αν θέλαμε να περιγράψουμε το GitHub, θα μπορούσαμε να πούμε ότι είναι ένα κεντρικό αποθετήριο λογισμικού και μία πλατφόρμα διαχείρισής του. Στο GitHub μπορεί να αναρτηθεί τόσο open source λογισμικό όσο και closed source. Μέσω του συστήματός του, δίνεται στους προγραμματιστές η δυνατότητα να ανεβάζουν το λογισμικό που δημιουργούν. Μπορούν να κρατήσουν αυτό το λογισμικό ιδιωτικό (private) και να έχουν μόνο αυτοί πρόσβαση σε αυτό, αν δεν είναι έτοιμοι να το διαθέσουν ακόμα, ή απλά πρόκειται για κάποιο έργο, το οποίο δεν επιθυμούν να εκδοθεί στο κοινό, ή μπορούν φυσικά να διαθέσουν ένα λογισμικό ανοικτού κώδικα με όποια διατεθεί ανοικτού λογισμικού αυτοί επιθυμούν (MIT, GPLv3 κ.λπ.). Μπορούν να φτιάξουν και τη δική τους άδεια διάθεσης. Παράλληλα μπορούν να κάνουν αλλαγές στον κώδικα τους, να αναβαθμίσουν το πρόγραμμα, και πολλά άλλα. Το GitHub, μέσω του συστήματος ελέγχου εκδόσεων που διαθέτει, διατηρεί όλες αυτές τις εκδόσεις του έργου μας ταξινομημένες. Με αυτόν τον τρόπο, μπορούμε εύκολα να τροποποιήσουμε το λογισμικό μας, να διαχειριστούμε τις αλλαγές, και να ανεβάσουμε ενημερώσεις. Επιπλέον, κάθε προγραμματιστής μπορεί να δει τον πηγαίο κώδικα οποιουδήποτε ανοικτού λογισμικού στο GitHub και να τον κατεβάσει. Μπορεί επίσης να κάνει τις δικές του τροποποιήσεις, και να τις μοιραστεί αν θέλει και με τους υπόλοιπους χρήστες. Ο κώδικας, εφ' όσον είναι διαθέσιμος στο κοινό, μπορεί να ανακτηθεί από οποιονδήποτε, ενώ θα πρέπει όσοι προβαίνουν σε λήψη λογισμικού να τηρούν τους κανόνες της εκάστοτε άδειας που διέπει το λογισμικό. (20)

3.1.4 Sublime Text Editor

Για το μεγαλύτερο κομμάτι της υλοποίησης χρησιμοποιήθηκε ο κειμενογράφος (editor) *Sublime Text editor*, ο οποίος είναι ένα πρόγραμμα επεξεργασίας πηγαίου κώδικα πολλαπλών πλατφορμών. Υποστηρίζει εγγενώς πολλές γλώσσες προγραμματισμού και γλώσσες σήμανσης, ενώ μπορούν να προστεθούν επιπλέον λειτουργίες από τους χρήστες. Οι λειτουργίες αυτές είναι συνήθως κατασκευασμένες από την κοινότητα και συντηρούνται με άδειες ελεύθερου λογισμικού. Γενικότερα, χρησιμοποιήθηκαν και άλλοι κειμενογράφοι όπως π.χ. Notepad++, Atom και VisualStudio Code αλλά η τελική επιλογή ήταν ο *Sublime Text editor*, διότι είχε όλες τις δυνατότητες που ήταν απαραίτητες (τόσο κλασικές όπως χρωματισμό του κώδικα, πρόβλεψη και αυτόματη συμπλήρωση, εύρεση και αντικατάσταση κομματιών του κώδικα κ.λπ., όσο και πιο σύγχρονες όπως επιλογές Goto Anything για να είναι δυνατό το άνοιγμα αρχείων με λίγα μόνο κλικ του ποντικιού, πολλαπλή επιλογή και πολλαπλή τροποποίηση κειμένου για να αλλάζουμε ταυτόχρονα σε δύο η περισσότερα μέρη τον κώδικά μας, καθώς και επιλογές για customization). Επίσης είχε το πλεονέκτημα ότι ανανεωνόταν άμεσα το κείμενο όταν γινόταν αλλαγή από εξωτερικό πρόγραμμα ή γινόταν pull από το Github. (21)

3.1.5 MySQL

Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων ανοικτού κώδικα (relational database management system – RDBMS), που χρησιμοποιεί την Structured Query Language (SQL), την πιο διαδεδομένη γλώσσα για την προσθήκη, την πρόσβαση και την επεξεργασία δεδομένων σε μία Βάση Δεδομένων. Επειδή είναι ανοικτού κώδικα (τουλάχιστον στον κλώνο MariaDB), οποιοσδήποτε μπορεί να κατεβάσει τη MySQL και να την διαμορφώσει με βάση τις ανάγκες του, σύμφωνα πάντα με τη γενική άδεια χρήσης. Η MySQL είναι γνωστή κυρίως για την ταχύτητα, την αξιοπιστία και την ευελιξία που παρέχει. Οι περισσότεροι συμφωνούν ωστόσο ότι είναι πιο αποτελεσματική για διαχείριση περιεχομένου και όχι για εκτέλεση συναλλαγών. Η MySQL αυτή τη στιγμή μπορεί να λειτουργήσει σε περιβάλλον Linux, Unix, και Windows. Ένας MySQL server διαχειρίζεται ένα σύνολο σχεσιακών βάσεων δεδομένων (databases). Κάθε σχεσιακή βάση δεδομένων είναι ένα σύνολο πινάκων (tables), με τις γραμμές κάθε του πίνακα να ονομάζονται εγγραφές (records) και τις στήλες του να ονομάζονται στήλες ή πεδία (columns/fields). Το κάθε πεδίο μπορεί να περιέχει μόνο ένα συγκεκριμένο τύπο πληροφορίας ο οποίος ορίζεται κατά τη δημιουργία του πίνακα. Σε κάθε πίνακα βάσης δεδομένων είναι σημαντικό να υπάρχει μια στήλη ή κάποιος συνδυασμός στηλών που σε κάθε γραμμή του πίνακα να έχει/έχουν διαφορετική τιμή έτσι ώστε κάθε εγγραφή να διαθέτει ένα μοναδικό χαρακτηριστικό, μία «ταυτότητα» με την οποία να είναι δυνατό να αναφερόμαστε σαφώς σε αυτήν. Το ιδιαίτερο αυτό πεδίο το ορίζουμε κατά την κατασκευή του πίνακα δίνοντάς του την ιδιότητα του πρωτεύοντος κλειδιού (primary key). (22)

3.2 Τεχνολογίες που χρησιμοποιήθηκαν

Παρακάτω παρατίθεται οι σημαντικότερες τεχνολογίες που αξιοποιήθηκαν για την υλοποίηση του κοινωνικού δικτύου. Σε αυτές περιλαμβάνονται όλες οι γλώσσες προγραμματισμού, σήμανσης και μορφοποίησης που χρησιμοποιήθηκαν, οι βιβλιοθήκες καθώς και τα frameworks που συνέβαλαν στην υλοποίηση του συστήματος.

3.2.1 HTML

Η **HTML** (HyperText Markup Language) είναι η κύρια γλώσσα σήμανσης για τις ιστοσελίδες του παγκόσμιου ιστού (world wide web) και τα στοιχεία της είναι τα βασικά δομικά στοιχεία όλων των ιστοσελίδων. Η HTML αποτελεί ένα σύνολο στοιχείων σήμανσης και κανόνων για τη διαμόρφωση της εμφάνισης και του περιεχομένου μιας ιστοσελίδας. Ουσιαστικά, δεν είναι γλώσσα προγραμματισμού, αλλά γλώσσα περιγραφής ιδιοτήτων των στοιχείων που αποτελούν μία ιστοσελίδα. Τα αρχεία HTML αποτελούνται από ειδικές, προκαθορισμένες ετικέτες (tags), οι οποίες περιγράφουν τον τρόπο με τον οποίο εμφανίζεται το περιεχόμενο HTML από την εφαρμογή πλοήγησης (browser), το πρόγραμμα δηλαδή που χρησιμοποιούμε για να προβάλουμε τις διάφορες σελίδες στο διαδίκτυο και περικλείονται πάντα μέσα σε σύμβολα «<» και «>» (π.χ. <html>). Οι ετικέτες HTML συνήθως λειτουργούν ανά ζεύγη (π.χ. <body> και </body>). Ανάμεσα στις ετικέτες, μας δίνεται η δυνατότητα να τοποθετήσουμε κείμενο, πίνακες, εικόνες, συνδέσμους κ.λπ. Ο σκοπός μιας εφαρμογής πλοήγησης είναι να διαβάζει τα έγγραφα HTML και να τα συνθέτει σε σελίδες που μπορεί ένας άνθρωπος διαβάσει ή να ακούσει, χρησιμοποιώντας τις ετικέτες που έχουμε τοποθετήσει στον κώδικά μας. Η HTML επιτρέπει την ενσωμάτωση εικόνων και άλλων αντικειμένων μέσα στη σελίδα, και μπορεί να χρησιμοποιηθεί για να εμφανίσει διαδραστικές φόρμες. Μπορούν επίσης να ενσωματωθούν σενάρια εντολών σε γλώσσες όπως η JavaScript, τα οποία επηρεάζουν τη συμπεριφορά των ιστοσελίδων HTML δίνοντας τους για παράδειγμα animations και κάνοντας τις γενικότερα διαδραστικές.

Η τρέχουσα έκδοση της HTML είναι η HTML5, η οποία έφερε αρκετές επαναστατικές αλλαγές στον τρόπο σύνταξης της HTML, εισάγοντας νέα tags τα οποία απλοποιούν την σήμανση του περιεχομένου και επιτρέπουν την καλύτερη δόμησή του, ενώ ταυτόχρονα παρέχει πολλές νέες δυνατότητες, όπως η δυνατότητα μορφοποίησης των εγγράφων, αλλάζοντας μεγέθη, χρώματα, γραμματοσειρές κ.λπ. με χρήση στοιχείων CSS που ενσωματώνονται στα HTML tags, χωρίς να χρειάζεται να συνδέσουμε κάποιο ξεχωριστό αρχείο τύπου CSS. Η HTML5, η οποία χρησιμοποιήθηκε στην υλοποίηση του συστήματος, είναι η πιο πρόσφατη έκδοση της HTML. Η προδιαγραφή της δημοσιεύθηκε τον Οκτώβρη του 2014 και αντικατέστησε την HTML 4.01, την XHTML 1.0 και την DOM Level 2 HTML. Η HTML5 προσφέρει επίσης πληθώρα από νέες δυνατότητες που απλοποιούν και διευκολύνουν την ανάπτυξη ενός ιστοτόπου. Προστίθενται νέες ετικέτες (tags) που προσφέρουν πιο ουσιαστική πληροφορία στις μηχανές αναζήτησης και νέες καινοτομίες που απλοποιούν την υλοποίηση διαφόρων εργασιών. Ο σκοπός της είναι η μείωση της ανάγκης για χρήση πολλών διαφορετικών plug-in εμπορικών εταιρειών και διαδικτυακών εφαρμογών όπως Adobe Flash, Microsoft Silverlight, Apache Pivot και Sun JavaFX . Τέλος, σε αυτή την έκδοση δίνεται μεγάλη έμφαση στην δημιουργία responsive ιστοσελίδων, σελίδων δηλαδή που προσαρμόζονται αυτοματοποιημένα σε διαφορετικές συσκευές. Η τεχνική της responsive σχεδίασης απευθύνεται κατά κύριο λόγο σε χρήστες

κινητών συσκευών (smartphones/tablets), που πιθανόν πλοηγούνται στην ιστοσελίδα με περιορισμένη ανάλυση οθόνης και πιθανώς περιορισμένο εύρος ζώνης (bandwidth). Θα πρέπει συνεπώς η ιστοσελίδα όχι μόνο να ανταποκρίνεται στον περιορισμένο «χώρο» της οθόνης τους, αλλά και να φορτώνεται όσο το δυνατόν ταχύτερα.

Η HTML5, χρησιμοποιήθηκε αρκετά για την υλοποίηση του κοινωνικού δικτύου, καθώς αποτέλεσε τη ραχοκοκαλιά για την εμφάνιση της διεπαφής του συστήματος που δημιουργήσαμε. Με χρήση αυτής, εμφανίζονται οι οποιεσδήποτε φόρμες, πίνακες, σχόλια, εικόνες και οποιοδήποτε άλλο στοιχείο της διεπαφής χρειάστηκε, ενώ μέσω κατάλληλης επικοινωνίας με τα αρχεία PHP (όπου υλοποιείται και η επικοινωνία με τη βάση δεδομένων) καθώς και JavaScript, επιτελέστηκε η επιτυχής λειτουργία του συστήματος. (23)

3.2.2 CSS

Η **CSS** (Cascading Style Sheets) είναι μια τεχνολογία που χρησιμοποιείται για τον έλεγχο της εμφάνισης και της μορφοποίησης ενός εγγράφου που έχει γραφτεί με μια γλώσσα σήμανσης (π.χ. HTML). Η CSS είναι μια γλώσσα προορισμένη να προσδιορίζει στυλιστικά μια ιστοσελίδα, δίνοντάς μας τη δυνατότητα να διαχωρίσουμε το περιεχόμενο της ιστοσελίδας μας από τον τρόπο παρουσίασής του. Μας δίνει δηλαδή τη δυνατότητα να τροποποιήσουμε πληθώρα στοιχείων εμφάνισης όπως π.χ. τα χρώματα, τη στοίχιση, την γραμματοσειρά και οποιαδήποτε άλλη πτυχή εμφάνισης της ιστοσελίδας μας με αρκετά απλό τρόπο. Επίσης, μας επιτρέπει να δημιουργήσουμε πρότυπα με τα οποία μπορούμε να μορφοποιήσουμε με ομοιόμορφο τρόπο κάθε σελίδα του ιστοτόπου μας, απλά εισάγοντας τον κατάλληλο σύνδεσμο σε οποιοδήποτε αρχείο HTML επιθυμούμε. Για μια καλαίσθητη και καλοσχεδιασμένη ιστοσελίδα, η χρήση της CSS κρίνεται ως απαραίτητη.

Ταυτόχρονα, με χρήση της CSS μπορούμε να κάνουμε μία ιστοσελίδα responsive, κάνοντας τη εύχρηστη και σε κινητές συσκευές (smartphones, tablets κ.λπ.), μία πτυχή που επίσης θεωρείται απαραίτητη πλέον, αφού όλοι διαθέτουν και χρησιμοποιούν τέτοιου είδους συσκευές. Για το σύστημα δημιουργήθηκε ένα βασικό CSS template, όπου καθορίζεται η εμφάνιση των περισσότερων στοιχείων της διεπαφής μας, όπως κουμπιά, πίνακες, επικεφαλίδες, headers, footers και άλλα. Ταυτόχρονα κατεβλήθη προσπάθεια ώστε να είναι η διεπαφή του συστήματος responsive, έτσι ώστε να μπορεί να χρησιμοποιηθεί άνετα και από κινητές συσκευές. Συντάσσοντας τις σελίδες μόνο με κώδικα HTML, θα ήταν εφικτό να οριστεί άμεσα το χρώμα και το μέγεθος του κειμένου αλλά και άλλων στοιχείων της σελίδας (όπως πίνακες, links, λίστες κ.τλ.). Κατά τη φάση της συντήρησης, το μόνο που θα χρειαζόταν θα ήταν να βρεθεί π.χ το χρώμα ενός στοιχείου μέσα στον κώδικα και να τροποποιηθεί κατάλληλα. Αυτή η διαδικασία μπορεί να φαντάζει εύκολη για ένα αρχείο, αλλά αν το site μας αποτελείται από δεκάδες σελίδες, αυτή η μέθοδος είναι χρονοβόρα και δεν μας επιτρέπει να διαχειριζόμαστε εύκολα και γρήγορα τη μορφοποίηση του ιστοτόπου μας, ενώ είναι επιρρεπής και σε παραλείψεις που θα οδηγήσουν σε ασυνέπειες στην εμφάνιση. Ας αναλογιστούμε, για παράδειγμα, πόσο χρονοβόρο θα είναι αν θελήσουμε κάποια στιγμή να αλλάξουμε τα χρώματα στο κύριο μενού του site μας, το οποίο επαναλαμβάνεται σε όλες τις σελίδες. Σε μια τέτοια περίπτωση θα χρειαζόταν να ανοίγουμε κάθε σελίδα του site και να αλλάζουμε τα χρώματα του φόντου και των links του μενού, διαδικασία που εκτός από χρονοβόρα είναι

και κουραστική και μπορεί τελικά να δημιουργήσει και ανομοιομορφίες στην εμφάνιση, καθώς υπάρχει υπολογίσιμη πιθανότητα οι οδηγίες μορφοποίησης να μην είναι ίδιες σε κάθε σελίδα. Εκτός από την ευκολία στην διαχείριση ενός site, ένα άλλο σημαντικό πλεονέκτημα της χρήσης CSS στις σελίδες είναι ο "καθαρότερος" κώδικας, χωρίς πολλές ιδιότητες στις ετικέτες οι οποίες τον κάνουν δυσανάγνωστο. Επιπλέον κάνει γρηγορότερη την πλοήγηση καθώς το αρχείο, μέσα στο οποίο ορίζονται τα στυλ, "διαβάζεται" από τον browser μόνο μια φορά και έπειτα αποθηκεύεται στην cache memory, μειώνοντας έτσι το μέγεθος της πληροφορίας που γίνεται download από τους browsers. (24)

3.2.3 JavaScript

Η **Javascript** είναι μία αντικειμενοστρεφής γλώσσα σεναρίων (scripting language) κατάλληλη για την ανάπτυξη διαδικτυακών εφαρμογών και σεναρίων στην πλευρά του client (εκτελείται στην εφαρμογή πλοήγησης). Υποστηρίζεται από όλους τους δημοφιλείς περιηγητές, όπως οι Microsoft Internet Explorer (ξεκινώντας με την έκδοση 3.0), Firefox, Safari, Opera, Google Chrome, κ.τ.λ. Η JavaScript ενσωματώνεται (έμμεσα ή άμεσα) στον κώδικα της HTML. Ο συνδυασμός της JavaScript με την HTML αποφέρει δυναμικές ιστοσελίδες (Ιστοσελίδες με δυνατότητα εμφάνισης δυναμικού περιεχομένου) αλλά και ιστοσελίδες με υψηλό βαθμό διαδραστικότητας. Με τη JavaScript μπορούμε να εισάγουμε λογική σε μία εφαρμογή, να τοποθετήσουμε «δυναμικό κείμενο», να κάνουμε την εφαρμογή ικανή να «αντιδρά σε γεγονότα», να εξακριβώσουμε την εγκυρότητα δεδομένων, να τροποποιήσουμε, να εμφανίσουμε ή να αποκρύψουμε δεδομένα, να εφαρμόσουμε animations και άλλα. Η JavaScript ήταν πολύ χρήσιμη στο πλαίσιο ανάπτυξης του συστήματος, καθώς αξιοποιήθηκαν πολλά στοιχεία από αυτά που αναφέρθηκαν. Επί παραδείγματι, χρησιμοποιήθηκε για να εισαχθεί η κατάλληλη λογική ελέγχου πριν την αποστολή των φορμών της HTML, να ελεγχθεί η εγκυρότητα των δεδομένων κ.ά. Γενικότερα αποτελεί μία από τις πιο ευέλικτες και αποτελεσματικές γλώσσες που μπορεί να χρησιμοποιηθούν σε διαδικτυακές εφαρμογές. Σύμφωνα με μία έρευνα, η JavaScript χρησιμοποιείται από το 88% όλων των ιστοτόπων. Κάνοντας περαιτέρω έρευνα παρατηρήσαμε ότι μπορούμε να βρούμε τη JavaScript όχι μόνο σε κάθε ιστότοπο, αλλά επίσης και σε εφαρμογές κινητών, σε παιχνίδια και διάφορες άλλες εφαρμογές.

Στη συνέχεια παραθέτουμε συνοπτικά τα πλεονεκτήματα της JavaScript, τα οποία μας οδήγησαν να την χρησιμοποιήσουμε, πλεονεκτήματα τα οποία την καθιστούν επίσης αρκετά δημοφιλή και στην κοινότητα γενικότερα:

- **Εύκολη εκμάθηση:** Το συντακτικό της JavaScript είναι αρκετά απλό και λιτό, ενώ αρκετές “δύσκολες” έννοιες, οι οποίες υπάρχουν σε άλλες γλώσσες (π.χ. C και Java) έχουν απλοποιηθεί αρκετά στην JavaScript, κάνοντας την εύκολη στην εκμάθηση για τους περισσότερους προγραμματιστές τόσο για τους έμπειρους, όσο και για τους νεοεισερχόμενους στον χώρο.
- **Ύπαρξη ελεύθερου λογισμικού και πακέτων τρίτων κατασκευαστών:** Ένα πολύ σημαντικό κίνητρο για τη χρησιμοποίηση της γλώσσας είναι η ύπαρξη πολυάριθμων πακέτων ελεύθερου λογισμικού, τα οποία ήταν διαθέσιμα στο Github και σε άλλα μέρη του διαδικτύου. Δεδομένου ότι η JavaScript είναι πλέον μία πολύ δημοφιλής και εύκολη στην εκμάθηση γλώσσα, υπάρχουν αρκετά πακέτα και προσθήκες τρίτων κατασκευαστών τα οποία υλοποιούν κοινές λειτουργίες, απλοποιώντας σημαντικά την εργασία ανάπτυξης, καθώς αξιοποιείται

αποτελεσματικά ώριμο και ελεγμένο λογισμικό που έχει δημιουργηθεί από την κοινότητα.

- **Επεξεργασία από την Πλευρά του Πελάτη (Client):** Αυτό σημαίνει ότι ο κώδικας εκτελείται εντός της εφαρμογής πλοήγησης στην πλευρά του χρήστη, αντί του εξυπηρετητή ιστού (web server), εξοικονομώντας έτσι εύρος ζώνης και περιορίζοντας την υπερφόρτωση του εξυπηρετητή, κάτι που κάποιες φορές μπορεί να είναι χρήσιμο, καθώς η επεξεργαστική ισχύς στην πλευρά του χρήστη είναι αρκετά μεγάλη, στις μέρες μας. Παράλληλα, η εκτέλεση εντός της εφαρμογής πλοήγησης αυξάνει σημαντικά τη διαδραστικότητα.
- **Η συγγραφή και η ενσωμάτωση της είναι απλή:** Για την συγγραφή και την εκτέλεση κώδικα δεν απαιτείται κανένας ειδικός μεταγλωττιστής ή συντάκτης. Το μόνο που χρειαζόμαστε για να γράψουμε JavaScript, είναι ένα πρόγραμμα επεξεργασίας κειμένου και ένας περιηγητής μέσω του οποίου θα εκτελεστεί ο κώδικας και θα εμφανιστεί το αποτέλεσμα. Η εύκολη δυνατότητα ενσωμάτωσης της γλώσσας στις σελίδες που βλέπει ο χρήστης και η εύκολη διαχείριση αλλαγών, αρκετές φορές καθιστά την εργασία των προγραμματιστών, ευκολότερη, δημιουργώντας ενδεχομένως άλλα προβλήματα π.χ. στην ασφάλεια, τα οποία θα αναλύσουμε αργότερα.
- **Άμεση ανάδραση και φιλικότητα προς τον τελικό χρήστη:** Με τη JavaScript, κάθε πεδίο μπορεί να επαληθεύεται καθώς συμπληρώνεται από τους χρήστες, γεγονός που παρέχει άμεση ανατροφοδότηση, όταν αυτοί κάνουν κάποιο λάθος. Γενικότερα δεν χρειάζεται, πλέον, οι επισκέπτες να συμπληρώσουν μία ολόκληρη φόρμα και να την υποβάλλουν, για να μάθουν πως υπάρχει κάποιο τυπογραφικό λάθος σε κάποιο πεδίο και ότι θα πρέπει να συμπληρώσουν ολόκληρη τη φόρμα ξανά. Κάνοντας χρήση JavaScript και άλλων τεχνολογιών μπορούμε να διαχειριστούμε τέτοια θέματα και να κάνουμε την σελίδα μας πιο διαδραστική και φιλική προς τον χρήστη.
- **Η JavaScript είναι ενσωματωμένη σε όλους τους σύγχρονους περιηγητές:** Οι χρήστες του ιστότοπου δεν χρειάζονται ειδικό λογισμικό και λήψεις επιπρόσθετων προγραμμάτων για να δουν το αποτέλεσμα της JavaScript. Έτσι μπορούμε να παρέχουμε σε κάθε χρήστη την ίδια εμπειρία και ταυτόχρονα να μην χρειάζεται και εμείς να χρησιμοποιήσουμε κάποιο άλλο λογισμικό. (25)

3.2.4 PHP

Η **PHP** (Hypertext Preprocessor) είναι μια γλώσσα προγραμματισμού για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο. Η γλώσσα PHP μπορεί να εγκατασταθεί σχεδόν σε όλα τα λειτουργικά συστήματα όπως Windows, Linux, Mac OS X, Risc OS κ.λπ. αλλά και υποστηρίζεται και από τους περισσότερους εξυπηρετητές ιστοσελίδων όπως ο Apache ή ο IIS. Η PHP μπορεί να λειτουργήσει είτε ως εγκατεστημένη μονάδα (module) στον εξυπηρετητή ιστοσελίδων είτε μέσω ενός επεξεργαστή CGI σεναρίων. Η PHP μπορεί να χρησιμοποιηθεί για εκτέλεση σεναρίων (scripts) από την πλευρά του απομακρυσμένου εξυπηρετητή ιστοσελίδων όπως γίνεται και με τα σενάρια CGI. Επίσης η PHP μπορεί να χρησιμοποιηθεί για είσοδο/έξοδο δεδομένων από τον χρήστη ή για την δυναμική δημιουργία σελίδων. Η PHP είναι μία γλώσσα σεναρίων (scripting language) κατάλληλη για την ανάπτυξη διαδικτυακών εφαρμογών και σεναρίων στην πλευρά του εξυπηρετητή (server side scripting). Μια σελίδα PHP περνά από επεξεργασία μέσα από έναν συμβατό server (π.χ. Apache). Χρησιμοποιώντας την

PHP, μπορούμε να κάνουμε την κατάλληλη επεξεργασία και διαχείριση της εισόδου του χρήστη καθώς και να επικοινωνήσουμε με την βάση αν χρειαστεί, κάνοντας τα κατάλληλα queries, τα οποία μπορούμε να αποθηκεύσουμε σε μεταβλητές της php. Μια ιστοσελίδα που περιέχει κάποιον κώδικα σε PHP υφίσταται προεπεξεργασία από τη μηχανή της PHP, που αποκαλείται διερμηνευτής (interpreter), και τα αποτελέσματα αυτής της επεξεργασίας στέλνονται πίσω στον web server και από εκεί στον φυλλομετρητή του χρήστη της ιστοσελίδας. Ο κώδικας PHP μπορεί να θέσει ερωτήματα σε βάσεις δεδομένων, να δημιουργήσει εικόνες, να διαβάσει και να γράψει αρχεία, να συνδεθεί με απομακρυσμένους υπολογιστές, κ.ο.κ. Τα αποτελέσματα της επεξεργασίας του κώδικα PHP είναι τα μόνα που στέλνονται στην εφαρμογή πλοήγησης, ο κώδικας που τα δημιούργησε παραμένει κρυφός και συνεπώς πολύ πιο ασφαλής. Αυτό το είδος της προεπεξεργασίας αποκαλείται server-side scripting και ενώ δεν παρέχει το ίδιο είδος δυναμικών εφέ όπως η JavaScript, οι PHP σελίδες αποκαλούνται δυναμικές. Για την ανάπτυξη του συστήματος η γλώσσα PHP χρησιμοποιήθηκε και για τους δύο λόγους, καθώς χρειάστηκε να λαμβάνονται τα δεδομένα που εισήγαγε ο χρήστης και να υπόκεινται σε επεξεργασία, όπως και να πραγματοποιείται επικοινωνία με τη βάση δεδομένων, όποτε αυτό ήταν απαραίτητο (π.χ. για να εισάγουμε κάποιον νέο χρήστη, κάποιο νέο post, like/comment, να πραγματοποιείται ενημέρωση όταν ο χρήστης θέλει να αλλάξει τα στοιχεία του και άλλα). (26)

3.2.5 AJAX

Η AJAX (Asynchronous JavaScript and XML) είναι ένα σύνολο τεχνικών ανάπτυξης ασύγχρονων εφαρμογών στο Διαδίκτυο, που με τη σειρά του χρησιμοποιεί πολλές τεχνολογίες από την πλευρά του προγράμματος πελάτη (client), πιθανώς σε συνεργασία με αντίστοιχα στοιχεία από την πλευρά του εξυπηρέτη. Με την AJAX οι εφαρμογές ιστού μπορούν να στέλνουν και να λαμβάνουν δεδομένα στο παρασκήνιο, ανεξάρτητα από το τι προβάλλεται στην σελίδα. Ο διαχωρισμός της ανταλλαγής δεδομένων από την προβολή τους, επιτρέπει στην ιστοσελίδα να αλλάζει το περιεχόμενο της δυναμικά χωρίς να χρειάζεται ολοκληρωτική ανανέωση της. Η χρήση της AJAX έχει συνεισφέρει στην ραγδαία εξέλιξη των διαδραστικών και δυναμικών εφαρμογών σε ιστοσελίδες. Αξίζει να σημειωθεί ότι η τεχνολογία της AJAX δεν είναι μία τεχνολογία από μόνη της, αλλά ένας συνδυασμός τεχνολογιών. Η AJAX χρησιμοποιεί HTML και CSS για τη σήμανση της δομής και της εμφάνισης. Η χρήση της JavaScript, σε συνδυασμό με το αντικείμενο XMLHttpRequest έρχεται να καλύψει τον χρόνο που κάνει μία σελίδα για να φορτώσει (page loading). Με άλλα λόγια, με τη χρήση της τεχνολογίας αυτής, δεν είναι απαραίτητο το πλήρες φόρτωμα της ιστοσελίδας, παρά μόνο φόρτωση συγκεκριμένης πληροφορίας (partial loading). Η δυνατότητα αυτή, μας προσφέρει μεγαλύτερη ταχύτητα και απαιτεί λιγότερο bandwidth – traffic, αφού πλέον δεν φορτώνεται ολόκληρη η σελίδα, αλλά μόνο το κομμάτι που θέλουμε να ανανεώσουμε. (27)

3.2.6 jQuery

Η jQuery είναι μια ελαφριά σε κατανάλωση πόρων (light-weight) βιβλιοθήκη JavaScript σχεδιασμένη να απλοποιήσει την υλοποίηση σεναρίων (scripting) στην πλευρά του πελάτη (client-side) της HTML και υποστηρίζει πολλαπλές εφαρμογές πλοήγησης. Είναι αρκετά εύκολη στην εκμάθηση ειδικά αν ο προγραμματιστής έχει γνώσεις Javascript. Η βιβλιοθήκη κυκλοφόρησε τον Ιανουάριο του 2006 από τον Τζον Ρέριγκ (John Resig). Χρησιμοποιείται σε πάνω από το 65% των 10.000 ιστοτόπων με τη μεγαλύτερη επισκεψιμότητα. Η jQuery είναι ελεύθερο λογισμικό, με άδεια MIT. Χρησιμοποιώντας jQuery, παρέχεται η δυνατότητα να δημιουργηθούν διάφορα εφέ στη σελίδα, χωρίς να χρειάζεται η ποσότητα κώδικα που δημιουργείται στην παραδοσιακή Javascript. Με την jQuery μπορούν να πραγματοποιηθούν πολλές λειτουργίες στις σελίδες όπως επιλογή και διαμόρφωση HTML στοιχείων, διαμόρφωση CSS στοιχείων, διεργασίες HTML γεγονότων, εφέ JavaScript και animations, διαμόρφωση του HTML DOM [Document Object Module], χρήση AJAX αλλά και πληθώρα άλλων εφαρμογών. (28)

3.2.7 Bootstrap

Το Bootstrap είναι ένα front-end framework μέσω του οποίου σχεδιαστές και προγραμματιστές, μπορούν να ελαχιστοποιήσουν τον χρόνο δημιουργίας μιας ιστοσελίδας, είτε όταν αυτή η δημιουργία ξεκινάει από το μηδέν είτε όταν χρησιμοποιούν templates που έχουν διατεθεί από την κοινότητα. Αυτό επιτυγχάνεται με χρήση των ενσωματωμένων στοιχείων τα οποία παρέχονται από το framework και είναι απαραίτητα για τον σχεδιασμό μιας σύγχρονης ιστοσελίδας. Περιέχει αρχεία HTML, CSS, JavaScript καθώς και εικόνες. Γενικότερα το bootstrap περιέχει μεταξύ άλλων εικονίδια, αναδυόμενα μενού, κουμπιά διαφόρων χρωμάτων και μεγεθών, ετικέτες, κεφαλίδες, εφέ φτιαγμένα σε JavaScript, λίστες, μενού περιήγησης και άλλα. Στο framework παρέχονται επίσης grids μέσω των οποίων μπορούμε να δομήσουμε την ιστοσελίδα μας καλύτερα. Βέβαια πλέον υπάρχει και το προκαθορισμένο CSS Grid, αλλά ακόμη πολλοί προγραμματιστές προτιμούν να δομούν τα στοιχεία της σελίδας τους μέσω του συστήματος του bootstrap. Τέλος μπορεί κάποιος να βρει ολόκληρα templates φτιαγμένα από την κοινότητα σε bootstrap και να τα χρησιμοποιήσει σε ερασιτεχνικές ή επαγγελματικές εφαρμογές. (29)

4 Υλοποίηση

4.1 Δημιουργία της Βάσης Δεδομένων

Ένα βασικό στοιχείο για την κατασκευή του κοινωνικού δικτύου είναι η ανάδειξη των βασικών οντοτήτων της εφαρμογής και ο σχεδιασμός του κατάλληλου σχήματος αποθήκευσής τους στη βάση δεδομένων. Στο πλαίσιο της MySQL η κωδικοποίηση που χρησιμοποιήθηκε είναι η utf8_unicode_ci προκειμένου να υπάρχει πλήρης υποστήριξη ελληνικών χαρακτήρων. Σε κάποια σημεία χρειάστηκε να χρησιμοποιηθεί η νεότερη κωδικοποίηση utf8_mb4_unicode_ci. Η διαφορά μεταξύ των δύο είναι ότι η κωδικοποίηση utf8 μπορεί να αποθηκεύσει μόνο 3 χαρακτήρες byte, ενώ η κωδικοποίηση utf8_mb4 μπορεί να αποθηκεύσει 4 χαρακτήρες byte, παρέχοντας πλήρη υποστήριξη για κάθε δυνατή γλώσσα. Σε όρους Unicode, το utf8 μπορεί να αποθηκεύσει μόνο χαρακτήρες στο Βασικό Πολύγλωσσο Επίπεδο, ενώ το utf8mb4 μπορεί να αποθηκεύσει οποιονδήποτε χαρακτήρα Unicode. Όταν π.χ. χρειάστηκε να προστεθούν emoticons στη βάση, τότε απαιτήθηκε μεγαλύτερο εύρος unicode χαρακτήρων, οπότε και άλλη κωδικοποίηση. Στη συνέχεια δημιουργήθηκαν και οι πίνακες (tables) που χρησιμοποιούνται για τις λειτουργίες του συστήματος, όπως αποθήκευση χρηστών, δημοσιεύσεων, likes/comments κ.λπ.

4.1.1 Οι πίνακες στη βάση δεδομένων

Οι πίνακες που δημιουργήθηκαν στη βάση δεδομένων ήταν έντεκα στον αριθμό και αφορούν τους εγγεγραμμένους χρήστες (members), τις φωτογραφίες των δημοσιεύσεων (images), τα likes (post_likes), τα σχόλια (comments), τα μηνύματα στο chat (chat_messages) κ.ά. Στη συνέχεια παρατίθενται όλοι οι πίνακες, ενώ μετέπειτα αναλύεται η δομή του κάθε ενός.

- members
- images
- post_likes
- post_comments
- comment_likes
- comment_replies
- hashtags
- ip_mac_addresses
- relationship
- saved_posts
- chat_message

4.1.1.1 Πίνακας members

Ο πίνακας members χρησιμοποιείται για την αποθήκευση των στοιχείων των εγγεγραμμένων στην εφαρμογή χρηστών και αποτελείται από 12 πεδία τα οποία φέρουν τα ονόματα id, username, password, fname, lname, email, date_of_registration, gender, date_of_birth, profile_pic, picture_path, bio και status.

Στον πίνακα που ακολουθεί καταγράφονται τα πεδία του πίνακα και ο τύπος του καθενός, ενώ παρατίθενται και σχετικά σχόλια και εξηγήσεις.

Όνομα πεδίου	Τύπος	Σχόλια
id	int(11)	Στο πεδίο αυτό παράγεται από τη βάση δεδομένων ένα μοναδικό αναγνωριστικό για κάθε εγγεγραμμένο χρήστη. Το πεδίο αυτό είναι το πρωτεύον κλειδί του πίνακα members.
username	varchar(40)	Στο πεδίο αυτό θα αποθηκεύεται το username που επέλεξε κατά την εγγραφή του ο χρήστης.
password	varchar(250)	Στο πεδίο αυτό θα αποθηκεύεται ο κωδικός του χρήστη. Ο κωδικός αποθηκεύεται κρυπτογραφημένος στη βάση με κωδικοποίηση SHA256. Η διαδικασία εκτελείται προγραμματιστικά με χρήση γλώσσας PHP κατά την εισαγωγή των κωδικών στη βάση.
fname	varchar(100)	Στο πεδίο αυτό αποθηκεύεται το όνομα του χρήστη. Ο τύπος δεδομένων του πεδίου είναι varchar.
lname	varchar(100)	Στο πεδίο αυτό θα αποθηκεύεται το επώνυμο του χρήστη. Ο τύπος δεδομένων του πεδίου είναι varchar.
email	varchar(100)	Στο πεδίο αυτό θα αποθηκεύεται το email του χρήστη.
date_of_registration	date	Πρόκειται για την ημερομηνία εγγραφής του χρήστη στο κοινωνικό δίκτυο.
gender	tinyint(1)	Το φύλο του χρήστη. Το φύλο αντιστοιχεί στο βιολογικό προσδιορισμό του χρήστη και είναι 0 - false αν αντιστοιχεί στο ανδρικό φύλο και 1 - true αν αντιστοιχεί στο γυναικείο.
picture_path	varchar(100)	Στο πεδίο αυτό αποθηκεύεται το path της εικόνας που επιλέγει ο χρήστης για το προφίλ του, το avatar του όπως ονομάζεται συνήθως. Θα ήταν δυνατό βέβαια να αποθηκεύεται και το όνομα της φωτογραφίας μαζί με το path σε ένα κοινό πεδίο, όμως με αυτόν τον τρόπο είχαμε μεγαλύτερη ευελιξία π.χ. να αλλάξουμε το μονοπάτι των αποθηκευμένων φωτογραφιών. Αυτή η ευελιξία αποδείχθηκε χρήσιμη σε διάφορα σημεία της υλοποίησης.
profile_pic	varchar(100)	Στο πεδίο αυτό αποθηκεύεται το όνομα της εικόνας που επιλέγει ο χρήστης για το προφίλ του.

bio	text	Στο πεδίο αυτό θα αποθηκεύεται η περιγραφή (σύντομο βιογραφικό) που εισάγει ο χρήστης για τον εαυτό του.
status	varchar(7)	Στο πεδίο αυτό αποθηκεύεται η κατάσταση λογαριασμού του χρήστη. Το πεδίο αυτό είναι τύπου varchar με μέγιστο μήκος τους 7 χαρακτήρες και default τιμή offline. Η τιμή αυτή αλλάζει online όταν ο χρήστης συνδεθεί για πρώτη φορά στο κοινωνικό δίκτυο. Εντός της εφαρμογής, ο χρήστης έχει τη δυνατότητα να απενεργοποιεί το λογαριασμό του οποτεδήποτε το επιθυμεί (αν δεν επιθυμεί να τον ενοχλούν για παράδειγμα), και άρα η τιμή αυτή θα αλλάζει από online σε offline. Σε μελλοντική έκδοση είναι δυνατόν να προστεθούν και άλλες καταστάσεις όπως away, do not disturb κ.ά. ή ακόμη και προσαρμοσμένες καταστάσεις οι οποίες θα πληκτρολογούνται από τον χρήστη.

Παρατήρηση: Τα υποχρεωτικά πεδία που θα πρέπει να έχουν συμπληρωμένα όλοι οι εγγεγραμμένοι χρήστες του δικτύου είναι το id τους - το οποίο παράγεται αυτόματα από τη βάση δεδομένων λόγω του AUTO_INCREMENT της στήλης id -, το ονοματεπώνυμο τους, το όνομα χρήστη, το email, ο κωδικός πρόσβασης τους, το φύλο τους και η ημερομηνία γέννησης τους. Τα υπόλοιπα πεδία που αφορούν την φωτογραφία, το bio του χρήστη κ.λπ. είναι προαιρετικά και θα συμπληρώνονται από τον ίδιο τον χρήστη, αν και όποτε το επιθυμεί. Ορισμένα από αυτά θα έχουν εξ αρχής μια ορισμένη default τιμή που θα έχουμε ορίσει εμείς οι ίδιοι (ένα τέτοιο πεδίο θα αποτελεί για παράδειγμα η profile_pic του χρήστη, έτσι ώστε όταν συνδέεται στο δίκτυο να υπάρχει ήδη μια εικόνα, αντί ενός άδειου λευκού πλαισίου και να μην τον υποχρεώνουμε να εισάγει μία φωτογραφία του).

4.1.1.2 Πίνακας images

Ο πίνακας images χρησιμοποιείται για την αποθήκευση των στοιχείων των φωτογραφιών - post στην εφαρμογή χρηστών και αποτελείται από τα παρακάτω πεδία:

Όνομα πεδίου	Τύπος	Σχόλια
photo_id	int(11)	Στο πεδίο αυτό παράγεται από τη βάση δεδομένων ένα μοναδικό αναγνωριστικό για κάθε εικόνα - post του κοινωνικού δικτύου. Το πεδίο αυτό είναι το Primary Key του πίνακα images. Είναι επίσης auto increment και παίρνει αύξουσα τιμή για κάθε νέα εγγραφή που εισάγεται.
photo_name	varchar(100)	Στο πεδίο αυτό αποθηκεύεται το όνομα της εικόνας του post που έχει δημοσιεύσει ο χρήστης στο κοινωνικό δίκτυο.

photo_path	varchar(100)	Στο πεδίο αυτό αποθηκεύεται το path της εικόνας του post που έχει δημοσιεύσει ο χρήστης.
photo_likes	bigint(20)	Στο πεδίο αυτό αποθηκεύεται ο αριθμός των likes που έχει κάθε φωτογραφία. Βέβαια διατηρείται σε ξεχωριστό πίνακα ποιος χρήστης έκανε like σε ποια φωτογραφία, το πότε και άλλες πιο λεπτομερής πληροφορίες, όμως η αποθήκευση του πλήθους των likes σε ένα πεδίο ήταν αρκετά χρήσιμη για επιτάχυνση της προβολής της πληροφορίας και την ταχεία ταξινόμηση των φωτογραφιών με βάση τα likes.
username	varchar(100)	Το όνομα του χρήστη που ανέβασε την φωτογραφία.
photo_tag	varchar(100)	Τα hashtags που έχει κάθε φωτογραφία. Μέσω αυτών επιτελείται και η λειτουργία της αναζήτησης. Το μέγιστο μήκος μπορεί να είναι περιοριστικό για κάποιες ακραίες περιπτώσεις. Μία εναλλακτική υλοποίηση θα ήταν να αποθηκεύονται τα hashtags ως οντότητες και να εισαχθεί η χρήση ενός πρόσθετου πίνακα για αποθήκευση της σύνδεσης μεταξύ φωτογραφιών και hashtags.
date_posted	datetime	Η ημερομηνία που ανέβηκε το post.

4.1.1.3 Πίνακας post_likes

Ο πίνακας post_likes χρησιμοποιείται για την αποθήκευση των στοιχείων των likes που μπορεί να έχει κάποια δημοσιευμένη φωτογραφία στην εφαρμογή χρηστών και αποτελείται από τα παρακάτω πεδία:

Όνομα πεδίου	Τύπος	Σχόλια
post_likes_id	int(11)	Στο πεδίο αυτό παράγεται από τη βάση δεδομένων ένα μοναδικό αναγνωριστικό για κάθε εικόνα - post του κοινωνικού δικτύου. Το πεδίο αυτό είναι το Primary Key του πίνακα post_likes. Είναι επίσης auto increment και παίρνει αύξουσα τιμή για κάθε νέα εγγραφή που εισάγεται.
liked_by_user	int(11)	Στο πεδίο αυτό αποθηκεύεται το id του χρήστη που δήλωσε την αρέσκεια του (like).
posted_photo_id	int(11)	Στο πεδίο αυτό αποθηκεύεται το id της εικόνας στην οποία έχει γίνει like.
time	datetime	Η ημερομηνία που δηλώθηκε η αρέσκεια προς την φωτογραφία (like).

4.1.1.4 Πίνακας post_comments

Ο πίνακας post_comments χρησιμοποιείται για την αποθήκευση των στοιχείων των σχολίων (comments) που μπορεί να έχει κάποια δημοσιευμένη φωτογραφία στην εφαρμογή χρηστών και αποτελείται από τα παρακάτω πεδία:

Όνομα πεδίου	Τύπος	Σχόλια
post_comments_id	int(11)	Στο πεδίο αυτό παράγεται από τη βάση δεδομένων ένα μοναδικό αναγνωριστικό για κάθε comment του κοινωνικού δικτύου. Το πεδίο αυτό είναι το Primary Key του πίνακα post_comments. Είναι επίσης auto increment και παίρνει αύξουσα τιμή για κάθε νέα εγγραφή που εισάγεται.
post_id	int(11)	Στο πεδίο αυτό αποθηκεύεται το id της δημοσίευσης στην οποία έγινε το σχόλιο.
user_id	int(11)	Στο πεδίο αυτό αποθηκεύεται το id του χρήστη που σχολίασε τη δημοσίευση.
comment_text	text	Το κείμενο του comment, αυτό δηλαδή που πληκτρολόγησε ο χρήστης στο πεδίο εισόδου. Η κωδικοποίηση σε αυτό το πεδίο είναι utf8_mb4, έτσι ώστε να δέχεται οποιοδήποτε unicode χαρακτήρα, ακόμα και emoticons που έχει την δυνατότητα ο χρήστης να εισάγει και φυσικά ελληνικούς ή άλλους χαρακτήρες unicode.
time_commented	datetime	Η ημερομηνία που ανέβηκε ο σχολιασμός της δημοσίευσης (comment).

4.1.1.5 Πίνακας comment_likes

Ο πίνακας χρησιμεύει για την αποθήκευση στοιχείων για τα likes που ενδεχομένως να έχει ένα σχόλιο (comment) και αποτελείται από τα παρακάτω πεδία:

Όνομα πεδίου	Τύπος	Σχόλια
comment_like_id	int(11)	Στο πεδίο αυτό παράγεται από τη βάση δεδομένων ένα μοναδικό αναγνωριστικό για κάθε like που γίνεται σε comment του κοινωνικού δικτύου. Το πεδίο αυτό είναι το Primary Key του πίνακα. Είναι επίσης auto increment και παίρνει αύξουσα τιμή για κάθε νέα εγγραφή που εισάγεται
liked_by	int(11)	Στο πεδίο αυτό αποθηκεύεται το id του χρήστη που έκανε like στο comment.
comment_id	int(11)	Στο πεδίο αυτό αποθηκεύεται το id του comment στο οποίο δηλώθηκε αρέσκεια.

time_liked	datetime	Η ημερομηνία που δηλώθηκε η αρέσκεια προς το comment.

4.1.1.6 Πίνακας comment_replies

Στην εφαρμογή υπάρχει επίσης η δυνατότητα άμεσης απάντησης σε σχόλιο που έχει κάνει κάποιος άλλος χρήστης, (όπως γίνεται π.χ. στο Facebook ή στο YouTube). Ο πίνακας "comment_replies" χρησιμεύει για την αποθήκευση των απαντήσεων (που είναι και αυτές σχόλια) σε σχόλια χρηστών και αποτελείται από τα παρακάτω πεδία:

Όνομα πεδίου	Τύπος	Σχόλια
reply_id	int(11)	Στο πεδίο αυτό παράγεται από τη βάση δεδομένων ένα μοναδικό αναγνωριστικό για κάθε comment του κοινωνικού δικτύου. Το πεδίο αυτό είναι το Primary Key του πίνακα. **comment_replies**. Είναι επίσης auto increment και παίρνει αύξουσα τιμή για κάθε νέα εγγραφή που εισάγεται.
comment_id	int(11)	Στο πεδίο αυτό αποθηκεύεται το id του comment στο οποίο ο χρήστης επιθυμούσε να απαντήσει (reply).
user_id	int(11)	Στο πεδίο αυτό αποθηκεύεται το id του χρήστη που απάντησε σε comment.
comment_text	text	Το κείμενο του comment, αυτό δηλαδή που πληκτρολόγησε ο χρήστης στο πεδίο εισόδου.
time_commented	datetime	Η ημερομηνία που ανέβηκε ο σχολιασμός του comment (reply).

4.1.1.7 Πίνακας hashtags

Ο πίνακας επιτρέπει την αποθήκευση των hashtags με έναν ιδιαίτερα δομημένο τρόπο. Τα πεδία του πίνακα παρατίθενται παρακάτω:

Όνομα πεδίου	Τύπος	Σχόλια
hashtag_id	int(11)	Το μοναδικό αναγνωριστικό για κάθε hashtag της δημοσίευσης. Το πεδίο αυτό είναι το Primary Key του πίνακα.
hashtag	varchar(255)	Η τιμή του hashtag.
post_id	int(11)	Το αναγνωριστικό id της δημοσίευσης που έχει το hashtag.
user_id	int(11)	Το αναγνωριστικό id του χρήστη που έβαλε το hashtag σε αυτή τη δημοσίευση.

Παρατήρηση: Αξίζει να αναφερθεί ότι αν και θεωρήσαμε αρχικά ότι η δημιουργία αυτού του πίνακα θα μας δώσει βελτιωμένη δυνατότητα αποθήκευσης των hashtags, τελικά ήταν υπερβολικά πολύπλοκη για την προοριζόμενη χρήση. Εν τέλει, τα hashtags έγιναν απλά πεδίο της δημοσίευσης και η διαχείρισή τους γίνεται αποκλειστικά προγραμματιστικά. Ο πίνακας ωστόσο δεν διαγράφηκε γιατί θεωρείται ότι ενδέχεται να μας χρησιμεύσει για μελλοντικές βελτιώσεις.

4.1.1.8 Πίνακας ip_mac_addresses

Αυτός ο πίνακας χρησιμεύει για να κρατήσουμε τις διευθύνσεις IP των χρηστών, το λειτουργικό τους σύστημα και άλλα χαρακτηριστικά που θα μας επιτρέψουν να διαχειριστούμε το κομμάτι της ασφάλειας του ιστότοπου το οποίο θα αναλύσουμε σε επόμενη ενότητα.

Όνομα πεδίου	Τύπος	Σχόλια
im_id	int(50)	Το αναγνωριστικό id για τον πίνακα. AUTO-INCREMENT
IP_address	varchar(50)	Η διεύθυνση IP του χρήστη.
user_name	varchar(255)	Το όνομα του χρήστη.
login_date	datetime	Πότε συνδέθηκε ο χρήστης τελευταία φορά με αυτή τη συσκευή από τη συγκεκριμένη διεύθυνση.
mobile	tinyint(1)	0 ή 1 ανάλογα αν η συσκευή είναι κινητή ή όχι για να διαχειριστούμε καλύτερα τις συσκευές IoT. Κάτι που βέβαια προκύπτει και από το πεδίο του λειτουργικού συστήματος τις περισσότερες φορές.
OS	varchar(255)	Το λειτουργικό σύστημα της συσκευής του χρήστη.
Browser	varchar(255)	Το πρόγραμμα περιηγητή του χρήστη.

Σημείωση 1: Το λειτουργικό σύστημα δεν είναι πάντοτε γνωστό ή μπορεί να μην το γνωρίζουμε με ασφάλεια. Η εξαγωγή του λειτουργικού συστήματος όπως και του περιηγητή από το signature του browser αποτελεί μόνο εκτίμηση. Μπορεί να αλλαχτεί από έναν έμπειρο χρήστη. Παρ' όλα αυτά είναι μια μέθοδος να διαχωρίζουμε τους εγκεκριμένους χρήστες.

Σημείωση 2: Αν και η διεύθυνση IP είναι εύκολο να αλλάξει κρατάμε τη διεύθυνση IP από όπου έχει συνδεθεί η συσκευή. Αν αυτή σε συνδυασμό με κάποια από τα πεδία αλλάξουν ειδοποιούμε τον χρήστη με χρήση email για ενδεχόμενο κενό ασφαλείας. Δεν πρόκειται για την απόλυτη λειτουργία ασφαλείας, και μπορεί να υπάρξουν false positives, αλλά γενικά μπορεί να προβλεφθούν αρκετά προβλήματα. Η όχληση του χρήστη από αυτά τα e-mail κρίνεται συνολικά περιορισμένη και γενικά η προκύπτουσα ωφέλεια στην ασφάλεια είναι σημαντικά μεγαλύτερη από το κόστος των οχλήσεων.

4.1.1.9 Πίνακας relationship

Αυτός ο πίνακας χρησιμεύει για να κρατάμε τις σχέσεις μεταξύ των χρηστών, κυρίως το ποιοι είναι φίλοι, ποιος έκανε αίτημα φιλίας σε άλλον κ.λπ. Χρησιμεύει όμως και για

άλλες ενέργειες που δηλώνουν τη σχέση μεταξύ χρηστών, όπως π.χ μπλοκάρισμα (block) ενός χρήστη.

Όνομα πεδίου	Τύπος	Σχόλια
relationship_id	int(11)	Το id για τον πίνακα.
user_one_id	int(11)	Το id του χρήστη που έκανε την ενέργεια π.χ αυτού που έστειλε ένα αίτημα φιλίας.
user_two_id	int(11)	Ο id του χρήστη που δέχτηκε την ενέργεια π.χ αυτού που έλαβε ένα αίτημα φιλίας.
status	tinyint(4)	Η κατάσταση της σχέσης. Αν είναι 0 σημαίνει ότι έχει σταλεί ένα αίτημα φιλίας. Αν είναι 1 σημαίνει ότι το αίτημα έχει γίνει αποδεκτό. Αν είναι 2 ότι έχει γίνει block κ.λπ. Για ευκολία θα μπορούσαν αντί για αριθμοί να εισάγονταν αλφαριθμητικά π.χ «pending» αν το αίτημα έχει σταλεί, αλλά δεν έχει γίνει ακόμη κάποια ενέργεια ή «accepted» αν έχει γίνει αποδεκτό. Επιλέχτηκαν αριθμοί για επίτευξη μεγαλύτερης απόδοσης και ταχύτητας.
action_user_id	int(11)	Πρόκειται για τον χρήστη που έκανε τελευταίος μία ενέργεια δηλαδή αυτόν που π.χ. αποδέχτηκε ένα αίτημα. Η γνώση αυτού του χρήστη ήταν αρκετά σημαντική στην υλοποίηση μας.

Σημείωση: Το action_user_id αντιπροσωπεύει το id του χρήστη ο οποίος εκτέλεσε την πιο πρόσφατη ενέργεια. Για παράδειγμα, αν ο χρήστης με id 1 έχει στείλει ένα αίτημα στο χρήστη με το id 2, το action user id θα γίνει 1. Οπότε ο χρήστης που έστειλε το αίτημα είναι ο 1. Αν ο χρήστης με id 2 έχει μπλοκάρει τον χρήστη με id 3, τότε 2 θα είναι και το action user id. Με αυτό τον τρόπο γνωρίζουμε ποιος χρήστης έκανε το μπλοκάρισμα. Μόνο αυτός ο χρήστης (με id 2) π.χ. στο παράδειγμα μας μπορεί να ξεκινήσει τη διαδικασία ξεμπλοκαρίσματος. Ο άλλος χρήστης πλέον δεν μπορεί να κάνει κάτι για αυτό.

Το action_user_id συνεπώς σε συνδυασμό με την κατάσταση status, χρησιμεύει για να αποφασιστεί ποιος χρήστης έχει την άδεια να εκτελέσει μία ενέργεια και ποιος όχι πάνω σε μία κατάσταση. Αν δεν υπήρχε το action_user_id, θα έπρεπε με κάποιο άλλο τρόπο να συμπεράνουμε ποιος χρήστης έχει δικαίωμα να κάνει κάποια ενέργεια.

4.1.1.10 Πίνακας saved_posts

Στο κοινωνικό δίκτυο υπάρχει η δυνατότητα ένας χρήστης να δημιουργήσει σελιδοδείκτες προς επιλεγμένες δημοσιεύσεις. Οι δημοσιεύσεις αυτές εμφανίζονται στο προφίλ του και μπορεί αργότερα να τις επισκεφτεί ξανά, όποτε αυτός θελήσει. Με τη χρήση αυτού του πίνακα αποθηκεύουμε το id του χρήστη που δημιούργησε τον σελιδοδείκτη, προς ποια δημοσίευση δείχνει ο σελιδοδείκτης και πότε έγινε η δημιουργία του σελιδοδείκτη.

Όνομα πεδίου	Τύπος	Σχόλια
saved_post_id	int(11)	Το αναγνωριστικό για τον πίνακα.
user_id	int(11)	Το αναγνωριστικό του χρήστη που αποθήκευσε τη δημοσίευση.
post_id	int(11)	Το αναγνωριστικό της δημοσίευσης.
time_saved	datetime	Πότε έγινε αποθήκευση της δημοσίευσης.

4.1.1.11 Πίνακας chat_message

Στην εφαρμογή υπάρχει τέλος η δυνατότητα ανταλλαγής μηνυμάτων (chatting) με άλλους χρήστες, συγκεκριμένα μόνο μεταξύ φίλων έτσι ώστε να διατηρηθεί και ένα επίπεδο ασφάλειας και ιδιωτικότητας. Στον πίνακα chat_message, αποθηκεύονται τα μηνύματα μεταξύ χρηστών, για μελλοντική αναφορά. Συγκεκριμένα αποθηκεύονται στοιχεία για τον αποστολέα και τον παραλήπτη, το κείμενο του μηνύματος αυτού και πότε συνέβη αυτή η ανταλλαγή μηνύματος.

Όνομα πεδίου	Τύπος	Σχόλια
chat_message_id	int(11)	Το αναγνωριστικό για τον πίνακα.
by_user_id	int(11)	Το αναγνωριστικό του χρήστη που στέλνει το μήνυμα.
to_user_id	int(11)	Το αναγνωριστικό του χρήστη που λαμβάνει το μήνυμα.
message	text	Το κείμενο του μηνύματος.
time_sent	datetime	Η ημερομηνία και η ώρα που στάλθηκε το μήνυμα.
status	enum('read', 'unread')	Το status χρησιμεύει για να κρατάμε την κατάσταση του μηνύματος, αν δηλαδή το μήνυμα έχει διαβαστεί ή όχι από τον παραλήπτη.

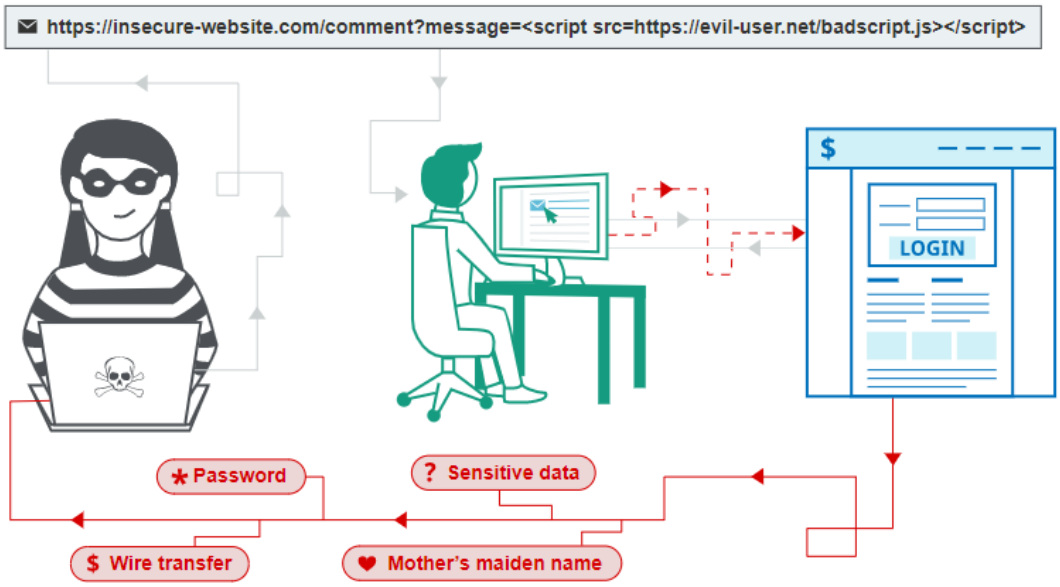
5 Κυριότερες επιθέσεις που μπορεί να δεχτεί το κοινωνικό δίκτυο

Στην παρούσα ενότητα περιγράφουμε ένα σύνολο από συνήθεις επιθέσεις που μπορεί να δεχθεί το σύστημα και τα αντίστοιχα μέτρα προστασίας.

5.1 Cross-site scripting (XSS)

Το Cross-site scripting (επίσης γνωστό ως XSS) (30) είναι μια ευπάθεια ασφαλείας που επιτρέπει σε έναν επιτιθέμενο να θέσει σε κίνδυνο τις αλληλεπιδράσεις που έχουν οι χρήστες με μια εφαρμογή. Επιτρέπει στον επιτιθέμενο να παρακάμψει ελέγχους πρόσβασης, όπως την πολιτική κοινής προέλευσης (Same-Origin Policy), η οποία έχει σχεδιαστεί για να διαχωρίζει διαφορετικούς ιστότοπους μεταξύ τους. Στις ευπάθειες αυτές συνήθως ο επιτιθέμενος καταφέρνει να υποκλέψει την ταυτότητα ενός χρήστη-θύματος αναφορικά με τις αλληλεπιδράσεις του με τον ιστοχώρο, αποκτώντας έτσι τη δυνατότητα να πραγματοποιήσει οποιεσδήποτε ενέργειες ο χρήστης-θύμα έχει τη δυνατότητα να εκτελέσει. Συνεπώς αποκτά τελικά πρόσβαση στα δεδομένα του χρήστη και λαμβάνει δυνατότητες πραγματοποίησης ενεργειών. Αν το θύμα έχει προνομιακή πρόσβαση στην εφαρμογή (π.χ. είναι admin), τότε ο εισβολέας μπορεί ακόμη και να αποκτήσει πλήρη έλεγχο όλων των λειτουργιών και των δεδομένων της εφαρμογής.

Το Cross-site scripting προϋποθέτει την ύπαρξη ευπαθειών σε έναν ιστότοπο, έτσι ώστε να επιτρέψει στον επιτιθέμενο να καταχωρήσει κακόβουλη Javascript (Javascript Injection), η οποία σε μεταγενέστερο σημείο θα επιστραφεί στον χρήστη-θύμα. Όταν ο κακόβουλος κώδικας εκτελείται μέσα στο πρόγραμμα περιήγησης του θύματος, ο εισβολέας μπορεί να έχει μια άμεση αλληλεπίδραση με την εφαρμογή, την οποία ελέγχει μέσω της λογικής του κώδικα. Στη συνέχεια μπορεί να έχει πρόσβαση, παθητικά ή ενεργά, σε ευαίσθητους πόρους του προγράμματος περιήγησης που σχετίζεται με την εφαρμογή π.χ. cookies, αναγνωριστικά περιόδου σύνδεσης (session IDs) και άλλα.



Εικόνα 26. Σχήμα Επίθεσης Cross Site Scripting

5.1.1 Τύποι επίθεσης XSS

- Reflected XSS
- Stored Xss
- DOM-based XSS

5.1.1.1 Reflected cross-site scripting

Είναι η απλούστερη μορφή για cross-site scripting. Προκύπτει όταν μια εφαρμογή λαμβάνει δεδομένα από ένα αίτημα HTTP και περιλαμβάνει αυτά τα δεδομένα μαζί με την απάντηση με μη ασφαλή τρόπο. Ένα παράδειγμα ακολουθεί παρακάτω:

```
https://insecure-website.com/status?message=All+is+well.

<p>Status: All is well.</p>
```

Η εφαρμογή δεν εκτελεί κάποια επεξεργασία των δεδομένων και έτσι ο επιτιθέμενος μπορεί εύκολα να εξαπολύσει την επίθεση του με το παρακάτω τρόπο:

```
https://insecure-
website.com/status?message=<script>/*+Bad+stuff+here...+*/</script>

<p>Status: <script>/* Bad stuff here... */</script></p>
```

Αν ο χρήστης επισκεφθεί τον σύνδεσμο URL που έχει φτιάξει ο κακόβουλος χρήστης, τότε το script θα εκτελεστεί στο browser του χρήστη. Σε αυτό το σημείο το script μπορεί να ανακτήσει δεδομένα στα οποία ο χρήστης έχει πρόσβαση. Ο κακόβουλος χρήστης μπορεί να έχει ενσωματώσει το URL σε σελίδες που περιέχουν απλές παραπομπές στον

ευπαθή ιστότοπο ή υπόσχονται πρόσβαση σε περιεχόμενο που αλλιώς είναι κλειδωμένο κ.λπ.

5.1.1.2 Stored cross-site scripting

Το Stored XSS (επίσης γνωστό ως persistent ή second-order XSS) προκύπτει όταν μια εφαρμογή λαμβάνει δεδομένα από μια πηγή η οποία δεν είναι αξιόπιστη και περιλαμβάνει αυτά τα δεδομένα στην HTTP απάντηση με μη ασφαλή τρόπο.

Τα εν λόγω δεδομένα ενδέχεται να καταχωρηθούν στην εφαρμογή μέσω αιτημάτων HTTP. Για παράδειγμα, σχόλια σε μια ανάρτηση ιστολογίου, ψευδώνυμα χρήστη σε ένα chat room ή στοιχεία επικοινωνίας σχετικά με μια παραγγελία ενός πελάτη. Σε άλλες περιπτώσεις, τα δεδομένα ενδέχεται να προέρχονται από άλλες μη αξιόπιστες πηγές. Για παράδειγμα, μια εφαρμογή webmail που εμφανίζει μηνύματα που λαμβάνονται μέσω SMTP, μια εφαρμογή μάρκετινγκ που εμφανίζει δημοσιεύσεις κοινωνικών μέσων ή μια εφαρμογή παρακολούθησης δικτύου που εμφανίζει δεδομένα από την κυκλοφορία πακέτων δικτύου.

Παρακάτω βλέπουμε ένα παράδειγμα όπου ένας πίνακας μηνυμάτων επιτρέπει στους χρήστες να καταχωρήσουν τα μηνύματά τους τα οποία εμφανίζονται σε άλλους χρήστες.

```
<p>Hello, this is my message!</p>
```

Η εφαρμογή δεν εκτελεί κάποια επεξεργασία των δεδομένων και έτσι ο επιτιθέμενος μπορεί εύκολα να στείλει μήνυμα το οποίο επιτίθεται σε άλλους χρήστες με το παρακάτω τρόπο:

```
<p><script>/* Bad stuff here... */</script></p>
```

Το μήνυμα αυτό θα αποθηκευθεί στη βάση δεδομένων και στη συνέχεια θα αποσταλεί στις εφαρμογές πλοήγησης όλων των χρηστών που θα ανακτήσουν αυτό το μήνυμα, με αποτέλεσμα να εκτελεστεί ο κακόβουλος κώδικας.

5.1.1.3 DOM-based cross-site scripting

Το DOM-based cross-site scripting (γνωστό και ως DOM XSS) προκύπτει όταν μια εφαρμογή περιέχει κάποια JavaScript από την πλευρά του πελάτη που επεξεργάζεται δεδομένα από μια μη αξιόπιστη πηγή με μη ασφαλή τρόπο, συνήθως γράφοντας τα δεδομένα πίσω στο DOM.

Στο ακόλουθο παράδειγμα, μια εφαρμογή χρησιμοποιεί κάποια JavaScript για να διαβάσει την τιμή από ένα πεδίο εισαγωγής και να γράψει αυτήν την τιμή σε ένα στοιχείο εντός του HTML:

```
var search = document.getElementById('search').value;
var results = document.getElementById('results');
results.innerHTML = 'You searched for: ' + search;
```

Εάν ο επιτιθέμενος μπορεί να ελέγξει την τιμή του πεδίου εισαγωγής, μπορεί εύκολα να δημιουργήσει μια κακόβουλη τιμή που προκαλεί την εκτέλεση του δικού του script:

```
You searched for: <img src=1 onerror='/* Bad stuff here... */'>
```

Σε μια τυπική περίπτωση, το πεδίο εισαγωγής θα συμπληρώνεται από μέρος του αιτήματος HTTP, όπως ένα URL query string parameter, επιτρέποντας στον εισβολέα να

παραδώσει μια επίθεση χρησιμοποιώντας μια κακόβουλη διεύθυνση URL, με τον ίδιο τρόπο που αντικατοπτρίζεται το XSS.

5.1.2 Κίνδυνοι που προκύπτουν από το XSS

- Ο επιτιθέμενος αντιποιείται (παριστάνει) τον χρήστη-θύμα.
- Ο επιτιθέμενος μπορεί να εκτελέσει οποιαδήποτε ενέργεια που μπορεί να εκτελέσει ο χρήστης-θύμα.
- Ο επιτιθέμενος έχει πρόσβαση σε οποιαδήποτε δεδομένα στα οποία ο χρήστης-θύμα έχει πρόσβαση
- Ο επιτιθέμενος μπορεί να καταγράψει τα διαπιστευτήρια σύνδεσης του χρήστη.
- Εκτελεί βανδαλισμό του ιστότοπου (Web defacement). Σε μία επίθεση βανδαλισμού, κακόβουλα μέρη διεισδύουν σε έναν ιστότοπο και αντικαθιστούν το περιεχόμενο του ιστότοπου με τα δικά τους μηνύματα. Τα μηνύματα μπορούν να μεταφέρουν ένα πολιτικό ή θρησκευτικό μήνυμα, βωμολοχίες ή άλλο ακατάλληλο περιεχόμενο που θα ενοχλούσε τους ιδιοκτήτες των ιστότοπων αυτών ή μια ειδοποίηση ότι ο ιστότοπος έχει παραβιαστεί από μια συγκεκριμένη ομάδα επιτιθέμενων. Οι περισσότεροι ιστότοποι και εφαρμογές ιστού αποθηκεύουν δεδομένα σε αρχεία περιβάλλοντος ή διαμόρφωσης, τα οποία επηρεάζουν το περιεχόμενο που εμφανίζεται στον ιστότοπο ή καθορίζουν πού βρίσκονται τα πρότυπα (templates) και το περιεχόμενο της σελίδας. Οι απροσδόκητες αλλαγές σε αυτά τα αρχεία μπορεί να σηματοδοτούν μια επίθεση. (31)
- Ο επιτιθέμενος μπορεί να εισάγει λειτουργικότητα τύπου δούρειου ίππου (trojan horse) στον ιστότοπο.

5.1.3 Επιπτώσεις από ευπάθειες XSS

Οι πραγματικές επιπτώσεις μιας επίθεσης XSS εξαρτάται γενικά από τη φύση της εφαρμογής, τη λειτουργικότητα, τα δεδομένα της και την κατάσταση του παραβιασμένου χρήστη. Για παράδειγμα:

- Σε μια εφαρμογή brochureware, όπου όλοι οι χρήστες είναι ανώνυμοι και όλες οι πληροφορίες είναι δημόσιες, ο αντίκτυπος θα είναι συχνά ελάχιστος.
- Σε μια εφαρμογή που περιέχει ευαίσθητα δεδομένα, όπως τραπεζικές συναλλαγές, email ή αρχεία υγειονομικής περίθαλψης, ο αντίκτυπος θα είναι συνήθως σοβαρός.
- Εάν ο χρήστης έχει αυξημένα δικαιώματα στην εφαρμογή, τότε ο αντίκτυπος θα είναι γενικά κρίσιμος, επιτρέποντας στον εισβολέα να πάρει τον πλήρη έλεγχο της ευάλωτης εφαρμογής και να θέσει σε κίνδυνο όλους τους χρήστες και τα δεδομένα τους.

5.1.4 Αποτροπή επιθέσεων τύπου XSS

Ο τρόπος αποφυγής μιας επίθεσης XSS μπορεί να είναι πολύ δύσκολος, ανάλογα με την πολυπλοκότητα της εφαρμογής και τους τρόπους με τους οποίους διαχειρίζεται δεδομένα που ελέγχονται από τον χρήστη.

Σε γενικές γραμμές, η αποτελεσματική πρόληψη της ευπάθειας XSS είναι πιθανό να περιλαμβάνει συνδυασμό των ακόλουθων μέτρων:

- *Filter input on arrival*: Στο σημείο όπου λαμβάνεται η είσοδος χρήστη, φιλτράρεται όσο το δυνατόν αυστηρότερα με βάση την αναμενόμενη ή έγκυρη εισαγωγή. Για παράδειγμα μπορεί να απορρίπτονται συμβολοσειρές που αντιστοιχούν σε HTML tags ή ειδικοί χαρακτήρες (αν δεν επιτρέπεται η χρήση τους) ή να μετατρέπονται οι ειδικοί χαρακτήρες σε HTML entities.
- *Encode data on output*: Στο σημείο όπου τα ελεγχόμενα δεδομένα εξάγονται στις απαντήσεις του HTTP, η έξοδος κωδικοποιείται για να αποτραπεί η ερμηνεία τους ως ενεργό περιεχομένο (π.χ. μέσω χρήσης HTML entities για τους ειδικούς χαρακτήρες). Ανάλογα με το εξαγόμενο περιεχόμενο, αυτό ενδέχεται να απαιτείται η εφαρμογή συνδυασμών κωδικοποίησης HTML, URL, JavaScript, CSS.
- *Χρήση κατάλληλων response headers*: Για να αποτραπεί το XSS στις απαντήσεις του HTTP που δε πρέπει να περιέχουν HTML ή JavaScript είναι δυνατόν να χρησιμοποιηθούν οι επικεφαλίδες (headers) Content-Type και οι X-Content-Type για να διασφαλιστεί ότι τα προγράμματα περιήγησης ερμηνεύουν τις απαντήσεις με τον τρόπο που επιθυμεί ο ιδιοκτήτης/κάτοχος του sitekαι όχι με τον τρόπο που επιθυμεί ο επιτιθέμενος.
- Πολιτική ασφαλείας του περιεχομένου (Content Security Policy): Ως τελευταία γραμμή άμυνας, μπορεί να χρησιμοποιηθεί η ασφάλειας περιεχομένου (CSP) για να περιοριστούν οι επιπτώσεις τυχόν ευπαθειών XSS που εξακολουθούν να εμφανίζονται. (30)

5.1.5 Άμυνα έναντι επιθέσεων XSS

5.1.5.1 Escaping user inputs

Τα δεδομένα που λαμβάνονται σε μια διαδικτυακή εφαρμογή πρέπει να εξετάζονται και να «γίνονται» ασφαλή πριν γίνουν διαθέσιμα στον τελικό χρήστη. Αυτό μπορεί να υλοποιηθεί με διαφυγή των ειδικών χαρακτήρων της εισόδου (escaping data ή escaping data inputs). Αυτή η μεθοδολογία εμποδίζει την παρερμηνεία των ληφθέντων δεδομένων και τη χρήση τους με κάποιον κακόβουλο τρόπο. Η εφαρμογή είναι σχεδιασμένη με τέτοιο τρόπο ώστε να φιλτράρονται τα δεδομένα και να μην αφήνει να περνάνε αμετάφραστοι χαρακτήρες που μπορούν δημιουργήσουν πρόβλημα στην εφαρμογή (π.χ. οι χαρακτήρες ‘<‘ και ’>’). Αν μια σελίδα δεν αφήνει τους χρήστες να εισάγουν τον δικό τους κώδικα, είναι εύκολο να διαχειριστούμε τέτοιες περιπτώσεις (escape) σε κώδικα JavaScript & HTML. Αλλά αν η εφαρμογή περιλαμβάνει χώρους για σχόλια χρηστών (comment boxes) όπως για παράδειγμα ένα φόρουμ τότε έχουμε πολύ λίγες επιλογές. Σε αυτήν την περίπτωση μπορούμε μόνο να επιλέξουμε προσεκτικά, ποιες οντότητες HTML θέλουμε να επιτρέπουμε να εμφανίζονται σε αυτή την εφαρμογή.

5.1.5.2 Input Validation

Η επικύρωση των δεδομένων των χρηστών εξασφαλίζει ότι η εφαρμογή παρέχει αξιόπιστα τα δεδομένα και διαγράφει τα μη αξιόπιστα ή κακόβουλα, αποτρέποντάς τα από το να βλάψουν τη βάση δεδομένων ή τα προσωπικά δεδομένα του χρήστη. Το whitelisting συνήθως σχετίζεται με το SQL Injection αλλά επιτρέποντας τους «ακίνδυνους» και μόνο χαρακτήρες είναι δυνατό να αποτρέψουμε και τις επιθέσεις XSS.

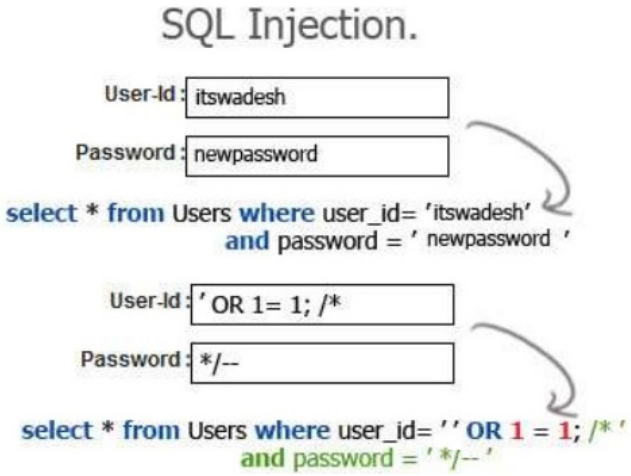
Έτσι, ένας από τους στόχους της επικύρωσης των δεδομένων είναι να αποτρέψουμε επιθέσεις XSS στους ιστοτόπους.

5.1.5.3 Sanitizing user input

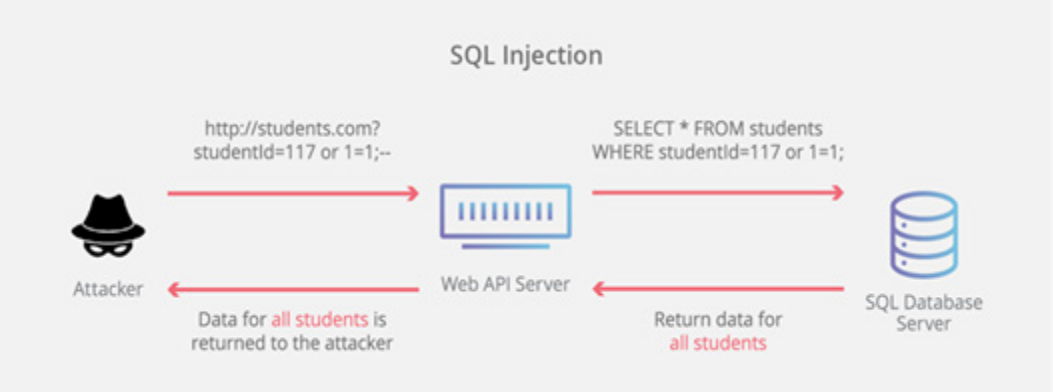
Η εξυγίανση (sanitization) των δεδομένων είναι η τροποποίηση των δεδομένων εισόδου για να βεβαιωθούμε ότι είναι έγκυρα. Αν αλλάξουμε τα μη έγκυρα (ίσως κακόβουλα) δεδομένα και τους δώσουμε έγκυρη μορφή, μπορούμε να βεβαιωθούμε ότι τα ληφθέντα δεδομένα δεν θα βλάψουν την εφαρμογή ή τη βάση δεδομένων μας. Η εξυγίανση της εισόδου είναι από τις πιο κοινές εργασίες σε διαδικτυακές εφαρμογές. Η PHP μας βοηθάει να κάνουμε αυτή την εργασία ευκολότερα, παρέχοντας μας εγγενή (native) φίλτρα που μπορούμε να χρησιμοποιήσουμε για να εντοπίσουμε μη έγκυρα e-mail, URLs, IP διευθύνσεις, κ.λπ. Ύστερα, χρησιμοποιώντας την οδηγία FILTER_SANITIZE_EMAIL, αφαιρούνται αυτοί οι ανεπιθύμητοι χαρακτήρες και μένει μόνο το εξυγιασμένο κείμενο που θέλουμε. Αυτές οι τρεις μέθοδοι που έχουμε αναφέρει έως τώρα, δεν είναι αρκετές από μόνες τους. Αλλά όταν υλοποιηθούν συνδυαστικά, μπορούν να προσφέρουν μια αξιόπιστη λύση για την καταπολέμηση των επιθέσεων XSS. (32)

5.2 SQL injection

Στην επίθεση τύπου SQL injection οι εισβολείς εισάγουν κακόβουλες εντολές SQL ή στοιχεία που θα οδηγήσουν στη διαμόρφωση κακόβουλων εντολών SQL στα πεδία εισαγωγής δεδομένων (entry fields) με σκοπό να εκτελεστούν έναντι της βάσης δεδομένων (back-end). Αυτό παρατηρείται ιδιαίτερα σε εφαρμογές που βασίζονται σε δεδομένα (data-driven applications). Με αυτές τις επιθέσεις, οι δράστες μπορούν να τροποποιήσουν ή να διαγράψουν υπάρχοντα δεδομένα και να δημιουργήσουν ψεύτικες ταυτότητες, ή συνολικότερα να τροποποιήσουν τις ερωτήσεις που υποβάλλονται στη βάση δεδομένων. Τα αποτελέσματα μπορεί να είναι πολύ σοβαρά, είναι δυνατόν να καταστραφούν όλα τα δεδομένα ή να αποκτήσει ο εισβολέας πλήρη έλεγχο της βάσης δεδομένων ή/και της εφαρμογής. Το Open Project Security Project (OWASP) ανέφερε το SQL injection ως την κορυφαία απειλή για την ασφάλεια στο διαδίκτυο στο αρχικό του έγγραφο OWASP 10 2018.



Εικόνα 27. Σύνδεση από χρήστη που εισάγει αναμενόμενες τιμές για τα στοιχεία (άνω μέρος), έναντι σύνδεσης όπου εισάγονται στοιχεία που μπορεί να προκαλέσουν SQLi (κάτω μέρος)



Εικόνα 28. Παραβίαση SQLi. Το query θα είναι πάντα αληθές και θα γίνει ανεπιθύμητη επιστροφή όλων των δεδομένων για τον πίνακα των μαθητών στον επιτιθέμενο

Στη συνέχεια καταγράφονται οι κύριες προσεγγίσεις για την αποτροπή SQL Injection στις εφαρμογές μας. Οι επιθέσεις του συγκεκριμένου τύπου είναι δυστυχώς πολύ συχνές, φαινόμενο που δεν αιτιολογείται, δεδομένου ότι είναι εξαιρετικά απλό να αποφευχθούν ευπάθειες SQL Injection χρησιμοποιώντας γνωστές τεχνικές, όπως αυτές που θα περιγράψουμε παρακάτω.

Όπως αναφέρθηκε παραπάνω, οι ευπάθειες SQL Injection εισάγονται όταν οι προγραμματιστές δημιουργούν δυναμικά ερωτήματα (queries), τα οποία περιλαμβάνουν είσοδο που παρέχεται από τον χρήστη και εκτελούνται στην βάση. Η αποφυγή ευπαθειών έγχυσης SQL μπορεί να επιτευχθεί αν οι προγραμματιστές:

i. σταματήσουν να γράφουν δυναμικά ερωτήματα χρησιμοποιώντας εναλλακτικές μεθόδους συμπερίληψης της εισόδου του χρήστη στις ερωτήσεις ή / και
ii. αποτρέψουν τις εισόδους των χρηστών, οι οποίες περιέχουν κακόβουλο SQL κώδικα.

Στη συνέχεια αναφέρονται οι κυριότερες τεχνικές για την αποτροπή ευπαθειών SQL Injection, με εφαρμογή των δύο προαναφερθεισών κατευθύνσεων. Αυτές οι τεχνικές μπορούν να χρησιμοποιηθούν σε σχεδόν οποιοδήποτε είδος γλώσσας προγραμματισμού με οποιονδήποτε τύπο βάσης δεδομένων. Υπάρχουν άλλοι τύποι βάσεων δεδομένων, όπως βάσεις δεδομένων XML, οι οποίες μπορεί να έχουν παρόμοια προβλήματα (π.χ. XPath και XQuery injection) στις οποίες αυτές οι τεχνικές μπορούν επίσης να χρησιμοποιηθούν για την προστασία τους.

5.2.1 Κυριότερες Λύσεις

- Άμυνα 1: Χρήση έτοιμων παραμετροποιημένων ερωτημάτων (parametrized queries)
- Άμυνα 2: Χρήση αποθηκευμένων διαδικασιών
- Άμυνα 3: Επικύρωση εισόδου με χρήση λίστας επιτρεπόμενων στοιχείων (whitelist)
- Άμυνα 4: Φιλτράρισμα δεδομένων και διαφυγή ειδικών χαρακτήρων (escape) όλων των παρεχόμενων από τον χρήστη εισόδων

Το ακόλουθο παράδειγμα (γραμμένο σε Java), δεν είναι ασφαλές και θα επέτρεπε σε έναν κακόβουλο χρήστη να εισάγει κώδικα στο ερώτημα που θα εκτελεστεί από τη βάση δεδομένων. Η μη επικυρωμένη παράμετρος "customerName" που ενσωματώνεται απλοϊκά στο query επιτρέπει σε έναν κακόβουλο χρήστη να εισάγει οποιονδήποτε κώδικα SQL επιθυμεί. Δυστυχώς, αυτή η μέθοδος πρόσβασης σε βάσεις δεδομένων είναι πολύ κοινή και σύντομα θα καταστεί σαφές γιατί πρέπει να αποφεύγεται.

```
String query = "SELECT account_balance FROM user_data WHERE
user_name = '" + 'request.getParameter("customerName") + "'";
try {
      Statement statement = connection.createStatement( ... );
      ResultSet results = statement.executeQuery(query);
}
catch (SQLException se) {
      // … logging and error handling
}
```

Αν ο κακόβουλος χρήστης διαμορφώσει την παράμετρο `customerName` να έχει την τιμή `tom' or 1=1 --` τότε η ερώτηση SQL θα διαμορφωθεί τελικά ως

```
SELECT account_balance FROM user_data WHERE user_name = 'tom' or
1=1 - '
```

με αποτέλεσμα να επιστρέφει όλα τα δεδομένα του πίνακα `account_balance`. Ο κακόβουλος χρήστης μπορεί επίσης να διαμορφώσει την παράμετρο `customerName` να έχει την τιμή `tom'; drop table account_balance --` με αποτέλεσμα η ερώτηση SQL να διαμορφωθεί τελικά ως

```
SELECT account_balance FROM user_data WHERE user_name = 'tom';
drop table account_balance; -- '
```

και τελική κατάληξη την καταστροφή του πίνακα `account_balance`.

5.2.1.1 *Επιλογή άμυνας 1*: Προπαραμετροποιημένα ερωτήματα (parametrized queries)

Η χρήση των προπαρασκευασμένων, παραμετροποιημένων ερωτημάτων (parametrized queries) είναι ο ενδεδειγμένος τρόπος τον οποίο όλοι οι προγραμματιστές θα έπρεπε να χρησιμοποιούν όταν κάνουν ερωτήματα σε μία βάση δεδομένων. Ένα παραμετροποιημένο query είναι απλό στη σύνταξη και ίσως πιο κατανοητό από ένα δυναμικό ερώτημα. Τα παραμετροποιημένα ερωτήματα αναγκάζουν τον προγραμματιστή να ορίσει πρώτα όλο τον κώδικα SQL ορίζοντας παραμέτρους για τις σταθερές που θα παρασχεθούν δυναμικά κατά τον χρόνο εκτέλεσης και στη συνέχεια να περάσει τιμές σε κάθε παράμετρο του ερωτήματος. Έτσι τα δεδομένα/παράμετροι διαχωρίζονται σαφώς από το κείμενο της ερώτησης (query). Αυτό το στυλ προγραμματισμού επιτρέπει συνεπώς στη βάση δεδομένων να κάνει διάκριση μεταξύ κώδικα και δεδομένων, ανεξάρτητα από το τι παρέχεται στο (ενδεχομένως κακόβουλο) κείμενο που δίνεται ως είσοδος από τον χρήστη.

Τα parametrized queries διασφαλίζουν ότι ένας επιτιθέμενος δεν θα μπορέσει να αλλάξει τις παραμέτρους ενός ερωτήματος και να εισάγει τη δική του λογική, με όποιο τρόπο και αν προσπαθήσει να εισάγει εντολές SQL. Στο παρακάτω «ασφαλές» παράδειγμα, αν ένας επιτιθέμενος εισάγει στο πεδίο userID την τιμή tom' or ' 1 '=' 1, το παραμετροποιημένο ερώτημα δεν θα είναι ευάλωτο και θα αναζητούσε ένα όνομα χρήστη που ταιριάζει κυριολεκτικά με ολόκληρη τη συμβολοσειρά tom ' or ' 1 '=' 1 και δεν θα εκτελούσε την κακόβουλη λογική όπως σε ένα δυναμικό ερώτημα.

Αναφορικά με τη γλώσσα PHP, με την οποία ασχοληθήκαμε στο έργο μας, μία λύση χρησιμοποιώντας αυτή την τεχνική, είναι η χρήση PDO με parametrized queries (χρησιμοποιώντας την μέθοδο bindParam()), ή στην περίπτωση μας επειδή χρησιμοποιήσαμε βάση δεδομένων MySQL, η δημιουργία των prepared statements γίνεται με χρήση των mysqli_prepare() ή prepare(), ανάλογα με το αν έχουμε επιλέξει να γράψουμε κώδικα με αντικειμενοστρεφές τρόπο ή πιο κλασικό procedural ύφος. Βέβαια σε μικρά έργα στα οποία δεν χρειάζεται απόλυτη επεκτασιμότητα και για απλές συνδέσεις και queries στην βάση, αυτοί οι τρόποι δεν έχουν σημαντικές διαφορές μεταξύ τους, αλλά φυσικά όσο η πολυπλοκότητα αυξάνεται αυτό αλλάζει. Αντίστοιχα χρησιμοποιούμε την μέθοδο bind_param() ή την συνάρτηση mysqli_stmt_bind_param() για να αναθέσουμε τις δικές μας μεταβλητές στο query όπως φαίνεται στα παρακάτω παραδείγματα:

1ος τρόπος: Αντικειμενοστρεφές (Object-Oriented) στυλ προγραμματισμού

Ακολουθώντας το αντικειμενοστρεφές στυλ προγραμματισμού χρησιμοποιούμε τις μεθόδους mysqli::prepare για τη δημιουργία του prepared statement και με χρήση της μεθόδου mysqli_stmt::bind_param παρέχουμε τις τιμές των μεταβλητών με το query.

```
$stmt = $mysqli -> prepare ("INSERT INTO CountryLanguage VALUES (?, ?, ?, ?)");
$stmt -> bind_param ('assd', $code, $language, $official, $percent);
```

Κατόπιν, με χρήση της μεθόδου mysqli_stmt::execute εκτελούμε το προετοιμασμένο query

```
/* execute prepared statement */
$stmt->execute();
```

2ος τρόπος: Διαδικαστικό (Procedural) στυλ προγραμματισμού

Ακολουθώντας το διαδικαστικό στυλ προγραμματισμού χρησιμοποιούμε τις συναρτήσεις mysqli_prepare για το prepared statement μας και με χρήση της συνάρτησης mysqli_stmt_bind_param παρέχουμε τις τιμές των μεταβλητών με το query.

```
$stmt = mysqli_prepare($link, "INSERT INTO CountryLanguage VALUES (?, ?, ?, ?)");
mysqli_stmt_bind_param($stmt, 'assd', $code, $language, $official, $percent);
```

Με χρήση της συνάρτησης **mysqli_stmt_execute()** εκτελούμε το προετοιμασμένο query.

```
/* execute prepared statement */
mysqli_stmt_execute($stmt);
```

Ενώ και για τα δύο παραδείγματα οι τιμές των μεταβλητών θα μπορούσαν να είναι οι παρακάτω:

```
$code = "GR";
$language = "Greek";
$official = "GR";
$percent = 89.5;
```

Σε σπάνιες περιπτώσεις, τα parametrized queries μπορούν να βλάψουν την απόδοση της εφαρμογής, ιδίως αν η ερώτηση δεν έχει διατυπωθεί ώστε να δίνονται σωστά οι πληροφορίες για τον τύπο των παραμέτρων και απαιτούνται μετατροπές τύπων. Όταν αντιμετωπίζουμε αυτού του είδους τα προβλήματα, είναι καλύτερο να εξετάσουμε εκ νέου τη διατύπωση της ερώτησης ώστε να μην πραγματοποιούνται αυτόματες μετατροπές τύπων. Αν αυτό δεν δώσει αποτέλεσμα, τότε θα πρέπει να α) επικυρώσουμε(validate) πολλαπλές φορές όλα τα δεδομένα ή β) να χρησιμοποιούμε τεχνικές διαφυγής (escaping) για όλες τις παρεχόμενες από τον χρήστη εισόδους χρησιμοποιώντας μια ρουτίνα διαφυγής συγκεκριμένη για τον προμηθευτή της βάσης δεδομένων μας. (33)

5.2.1.2 *Επιλογή άμυνας 2: Χρήση αποθηκευμένων διαδικασιών*

Οι αποθηκευμένες διαδικασίες (stored procedures) δεν είναι πάντα ασφαλείς από εγχύσεις SQL. Ωστόσο, ορισμένες τυποποιημένες δομές προγραμματισμού αποθηκευμένης διαδικασίας έχουν το ίδιο αποτέλεσμα με τη χρήση παραμετροποιημένων ερωτημάτων όταν εφαρμόζονται με ασφάλεια, που είναι ο κανόνας για τις περισσότερες αποθηκευμένες γλώσσες διαδικασίας. Απαιτούν από τον προγραμματιστή να δημιουργήσει απλώς δηλώσεις SQL με παραμέτρους που παραμετροποιούνται αυτόματα εκτός αν ο προγραμματιστής δοκιμάσει κάτι που ξεφεύγει σε μεγάλο βαθμό από το φυσιολογικό. Η διαφορά μεταξύ των parametrized queries και των αποθηκευμένων διαδικασιών είναι ότι ο κώδικας SQL για μια αποθηκευμένη διαδικασία ορίζεται και αποθηκεύεται στην ίδια τη βάση δεδομένων και στη συνέχεια καλείται από την εφαρμογή. Και οι δύο αυτές τεχνικές έχουν την ίδια αποτελεσματικότητα στην αποτροπή της έγχυσης SQL, οπότε ένας οργανισμός ή ένας προγραμματιστής επιλέγει ποια προσέγγιση τον συμφέρει καλύτερα να ακολουθήσει.

Σημείωση: «*Εφαρμόζεται με ασφάλεια*» σημαίνει ότι η αποθηκευμένη διαδικασία δεν πραγματοποιεί δυναμική παραγωγή SQL. Οι προγραμματιστές συνήθως δεν δημιουργούν δυναμικό SQL κώδικα μέσα σε αποθηκευμένες διαδικασίες. Ωστόσο, αν και η δυναμική πραγωγή SQL κώδικα πρέπει να αποφεύγεται, μπορεί να γίνει. Αν ένα τέτοιο σενάριο δεν μπορεί, να αποφευχθεί, η αποθηκευμένη διαδικασία πρέπει να χρησιμοποιεί επικύρωση εισόδου ή κατάλληλη διαφυγή, όπως περιγράφεται σε άλλα σημεία της διπλωματικής.

Υπάρχουν επίσης πολλές περιπτώσεις όπου οι αποθηκευμένες διαδικασίες μπορούν να αυξήσουν τον κίνδυνο. Για παράδειγμα, στον διακομιστή MS SQL, υπάρχουν 3 κύριοι προεπιλεγμένους ρόλοι: db_datareader, db_datawriter και db_owner. Προτού τεθούν σε χρήση οι αποθηκευμένες διαδικασίες, τα DBA δίνουν στο db_datareader ή στο db_datawriter δικαιώματα στο χρήστη της διαδικτυακής υπηρεσίας, ανάλογα με τις απαιτήσεις. Ωστόσο, οι αποθηκευμένες διαδικασίες απαιτούν δικαιώματα εκτέλεσης, έναν ρόλο που δεν είναι διαθέσιμος από default ως προεπιλογή. Ορισμένες ρυθμίσεις όπου η διαχείριση χρηστών έχει συγκεντρωθεί, αλλά περιορίζεται σε αυτούς τους 3 ρόλους, προκαλούν την εκτέλεση όλων των εφαρμογών ιστού με δικαιώματα db_owner, ώστε οι αποθηκευμένες διαδικασίες να μπορούν να λειτουργήσουν. Φυσικά, αυτό σημαίνει ότι εάν

παραβιαστεί ένας διακομιστής, ο εισβολέας έχει πλήρη δικαιώματα στη βάση δεδομένων, όπου στο παρελθόν μπορεί να είχε μόνο πρόσβαση ανάγνωσης.

5.2.1.3 *Επιλογή άμυνας 3: Επικύρωση εισαγωγής στη λίστα επιτρεπόμενων (whitelist)*

Διάφορα τμήματα των ερωτημάτων SQL δεν είναι παραδεκτές τοποθεσίες για τη χρήση μεταβλητών με οποιαδήποτε τιμή: τέτοια τμήματα είναι τα ονόματα πινάκων ή στηλών και η ένδειξη ταξινόμησης ταξινόμησης (ASC ή DESC), όπου μπορούν κανονικά να εμφανιστούν μόνο προσδιοριστές (identifiers) και όχι αυθαίρετες συμβολοσειρές. Σε τέτοιες περιπτώσεις, η επικύρωση εισαγωγής ή ο επανασχεδιασμός του ερωτήματος είναι η καταλληλότερη άμυνα. Για τα ονόματα πινάκων ή στηλών, ιδανικά αυτές οι τιμές προέρχονται από τον κώδικα και όχι από τις παραμέτρους του χρήστη.

Αλλά εάν οι τιμές παραμέτρων χρήστη χρησιμοποιούνται για τη στόχευση διαφορετικών ονομάτων πινάκων και ονομάτων στηλών, τότε οι τιμές παραμέτρων θα πρέπει να αντιστοιχιστούν στα παραδεκτά / αναμενόμενα ονόματα πινάκων ή στηλών για να διασφαλιστεί ότι η μη είσοδος χρήστη δεν καταλήγει στο ερώτημα χωρίς να έχει επικυρωθεί και ελεγχθεί. Ας ληφθεί υπ' όψιν ότι αυτό είναι ένα αποτέλεσμα κακής σχεδίασης.

5.2.1.4 *Επιλογή άμυνας 4: Φιλτράρισμα δεδομένων και διαφυγή ειδικών χαρακτήρων*

Η άλλη λοιπόν βασική λύση σε αυτή την επίθεση είναι όλα τα πεδία εισαγωγής (όπως πεδία κειμένου, πλαίσια σχολίων κ.λπ.) μιας εφαρμογής να ελεγχθούν πολλαπλές φορές. Θα πρέπει οπωσδήποτε να φιλτράρουμε τις εισόδους που μας έρχονται από τον χρήστη και να αφαιρούμε ειδικά σύμβολα ('=','΄μονά και διπλά εισαγωγικά κ.λπ.) και γενικότερα να χρησιμοποιούμε input sanitization, το οποίο περιγράψαμε παραπάνω. Για να φιλτράρονται οι μη επικυρωμένες εντολές SQL, μπορούμε να ενσωματώσουμε ένα τείχος προστασίας (firewall) με βαθιά επισκόπηση πακέτων (deep packet inspection) στο σύστημα ασφαλείας μας.

Η τελευταία τεχνική, η οποία πρέπει να χρησιμοποιείται, όταν κανένα από τα παραπάνω δεν είναι εφικτό ή σε συνδυασμό με άλλες τεχνικές είναι το escape της εισόδου του χρήστη. Η μεθοδολογία αυτή της διαφυγής εισόδων, είναι μία ακόμα επιλογή αν και θα πρέπει ο προγραμματιστής να είναι ιδιαίτερα προσεκτικός ώστε να μην παραλείπει ποτέ να εισάγει τις κατάλληλες εντολές για αντικατάσταση των «επικίνδυνων» χαρακτήρων με κατάλληλους χαρακτήρες διαφυγής.

Αυτή η τεχνική είναι να χρησιμοποιήσουμε διαφυγή στην είσοδο του χρήστη πριν αυτή τοποθετηθεί σε ένα ερώτημα. Αυτός είναι ένας απλός τρόπος που φέρει μία αποτελεσματικότητα ειδικά σε εφαρμογές που έχουν φτιαχτεί παλαιότερα χρησιμοποιώντας πρότερες εκδόσεις της php. Το εγχειρίδιο για ένα SQL DBMS εξηγεί ποιοι χαρακτήρες έχουν ειδική σημασία, γεγονός το οποίο μας επιτρέπει τη δημιουργία μιας ολοκληρωμένης μαύρης λίστας χαρακτήρων που χρειάζονται αντικατάσταση - μετάφραση. Για παράδειγμα, κάθε εμφάνιση ενός μονού εισαγωγικού (') σε μια παράμετρο πρέπει να αντικατασταθεί από διπλά εισαγωγικά (") για να σχηματιστεί μια έγκυρη κυριολεκτική συμβολοσειρά (string literal) SQL. Για παράδειγμα, στην PHP είναι συνηθισμένο να χρησιμοποιείται διαφυγή χαρακτήρων για τις παραμέτρους χρησιμοποιώντας τη συνάρτηση mysqli_real_escape_string (); πριν από την αποστολή του ερωτήματος SQL. Αυτή η συνάρτηση εισάγει την ανάστροφη κάθετο (backslash) πριν από ακόλουθους χαρακτήρες: \x00, \n, \r, \, ' (μονό εισαγωγικό), " (διπλό εισαγωγικό) και \

x1a (CTRL-Z, νοείται ως end-of-file). Αυτή η λειτουργία χρησιμοποιείται συνήθως για να μετατρέψει τα δεδομένα σε ασφαλή μορφή πριν από την αποστολή ενός ερωτήματος στη MySQL.

Η PHP έχει παρόμοιες λειτουργίες για άλλα συστήματα βάσεων δεδομένων, όπως pg_escape_string () για την PostgreSQL. Η συνάρτηση `addslashes` λειτουργεί για διαφυγή χαρακτήρων και χρησιμοποιείται ειδικά για αναζήτηση σε βάσεις δεδομένων που δεν έχουν λειτουργίες διαφυγής στην PHP. Επιστρέφει μια συμβολοσειρά με ανάστροφη κάθετο πριν από χαρακτήρες για τους οποίους πρέπει να χρησιμοποιηθεί διαφυγή σε ερωτήματα βάσης δεδομένων, κ.λπ. Αυτοί οι χαρακτήρες περιλαμβάνουν τα μονά εισαγωγικά ('), τα διπλά εισαγωγικά ("), την ανάστροφη κάθετο (\) και το NUL (το NULL byte, 0x00).

Κάθε DBMS υποστηρίζει ένα ή περισσότερα σχήματα διαφυγής χαρακτήρων προσαρμοσμένα σε συγκεκριμένα είδη ερωτημάτων. Εάν επιτύχουμε να αποφύγουμε όλες τις παρεχόμενες από τον χρήστη εισόδους χρησιμοποιώντας το σωστό σχήμα διαφυγής για τη βάση δεδομένων που χρησιμοποιούμε, το DBMS δεν θα συγχέει αυτήν την είσοδο με τον κώδικα SQL που έχει γράψει ο προγραμματιστής, αποφεύγοντας έτσι τυχόν ευπάθειες έγχυσης SQL.

Η τακτική μετάδοση συμβολοσειρών διαφυγής στη SQL είναι επιρρεπής σε σφάλματα επειδή είναι εύκολο να ξεχάσουμε να διαφύγουμε από μια δεδομένη συμβολοσειρά. Επίσης ακόμα και να τα κάνουμε όλα σωστά, δεν είμαστε πάντα ασφαλείς ότι με την αποφυγή κάποιων ειδικών χαρακτήρων θα είμαστε ασφαλείς σε όλες τις περιπτώσεις επιθέσεων SQLi. Η δημιουργία ενός διαφανούς επιπέδου για τη διασφάλιση της εισόδου μπορεί να μειώσει (ή και να εξαλείψει) αυτήν την εμφάνιση λάθους.

5.3 Απειλές από αυτοματοποιημένα bots

Οι απειλές από αυτοματοποιημένα bots προέρχονται κυρίως από ρομπότ που έχουν σχεδιαστεί με τέτοιο τρόπο ώστε να εκτελούν ένα μεγάλο αριθμό επαναλαμβανόμενων εργασιών, χωρίς την ανθρώπινη παρέμβαση. Στη σημερινή εποχή, υπάρχει μεγάλη πιθανότητα κάποιες αιτήσεις να προέρχονται από μηχανές και όχι από ανθρώπους. Αυτές οι μηχανές - “κακόβουλα” διαδικτυακά bots χρησιμοποιούνται εδώ και πολλά χρόνια στο διαδίκτυο. Μπορούν να χρησιμοποιηθούν με διάφορους στόχους, από τη δημιουργία ψεύτικων λογαριασμών, έως την ταχεία κράτηση όλων των εισιτηρίων για μια δημοφιλή συναυλία και την ενορχήστρωση μιας μεγάλης κλίμακας επίθεσης άρνησης παροχής υπηρεσίας (DoS). Χρειαζόμαστε έναν αξιόπιστο τρόπο για να ξεχωρίσουμε ένα κακόβουλο bot από έναν νομότυπο χρήστη. Ένας τρόπος για να ελέγξουμε αν ο χρήστης είναι μηχανή ή άνθρωπος, μπορούμε να χρησιμοποιήσουμε την τεχνολογία CAPTCHA. Ακριβώς όπως τα ίδια τα bots στο Διαδίκτυο, και όπως και πολλές από τις καινοτομίες στο Διαδίκτυο, τα CAPTCHA βρίσκουν την καταγωγή τους στην κοινότητα των χάκερ. Πίσω στη δεκαετία του 1980, οι χάκερ εφηύραν το leetspeek για να παρακάμψουν το φιλτράρισμα ασφαλείας σε φόρουμ συνομιλίας μέσω Διαδικτύου. Το Leet είναι μια μέθοδος μετατροπής λέξεων σε όμοιους χαρακτήρες ή συντομογραφίες που δεν μπορούν εύκολα να ερμηνευτούν από έναν υπολογιστή. Βέβαια από τότε έχει γίνει μεγάλη εξέλιξη αυτού του τύπου εργαλείων.

Είναι εύκολο να γίνει διάκριση μεταξύ δεδομένων που έχουν εισαχθεί από τον χρήστη και αυτοματοποιημένων δεδομένων. Η τεχνολογία ανίχνευσης bot σε πραγματικό χρόνο μπορεί να μας βοηθήσει να εξαλείψουμε σε μεγάλο βαθμό τις αυτοματοποιημένες απειλές. Η τεχνολογία που χρησιμοποιήσαμε εμείς είναι το σύστημα ReCaptcha της

Google. Η Google, κατά καιρούς έχει εφαρμόσει πολλές μεθόδους για να διακρίνονται τα bots από τους ανθρώπους. Η τρέχουσα τεχνολογία (reCAPTCHAv3) τυπικά εμφανίζει τη φράση «δεν είμαι ρομπότ», και θα πρέπει ο άνθρωπος-χρήστης να περάσει επιτυχώς κάποιον έλεγχο για να αποδείξει στο σύστημα την ανθρώπινη φύση του και να του επιτραπεί η πρόσβαση. Πλέον, με το reCAPTCHA δε χρειάζεται να υποβάλλεται ο χρήστης σε κοπιαστικές δοκιμασίες όπως π.χ. να διακρίνει θολές ταμπέλες και περίεργες εικόνες ή να επιλέγει τις σωστές απαντήσεις σε ερωτήσεις, καθώς η τρέχουσα τεχνολογία αυτοματοποιεί σε μεγάλο βαθμό τον έλεγχο. Χρησιμοποιώντας έναν συνδυασμό μηχανικής εκμάθησης (machine learning) και προηγμένη ανάλυση επικινδυνότητας, η Google ενημέρωσε το σύστημα να εντοπίζει τις συνήθειες των χρηστών, χωρίς κάποιο πρόσθετο εργαλείο. Όταν εισερχόμαστε σε κάποια σελίδα, εξαφανίζονται τα ελεγκτικά εργαλεία και φαίνεται μόνο το σχετικό περιεχόμενο. Παρόλα αυτά, σε κάποιες περιπτώσεις, ίσως χρειαστεί να λύσουμε κάποιο απλό παζλ. Ενώ το νέο σύστημα είναι διαφανές, λαμβάνει και πάλι υπόψη του μεταβλητές όπως την διεύθυνση IP και τις κινήσεις του ποντικιού μας. Η Google αναφέρει ότι η τεχνολογία ελέγχει και τη γενικότερη συμπεριφορά του χρήστη για να εξακριβώσει ότι είναι άνθρωπος. Χάρη σε αυτό το σύστημα της Google, η πρόσβασή πλέον είναι πολύ πιο εύκολη και γρήγορη, ενώ ταυτόχρονα είμαστε σε μεγάλο βαθμό ασφαλείς από αυτού του είδους τις επιθέσεις.

5.4 File Path Traversal

Το file path traversal είναι επίσης γνωστό ως διάσχιση καταλόγου ή backtracking. Ο πρωταρχικός στόχος αυτής της επίθεσης εφαρμογών ιστού είναι η πρόσβαση σε αρχεία και καταλόγους που δεν τοποθετούνται κάτω από το root directory του ιστοχώρου, αλλά βρίσκονται σε αυθαίρετα σημεία στο σύστημα αρχείων. Οι hackers έχουν πρόσβαση σε αυθαίρετα αρχεία και καταλόγους μεταβάλλοντας τις μεταβλητές αρχείων (όπως χρησιμοποιώντας dot-dot-slash, ../).

Αυτή η επίθεση στις web εφαρμογές, μπορεί να αποφευχθεί με input validation. Η εφαρμογή των απαιτούμενων φίλτρων στην εφαρμογή μας μπορεί να εξαλείψει τις πιθανότητες οι hackers να πάρουν στην κατοχή τους αυθαίρετα αρχεία και φακέλους. Οι πιο σημαίνοντες μηχανισμοί που μπορούν να εφαρμοστούν με σχετική ευκολία είναι η χρήση της συνάρτησης realpath και την ρύθμιση open_basedir. Επιπλέον, η αναβάθμιση του λογισμικού του web server ή οποιοδήποτε λογισμικό επιδιόρθωσης, μπορεί να προστατεύσει την εφαρμογή μας από το file path traversal.

Στη συνέχεια παρουσιάζονται κάποιοι από τους πιο αποτελεσματικούς μηχανισμούς που μπορούν να χρησιμοποιηθούν για την αποφυγή τέτοιων επιθέσεων.

- *Realpath*: Επιστρέφει το κανονικοποιημένο, απόλυτο (absolute) pathname (34)
  `realpath ( string $path ) : string`

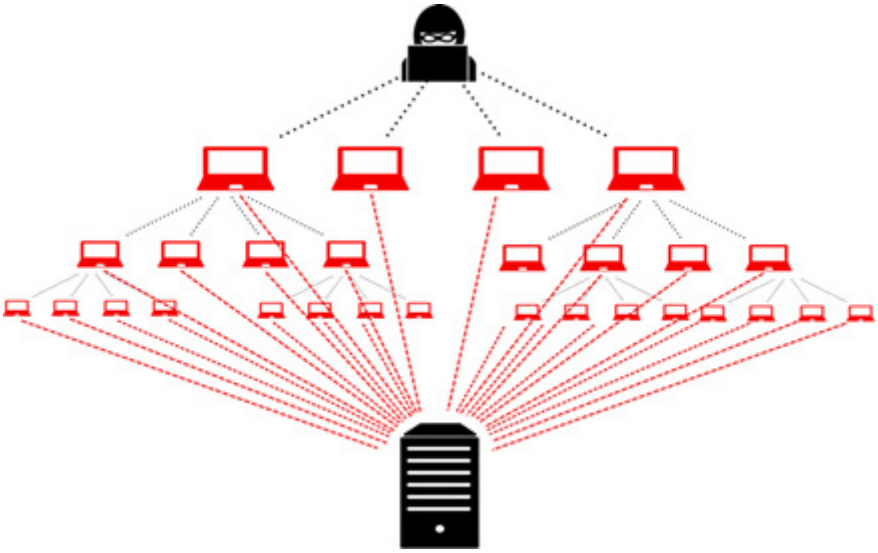
  Η συνάρτηση realpath () επεκτείνει όλους τους συμβολικούς συνδέσμους (links) και επιλύει αναφορές σε /./, /../ και επιπλέον / χαρακτήρες στη διαδρομή εισόδου, επιστρέφοντας το pathname σε κανονικοποιημένη μορφή, χωρίς να περιέχονται σε αυτό χαρακτήρες οι οποίοι μπορούν να θέσουν σε κίνδυνο την εφαρμογή. (34)

- *Open_basedir*: (35) Είναι ένας περιορισμός, ο οποίος χρησιμοποιείται για να αποτρέψει σε ένα κακόβουλο χρήστη να αποκτήσει πρόσβαση σε μη εξουσιοδοτημένα paths στο server όπως σε άλλα ονόματα domain. Αν ένας server φιλοξενεί πολλά sites και ένα από αυτά δεν έχουν επαρκή ασφάλεια και δεν χρησιμοποιούν το open_basedir, μπορεί ένας κακόβουλος χρήστης μέσω αυτού

του site να αποκτήσει πρόσβαση σε κάποιο άλλο ή σε αυθαίρετα αρχεία στον υπολογιστή. Η δήλωση/ρύθμιση *open_basedir* καθορίζει τις τοποθεσίες ή τις διαδρομές από τις οποίες επιτρέπεται η πρόσβαση της PHP σε αρχεία χρησιμοποιώντας λειτουργίες όπως *fopen ()* και *gzopen ()* ή εν γένει οποιαδήποτε πρόσβαση. Εάν ένα αρχείο βρίσκεται εκτός των διαδρομών που ορίζονται από το open_basedir, η PHP θα αρνηθεί να το ανοίξει. (36)

5.5 Distributed Denial of Service (DDoS)

Αυτού του είδους οι επιθέσεις περιλαμβάνουν τον έλεγχο πολυάριθμων υπολογιστών από πλευράς του επιτιθέμενου. Συνήθως είναι υπολογιστές με χαμηλές δυνατότητες, οι οποίοι οργανώνονται σε κάποιο botnet. Για παράδειγμα, μπορεί να είναι πολλές συσκευές IoT . Αυτοί οι υπολογιστές ύστερα «βομβαρδίζουν» με αιτήσεις ένα server στον οποίο έχουν στοχεύσει, έτσι ώστε να τον κάνουν να μην μπορεί να ανταποκριθεί στα αιτήματα των νόμιμων επισκεπτών του (επίθεση στη διαθεσιμότητα των υπηρεσιών). Παρόλο που το DDoS δεν παρέχει από μόνο του πρόσβαση στα δεδομένα στον επιτιθέμενο, τα προηγούμενα χρόνια παρατηρήσαμε μια τάση των επιτιθέμενων να χρησιμοποιούν όλο και περισσότερο το DDoS παράλληλα με άλλες επιθέσεις, για να κρατήσουν απασχολημένους τους αυτοματοποιημένους αμυντικούς μηχανισμούς.



Για μεγάλη μερίδα των σημερινών δικτύων υπολογιστών υπάρχει η δυνατότητα να αντιμετωπισθούν με επιτυχία απλές επιθέσεις DDoS. Τα βασικά μέτρα αντιμετώπισης, είναι είτε να έχουμε προληπτικούς μηχανισμούς αποτροπής των επιθέσεων, είτε να κάνουμε φιλτράρισμα των πακέτων που έχουν ύποπτη συμπεριφορά κατά την επίθεση, είτε να προσδιορίσουμε την πηγή - ταυτότητα της επίθεσης και να φιλτράρουμε τις αιτήσεις από αυτόν τον επιτιθέμενο.

Για παράδειγμα, για να αμυνθούμε έναντι επιθέσεων DDoS, μπορούμε να καταγράφουμε τον αριθμό των συνδέσεων που έχει ανοίξει κάθε client και να απαγορευτεί η δημιουργία νέας σύνδεσης, όταν αυτός ξεπεράσει το όριο που έχει τεθεί ή όταν ο ρυθμός δημιουργίας συνδέσεων υπερβαίνει κάποιο όριο. Μπορούμε ακόμη να απαγορεύσουμε μέσω του firewall την πρόσβαση σε συγκεκριμένες διευθύνσεις IP οι οποίες δρουν ύποπτα. Βέβαια σε επιθέσεις DDoS, όπου έχουμε επιθέσεις από εξαιρετικά μεγάλο αριθμό συσκευών αυτή

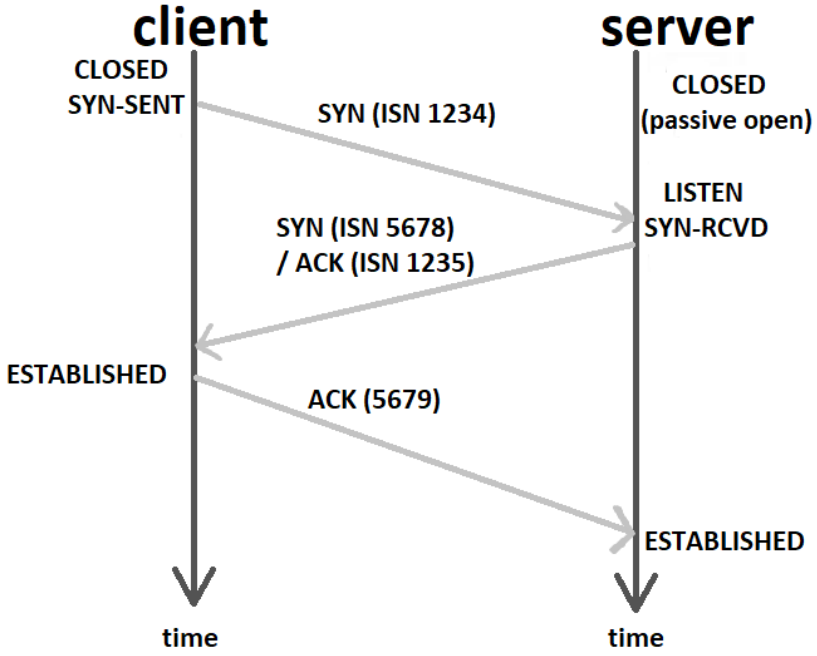
η τακτική μπορεί να μην είναι τόσο αποτελεσματική. Γενικότερα οι επιθέσεις DDoS γίνονται ολοένα και πιο σύνθετες οπότε χρειαζόμαστε πάντα νέα μέτρα ασφαλείας. (37)

Οι συχνές δυσλειτουργίες ενός ιστοχώρου/υπολογιστικού συστήματος, τα απρόσμενα υψηλά φορτία καθώς και τα ασυνήθιστα μηνύματα σφάλματος και οι εξάντληση πόρων, είναι σοβαρές ενδείξεις για την ύπαρξη μιας κακόβουλης πράξης DDoS.

Στη συνέχεια παρουσιάζονται κάποια μέτρα πρόληψης που μπορεί να ακολουθηθούν για να αποτρέψουμε τέτοιου είδους επιθέσεις

- Πρέπει να γίνεται συνέχεια *έλεγχος της κίνησης των δεδομένων* με χρήση διάφορων στοιχείων υλικού και λογισμικού ώστε αν ξαφνικά εντοπιστεί υπερβολικά υψηλός από αιτήματα αυτά να φιλτράρονται και να υπάρχει επαρκής χρόνος ώστε να αποφευχθεί η κατάρρευση του διακομιστή. Αυτό θα γίνει αν μπουν όσο το δυνατόν πιο σύντομα περιορισμοί στη κίνηση. Είναι σημαντικό λοιπόν να γνωρίζουμε τι είναι κανονική, χαμηλή και υψηλή κυκλοφορία και ποιες είναι οι ώρες αιχμής που έχουμε περισσότερη κίνηση και πότε έχουμε χαμηλή. Με αυτή τη γνώση είναι δυνατόν να τίθενται πιο ρεαλιστικοί και αποτελεσματικοί περιορισμοί.
- *Προμήθεια μεγαλύτερου εύρους ζώνης*. Η απόκτηση μεγαλύτερου εύρους ζώνης διακομιστή από ό,τι πραγματικά χρειαζόμαστε, μπορεί να κοστίζει, ωστόσο μας εξασφαλίζει καλύτερη αντίδραση σε περίπτωση επίθεσης DDoS πριν η ιστοσελίδα, ο διακομιστής ή η εφαρμογή υπερφορτωθεί εντελώς.
- *Χρήση Content Distribution Network (CDN)*. Ο στόχος ενός DDoS είναι η υπερφόρτωση του διακομιστή που φιλοξενεί την εφαρμογή. Μια λύση λοιπόν είναι να αποθηκευθούν τα δεδομένα σε πολλούς διακομιστές σε όλο τον κόσμο. Αυτό ακριβώς κάνει το δίκτυο διανομής περιεχομένου (Content Distribution Network). Τα CDN εξυπηρετούν τον ιστότοπο ή τα δεδομένα σε χρήστες από ένα διακομιστή που βρίσκεται κοντά σε κάθε χρήστη για ταχύτερη απόδοση. Χρησιμοποιώντας ένα τέτοιο δίκτυο γινόμαστε λιγότερο ευάλωτοι σε επιθέσεις, επειδή εάν ένας διακομιστής υπερφορτωθεί, υπάρχουν πολλοί ακόμη διαθέσιμοι σε λειτουργία.
- Οι διακομιστές θα πρέπει να προστατεύονται από τείχη προστασίας δικτύου και πιο εξειδικευμένα τείχη προστασίας εφαρμογών ιστού, καθώς και θα πρέπει να χρησιμοποιούνται εξισορροπιστές φορτίου. Πολλοί προμηθευτές υλικού περιλαμβάνουν τώρα προστασία λογισμικού από επιθέσεις πρωτοκόλλου DDoS, όπως π.χ. επιθέσεις τύπου SYN flood. Για να εξηγήσουμε τις επιθέσεις SYN flood θα ξεκινήσουμε από το επίπεδο των δικτύων και συγκεκριμένα πρέπει να γίνει κατανοητή η διαδικασία σύνδεσης δύο υπολογιστών με το πρωτόκολλο TCP. Στην επικοινωνία με βάση των πρωτόκολλο TCP, η τριμερής χειραψία (3-way handshake) σηματοδοτεί την αρχή της σύνδεσης μεταξύ ενός πελάτη (client) και ενός εξυπηρετητή (server) στο πρωτόκολλο μεταφοράς TCP, το οποίο βρίσκεται πάνω από το πρωτόκολλο IP. Ο server πρώτα αφού έχει δημιουργήσει μία δίοδο (socket) και έχει πραγματοποιήσει τις κατάλληλες ενέργειες για να αντιστοιχήσει τη δίοδο σε έναν αριθμό θύρας του TCP/IP, στη συνέχεια το ανοίγει έτσι ώστε να αρχίσει να δέχεται τις συνδέσεις που περιμένουν στην ουρά (passive open). Όταν γίνει αυτό, ο client με τη σειρά του μπορεί να αρχίσει τη σύνδεση (active open) με το 3-way handshake. Η διαδικασία έχει ως ακολούθως:
    - Αρχικά αποστέλλεται από τον client ένα πακέτο SYN (synchronize). Ο client θέτει το πεδίο αριθμού ακολουθίας στην επικεφαλίδα (TCP header) στον αρχικό αριθμό ακολουθίας του (ISN - initial sequence number).

- Ο διακομιστής (server) από την άλλη απαντάει:
  - είτε με SYN-ACK (acknowledge) που σημαίνει ότι αποδέχεται τη σύνδεση (στέλνει το δικό του ISN που έχει το ISN+1 του πρώτου πακέτου του client),
  - είτε με SYN/RST (reset) για να αρνηθεί για κάποιο λόγο τη σύνδεση στον client και η διαδικασία σταματά.
- Τέλος, όταν ο client πάρει το πακέτο SYN/ACK απαντάει και αυτός τελευταίος στέλνοντας ένα ACK (acknowledgment), επιβεβαιώνοντας έτσι τη σύνδεση και δείχνοντας ότι από εδώ και στο εξής μπορεί να ξεκινήσει η αποστολή των πακέτων με τα δεδομένα. (38)



Εικόνα 29. Χειραψία εγκαθίδρυσης σύνδεσης στο TCP/IP

Η διαδικασία της τριμερούς χειραψίας όμως μπορεί να χρησιμοποιηθεί και από κάποιον κακόβουλο. Η πιο γνωστή επίθεση είναι η επίθεση SYN flooding (39), η οποία είναι επίθεση άρνησης παροχής υπηρεσίας (DoS), όπου ο επιτιθέμενος επιτίθεται είτε από ένα μόνο μηχάνημα είτε από πλήθος μηχανημάτων, στέλνοντας σειρά αιτήσεων με SYN πακέτα σε κάθε port του server, αλλά δεν απαντάει πάλι με ACK σε αυτόν ούτως ώστε να ολοκληρωθεί φυσιολογικά η τριμερής χειραψία του TCP. Ο διακομιστής θεωρεί ότι τα πακέτα αυτά προέρχονται από έναν κανονικό νόμιμο χρήστη, οπότε απαντά με πακέτα SYN-ACK σύμφωνα με τη διαδικασία χειραψίας του πρωτοκόλλου TCP. Ο επιτιθέμενος όμως όπως είπαμε δεν απαντά και λίγο πριν έρθει το timeout στέλνει και άλλο SYN κρατώντας τον server μερικά, ή ολοκληρωτικά απασχολημένο, εξαντλώντας το πλήθος των διαθέσιμων συνδέσεων TCP/IP. Για κάθε σύνδεση που δεν ολοκληρώνεται, ο server δαπανά υπολογιστικούς πόρους. Αυτό έχει ως αποτέλεσμα μετά από κάποιο συγκεκριμένο αριθμό τέτοιων συνδέσεων ο διακομιστής να φτάσει στα όριά του και κάποιος άλλος νόμιμος χρήστης να μην

μπορεί να συνδεθεί με τον διακομιστή ή ακόμα και να μην μπορεί να εξυπηρετηθεί κανένας χρήστης, διότι ο διακομιστής ήταν απασχολημένος με τις συνδέσεις που είχε ανοίξει ο επιτιθέμενος. Η επίθεση SYN flood είναι αρκετά συνηθισμένη και η πλειοψηφία των σημερινών δικτύων υπολογιστών είναι σε θέση να την αντιμετωπίσει με επιτυχία π.χ. καταγράφοντας τον αριθμό των συνδέσεων που έχει ανοίξει κάθε client και η απαγόρευση δημιουργίας νέου όταν αυτός ξεπεράσει αυτό το όριο (39) παρακολουθώντας πόσες συνδέσεις βρίσκονται στο στάδιο της εκκρεμούς εγκαθίδρυσής τους, και διαγράφοντας κάποιες από αυτές όταν ο αριθμός φτάσει σε μια τιμή που έχει οριστεί. Συγκεκριμένες ενότητες λογισμικού μπορούν επίσης να προστεθούν σε κάποιο λογισμικό διακομιστή ιστού για να παρέχουν κάποια λειτουργικότητα πρόληψης DDoS. Για παράδειγμα, το λογισμικό Apache 2.2.15 περιλαμβάνει με μια μονάδα που ονομάζεται mod_reqtimeout για να προστατευτεί από επιθέσεις επιπέδου εφαρμογής, όπως η επίθεση Slowloris, η οποία ανοίγει συνδέσεις με έναν διακομιστή ιστού και στη συνέχεια τις κρατά ανοιχτές για όσο το δυνατόν περισσότερο στέλνοντας μερικές αιτήσεις μέχρι ο διακομιστής να μην μπορεί να δεχτεί πλέον νέες συνδέσεις.

- Πολλοί προμηθευτές ασφαλείας, συμπεριλαμβανομένων των NetScout Arbor, Fortinet, Check Point, Cisco και Radware, προσφέρουν συσκευές που τοποθετούνται μπροστά από τα τείχη προστασίας του δικτύου και έχουν σχεδιαστεί για να εμποδίζουν τις επιθέσεις DDoS πριν μπορέσουν να εκτυλιχθούν πλήρως. Το κάνουν αυτό χρησιμοποιώντας μια σειρά τεχνικών, όπως το να ελέγχουν τη συμπεριφορά της κυκλοφορίας της γραμμής και στη συνέχεια του αποκλεισμού της μη φυσιολογικής κυκλοφορίας και του αποκλεισμού της κυκλοφορίας βάσει γνωστών υπογραφών επίθεσης. Η κύρια αδυναμία αυτού του τύπου προσέγγισης για την πρόληψη επιθέσεων DDoS είναι ότι οι ίδιες οι συσκευές είναι περιορισμένες ως προς την ποσότητα της κυκλοφορίας που μπορούν να χειριστούν. Ενώ οι συσκευές προηγμένης τεχνολογίας ενδέχεται να είναι σε θέση να ελέγχουν την κυκλοφορία που φτάνει με ρυθμό έως και 80 Gbps, οι σημερινές επιθέσεις DDoS μπορούν εύκολα να είναι τάξεις μεγέθους μεγαλύτερες από αυτήν. (40)

5.6 Ευπάθειες σε IoT συσκευές (κυρίως μέσω κάμερας)

Τέλος μία πολύ σημαντική απειλή για την ιδιωτικότητα των χρηστών προέρχεται πλέον από τις IoT συσκευές. Η απειλή για την ιδιωτικότητα προέρχεται αφ' ενός από τη δυνατότητα μη εξουσιοδοτημένης πρόσβασης στις συσκευές και τη συλλογή σχετικών στοιχείων, όπως π.χ. τη λήψη στοιχείων για την κατανάλωση ρεύματος ή τη λειτουργία του φωτισμού και της θέρμανσης. Ειδικότερα για τις κάμερες και τις συσκευές καταγραφής ήχου, πέραν της ανωτέρω απειλών υπάρχει και το πρόβλημα της αυτόματης δημοσιοποίησης των ήχων ή των φωτογραφιών/video χωρίς προγενέστερο έλεγχο του τι υλικό περιλαμβάνει ο ήχος ή τι απεικονίζουν οι φωτογραφίες ή τα βίντεο.

Αναφορικά με το θέμα των ευπαθειών, αν ένας κακόβουλος χρήστης, αποκτήσει πρόσβαση σε μία κάμερα π.χ. μιας κινητής συσκευής, τότε οι επιπτώσεις για την ασφάλεια του χρήστη είναι πολυάριθμες. Ο επιτιθέμενος μπορεί να συγκεντρώσει δεδομένα του χρήστη χωρίς την άδειά του ακόμη και όταν ο χρήστης αποσυνδεθεί από το κοινωνικό δίκτυο, εφ΄όσον ο επιτιθέμενος αυτός έχει ακόμη την πρόσβαση, ή ακόμη χειρότερα μπορεί να δημοσιεύσει αυτό το περιεχόμενο του χρήστη στο κοινωνικό δίκτυο ή ευρύτερα στο διαδίκτυο χωρίς τη θέληση του χρήστη. Γενικότερα, εκτός από τη χρήση

τέτοιων συσκευών για επιθέσεις τύπου DDoS που αναφέραμε παραπάνω, αναλύουμε εκτενέστερα τη χρήση κάμερας σε συσκευές IoT, καθώς αυτό είναι ένα από τα σημαντικότερα κομμάτια του κοινωνικού μας δικτύου και θα πρέπει να δομηθεί η κατάλληλη ασφάλεια γύρω από αυτές.

Κάποιες ευπάθειες σε συσκευές IoT που έχουν αναλυθεί από άλλες έρευνες είναι οι παρακάτω:

1. Ελλιπείς έλεγχοι από πλευράς διακομιστών (servers) που είναι ενσωματωμένοι στις συσκευές, π.χ. web servers για διαχείριση.
2. Χρήση μη κρυπτογραφημένης επικοινωνίας ή προβληματικών μηχανισμών κρυπτογράφησης.
3. Έλλειψη δυαδικής προστασίας (χωρίς την οποία μπορεί να γίνει ευκολότερο reverse engineering).
4. Κακή εφαρμογή ελέγχων πιστοποίησης και αυθεντικοποίησης.
5. Μη κατάλληλη χρήση υπηρεσιών δικτύου.

Το κακόβουλο λογισμικό σε συσκευές IoT έχει σχεδιαστεί με σκοπό να επιτύχει μια ποικιλία επιθέσεων που κυμαίνονται από την κλοπή ευαίσθητων δεδομένων έως τη χρήση της συσκευής ως ορμητηρίου για πραγματοποίηση επιθέσεων σε άλλες συσκευές, π.χ. την κατασκευή ενός botnet. Το Mirai (41) (42) είναι ένα παράδειγμα σχηματισμού ενός μεγάλου κακόβουλου δικτύου botnet στο δίκτυο IoT, που προκάλεσε σημαντική ζημιά. Το συγκεκριμένο κακόβουλο λογισμικό εντοπίστηκε για πρώτη φορά κατά τη διάρκεια μιας επίθεσης DDoS στον ιστότοπο του δημοσιογράφου Brian Krebs (43). Το Mirai εκμεταλλεύεται βασικά ελαττώματα ασφάλειας σε συσκευές IoT, όπως προκαθορισμένα (hard-coded) ονόματα χρήστη και κωδικούς πρόσβασης για σύνδεση με το πρωτόκολλο telnet. Το λογισμικό έχει ένα προεγκατεστημένο σύνολο συνδυασμών ονόματος χρήστη και κωδικού πρόσβασης που χρησιμοποιεί για να κάνει επιθέσεις εξαντλητικής αναζήτησης (brute force) στη συσκευή. Το Mirai στοχεύει κυρίως τις κάμερες, καθώς έχουν υψηλή υπολογιστική ισχύ σε σύγκριση με άλλες συσκευές IoT. Μόλις το Mirai εκμεταλλευτεί επιτυχώς μια ευπάθεια στη συσκευή και αποκτήσει τον έλεγχό της, μετατρέπει τη συσκευή σε ένα bot που ελέγχεται από το διακομιστή των επιτιθέμενων. Το Mirai έχει την ικανότητα να εκτελεί διάφορους τύπους επιθέσεων DDoS όπως DNS, UDP, SYN και ACK flooding. (44) Υπάρχουν πολλοί άλλοι τύποι κακόβουλου λογισμικού προσανατολισμένου στο IoT που έχουν προκαλέσει σημαντική ζημιά. Ένα θέμα ευπάθειας σε μία ή περισσότερες κάμερες, θα μπορούσε να επιτρέψει στους επιτιθέμενους να αποκτήσουν απομακρυσμένο έλεγχο συσκευών για χρήση ως εργαλείο παρακολούθησης, με τη δυνατότητα να παρακολουθούν (snoop) οποιοδήποτε ήχο ή βίντεο που έχει εγγραφεί. Ένα παράδειγμα είναι η ευπάθεια που βρέθηκε σε σχεδόν 400 μοντέλα βιντεοκαμερών με σύνδεση στο Διαδίκτυο γνωστού κατασκευαστή. Αξιοποιώντας τις ευπάθειες στις κάμερες που συνδέονται στο Διαδίκτυο από την εταιρεία Axis Communication, οι ερευνητές της εταιρείας ασφαλείας VDOO διαπίστωσαν ότι οι απομακρυσμένοι εισβολείς μπορούσαν να καταλάβουν συσκευές χρησιμοποιώντας μόνο τη διεύθυνση IP και χωρίς προηγούμενη πρόσβαση στην κάμερα ή γνώση των διαπιστευτηρίων σύνδεσης (45). Οι ευπάθειες έχουν αποκαλυφθεί στον οργανισμό Axis, ο οποίος έχει ενημερώσει το υλικολογισμικό όλων των επηρεαζόμενων προϊόντων, προκειμένου να προστατεύσει τους χρήστες από το να πέσουν θύματα επιθέσεων. Συνολικά ανακαλύφθηκαν επτά ευπάθειες στις κάμερες και οι ερευνητές έχουν αναλύσει πως τρεις από αυτές θα μπορούσαν να συνδέονται για να παρέχουν απομακρυσμένη πρόσβαση στις κάμερες και να εκτελούν απομακρυσμένα εντολές κελύφους (shell commands) με δικαιώματα πρόσβασης διαχειριστή (root). Σε αυτά περιλαμβάνεται η

παροχή πρόσβασης στη ροή βίντεο της κάμερας, η δυνατότητα ελέγχου της θέσης της κάμερας και ο έλεγχος της ανίχνευσης κίνησης και η δυνατότητα ακρόασης ήχου. Υπάρχει επίσης η δυνατότητα εκμετάλλευσης των καμερών με τέτοιο τρόπο ώστε να μπορούν να χρησιμοποιηθούν ως σημείο εισόδου στο δίκτυο για μια ευρύτερη επίθεση, καθώς και η πιθανότητα η κάμερα να γίνει μέρος ενός κακόβουλου δικτύου botnet. Ο λόγος που οι ευπάθειες οι οποίες επιτρέπουν την πρόσβαση διαχειριστή είναι τόσο υψηλού κινδύνου, είναι ότι ο εισβολέας μπορεί πρακτικά να χρησιμοποιήσει οποιοδήποτε χαρακτηριστικό της κάμερας. Αν κάποιος εισβολέας διαθέτει τους κατάλληλους πόρους, και γνωρίζει τέτοια τρωτά σημεία, θα μπορούσε σίγουρα να παραβιάσει την ιδιωτική ζωή και την ασφάλεια του οργανισμού σε σοβαρό βαθμό, ιδιαίτερα λαμβάνοντας υπ' όψιν ότι από το χρονικό σημείο που οι επιτιθέμενοι ανακαλύπτουν μία ευπάθεια μέχρι τη δημιουργία της κατάλληλης επιδιόρθωσης και της εγκατάστασής της, μεσολαβεί συνήθως σημαντικό χρονικό διάστημα. Επίσης, θα μπορούσε να επιτεθεί σε άλλους στόχους χρησιμοποιώντας πολλές από τις επηρεαζόμενες κάμερες. (45)

Όσον αφορά το 2ο ζήτημα ως προς την αντιμετώπιση θεμάτων με την ανωνυμοποίηση / αφαίρεση προσωπικών δεδομένων από τις φωτογραφίες, η σκέψη είναι να υλοποιηθεί λειτουργία ή οποία θα σχετίζεται άμεσα με την ιδιωτικότητα και θα αποτελέσει ενδεχομένως το υπόβαθρο για τη μελλοντική λειτουργία αυτόματου ανεβάσματος από IP κάμερα. Ένα σενάριο είναι το ακόλουθο: Όταν ανεβάζουμε μία φωτογραφία, μπορεί να υπάρχουν μέσα πρόσωπα που δεν θέλουμε ή δεν θα πρέπει να εμφανίζονται εκεί. Σκοπός είναι να χρησιμοποιηθεί κάποια βιβλιοθήκη η οποία θα επιτρέπει την αναγνώριση περιοχών της φωτογραφίας που είναι πρόσωπα (ενδεχομένως και πινακίδες αυτοκινήτων και άλλα ευαίσθητα δεδομένα στο μέλλον). Για παράδειγμα θα μπορούσε το σύστημα να εντοπίζει ποιες περιοχές είναι πρόσωπα και να προτρέπει τον χρήστη να επιλέξει ποιες θα διατηρηθούν και ποιες θα "θολωθούν" (με κάποιο μωσαϊκό ή blurring ή ακόμη και cropping, εάν είναι στα όρια της φωτογραφίας). Θα μπορούσε γενικότερα να γίνεται αυτόματη συσκότιση προσώπων, πινακίδων αυτοκινήτων ή οποιουδήποτε άλλου στοιχείου θεωρείται ιδιωτική πληροφορία. Στην υλοποίηση μας, την οποία θα αναφέρουμε αναλυτικότερα στην ενότητα 6, χρησιμοποιούμε μία βιβλιοθήκη για να εξάγουμε συγκεκριμένα δεδομένα από μία φωτογραφία μέσω της βιβλιοθήκης face-api.js η οποία είναι κατά βάση ένα περιτύλιγμα (wrapper) της δημοφιλούς βιβλιοθήκης *tensorflow* (46). Μέσα από αυτό το API παρέχεται η δυνατότητα να υλοποιηθεί σε Javascript το κατάλληλο classification και να εντοπίζονται δεδομένα όπως πρόσωπα και πινακίδες αυτοκινήτων (για την ώρα υπάρχει υλοποίηση μόνο για την αναγνώριση προσώπων, αλλά με αντίστοιχο τρόπο μπορεί να υλοποιηθεί μία κατάλληλη επέκταση). Από τη στιγμή που οι κατάλληλες περιοχές έχουν εντοπισθεί, είναι εύκολο να σκιασθούν ή να καλυφθούν με μωσαϊκό. Άλλα αντικείμενα των οποίων η προβολή ενέχει κινδύνους ιδιωτικότητας, εκτός από ανθρώπινα πρόσωπα και πινακίδες αυτοκινήτων, θεωρούμε ότι είναι τμήματα ανθρώπινων σωμάτων, είναι στοιχεία πιστωτικών καρτών και γενικότερα ευαίσθητοι αριθμοί, όπως ΑΦΜ, ΑΜΚΑ, αριθμοί σπιτιών κ.λπ. Γενικότερα οποιαδήποτε κείμενα με γράμματα και αριθμούς θα μπορούσαν να αναγνωρίζονται και να σκιάζονται. Ενδέχεται επίσης να γίνουν δημοσιεύσεις στο κοινωνικό δίκτυο από φορείς και οργανισμούς, όπως εταιρείες, στρατιωτικά τμήματα, κυβερνητικούς οργανισμούς και χρηματοπιστωτικά ιδρύματα, και σε όλους αυτούς τους οργανισμούς παράγεται σε καθημερινή βάση, ένας τεράστιος όγκος πληροφοριών. Πολλές από τις πληροφορίες αυτές είναι απόρρητες και δεν θα πρέπει να γίνονται προσβάσιμες σε εξωτερικούς (κακόβουλους ή μη) χρήστες. Άρα λοιπόν, δημιουργείται η ανάγκη για την προστασία των απόρρητων πληροφοριών αυτών, κάτι που αντιμετωπίζεται, έως ένα βαθμό, από την εφαρμογή τεχνικών εντοπισμού και σκίασης/συσκότισης. Σε πρώτο στάδιο

αναγνωρίζονται μόνο τα ανθρώπινα πρόσωπα και σκιάζονται, με εφαρμογή ενός συνδυασμού μάσκας και ενός ποσοστού θολώματος – blurring. Στο μέλλον, θα ήταν επιθυμητό να έχει και ο χρήστης έλεγχο στη διαδικασία και να θολώνει / ξεθολώνει τα σημεία που αυτός επιθυμεί ακόμη και αν αυτό ενέχει κινδύνους για την ιδιωτικότητα, καθώς μπορεί να μη γνωρίζει ένα στοιχείο το οποίο είναι απόρρητο ή απλά να θέλει να συμπεριλάβει το δικό του πρόσωπο ή άτομα τα οποία έχουν δώσει την συγκατάθεσή τους σε αυτόν, κάτι που δεν είναι γνωστό στο σύστημα. Πάντως είναι πιο ασφαλές μία εικόνα τελικά να αξιολογείται από τον χρήστη, πριν τη δημοσίευσή της.

Αναφορικά με άλλα μέτρα προστασίας, όταν αλλάξει η IP μιας συσκευής ενημερώνεται ο χρήστη και κατόπιν της έγκρισής του διακόπτουμε την πρόσβαση από την ενδεχομένως κακόβουλη συσκευή. Βέβαια η IP μπορεί να αλλάξει με πολυάριθμους τρόπους π.χ. αν συνδεόμαστε αρχικά με δεδομένα (data) και ενεργοποιήσουμε τη σύνδεση με το wifi ή αν κλείσουμε το wifi και γυρίσουμε σε δεδομένα και γενικότερα δεν είναι στατική για τους περισσότερους χρήστες. Παρ'όλα αυτά αν αλλάξει ένας συνδυασμός πραγμάτων μαζί με την διεύθυνση IP του, τον ειδοποιούμε στέλνοντας του μήνυμα email για να ερευνήσει αν υπάρχει κάποιο πρόβλημα. Ο μηχανισμός δεν είναι τέλειος και μπορεί να εμφανίσει false positives ενδεχομένως, αλλά είναι ένα επιπλέον βήμα ασφαλείας που μαζί με τους υπόλοιπους μηχανισμούς δρα θετικά ως προς την ασφάλεια. Γενικότερα καταγράφουμε με μία πιθανότητα λάθους, αν ο χρήστης έχει συνδεθεί από συσκευή IoT, την IP του και το λειτουργικό σύστημα που χρησιμοποίησε, προσπαθώντας να ειδοποιούμε τον χρήστη κάθε φορά που ο υπολογιστής ή το δίκτυο του αλλάξουν. Όπως είπαμε δεν μπορούμε να εμπιστευθούμε απόλυτα τον μηχανισμό, ειδικά για την καταγραφή του περιηγητή και του λειτουργικού συστήματος καθώς ο χρήστης μπορεί να αλλοιώσει αυτά τα δεδομένα και στο επίπεδο της PHP δεν μπορούμε να κάνουμε ελέγχους σε χαμηλότερο επίπεδο, αλλά είναι θεωρούμε και ένα καλό βήμα για μελλοντικές επεκτάσεις και η επέκταση του μηχανισμού λειτουργεί όπως είδαμε σε άλλες μεγαλύτερες εφαρμογές. Στην υπάρχουσα εφαρμογή εν τέλει συνήθως ειδοποιούμε τον χρήστη αν κάποιο από αυτά τα δεδομένα που αναφέρθηκαν παραπάνω αλλάξει, ή απλά μπλοκάρουμε κάποια πρόσβαση. Στο μέλλον αυτά τα στοιχεία μπορούν να χρησιμοποιηθούν με ακόμη πιο έξυπνο τρόπο. Επιπροσθέτως, αν αλλάξει η IP, θα είναι πολύ δύσκολο να χρησιμοποιηθεί και η λειτουργία αυτόματου ανεβάσματος από IP κάμερα (η οποία λειτουργία θα μπορούσε να αποτελέσει μελλοντική επέκταση στην εφαρμογής μας). Ο λόγος είναι ότι δεν θα είναι δυνατή η άντληση δεδομένων από τη συγκεκριμένη κάμερα, γιατί πλέον περιμένουμε διαφορετική διεύθυνση IP από αυτή που η συσκευή έχει: η κάμερα θα μπορούσε να λειτουργεί σε περίπτωση αλλαγής διεύθυνσης μόνο εάν η επικοινωνία βασίζεται σε κάποιο δυναμικό πρωτόκολλο ανακάλυψης συσκευής ή κάποιο δυναμικό πρωτόκολλο αντιστοίχισης διεύθυνσης όπως π.χ. το NetBios (47) ή το DDNS (48). Σε αυτή την περίπτωση, το μόνο που μπορούμε να κάνουμε είναι να ειδοποιήσουμε τον χρήστη ότι η διεύθυνση της κάμερας έχει αλλάξει και δεν μπορεί πλέον να εντοπιστεί από την εφαρμογή.

6 Ενισχύοντας την ασφάλεια του κοινωνικού δικτύου

Σε αυτό το κεφάλαιο παρουσιάζουμε πιο αναλυτικά μεθόδους που χρησιμοποιήσαμε για να ενισχύσουμε την ασφάλεια του κοινωνικού δικτύου. Χρησιμοποιήσαμε τόσο κάποιες κλασικές μεθόδους, όπως ελέγχους στα πεδία εισόδου και captcha, αλλά και μεθόδους που λαμβάνουν υπ' όψιν τα συγκεκριμένα χαρακτηριστικά της εφαρμογής, όπως ασφαλέστερο σχεδιασμό των σχέσεων των χρηστών μας, κρυπτογράφηση δεδομένων και κατάλληλη διαχείριση συσκευών IoT και ευπαθειών σε αυτές.

6.1 Μέθοδοι που χρησιμοποιήσαμε

6.1.1 Έλεγχοι στα πεδία εισόδου

Όπως αναφέρθηκε και ανωτέρω, για να αποφύγουμε επιθέσεις όπως SQL injection και cross-site scripting (XSS) χρειαζόμαστε έναν συνδυασμό μεθόδων για το φιλτράρισμα των δεδομένων του χρήστη και την αποβολή μη αποδεκτών χαρακτήρων Για την καλύτερη πρόληψη επιθέσεων τύπου SQL injection χρησιμοποιείται και η τεχνική των parametrized queries, όπου τα δεδομένα / παράμετροι ξεχωρίζονται από το query, με αποτέλεσμα να μην είναι δυνατόν να ερμηνευτούν τα δεδομένα εισόδου ως άλλα δομικά στοιχεία του query (π.χ. ονόματα πινάκων ή τμήματα της συνθήκης). Με άλλα λόγια, οι δύο πιο συνηθισμένες επιθέσεις στον ιστότοπο - SQL injection και cross-site scripting (XSS), μοιράζονται μια ριζική κοινή αιτία, την έλλειψη κατάλληλου φιλτραρίσματος στα πεδία εισόδου.

Όταν ο χρήστης δεν είναι εξουσιοδοτημένος, τότε μπορεί να εκμεταλλευτεί τον μηχανισμό και να εισάγει κακόβουλα δεδομένα στα πεδία μας. Με αυτόν τον τρόπο μπορεί π.χ. να κάνει μία επίθεση τύπου SQL injection και να προβάλλει δεδομένα από τη βάση ή ακόμη και να διαγράψει ολόκληρους πίνακες στην βάση μας ή να υποκλέψει δεδομένα πιστωτικών καρτών των χρηστών μας. Στην εφαρμογή μας συνεπώς κάναμε κατάλληλους ελέγχους του κειμένου που εισάγεται στα πεδία (sanitization κ.λπ.) ώστε να αποτραπεί η επιτυχία επιθέσεων τύπου SQL injection και άλλες κακόβουλες ενέργειες.

6.1.2 Έλεγχος για πρόσβαση από bots (Μηχανισμοί «I'm not a robot»)

Τα bots, όπως αναφέραμε, είναι μία πολύ συχνή απειλή στον χώρο των διαδικτυακών εφαρμογών. Η τεχνολογία που χρησιμοποιήσαμε για να αποφύγουμε τέτοια προβλήματα ασφαλείας ήταν ο μηχανισμός Captcha reCaptcha της Google. Η τεχνολογία αυτή έχει σκοπό την ανίχνευση ρομπότ σε πραγματικό χρόνο και μπορεί να μας βοηθήσει να εξαλείψουμε σε μεγάλο βαθμό τις αυτοματοποιημένες απειλές. Σε αυτή τη κατηγορία υπάρχουν πλέον αρκετές μέθοδοι και λύσεις. Μία από αυτές ήταν το reCAPTCHA της εταιρίας Google. Με το reCAPTCHA ο χρήστης τικάρει ένα checkbox. Ύστερα ανάλογα με τον τρόπο που ο χρήστης πάτησε το κουμπί χρησιμοποιώντας έναν συνδυασμό μηχανικής μάθησης (machine learning) και ανάλυσης υψηλών ρίσκων, το σύστημα αποφασίζει αν πρόκειται για πραγματικό χρήστη. Εμείς χρειάζεται να ελέγξουμε αν ο χρήστης πάτησε πάνω σε αυτό το κουμπί και αν πέρασε με επιτυχία τη διαδικασία ή όχι. Από την οπτική γωνία του developer, η διαδικασία ενσωμάτωσης της λειτουργικότητας

είναι αρκετά απλή. Το μόνο που χρειάζεται είναι να γίνει λήψη ενός κωδικού (API key) από τον αντίστοιχο ιστότοπο και μετά η ενσωμάτωση στην σελίδα μας γίνεται μία αρκετά εύκολη διαδικασία.

6.1.3 Έλεγχος σύνδεσης από διαφορετικές συσκευές και τοποθεσίες και κατάλληλες ενημερώσεις των χρηστών.

Σε αυτό το στάδιο καταγράφουμε στη βάση δεδομένων το λειτουργικό σύστημα το οποίο χρησιμοποιεί ο χρήστης όταν συνδέεται στην εφαρμογή, καθώς και τη διεύθυνση IP της σύνδεσης. Αρχικά υπήρχε η σκέψη να καταγράφεται μόνο η διεύθυνση IP αλλά επειδή δεν είναι πάντα στατική και μπορεί να αλλάζει, διερευνήθηκε η δυνατότητα να φυλάσσονται και άλλα στοιχεία για την συσκευή. Μία ιδέα ήταν να κρατείται η διεύθυνση MAC, αλλά αυτή η ιδέα είχε τελικά αρκετά προβλήματα: κατ'αρχάς η διεύθυνση MAC δεν είναι απαραίτητα μοναδική αλλά επίσης δεν είναι και σταθερή, καθώς σε σύγχρονα συστήματα κινητών ή άλλες νεότερες συσκευές αλλάζει, με σκοπό την προστασία της ιδιωτικότητας. Έτσι υπήρξε η σκέψη να φυλάσσεται με κάποιο τρόπο η MAC του router, η οποία αλλάζει πολύ δυσκολότερα, οπότε για τους χρήστες που συνδέονται συνήθως από το ίδιο μέρος (π.χ. την ίδια οικία) μπορούμε να διατηρήσουμε αυτή την πληροφορία, όμως τελικά δεν ήταν δυνατό να αποκτηθεί αυτόματα η συγκεκριμένη διεύθυνση και θα χρειαζόταν κάποια χειροκίνητη ρύθμιση, η οποία ωστόσο δεν είναι ασφαλέστερη από τη χρήση ενός διαμοιραζόμενου μυστικού. Έτσι, εγκαταλείφθηκε αυτή η ιδέα. Φυσικά από το επίπεδο που βρίσκεται η εφαρμογή μας τέτοιοι έλεγχοι γενικότερα δεν είναι απόλυτα έμπιστοι, αλλά αν συνδυαστούν μπορούν να μας προσδώσουν ένα επίπεδο ασφαλείας. Οι έλεγχοι κυρίως για την διεύθυνση IP, γίνονται ώστε σε περίπτωση που ο ίδιος λογαριασμός συνδεθεί από άλλο δίκτυο ή και άλλη συσκευή να ενημερώνεται ο χρήστης με ένα email και να προλαβαίνει τον κίνδυνο. Επίσης μέσω της καταγραφής του λειτουργικού συστήματος συμπεραίνουμε με κάποια πιθανότητα λάθους αν η συσκευή του χρήστη είναι συσκευή IoT ή αν είναι άλλο περιβάλλον. Μία άλλη σκέψη ήταν να καταγράψουμε και την τοποθεσία του χρήστη, ωστόσο είναι ασαφές αν αυτό μπορεί να συνιστά παραβίαση των προσωπικών δεδομένων του χρήστη, και έτσι αποφασίστηκε στην παρούσα φάση να μην καταγράφεται η πληροφορία αυτή.

6.1.4 Έλεγχος δικαιωμάτων πρόσβασης στην κάμερα ή σε άλλες IoT συσκευές καθώς και ενημέρωση των χρηστών σε περίπτωση μεταβολής των συσκευών.

Ένας μηχανισμός που προς το παρόν βρίσκεται ακόμη στο στάδιο της διερεύνησης είναι να ελέγχονται οι συσκευές που χρησιμοποιεί ο χρήστης για να πραγματοποιούνται λειτουργίες του συστήματος, π.χ. για να ανεβάζει φωτογραφίες, ώστε αν αλλάξει η διεύθυνση IP να ενημερώνεται ο χρήστης και να προβαίνει στις ενέργειες που κρίνει σκόπιμες, λ.χ. να διακόπτεται η πρόσβαση από την ενδεχομένως κακόβουλη συσκευή. Στην υπάρχουσα εφαρμογή η επιλογή που έχει γίνει είναι να ειδοποιούμε τον χρήστη αν εντοπίσουμε κάποια ενδεχόμενη απειλή. Στο μέλλον θα μπορούσε να εξετασθεί το μπλοκάρισμα της διεύθυνσης IP της κακόβουλης συσκευής ή η απενεργοποίηση κάποιας αυτόματης λειτουργίας (π.χ. ανεβάσματος φωτογραφιών).

Μία ακόμη λειτουργία που κρίθηκε απαραίτητη έτσι ώστε να γίνει πρόληψη θεμάτων ιδιωτικότητας ήταν ο εντοπισμός και η αυτόματη συσκότιση περιοχών που αντιστοιχούν σε ευαίσθητα δεδομένα τα οποία θα μπορούσαν να προκαλέσουν προβλήματα αν δημοσιευθούν αφιλτράριστα (όπως π.χ. δημοσίευση προσώπων ατόμων χωρίς τη

συγκατάθεση αυτών). Πιο συγκεκριμένα, όταν ανεβάζουμε μία φωτογραφία, μπορεί να υπάρχουν μέσα σε αυτή πρόσωπα που δεν είναι επιθυμητό ή δεν θα πρέπει να εμφανίζονται εκεί. Σκοπός ήταν να χρησιμοποιηθεί κάποια βιβλιοθήκη η οποία θα επιτρέπει την αναγνώριση περιοχών της φωτογραφίας που είναι πρόσωπα (ενδεχομένως και πινακίδες αυτοκινήτων και άλλα ευαίσθητα δεδομένα στο μέλλον) και ύστερα αυτά να θολώνονται.

Κατόπιν έρευνας στις διαθέσιμες βιβλιοθήκες ανοικτού λογισμικού, εντοπίσθηκαν βιβλιοθήκες αναγνώρισης περιοχών που αποτελούν πρόσωπα και λοιπά αντικείμενα σε γλώσσες PHP και Javascript. Κάποιες από αυτές, πέραν από αυτές που τελικά χρησιμοποιήθηκαν είναι οι παρακάτω:

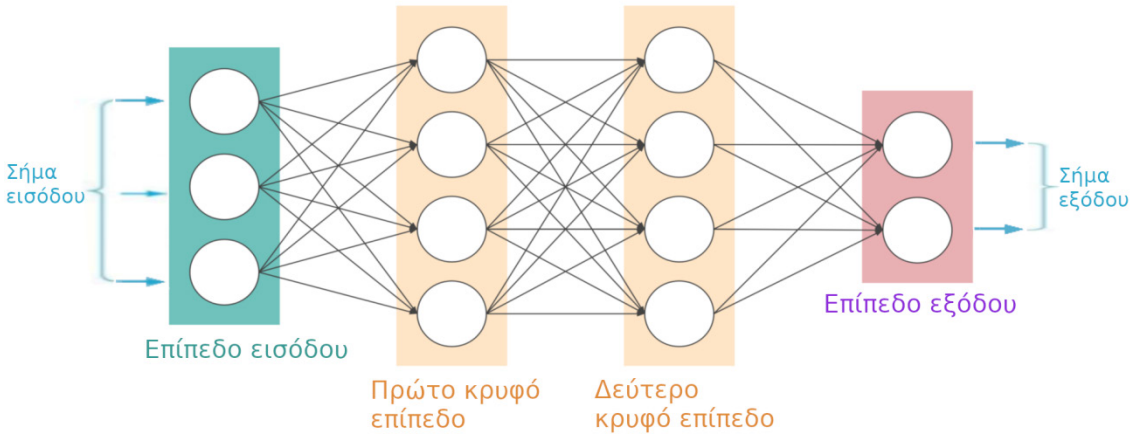
- **PHP Face Detection Library**: Η συγκεκριμένη βιβλιοθήκη είναι υλοποιημένη σε γλώσσα PHP (Έκδοση PHP5). Μέσω αυτής δίνεται η δυνατότητα να σαρωθούν φωτογραφίες και να εντοπισθούν περιοχές σε αυτές που αντιστοιχούν σε πρόσωπα. Η βιβλιοθήκη μας δίνει ακόμη την δυνατότητα να αποκόψουμε τα πρόσωπα αυτά, ακόμη και να τα αποθηκεύσουμε σε άλλο αρχείο εικόνας μετά την αποκοπή. Γενικότερα, η βιβλιοθήκη μπορεί να φανεί πολύ χρήσιμη για την ανίχνευση προσώπων και για μέσω webcam, όπως στη δική μας περίπτωση. Επίσης η διαδικασία περικοπής είναι αρκετά ακριβής και αφαιρείται μόνο το μέρος της εικόνας που χρειάζεται αφού η βιβλιοθήκη έχει ανιχνεύσει με ακρίβεια τα κύρια σημεία και γραμμές ενός προσώπου. Η βιβλιοθήκη είναι διαθέσιμη στον σύνδεσμο https://www.phpclasses.org/package/11178-PHP-Scan-recognize-and-crop-faces-in-images.html
- **Cloud Vision API**: Αυτή η βιβλιοθήκη - διεπαφή προγραμματισμού εφαρμογών (API) παρέχεται από τη Google. Παρέχει, μεταξύ άλλων, τη λειτουργία ανίχνευσης προσώπων. Με αυτή τη λειτουργία μας παρέχεται η δυνατότητα ανίχνευσης πολλαπλών προσώπων μέσα σε μια εικόνα. Μπορεί να γίνει ακριβής ανίχνευση ακόμα και αν τα πρόσωπα βρίσκονται σε διάφορες συναισθηματικές καταστάσεις ή φορούν αξεσουάρ όπως γυαλιά και καπέλα, σύμφωνα πάντα με την κατασκευάστρια εταιρία. Δεν υποστηρίζεται μεμονωμένη αναγνώριση - ταυτοποίηση για ένα συγκεκριμένο πρόσωπο. Το API είναι διαθέσιμο στον σύνδεσμο https://cloud.google.com/vision/docs/detecting-faces.

Εικόνα 30.Ανίχνευση Προσώπου με χρήση του Cloud Vision API

Στην υλοποίηση του Pik-Pok χρησιμοποιήθηκε το λογισμικό face-api.js, το οποίο είναι ένα API υλοποιημένο σε JavaScript που παρέχει τη δυνατότητα να πραγματοποιηθεί αναγνώριση προσώπων. Το API είναι περιτύλιγμα (wrapper) της γνωστής βιβλιοθήκης

tensorflow.js και συνεπώς χρησιμοποιεί το *tensorflow* ως πυρήνα του. Μέσω του API καθίστανται διαθέσιμα αρκετά χαρακτηριστικά του tensorflow, εκτός από την απλή αναγνώριση προσώπων, όπως την αναγνώριση ηλικιών, συναισθημάτων, φύλου (gender) κ.ά.

Το TensorFlow είναι μία βιβλιοθήκη για μηχανική μάθηση (machine learning), το οποίο μας δίνει δυνατότητες αναγνώρισης αντικειμένων χρησιμοποιώντας νευρωνικά δίκτυα. Ένα νευρωνικό δίκτυο, είναι ένας αλγόριθμος μάθησης, που εμπνέεται από τη δομή και τις λειτουργικές πτυχές των βιολογικών νευρωνικών δικτύων. Η δομή των υπολογισμών βασίζεται σε μια ομάδα εσωτερικά διασυνδεδεμένων τεχνητών νευρώνων, οι οποίοι επεξεργάζονται την πληροφορία και εκτελούν υπολογισμούς επικοινωνώντας μεταξύ τους. Τα τεχνητά νευρωνικά δίκτυα είναι απλοποιημένα μοντέλα του κεντρικού νευρικού συστήματος του ανθρώπου. Μιμούνται τη λειτουργία των βιολογικών νευρώνων του εγκεφάλου και τη δομή των βιολογικών νευρωνικών δικτύων. Αποτελούνται από διασυνδεδεμένα υπολογιστικά στοιχεία που έχουν την ικανότητα να ανταποκρίνονται σε ερεθίσματα που δέχονται στην είσοδό τους και να μαθαίνουν να προσαρμόζονται στο περιβάλλον τους. Γενικότερα προσομοιώνοντας τη λειτουργία ενός νευρωνικού δικτύου που βασίζεται στον ανθρώπινο εγκέφαλο, η ελπίδα είναι να αναπτυχθεί μία διαδικασία-ευφυής αλγόριθμος που θα είναι σε θέση να προσομοιώνει λειτουργίες που έχει ο άνθρωπος όπως η μάθηση, η μνήμη, η γενίκευση, η ομαδοποίηση προτύπων κ.λπ., χωρίς να χρειάζεται να γίνει ανάλυση όλων των πιθανών περιπτώσεων που μπορούν να προκύψουν και να συγγράφεται κώδικας για όλες αυτές. Η ιδέα είναι ότι η μηχανή εκπαιδεύεται με βάση τα πρότυπα που έχει και «μαθαίνει από τα λάθη της», όπως (συνήθως) και ένας άνθρωπος. Υπάρχουν διάφορα είδη νευρωνικών δικτύων. Το tensorflow χρησιμοποιεί πολυεπίπεδα νευρωνικά δίκτυα (MLP). Ένα τέτοιο πολυεπίπεδο δίκτυο αποτελείται από ένα σύνολο αισθητήρων (πηγαίοι κόμβοι) που αποτελούν το επίπεδο εισόδου, ένα ή περισσότερα κρυφά επίπεδα (hidden layers) υπολογιστικών κόμβων και ένα επίπεδο υπολογιστικών κόμβων εξόδου. Το βασικό χαρακτηριστικό του δικτύου αυτού είναι ότι οι νευρώνες του οποιουδήποτε στρώματος *l*, τροφοδοτούν αποκλειστικά τους νευρώνες του επόμενου στρώματος *l+1* και τροφοδοτούνται αποκλειστικά από τους νευρώνες του προηγούμενου στρώματος *l-1*. Η αρχιτεκτονική ενός δικτύου MLP τριών επιπέδων με δύο κρυφά επίπεδα απεικονίζεται στο παρακάτω σχήμα:



Εικόνα 31. Αρχιτεκτονική ενός δικτύου MLP τριών επιπέδων με δύο κρυφά επίπεδα

Το δίκτυο αυτό είναι πλήρως διασυνδεδεμένο, δηλαδή ένας νευρώνας σε κάθε επίπεδο είναι συνδεδεμένος με όλους τους νευρώνες του προηγούμενου επιπέδου. Επιπλέον το σήμα ρέει μέσα στο δίκτυο σε μία προς τα εμπρός κατεύθυνση (στην εικόνα, από τα αριστερά προς τα δεξιά) και από επίπεδο σε επίπεδο. Η εκπαίδευση ενός δικτύου πολλών επιπέδων είναι η διαδικασία ρύθμισης των βαρών των συνάψεών του, έτσι ώστε να ικανοποιείται κάποιο κριτήριο βελτιστοποίησης. Στόχος της εκπαίδευσης είναι το δίκτυο να μάθει οποιαδήποτε επιθυμητή συνάρτηση με οποιοδήποτε βαθμό προσέγγισης. Ο αλγόριθμος που χρησιμοποιείται κυρίως για την εκπαίδευση των MLP είναι ο αλγόριθμος της αντίστροφης διάδοσης σφάλματος (Error Back Propagation). Ο αλγόριθμος αυτός βασίζεται στον κανόνα μάθησης διόρθωσης σφαλμάτων (error correction learning rule). Βασικό στοιχείο της εκπαίδευσης αυτής είναι η ύπαρξη στόχων, δηλαδή το δίκτυο εκπαιδεύεται με επίβλεψη. Η διαδικασία εκπαίδευσης αποτελείται από δύο περάσματα διαμέσου των επιπέδων του δικτύου: ένα πέρασμα προς τα εμπρός και ένα πέρασμα προς τα πίσω. Στο πέρασμα προς τα εμπρός ένα διάνυσμα εισόδου εφαρμόζεται στους νευρώνες εισόδου του δικτύου και η επίδραση του διαδίδεται μέσα στο δίκτυο από επίπεδο σε επίπεδο. Στο τέλος παράγονται οι έξοδοι που είναι η πραγματική απόκριση του δικτύου. Σε αυτό το πέρασμα τα βάρη του δικτύου είναι σταθερά. Στο πέρασμα προς τα πίσω τα βάρη μεταβάλλονται σύμφωνα με τον κανόνα διόρθωσης σφαλμάτων. Συγκεκριμένα η πραγματική απόκριση του δικτύου αφαιρείται από την επιθυμητή έξοδο για να δημιουργηθεί ένα σήμα λάθους το οποίο διαδίδεται προς τα πίσω στο δίκτυο. Τα βάρη των συνάψεων προσαρμόζονται έτσι ώστε η πραγματική απόκριση να πλησιάσει την επιθυμητή. (49)

Τα σύγχρονα νευρωνικά δίκτυα συνήθως χρησιμοποιούνται για τη μοντελοποίηση σύνθετων σχέσεων μεταξύ δεδομένων εισόδου και εξόδου, για την ανακάλυψη προτύπων στα δεδομένα, ή για τον εντοπισμό στατιστικής δομής σε μία άγνωστη κοινή κατανομή πιθανότητας μεταξύ των παρατηρούμενων μεταβλητών. Συγκεκριμένα, για την αναγνώριση προτύπων σε εικόνες (που χρειάστηκε στο έργο μας) χρησιμοποιούνται κυρίως τα συνελικτικά δίκτυα (CNN), τα οποία αποτελούν ένα είδος νευρωνικών δικτύων και ακολουθούν τις βασικές αρχές που περιγράφηκαν παραπάνω. Η εκπαίδευση των δικτύων γίνεται με τον ίδιο τρόπο, με ορισμένες μικρές προσθήκες ή αλλαγές, οι οποίες είναι φυσικό επακόλουθο της διαφορετικής δομής αυτών των δικτύων. Η βασική ιδιαιτερότητα των συνελικτικών δικτύων είναι ότι είναι πολύ πιο φιλικά στην υποδοχή εικόνων ως μεταβλητή εισόδου. Οι έγχρωμες εικόνες έχουν 3 διαστάσεις: πλάτος, ύψος και βάθος όπου ως βάθος ορίζονται τα 3 επίπεδα χρωμάτων Κόκκινο-Πράσινο-Μπλε. Κάθε εικόνα περιέχει έναν αριθμό από εικονοστοιχεία (pixels), που υπολογίζεται πολλαπλασιάζοντας κάθε της διάσταση. Τα απλά νευρωνικά δίκτυα θα αποσύνθεταν την μορφή των εικόνων αυτών από τρισδιάστατη και θα δημιουργούσαν ένα διάνυσμα μίας διάστασης παραθέτοντας γραμμικά τα στοιχεία από όλες τις διαστάσεις. Για παράδειγμα, θέτοντας μια εικόνα μεγέθους 240 * 240 * 3 ως είσοδο, ένας πλήρως συνδεδεμένος, με την είσοδο, νευρώνας θα αποτελούνταν από 172800 (δηλ. 240 * 240 * 3) βάρη. Επιπρόσθετα, σε ένα κρυφό επίπεδο δεν θα είχαμε μόνο έναν νευρώνα, και ίσως να χρειαζόμασταν περισσότερα από ένα επίπεδα. Επομένως, οι υπό μάθηση παράμετροι θα αυξάνονταν δραματικά, δημιουργώντας αχανή, μη πρακτικά και επιρρεπή στην υπερεκπαίδευση δίκτυα. Επιπλέον, τα συγκεκριμένα δίκτυα θα αγνοούσαν την αυξημένη συσχέτιση που υπάρχει μεταξύ των γειτονικών εικονοστοιχείων σε σχέση με πιο απόμακρα, κάτι το οποίο αποτελεί κύριο χαρακτηριστικό γνώρισμα των φυσικών εικόνων (50). Η μεγάλη εξέλιξη των συνελικτικών δικτύων είναι ότι σαν δεδομένο εισόδου δεν δέχονται ένα διάνυσμα μίας διάστασης αλλά ένα πολυδιάστατο πίνακα. Αυτό επιτρέπει την διατήρηση της τρισδιάστατης μορφής των έγχρωμων φωτογραφιών και την καλύτερη

ανάλυσή τους. Η χρήση των ΣΝΔ εκμεταλλεύεται το γεγονός ότι η είσοδος αποτελείται από εικόνες, και περιορίζει την αρχιτεκτονική με έξυπνο τρόπο. Γενικότερα, στα ΣΝΔ, οι νευρώνες έχουν 3 διαστάσεις (πλάτος-ύψος-βάθος), και έχουν την ιδιαιτερότητα να συνδέονται σε μία μικρή περιοχή του προηγούμενου επιπέδου, αντί με όλους τους νευρώνες όπως θα γινόταν σε μία πλήρη σύνδεση (51). Η βασική διαφορά επομένως είναι ότι τα συνελικτικά δίκτυα έχουν τρισδιάστατη μορφή και από επίπεδο σε επίπεδο αλλάζει αυτή η τρισδιάστατη μορφή, ενώ προηγουμένως στα απλά νευρωνικά δίκτυα άλλαζε μόνο η μία διάσταση. Η εκπαίδευση ωστόσο, πραγματοποιείται με παρόμοιο τρόπο.

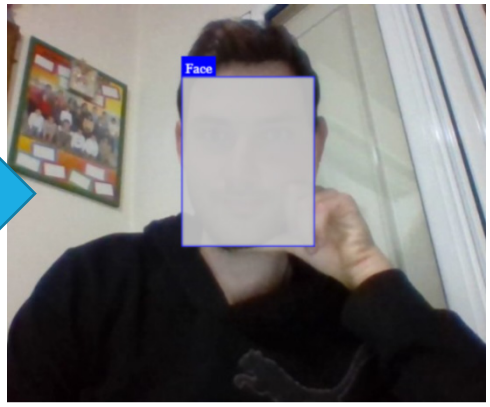
Το θετικό με το Tensorflow είναι ότι μπορούμε να παρακάμψουμε τη διαδικασία αυτής της εκπαίδευσης καθώς υπάρχουν σε αυτό και κατ' επέκταση στον wrapper που χρησιμοποιήσαμε έτοιμα pre-trained μοντέλα για αναγνώριση διάφορων αντικειμένων. Μπορούμε βέβαια να χρησιμοποιήσουμε τα δικά μας δεδομένα και να επανεκπαιδεύσουμε τα υπάρχοντα μοντέλα ή ακόμη και να δημιουργήσουμε και να εκπαιδεύσουμε ένα νευρωνικό δίκτυο από την αρχή. Οπότε στο μέλλον μπορούμε να επεκτείνουμε την εφαρμογή μας πέρα από την αναγνώριση και απόκρυψη προσώπων στην αναγνώριση και απόκρυψη πλήθους διαφορετικών τύπων αντικειμένων. Το tensorflow παρέχει υλοποιήσεις σε διάφορες γλώσσες. Η JavaScript είναι μία από αυτές, όπου υπάρχει το *tensorflow.js*. Εν τέλει η βιβλιοθήκη που χρησιμοποιήσαμε είναι λογισμικό ανοιχτού κώδικα (ΕΛ/ΛΑΚ), με άδεια MIT, η οποία παρέχει στους χρήστες απεριόριστη ελευθερία στη χρήση του κώδικα, με βασικό όρο την αναφορά του αρχικού δημιουργού, ενώ ταυτόχρονα απαλλάσσει τους δημιουργούς από κάθε ευθύνη, αν συμβεί ζημία από τη χρήση αυτού. Επιτρέπει προσωπική και εμπορική χρήση, αντιγραφή, επαναδειοδότηση (sublicensing), διανομή και τροποποίηση, αρκεί η άδεια και το copyright να συνοδεύονται μαζί με το software. Η άδεια δεν παρέχει κανενός είδους εγγύηση. (52) Ο σύνδεσμος του έργου είναι ο παρακάτω: https://github.com/justadudewhohacks/face-api.js. Έτσι, συνδυάζοντας το API με πρόσθετο κώδικα, μπορούμε να αναγνωρίσουμε πρόσωπα, όπως φαίνεται στην παρακάτω εικόνα:

Εικόνα 32. Ανίχνευση Προσώπων σε οικογενειακή φωτογραφία

Η φωτογραφία είναι royalty-free και η πηγή της είναι η ιστοσελίδα burst (https://burst.shopify.com/photos/winter-family-love?c=family).

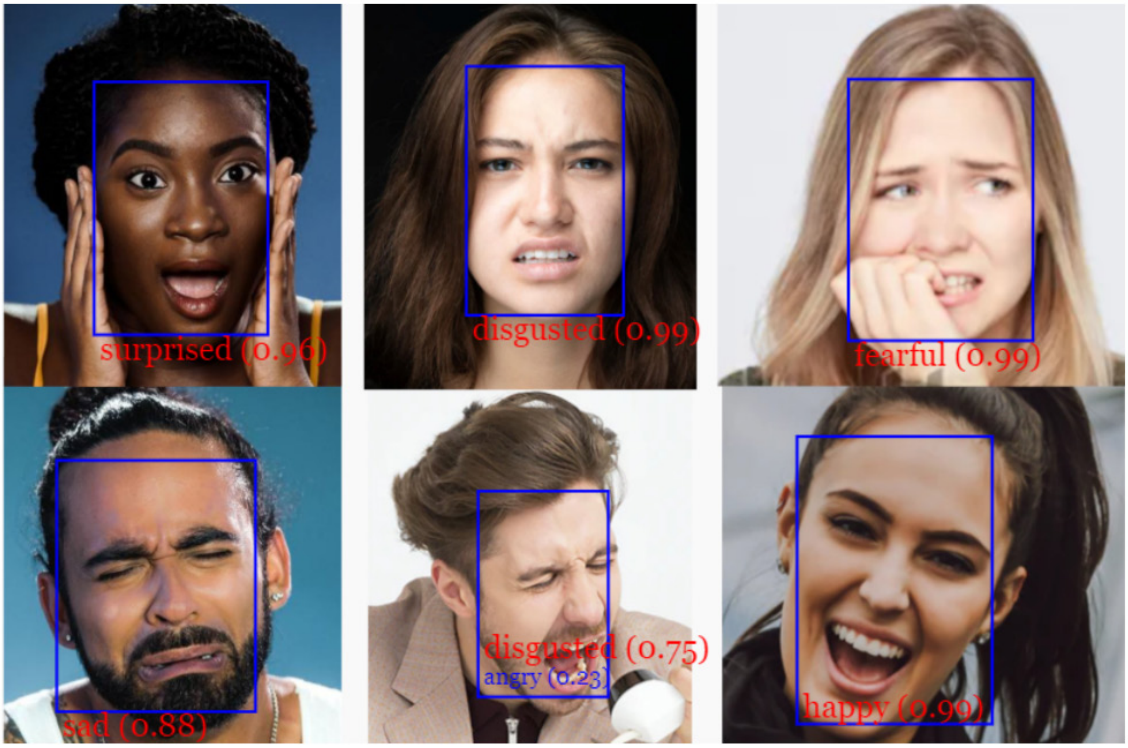
Στην υλοποίηση μας, χρησιμοποιούμε μία βιβλιοθήκη για να εξάγουμε συγκεκριμένα χαρακτηριστικά μίας φωτογραφίας. Η βιβλιοθήκης ονομάζεται *face-api.js* και είναι κατά βάση ένα περιτύλιγμα (wrapper) της βιβλιοθήκης *tensorflow*. Μέσα από αυτό το API παρέχεται η δυνατότητα να υλοποιηθεί σε Javascript το κατάλληλο classification και να εντοπίζονται δεδομένα όπως πρόσωπα και πινακίδες αυτοκινήτων (για την ώρα υπάρχει υλοποίηση μόνο για την αναγνώριση προσώπων, αλλά με αντίστοιχο τρόπο μπορεί να υλοποιηθεί μία κατάλληλη επέκταση). Η εκπαίδευση του νευρωνικού δικτύου όπως είπαμε έχει γίνει ήδη από τους κατασκευαστές του API, συνεπώς το μόνο που χρειάζεται για τη χρήση του μοντέλου είναι να φορτωθούν τα κατάλληλα μοντέλα, τα οποία θα επιτρέψουν την ανίχνευση προσώπων. Στην παρούσα υλοποίηση έχουν φορτωθεί και μοντέλα τα οποία βοηθούν στην ανίχνευση συναισθημάτων και άλλων χαρακτηριστικών, αλλά αυτά θα αφαιρεθούν στο μέλλον προκειμένου να υπάρξει κέρδος για την ταχύτητα, καθώς τα δεδομένα συναισθημάτων δεν αξιοποιούνται. Αφού φορτωθούν τα μοντέλα, πραγματοποιείται εντοπισμός των προσώπων και σχηματίζουμε στον καμβά της φωτογραφίας τα κατάλληλα τετράγωνα γύρω από τα πρόσωπα. Ταυτόχρονα μπορούμε να εισάγουμε πρόσθετες ετικέτες (labels), όπως και να δείξουμε χαρακτηριστικά για την ηλικία και τα συναισθήματα του συγκεκριμένου προσώπου, όπως θα δειχθεί παρακάτω. Κατόπιν, σκιάζονται αυτόματα τα πρόσωπα των χρηστών. Τα ανθρώπινα πρόσωπα σκιάζονται με εφαρμογή ενός συνδυασμού μάσκας και ενός ποσοστού θολώματος – blurring. Χρειάζονται περισσότερα πειράματα για να προσδιοριστεί το καλύτερο ποσοστό του blurring effect και πόσο «σκοτεινή» θα πρέπει να είναι η μάσκα. Ο συνδυασμός αυτών θα ήταν επιθυμητό ταυτόχρονα να κρύβει όσο χρειάζεται το πρόσωπο (σε βαθμό που δεν είναι αναγνωρίσιμο), αλλά ταυτόχρονα να διατηρεί στον μέγιστο δυνατό βαθμό την τεχνοτροπία της φωτογραφίας. Στη συνέχεια παρατίθενται ένα απλό παράδειγμα θολώματος.

Εικόνα 33. Αρχική φωτογραφία, πριν την εφαρμογή του αλγορίθμου



Εικόνα 34. Τελική φωτογραφία, μετά την εφαρμογή του αλγορίθμου

Όπως προαναφέρθηκε, μπορούν να αναγνωριστούν και συναισθήματα καθώς και ηλικίες. Στη συνέχεια παρατίθεται μία συνοπτική παρουσίαση αυτών των λειτουργιών καθώς και μία σύγκριση όσον αφορά τον τομέα των συναισθημάτων μεταξύ των διάφορων μοντέλων που μας προσφέρει η βιβλιοθήκη.



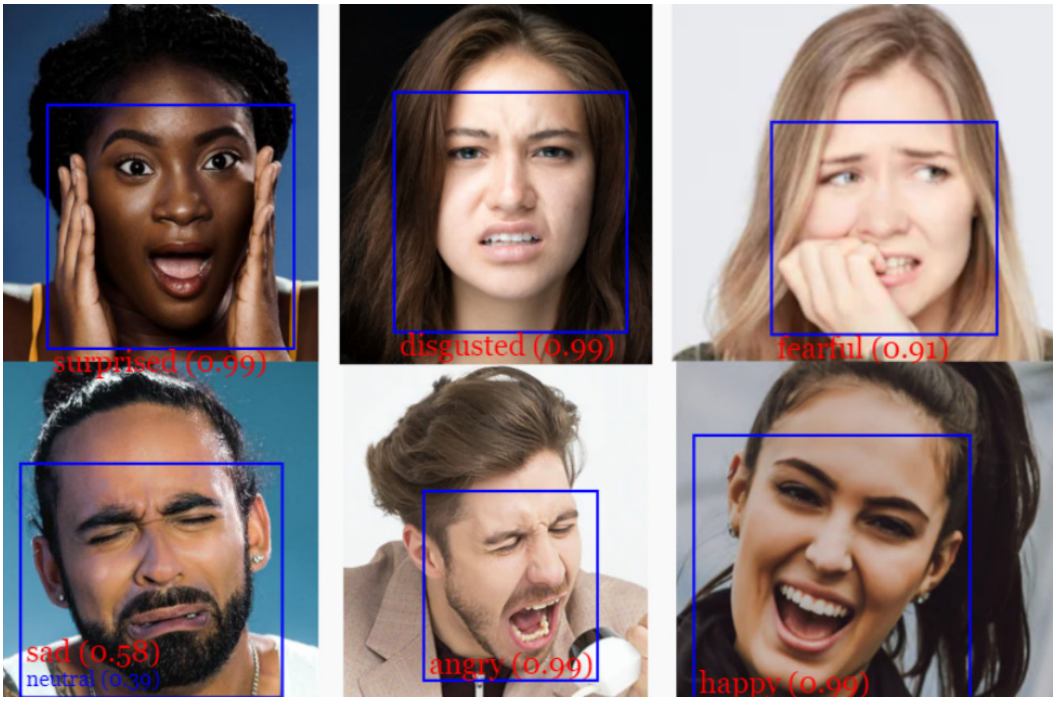
Εικόνα 35. Προσδιορισμός Συναισθημάτων με βάση το μοντέλο SSD mobilenet V1

Γενικά, ανάλογα με τα δεδομένα, την εκπαίδευση που έχει γίνει και την αρχιτεκτονική που χρησιμοποιείται, παρατηρούμε ότι έχουμε διαφορετικά αποτελέσματα, όπως ήταν άλλωστε αναμενόμενο. Από τη βιβλιοθήκη μας παρέχονται 3 μοντέλα:

- Το SSD mobilenet V1 (53), το οποίο είναι το πιο βαρύ και ολοκληρωμένο από όλα για τις περισσότερες εφαρμογές. Είναι βασισμένο στο κλασικό μοντέλο - CNN MobileNet V1, αλλά έχουν εισαχθεί κάποια επιπρόσθετα επίπεδα για ακριβέστερη πρόβλεψη των πλαισίων. Έχει γενικά πολύ καλή ακρίβεια και μπορεί σε συγκεκριμένες περιπτώσεις να χρησιμοποιηθεί για εφαρμογές πραγματικού χρόνου,
- το MTCNN (Multi-task Cascaded Convolutional Neural Networks), το οποίο είναι ένα πολύ ελαφρύ μοντέλο, πετυχαίνει αρκετά ακριβή αποτελέσματα και δουλεύει καλά σε εφαρμογές πραγματικού χρόνου, ενώ τέλος υπάρχει και
- το Tiny Face Detector, το οποίο είναι πιο αποδοτικό σε ταχύτητα από το SSD mobilenet V1 αλλά υστερεί σε θέμα ακρίβειας. Μπορεί και αυτό να χρησιμοποιηθεί σε εφαρμογές πραγματικού χρόνου, καθώς είναι εξαιρετικά ελαφρύ. Στο σχετικά απλό παράδειγμά μας, είχαν και τα 3 μοντέλα αρκετά καλή απόδοση αφού και τα 3 προσδιόρισαν σωστά (σχεδόν) όλα τα συναισθήματα.

Στην εικόνα (Εικόνα 35), φαίνονται μερικά αποτελέσματα της χρήσης του SSD mobilenet V1 για αναγνώριση συναισθημάτων. Το SSD mobilenet V1, θεωρούμε ότι έκανε ένα λάθος στην εικόνα 5, όπου το συναίσθημα ήταν περισσότερο “anger” παρά “disgustment”, αν και ήταν λίγο δύσκολο παράδειγμα και σε ένα βαθμό υποκειμενικό. Με το πρώτο μοντέλο, παρατηρούμε ότι έχουμε μία ανακρίβεια στον προσδιορισμό των συναισθημάτων στην 5η εικόνα όπως αναφέραμε, αλλά γενικότερα το μοντέλο έχει μεγάλη ακρίβεια και έχει εξάγει τις σωστές ετικέτες συναισθημάτων. Το Tiny Face Detector είχε καλή ακρίβεια,

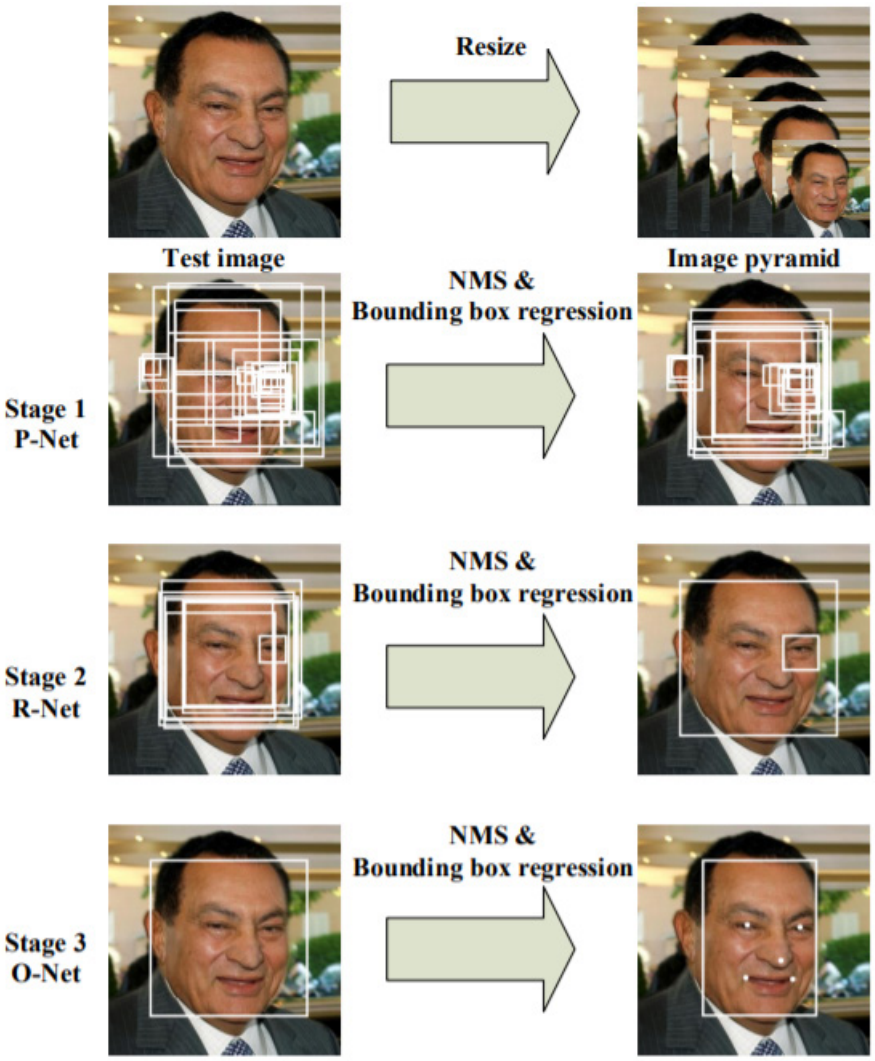
αφού προσδιόρισε σωστά τα περισσότερα συναισθήματα, απλά με μικρότερη βεβαιότητα (confidence) από τα υπόλοιπα και έκανε σχεδόν λάθος στην εικόνα 4 όπου το συναίσθημα ήταν ξεκάθαρα “sad” (ενώ το μοντέλο δίνει πιθανότητα 0.39 για “neutral”). Ταυτόχρονα είναι αξιοσημείωτα αποδοτικό σε ταχύτητα, καθώς είναι μία βελτιωμένη έκδοση του μοντέλου Tiny YOLO V2 (54) (55) (56), σύμφωνα με τους κατασκευαστές του API. Τα αποτελέσματα του Tiny Face Detector φαίνονται στην Εικόνα 36.



Εικόνα 36. Προσδιορισμός συναισθημάτων με το μοντέλο TinyFaceDetector

Τέλος παρατηρήσαμε ότι με τη χρήση του μοντέλου MTCNN, είχαμε τα καλύτερα αποτελέσματα. Το MTCNN είναι επίσης ένα αρκετά ελαφρύ μοντέλο και μπορεί να χρησιμοποιηθεί και σε εφαρμογές πραγματικού χρόνου. Θα μπορούσε να είναι συμπτωματικό να σημειωθούν τα καλύτερα αποτελέσματα στο συγκεκριμένο παράδειγμα, ή ίσως το σύστημα με τα 5 Point Face Landmarks να δουλεύει καλύτερα στην ανίχνευση απλών συναισθημάτων. Χρειαζόμαστε σίγουρα περισσότερα πειράματα για να κάνουμε μία καλύτερη σύγκριση μεταξύ των τριών. Ο αλγόριθμος έδινε επίσης μεγαλύτερα ποσοστά βεβαιότητας για τα αποτελέσματα και είχε τέλεια ακρίβεια, όπως φαίνεται στην Εικόνα 37.

Αναλύοντας λίγο περισσότερο τον αλγόριθμο, το MTCNN (Multi-task Cascaded Convolutional Neural Networks), αυτός αποτελείται από 3 επίπεδα. Χρησιμοποιείται για να αναγνωρίσει τα σημεία της φωτογραφίας που είναι πρόσωπα και να θέσει τα πλαίσια γύρω από αυτά τα πρόσωπα. Για να το κάνει αυτό χρησιμοποιεί μία απεικόνιση με μόλις 5 σημεία ενδιαφέροντος (5 Point Face Landmarks.) Για να κάνουμε μία σύγκριση ο αλγόριθμος SSD Mobilenet V1 χρειάζεται 68 σημεία, για αυτό και έχει στις περισσότερες περιπτώσεις καλύτερη ακρίβεια.



Εικόνα 37. Τα 3 στάδια εκτέλεσης του convolutional δικτύου.

Κάθε επίπεδο βελτιώνει σταδιακά τα αποτελέσματα της αναγνώρισης. Αυτό επιτυγχάνεται λαμβάνοντας τις εισόδους από ένα CNN (Convolutional Neural Network), το οποίο επιστρέφει τα κατάλληλα πλαίσια με τα scores τους, ακολουθούμενα από το non max suppression (χρησιμοποιείται για την επιλογή του κατάλληλου πλαισίου). (57)

Στο 1ο στάδιο, η εικόνα εισόδου μικραίνει πολλές φορές έτσι ώστε να δημιουργηθεί μία πυραμίδα εικόνα και κάθε κλιμακούμενη έκδοση της εικόνας περνά μέσω του CNN.

Στα στάδια 2 και 3 εξάγονται ψηφίδες (patches) από την εικόνα για κάθε πλαίσιο και αλλάζουμε το μέγεθός τους (24x24 στο στάδιο 2 και 48x48 στο στάδιο 3) και τα προωθούμε μέσω του CNN αυτού του σταδίου. Εκτός από τα πλαίσια οριοθέτησης και τα σκορ, το στάδιο 3 υπολογίζει επιπλέον 5 σημεία ορόσημων προσώπου για κάθε κουτί.

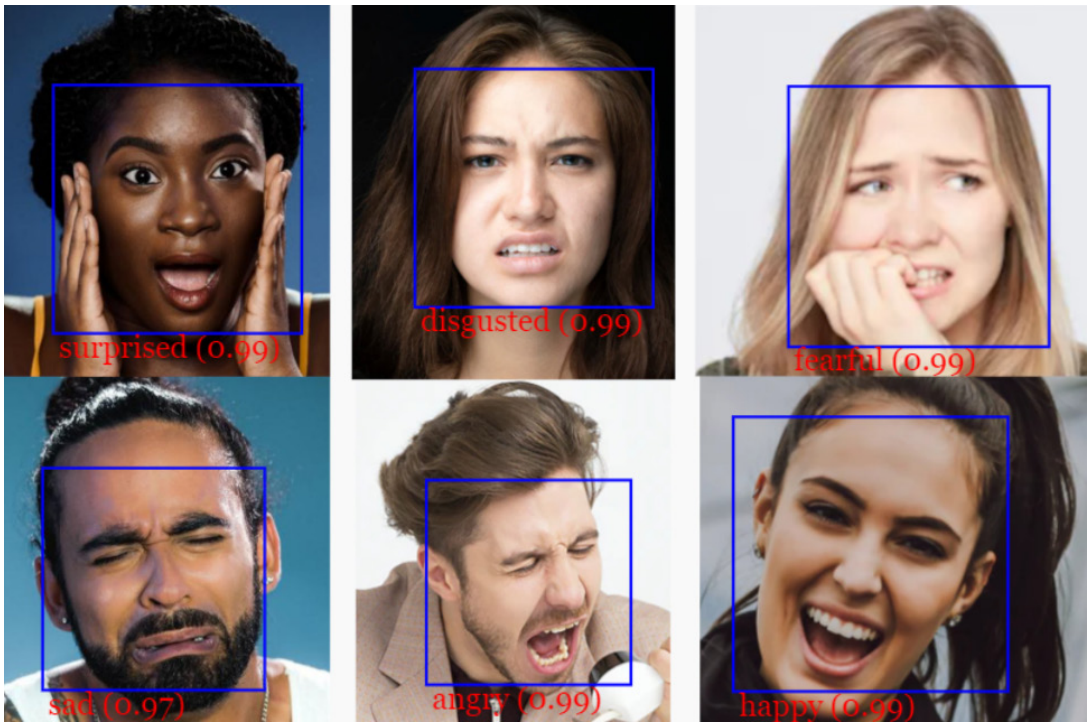
Αφού εργαστούμε με τους τρεις αλγορίθμους καταλαβαίνουμε και στην πράξη ότι μπορούμε με τον MTCNN να πάρουμε αρκετά καλά αποτελέσματα με με πολύ καλή απόδοση σε σύγκριση με το SSD Mobilenet v1.

Πλεονεκτήματα:

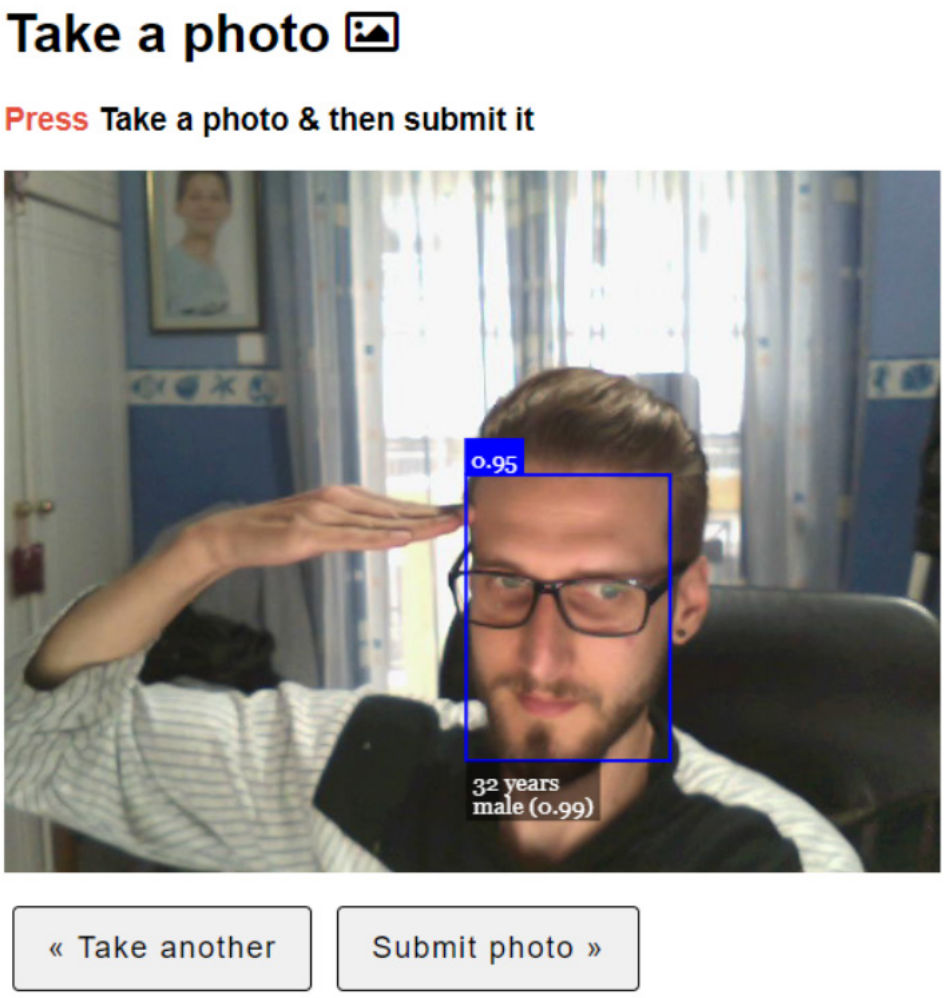
- μικρότεροι χρόνοι συμπερασμού (inference) (μεγαλύτερη ταχύτητα ανίχνευσης).
- ταυτόχρονη ανίχνευση 5 σημείων προσώπου.
- Πολύ μικρότερο μέγεθος μοντέλου: μόνο ~2MB σε σχέση με τα ~6MB που χρειάζονται τα βάρη στο quantized SSD Mobilenet v1.
- Επιτρέπει αλλαγές (configurable): Υπάρχουν παράμετροι που μπορούμε να ρυθμίσουμε για να πετύχουμε καλύτερη απόδοση ανάλογα με την εκάστοτε εφαρμογή.

Μειονεκτήματα:

- Λιγότερο ακριβής από το SSD Mobilenet v1 (αν και δεν ισχύει στο παράδειγμα που παρουσιάσθηκε). (58)



Εικόνα 38. Προσδιορισμός συναισθημάτων με το μοντέλο MTCNN



Εικόνα 39. Σύμφωνα με τον αλγόριθμο, στην εικόνα απεικονίζεται ένα ανθρώπινο πρόσωπο (95% πιθανότητα), 32 χρονών ενδεχομένως και κατά 99% γένους αρσενικού

6.1.5 Χρήση κατάλληλων τεχνικών μετάδοσης για την προστασία των δεδομένων κατά τη μεταφορά.

Σε αυτήν την ενότητα θα μιλήσουμε για το πρωτόκολλο HTTPS και τα πιστοποιητικά SSL των οποίων η χρήση κρίνεται απαραίτητη όταν το site φιλοξενηθεί σε κάποιον διακομιστή, έτσι ώστε να διασφαλιστεί η προστασία των δεδομένων στο επίπεδο μεταφοράς. Αρχικά το απλό HTTP βγαίνει από τα αρχικά του Hypertext Transfer Protocol και σημαίνει πρωτόκολλο μεταφοράς υπερκειμένου. Χρησιμοποιείται για την αποστολή πληροφοριών μεταξύ δύο συστημάτων και χρησιμοποιείται συχνότερα μεταξύ ενός διακομιστή ιστού (web server) και ενός οικιακού υπολογιστή. Το HTTPS αντιπροσωπεύει το ασφαλές πρωτόκολλο μεταφοράς υπερκειμένου με την προσθήκη του S και βγαίνει από το Hypertext Transfer Protocol Secure. Το συγκεκριμένο πρωτόκολλο, χρησιμοποιείται επίσης για την αποστολή πληροφοριών μεταξύ συστημάτων, αλλά με περισσότερη ασφάλεια. Στις πρώιμες μέρες του Διαδικτύου, όλοι οι εξυπηρετητές λειτουργούσαν με το HTTP. Το HTTP εισήχθη το 1991 ενώ το HTTPS αν και εισήχθη το 1994, για μεγάλο χρονικό διάστημα χρησιμοποιείτο μόνο το HTTP. Αρχικά το πρωτόκολλο δεν μετέφερε καμία πληροφορία σχετικά με το πρόγραμμα-πελάτη και η

μόνη επιλογή που υπήρχε ήταν η αίτηση από τον εξυπηρετητή μιας σελίδας υπερκειμένου το οποίο περιείχε κείμενο με σήμανση HTML καθώς και εικόνες. Το HTTP χρησιμοποιεί για την επίτευξη επικοινωνίας το κλασικό πρωτόκολλο του TCP, όπου ο client πραγματοποιεί μια σύνδεση TCP με τον host χρησιμοποιώντας (α) το domain name ή τη διεύθυνση IP και (β) τον αριθμό της θύρας που έχει δοθεί στη διεύθυνση. Αν ο αριθμός της θύρας δεν έχει ορισθεί ο προεπιλεγμένος είναι το 80 για το απλό HTTP και το 443 για το HTTPS. Έπειτα ο server αποδέχεται τη σύνδεση. Αν και το HTTP πρωτόκολλο σχεδιάστηκε για χρήση στο Ιστό, υποστηρίζει λειτουργίες που είναι πιο γενικευμένες από ό,τι απαιτείται για τη λειτουργία του web μέσα από εφαρμογές πλοήγησης. Οι λειτουργίες αυτές ονομάζονται μέθοδοι. Κάθε αίτηση αποτελείται από μία ή και περισσότερες γραμμές κειμένου ASCII. Η πρώτη λέξη της πρώτης γραμμής της αίτησης είναι το όνομα της ζητούμενης μεθόδου. Κάποιες από τις ενσωματωμένες μεθόδους αίτησης είναι οι παρακάτω:

- GET
- HEAD
- POST
- PUT
- DELETE
- TRACE
- CONNECT
- OPTIONS

Κάθε αίτηση λαμβάνει μια απάντηση η οποία αποτελείται από μια γραμμή κατάστασης και πιθανό πρόσθετες πληροφορίες. Η γραμμή κατάστασης περιέχει ένα τριψήφιο κωδικό κατάστασης, ο οποίος δηλώνει κατά πόσον εξυπηρετήθηκε η αίτηση και αν δεν εξυπηρετήθηκε, γιατί συνέβη αυτό. Το πρώτο ψηφίο χρησιμοποιείται για τη υποδιαίρεση τον αιτήσεων σε 5 κατηγορίες.

Οι κωδικοί 1xx χρησιμοποιούνται σπανίως στη πράξη. Παράδειγμα: 100 = ο διακομιστής συμφωνεί να χειριστεί την αίτηση του πελάτη.

Οι κωδικοί 2xx σημαίνουν ότι ο χειρισμός της αίτησης έγινε με επιτυχία και ότι επιστρέφεται το περιεχόμενο, αν υπάρχει. Παράδειγμα: 200 = η αίτηση πέτυχε, 204 = δεν υπάρχει περιεχόμενο.

Οι κωδικοί 3xx λένε στον πελάτη να αναζητήσει την πληροφορία αλλού, είτε χρησιμοποιώντας μια διαφορετική διεύθυνση URL, είτε εξετάζοντας την κρυφή μνήμη. Παράδειγμα: 301 = η σελίδα μετακινήθηκε.

Οι κωδικοί 4xx σημαίνουν ότι η αίτηση απέτυχε λόγω κάποιοι σφάλματος του πελάτη, όπως άκυρη αίτηση ή ανύπαρκτη σελίδα. Παράδειγμα: 403 = απαγόρευση πρόσβασης στη σελίδα, 404 = η σελίδα δεν βρέθηκε.

Οι κωδικοί 5xx σημαίνουν ότι η αίτηση απέτυχε λόγω κάποιου σφάλματος από την πλευρά του διακομιστή. Παράδειγμα 500 είναι το γενικευμένο “Internal Server Error”

Η ανάπτυξη του HTTP έγινε υπό την εποπτεία του World Wide Web Consortium και του Internet Engineering Task Force (IETF). Το απλό πρωτόκολλο HTTP δεν εγγυάται καμία ασφάλεια. (59)

Το διαδίκτυο το 1991 δεν ήταν ευρέως διαθέσιμο και προσβάσιμο από όλους, αλλά και οι προσωπικές συσκευές ήταν ακριβές και πολύ λιγότερες. Δεν ήταν όμως μόνο τα ποσοστά

χρήσης του διαδικτύου αλλά και η φύση των πληροφοριών που αποστέλλονται. Με την εξάπλωση του ηλεκτρονικού ταχυδρομείου μέσω των πανεπιστημίων, εμφανίστηκε η ανάγκη για ασφαλή κανάλια επικοινωνίας. Το διαδίκτυο συνέχισε να ωριμάζει και με την άφιξη του ηλεκτρονικού εμπορίου οι τράπεζες άρχισαν να χρησιμοποιούν το ασφαλές πρωτόκολλο στις μεθόδους πληρωμής των ηλεκτρονικών καταστημάτων. Οι πληροφορίες έπρεπε να αποστέλλονται με ασφάλεια και ο μόνος τρόπος για να γίνει αυτό ήταν το πρωτόκολλο HTTPS. Το HTTPS χρησιμοποιούσε αρχικά το πρωτόκολλο Secure Socket Layer για την ασφαλή μετάδοση δεδομένων. Το SSL όπως είναι γνωστό αναπτύχθηκε για αυτόν ακριβώς τον σκοπό. Αρχικά, χρησιμοποιήθηκε από ιστότοπους ηλεκτρονικού ταχυδρομείου, ηλεκτρονικού εμπορίου και σε portals πληρωμών όπως την Paypal. Ο στόχος του SSL, ήταν και παραμένει να μην μπορεί να χρησιμοποιηθεί από κάποια μη ασφαλή σελίδα, που προσπαθεί να ξεγελάσει τους επισκέπτες της, για την διανομή κακόβουλου λογισμικού από το πρόγραμμα περιήγησης.
Αξίζει να σημειωθεί ότι το Secure Socket Layer έχει εξελιχθεί από τότε που αναπτύχθηκε για πρώτη φορά. Έχει αντικατασταθεί από το Security Layer Security (TLS). Το TLS παρέχει ένα πολύ καλύτερο επίπεδο ασφάλειας προστασίας της ιδιωτικής ζωής. Σήμερα, η ανάγκη για ασφάλεια δεν αφορά απλά την μη διαρροή ευαίσθητων πληροφοριών, αλλά και για τον αποκλεισμό παρακολούθησης από διάφορους τρίτους.

Τα παραδοσιακά IDS, IPS και firewalls βασίζονται στα ports για να διαπιστώσουν ποιους μηχανισμούς θα χρησιμοποιήσουν για ανίχνευση, για ανάλυση και ποιες υπογραφές να προσέξουν. Τα Malwares βασίζονται στο secure socket layer(SSL) encryption με σκοπό να κρύψουν το malicious περιεχόμενο όπως το CnC traffic. Το SSL χρησιμοποιεί μεθόδους κρυπτογράφησης των δεδομένων που ανταλλάσσονται μεταξύ δύο συσκευών (συνηθέστερα Ηλεκτρονικών Υπολογιστών) εγκαθιδρύοντας μία ασφαλή σύνδεση μεταξύ τους μέσω του διαδικτύου. Το πρωτόκολλο(SSL) είναι πρωτόκολλο επικοινωνίας και χρησιμοποιείται ευρέως για ηλεκτρονικές αγορές, για χρηματικές συναλλαγές, για music streaming και εξυπηρετεί πολύ για malware delivery σε κάποιον ανυποψίαστο χρήστη.Tunneling είναι η διαδικασία χρήσης υποδομής των ηλεκτρονικών δικτύων για τη μετακίνηση στοιχείων μεταξύ των δικτύων. Τα στοιχεία προς μετακίνηση είναι δυνατόν να εφαρμόζονται σε πλαίσια ή πακέτα πρωτοκόλλου τα οποία πραγματοποιούν διαδρομή ηλεκτρονικά σε ένα διαδικτυακό μονοπάτι το λεγόμενο tunnel. Από τη στιγμή που τα πακέτα φτάσουν στον προορισμό τους, τότε αφαιρούνται οι πρόσθετες κεφαλίδες. Το Tunneling καθιστά τα IDS και τα firewalls αναποτελεσματικά στα smart grids. Επιτρέπει στους attackers να κρύψουν το κακόβουλο λογισμικό μέσα σε νόμιμες εφαρμογές και πρωτόκολλα σε μια κρυπτογραφημένη κίνηση δεδομένων. Εγκαθιστώντας proxy servers σε συσκευές infected επιτρέπουν στα bots να κρύψουν την επικοινωνία τους καθώς και οποιοδήποτε ίχνος. Ανώνυμα εργαλεία όπως το Tor, Himachi και το UltraSurf βοηθούν τους attackers. Τα social media είναι ένας χώρος που είναι επιρρεπής σε malware infection, και CnC. Οι συσκευές IoT χρησιμοποιούν web-based e-mail, instant messaging, web-based file transfer, blogs, message boards, and microblogging και είναι εύκολο κάποιος ανυποψίαστος χρήστης να τρέξει κατά λάθος στη συσκευή του ένα κακόβουλο script μέσω μιας διαφήμισης ή μιας εικόνας και όλο αυτό μπορεί να οδηγήσει στη κατάρρευση ενός smart grid network. Για αυτό το λόγο με σκοπό να αυξηθεί η ασφάλεια οι εφαρμογές χρησιμοποιούν SSL encryption σαν προεπιλεγμένη ασφάλεια στη κίνηση δεδομένων. Από την οπτική της ασφάλειας, η παροχή της δυνατότητας σε μία κακόβουλη εφαρμογή να χρησιμοποιεί κρυπτογραφημένη επικοινωνία είναι ελάττωμα, καθώς η κρυπτογράφηση του SSL διευκολύνει το malware να κρύψει την παρουσία του και το περιεχόμενο της επικοινωνίας. Ο attacker μπορεί να κρυφτεί μέσα σε μια SSL encrypted connection μεταξύ του χρήστη και της εφαρμογής. Παλιά τα malware είχαν τη

δυνατότητα να πολλαπλασιάζονται και να μολύνουν πολλούς hosts. Τα εξελιγμένα malware είναι πιο ποιοτικά και δε χρειάζεται να το κάνουν αυτό καθώς δίνουν τη δυνατότητα στον attacker να χειρίζεται το malware απομακρυσμένα. Μια Deadly attack μπορεί να γίνει από έναν και μόνο υπολογιστή. Κάποια ανεπτυγμένα malware έχουν επίσης συγκεκριμένα κομμάτια κώδικα που έχουν σκοπό να αλλάξουν το malware signature.

6.1.6 Διασφάλιση σχέσεων εμπιστοσύνης μέσω σύναψης φιλίας (Friend requests)

Μια ακόμη λειτουργία που υλοποιείται στο σύστημα κοινωνικής δικτύωσης είναι τα αιτήματα φιλίας. Όπως και σε άλλα κοινωνικά δίκτυα για να διασφαλιστεί η ιδιωτικότητα και η ασφάλεια των διαπροσωπικών σχέσεων των χρηστών κρίθηκε απαραίτητο ένας χρήστης να μπορεί να ανταλλάξει μηνύματα μόνο με φίλους. Σε άλλα κοινωνικά δίκτυα υπάρχει επίσης η δυνατότητα αποστολής μηνυμάτων σε προφίλ που είναι δημόσια (public). Εμείς κρίναμε απαραίτητο να μην επιτρέπεται ούτε αυτή η λειτουργικότητα, διότι υπάρχει και πάλι ο κίνδυνος να χρησιμοποιηθεί το σύστημα επικοινωνίας με τρόπο κακόβουλο. Βέβαια ο διαχωρισμός των προφίλ σε public και private είναι λειτουργία που θα θέλαμε να υλοποιήσουμε στο μέλλον. Γενικότερα οι άμεσες και προσωπικές αλληλεπιδράσεις μεταξύ χρηστών (π.χ. chatting) μέσα στο κοινωνικό δίκτυο, ειδικότερα αυτές που περιλαμβάνουν ανταλλαγή κειμένων ή/και πολυμέσων μπορούν να γίνουν μόνο από άτομα που έχουν συνάψει φιλία. Για να γίνει ένας χρήστης φίλος με κάποιον άλλο θα πρέπει να στείλει αίτημα φιλίας και ο δεύτερος να αποδεχτεί το αίτημα φιλίας του πρώτου. Αυτός ο περιορισμός εισήχθη με σκοπό το κοινωνικό μας δίκτυο να είναι πιο ασφαλές, αφού επιτρέπεται η αλληλεπίδραση των χρηστών μόνο με άτομα που στη ουσία τους έχουμε δώσει εξουσιοδότηση εμείς και τα θεωρούμε έμπιστα. Αν το αίτημα δεν γίνει δεκτό, τότε ο χρήστης δεν μπορεί να στείλει άλλο αίτημα για μεγάλο χρονικό διάστημα, προκειμένου να αποφεύγεται η υπερφόρτωση ενός χρήστη με αιτήματα από κακόβουλους ή επίμονους και ενοχλητικούς χρήστες.

Μία άλλη ιδέα που εξετάστηκε είναι να υπάρχει μία μπάρα αναζήτησης και αν ο χρήστης γράψει επακριβώς το username ενός άλλου χρήστη, τότε και μόνο τότε να έχει τη δυνατότητα να αλληλεπιδράσει μαζί του, ενώ να δίνεται στον άλλο χρήστη η δυνατότητα να τον μπλοκάρει. Όμως αναλύοντας το σχέδιο αυτό, θεωρήθηκε ότι θα έκανε την εφαρμογή μας αρκετά δύσκολη στη χρήση, ενώ θα υπήρχαν και πιθανότητες αποκάλυψης ονομάτων χρηστών με εξαντλητικές αναζητήσεις (brute force). Τέλος, θέλαμε να αποφύγουμε λειτουργίες όπως το απλό follow καθώς, αν και η λειτουργία δουλεύει καλά σε αρκετά δίκτυα έχει ζητήματα ιδιωτικότητας, καθώς οποιοσδήποτε χρήστης μπορεί να κάνει follow κάποιον άλλο και να αλληλεπιδράσει μαζί του, μεταφέροντας στον άλλο χρήστη την υπευθυνότητα λήψης περιοριστικών μέτρων (π.χ. block). Έτσι καταλήξαμε στην λειτουργία αυτή που είναι πρακτικά μία επέκταση των αιτημάτων φιλίας τα οποία γίνονται σε πολλά κοινωνικά δίκτυα όπως π.χ. το Facebook.

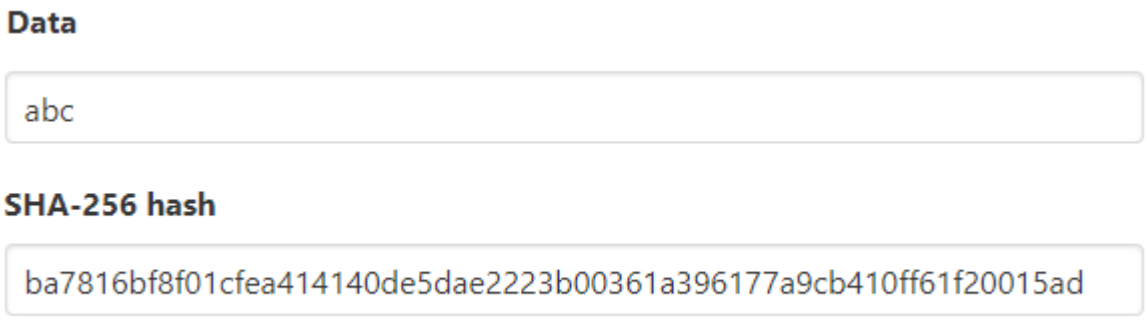
6.1.7 Κρυπτογράφηση κωδικών και ευαίσθητων δεδομένων στην βάση δεδομένων

6.1.7.1 Οι συναρτήσεις σύνοψης (hash functions) και ο ρόλος τους στην ασφάλεια

Πριν αναφερθούμε στο πρόβλημα παραθέτουμε μία σύντομη εισαγωγή στις συναρτήσεις κερματισμού (hash functions), καθώς αυτές διαδραματίζουν σημαντικό ρόλο στην κρυπτογράφηση ευαίσθητων στοιχείων που αποθηκεύονται στη βάση δεδομένων, συμπεριλαμβανομένων και των κωδικών πρόσβασης. Οι συναρτήσεις κερματισμού είναι συναρτήσεις οι οποίες δέχονται σαν είσοδο μια ακολουθία χαρακτήρων αυθαίρετου μήκους και παράγουν ένα μήνυμα σταθερού μεγέθους (γενικά μικρότερο) που ονομάζεται τιμή κερματισμού (hash value). Οι μονόδρομες συναρτήσεις σύνοψης (MD4, MD5, SHA) είναι συναρτήσεις οι οποίες λειτουργούν μόνο προς την μία κατεύθυνση, δηλαδή ενώ είναι εύκολο να υπολογιστεί μια τιμή κερματισμού για κάποιο δεδομένο όρισμα, είναι υπολογιστικά ανέφικτο να βρεθεί το όρισμα στο οποίο αντιστοιχεί μια συγκεκριμένη τιμή κερματισμού. Με άλλα λόγια είναι εύκολο η συνάρτηση να υπολογιστεί, αλλά “δύσκολο” να αντιστραφεί. Ο όρος «υπολογιστικά ανέφικτο» αναφέρεται στο γεγονός ότι είναι εν τέλει δυνατό να αντιστρέψουμε μια μονόδρομη συνάρτηση, ιδίως με χρήση εξαντλητικής αναζήτησης , ο χρόνος όμως που απαιτείται για την επίτευξη της αντιστροφής αυξάνεται εκθετικά σε σχέση το μήκος ορίσματος, συνεπώς η αντιστροφή θα χρειαστεί πάρα πολύ χρόνο. Υπάρχουν ωστόσο και άλλοι τύποι επιθέσεων, όπως π.χ. επιθέσεις βασισμένες σε λεξικά, οι οποίες μπορεί να είναι αποτελεσματικές για την εύρεση απλών συνθηματικών ή άλλων ευαίσθητων στοιχείων.

Σε μία καλά σχεδιασμένη μονόδρομη συνάρτηση κερματισμού είναι επίσης δύσκολο να βρεθούν δύο ορίσματα που δίνουν την ίδια τιμή. Αυτό σημαίνει ότι έχουμε μικρή πιθανότητα συγκρούσεων. Οι μονόδρομες συναρτήσεις σύνοψης μπορούν να διασφαλίσουν σε μεγάλο βαθμό το απόρρητο των πληροφοριών, καθώς είναι πάρα πολύ δύσκολο να βρεθεί η είσοδος που αντιστοιχεί στο αρχικό όρισμα από την τιμή κερματισμού. Η αλλαγή επίσης έστω και ενός bit στο αρχικό μήνυμα προκαλεί ολοκληρωτική αλλαγή στην έξοδο, αλλάζοντας κατά μέσο όρο τα μισά bits της τιμής σύνοψης, χαρακτηριστικό πολύ χρήσιμο για την υλοποίηση αξιόπιστων μηχανισμών κρυπτογράφησης.

Οι συναρτήσεις αυτές χρησιμοποιούνται σε πλήθος εφαρμογών, όπως σε ψηφιακές υπογραφές, όπου η σύνοψη του μηνύματος κρυπτογραφείται με το ιδιωτικό κλειδί (ασύμμετρη κρυπτογράφηση) του ιδιοκτήτη/αποστολέα του μηνύματος και ύστερα ο έλεγχος της ψηφιακής υπογραφής γίνεται με την ανάποδη διαδικασία: υπολογίζεται η σύνοψη του μηνύματος, αποκρυπτογραφείται η ψηφιακή υπογραφή με το δημόσιο κλειδί του ιδιοκτήτη/αποστολέα και συγκρίνονται οι δύο συνόψεις. Αν είναι ίδιες το έγγραφο ανήκει στον ιδιοκτήτη/αποστολέα που το έχει υπογράψει και δεν έχει παραποιηθεί. Επίσης hash functions όπως το MD5 χρησιμοποιούνται για κωδικούς ασφαλείας όπου στη θέση τους αποθηκεύεται η σύνοψη του κωδικού και μετά ελέγχονται οι συνόψεις για το αν ο κωδικός είναι έγκυρος. Βέβαια, η ασφάλεια και εδώ δεν είναι απόλυτα διασφαλισμένη, καθώς είναι δυνατή η εύρεση του κωδικού με εξαντλητικές επιθέσεις (Brute force attack) ή επιθέσεις βασισμένες σε λεξικά και τα σωστά εργαλεία. Τέλος οι κώδικες πιστοποίησης MACs είναι επίσης μονόδρομες συναρτήσεις σύνοψης οι οποίες βασίζονται σε μυστικό κλειδί έτσι ώστε μόνο κάποιος που γνωρίζει το κλειδί αυτό μπορεί να επιβεβαιώσει την τιμή σύνοψης.



Εικόνα 40. Παράδειγμα κρυπτογράφησης κειμένου με τον αλγόριθμο SHA-256

6.1.7.2 Υλοποίηση κρυπτογράφησης ευαίσθητων δεδομένων στη βάση δεδομένων και ο κρυπταλγόριθμος που χρησιμοποιήθηκε

Οι κωδικοί δεν αποθηκεύονται στη βάση δεδομένων σε μορφή απλού κειμένου (plaintext) αλλά σε μορφή κρυπτογραφημένου κειμένου, έτσι ώστε ακόμα και αν αυτοί διαρρεύσουν, να μην είναι δυνατή η ανάκτηση των πραγματικών κωδικών. Αν οι κωδικοί βέβαια είναι πολύ απλοί της μορφής π.χ 1234, abcd, password κ.λπ., τότε όπως αναφέρθηκε θα είναι εφικτό με μία επίθεση λεξικού να εντοπισθούν. Για τον λόγο αυτό, σε προηγούμενο βήμα έχουμε υποχρεώσει τον χρήστη να χρησιμοποιεί ασφαλείς κωδικούς που να είναι π.χ. μεγαλύτεροι σε μήκος από κάποιους χαρακτήρες και να περιέχουν αριθμούς, γράμματα και σύμβολα. Μπορούμε επίσης να χρησιμοποιήσουμε και κάποια τεχνική salting για να αυξηθεί η αντοχή του συστήματος κρυπτογράφησης απέναντι σε επιθέσεις με χρήση προϋπολογισμένων λεξικών από κρυπτογραφήματα κωδικών.

Ο αλγόριθμος που χρησιμοποιήσαμε στην υλοποίησή μας είναι ο SHA256, ένας αρκετά ασφαλής αλγόριθμος και ταυτόχρονα σχετικά αποδοτικός σε ταχύτητα. Να αναφέρουμε ότι για ερευνητικούς σκοπούς χρησιμοποιήσαμε αρχικά τον αλγόριθμο MD5, ο οποίος ωστόσο δεν είναι ιδιαίτερα ασφαλής καθώς έχουν εισαχθεί γνωστές αποδοτικές τεχνικές για την παραβίασή του. Με τον MD5, παρατηρήσαμε ότι ο μέσος χρόνος για την αποκρυπτογράφηση-σπάσιμο(cracking) κωδικών για κωδικούς της τάξης των 8 ψηφίων, οι οποίοι περιείχαν μικρά, κεφαλαία γράμματα και αριθμούς μόνο, ήταν λιγότερο από μισή ώρα, συνήθως μόλις 22-25 λεπτά. Το configuration του συγκεκριμένου φορητού συστήματος αντιστοιχούσε σε ένα κοινό μηχάνημα, για τα σημερινά δεδομένα. Αποτελούταν από ένα επεξεργαστή Intel i7 7500U, 2 cores / 4 threads, της σειράς Kaby Lake στα 2.7GHz, 16GB μνήμη RAM DDR4 2133MHz - CL15. Ανάλογα με τους πόρους του συστήματος που χρησιμοποιούσαμε ο χρόνος αυτός μπορεί να μειωνόταν και άλλο και συνεπώς μέτριας ασφάλειας κωδικοί που εισάγουν συνήθως οι χρήστες (έτσι ώστε να τους θυμούνται) δεν είναι απίθανο να έσπαγαν σε μόλις 15 λεπτά ή και λιγότερο. Οπότε ο επόμενος αλγόριθμος που εξετάστηκε ήταν της οικογένειας SHA. Ο SHA1 όπως ερευνήσαμε πάσχει επίσης από παρόμοια προβλήματα, οπότε καταλήξαμε στον αλγόριθμο SHA256.

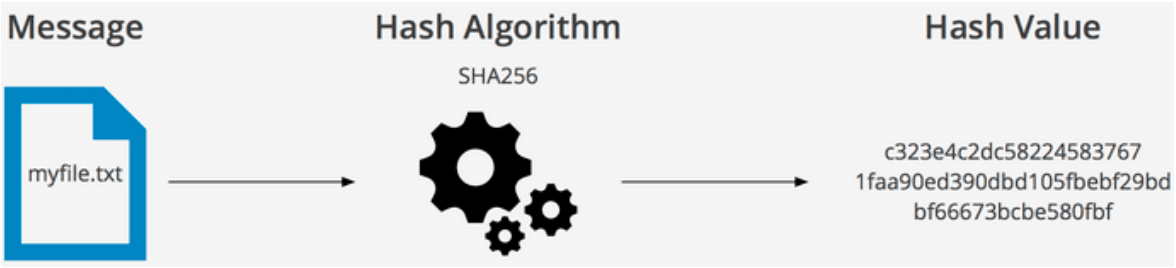
Το όνομα του αλγόριθμου SHA256, είναι ακρωνύμιο των λέξεων Secure Hashing Algorithm, και ο SHA256 είναι ένας δημοφιλής μηχανισμός κατακερματισμού που δημιουργήθηκε από ειδικούς της NSA. Σε αυτό το σημείο να αναφέρουμε ότι ο αλγόριθμος κατακερματισμού SHA αρχικά αναπτύχθηκε από τον οργανισμό NIST

(National Institute of Standards and technology). Η πρώτη έκδοση του αλγορίθμου, γνωστή ως SHA-0 δημοσιεύθηκε ως πρότυπο επεξεργασίας ομοσπονδιακών πληροφοριών (Federal Information Processing Standard - FIPS 180) το 1993. Το 1995 διανεμήθηκε μία αναθεωρημένη έκδοση ως FIPS-180-1, που αναφέρεται γενικά ως sha-1. Το σημερινό πρότυπο καλείται Secure Hash Standard. Ο SHA στηρίζεται στη συνάρτηση κατακερματισμού MD4. Ο SHA-1 καθορίζεται επίσης στο RFC 3174, που ουσιαστικά επαναλαμβάνει την περιγραφή του FIPS 180-1, αλλά προσθέτει μία υλοποίηση σε κώδικα C. (37) (60) (61)

Η βασική λειτουργία του αλγορίθμου είναι η μετατροπή των τυχαίων πληροφοριών σε τιμές με σταθερό μήκος, οι οποίες στο ανθρώπινο μάτι φαίνονται τυχαίες. Επαναλαμβάνουμε ότι πρόκειται για αλγόριθμο δεύτερης γενιάς που δημιουργήθηκε με βάση τον προκάτοχό του SHA-1, ο οποίος με τη σειρά του αναπτύχθηκε το 1995. Μια ενημερωμένη έκδοση του πλέον δημοφιλούς αλγορίθμου δημιουργήθηκε από τον Οργανισμό Εθνικής Ασφάλειας των Η.Π.Α. το 2002. Τρία χρόνια αργότερα, εμφανίστηκε ένα δίπλωμα ευρεσιτεχνίας, επιτρέποντας στον αλγόριθμο να χρησιμοποιηθεί για πολιτικούς σκοπούς. Η τρίτη έκδοση του δημοφιλούς μηχανισμού εμφανίστηκε το 2012, αναπτύχθηκε από ειδικούς του Εθνικού Οργανισμού Τυποποίησης. Με την πάροδο του χρόνου, ο SHA-3 αντικατέστησε πλήρως τους προκατόχους της.

Είναι υπολογιστικά ανέφικτο να αντιστραφεί μία τιμή κερματισμού SHA256. Αξίζει να σημειωθεί ότι όλες οι υπάρχουσες εκδόσεις του αλγορίθμου Algorithm Secure Hashing δημιουργήθηκαν σύμφωνα με την αρχή Merkle-Damgard: οι πληροφορίες χωρίζονται τμήματα και για κάθε τμήμα υπολογίζεται μία μερική συνάρτηση μονής κατεύθυνσης, με αποτέλεσμα το μήκος των δεδομένων να μειώνεται σημαντικά.
Η επιλογή του SHA-256 έγινε διότι ο αλγόριθμος αυτός διαθέτει πολλά πλεονεκτήματα:

- Ο υπολογισμός της συνάρτησης κερματισμού πραγματοποιείται πολύ γρήγορα και συνεπώς ο αλγόριθμος είναι πολύ αποδοτικός.
- Είναι συνάρτηση μίας κατεύθυνσης, οπότε είναι αδύνατο να υπολογιστεί η είσοδος που αντιστοιχεί στο αρχικό όρισμα από την τιμή κερματισμού.
- Η πιθανότητα συγκρούσεων είναι σχεδόν μηδενική.



Εικόνα 41. Επισκόπηση λειτουργίας του αλγορίθμου

Αξίζει να αναφερθεί ότι μπορούν να χρησιμοποιηθούν και άλλοι αλγόριθμοι για την κρυπτογράφηση των ευαίσθητων δεδομένων. Σε αυτή τη κατηγορία είναι γενικά καλύτερο να χρησιμοποιούνται λιγότερο αποδοτικοί αλγόριθμοι, καθώς με την χρήση «αργών» αλγορίθμων, κάνουμε τη διαδικασία των εξαντλητικών επιθέσεων δυσκολότερη για τους επιτιθέμενους, βελτιώνοντας την ασφάλεια. Κάποιοι αλγόριθμοι που πλέον προτιμώνται

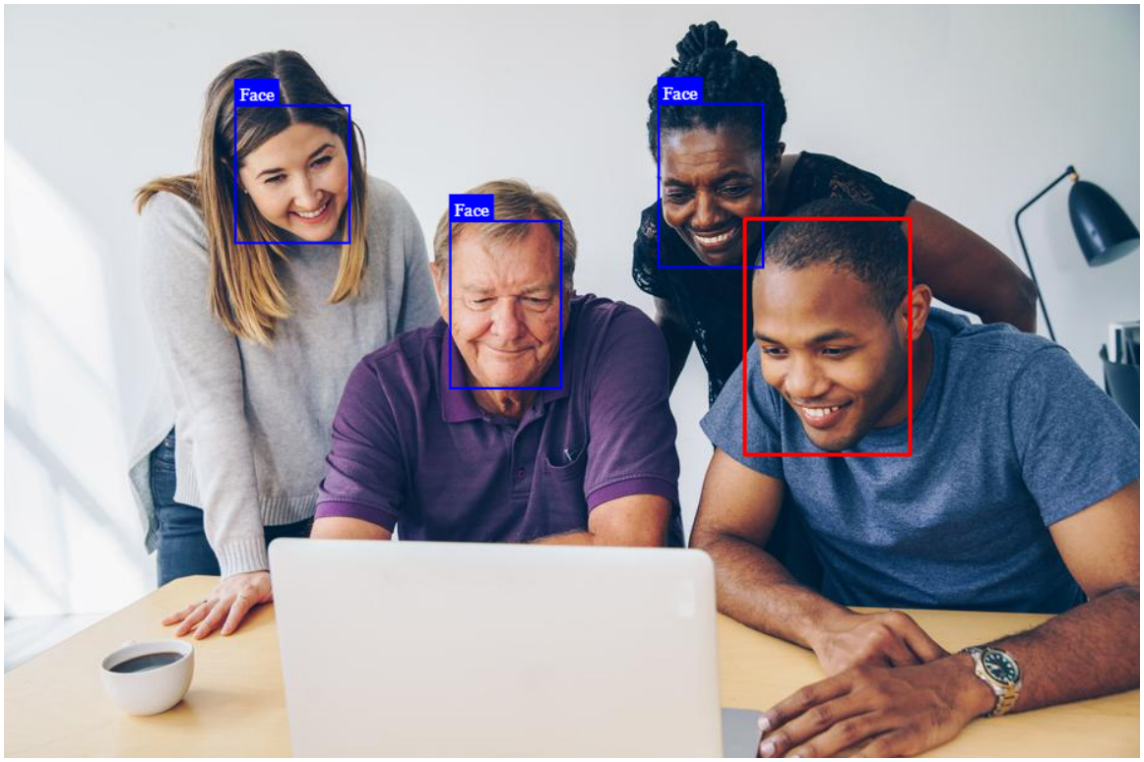
από πολλά λειτουργικά συστήματα και frameworks είναι ενδεικτικά οι αλγόριθμοι bcrypt, scrypt και Argon2. Οι αλγόριθμοι αυτοί έχουν σχεδιαστεί ειδικά για την κρυπτογράφηση κωδικών. Αναλυτικότερα, το γνωστότερο από τα προαναφερθέντα bcrypt, είναι μια συνάρτηση κατακερματισμού σχεδιασμένη για κρυπτογράφηση κωδικών πρόσβασης που σχεδιάστηκε από τους Niels Provos και David Mazières, με βάση την συνάρτηση κατακερματισμού Blowfish και παρουσιάστηκε στο USENIX το 1999 (62). Εκτός από την ενσωμάτωση μίας τυχαίας ποσότητας, δηλαδή ενός salt, για προστασία από τις επιθέσεις ουράνιου τόξου (rainbow table attacks)[1], η συνάρτηση bcrypt προσαρμόζεται με την πάροδο του χρόνου. Δηλαδή, ο αριθμός των επαναλήψεων μπορεί να αυξηθεί για να κάνει τον αλγόριθμο πιο αργό, οπότε παραμένει ανθεκτικός στις επιθέσεις εξαντλητικής αναζήτησης (brute-force) ακόμη και με αυξημένη υπολογιστική ισχύ. Αυτό υλοποιείται εισάγοντας bcrypt την έννοια των γύρων. Σε κάθε γύρο προστίθεται στον κωδικό ένα νέο salt, που κάνει την εύρεση του κωδικού πιο δύσκολη, γεγονός το οποίο κάνει τον αλγόριθμο αρκετά πιο ανθεκτικό σε μελλοντικές εξελίξεις (future proof), καθώς όση υπολογιστική ισχύ και να κατέχει ο επιτιθέμενος, αν οι γύροι είναι αρκετοί, ο χρόνος εύρεσης του κωδικού είναι τόσο μεγάλος που δεν αξίζει ο επιτιθέμενος να σπαταλήσει πόρους (χρονικούς και οικονομικούς) για να αποκαλύψει τον κωδικό. Αυτή η ποσότητα του κόστους μπορεί επίσης να καθοριστεί, έτσι ώστε να υπάρχει στην εφαρμογή μία ισορροπία μεταξύ ταχύτητας και ασφάλειας. Δεν είναι τυχαίο ότι ο αλγόριθμος χρησιμοποιείται με επιτυχία από το 1999. Ο αλγόριθμος bcrypt είναι ο προεπιλεγμένος αλγόριθμος κατακερματισμού κωδικών πρόσβασης για το λειτουργικό σύστημα OpenBSD και άλλα λειτουργικά συστήματα, συμπεριλαμβανομένων ορισμένων διανομών Linux, όπως το SUSE Linux. Υπάρχουν υλοποιήσεις του αλγορίθμου bcrypt για C, C ++, C #, Elixir, Go, Java, JavaScript, Perl, PHP (63), Python, Ruby και άλλες γλώσσες. Η χρήση των συγκεκριμένων αλγορίθμων θα διερευνηθεί στο μέλλον. Στα πλαίσια της διπλωματικής μας εργασίας δεν ήταν απαραίτητη η χρήση τους, αφού η συνάρτηση SHA256 ήταν αρκετά αποτελεσματική ως προς την ασφάλεια, ειδικά αν εφαρμοστεί το κατάλληλο salting. Το συγκεκριμένο ζήτημα θα αναλυθεί εκτενέστερα και ίσως να εφαρμοστεί κάποιος πιο αργός και ασφαλής αλγόριθμος για κρυπτογράφηση ευαίσθητων δεδομένων στη βάση, δεδομένου ότι η αλλαγή του αλγόριθμου κατακερματισμού, μπορεί να γίνει με αρκετά απλό τρόπο στην υλοποίησή μας, οπότε μία ενδεχόμενη αλλαγή δεν θα επηρέαζε τις υπόλοιπες λειτουργίες της εφαρμογής. Ως γενικότερο συμπέρασμα της έρευνας μας, αποτυπώνεται η ανάγκη που υπάρχει για χρήση αργών αλγορίθμων όπως ο bcrypt για την κρυπτογράφηση κωδικών πρόσβασης και ευαίσθητων δεδομένων, κάτι που σύμφωνα με έρευνες δεν ακολουθείται πάντα και γνωστά CMS (π.χ. WordPress) χρησιμοποιούν ακόμη και MD5 εγγενώς (64).

[1] Ένας πίνακας ουράνιου τόξου (rainbow table) είναι ένα ευμέγεθες αποθετήριο δεδομένων που χρησιμοποιείται για να πραγματοποιηθεί επίθεση στη μέθοδο με την οποία κρυπτογραφούνται τα δεδομένα (όχι στον ίδιο τον κωδικό). Πρόκειται για μια τεράστια βιβλιοθήκη κωδικών πρόσβασης και τις τιμές κατακερματισμού που αντιστοιχούν σε κάθε κωδικό από αυτούς. Σε μία επίθεση ουράνιου τόξου, οι επιτιθέμενοι συγκρίνουν το hash του κωδικού ενός χρήστη με όλες τις υπάρχουσες τιμές κατακερματισμού στη βάση δεδομένων. Έτσι μπορεί να αποκαλυφτεί ο κωδικός που αντιστοιχεί σε ένα συγκεκριμένο hash. Επιπλέον, περισσότερα από ένα κείμενα plaintext μπορεί να παράγουν το ίδιο hash, γεγονός που εκμεταλλεύονται οι κυβερνοεγκληματίες, καθώς δεν χρειάζεται να γνωρίζουν τον πραγματικό κωδικό πρόσβασης, αλλά οποιοσδήποτε συνδυασμό συμβόλων που πιστοποιεί την πρόσβασή τους.

6.1.8 Προβλήματα ασφαλείας που προκύπτουν από την λειτουργία αυτόματης δημοσίευσης υλικού από κάμερα IP

Η λειτουργία αυτή δημιουργεί αρκετά προβλήματα ασφάλειας αλλά κυρίως ιδιωτικότητας καθώς αν κάποιος θελήσει να τραβάει φωτογραφίες ανά τακτά χρονικά διαστήματα από μία κάμερα IP η οποία ενδεχομένως να βρίσκεται και σε έναν κοινόχρηστο χώρο, θα λαμβάνει στιγμιότυπα από πρόσωπα από τα οποία δεν έχουμε πάρει την συγκατάθεση τους και ενδεχομένως να μην επιθυμούν να καταγραφούν. Ακόμη και σε ιδιωτικό χώρο, όμως μπορεί να δημιουργηθούν θέματα ιδιωτικότητας καθώς μία κάμερα μπορεί να παίρνει τα πρόσωπα επισκεπτών ή το πρόσωπο του ιδιοκτήτη σε περιστάσεις που δεν είναι κατάλληλες για αναμετάδοση (από αστείες γκριμάτσες μέχρι οτιδήποτε άλλο), ή μπορεί να πάρει κάποιους αριθμούς και προσωπικά του στοιχεία τα οποία να μην επιθυμεί και να μην πρέπει να δημοσιευθούν, όπως αναλύσαμε στην ενότητα 5. Έτσι προκύπτουν προβλήματα και η πράξη γενικότερα αντιβαίνει στους κανόνες που διέπονται από το νομοθετικό πλαίσιο GDPR, το οποίο έχουμε αναφέρει σε προηγούμενη ενότητα. (1) Δεδομένων αυτών των προβλημάτων, αποφασίσθηκε να υλοποιηθεί η λειτουργία αυτόματης αναγνώρισης και συσκότισης προσώπων, η οποία περιγράφεται στην ενότητα 6.1.4., χρησιμοποιώντας δηλαδή μία βιβλιοθήκη, η οποία θα επιτρέπει την αναγνώριση περιοχών της φωτογραφίας που είναι πρόσωπα ενώ ύστερα αυτά θολώνονται με κατάλληλο κώδικα.

Έτσι μπορεί να γίνεται αυτόματη συσκότιση προσώπων, όταν ανεβάζουμε φωτογραφίες και κατ' επέκταση και άλλων αντικειμένων τα οποία ενδεχομένως θα κριθεί ότι παραβιάζουν την ιδιωτικότητα των χρηστών. Βέβαια η λειτουργία έχει κάποια πιθανότητα σφάλματος. Με άλλα λόγια, αν και ο αλγόριθμος ήταν αρκετά αποτελεσματικός και έχει μικρές πιθανότητες λάθους, υπήρξε μία περίπτωση που δεν κατάφερε να ανιχνεύσει αν όλα τα πρόσωπα ήταν άνθρωποι, ίσως λόγω κλίσης, μη επαρκούς φωτισμού και άλλων συνθηκών.



Συνεπώς στο ανέβασμα πολλαπλών φωτογραφιών ή και βίντεο από μία IP Camera μπορεί και πάλι να δημιουργηθούν προβλήματα. Η απλή λύση είναι να μειωθεί λίγο το κατώφλι εμπιστοσύνης (confidence) του αλγορίθμου έτσι ώστε να συσκοτίζονται και πρόσωπα για τα οποία ο αλγόριθμος είναι σίγουρος σε ποσοστό μικρότερο του 50%, ενδεικτικά στην περιοχή 30-40%. Βέβαια σε αυτή την περίπτωση μπορεί να δημιουργηθούν αλλά πρακτικά προβλήματα καθώς θα θολώνονται ενδεχομένως και αντικείμενα τα οποία δεν πρέπει να θολωθούν.

7 Επίλογος

Στις μέρες μας, η χρήση κοινωνικών δικτύων είναι σχεδόν καθολική, καθώς θεωρούνται όχι μόνο χρήσιμα, αλλά και απαραίτητα στοιχεία για πλήθος δραστηριοτήτων. Ένας μεγάλος αριθμός ανθρώπων καθημερινά χρησιμοποιεί τα κοινωνικά δίκτυα τα οποία έχουν ως στόχο τη μετάδοση πληροφοριών με απλούστερο και γρηγορότερο τρόπο. Ταυτόχρονα, ο άνθρωπος από τη φύση του είναι κοινωνικό ον με αποτέλεσμα η χρήση των κοινωνικών δικτύων να είναι ιδιαίτερα ελκυστική, καθώς μέσω αυτής επιτυγχάνεται κοινωνική επαφή και ανταλλαγή απόψεων. Τα παιδιά ιδιαίτερα είναι επιρρεπή στο να εθιστούν σε διάφορες πλατφόρμες κοινωνικών δικτύων, καθώς έχουν μεγαλώσει σε περιβάλλον με ευρεία χρήση κοινωνικών δικτύων και αγνοούν σε πολλές περιπτώσεις τους κινδύνους που μπορεί να επιφέρει η αλόγιστη χρήση τους ή δεν είναι σε θέση να αναπτύξουν αποτελεσματικές άμυνες. Συνακόλουθα, η ασφάλεια στα κοινωνικά δίκτυα είναι ένα πολύ σημαντικό θέμα στις μέρες μας και θα πρέπει να μελετηθεί εκτενέστερα.

Στην παρούσα εργασία αναπτύχθηκε μία ολοκληρωμένη διαδραστική και responsive εφαρμογή κοινωνικής δικτύωσης, η οποία περιλαμβάνει όλες τις απαραίτητες λειτουργίες που χρειάζεται ένας χρήστης σε ένα σύγχρονο κοινωνικό δίκτυο και να διαθέτει αρκετές βαθμίδες ασφαλείας.

Κατά τη διάρκεια εκπόνησης της παρούσας εργασίας, οι δυσκολίες που κληθήκαμε να αντιμετωπίσουμε ήταν πολλές. Ας αναφερθούμε συνοπτικά σε κάποιες από αυτές. Ως προς την υλοποίηση, η πρώτη δυσκολία που αντιμετωπίσαμε ήταν ο σχεδιασμός της ιστοσελίδας καθώς έπρεπε η ιστοσελίδα μας να είναι διαδραστική και responsive και η διεπαφή να ανταποκρίνεται στα πρότυπα ενός σύγχρονου κοινωνικού δικτύου. Κατόπιν αντιμετωπίσαμε αρκετές δυσκολίες με τους μηχανισμούς των Like/Unlike και των Comments. Χρειαζόταν κατάλληλη αρχιτεκτονική και σχεδιασμός στη βάση δεδομένων και αρκετή αποσφαλμάτωση (debugging) στο back end, έτσι ώστε να λειτουργήσουν όπως σε ένα σύγχρονο κοινωνικό δίκτυο. Παρόμοιες δυσκολίες αντιμετωπίσαμε και σε άλλους μηχανισμούς όπως η δημιουργία φίλων, η απάντηση σε comments και η συνομιλία με άλλους χρήστες. Μία από τις κυριότερες δυσκολίες που αντιμετωπίσαμε ήταν το πως θα κάνουμε τη σελίδα μας διαδραστική, έτσι ώστε όλα να γίνονται σε πραγματικό χρόνο, όπως πρέπει σε ένα σύγχρονο κοινωνικό δίκτυο (π.χ. για να γίνεται διαδραστικά το unlike και να αλλάζουν τα στοιχεία χωρίς ανανέωση της σελίδας). Αυτό τελικά καταφέραμε και το λύσαμε με τη χρήση Ajax και JQuery ή πιο σπάνια με χρήση απλής JavaScript χωρίς επιπλέον βιβλιοθήκη, ανάλογα την περίπτωση. Ως προς το κομμάτι της ασφάλειας χρειάστηκε να μελετήσουμε αρκετούς τύπους ευπαθειών, έτσι ώστε να βρούμε ενδεχόμενες ευπάθειες στην διαδικτυακή μας εφαρμογή, ενώ για να τις λύσουμε χρειάστηκε να υλοποιήσουμε πολυάριθμες τακτικές, κάποιες από τις οποίες περιγράφηκαν σε αυτή την διπλωματική εργασία. Εν τέλει, ως προς το κομμάτι της υλοποίησης κατάλληλης λειτουργίας για αυτόματη αναγνώριση και συσκότιση προσώπων, χρειάστηκε να μελετήσουμε εκτενώς αλγόριθμους και μοντέλα νευρωνικών δικτύων και να ερευνήσουμε για ενδεχόμενες βιβλιοθήκες οι οποίες πραγματεύονται το συγκεκριμένο ζήτημα και ύστερα να συγγράψουμε κατάλληλο κώδικα για την ενοποίηση της βιβλιοθήκης με την εφαρμογή μας.

7.1 Μελλοντικές επεκτάσεις

Η εφαρμογή μας είναι μια εφαρμογή διαδικτυακού λογισμικού γραμμένη κυρίως σε κώδικα PHP, ενώ γίνεται χρήση της βάσης δεδομένων MySQL για την αποθήκευση των περισσότερων από τα δεδομένα μας. Χρησιμοποιήσαμε βέβαια αρκετά και την JavaScript στο front end κομμάτι καθώς και τα εργαλεία Ajax και JQuery. Η παρούσα διπλωματική εργασία θεωρούμε ότι αποτελεί ένα ολοκληρωμένο κοινωνικό δίκτυο με αρκετά επίπεδα ασφάλειας. Οι τεχνικές και οι τεχνολογίες άλλωστε που χρησιμοποιήθηκαν δημιουργήθηκαν μέσα στα χρονικά πλαίσια εκπόνησης μιας μεταπτυχιακής εργασίας και έγινε σημαντική προσπάθεια τις περισσότερες λειτουργίες να τις θεωρήσουμε σε χαμηλότερο επίπεδο, εις βάθος και να τις υλοποιήσουμε μόνοι μας σχεδόν από το μηδέν, έτσι ώστε να καταλάβουμε την πραγματική πολυπλοκότητα ενός τέτοιου έργου και να αποκομίσουμε γνώσεις στα θέματα του προγραμματισμού και της ασφάλειας. Χρησιμοποιήθηκαν βέβαια και βιβλιοθήκες ελεύθερου λογισμικού κατά σημεία (π.χ. στην εισαγωγή emoticons, ή για συγκεκριμένες δυνατότητες επεξεργασίας εικόνας, ή στο κομμάτι της ασφάλειας), καθώς σε τέτοιου είδους προβλήματα οι έτοιμες λύσεις μπορούν να εξοικονομήσουν χρόνο, χωρίς να μειώσουν την ποιότητα του λογισμικού μας.

Για την εφαρμογή υπάρχουν μελλοντικά περιθώρια βελτίωσης. Μερικά από αυτά είναι:

- Βελτιώσεις στη διεπαφή χρήστη και polishing, καθώς η διεπαφή ειδικά σε ένα τέτοιο έργο έχει μεγάλη αξία.
- Βελτιώσεις στην λειτουργικότητα επιμέρους λειτουργιών όπως π.χ. στο chatting, όπου λόγω της υλοποίησης σε PHP, έχει τεθεί ένας χρόνος ανανέωσης των μηνυμάτων (λίγων δευτερολέπτων) τα οποία ανακτώνται από την βάση με JQuery και Ajax. Θα μπορούσαν οι συνομιλίες στο μέλλον να γίνονται σε πραγματικό χρόνο. Θα μπορούσε για παράδειγμα να χρησιμοποιηθεί το πρωτόκολλο XMPP (Extensible Messaging and Presence Protocol), το οποίο χρησιμοποιείται και απο εφαρμογές όπως Whatsup και Zoom (65). Επίσης θα μπορούσε να προστεθεί και η επισύναψη αρχείων και φωτογραφιών. Σε άλλες λειτουργικότητες επίσης όπως το ανέβασμα νέας δημοσίευσης, θα ήταν επιθυμητή η ύπαρξη φίλτρων για τις φωτογραφίες που ανεβάζει ο χρήστης καθώς πολλοί χρήστες έχουν συνηθίσει αυτή τη λειτουργικότητα σε κοινωνικά δίκτυα όπως το Instagram. Ο διαχωρισμός των προφίλ σε δημόσια και ιδιωτικά θα ήταν επίσης μία θεμιτή λειτουργία.
- Εύρεση και διόρθωση bugs που είτε ενδεχομένως υπάρχουν τώρα, είτε θα προκύψουν στο μέλλον από τη χρήση του κοινωνικού δικτύου από πλήθος χρηστών, καθώς η εκτενής του χρήση θα οδηγήσει στην αποκάλυψη των τυχόντων bugs.
- Υλοποίηση περισσότερων και πιο καινοτόμων συστημάτων ασφαλείας κυρίως γύρω από IoT συσκευές και την ειδική διαχείριση τους.

8 Βιβλιογραφία

1. Νομοθεσία και Κανονιστικό πλαίσιο General Data Protection Regulation (GDPR). [Ηλεκτρονικό] 23 May 2018. [Παραπομπή: 29 November 2020.] https://gdpr-info.eu/.

2. **Barnes, J. A.** Human Relations, 7: 39-58. «*Class and Committees in a Norwegian Island Parish*». 1954.

3. «*Social Support Networks and the crisis of Bereavement*». **Walker, K., Macbride, A., Vachnon, M.** Community Resources Service Clarke Institute of Psychiatry Toronto Ontario Canada, 11(1): 35-41. : s.n., 1967.

4. **Χτούρης, Σ.** *Ορθολογικά Συμβολικά Δίκτυα.* Αθήνα : Εκδόσεις Νήσος, (2004).

5. **Amichai-Hamburger, Yair.** «Personality and the Internet». *The social net: Human behavior in cyberspace, 35: 27-55.* 2005.

6. **Amichai-Hamburger, Y., & Barak, A.** Technology and Well-being, 10: 34-76. «*Internet and well-being*». 2009.

7. **Michael Hauben, Ronda Hauben.** Netizens: On the History and Impact of Usenet and the Internet. [Ηλεκτρονικό] [Παραπομπή: 06 July 1998.] https://journals.uic.edu/ojs/index.php/fm/article/view/605.

8. **Britannica, The Editors of Encyclopaedia.** USENET. *Britannica.* [Ηλεκτρονικό] [Παραπομπή: 1 December 2020.] https://www.britannica.com/technology/USENET.

9. **O'Reilly, Tim και Todino, Grace.** *Managing UUCP and Usenet.* s.l. : O'Reilly & Associates, Inc., 1992.

10. ArpaNet. [Ηλεκτρονικό] [Παραπομπή: 29 November 2020.] http://pacific.jour.auth.gr/internet/page%201.2.htm.

11. *The co-evolution of two Chinese mobile short video apps: Parallel platformization of Douyin and TikTok.* **D. Bondy Valdovinos Kaye, Xu Chen, Jing Zeng.** 2020.

12. *Second Life: a Social Network of Humans and Bots.* **Matteo Varvello, Geoffrey M. Voelker.** Pages 9-14, s.l. : NOSSDAV '10, 2010.

13. **Λουκέρη, Νίκη.** Παρουσίαση και ανάλυση των μέσων κοινωνικής δικτύωσης. [Ηλεκτρονικό] http://telematics.upatras.gr/telematics/system/files/bouras_site/ergasies_foithtwn/%CE%A0%CE%B1%CF%81%CE%BF%CF%85%CF%83%CE%AF%CE%B1%CF%83%CE%B7%20%26%20%CE%91%CE%BD%CE%AC%CE%BB%CF%85%CF%83%CE%B7%20%CF%84%CF%89%CE%BD%20%CE%9C%CE%AD%CF%83%CF%89%CE%BD%20%CE%.

14. *Smart Age Detection for Social Media Using Deep Learning Techniques via Ear Shape.* **Manal Alghieth, Jawaher Alhuthail, Kholod Aldhubiay, Rotan Alshowaye.** Qassim, Saudi Arabi : Information Technology, Qassim University, 2019, (IJACSA) International Journal of Advanced Computer Science and Applications, Τόμ. 10.

15. *Ear Recognition: More Than a Survey.* **Žiga Emeršič, Vitomir Štruc, Peter Peer.** Ljubljana, Slovenia : s.n., 18 November 2016.

16. **apachefriends.** [Ηλεκτρονικό] [Παραπομπή: 30 11 2020.] https://www.apachefriends.org/docs/.

17. phpmyadmin. [Ηλεκτρονικό] [Παραπομπή: 25 10 2020.] https://www.phpmyadmin.net/docs/.

18. Git Branches. *Atlassian.* [Ηλεκτρονικό] [Παραπομπή: 29 November 2020.] https://www.atlassian.com/git/tutorials/using-branches.

19. Git Merges. *Atlassian.* [Ηλεκτρονικό] [Παραπομπή: 29 November 2020.] https://www.atlassian.com/git/tutorials/using-branches/git-merge.

20. github. [Ηλεκτρονικό] 20 10 2020. https://docs.github.com/en.

21. sublimetext. [Ηλεκτρονικό] [Παραπομπή: 20 10 2020.] https://www.sublimetext.com/docs/3/.

22. mysql. [Ηλεκτρονικό] [Παραπομπή: 20 10 2020.] https://dev.mysql.com/doc/.

23. html. [Ηλεκτρονικό] [Παραπομπή: 20 10 2020.] https://devdocs.io/html/.

24. css. [Ηλεκτρονικό] [Παραπομπή: 20 10 2020.] https://devdocs.io/css/.

25. JavaScript. [Ηλεκτρονικό] [Παραπομπή: 20 10 2020.] https://developer.mozilla.org/en-US/docs/Web/JavaScript.

26. php. [Ηλεκτρονικό] [Παραπομπή: 20 10 2020.] https://www.php.net/docs.php.

27. ajax. [Ηλεκτρονικό] [Παραπομπή: 19 10 2020.] https://api.jquery.com/category/ajax/.

28. jquery.com. [Ηλεκτρονικό] [Παραπομπή: 29 November 2020.] https://jquery.com/.

29. bootstrap. [Ηλεκτρονικό] [Παραπομπή: 20 10 2020.] https://getbootstrap.com/docs/4.5/getting-started/introduction/.

30. PortSwigger. [Ηλεκτρονικό] [Παραπομπή: 1 12 2020.] https://portswigger.net/web-security/cross-site-scripting.

31. Website Defacement Attack. *Imperva.* [Ηλεκτρονικό] [Παραπομπή: 27 December 2020.] https://www.imperva.com/learn/application-security/website-defacement-attack/.

32. Prevent Web Attacks Using Input Sanitization. *esecurityplanet.com.* [Ηλεκτρονικό] [Παραπομπή: 28 December 2020.] https://www.esecurityplanet.com/endpoint/prevent-web-attacks-using-input-sanitization/#:~:text=Input%20sanitization%20describes%20cleansing%20and,a%20false%20sense%20of%20security..

33. SQL Injection Prevention Cheat Sheet. *owasp.org.* [Ηλεκτρονικό] [Παραπομπή: 5 December 2020.] https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html.

34. realpath function. *php.* [Ηλεκτρονικό] 06 December 2020. https://www.php.net/manual/en/function.realpath.php.

35. Description of core php.ini directives. *php.net.* [Ηλεκτρονικό] [Παραπομπή: 28 December 2020.] https://www.php.net/manual/en/ini.core.php.

36. open_basedir. *mediatemple.net.* [Ηλεκτρονικό] [Παραπομπή: 27 December 2020.] https://mediatemple.net/community/products/dv/204404214/how-do-i-set-the-path-for-open_basedir.

37. **Stallings, William.** *Κρυπτογραφία και Ασφάλεια Δικτύων, Αρχές και Εφαρμογές.* Πρώτη Ελληνική έκδοση.

38. **James F. Kurose, Keith W. Ross.** *Computer Networking A Top-Down Approach.* Sixth Edition. s.l. : Pearson.

39. SYN-Flooding. *techtarget.* [Ηλεκτρονικό] [Παραπομπή: 27 December 2020.] https://searchsecurity.techtarget.com/definition/SYN-flooding.

40. eSecurity Planet. [Ηλεκτρονικό] 06 12 2020. https://www.esecurityplanet.com/networks/how-to-prevent-ddos-attacks-tips-to-keep-your-website-safe/.

41. *Analysis of Mirai Malicious Software.* **Hamdija Sinanovic, Sasa Mrdovic.** Sarajevo, Bosnia and Herzegovina : University of Sarajevo.

42. *The Day After Mirai: A Survey on MQTT Security Solutions After the Largest Cyber-attack Carried Out through an Army of IoT Devices.* **Giovanni Perrone, Massimo Vecchio, Riccardo Pecori, Raffaele Giaffreda.** 2017.

43. KrebsOnSecurity Hit With Record DDoS. *KrebsOnSecurity.* [Ηλεκτρονικό] 16 September 2016. [Παραπομπή: 2 January 2021.] https://krebsonsecurity.com/2016/09/krebsonsecurity-hit-with-record-ddos/.

44. *IoT Security Vulnerability: A Case Study of a Web Camera.* **Yogeesh Seralathan, Tae (Tom) Oh, Suyash Jadhav, Jonathan Myers, Jaehoon (Paul) Jeong, Young Hoand and Jeong Neyo Kim.** s.l. : International Conference on Advanced Communications Technology(ICACT), 11 ~ 14 February 2018.

45. Vulnerabilities in these IoT cameras could give attackers full control, warn researchers. *zdnet.com.* [Ηλεκτρονικό] https://www.zdnet.com/article/vulnerabilities-in-these-iot-cameras-could-give-attackers-full-control-warn-researchers/.

46. An end-to-end open source machine learning platform. *TensorFlow.* [Ηλεκτρονικό] [Παραπομπή: 02 January 2021.] https://www.tensorflow.org/.

47. **Corporation, Architecture Technology.** *Netbios Report and Reference.* 2nd Edition. s.l. : Elsevier Ltd., 1991.

48. **Aitchison, Ron.** *Pro DNS and BIND.* s.l. : Apress Media LLC, 2005.

49. **ΛΑΟΥΡΑ, ΘΕΟΔΟΣΗ - ΚΟΚΚΙΝΟΥ.** Τεχνητά Νευρωνικά Δίκτυα και εφαρμογές στα Συστήματα Αυτόματου Ελέγχου. [Ηλεκτρονικό] [Παραπομπή: 28 December 2020.] https://nemertes.lis.upatras.gr/jspui/bitstream/10889/6401/1/%CE%B4%CE%B9%CF%80%CE%BB%CF%89%CE%BC%CE%B1%CF%84%CE%B9%CE%BA%CE%B7.pdf.

50. *Υλοποίηση βιβλιοθήκης στο MATLAB για την ταχεία έρευνα και προτυποποίηση νέων μεθόδων εκπαίδευσης Βαθέων Συνελικτικών Νευρωνικών Δικτύων με χρήση του CAFFE.* **Πρόβος, Αλέξης.** s.l. : Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, 2015.

51. *Αναγνώριση Προσώπου με χρήση Συνελικτικών Νευρωνικών Δικτύων.* **Κανελλόπουλος, Στέφανος Π.** Αθήνα : ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ, 2019.

52. The MIT License. *opensource.org.* [Ηλεκτρονικό] [Παραπομπή: 28 December 2020.] https://opensource.org/licenses/MIT.

53. *SSD: Single Shot MultiBox Detector.* **Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg.** 8 December 2015, Τόμ. last revised 29 December 2016 (v5).

54. *Tinier-YOLO: A Real-Time Object Detection.* **WEI FANG, LIN WANG, PEIMING REN.** Wuxi, China : s.n., 2019.

55. tensorflow Keras YOLOv4/v3/v2 Modelset. *github.* [Ηλεκτρονικό] [Παραπομπή: 03 January 2021.] https://github.com/david8862/keras-YOLOv3-model-set.

56. *Fixed Point Implementation of Tiny-Yolo-v2 using OpenCL on FPGA.* **Yap June Wai, Zulkalnain bin Mohd Yussof, Sani Irwan bin Salim, Lim Kim Chuan.** Melaka, Malaysia : s.n., 2018, Τόμ. 9.

57. *Joint Face Detection and Alignment using Convolutional Networks.* **Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, Senior Member, IEEE, and Yu Qiao, Senior Member, IEEE.** https://kpzhang93.github.io/MTCNN_face_detection_alignment/paper/spl.pdf.

58. Realtime JavaScript Face Tracking and Face Recognition using face-api.js' MTCNN Face Detector. *https://itnext.io/.* [Ηλεκτρονικό] [Παραπομπή: 29 December 2020.] https://itnext.io/realtime-javascript-face-tracking-and-face-recognition-using-face-api-js-mtcnn-face-detector-d924dd8b5740.

59. W3C. [Ηλεκτρονικό] 10 12 2020. https://www.w3.org/Protocols/HTTP/AsImplemented.html.

60. **U.S. Department of Commerce, National Institute of Standards and Technology.** [Ηλεκτρονικό] August 2015. [Παραπομπή: 29 October 2020.] https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf.

61. **Joux, Florent Chabaud and Antoine, Centre d'l~lectronique de l'Armement.** Differential Collisions in SHA-0. [Ηλεκτρονικό] [Παραπομπή: 25 October 2020.] https://link.springer.com/chapter/10.1007/BFb0055720.

62. *A Future-Adaptable Password Scheme.* **Niels Provos, David Mazieres.** Monterey, California, USA : s.n., 6-11 June 1999, Proceedings of the FREENIX Track.

63. https://www.php.net. *password_hash.* [Ηλεκτρονικό] [Παραπομπή: 22 January 2021.]

64. *Evaluation of password hashing schemes in open source web platforms.* **Christoforos Ntantogian, Stefanos Malliaros, Christos Xenakis.** Department of Digital Systems, University of Piraeus, Piraeus, Greece : Elsevier, 2019.

65. XMPP - Extensible Messaging and Presence Protocol. *xmpp.org.* [Ηλεκτρονικό] [Παραπομπή: 28 December 2020.] https://xmpp.org/.

66. **Amichai-Hamburger Yair.** Research Gate. [Ηλεκτρονικό] January 2002. https://www.researchgate.net/publication/223879534_Internet_and_personality.

67. **University, George Mason.** Social Engineering: Hacking the Wetware. Article in Information Security Journal. A GlobaScott Applegate [Online]. [Ηλεκτρονικό] https://www.researchgate.net/publication/220449972_Social_Engineering_Hacking_the_Wetware.

68. *Prevailing and emerging cyber threats and security practices in IoT-Enabled smart grids: A survey.* **Abhishek Gupta, Alagan Anpalagan, Glaucio H.S. Carvalho, Ling Guan, Isaac Woungang.** Ryerson University, Toronto, Canada : s.n., 5 September 2020, Journal of Network and Computer Applications, p. 31.

69. "addslashes - PHP Manual". *php.net.* [Ηλεκτρονικό] [Παραπομπή: 5 December 2020.] https://www.php.net/manual/en/function.addslashes.