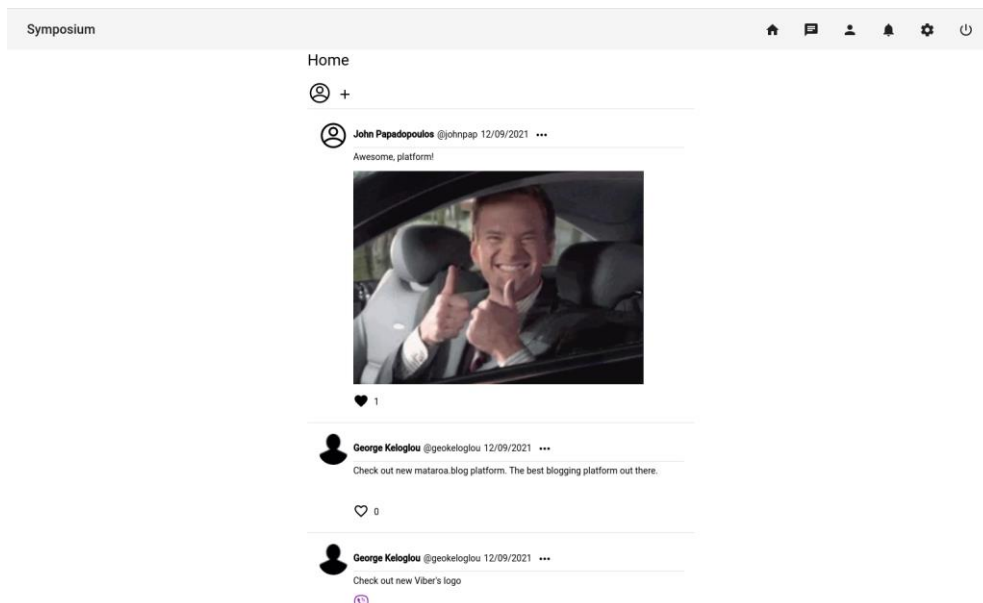


ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ
ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Τίτλος:

«Ανάπτυξη εφαρμογής κοινωνικής δικτύωσης: *Symposium*»



Φοιτητής:
Κέλογλου Γεώργιος
Αρ. μητρώου: 154466

Επιβλέπων:
κ. Σιδηρόπουλος Αντώνιος
Καθηγητής

Θεσσαλονίκη, 2021

Τίτλος Δ.Ε.: «Ανάπτυξη εφαρμογής κοινωνικής δικτύωσης: Symposium»

Κωδικός Δ.Ε.: 20192

Όνοματεπώνυμο φοιτητή: Κέλογλου Γεώργιος

Όνοματεπώνυμο εισηγητή: Σιδηρόπουλος Αντώνιος

Ημερομηνία ανάληψης: Δ.Ε. 02-11-2020

Ημερομηνία περάτωσης: Δ.Ε. 15-09-2021

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία εκπονήθηκε και αποτελεί πνευματική ιδιοκτησία του φοιτητή Κέλογλου Γεώργιου. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

Πρόλογος

Ο τομέας της πληροφορικής που μου έχει προκαλέσει ιδιαίτερο ενδιαφέρον κατά την διάρκεια της φοίτησης και μετέπειτα της επαγγελματικής σταδιοδρομίας στο Web development αποτέλεσαν έναυσμα για την ενασχόλησή μου με το συγκεκριμένο θέμα της πτυχιακής διατριβής το οποίο είναι η «Ανάπτυξη εφαρμογής κοινωνικής δικτύωσης: Symposium» στο Web.

Στην παρούσα διατριβή η οποία χωρίζεται σε πέντε κεφάλαια- ασχολήθηκα με την υλοποίηση ενός συστήματος κοινωνικής δικτύωσης. Στο πρώτο κεφάλαιο γίνεται μία ανάλυση για την εφαρμογή και τα πλεονεκτήματα υλοποίησής της στο Web. Στο δεύτερο κεφάλαιο, αναφέρεται το Διαδίκτυο και ο Παγκόσμιος Ιστός καθώς και τα επιμέρους τμήματα των τεχνολογιών Διαδικτύου.

Έπειτα, στο επόμενο κεφάλαιο αναλύονται εκτενώς όλες οι τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής. Στη συνέχεια, ακολουθεί το σημαντικότερο κεφάλαιο της εργασίας στο οποίο πραγματώνεται η εφαρμογή και επεξηγείται και δίνονται οδηγίες για την χρήση της. Το τελευταίο κεφάλαιο αποτελείται από τα συμπεράσματα της εργασίας και τις προτάσεις για βελτίωση της εφαρμογής.

Σε αυτό το σημείο, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή, της παρούσας διατριβής, κ. Σιδηρόπουλο Αντώνιο καθώς και την οικογένεια μου, οι οποίοι με στηρίζουν όλα αυτά τα χρόνια.

Περίληψη

Τα τελευταία χρόνια μπορεί κανείς να παρατηρήσει ότι με την πάροδο του χρόνου όλοι οι άνθρωποι, μικροί και μεγάλοι, χρησιμοποιούν στην εποχή μας τα κοινωνικά δίκτυα. Άλλοι σε καθημερινή βάση, άλλοι σε μικρότερη συχνότητα, η ουσία όμως είναι ότι τα χρειαζόμαστε. Ο λόγος είναι προφανής και ονομάζεται επικοινωνία. Συνεπώς, με αυτήν την σκέψη και πρόφαση αποφάσισα στην πτυχιακή εργασία να υλοποιήσω μία αντίστοιχη πλατφόρμα. Προφανώς, όχι του ίδιου βεληνεκούς, όπως των γνωστών εταιρειών, αλλά μία μινιμαλιστική προσέγγιση αυτών.

Μόλις αρχίσει κάποιος να σκέφτεται πως θα σχεδιάσει, θα αναπτύξει και θα ανεβάσει στο Διαδίκτυο μία τέτοια διαδικτυακή εφαρμογή πρέπει να αναλογιστεί αρχικά ότι θα βασιστεί σε τεχνολογίες που υπάρχουν και είναι καλό να υπάρχουν για τα επόμενα χρόνια. Επομένως, η απόφαση πρέπει να είναι με γνώμονα το μέλλον και όχι την τρέχουσα κατάσταση της μόδας.

Η υλοποίηση μιας τέτοιας πλατφόρμας δεν είναι απλή. Αρχικά, ο όγκος των δεδομένων των χρηστών είναι τεράστιος. Σκεφτείτε πόσες φωτογραφίες καθημερινά ανεβάζει ο κόσμος, πόσα μηνύματα ανταλλάσει, πόσα likes (μου αρέσει) πατάει. Πέρα από τον όγκο των δεδομένων, ο σχεδιασμός της εφαρμογής θα πρέπει να είναι απλός, κατανοητός και φυσικά όσον αφορά την λειτουργικότητα απροβλημάτιστος.

Συνεπώς, στην παρούσα εργασία, παραθέτω την δική μου προσπάθεια και οπτική υλοποίησης μιας αντίστοιχης εφαρμογής κοινωνικής δικτύωσης στο Web, αποδεχόμενος την πρόκληση και αναλύοντας φυσικά όλες τις τεχνολογίες που χρησιμοποιήθηκαν με μία εισαγωγή σε αυτές που βασίστηκαν.

Development of a social networking application: «Symposium»

Georgios Keloglou

Abstract

In recent years one can observe that over time all people, young and old, use social networks in our time. Some on a daily basis, others on a lower frequency, but the bottom line is that we need them. The reason is obvious and is called communication. Therefore, with this thought and pretext, I decided in the dissertation to implement a corresponding platform. Obviously, not of the same scope as the well-known companies, but a minimalist approach to them.

Once one starts thinking about how to design, develop and upload such an online application on the Internet, one must first consider that it will be based on technologies that exist and are good to have for years to come. Therefore, the decision should be guided by the future and not the current state of fashion.

The implementation of such a platform is not simple. Initially, the volume of user data is huge. Think about how many photos people upload daily, how many messages they exchange, how many likes they click. Beyond the volume of data, the design of the application should be simple, understandable and of course in terms of functionality trouble-free.

Therefore, in the present work, I present my own effort and perspective of implementing a corresponding social networking application on the Web, accepting the challenge and of course analyzing all the technologies used with an introduction to the ones based.

Περιεχόμενα

Πρόλογος.....	iv
Περίληψη	v
Abstract.....	vi
Περιεχόμενα	vii
Κεφάλαιο 1ο: Εισαγωγή.....	1
1.1 Η εφαρμογή και το πρόβλημα.....	1
1.2 Γιατί Web εφαρμογή;	1
Πλεονεκτήματα	1
1.2.2 Μειονεκτήματα	2
Κεφάλαιο 2ο: Διαδίκτυο - Web	5
2.1 Ο παγκόσμιος ιστός.....	5
2.2 Το Διαδίκτυο.....	6
2.2.1 Ορισμός	6
2.2.2 Το Διαδίκτυο και η επικοινωνία	6
2.2.3 Η τεχνολογία του Διαδικτύου	6
2.2.4 Η ιστορία του Διαδικτύου	7
2.2.5 Οι πληροφορίες στο Διαδίκτυο	8
2.2.6 Νομικά και ηθικά ζητήματα	9
2.2.7 Πρόσβαση στο Διαδίκτυο	9
2.2.8 Διαδίκτυο και Ευρωπαϊκή Ένωση	10
2.2.9 Κίνδυνοι του Διαδικτύου	11
2.3 Διαδικτυακά πρωτόκολλα και υπηρεσίες	13
2.4 Υπηρεσίες Διαδικτύου.....	14
2.5 Ιστοσελίδα (web page)	15
2.6 Κατηγορίες ιστοσελίδων	15
2.6.1 Στατική ιστοσελίδα	15
2.6.2 Δυναμική ιστοσελίδα	15
2.6.3 Διαφορές στατικής και δυναμικής ιστοσελίδας	16
2.7 Διακομιστής Ιστού (Web Server).....	16
2.8 Φυλλομετρητής (web browser).....	17
2.9 Όνομα τομέα / περιοχής (domain name)	18
2.10 Φιλοξενία Ιστοσελίδων (web hosting)	18

2.11 JSON (JavaScript Object Notation).....	19
2.11.1 Τι είναι το πρωτόκολλο HTTP	20
2.11.2 Τι είναι το πρωτόκολλο HTTPS	21
2.11.3 Δομή ενός HTTP μηνύματος απάντησης	22
2.11.4 Δομή ενός HTTP μηνύματος αίτησης	23
2.11.5 URIs (Uniform Resource Identifiers)	24
2.11.6 SOAP (Simple Object Access Protocol)	24
2.11.7 REST (Representational State Transfer)	24
2.11.8 REST VS SOAP	25
Κεφάλαιο 3ο: Τεχνολογίες, frameworks και libraries	27
3.1 Τεχνολογίες front-end	27
3.1.1 HTML / HTML5	27
3.1.2 CSS	28
3.1.3 JavaScript	28
3.1.4 TypeScript	29
3.1.5 NPM (Node Package Manager)	29
3.1.5 Angular (Web framework)	30
3.1.6 Material Design / Angular Material	30
3.1.7 RxJS	30
3.1.8 JSON Web Token	31
3.2 Τεχνολογίες back-end.....	34
3.2.1 C#	34
3.2.2 .NET Core	34
3.2.3 Entity Framework Core	34
3.2.4 Dapper	35
3.2.5 SQL / PostgreSQL	35
3.2.6 SignalR	36
3.2.7 Docker	38
3.2.8 Maildev	41
3.2.9 Microsoft Azure και blob	41
3.2.10 Repository Pattern	42
Κεφάλαιο 4: Προγράμματα, οδηγίες, περιβάλλον και η εφαρμογή.....	45
4.1 Προγράμματα και οδηγίες	45
4.1.1 Προγράμματα που χρησιμοποιήθηκαν	45
4.1.2 Πως τρέχουμε την εφαρμογή σε local development environment	45

4.2 Η εφαρμογή - Symposium.....	46
4.2.1 Σχεδιάγραμμα βάσης δεδομένων - εφαρμογής	51
4.2.2 Αρχιτεκτονική εφαρμογής	51
4.2.3 Αναλυτική επεξήγηση front-end, back-end	51
4.3 Προστιθέμενη αξία.....	55
4.4 Δημιουργία εσόδων (monetization)	55
Κεφάλαιο 5: Προτάσεις για βελτίωση, συμπεράσματα και επίλογος.....	57
Βιβλιογραφία.....	59

Κεφάλαιο 1ο: Εισαγωγή

1.1 Η εφαρμογή και το πρόβλημα

Η εφαρμογή αποτελεί ένα κοινωνικό δίκτυο το οποίο έχει στοιχεία blogging. Η βασική λειτουργία της πλατφόρμας είναι ο εγγεγραμμένος χρήστης να δημοσιοποιεί κείμενα, φωτογραφίες κ.ά. Η εγγραφή στην πλατφόρμα είναι υποχρεωτική, καθώς ο μη εγγεγραμμένος χρήστης δεν έχει καθόλου πρόσβαση. Οι τεχνολογίες που χρησιμοποιήθηκαν περιγράφονται στα επόμενα κεφάλαια, όπως επίσης η αρχιτεκτονική και η λειτουργικότητα της. Επίσης, είναι responsive, οπότε η σύνδεση από κινητό ή tablet είναι δυνατή. Η εφαρμογή γενικότερα είναι πολύ απλή στην χρήση της και υπάρχουν σημάνσεις για οποιαδήποτε λειτουργικότητα.

1.2 Γιατί Web εφαρμογή;

Η επιλογή της πλατφόρμας υλοποίησης της εφαρμογής αποφασίστηκε να γίνει στο Web (διαδικτυακή εφαρμογή), καθώς δεν υπάρχει κάποια πρωταρχική ανάγκη υλοποίησης σε mobile ή desktop συσκευής. Η πρόσβαση μπορεί να γίνει και από smartphone που διαθέτουν περιηγητές (browsers). Οι τοπικές εφαρμογές υπολογιστών παραμένουν μέχρι και σήμερα οι πιο δημοφιλείς εφαρμογές και αυτό οφείλεται εν μέρει στο ότι οι τοπικές εφαρμογές ήταν και οι πρώτες εφαρμογές που δημιουργήθηκαν. Όμως, με την πάροδο του χρόνου όλο και περισσότερες διαδικτυακές εφαρμογές κάνουν την εμφάνισή τους. Συγκρίνοντας τις δύο κατηγορίες εφαρμογών, οι χρήστες θα εντοπίσουν αντίστοιχα τόσο θετικά όσο και αρνητικά στοιχεία. Οπότε, σε συνάρτηση με τις εργασίες που επιθυμούν η εφαρμογή τους να πραγματοποιεί καλούνται να αποφασίσουν ποια από τις δύο κατηγορίες τους ταιριάζει.

Πλεονεκτήματα

- **Άμεση πρόσβαση από οποιαδήποτε συσκευή:** Οι χρήστες των διαδικτυακών εφαρμογών έχουν άμεση προσβασιμότητα στις εφαρμογές που θέλουν να χρησιμοποιήσουν από οποιονδήποτε υπολογιστή ή άλλη συσκευή έχει ίντερνετ χωρίς την εγκατάσταση κάποιου επιπρόσθετου λογισμικού. Η μόνη απαραίτητη εφαρμογή είναι ο περιηγητής Διαδικτύου ο οποίος είναι προεγκατεστημένος σε όλα τα λειτουργικά συστήματα ακόμα και στις φορητές συσκευές αλλά και στα κινητά τηλέφωνα. Η ιδιότητα αυτή των διαδικτυακών εφαρμογών είναι ιδιαίτερα σημαντική για μεγάλες επιχειρήσεις με πολλούς χρήστες που στην περίπτωση της τοπικής εφαρμογής θα έπρεπε να εγκατασταθεί η εφαρμογή σε κάθε ένα υπολογιστή ξεχωριστά.
- **Δυνατότητα χρήσης ανεξαρτήτως τοποθεσίας:** Ως συνέχεια των παραπάνω, οι χρήστες των διαδικτυακών εφαρμογών μπορούν να τις χρησιμοποιούν ακόμα και αν δεν βρίσκονται στο χώρο εργασίας τους. Η δυνατότητα αυτή δίνει ευελιξία στους χρήστες, ώστε να χρησιμοποιούν τις εφαρμογές οπουδήποτε επιθυμούν, επιτρέποντάς τους να εργάζονται ακόμα και από απομακρυσμένες περιοχές ή και από το σπίτι τους.
- **Συμβατές με όλα τα λειτουργικά συστήματα:** Ένα ακόμα πλεονέκτημα των διαδικτυακών εφαρμογών είναι ότι είναι συμβατές με όλα τα λειτουργικά συστήματα. Η εφαρμογή εκτελείται μέσω του περιηγητή του Διαδικτύου και όχι στον υπολογιστή του χρήστη, οπότε την κάνει ικανή να εκτελείται σε όλα τα λειτουργικά συστήματα. Η

ιδιότητα αυτή οφείλεται επίσης και στην προτυποποίηση των γλωσσών προγραμματισμού τις οποίες χρησιμοποιεί η εφαρμογή.

- **Δεν καταναλώνουν πόρους:** Ως συνέχεια των παραπάνω και εφόσον οι διαδικτυακές εφαρμογές δεν εκτελούνται στον υπολογιστή του χρήστη, δεν καταναλώνουν και πόρους από το σύστημα. Για τον λόγο αυτό, οι εφαρμογές Διαδικτύου είναι ιδιαίτερα ελαφριές για την υπολογιστική μονάδα.
- **Δεν καταλαμβάνουν χώρο:** Ακολουθώντας την προαναφερθείσα λογική, οι εφαρμογές αυτές δεν καταλαμβάνουν καθόλου ή σχεδόν καθόλου χώρο στο δίσκο του χρήστη. Γιατί, το σύνολο της εφαρμογής είναι αποθηκευμένο στον εξυπηρετητή και μόνο κατά την χρήση της εφαρμογής μπορεί να υπάρχει μεταφορά δεδομένων προς την υπολογιστική μονάδα του χρήστη και μόνο στην περίπτωση που ο χρήστης το επιθυμεί.
- **Γρήγορη αναβάθμιση:** Σημαντικό πλεονέκτημα, συγκριτικά με τις τοπικές εφαρμογές, εμφανίζεται στις περιπτώσεις που η εφαρμογή χρειάζεται κάποια αναβάθμιση. Σε μια κλασική τοπική εφαρμογή η αναβάθμιση του συστήματος θα πρέπει να γίνει σε κάθε έναν υπολογιστή ξεχωριστά, πράγμα που απαιτεί χρόνο και χρήμα. Αντίθετα, σε μια διαδικτυακή εφαρμογή η αναβάθμιση πραγματοποιείται μόνο στον εξυπηρετητή που φιλοξενεί την εφαρμογή και ταυτόχρονα το αναβαθμισμένο πρόγραμμα είναι διαθέσιμο σε όλους τους χρήστες. Με τον τρόπο αυτό, εξοικονομείται χρόνος, ο οποίος είναι ιδιαίτερα πολύτιμος κυρίως για τις μεγάλες επιχειρήσεις, με συνέπεια να μειώνεται σημαντικά και το κόστος της αναβάθμισης μιας και απαιτείται λιγότερο εργατικό δυναμικό για την διεκπεραίωσή της.
- **Νέο, βελτιωμένο περιβάλλον:** Ένα ακόμα πλεονέκτημα των διαδικτυακών εφαρμογών είναι ότι πλέον με την εμφάνιση της HTML5 είναι δυνατό ο δημιουργός της εφαρμογής να την εμπλουτίσει ώστε να είναι πιο φιλική, εύχρηστη και ευχάριστη προς τον χρήστη. Παλαιότερα, οι εφαρμογές αυτές υστερούσαν στην εμφάνιση, όμως πλέον είναι ιδιαίτερα εύκολα να εμπλουτιστούν.
- **Δυνατότητα χρήσης και εκτός Διαδικτύου - ενδοδικτύου:** Η δυνατότητα της χρήσης μιας διαδικτυακής εφαρμογής εκτός Διαδικτύου, με την προϋπόθεση ότι η εφαρμογή έχει κατασκευαστεί με ανάλογο τρόπο, αποτελεί πλεονέκτημα των σύγχρονων διαδικτυακών εφαρμογών (εφαρμογές με χρήση HTML5). Για παράδειγμα, αν για κάποιο λόγο η σύνδεση στο Διαδίκτυο διακοπεί αυτό δεν επηρεάζει τον χρήστη, ο οποίος συνεχίζει να χρησιμοποιεί την εφαρμογή κανονικά. Αυτό επιτυγχάνεται από τον περιηγητή, ο οποίος κρατάει ένα αντίγραφο από τα αρχεία στον υπολογιστή του χρήστη, που είναι απαραίτητα για την εκτός δικτύου χρήση της εφαρμογής, και τα χρησιμοποιεί όταν αυτό κριθεί απαραίτητο. Η συγκεκριμένη δυνατότητα δεν είναι διαθέσιμη σε όλες τις εφαρμογές που χρησιμοποιούν HTML5 αλλά μόνο σε αυτές που έχει υπάρξει πρόβλεψη για χρήση της εφαρμογής και εκτός Διαδικτύου ή ενδοδικτύου.

1.2.2 Μειονεκτήματα

- **Χρήση της εφαρμογής εκτός Διαδικτύου:** Ένα μειονέκτημα που ταυτόχρονα είναι και πλεονέκτημα είναι η χρήση της εφαρμογής εκτός Διαδικτύου. Προκειμένου να γίνει αυτό εφικτό θα πρέπει να έχει γίνει πρόβλεψη κατά την σχεδίαση της εφαρμογής και να έχουν ληφθεί τα κατάλληλα μέτρα. Σε περίπτωση που κάτι τέτοιο δεν έχει γίνει τότε η

εφαρμογή δεν είναι δυνατό να χρησιμοποιηθεί χωρίς την σύνδεση του χρήστη στο Διαδίκτυο ή το ενδοδίκτυο της εταιρείας.

- **Αδυναμία χρήσης χωρίς σύνδεση στο Διαδίκτυο:** Σαν συνέχεια των παραπάνω οι εφαρμογές που δεν έχουν κατασκευαστεί με χρήση της τελευταίας έκδοσης της HTML, δηλαδή παλαιότερες εφαρμογές, είναι αδύνατο να χρησιμοποιηθούν αν δεν υπάρχει σύνδεση με το Διαδίκτυο ή το ενδοδίκτυο. Αυτό αποτελεί και το κύριο μειονέκτημα των παλαιότερων διαδικτυακών εφαρμογών, καθώς αν για κάποιο λόγο η σύνδεση του χρήστη διακοπεί τότε αυτός δεν μπορεί να χρησιμοποιήσει καθόλου την εφαρμογή. Σε περίπτωση που μια παλαιότερη διαδικτυακή εφαρμογή θελήσει να αυξήσει τις δυνατότητές της προκειμένου να μπορεί να λειτουργεί και εκτός Διαδικτύου, τότε αυτή θα πρέπει να ανακατασκευαστεί σε μεγάλο μέρος από την αρχή.
- **Μη πλήρης συμβατότητα των περιηγητών:** Ένα ακόμα μειονέκτημα που αφορά την τελευταία έκδοση της HTML είναι η μη πλήρης συμβατότητα των περιηγητών με την έκδοση αυτή. Αν και υπάρχει πληθώρα δυνατοτήτων και πλεονεκτημάτων από τη χρήση της HTML5, αρκετοί από τους περιηγητές δεν είναι ακόμα πλήρως συμβατοί με αυτά. Έτσι, δεν γίνεται πλήρης χρήση των δυνατοτήτων αυτών, πράγμα που περιορίζει τους προγραμματιστές που έχουν αναλάβει ένα έργο. Επίσης, σε περίπτωση που δεν έχει προβλεφθεί η μη λειτουργία κάποιου χαρακτηριστικού της εφαρμογής σε κάποιον περιηγητή, μπορεί να δημιουργήσει προβλήματα στην εφαρμογή με αποτέλεσμα να μην λειτουργεί σωστά ή να μην λειτουργεί καθόλου. Για τον λόγο αυτό, ο κατασκευαστής της εφαρμογής με τον πελάτη πρέπει από κοινού να αποφασίζουν ποιος περιηγητής θα είναι ο προτεινόμενος για την εφαρμογή αλλά ταυτόχρονα να προβλέπεται και η περίπτωση χρήσης άλλων περιηγητών. Ένας καλός τρόπος για να ελεγχθεί η συμβατότητα του περιηγητή μας με την HTML5 είναι τα διάφορα διαδικτυακά τεστ που αξιολογούν τις δυνατότητές του.
- **Άμεση αναβάθμιση:** Η αναβάθμιση της εφαρμογής, αποτελεί μειονέκτημα και πλεονέκτημα ταυτοχρόνως. Στην περίπτωση της τοπικής εφαρμογής μια επιχείρηση μπορεί να αναβαθμίσει την εφαρμογή που χρησιμοποιεί, όποτε αυτή το θεωρεί αναγκαίο, κρίνοντας το κόστος αναβάθμισης, την αξιοπιστία της νέας εφαρμογής αλλά και το χρόνο που θα χρειαστούν οι υπάλληλοί της, ώστε να προσαρμοστούν στη νέα έκδοση. Αντίθετα, στις διαδικτυακές εφαρμογές η αναβάθμιση γίνεται χωρίς πρώτα να ερωτηθούν όλοι οι χρήστες. Για παράδειγμα στην περίπτωση που η ερχόμενη αναβάθμιση μιας τοπικής εφαρμογής έχει σφάλματα τότε μπορεί κάποιος χρήστης (εταιρεία) να μην πραγματοποιήσει την αναβάθμιση έως ότου διορθωθούν αυτά. Στην περίπτωση όμως της διαδικτυακής εφαρμογής ο χρήστης (εταιρεία) δεν μπορεί να αποτρέψει την αναβάθμιση αυτή.
- **Πιθανή μη συμβατότητα κάποιων στοιχείων της εφαρμογής με μια μελλοντική έκδοση του περιηγητή σας:** Τέλος, στα μειονεκτήματα εντάσσεται και η πιθανή μη συμβατότητα κάποιων στοιχείων του προγράμματος με μια μελλοντική έκδοση του περιηγητή μας. Αυτό συμβαίνει συχνά, διότι παρατηρείται το φαινόμενο να εγκαταλείπονται κάποια υποστηριζόμενα στοιχεία από τους περιηγητές, αν η δημιουργός εταιρία κρίνει ότι αυτά δεν έχουν μέλλον στις εφαρμογές Διαδικτύου. Αποτέλεσμα αυτών των αποφάσεων, είναι η δυσλειτουργία κάποιων παλαιότερων εφαρμογών καθιστώντας αναγκαία την αναβάθμισή τους.

Κεφάλαιο 1ο: Διαδίκτυο - Web

1.1 Ο παγκόσμιος ιστός

Παγκόσμιος ιστός (αγγ.: World Wide Web ή www) είναι το ανοιχτό σύστημα διασυνδεδεμένων πληροφοριών και περιεχομένου, το οποίο επιτρέπει στους χρήστες του Διαδικτύου να κάνουν αναζήτηση πληροφοριών μεταβαίνοντας από το ένα έγγραφο στο άλλο. Όλες οι δικτυακές μονάδες του Διαδικτύου αποτελούνται από συνδεδεμένους υπολογιστές στο τοπικό επίπεδο, παράδειγμα αποτελεί ένα δίκτυο υπολογιστών στα κεντρικά γραφεία μίας εταιρίας. Τα δίκτυα αυτά με τη σειρά τους συνδέονται σε ευρύτερα δίκτυα, για παράδειγμα σε εθνικά και υπερεθνικά.

Το ευρύτερο δίκτυο στον κόσμο λέγεται παγκόσμιος ιστός και είναι μοναδικό, αυτό σημαίνει ότι δεν υπάρχουν παραπάνω από ένα δίκτυα υπολογιστών παγκόσμιας κλίμακας. Σε αυτό το παγκόσμιο δίκτυο συμπεριλαμβάνονται τα γήινα δίκτυα και τα δίκτυα των δορυφόρων του και άλλων συσκευών που είναι συνδεδεμένα σε αυτό. Ακόμη, η τεχνολογία του ιστού καθιστά δυνατή τη δημιουργία "υπερκειμένων", δηλαδή μία διασύνδεση πολλών μη ιεραρχημένων στοιχείων τα οποία παλαιότερα ήταν απομονωμένα. Αυτά τα στοιχεία μπορούν να αποκτήσουν κι άλλες μορφές εκτός της μορφής του γραπτού κειμένου, όπως εικόνας και ήχου.

Η τεχνολογία του ιστού που δημιουργήθηκε το 1989 από το Βρετανό Τιμ Μπέρνερς Λι (Tim Berners-Lee), η οποία εκείνη την εποχή εργαζόταν στον Ευρωπαϊκό Οργανισμό Πυρηνικών Ερευνών (CERN) στη Γενεύη της Ελβετίας. Το όνομα που είχε δώσει στην εφεύρεσή του ο ίδιος ο Lee ήταν World Wide Web, όρος που είναι γνωστός στους περισσότερους από το "www". Ο Lee οδηγήθηκε στην εφεύρεση του Παγκόσμιου ιστού μέσω του οράματός του για έναν κόσμο όπου για τον καθένα θα ήταν εφικτό να ανταλλάσσει πληροφορίες και ιδέες άμεσα, οι οποίες θα ήταν προσβάσιμες από τους υπόλοιπους.

Ακόμη, το σημείο στο οποίο δόθηκε ιδιαίτερο βάρος ήταν η μη ιεράρχηση των διασυνδεδεμένων στοιχείων. Οραματίστηκε το κάθε στοιχείο, κάθε κόμβο του ιστού ίσο ως προς την προσβασιμότητα με τα υπόλοιπα. Αν σκεφτεί κανείς το βαθμό ιεράρχησης με τον οποίο λειτουργούν οι μηχανές αναζήτησης του ιστού, για παράδειγμα το google, γίνεται εύκολα κατανοητό ότι κατά την πράξη κάτι τέτοιο δε συμβαίνει, τουλάχιστον στο βαθμό που το είχε οραματιστεί ο Lee. Σαν πληροφοριακό σύστημα παρέχει συγκεκριμένο μοντέλο δεδομένων που βασίζεται σε κόμβους και υπερσυνδέσμους. Το μοντέλο δεδομένων του παγκόσμιου ιστού παρέχει κόμβους (αγγλ. nodes), άγκυρες (αγγλ. anchors) και συνδέσμους (ή υπερσυνδέσμους) (αγγλ. links ή hyperlinks).

Οι κόμβοι είναι οι φορείς του περιεχομένου, ενώ οι άγκυρες είναι οι περιοχές του περιεχομένου των κόμβων από όπου ξεκινούν ή καταλήγουν οι σύνδεσμοι. Οι σύνδεσμοι συνδέουν δύο κόμβους. Αρχικά ο Παγκόσμιος Ιστός (WEB 1.0) συνέδεε δεδομένα και πληροφορίες τα οποία μπορούσαν να παρέχουν στο Διαδίκτυο μόνο όσοι γνώριζαν HTML (Hyper Text Markup Language). Ο χρήστης του Διαδικτύου ήταν παθητικός δέκτης των πληροφοριών, πράγμα το οποίο άλλαξε με την έλευση του WEB 2.0, του διαδραστικού ιστού. Οι χρήστες του Διαδικτύου μπορούν πλέον να αλληλοεπιδρούν μεταξύ τους, να ανταλλάσσουν πληροφορίες, να διαμορφώνουν το περιεχόμενο μιας ιστοσελίδας. Ο Web 2.0 συνδέει άτομα μεταξύ τους και είναι αποτέλεσμα κυρίως του εύκολου τρόπου κατασκευής ιστοσελίδων. Τέλος, τα πρώτα WEB 2.0 εργαλεία είναι τα ιστολόγια (blogs), στα οποία οι χρήστες του Διαδικτύου μπορούν να αφήνουν σχόλια. Άλλα παραδείγματα WEB 2.0 εργαλείων είναι τα wikis και οι ιστοσελίδες κοινωνικής δικτύωσης.

1.2 Το Διαδίκτυο

2.2.1 Ορισμός

Το Διαδίκτυο είναι ένα παγκόσμιο σύστημα διασυνδεδεμένων δικτύων υπολογιστών, τα οποία χρησιμοποιούν μία καθιερωμένη ομάδα πρωτοκόλλων, που συχνά αποκαλείται "TCP/IP" (αν και αυτή δεν χρησιμοποιείται από όλες τις υπηρεσίες του Διαδικτύου) για να εξυπηρετεί δισεκατομμύρια χρήστες καθημερινά στον κόσμο. Οι διασυνδεδεμένοι ηλεκτρονικοί υπολογιστές ανά τον κόσμο, που βρίσκονται σε ένα κοινό δίκτυο επικοινωνίας, ανταλλάσσουν μηνύματα (πακέτα) με την χρήση διαφόρων πρωτοκόλλων (τυποποιημένοι κανόνες επικοινωνίας), που υλοποιούνται σε επίπεδο υλικού και λογισμικού. Το κοινό αυτό δίκτυο καλείται Διαδίκτυο.

2.2.2 Το Διαδίκτυο και η επικοινωνία

Σύμφωνα με την προσέγγιση της "ιντερνετοφιλίας" (ένα μείγμα κλασικής "πλουραλιστικής" προσέγγισης και τεχνολογικού "ντετερμινισμού"), το Διαδίκτυο, αλλά και η ψηφιακή τεχνολογία γενικότερα, παρέχουν την ικανότητα να δημιουργούν "εικονικούς χώρους", "εικονικές κοινότητες", όπου παύουν να υφίστανται οι κοινωνικές και πολιτιστικές διαχωριστικές γραμμές, οι οποίες υπάρχουν στον πραγματικό κόσμο και που τα παραδοσιακά μέσα επικοινωνίας αδυνατούν να ξεπεράσουν εύκολα. Επίσης, η επικοινωνία μέσω του Διαδικτύου καθίσταται άμεση και αμφίδρομη. Κάθε χρήστης δύναται μέσω ενός ηλεκτρονικού υπολογιστή συνδεδεμένου στο Διαδίκτυο, να πληροφορηθεί αλλά και να πληροφορήσει ανταλλάσσοντας απόψεις μέσω ενός πιο συμμετοχικού και λιγότερο ελεγχόμενου διαύλου επικοινωνίας.

Με τον τρόπο αυτό, αποκτάται από του χρήστες ολοένα και περισσότερο την ιδιότητα του παγκόσμιου πολίτη. Υπάρχει έντονη τάση από την αρχή της εμφάνισής του Διαδικτύου, να θεωρείται ένα άκρως δημοκρατικό μέσο μαζικής επικοινωνίας, που αποδιαμεσολαβεί την επικοινωνία και καθιστά ισχυρότερο το μέσο άνθρωπο, καθώς δίνει στον τελευταίο την δυνατότητα πρόσβασης σε μεγάλο όγκο πληροφοριών συγκεντρωμένων σε έναν "χώρο" και τη δυνατότητα της προσωπικής επιλογής των πληροφοριών αυτών. Συνεπώς, η βασική θέση της προσέγγισης αυτής είναι ότι το Διαδίκτυο θα εκδημοκρατίσει την κοινωνία με το να βελτιώσει την επικοινωνία καταργώντας την ανάγκη για διαμεσολάβηση.

2.2.3 Η τεχνολογία του Διαδικτύου

Το Διαδίκτυο είναι ένα επικοινωνιακό δίκτυο που επιτρέπει την ανταλλαγή δεδομένων μεταξύ οποιωνδήποτε διασυνδεδεμένων υπολογιστών. Η τεχνολογία του είναι κυρίως βασισμένη στη διασύνδεση επιμέρους δικτύων ανά τον κόσμο και σε πολυάριθμα πρωτόκολλα επικοινωνίας. Στην πιο εξειδικευμένη και περισσότερο χρησιμοποιούμενη μορφή του, ο όρος Διαδίκτυο περιγράφει το παγκόσμιο πλέγμα διασυνδεδεμένων υπολογιστών και υπηρεσιών και πληροφοριών που παρέχει στους χρήστες του. Το Διαδίκτυο χρησιμοποιεί την μεταγωγή πακέτων και τη στοιβά πρωτοκόλλων. Σήμερα, ο συγκεκριμένος όρος έχει καταλήξει να αναφέρεται στο παγκόσμιο αυτό δίκτυο. Για να ξεχωρίζει, το παγκόσμιο αυτό δίκτυο γράφεται με κεφαλαίο το αρχικό "Δ". Η τεχνική της διασύνδεσης δικτύων μέσω μεταγωγής πακέτων και της στοιβάς πρωτοκόλλων ονομάζεται «διαδικτύωση».

2.2.4 Η ιστορία του Διαδικτύου

Οι πρώτες απόπειρες για τη δημιουργία ενός Διαδικτύου ξεκίνησαν στις ΗΠΑ κατά τη διάρκεια του ψυχρού πολέμου. Η Σοβιετική Ένωση είχε ήδη στείλει στο διάστημα το δορυφόρο Σπούτνικ 1 κάνοντας τους Αμερικανούς να φοβούνται όλο και περισσότερο για την ασφάλεια της χώρας τους. Θέλοντας λοιπόν να προστατευτούν από μια πιθανή πυρηνική επίθεση των Ρώσων δημιούργησαν την υπηρεσία των προηγμένων αμυντικών ερευνών ARPA (Advanced Research Project Agency) γνωστή ως DARPA (Defense Advanced Research Projects Agency) στις μέρες μας.

Αποστολή της συγκεκριμένης υπηρεσίας ήταν να βοηθήσει τις στρατιωτικές δυνάμεις των ΗΠΑ να αναπτυχθούν τεχνολογικά και να δημιουργηθεί ένα δίκτυο επικοινωνίας που θα μπορούσε να επιβιώσει σε μια ενδεχόμενη πυρηνική επίθεση. Το αρχικό θεωρητικό υπόβαθρο δόθηκε από τον Τζ. Λικλάιντερ (J.C.R. Licklider) ο οποίος μνημόνευε σε συγγράμματά του το "γαλαξιακό δίκτυο". Η θεωρία αυτή υποστήριζε την ύπαρξη ενός δικτύου υπολογιστών που θα ήταν συνδεδεμένοι μεταξύ τους και θα μπορούσαν να ανταλλάσσουν γρήγορα στοιχεία και προγράμματα.

Το επόμενο θέμα που ανακύπτει ήταν ότι το δίκτυο αυτό θα έπρεπε να ήταν αποκεντρωμένο έτσι ώστε ακόμη κι αν κάποιος κόμβος του δεχόταν επίθεση να υπήρχε δίοδος επικοινωνίας για τους άλλους υπολογιστές. Την λύση σε αυτό έδωσε ο Πολ Μπάραν (Paul Baran) με το σχεδιασμό ενός κατακεντρωμένου δικτύου επικοινωνίας που χρησιμοποιούσε την ψηφιακή τεχνολογία. Πολύ σημαντικό ρόλο έπαιξε και η θεωρία ανταλλαγής πακέτων του Λέοναρντ Κλάινροκ (Leonard Kleinrock), ο οποίος υποστήριζε ότι πακέτα πληροφοριών που θα περιείχαν την προέλευση και τον προορισμό τους μπορούσαν να σταλούν από έναν υπολογιστή σε έναν άλλο.

Στηριζόμενο λοιπόν σε αυτές τις θεωρίες δημιουργήθηκε το πρώτο είδος Διαδικτύου γνωστό ως ARPANET. Εγκαταστάθηκε και λειτούργησε για πρώτη φορά κατά το 1969 με 4 κόμβους μέσω των οποίων συνδέονται 4 μίνι υπολογιστές (mini computers 12k): του πανεπιστημίου της Καλιφόρνια στην Σάντα Μάρμπαρα του πανεπιστημίου της Καλιφόρνια στο Λος Άντζελες, το S.R.I International στο Στάνφορντ και το πανεπιστήμιο της Γιούτα. Η ταχύτητα του δικτύου έφθανε τα 50 kbps και έτσι επιτεύχθηκε η πρώτη dial up σύνδεση μέσω γραμμών τηλεφώνου.

Μέχρι το 1972 οι συνδεδεμένοι στο ARPANET υπολογιστές έχουν φτάσει τους 23, οπότε και εφαρμόζεται για πρώτη φορά το σύστημα διαχείρισης ηλεκτρονικού ταχυδρομείου (e-mail). Παράλληλα δημιουργήθηκαν και άλλα δίκτυα, που χρησιμοποιούσαν διαφορετικά πρωτόκολλα (όπως το X.25 και το UUCP) τα οποία συνδέονται με το ARPANET. Το πρωτόκολλο που χρησιμοποιούσε το ARPANET ήταν το NCP (Network Control Protocol), το οποίο, όμως, είχε το μειονέκτημα ότι λειτουργούσε μόνο με συγκεκριμένους τύπους υπολογιστών. Έτσι, δημιουργήθηκε η ανάγκη στις αρχές του 1970 για ένα πρωτόκολλο που θα έωνε όλα τα δίκτυα που είχαν δημιουργηθεί μέχρι τότε.

Το 1974 λοιπόν, δημοσιεύεται η μελέτη των Βιντ Σερφ (Vint Cerf) και Μπομπ Κάαν (Bob Kahn) από την οποία προέκυψε το πρωτόκολλο TCP (Transmission Control Protocol) που αργότερα, το 1978 έγινε TCP/IP, προστέθηκε δηλαδή το Internet Protocol (IP), ώσπου το 1983 έγινε το μοναδικό πρωτόκολλο το οποίο ακολουθούσε το ARPANET. Το 1984 υλοποιείται το πρώτο DNS (Domain Name System) σύστημα στο οποίο καταγράφονται 1.000 κεντρικοί κόμβοι και οι υπολογιστές του Διαδικτύου πλέον αναγνωρίζονται από διευθύνσεις κωδικοποιημένων αριθμών.

Ένα ακόμη σημαντικό βήμα στην ανάπτυξη του Διαδικτύου έκανε το Εθνικό Ίδρυμα Επιστημών (National Science Foundation, NSF) των ΗΠΑ, το οποίο δημιούργησε την πρώτη διαδικτυακή πανεπιστημιακή ραχοκοκαλιά (backbone), το NSFNet, το 1986. Ακολούθησε η ενσωμάτωση άλλων σημαντικών δικτύων, όπως το Usenet, το Fidonet και το Bitnet. Ο όρος Διαδίκτυο/Ίντερνετ ξεκίνησε να χρησιμοποιείται ευρέως την εποχή που συνδέθηκε το ARPANET με

το NSFNet και Internet σήμαινε οποιοδήποτε δίκτυο χρησιμοποιούσε TCP/IP. Η μεγάλη άνθιση του Διαδικτύου όμως, ξεκίνησε με την εφαρμογή της υπηρεσίας του Παγκόσμιου Ιστού από τον Τιμ Μπέρνερς-Λι στο ερευνητικό ίδρυμα CERN το 1989, ο οποίος είναι στην ουσία, η "πλατφόρμα", η οποία κάνει εύκολη την πρόσβαση στο Ίντερνετ, ακόμη και στη μορφή που είναι γνωστό σήμερα.

2.2.5 Οι πληροφορίες στο Διαδίκτυο

Το Διαδίκτυο, σε συνδυασμό με την ολοένα αναπτυσσόμενη ψηφιακή τεχνολογία, έχει δημιουργήσει μία τεράστια αγορά γνώσεων/πληροφοριών. Παραδοσιακές μορφές τέχνης, όπως για παράδειγμα ο κινηματογράφος και η μουσική, μέσω της ψηφιακής τεχνολογίας παίρνουν την ίδια μορφή (αρχείων δεδομένων) με αντικείμενα που εκ πρώτης όψεως είναι εντελώς διαφορετικά, όπως για παράδειγμα η ιατρική επιστήμη ή κάποιο πρόγραμμα λογισμικού. Παρατηρείται λοιπόν, μία συγκέντρωση γνώσης ή, αν είναι δυνατόν να λεχθεί, πολιτιστικής κληρονομιάς, η οποία σχετίζεται άμεσα με το Ίντερνετ. Το μεγάλο ερώτημα που προκύπτει πλέον είναι το "ποιος θα διοικήσει, ποιος θα ελέγξει τη γνώση αυτή".

Από την στιγμή που το Διαδίκτυο είναι ένα δίκτυο συνδεδεμένων υπολογιστών, κάθε χρήστης έχει την δυνατότητα να μοιραστεί πληροφορίες με άλλους χρήστες γενόμενος, αρκετές φορές, ο ίδιος δημιουργός και πάροχος των πληροφοριών αυτών. Δεν υπάρχει άμεσος έλεγχος των πληροφοριών που "ανεβαίνουν" στο Διαδίκτυο από κάποιον ιεραρχικά ανώτερο χρήστη ή οργανισμό. Το θέμα της μη ιεραρχημένης πληροφορίας τίθεται υπό αμφισβήτηση. Ο όγκος της πληροφορίας στο Διαδίκτυο είναι πράγματι μεγάλος. Παρόλα αυτά, υπάρχουν πληροφορίες ευκολότερα και δυσκολότερα προσβάσιμες από το χρήστη.

Το Διαδίκτυο κατέστησε εφικτή την συγκέντρωση μεγάλου όγκου πληροφοριών και επηρέασε σημαντικά τον τρόπο διάθεσής τους, δεν συμβαίνει όμως στον ίδιο βαθμό το ίδιο και στον τρόπο παραγωγής αυτών. Για παράδειγμα, ο τρόπος παραγωγής μιας κινηματογραφικής ταινίας δεν έχει επηρεαστεί σημαντικά από την ύπαρξη του Διαδικτύου, ανεξάρτητα από το αν έχει επηρεαστεί ή όχι από την ψηφιακή τεχνολογία. Παρ' όλα αυτά, και σύμφωνα με την "ιντερνετοφιλική" προσέγγιση, το Διαδίκτυο ασκεί μεγάλη επίδραση στην διαδικασία παραγωγής δημοσιογραφικών προϊόντων. Η δημιουργία της είδησης παύει να είναι πλέον μονοπώλιο λίγων, αφού ο κάθε χρήστης μπορεί αν το επιθυμεί να δημιουργήσει πληροφορία ανά πάσα στιγμή.

Το πιο τρανταχτό παράδειγμα της επίδρασης αυτής είναι τα ιστολόγια (blogs), στα οποία μπορεί κανείς να εκφέρει απόψεις και να σχολιάσει γεγονότα πάσης φύσεως (βλ. δημοσιογραφία στον ιστό και δημοσιογραφία των πολιτών). Σαν αποτέλεσμα της επιρροής αυτής του Ίντερνετ στην παραγωγή ειδήσεων, τα όρια μεταξύ ενός απλού χρήστη του Διαδικτύου και ενός επαγγελματία δημοσιογράφου γίνονται περισσότερο δυσδιάκριτα. Αυτό με την σειρά του οδηγεί στην ανάγκη για επαναπροσδιορισμό της έννοιας της δημοσιογραφίας καθώς και της απαραίτητης εκπαίδευσης των δημοσιογράφων. Η ανάγκη για τον επαναπροσδιορισμό της δημοσιογραφίας, όμως, δεν είναι τόσο μεγάλη σύμφωνα με τους υποστηρικτές της "αντι-πλουραλιστικής" προσέγγισης, καθώς θεωρούν πως το Ίντερνετ δεν μπορεί να ασκήσει ουσιαστική επίδραση στην επικοινωνία γενικότερα και στην δημοσιογραφία ειδικότερα.

Επιπλέον, λόγω της μεγάλης συγκέντρωσης γνώσης στο Διαδίκτυο, η έννοια της κοινωνικής ισότητας παίρνει και πάλι μεγάλη σημασία. Το χάσμα ανάμεσα σε πληροφοριακά πλούσιους και πληροφοριακά φτωχούς θα διευρύνεται όσο αυξάνεται η συγκέντρωση της γνώσης αυτής. Το παραπάνω αποτελεί έναν ακόμη λόγο που κάνει πιο επιτακτική την ανάγκη για διερεύνηση του αρχικού ερωτήματος "ποιος θα ελέγξει τη γνώση αυτή". Η γλώσσα που χρησιμοποιείται περισσότερο στη διακίνηση της πληροφορίας στο Διαδίκτυο είναι η Αγγλική. Έχοντας αναπτυχθεί, τα τελευταία

χρόνια, το Διαδίκτυο περιλαμβάνει πια ποιοτικά και ποσοτικά ευρύ περιεχόμενο και στις υπόλοιπες γλώσσες των περισσότερο ανεπτυγμένων χωρών. Παρόλα αυτά, υπάρχουν ακόμη δυσλειτουργίες και τεχνικά προβλήματα σχετικά με τη κωδικοποίηση, όπως το mojibake.

2.2.6 Νομικά και ηθικά ζητήματα

Η παραβίαση των πνευματικών δικαιωμάτων, η πορνογραφία, η ψευδοπροσωπία και η προσφορά παρανόμων προϊόντων αποτελούν φαινόμενα υπαρκτά στο Ίντερνετ και ο περιορισμός τους είναι ιδιαίτερα δύσκολος. Για παράδειγμα, η λέξη "sex" είναι μία από τις πλέον δημοφιλείς στις μηχανές αναζήτησης. Συχνά, η ανησυχία αυτή, που θεωρείται αβάσιμη, μπορεί να υποστηριχθεί και από κάποια εγκλήματα ή αποτρόπαιες καταστάσεις (συνήθως περιπτώσεις παιδεραστίας κ.ά.). Το Διαδίκτυο έχει κατηγορηθεί και ως παράγοντας που έπαιξε ρόλο σε θανάτους.

Ο Μπράντον Βέντας (Brandon Vedas) πέθανε από την υπερβολική δόση ενός μίγματος νόμιμων και παράνομων ναρκωτικών παρακινούμενος από συνομιλητές του στο IRC. Ο Σων Γούλεϊ (Shawn Woolley) αυτοκτόνησε με πιστόλι για λόγους που οι οποίοι σχετίζονται με τον εθισμό του με το EverQuest, ένα Μαζικά Πολυχρηστικό Διαδικτυακό Παιχνίδι Ρόλων (MMORPG), όπως ισχυρίστηκε η μητέρα του. Ο Άρμιν Μάιβες (Armin Meiwes) μαχαίρωσε μέχρι θανάτου και έφαγε μέρος του σώματος του Μπερντ-Γιούργκεν Μπράντες (Bernd Jürgen Brandes) όταν ο τελευταίος απάντησε στην αγγελία του πρώτου που ζητούσε έναν «μεγαλόσωμο άντρα έτοιμο να σφαγιαστεί και μετά να καταβροχθιστεί».

Ακόμη, το Διαδίκτυο είναι μη ελεγχόμενο, υπό την έννοια ότι δεν υπάρχει κάποια ενιαία κυβερνητική ή άλλη αντίστοιχη αρχή, που θα ελέγχει το περιεχόμενό του πριν αυτό δημοσιευθεί -με βάση πολλούς χρήστες αυτό θα αποτελούσε λογοκρισία. Είναι χαρακτηριστικό ότι λέγεται πως "το Διαδίκτυο ελέγχεται από τους χρήστες του". Βέβαια, οι κρατικές υπηρεσίες και αστυνομίες σε κάθε χώρα, καθώς και οι αντίστοιχες νομοθετικές ρυθμίσεις, παρεμβαίνουν για την αναστολή των αξιόποινων πράξεων που διαπράττονται μέσω Διαδικτύου.

Επίσης, ηθικό ζήτημα αποτελεί και ο συγκεντρωτισμός των Μ.Μ.Ε. και αναφέρεται στο ολιγοπώλιο, μικρού σχετικά αριθμού, εταιριών οι οποίες κατέχουν τα μέσα και ελέγχουν όλη την αλυσίδα διανομής του προϊόντος. Στα πλαίσια του Διαδικτύου τίθεται το ερώτημα του κατά πόσο οι οικονομικές διαδικασίες του παρόντος καπιταλιστικού γίνεσθαι περιορίζουν την δημόσια σφαίρα και το αν είναι αποδεκτή ή κατακριτέα η πρωτοφανής ισοτιμία στην παρουσία και διαχείριση της πληροφορίας και του εμπορεύματος στον χώρο του Ίντερνετ. Επιπλέον, παρά το γεγονός ότι το Ίντερνετ συχνά περιγράφεται και ως αποκεντρωμένο, με απροσπέλαστο όγκο πληροφοριών και, συνεπώς, χωρίς κεντρικό έλεγχο, είναι εμφανής η εκτενής ιεράρχηση του περιεχομένου από μηχανές αναζήτησης και η γενικότερη διαίωνιση των ιστοτόπων με την υψηλότερη επισκεψιμότητα.

2.2.7 Πρόσβαση στο Διαδίκτυο

Οι κοινές μέθοδοι πρόσβασης στο Διαδίκτυο είναι η επιλογική και η ευρυζωνική. Δημόσιοι χώροι για χρήση του Διαδικτύου περιλαμβάνουν βιβλιοθήκες και Internet cafes, στα οποία υπάρχουν διαθέσιμοι υπολογιστές με σύνδεση στο Διαδίκτυο. Υπάρχουν επίσης, και σημεία πρόσβασης στο Διαδίκτυο σε δημόσιους χώρους όπως είναι οι αίθουσες αναμονής αεροδρομίων, μερικές φορές μόνο για σύντομη χρήση όσο περιμένουμε. Τέτοια σημεία είναι γνωστά και με διάφορους άλλους όρους, όπως «δημόσια περίπτερα Διαδικτύου», «δημόσια τερματικά Διαδικτύου» και «ιστο - τηλέφωνα».

Η δικτύωση μέσω Wi-Fi παρέχει μία ασύρματη πρόσβαση στο Διαδίκτυο. Ασύρματα σημεία πρόσβασης (hotspot) τα οποία παρέχουν τέτοια πρόσβαση περιλαμβάνουν τα Wifi-cafes, όπου

κάποιος αρκεί να φέρει τις δικές του/της ασύρματες συσκευές όπως για παράδειγμα φορητό Η/Υ ή PDA. Αυτές οι υπηρεσίες μπορεί να είναι δωρεάν σε όλους, είτε δωρεάν μόνο σε πελάτες, είτε επί πληρωμή. Ένα hotspot δε χρειάζεται να περιορίζεται σε ένα μόνο συγκεκριμένο περιβάλλον. Ολόκληρες πανεπιστημιούπολεις και πάρκα έχουν αυτήν την δυνατότητα, ακόμη και ολόκληρες περιοχές.

Επίσης, οι προσπάθειες που έχουν γίνει να συνδεθεί και ο αγροτικός πληθυσμός οδήγησαν στα ασύρματα κοινοτικά δίκτυα. Τα πλεονεκτήματα της πρόσβασης ενός χρήστη μέσω του δικού του υπολογιστή (αντί μέσω δημόσιου τερματικού) περιλαμβάνουν την δυνατότητα για κατέβασμα και ανέβασμα αρχείων χωρίς περιορισμούς, την χρήση του αγαπημένου του φυλλομετρητή (ή προγράμματος ανάγνωσης ιστοσελίδων, που αποτελεί ορθότερη ορολογία των λέξεων: web browser) και των ρυθμίσεων αυτού (το μενού των ρυθμίσεων μπορεί να απενεργοποιηθεί σε ένα δημόσιο υπολογιστή) και την εκτέλεση δραστηριοτήτων στο Διαδίκτυο με την χρήση δικών του προγραμμάτων και δεδομένων. Χώρες με πολύ καλή πρόσβαση στο Ίντερνετ περιλαμβάνουν τη Νότια Κορέα, όπου το 50% του πληθυσμού έχει ευρυζωνική πρόσβαση, τη Σουηδία και τις ΗΠΑ.

2.2.8 Διαδίκτυο και Ευρωπαϊκή Ένωση

Το δικαίωμα των Ευρωπαίων πολιτών για την ελεύθερη πρόσβαση στο Διαδίκτυο κατοχυρώνεται στο άρθρο 11 του Χάρτη των Θεμελιωδών Δικαιωμάτων της Ευρωπαϊκής Ένωσης περί ελευθερίας της έκφρασης και της ενημέρωσης. Προσφάτως, στο Ευρωπαϊκό Κοινοβούλιο ψηφίστηκε τροπολογία σύμφωνα με την οποία «δεν μπορεί να επιβάλλεται περιορισμός επί των θεμελιωδών δικαιωμάτων και ελευθεριών των τελικών χρηστών, χωρίς να προηγηθεί δικαστική απόφαση... εκτός από περιπτώσεις όπου απειλείται η ασφάλεια των πολιτών και στις οποίες η απόφαση δύναται να είναι αντίστοιχη».

Επιπλέον, με την εν λόγω τροπολογία η πρόσβαση στο Διαδίκτυο θα μπορεί να απαγορευτεί με σχετικές δικαστικές αποφάσεις που θα επιβάλλει η εκάστοτε εθνική νομοθεσία στο όνομα της απειλής της ασφάλειας. Συγκεκριμένα, η τροπολογία αναφέρει επίσης «...η πρόσβαση στο Διαδίκτυο δεν μπορεί να περιοριστεί χωρίς να προηγηθεί δικαστική απόφαση. Εξαιρούνται οι περιπτώσεις όπου απειλείται η ασφάλεια των πολιτών.» Χαρακτηριστικό παράδειγμα αποτελεί η Βρετανία, στην οποία οι πάροχοι απαγόρευαν την πρόσβαση σε μια λίστα ιστοσελίδων στην οποία μέχρι τώρα βρίσκονταν σελίδες παιδικής πορνογραφίας, όμως πρόσφατα προστέθηκαν και άλλες, όπως αυτή που αφορά το χάκινγκ (hacking).

Στους χρήστες που θα επιχειρούν να εισέλθουν σε κάποια από αυτές τις σελίδες θα απαγορεύεται η είσοδος, ενώ τα ηλεκτρονικά τους ίχνη θα καταγράφονται. Έτσι, παρά την εν λόγω τροπολογία, εξακολουθεί να μην λαμβάνεται υπόψη ότι το αδιάσειστο δικαίωμα της πρόσβασης των πολιτών στο Διαδίκτυο αποτελεί προαπαιτούμενο για την προάσπιση και άλλων θεμελιωδών δικαιωμάτων όπως η γνώση, η παιδεία η ελευθερία έκφρασης και πολιτικής δράσης. Είναι σημαντικό, επίσης, να κατανοηθεί πως οι χρήστες του Διαδικτύου δεν είναι πελάτες αλλά πολίτες και ως τέτοιοι θα πρέπει να λογίζονται σε θέματα που αφορούν αφενός την υποδομή του Διαδικτύου και αφετέρου το δικαίωμα πρόσβασης σε αυτό.

Τέλος, όσον αφορά την υποδομή, οφείλει η εκάστοτε εθνική αρχή να φροντίζει για την επέκταση του δικτύου, ακόμη και σε περιοχές στις οποίες η ιδιωτική πρωτοβουλία αρνείται να προβεί στην απαιτούμενη επένδυση, αν τη θεωρεί οικονομικά ασύμφορη. Με αυτόν τον τρόπο, θα διασφαλιστεί το δικαίωμα των πολιτών για ενημέρωση και ελευθερία έκφρασης. Όσον αφορά την πρόσβαση πρέπει να κατοχυρώνεται το δικαίωμα των πολιτών για ελεύθερη και ισότιμη πρόσβαση όπως αναφέρθηκε και με τα παραπάνω.

2.2.9 Κίνδυνοι του Διαδικτύου

Εγκυμονούν πολλοί κίνδυνοι κατά την πρόσβασή στο Διαδίκτυο, ανεξαρτήτως από τον τρόπο χρήσης των υπηρεσιών του. Βρίσκονται κακόβουλοι χρήστες με αρκετές δυνατότητες πρόκλησης ζημιών, τόσο στο επίπεδο του χρησιμοποιούμενου λογισμικού και υλικού, όσο και σε προσωπικό επίπεδο.

Πρόκληση ζημιών στο υπολογιστικό σύστημα

Ο κύριος κίνδυνος πρόκλησης ζημιών στο υπολογιστικό σύστημα ενός ανυποψίαστου χρήστη είναι η μόλυνση του συστήματος με κάποιον ιό. Η μόλυνση γίνεται όταν ο χρήστης καλείται να λάβει κάποιο -φαινομενικά αθώο- αρχείο όπως ένα κείμενο ή μια φωτογραφία και, όταν δοκιμάσει να το χρησιμοποιήσει, ο ιός αναλαμβάνει δράση επί μολύνοντας το σύστημα. Μπορεί να καταστρέψει αρχεία ή και ολόκληρο τον σκληρό δίσκο του συστήματος. Άλλες φορές είναι δυνατή η αποστολή ιού απευθείας από τον ιστοτόπο που επισκέπτεται ο χρήστης, χωρίς να εμφανιστεί κάποια ένδειξη λήψης αρχείου. Η περίπτωση αυτή εκμεταλλεύεται κενά ασφαλείας στο λογισμικό του χρήστη (φυλλομετρητή ή Λειτουργικό σύστημα).

Παρόμοιας δράσης είναι και ένα πρόγραμμα που αποκαλείται worm (=σκουλήκι). Είναι παρόμοιο σε αποτέλεσμα με τον ιό, αλλά, αντίθετα από αυτόν, δεν απαιτεί την "προσκόλλησή" του σε ένα αρχείο, έχοντας έτσι περισσότερη αυτονομία. Η βλάβη που προκαλεί το worm δεν είναι τόσο ευρεία στο σύστημα, όσο στο δίκτυο σύνδεσης, επειδή καταναλώνει σημαντικό εύρος ζώνης (bandwidth). Άλλος κίνδυνος είναι ο Δούρειος Ίππος, ένα πρόγραμμα που ξεγελά τον χρήστη του, ο οποίος χρησιμοποιώντας το νομίζει ότι εκτελεί κάποια εργασία, ενώ στην πραγματικότητα εκτελεί κάποια άλλη, συνήθως εγκατάσταση άλλων κακόβουλων προγραμμάτων. Αντίθετα από τους ιούς, οι δούρειοι ίπποι δεν επιμολύνουν αρχεία.

Πρόκληση ζημιών σε προσωπικά δεδομένα

Στην κατηγορία αυτή υπάγονται τόσο οι δούρειοι ίπποι που προαναφέρθηκαν, όσο και κακόβουλα μηνύματα ηλεκτρονικού ταχυδρομείου. Με τον τρόπο αυτό όχι μόνον είναι δυνατό να υφαρπαγούν προσωπικά δεδομένα κάποιου χρήστη, όπως ο αριθμός ταυτότητάς του ή το ΑΦΜ του, όσο και, πιο σημαντικό, αριθμοί πιστωτικών καρτών, λογαριασμών τραπεζής κτλ. Ανάλογη μέθοδος ακολουθείται και από ορισμένους ιστοτόπους, στους οποίους ο ανύποπτος χρήστης καταχωρεί παρόμοια στοιχεία παραγγέλοντας ένα προϊόν, το οποίο όχι μόνο δε θα λάβει ποτέ, αλλά τα δεδομένα του μπορούν να χρησιμοποιηθούν από τους δημιουργούς του ιστοτόπου για να πραγματοποιήσουν οι ίδιοι αγορές, χρεώνοντας τον "πελάτη" τους.

Η μέθοδος υφαρπαγής προσωπικών δεδομένων μέσω ηλεκτρονικού ταχυδρομείου αποκαλείται "Phishing" (παραφθορά της λέξης fishing = ψάρεμα). Αρκετά προγράμματα περιήγησης (browsers) αναγνωρίζουν τους ιστοτόπους στους οποίους παραπέμπουν τα παραπλανητικά μηνύματα, ωστόσο αυτό δεν συμβαίνει σε ποσοστό 100%. Οι χρήστες είναι καλό να γνωρίζουν ότι κανείς χρηματοπιστωτικός φορέας δεν χρησιμοποιεί το Διαδίκτυο για να ανανεώσει προσωπικές πληροφορίες, ενώ ένας προστατευμένος ιστοτόπος αρχίζει πάντα με το πρόθεμα https (secure, ασφαλής).

Παραπλάνηση

Συχνά, οι χρήστες του Διαδικτύου χρησιμοποιούν τις υπηρεσίες του για να βρουν πληροφορίες που χρειάζονται. Πληθώρα ιστοτόπων εμφανίζουν πληροφορίες, οι οποίες φαινομενικά είναι ακριβείς ή αναφέρουν απόλυτα αξιόπιστους δημιουργούς ή πηγές. Το κίνητρο για τέτοιες πράξεις μπορεί να

είναι είτε η αποκομιδή ιδίου οφέλους είτε, απλά, η χαρά της παραπλάνησης των (αγνώστων) χρηστών. Ο όρος που περιγράφει αυτού του τύπου την παραπλάνηση είναι "Hoax". Για περισσότερες πληροφορίες δείτε το Μουσείο των Hoaxes (αγγλικά).

Προστασία (phishing & pharming)

Υπάρχουν τρεις τρόποι προστασίας, οι οποίοι θα πρέπει να χρησιμοποιούνται σε συνδυασμό:

- Χρήση τείχους προστασίας (firewall)
- Χρήση λογισμικού προστασίας ενάντια σε ιούς και προγράμματα κατασκοπείας (spyware).
- Συνεχής ενημέρωση των χρηστών.

Οι πλέον συνηθισμένοι τρόποι εξαπάτησης των χρηστών είναι το **Phishing** και το **Pharming**:

Phishing

Το phishing είναι η πράξη με την οποία κάποιος προσπαθεί να αποκτήσει πληροφορίες, όπως ονόματα χρηστών, κωδικούς πρόσβασης καθώς και στοιχεία πιστωτικών καρτών (και συνήθως άμεσα ή έμμεσα χρήματα), αφού έχει μεταμφιεστεί σε μια αξιόπιστη οντότητα μιας ηλεκτρονικής επικοινωνίας.

Οι ανακοινώσεις που υποτίθεται ότι είναι από δημοφιλείς κοινωνικές ιστοσελίδες, ιστοσελίδες δημοπρασιών, απευθείας σύνδεση επεξεργαστών πληρωμής ή διαχειριστές, συνήθως χρησιμοποιούνται για να δελεάσουν το ανυποψίαστο κοινό. Phishing emails μπορεί να περιέχουν συνδέσμους προς ιστοσελίδες που έχουν μολυνθεί με κακόβουλο λογισμικό.

Το phishing συνήθως εκτελείται από πλαστογραφημένο e-mail ή instant messaging, και συχνά κατευθύνει τους χρήστες να εισάγουν τα στοιχεία σε μια πλαστή ιστοσελίδα, η εμφάνιση και η αίσθηση της οποίας είναι σχεδόν πανομοιότυπη με τη νόμιμη. Το phishing είναι ένα από τα παραδείγματα της τεχνικής κοινωνικής μηχανικής που χρησιμοποιούνται για να εξαπατήσουν τους χρήστες, και εκμεταλλεύεται την κακή χρηστικότητα των σημερινών τεχνολογιών ασφαλείας web. Οι προσπάθειες για την αντιμετώπιση του αυξανόμενου αριθμού των αναφερόμενων περιστατικών phishing περιλαμβάνουν τη νομοθεσία, την εκπαίδευση των χρηστών, την ευαισθητοποίηση του κοινού και τεχνικά μέτρα ασφαλείας.

Μια τεχνική phishing περιγράφεται με λεπτομέρεια το 1987 και (σύμφωνα με τον δημιουργό του) η πρώτη καταγεγραμμένη χρήση του όρου «phishing» έγινε το 1995. Ο όρος είναι μια παραλλαγή της αλιείας, πιθανώς επηρεασμένος από phreaking και παραπέμπει στο «δόλωμα». Χρησιμοποιείται με την ελπίδα ότι το ενδεχόμενο θύμα θα "δαγκώσει" κάνοντας κλικ σε ένα κακόβουλο link ή το άνοιγμα ενός κακόβουλου αρχείου, προγραμματισμένου να αντιγράψει οικονομικά στοιχεία ή και κωδικούς πρόσβασης.

Η μέθοδος "phishing" συχνά συνδέεται με την αθέμιτη απόκτηση δεδομένων ή τη διάπραξη απάτης στο Διαδίκτυο. Η έκφραση "phishing" προέρχεται από την συνήθεια των hackers να χαρακτηρίζουν τους ηλεκτρονικούς τόπους στους οποίους έχουν πρόσβαση "phish".

Ειδικότερα, ως "phishing" χαρακτηρίζεται η αποστολή ηλεκτρονικών μηνυμάτων (e-mails) που σκοπό έχουν να προκαλέσουν την κλοπή εμπιστευτικών στοιχείων που ανήκουν στον παραλήπτη του ηλεκτρονικού μηνύματος. Τα ηλεκτρονικά αυτά μηνύματα δίνουν την εντύπωση πως προέρχονται από κάποια τράπεζα και ζητούν από τον παραλήπτη με διάφορες δικαιολογίες και προφάσεις την αποκάλυψη ευαίσθητων δεδομένων, π.χ. τον αριθμό τραπεζικού λογαριασμού του, τον προσωπικό αριθμό αναγνώρισης (PIN). Αν ο ανυποψίαστος παραλήπτης αποκαλύψει τις πληροφορίες

αυτές, οι δράστες (phishers) “εισβάλλουν” άμεσα στο λογαριασμό του και, αφού μεταφέρουν χρήματα από αυτόν τον λογαριασμό σε άλλον, τον αδειάζουν.

Επειδή η μέθοδος “phishing” βασίζεται στην πλάνη του θύματος με σκοπό την περιουσιακή του ζημία, είναι προφανές ότι οι Phishers μέσω αυτής προσπορίζουν στον εαυτό τους ή/και σε τρίτους παράνομο περιουσιακό όφελος. Επειδή δε οι δράστες έχουν γνώση και θέληση σχετικά με την παράνομη δραστηριότητά τους, συμπεραίνεται ότι το “phishing” συνιστά απάτη, κατά το άρθρο 386 του Ποινικού Κώδικα, σύμφωνα με το οποίο «όποιος με σκοπό να αποκομίσει ο ίδιος ή άλλος παράνομο περιουσιακό όφελος βλάπτει ξένη περιουσία πείθοντας κάποιον σε πράξη, παράλειψη ή ανοχή με την εν γνώσει παράσταση ψευδών γεγονότων ως αληθινών ή την αθέμιτη απόκρυψη ή παρασιώπηση αληθινών γεγονότων τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών και αν η ζημία που προξενήθηκε είναι ιδιαίτερα μεγάλη, με φυλάκιση τουλάχιστον δύο ετών».

Pharming

Η τεχνική του “pharming” αποτελεί μέθοδο εξαπάτησης μέσω του Διαδικτύου παρόμοια με το “phishing” αλλά σαφώς πιο επικίνδυνη από αυτό. Ένα ειδικό πρόγραμμα εκμεταλλεύεται κενά ασφαλείας του συστήματος, διεισδύει στον υπολογιστή του θύματος και το επηρεάζει κατά τέτοιο τρόπο, ώστε, ακόμη και αν ο χρήστης πληκτρολογεί τη σωστή διεύθυνση του διαδικτυακού τόπου που θέλει να επισκεφτεί, θεωρώντας πως βρίσκεται σε ασφαλή χώρο, ο συγκεκριμένος υπολογιστής τον “οδηγεί” μόνο σε πλαστές ιστοσελίδες. Ειδικότερα, αν πρόκειται για ιστοσελίδα τράπεζας, η προσπάθεια του θύματος να πραγματοποιήσει τις συναλλαγές του μέσω on-line banking καταλήγει στη μεταφορά των χρημάτων του στους δράστες (pharmers).

Είναι σαφές ότι η αύξηση των ωρών χρήσης του Διαδικτύου πολλαπλασιάζει τον κίνδυνο εγκατάστασης προγραμμάτων που καθιστούν δυνατό το “pharming”, το οποίο βαθμιαία εξελίσσεται σε μία από τις σοβαρότερες μορφές εγκληματικότητας στο Διαδίκτυο.

Η μέθοδος “pharming” αποτελεί ένα είδος διείσδυσης μέσω του Διαδικτύου, χωρίς τη συναίνεση του νόμιμου κατόχου των στοιχείων. Συνεπώς, η μέθοδος αυτή, εφόσον είναι ολοφάνερο ότι τελείται με δόλο, συνιστά παραβίαση απορρήτου κατά το άρθρο 370Γ § 2 του Ποινικού Κώδικα, σύμφωνα με το οποίο «όποιος αποκτά πρόσβαση σε στοιχεία που έχουν εισαχθεί σε υπολογιστή ή σε περιφερειακή μνήμη υπολογιστή ή μεταδίδονται με συστήματα τηλεπικοινωνιών, εφόσον οι πράξεις αυτές έγιναν χωρίς δικαίωμα, ιδίως με παραβίαση απαγορεύσεων ή μέτρων ασφαλείας που είχε λάβει ο νόμιμος κάτοχός τους, τιμωρείται με φυλάκιση μέχρι τρεις μήνες ή με χρηματική ποινή τουλάχιστον 29,00 € (...)».

Συμπερασματικά, οι ανωτέρω δύο μέθοδοι μπορούν να τιμωρηθούν, σύμφωνα με τις ισχύουσες διατάξεις του Ποινικού Κώδικα. Για την αντιμετώπιση τέτοιων φαινομένων κρίνεται απαραίτητη η λήψη τεχνικών μέτρων ασφαλείας, καθώς και η ευαισθητοποίηση των χρηστών του Ίντερνετ, ώστε να μην γίνονται εύκολα θύματα των phishers και των pharmers.

1.3 Διαδικτυακά πρωτόκολλα και υπηρεσίες

Τα συνηθέστερα διαδικτυακά πρωτόκολλα είναι τα TCP, IP, UDP, FTP, IMAP, POP3, SMTP, SMTP, DNS, HTTP/HTTPS, SSL, SSH. Ακόμα, μερικές από τις πιο γνωστές διαδικτυακές υπηρεσίες που κάνουν χρήση των πρωτοκόλλων αυτών είναι το ηλεκτρονικό ταχυδρομείο (e-mail), οι ομάδες συζητήσεων (newsgroups), η δια-μοίραση αρχείων (file sharing), η μεταφορά αρχείων (file transfer), ο Παγκόσμιος Ιστός (World Wide Web) και η ροή μέσων σε πραγματικό χρόνο (streaming media)

και η τηλεφωνία μέσω IP (voice telephony – VoIP). Από τις παραπάνω υπηρεσίες, το ηλεκτρονικό ταχυδρομείο και ο παγκόσμιος ιστός είναι οι πιο ευρέως χρησιμοποιημένες ενώ πολλές άλλες έχουν βασιστεί πάνω σε αυτές.

2.4 Υπηρεσίες Διαδικτύου

Οι περισσότερο δημοφιλείς και διαδεδομένες υπηρεσίες εφαρμογών του Διαδικτύου είναι οι παρακάτω:

- Παγκόσμιος Ιστός ή Ιστός (World Wide Web). Ο Ιστός παρέχει τη δυνατότητα στους χρήστες να εμφανίζουν έγγραφα που περιέχουν κείμενο και γραφικά, καθώς και να ακολουθούν υπερσυνδέσμους για να κινηθούν μεταξύ των εγγράφων. Ο Ιστός αναπτύχθηκε σε τέτοιο βαθμό που έγινε η μεγαλύτερη πηγή διακίνησης πληροφοριών στο παγκόσμιο Internet, και εξακολουθεί να κατέχει κυρίαρχη θέση.
- Ηλεκτρονικό ταχυδρομείο (e-mail). Το ηλεκτρονικό ταχυδρομείο δίνει τη δυνατότητα στον χρήστη να συνθέσει ένα μήνυμα και να στείλει αντίγραφό του σε μεμονωμένα άτομα ή ομάδες ατόμων. Ένα άλλο τμήμα δίνει τη δυνατότητα στους χρήστες να διαβάσουν τα μηνύματα που έχουν λάβει.
- Μεταφορά αρχείων (File transfer). Η εφαρμογή μεταφοράς αρχείων παρέχει τη δυνατότητα στο χρήστη να στέλνει και να λαμβάνει αντίγραφα αρχείων δεδομένων. Η μεταφορά αρχείων είναι μια από τις πιο ευρέως χρησιμοποιούμενες, υπηρεσίες εφαρμογών στο Internet. Παρά το ότι τα μικρά αρχεία μπορούν πλέον να επισυνάπτονται σε ένα μήνυμα ηλεκτρονικού ταχυδρομείου, η υπηρεσία μεταφοράς αρχείων εξακολουθεί να είναι απαραίτητη για τη διαχείριση μεγάλων αρχείων.
- Τηλεσύνδεση (Remote login). Η τηλεσύνδεση παρέχει τη δυνατότητα σε έναν χρήστη να συνδεθεί με μια απομακρυσμένη μηχανή και να δημιουργήσει αλληλεπιδραστική ή περίοδο συνδεδεμένης λειτουργίας (login session). Μέσω της τηλεσύνδεσης, η οθόνη του χρήστη συνδέεται κατευθείαν με την απομακρυσμένη μηχανή: κάθε πληκτρολόγηση του χρήστη μεταφέρεται στην απομακρυσμένη μηχανή και, συνεπώς, κάθε χαρακτήρας που στέλνει η απομακρυσμένη μηχανή εμφανίζεται στην οθόνη του χρήστη. Μόλις τερματίσει η περίοδος τηλεσύνδεσης, η εφαρμογή επαναφέρει τον χρήστη στο τοπικό σύστημα.

2.5 Ιστοσελίδα (web page)

Η ιστοσελίδα είναι ένα αρχείο που εμπεριέχει πληροφορίες, οι οποίες προορίζονται για δημοσίευση στον Παγκόσμιο Ιστό. Οι πληροφορίες του Παγκόσμιου Ιστού εμφανίζονται μορφοποιημένες με τη γλώσσα HTML (Hypertext Markup Language) σε μορφή ιστοσελίδων (web pages) και με την κατάληξη .htm ή .html. Υπάρχουν όμως και διαφορετικές μορφοποιήσεις ιστοσελίδων, όπως για παράδειγμα .aspx. Οι ιστοσελίδες είναι εφικτό να περιέχουν εκτός από στατικό κείμενο, εικόνες, video, ήχο, κινούμενες εικόνες (animation), δυναμικό κείμενο κτλ. Ένα πλεονέκτημα τους είναι ότι αλληλοσυνδέονται μεταξύ τους, με αποτέλεσμα ο χρήστης να μπορεί να μεταβεί με ένα κλικ από την μια ιστοσελίδα στην άλλη.

Η κατασκευή ιστοσελίδων είναι κάτι που μπορεί να πραγματοποιηθεί εύκολα στις μέρες μας καθώς υπάρχουν αρκετά ελεύθερα προγράμματα καθώς και αυτοματοποιημένοι μηχανισμοί ιστοσελίδων που επιτρέπουν σε απλούς χρήστες εύκολα και γρήγορα να δημιουργήσουν την δική τους προσωπική ιστοσελίδα. Από την άλλη μεριά υπάρχουν και πολλές εταιρείες, που ειδικεύονται στη δημιουργία ελκυστικών και λειτουργικών ιστοσελίδων που έχουν σαν κύριο στόχο να οδηγήσουν τους επισκέπτες στην αγορά κάποιου προϊόντος, στην επικοινωνία με τον ιδιοκτήτη του ιστοτόπου ή απλά στο ανέβασμα του εταιρικού προφίλ μιας επιχείρησης.

2.6 Κατηγορίες ιστοσελίδων

Οι δύο βασικές κατηγορίες των ιστοσελίδων είναι οι δυναμικές (Dynamic) και οι στατικές (Static) ιστοσελίδες. Θα αναφερθούμε σε αυτές αναλυτικότερα με την κάθε μια ξεχωριστά και θα δούμε τις βασικές διαφορές τους.

2.6.1 Στατική ιστοσελίδα

Στατική ιστοσελίδα (Static Web Page) ονομάζεται μια ιστοσελίδα της οποίας το περιεχόμενο μεταφέρεται στον χρήστη ακριβώς στην μορφή που είναι αποθηκευμένο στον εξυπηρετητή ιστοσελίδων (Web Server). Συνεπώς, τα στατικά websites είναι αυτά που δεν ανανεώνονται συχνά και δεν προσφέρουν υπηρεσίες αλληλεπίδρασης με τους χρήστες. Οι στατικές ιστοσελίδες χρησιμοποιούν το πρωτόκολλο HTTP κατα κύριο λόγο για να μεταφέρονται και αποθηκεύονται σε μορφή HTML.

Βασικό πλεονέκτημα των στατικών ιστοσελίδων είναι ότι δεν χρειάζονται ιδιαίτερες γνώσεις και προγραμματιστικές δεξιότητες για να τις δημιουργήσεις καθώς και ότι η σελίδα μπορεί να διατίθεται στον φυλλομετρητή χωρίς να είναι απαραίτητη η μεσολάβηση κάποιου εξυπηρετητή ιστοσελίδων. Αυτό μπορεί να συμβεί με την χρήση ενός αποθηκευτικού μέσου όπως ένα CD-ROM ή USB. Επίσης, είναι εφικτό να κλωνοποιηθεί σε περισσότερους από έναν εξυπηρετητές. Στα αρνητικά των στατικών ιστοσελίδων παρουσιάζεται η άβολη χρήση τους απέναντι στον προγραμματιστή ή τον χρήστη καθώς και η δύσκολη διαχείρισή τους όταν ο αριθμός των ιστοσελίδων είναι πολύ μεγάλος. Σε αυτή την περίπτωση απαιτούνται αυτόματα εργαλεία διαχείρισης.

2.6.2 Δυναμική ιστοσελίδα

Δυναμική ιστοσελίδα (Dynamic Web Page) είναι μια ιστοσελίδα η οποία δημιουργείται δυναμικά την στιγμή της πρόσβασης σε αυτή ή την στιγμή που ο χρήστης αρχίζει να αλληλεπιδρά με τον εξυπηρετητή ιστοσελίδων. Με απλά λόγια, με τις δυναμικές ιστοσελίδες μπορούν συγκεκριμένοι

χρήστες (διαχειριστές) να προχωρούν σε αλλαγές εύκολα και γρήγορα χρησιμοποιώντας έναν πίνακα διαχείρισης στον οποίο έχουν αποκλειστική πρόσβαση. Η έννοια “δυναμική” πηγάζει από την τεχνοτροπία με την οποία είναι κατασκευασμένη η ιστοσελίδα. Δυναμικά websites ονομάζονται τα ηλεκτρονικά καταστήματα, τα συστήματα διαχείρισης περιεχομένου (CMS), τα forums, τα blogs κ.ά.

Βασικά πλεονεκτήματα της δυναμικής ιστοσελίδας είναι ότι έχουν την δυνατότητα άμεσης επέμβασης και τροποποίησης του περιεχομένου της από τον διαχειριστή της και επίσης ότι δεν υπάρχουν σχέσεις άμεσης εξάρτησης με κατασκευαστές και εταιρείες κατασκευής ιστοσελίδων. Δεν υπάρχουν περιορισμοί στον όγκο που μπορεί να αποκτήσει ένας ιστότοπος, η εξοικονόμηση πόρων και χρημάτων ακόμη είναι πολύ σημαντικός παράγοντας της χρήσης τους και τέλος υπάρχει διαδεδομένη τεχνογνωσία σε παγκόσμιο επίπεδο. Το βασικό τους μειονέκτημα είναι ότι υπάρχει μεγάλη εξάρτηση λειτουργίας της ιστοσελίδας από πλήθος ιδιοτήτων του διακομιστή στον οποίο πραγματοποιείται η φιλοξενία της ιστοσελίδας. Επίσης, δυσκολότερη είναι και η αντιμετώπιση προβλημάτων καθώς και τεχνικών δυσκολιών που προκύπτουν.

2.6.3 Διαφορές στατικής και δυναμικής ιστοσελίδας

Η κυριότερη διαφορά των δυναμικών ιστοσελίδων έναντι των στατικών είναι ότι στις δυναμικές είναι απεριόριστο το περιεχόμενο των πληροφοριών που μπορεί κάποιος να αποθηκεύσει ενώ αντιθέτως στις στατικές το περιεχόμενο είναι πολύ περιορισμένο. Επίσης, όπως προαναφέρθηκε, οι στατικές ιστοσελίδες παρουσιάζουν ελάχιστη ακόμα και καμία αλληλεπίδραση με τους χρήστες ενώ στις δυναμικές ιστοσελίδες η αλληλεπίδραση με τους χρήστες είναι δεδομένη.

Σε αντίθεση με τις στατικές, οι οποίες είναι αρχεία “ανεβασμένα” σε έναν διακομιστή (server) οι δυναμικές ιστοσελίδες είναι ολόκληρες εφαρμογές που απαιτούν μια βάση δεδομένων για να λειτουργήσουν. Το κυριότερο πλεονέκτημα των δυναμικών ιστοσελίδων έναντι των στατικών, είναι σε ότι αφορά το κόστος για την αλλαγή περιεχομένου, καθώς στις πρώτες το κόστος είναι μηδενικό ενώ στις δεύτερες είναι αρκετά υψηλότερο. Σε ότι αφορά το κόστος κατασκευής όμως, τα πράγματα εδώ αντιστρέφονται, με τις στατικές ιστοσελίδες να υπερτερούν έναντι των δυναμικών αφού το αρχικό κόστος κατασκευής είναι κατά πολύ μικρότερο. Τέλος, σχετικά με την εξέλιξη οι στατικές ιστοσελίδες παύουν να εξελίσσονται ενώ σε αντίθεση, οι δυναμικές εξελίσσονται μέρα με την μέρα.

2.7 Διακομιστής Ιστού (Web Server)

Κάθε ιστοσελίδα βρίσκεται με τη μορφή αρχείου σε κάποιον διακομιστή Ιστού (web Server). Οι διακομιστές Ιστού είναι ειδικοί υπολογιστές με ειδικό λογισμικό και κατάλληλες δικτυακές συνδέσεις, οι οποίοι επιτρέπουν τη διάθεση των ιστοσελίδων σε ολόκληρο τον κόσμο. Ο Web Server, ουσιαστικά είναι ο υπολογιστής εκείνος ο οποίος αναλαμβάνει τη δημοσίευση της ιστοσελίδας στο Διαδίκτυο. Ο χρήστης του Διαδικτύου που θέλει να δει μια ιστοσελίδα, τη ζητάει από τον διακομιστή Ιστού στον οποίο αυτή βρίσκεται, και ο διακομιστής Ιστού με τη σειρά του την στέλνει. Ο server πρέπει απαραίτητα να βρίσκεται σε συνεχή διαθεσιμότητα, ώστε να είναι εφικτό να ανταποκρίνεται στις κλήσεις και να παρέχει τις ιστοσελίδες.

Το λογισμικό το οποίο τρέχει σε έναν κόμβο του Διαδικτύου ή στον υπολογιστή όταν εκτελεί προγράμματα εξυπηρετητές συνεχόμενα και για πολλές ώρες, ονομάζεται διακομιστής ή εξυπηρετητής. Οι εξυπηρετήσεις αυτές αφορούν άλλα προγράμματα τα οποία ονομάζονται πελάτες

(clients). Ο δημοφιλέστερος web server είναι ο Apache, τον οποίο θα δούμε αναλυτικά στην συνέχεια.

Οι εξυπηρετητές-προγράμματα που συναντάμε συχνότερα σε περιβάλλον γραφείου είναι οι εξής:

- **Εξυπηρετητής αρχείων** (file server)
- **Εξυπηρετητής αντιγράφων ασφαλείας** (backup server)
- **Εξυπηρετητής βάσεων δεδομένων** (database server)
- **Εξυπηρετητής διαμεσολαβητή** (proxy server)
- **Εξυπηρετητής ηλεκτρονικού ταχυδρομείου** (mail server)
- **Εξυπηρετητής φαξ** (fax server)
- **Εξυπηρετητής εκτυπωτών** (printer server)

Οι εξυπηρετητές-προγράμματα που συναντάμε συχνότερα στο Διαδίκτυο είναι οι εξής:

- **Εξυπηρετητής παγκόσμιου ιστού με το πρωτόκολλο http** (http server)
- **Εξυπηρετητής ηλεκτρονικού ταχυδρομείου** (mail server)
- **Domain Name System** (DNS server)
- **Εξυπηρετητής μεταφοράς αρχείων με το πρωτόκολλο ftp** (ftp server)

2.8 Φυλλομετρητής (web browser)

Ένας φυλλομετρητής ιστοσελίδων ή περιηγητής ιστού είναι ένα λογισμικό το οποίο επικοινωνεί με τους διακομιστές ιστού (web servers) μέσω του πρωτοκόλλου HTTP. Χαρακτηριστικά, αλληλεπιδρά με κείμενα, εικόνες, βίντεο και πληθώρα πληροφοριών που βρίσκονται συνήθως αναρτημένες σε μια ιστοσελίδα ενός ιστότοπου στον παγκόσμιο ιστό ή σε ένα τοπικό δίκτυο. Ο Web browser επιτρέπει στον χρήστη την γρήγορη και εύκολη πρόσβαση σε διάφορες ιστοσελίδες και ιστότοπους εναλλάσσοντας τις ιστοσελίδες διαμέσου υπερσυνδέσμων. Τα κείμενα, οι εικόνες, τα βίντεο μπορεί να περιέχουν υπερσυνδέσμους προς άλλες ιστοσελίδες. Οι φυλλομετρητές χρησιμοποιούν τη γλώσσα μορφοποίησης HTML για την προβολή των ιστοσελίδων.

Κάποιοι από τους δημοφιλέστερους Web browsers είναι οι:

- Windows Internet Explorer
- Mozilla Firefox
- Opera
- Apple Safari
- Google Chrome

Για κάθε φυλλομετρητή είναι διαθέσιμα είτε δωρεάν είτε επί πληρωμή, αρκετά πρόσθετα στοιχεία (add-ons, plug-ins) που βοηθούν στην αύξηση των δυνατοτήτων τους, την βελτίωση όσον αφορά την χρηστικότητά τους καθώς και την προστασία των πελατών ακόμα και σε θέματα ασφαλείας.

2.9 Όνομα τομέα / περιοχής (domain name)

Όνομα χώρου ή τομέα ή περιοχής (domain name) στο Διαδίκτυο είναι ένας περιορισμένος τομέας των διεθνών πόρων του Συστήματος Ονομάτων Χώρου (DNS) ο οποίος εκχωρείται για αποκλειστική χρήση σε ένα φυσικό ή νομικό πρόσωπο. Το όνομα τομέα / χώρου δεν ανήκει στο πρόσωπο που του έχει εκχωρηθεί αλλά έχει μόνο την αποκλειστική δυνατότητα χρήσης του για όσο διάστημα έχει καταβάλει τα τέλη κατοχύρωσης. Ένα όνομα χώρου είναι εφικτό να έχει διάφορες κατάληξεις όπως “.com”, “.eu”, “.gr”, “.net”, “.org”, “.info”, “.biz”, “.de”, “.it”, “.es” κ.λ.π., ανάλογα με τη χρήση και τη χώρα προέλευσής του.

Στα ονόματα χώρου επιτρέπεται αποκλειστικά η χρήση αλφαριθμητικών στοιχείων και παυλών. Για τα ονόματα χώρου με κατάληξη “.gr” υπάρχουν απαγορευμένες κατηγορίες. Αν ένα όνομα χώρου θεωρείται κοινόχρηστο ή γεωγραφικός όρος εκχωρείται μόνο στους αντίστοιχους οργανισμούς τοπικής αυτοδιοίκησης ανεξάρτητα από τον τρόπο γραφής του με λατινικά στοιχεία. Επιπλέον, δεν είναι εφικτή η εκχώρηση ονομάτων χώρου με κατάληξη .gr που αποτελούν λέξεις κλειδιά στο Διαδίκτυο.

Τα κατοχυρωμένα ονόματα χώρου είναι συνήθως τα ονόματα των τριών ή τεσσάρων πρώτων επιπέδων. Τα υπόλοιπα ονόματα χώρου δεν απαιτούν κατοχύρωση. Στα ονόματα χώρου κάθε τελεία δείχνει την αλλαγή επίπεδου ή αρχή ενός υποσυνόλου - υποτομέα και το σύνολο - χώρος που περιλαμβάνει όλα τα σύνολα είναι η πιο δεξιά τελεία που συνήθως παραλείπεται. Οι λύτες είναι το λογισμικό που μας συμβάλλει έτσι ώστε να χρησιμοποιήσουμε τα ονόματα χώρου. Οι λύτες διαβάζουν τα ονόματα του DNS από δεξιά προς τα αριστερά.

2.10 Φιλοξενία Ιστοσελίδων (web hosting)

Όταν μια ιστοσελίδα θα πρέπει να βρίσκεται συνεχώς αναρτημένη στο Διαδίκτυο θα πρέπει πρώτα να αποθηκευτεί σε ένα Web server για να χρησιμοποιηθεί. Για αυτό υπάρχουν οι διαδικτυακές υπηρεσίες που επιτρέπουν σε εταιρείες και ιδιώτες να διαθέτουν μια ιστοσελίδα αναρτημένη στο Διαδίκτυο συνεχώς και με την απαιτούμενη ασφάλεια. Γιατί θα ήταν αδύνατο ο κάθε χρήστης ξεχωριστά να διαθέτει τον απαιτούμενο εξοπλισμό. Η φιλοξενία ιστοσελίδων λοιπόν, είναι αυτή η διαδικτυακή υπηρεσία που επιτρέπει όλο αυτό χωρίς επιβαρύνσεις με το κόστος του ανάλογου εξοπλισμού.

Με πιο απλά λόγια μπορούμε να πούμε πως ο όρος Web Hosting είναι η ενοικίαση του χώρου στον ιδιοκτήτη μιας ιστοσελίδας σε υπολογιστές (διακομιστές) για να τοποθετήσει τα αρχεία του. Κάθε εταιρία ή ιδιώτης που παρέχουν φιλοξενία σε ιδιοκτήτες ιστοσελίδων διαθέτει τα δικά της πακέτα hosting με τιμές, ταχύτητες, διάρκεια και χαρακτηριστικά ανάλογα με τα θέλω του πελάτη. Οπότε σύμφωνα με την χρήση ή την κίνηση και την δημοφιλία της ιστοσελίδας του κάθε ενδιαφερόμενου επιλέγει το πακέτο που ταιριάζει καλύτερα σε αυτόν.

Τα πακέτα φιλοξενίας όπως κατηγοριοποιούνται στις παρακάτω οικογένειες:

- **Shared Hosting**, όπου παρέχεται μέρος του διακομιστή στον οποίο φιλοξενούνται και άλλοι χρήστες.
- **Reseller Hosting**, όπου υπάρχει η δυνατότητα της μεταπώλησης εργαλείων φιλοξενίας ιστοσελίδων και χώρου.

- **Cloud Hosting**, όπου τα εισερχόμενα αιτήματα εξυπηρέτησης μοιράζονται σε μηχανήματα που έχουν το μικρότερο φόρτο εργασίας με την χρήση της τεχνολογίας διαμοιρασμού φόρτου εργασίας σε πολλούς διακομιστές ταυτόχρονα, ενώ παράλληλα τα αντίγραφα σε κάθε server εξασφαλίζουν την διαθεσιμότητα και την ακεραιότητα των αρχείων.
- **Dedicated Servers**, όπου παρέχεται ολόκληρος ο διακομιστής για αποκλειστική χρήση και διαχείριση από τον κάτοχο του ιστότοπου.
- **Virtual Private Server**, όπου μέσω λογισμικού εικονικοποίησης διακομιστή (virtualization) παρέχεται στον διακομιστή ένας απομονωμένος χώρος με δικούς του –αποκλειστικής χρήσης– πόρους συστήματος και κεντρική πρόσβαση.

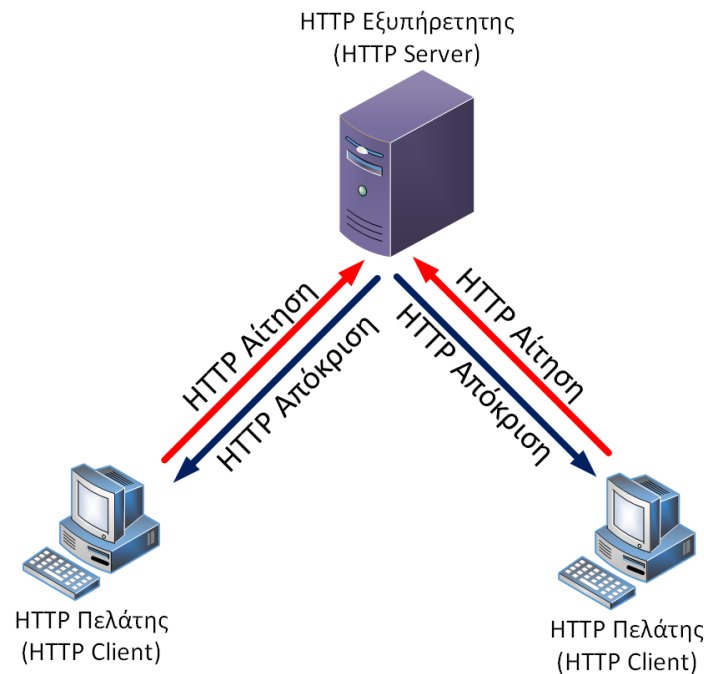
2.11 JSON (JavaScript Object Notation)

Το JSON είναι έγγραφο ή αρχείο που περιέχει δεδομένα σε μορφή κειμένου και χρησιμοποιείται για την εναλλαγή των δεδομένων μεταξύ συστημάτων. Η δομή του αποτελείται κυρίως από συλλογές ή λίστες ονομάτων και τιμών σε διάφορες γλώσσες που μπορούν να περιγράψουν κάποιο αντικείμενο, δομή, πίνακα ή κάποιο διάνυσμα, λίστα ή ακολουθία. Η μορφή JSON καθορίστηκε από τον Douglas Crockford.

2.11.1 Τι είναι το πρωτόκολλο HTTP

Το πρωτόκολλο μεταφοράς υπερκειμένου (HyperText Transfer Protocol - HTTP) βασίζεται στο μοντέλο πελάτη - εξυπηρετητή (client - server). Δηλαδή διαμοιράζεται σε δύο προγράμματα, που το ένα είναι του πελάτη (client) και το άλλο είναι του εξυπηρετητή (server). Τα δύο αυτά προγράμματα συνομιλούν μεταξύ τους ανταλλάσσοντας HTTP μηνύματα. Το HTTP ορίζει την δομή που θα έχουν αυτά τα μηνύματα. Οι περιηγητές (browsers) που «τρέχουν» σε έναν σταθμό εργασίας αναλαμβάνουν το ρόλο του HTTP πελάτη και την δουλειά του HTTP εξυπηρετητή αναλαμβάνει κάποιος απομακρυσμένος server στον οποίο απαιτεί πρόσβαση ο HTTP πελάτης. Συνήθως οι HTTP πελάτες που τρέχουν περιηγητές (browsers) είναι απλοί υπολογιστές οι οποίοι είτε βρίσκονται σε κάποια κατοικία είτε σε κάποιον οργανισμό (π.χ. εταιρίες, πανεπιστήμια, ιδρύματα έρευνας όπως το ΙΤΕ κλπ). Οι δε HTTP εξυπηρετητές όπως προαναφέραμε είναι απομακρυσμένοι servers οι οποίοι είναι πιθανό να βρίσκονται σε κέντρα πληροφοριών (data centers) όπου φιλοξενούν αντικείμενα τα οποία μπορεί να ζητήσει ένας HTTP πελάτης.

Για παράδειγμα: Μόλις ένας χρήστης ζητάει μια ιστοσελίδα πληκτρολογεί τον αντίστοιχο σύνδεσμο (URL). Ο browser στέλνει ένα μήνυμα HTTP αίτησης για το αντικείμενο που έχει ο εξυπηρετητής. Ο εξυπηρετητής λαμβάνει αυτήν την αίτηση και απαντά με HTTP μηνύματα απόκρισης τα οποία περιέχουν τα αντικείμενα. Το HTTP χρησιμοποιεί το πρωτόκολλο TCP και την θύρα επικοινωνίας 80. Υπάρχει επίσης και το HTTPS το οποίο προσφέρει ασφαλής επικοινωνία μεταξύ client και server χρησιμοποιώντας την θύρα 443.



Εικόνα 2.1: Παράδειγμα αίτησης και απόκρισης HTTP μηνυμάτων

Στο παραπάνω σχήμα έχουμε δύο HTTP πελάτες (HTTP clients) και ένα HTTP εξυπηρετητή (HTTP Server). Το πρωτόκολλο μεταφοράς που χρησιμοποιείται είναι το TCP διότι όπως αναφέραμε παραπάνω το HTTP υποστηρίζει το TCP ως πρωτόκολλο μεταφοράς δεδομένων. Θα αναφερθούμε εκτενέστερα γι' αυτό στο επίπεδο μεταφοράς που θα δούμε στο επόμενο κεφάλαιο. Προς το παρόν ας συγκρατήσουμε ότι το HTTP χρησιμοποιεί TCP επικοινωνία και ότι αυτό προσφέρει αξιόπιστη μεταφορά δεδομένων. Ο πελάτης HTTP λοιπόν είναι αυτός που ξεκινάει πρώτος την σύνδεση με τον HTTP εξυπηρετητή. Όταν πραγματοποιηθεί η σύνδεση οι διεργασίες του browser και του εξυπηρετητή προσπελούν το TCP μέσω των διεπαφών socket τους. Ο πελάτης στέλνει μήνυμα αιτήσεων HTTP στην διεπαφή socket και στη συνέχεια λαμβάνει μηνύματα απόκρισης HTTP από

την διεπαφή socket. Με πανομοιότυπο τρόπο ο HTTP εξυπηρετητής λαμβάνει μηνύματα αιτήσεως από την διεπαφή του socket του και στέλνει απαντητικά μηνύματα στην ίδια διεπαφή του.

Υπάρχουν δύο τύποι HTTP συνδέσεων:

- **HTTP με μη παραμένουσες συνδέσεις (Non persistent HTTP):** Όπου ένα και μόνο αντικείμενο μπορεί να σταλεί μόνο μέσω μιας σύνδεσης. Παράδειγμα το HTTP/1.0
- **HTTP με παραμένουσες συνδέσεις (persistent HTTP):** Πολλά αντικείμενα μπορούν να σταλούν μέσω μιας TCP σύνδεσης. Παράδειγμα το HTTP/1.1

2.11.2 Τι είναι το πρωτόκολλο HTTPS

Το HTTPS δεν αποτελεί ένα ξεχωριστό πρωτόκολλο αλλά έναν συνδυασμό του πρωτοκόλλου HTTP και των δυνατοτήτων κρυπτογράφησης που παρέχει το πρωτόκολλο Secure Sockets Layer (SSL). Η κρυπτογράφηση που χρησιμοποιείται εξασφαλίζει με τη σειρά του ότι τα κρυπτογραφημένα δεδομένα δεν θα είναι εφικτό να υποκλαπούν από άλλους κακόβουλους χρήστες. Για να χρησιμοποιηθεί το HTTPS σε κάποιο server θα πρέπει ο διαχειριστής του να δημιουργήσει ένα ζεύγος κλειδιών (δημόσιο/ιδιωτικό). Στην συνέχεια το δημόσιο κλειδί θα πρέπει να υπογραφεί από μία αρχή πιστοποίησης (certificate authority), η οποία εκδίδει ένα ψηφιακό πιστοποιητικό και πιστοποιεί ότι ο server που εμφανίζεται στο πιστοποιητικό είναι νόμιμος, όπως και ότι το πιστοποιητικό είναι έγκυρο.

Με τον τρόπο αυτό οι χρήστες είναι εφικτό να δουν την υπογραφή της αρχής πιστοποίησης και να βεβαιωθούν ότι το πιστοποιητικό είναι απολύτως έγκυρο και ότι κανένας κακόβουλος χρήστης δεν το έχει πλαστογραφήσει. Το HTTPS χρησιμοποιείται κυρίως σε περιπτώσεις που απαιτείται μεταφορά ευαίσθητων προσωπικών δεδομένων (π.χ. υπηρεσίες cloud service όπως dropbox.com). Το επίπεδο προστασίας δεδομένων εξαρτάται από το αν έχει εφαρμοστεί σωστά η διαδικασία ασφάλειας και από το πόσο ισχυροί είναι οι αλγόριθμοι κρυπτογράφησης που χρησιμοποιούνται.

Πολλοί χρήστες πιστωτικών καρτών θεωρούν ότι το HTTPS προστατεύει ολοκληρωτικά τα στοιχεία της πιστωτικής τους κάρτας από κατάχρηση. Το HTTPS χρησιμοποιεί την κρυπτογράφηση για να μεταδώσει τον αριθμό από τον υπολογιστή του πελάτη προς τον server. Η μετάδοση είναι απολύτως ασφαλής και τα δεδομένα φτάνουν στον server χωρίς κανείς να καταφέρει να τα υποκλέψει.

Οπότε, από τα παραπάνω συμπεραίνουμε ότι το HTTPS εγκαθιδρύει ένα ασφαλές κανάλι ανταλλαγής δεδομένων πάνω από ένα μη ασφαλές δίκτυο. Με αυτόν τον τρόπο καθορίζει ασφάλεια από συνακροάσεις και επιθέσεις man-in-the-middle παρέχοντας επαρκείς σουίτες κρυπτογράφησης (cipher suites) και ότι το πιστοποιητικό (certificate) του server είναι αξιόπιστο. Οπότε το HTTPS επί της ουσίας κρυπτογραφεί το ζητούμενο από τον client URL, τις παραμέτρους ουράς (queries), τις επικεφαλίδες (headers) και τα cookies (που συνήθως περιέχουν πληροφορίες για τον εντοπισμό του client). Παρόλα αυτά οι θύρες επικοινωνίας και η διεύθυνση του host δεν καταφέρνουν να προστατευτούν από το HTTPS. Οι περιηγητές διαδικτύου (web browsers) γνωρίζουν ποιοι HTTPS servers είναι αξιόπιστοι με βάση το πιστοποιητικό που διαθέτουν. Επομένως, ένας χρήστης μπορεί να εμπιστευτεί μια HTTPS σύνδεση όταν ισχύουν τα εξής:

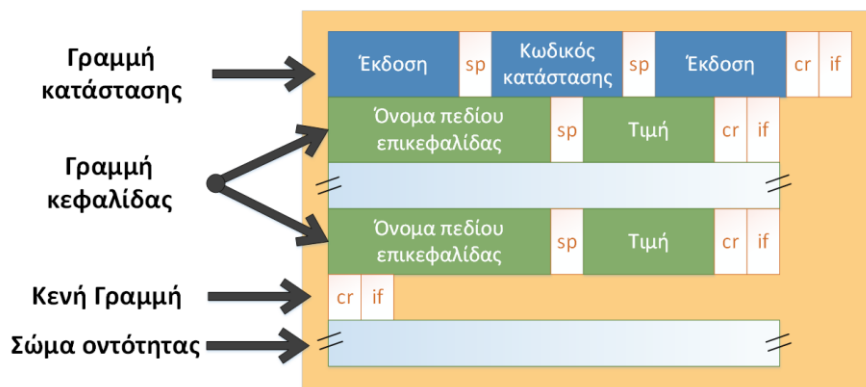
- Ο χρήστης εμπιστεύεται ότι ο browser του εκτελεί το HTTPS με σωστά προ εγκατεστημένα πιστοποιητικά εξουσιοδότησης
- Ο χρήστης εμπιστεύεται το πιστοποιητικό μιας σελίδας η οποία είναι νόμιμη
- Η ιστοσελίδα παρέχει ένα έγκυρο πιστοποιητικό, που σημαίνει ότι υπογράφηκε από έναν έμπιστο οργανισμό
- Το πιστοποιητικό εντοπίζει επιτυχώς την ιστοσελίδα
- Ο χρήστης εμπιστεύεται ότι το πρωτόκολλο κρυπτογράφησης (SSL/TSL) είναι επαρκώς ασφαλή ενάντια σε επιθέσεις από τρίτους.

Το HTTPS είναι ιδιαίτερα χρήσιμο για τα μη ασφαλή δίκτυα (όπως δημόσια WiFi access points - hotspots) καθότι κάποιος που έχει συνδεθεί στο ίδιο δίκτυο μπορεί να παρακολουθεί την κίνηση των πακέτων (packet sniff) με αποτέλεσμα να ανακαλύψει σημαντικές πληροφορίες αν και μόνο αν το HTTPS δεν χρησιμοποιείται. Επιπρόσθετα, πολλά ελεύθερα WLAN δίκτυα εφαρμόζουν packet injection προκειμένου να «σερβίρουν» στον client τις διαφημίσεις τους σε ιστοσελίδες που επισκεπτόμαστε. Αυτό μπορεί να το εκμεταλλευτούν αυτοί που θα κάνουν επίθεση σε ένα δίκτυο προσθέτοντας στο πακέτο αντί για τις διαφημίσεις κάποιο malware στις ιστοσελίδες και να υποκλέπτουν τα στοιχεία του εκάστοτε χρήστη. Το HTTPS είναι επίσης χρήσιμο σε συνδέσεις που παρέχουν ανωνυμία (όπως π.χ. Tor) καθότι κακόβολοι Tor κόμβοι μπορούν να καταστρέψουν ή να αλλοιώσουν το περιεχόμενο της πληροφορίας έτσι ώστε να «ρίξουν» malware στην Tor σύνδεση. Αυτός είναι ένας λόγος όπου το Electronic Frontier Foundation και η Tor project ξεκίνησαν να αναπτύσσουν την ιδέα ότι όλες οι ιστοσελίδες οφείλουν να χρησιμοποιούν το πρωτόκολλο HTTPS.

2.11.3 Δομή ενός HTTP μηνύματος απάντησης

Η γραμμή κεφαλίδας (header section) καθορίζει τον υπολογιστή στον οποίο βρίσκεται το αντικείμενο το οποίο ζητάμε. Το σώμα οντότητας είναι κενό με την τιμή GET αλλά χρησιμοποιείται όταν αποδίδεται στο πεδίο μεθόδου η τιμή POST. Προηγουμένως αναλύσαμε την δομή από την οποία αποτελείται ένα HTTP μήνυμα αίτησης. Τι γίνεται όμως στη περίπτωση που ο Server επιθυμεί να δώσει μια απάντηση στον HTTP client; Θα υπάρξουν διαφορές ως προς την δομή του πακέτου;

Ας δούμε σε αυτό το σημείο την μορφή ενός HTTP μηνύματος απάντησης. Έχει τρία σημεία τα οποία είναι μια γραμμή κατάστασης (status line), έξι γραμμές επικεφαλίδας και μετά το σώμα οντότητας. Το σώμα οντότητας αφορά το περιεχόμενο του μηνύματος. Η γραμμή κατάστασης περιέχει τρία πεδία, το πεδίο έκδοσης πρωτοκόλλου έναν κωδικό κατάστασης και ένα αντίστοιχο μήνυμα κατάστασης. Η εικόνα της παρακάτω διαφάνειας δίνει μια σαφέστατη απάντηση για την δομή ενός απαντητικού μηνύματος HTTP.



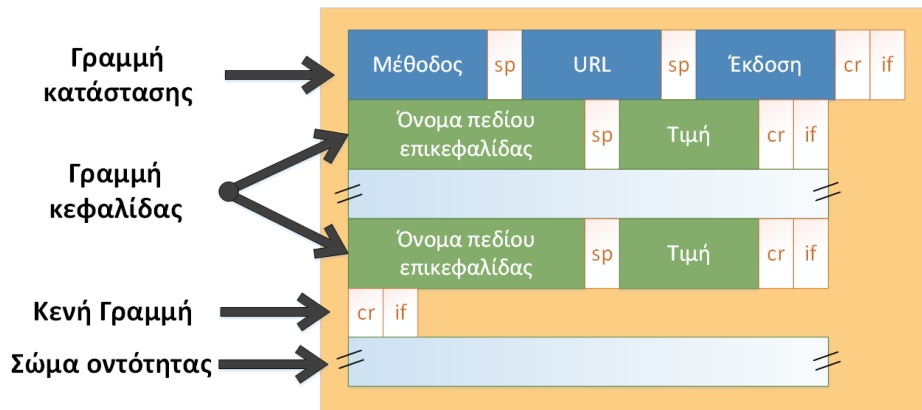
Εικόνα 2.3: HTTP μήνυμα απάντησης

Το πεδίο κωδικός κατάστασης είναι πιθανό να έχει διαφορετικές τιμές οι οποίες μπορεί να είναι 200, 301, 400, 404, 505 κλπ. Κάθε ένας απ' αυτούς τους κωδικούς ορίζει την απάντηση που θα δώσει στον client. Για παράδειγμα, έστω ότι ένας HTTP πελάτης (client) στέλνει μια αίτηση στον HTTP εξυπηρετητή (server). Όταν λάβει ο HTTP εξυπηρετητής το αίτημα απ' τον HTTP πελάτη (client) θα ελέγξει με την σειρά του αν όλα είναι εντάξει και αφού το επιβεβαιώσει θα στείλει πίσω στον HTTP πελάτη ένα HTTP απαντητικό μήνυμα με κωδικό 200 που σημαίνει OK.

2.11.4 Δομή ενός HTTP μηνύματος αίτησης

Ας αναλύσουμε λίγο την δομή ενός HTTP μηνύματος αίτησης. Η πρώτη γραμμή που καλείται γραμμή κατάστασης (status line) περιέχει τρία πεδία:

1. το πεδίο μεθόδου
2. το πεδίο URL
3. και το πεδίο έκδοσης HTTP



Εικόνα 2.2: HTTP μήνυμα αίτησης

Το πεδίο μέθοδος είναι πιθανό να έχει διαφορετικές τιμές οι οποίες μπορεί να είναι GET, POST, HEAD, PUT, DELETE κλπ. Η μέθοδος GET χρησιμοποιείται όταν ζητάμε ένα αντικείμενο ή μια πληροφορία από το URL. Στον πίνακα της παρακάτω διαφάνειας ακολουθεί η κάθε τιμή που μπορεί να πάρει το πεδίο μεθόδου.

1. **GET:** Η τιμή GET χρησιμοποιείται όταν θέλουμε να ανακτήσουμε μια πληροφορία που περιέχεται στο πεδίο URL (που είναι ένας server). Οι αιτήσεις που έχουν την τιμή GET μπορούν μόνο να ανακτήσουν δεδομένα.
2. **HEAD:** Λειτουργεί με τον ίδιο τρόπο όπως η GET, με την διαφορά ότι μεταφέρει μόνο την γραμμή κατάστασης status line και την γραμμή επικεφαλίδας (header section) στο πακέτο HTTP.
3. **POST:** Η τιμή POST χρησιμοποιείται σε HTTP μηνύματα αίτησης για να στείλουμε πληροφορία στον server, για παράδειγμα, τις πληροφορίες που θέλει να στείλει ένας πελάτης, ανέβασμα αρχείων, κλπ μέσω μιας φόρμας HTML.
4. **PUT:** Αντικαθιστά όλες τις τρέχουσες αναπαραστάσεις των πόρων προορισμού με το ανεβασμένο περιεχόμενο (uploaded content)
5. **DELETE:** Διαγράφει όλες τις τρέχουσες αναπαραστάσεις των πόρων προορισμού που έχουν δοθεί από ένα URL.
6. **CONNECT:** Εγκαθιδρύει ένα επικοινωνιακό κανάλι με τον εξυπηρετητή (server) το οποίο ταχτοποιείται από το δοθέν URL
7. **OPTIONS:** Περιγράφει όλες τις ιδιότητες επικοινωνίας για τους πόρους προορισμού
8. **TRACE:** Εκτελεί μια διαδικασία από τεστ βρόγχων επαναληπτικών μηνυμάτων (message loop-back test) σε όλη τη σύνδεση ως τους πόρους προορισμού.

2.11.5 URIs (Uniform Resource Identifiers)

Ενιαία Αναγνωριστικά Πόρων, (Uniform Resources Identifiers ή URIs) είναι σύντομες χορδές χαρακτήρων που χρησιμοποιούνται για τον εντοπισμό ονομάτων ή πόρων στο Διαδίκτυο: έγγραφα, εικόνες, αρχεία, υπηρεσίες, ηλεκτρονικά γραμματοκιβώτια, και άλλους πόρους. Καθιστούν τους πόρους διαθέσιμους κάτω από μια ποικιλία των συστημάτων και μεθόδων πρόσβασης, όπως HTTP, FTP, και το ηλεκτρονικό ταχυδρομείο και προσπελάσιμους με τον ίδιο απλό τρόπο ονοματοδοσίας.

2.11.6 SOAP (Simple Object Access Protocol)

SOAP ονομάζεται ένα πρωτόκολλο το οποίο βασίζεται σε XML αρχεία για την ανταλλαγή δεδομένων μέσω HTTP. Παρέχει απλές μεθόδους, που βασίζονται σε πρότυπα για την αποστολή μηνυμάτων σε μορφή XML μεταξύ των εφαρμογών σε μια ποικιλία τυποποιημένων τεχνολογιών Διαδικτύου, συμπεριλαμβανομένων SMTP, HTTP, και FTP. Οι υπηρεσίες Web χρησιμοποιούν το πρωτόκολλο SOAP για την αποστολή μηνυμάτων μεταξύ της υπηρεσίας και του πελάτη.

Επειδή το HTTP υποστηρίζεται από όλους τους διακομιστές και Web browsers, τα μηνύματα SOAP είναι εφικτό να σταλούν μεταξύ των εφαρμογών, ανεξάρτητα από πλατφόρμα ή γλώσσα προγραμματισμού. Τα δεδομένα αποστέλλονται μεταξύ του πελάτη και της υπηρεσίας Web χρησιμοποιώντας μηνύματα αιτήματος - απόκρισης SOAP, η μορφή των οποίων καθορίζεται από το WSDL. Επειδή ο πελάτης και ο διακομιστής τηρούν τη σύμβαση WSDL, κατά τη δημιουργία SOAP μηνυμάτων η συμβατότητα είναι εγγυημένη.

2.11.7 REST (Representational State Transfer)

Ο όρος Representational State Transfer (REST) επινοήθηκε από τον Roy Fielding με σκοπό να ορίσει μια αρχιτεκτονική που βασίζεται σε ένα σύνολο αρχών για το σχεδιασμό λογισμικού του διαδικτύου. Η REST ορίζει ένα σύνολο αρχιτεκτονικών αρχών για την σχεδίαση υπηρεσιών Web στο οποίο τα δεδομένα και η πληροφορία θεωρούνται πόροι του συστήματος και η πρόσβαση σε αυτούς δίνεται μέσω των Ενιαίων Αναγνωριστικών Πόρων (URIs), που αποτελούν συνδέσμους στο διαδίκτυο. Ουσιαστικά, ορίζει το πώς οι πόροι αντιμετωπίζονται και μεταφέρονται μέσω HTTP από ένα ευρύ φάσμα πελατών (clients) σε διαφορετικές γλώσσες.

Η βασική ιδέα σε κάθε REST API είναι ο πόρος. Ένας πόρος χαρακτηρίζεται ως ένα αντικείμενο που έχει έναν τύπο, συσχετιζόμενα δεδομένα, σχέσεις με άλλους πόρους, καθώς και μια σειρά από μεθόδους που λειτουργούν πάνω σε αυτό. Είναι παρόμοιο με ένα αντικείμενο σε μία αντικειμενοστραφή γλώσσα προγραμματισμού, με τη σημαντική διαφορά ότι μόνο μερικές πρότυπες μέθοδοι ορίζονται για τον πόρο (πρότυπο HTTP : GET, POST, PUT και DELETE μέθοδοι), ενώ ένα αντικείμενο συνήθως έχει πολλές μεθόδους. Οι πόροι μπορούν ακόμα και να ταξινομηθούν σε συλλογές - σορούς. Κάθε συλλογή είναι ομοιογενής, ώστε να περιέχει μόνο ένα είδος πόρων, χωρίς συγκεκριμένη τάξη.

Οι πόροι μπορούν επίσης, είναι εφικτό να υπάρχουν έξω από κάποια συλλογή. Κάθε αντικείμενο και πόρος στο σύστημα είναι προσβάσιμο μέσω ενός μοναδικού αναγνωριστικού. Στο REST αυτό επιτυγχάνεται με την χρήση των URIs. Όταν πραγματοποιείται μια αίτηση HTTP στον φυλλομετρητή πρέπει να περιέχει το URI του αντικειμένου που ζητείται από τον server ή τοποθετείται σε αυτόν.

Η αρχιτεκτονική REST παρουσιάζει τα ακόλουθα πλεονεκτήματα:

- Δεν υπάρχει ανάγκη για διατήρηση της κατάσταση των εφαρμογών, οδηγώντας σε απλοποιημένα και επεκτάσιμα συστατικά.
- Οι αιτήσεις μπορούν να υποβάλλονται σε επεξεργασία παράλληλα.
- Οι αιτήσεις μπορούν να γίνουν κατανοητές αν απομονωθούν, οδηγώντας στην απλουστευμένη οργάνωση και δυναμική αναδιοργάνωση των υπηρεσιών.
- Εξυπηρετεί στην καλή χρήση του HTTP cache και του διακομιστή μεσολάβησης (proxy server) που βοηθάει στον χειρισμό μεγαλύτερου φορτίου.

2.11.8 REST VS SOAP

Συγκρίνοντας το REST με το SOAP στην εφαρμογή τους σε διαδικτυακές υπηρεσίες καταλήγουμε στο εξής συμπέρασμα. Το REST είναι στο μεγαλύτερο μέρος του πιο εύκολο στην χρήση του και πιο ευέλικτο.

Παρουσιάζει τα παρακάτω πλεονεκτήματα συγκριτικά με το SOAP:

- Μικρότερη καμπύλη εκμάθησης.
- Καταλληλότερο για περιβάλλον σημείο-προς-σημείο ή όταν ο ενδιαμέσος δεν παίζει σημαντικό ρόλο.
- Δεν χρειάζονται ακριβά εργαλεία για να αλληλεπιδρούν με την διαδικτυακή υπηρεσία. • Είναι αποτελεσματικότερο διότι το SOAP χρησιμοποιεί XML αρχεία για όλα τα μηνύματα ενώ το REST μπορεί να χρησιμοποιήσει μικρότερα μηνύματα.
- Είναι πολύ γρήγορο διότι δεν απαιτείται εκτεταμένη επεξεργασία.
- Ελάχιστα εργαλεία και ενδιαμέσα πρόγραμμα απαιτούνται. Απαιτείται μόνο η υποστήριξη HTTP. Κωνσταντίνος Κοροβέσης, Χρυσόστομος Μανώλης 22 Σχεδιασμός και Ανάπτυξη Διαδικτυακών Υπηρεσιών Αρχιτεκτονικής REST
- Ενσωματωμένη διαχείριση σφαλμάτων.

Το REST παρουσιάζει μερικά μειονεκτήματα συγκριτικά με το SOAP. Το REST για να πέτυχει πρέπει να είναι απλό και γρήγορο, αλλά στερείται αξιοπιστίας καθώς είναι πιθανό για παράδειγμα κάποιες HTTP μέθοδοι όπως η DELETE να επιστρέψουν OK χωρίς οι πόροι να έχουν διαγραφεί. Επίσης το SOAP εμπνέει μεγαλύτερη ασφάλεια λόγω των πολλών προτύπων για την αξιοπιστία, την ασφάλεια και τις συναλλαγές.

Χρησιμοποιούμε τον όρο RESTful για να αναφερθούμε σε υπηρεσίες Web χτισμένες σύμφωνα με την αρχιτεκτονική REST. Η REST έχει γίνει τα τελευταία χρόνια ένα κυρίαρχο μοντέλο σχεδιασμού υπηρεσιών Web. Η αρχιτεκτονική REST είχε τρανό αντίκτυπο στο διαδίκτυο και έχει κυριαρχήσει έναντι υπηρεσιών γραμμένες με βάση τα SOAP και WSDL πρότυπα, καθώς είναι πολύ απλούστερη στη χρήση, παρέχει επεκτασιμότητα και κυριότερα, έχει καλύτερη απόδοση.

Κεφάλαιο 2ο: Τεχνολογίες, frameworks και libraries

3.1 Τεχνολογίες front-end

3.1.1 HTML / HTML5

Η HTML [1] ξεκίνησε να δημιουργείται στις αρχές της δεκαετίας του 1990 έως ένα σύντομο έγγραφο το οποίο παρέθετε κάποια στοιχεία όπου χρησιμοποιούνταν για την κατασκευή ιστοσελίδων. Πολλά από τα στοιχεία αυτά αφορούσαν την περιγραφή περιεχομένου ιστοσελίδων όπως οι επικεφαλίδες, οι παράγραφοι, οι αλλαγές γραμής και οι λίστες. Ο αριθμός έκδοσης της HTML αυξανόταν, καθώς η γλώσσα εξελισσόταν με την εισαγωγή νέων στοιχείων και διαφοροποιήσεων στους κανόνες της. Η τελευταία έκδοση της είναι η HTML5.



Η HTML5 αποτελεί φυσική εξέλιξη των προηγούμενων εκδόσεων της HTML και προσπαθεί να καλύψει τις ανάγκες όλων των υπαρχόντων και των μελλοντικών ιστότοπων. Κληρονομεί τη συντριπτική πλειονότητα των λειτουργιών από τους προκατόχους της, γεγονός που σημαίνει ότι αν κάποιος έχει γράψει κώδικα HTML πριν από την έλευση της HTML5, γνωρίζει ήδη αρκετά για την HTML5. Αυτό σημαίνει επίσης ότι μεγάλο μέρος της HTML5 λειτουργεί τόσο σε παλιούς όσο και σε νέους φυλλομετρητές. Η προς τα πίσω συμβατότητα αποτελεί βασική αρχή στον σχεδιασμό της HTML5.

Η HTML (HyperText Markup Language - Γλώσσα Σήμανσης Υπερκειμένου) είναι η κύρια γλώσσα σήμανσης για τις ιστοσελίδες, και τα στοιχεία της είναι τα βασικά δομικά στοιχεία των websites. Η HTML γράφεται υπό μορφή στοιχείων HTML τα οποία αποτελούνται από ετικέτες (tags) που περικλείονται μέσα σε σύμβολα «μεγαλύτερο από» και «μικρότερο από» (για παράδειγμα), μέσα στο περιεχόμενο της ιστοσελίδας. Οι ετικέτες HTML συνήθως λειτουργούν ανά ζεύγη (για παράδειγμα και), με την πρώτη να ονομάζεται ετικέτα έναρξης και τη δεύτερη ετικέτα λήξης (ή σε άλλες περιπτώσεις ετικέτα ανοίγματος και ετικέτα κλεισίματος αντίστοιχα). Ανάμεσα στις ετικέτες, οι σχεδιαστές ιστοσελίδων μπορούν να τοποθετήσουν κείμενο, πίνακες, εικόνες κλπ.

Ο σκοπός ενός πλοηγητή είναι να διαβάσει τα έγγραφα HTML και να τα συνθέσει σε σελίδες που μπορεί κανείς να διαβάσει ή να ακούσει. Ο browser δεν εμφανίζει τις ετικέτες HTML, αλλά τις χρησιμοποιεί για να ερμηνεύσει το περιεχόμενο της σελίδας.

Τα στοιχεία της HTML χρησιμοποιούνται για να κτίσουν όλους του ιστότοπους. Η HTML επιτρέπει την ενσωμάτωση εικόνων και άλλων αντικειμένων μέσα στη σελίδα, ενώ μπορεί να χρησιμοποιηθεί για να εμφανίσει διαδραστικές φόρμες. Παρέχει τις μεθόδους της δημιουργίας δομημένων εγγράφων (δηλαδή εγγράφων που αποτελούνται από το περιεχόμενο που μεταφέρουν και από τον κώδικα μορφοποίησης του περιεχομένου) καθορίζοντας δομικά σημαντικά στοιχεία για το κείμενο, όπως κεφαλίδες, παραγράφους, λίστες κ.α. Μπορούν επίσης να ενσωματώνονται σενάρια εντολών σε γλώσσες όπως η JavaScript, τα οποία επηρεάζουν τη συμπεριφορά των ιστοσελίδων HTML.

Οι πλοηγητές μπορούν επίσης να αναφέρονται σε στυλ μορφοποίησης CSS για να ορίζουν την εμφάνιση και τη διάταξη του κειμένου και του υπόλοιπου υλικού. Ο οργανισμός W3C, ο οποίος δημιουργεί και συντηρεί τα πρότυπα για την HTML και τα CSS, ενθαρρύνει τη χρήση των CSS αντί διαφόρων στοιχείων της HTML για σκοπούς παρουσίασης του περιεχομένου.

3.1.2 CSS

Το CSS [2] είναι ένα εργαλείο όπου χρησιμοποιούμε για την κατασκευή ενός web site με σκοπό την βελτίωση της ταχύτητας φόρτωσης αλλά και την μείωση του χώρου που καταλαμβάνει ένα website σε έναν server. Αυτό επιτυγχάνεται γιατί αυτού του είδους η σχεδιαστική προσέγγιση ενός web site συγκεντρώνει όλο τον κώδικα που γράφεται και αφορά την μορφή και την εμφάνιση μια σελίδας σε ένα αρχείο με κατάληξη .css το οποίο και συνδέεται με όλες τις σελίδες και υποσελίδες και τις τροφοδοτεί με τον αντίστοιχο κώδικα ανά πάσα στιγμή.

Οι μηχανές αναζήτησης έχουν μεγάλη αδυναμία στις ιστοσελίδες που είναι φτιαγμένες με αυτόν τον τρόπο και τα κατατάσσουν υψηλότερα από άλλες με αντίστοιχη σημασία. Ο τρόπος αυτός προσφέρει και συνοχή και ομοιομορφία ανάμεσα στις σελίδες των ιστοσελίδων που τον χρησιμοποιούν. Επίσης, μπορεί να χρησιμοποιηθεί για να διαμορφώσει τα χρώματα και το μέγεθος διαφόρων στοιχείων της ιστοσελίδας, την συμπεριφορά τους σε διάφορες ενέργειες και παλιότερα, συνηθίζονταν να διαμορφώνεται κάθε στοιχείο της ιστοσελίδας ξεχωριστά, σε κάθε μία ξεχωριστή σελίδα. Η μεγάλη ευκολία που προσφέρει η χρήση ενός CSS, είναι ότι οι ενδεχόμενες αλλαγές, γίνονται μόνο σε ένα ενιαίο έγγραφο και αυτόματα εφαρμόζονται σε όλες τις σελίδες που το χρησιμοποιούν. Έτσι μία ιστοσελίδα που χρησιμοποιεί CSS, μπορεί πολύ πιο εύκολα να αλλάξει όψη. Για τον έλεγχο της εγκυρότητας ενός CSS εγγράφου, με βάση τα web standards, μπορεί να χρησιμοποιηθεί ένας CSS validator. Για την δημιουργία και επεξεργασία εγγράφων CSS, μπορεί να χρησιμοποιηθεί ένας επεξεργαστής απλού κειμένου ή κάποιο πρόγραμμα ειδικό για CSS που προσφέρει αρκετές ευκολίες στην διαμόρφωση του κώδικα CSS. Μπορούν να οριστούν κατευθείαν σε μια HTML σελίδα ή σε ένα εξωτερικό αρχείο CSS. Τα αρχεία CSS έχουν την κατάληξη “.css” και γενικά τοποθετούνται στον ίδιο φάκελο μαζί με τις html σελίδες. Περιέχουν εντολές σε δομημένη μορφή. Η εισαγωγή του στην ιστοσελίδα γίνεται με την ετικέτα, η οποία τοποθετείται στην ετικέτα επικεφαλίδας. Τα φύλλα στυλ επιτρέπουν στους σχεδιαστές να ορίσουν μια φορά τα χαρακτηριστικά όπου θα εφαρμόσουν στην ιστοσελίδα τους και να καλούν όποτε θέλουν χωρίς να επαναλαμβάνουν κάθε φορά τις ετικέτες μορφοποίησης της HTML.

3.1.3 JavaScript

Η JavaScript (JS) [3] είναι μία διερμηνευόμενη γλώσσα προγραμματισμού για H/Y, Αρχικά, αποτέλεσε ένα μέρος της υλοποίησης των web browsers, ώστε τα σενάρια από την πλευρά του πελάτη (client-side) να μπορούν να επικοινωνούν με τον χρήστη, να ανταλλάσσουν δεδομένα ασύγχρονα και να αλλάζουν δυναμικά το περιεχόμενο του εγγράφου που εμφανίζεται.

Η JavaScript είναι μια γλώσσα scripting που βασίζεται στα πρωτότυπα (prototype-based), είναι δυναμική, με ασθενείς τύπους και έχει συναρτήσεις ως first class objects. Η σύνταξή της είναι επηρεασμένη από τη C. Η JavaScript αντιγράφει πολλά ονόματα και naming conventions από τη Java, αλλά γενικά οι δύο αυτές γλώσσες δε σχετίζονται και έχουν πολύ διαφορετική σημασιολογία. Οι βασικές αρχές σχεδιασμού της JavaScript προέρχονται από τις γλώσσες προγραμματισμού Self και Scheme.

Ακόμη, είναι γλώσσα βασισμένη σε προγραμματιστικά παραδείγματα (multi-paradigm), υποστηρίζοντας αντικειμενοστρεφές, προστακτικό και συναρτησιακό στυλ προγραμματισμού. Η JavaScript χρησιμοποιείται και σε εφαρμογές εκτός ιστοσελίδων — τέτοια παραδείγματα είναι τα έγγραφα PDF, οι site specific browsers και οι μικρές εφαρμογές της επιφάνειας εργασίας (desktop

widgets). Πρόσφατες εικονικές μηχανές και λόγω της ανάπτυξης για JavaScript έχουν επίσης κάνει τη JavaScript πιο δημοφιλή για την ανάπτυξη εφαρμογών Ιστού στην πλευρά του διακομιστή (server-side). Το πρότυπο της JS κατά τον οργανισμό τυποποίησης ECMA ονομάζεται ECMAScript.

3.1.4 TypeScript

Η TypeScript [4] είναι μια γλώσσα προγραμματισμού που αναπτύχθηκε και συντηρείται από τη Microsoft. Είναι ένα αυστηρό συντακτικό υπερσύνολο JavaScript και προσθέτει προαιρετική στατική πληκτρολόγηση στη γλώσσα. Το TypeScript έχει σχεδιαστεί για την ανάπτυξη μεγάλων εφαρμογών και μεταγλωττίζει σε JavaScript. Καθώς η TypeScript είναι ένα υπερσύνολο JavaScript, τα υπάρχοντα προγράμματα JavaScript είναι επίσης έγκυρα προγράμματα TypeScript.

Η TypeScript μπορεί να χρησιμοποιηθεί για την ανάπτυξη εφαρμογών JavaScript για εκτέλεση τόσο από πλευράς πελάτη όσο και από πλευράς διακομιστή (όπως με το Node.js ή το Deno). Υπάρχουν πολλές διαθέσιμες επιλογές για μετάθεση. Μπορεί να χρησιμοποιηθεί ο προεπιλεγμένος έλεγχος TypeScript, ή ο μεταγλωττιστής Babel μπορεί να κληθεί για να μετατρέψει το TypeScript σε JavaScript.

Η TypeScript υποστηρίζει αρχεία ορισμού που μπορούν να περιέχουν πληροφορίες τύπου υφιστάμενων βιβλιοθηκών JavaScript, όπως και τα αρχεία κεφαλίδας C++ μπορούν να περιγράψουν τη δομή των υφιστάμενων αρχείων αντικειμένων. Αυτό επιτρέπει σε άλλα προγράμματα να χρησιμοποιούν τις τιμές που ορίζονται στα αρχεία σαν να ήταν οντότητες TypeScript με στατική πληκτρολόγηση. Οι κεφαλίδες TypeScript για τις βασικές ενότητες της Node.js είναι επίσης διαθέσιμες, επιτρέποντας την μεγάλη ανάπτυξη προγραμμάτων Node.js εντός του TypeScript.

Ο μεταγλωττιστής της TypeScript είναι ο ίδιος γραμμένος σε TypeScript και μεταγλωτισμένος σε JavaScript. Έχει άδεια σε Apache 2.0. Η TypeScript περιλαμβάνεται ως γλώσσα προγραμματισμού πρώτης κατηγορίας στο Microsoft Visual Studio 2013 Update 2 και νεότερη έκδοση, μαζί με την C# και άλλες γλώσσες της Microsoft. Μια επίσημη επέκταση επιτρέπει στο Visual Studio 2012 να υποστηρίζει επίσης TypeScript. Ο Anders Hejlsberg, επικεφαλής αρχιτέκτονας της C# και δημιουργός των Delphi και το της Turbo Pascal, εργάστηκε για την ανάπτυξη και τον σχεδιασμό της TypeScript.

3.1.5 NPM (Node Package Manager)

Το NPM (Node Package Manager) [5] είναι ένας διαχειριστής πακέτων για τη γλώσσα προγραμματισμού JavaScript. Είναι ένας προεπιλεγμένος διαχειριστής πακέτων για το περιβάλλον εκτέλεσης JavaScript Node.js (JavaScript Runtime Environment). Αποτελείται από έναν client γραμμής εντολών, που ονομάζεται επίσης NPM, και μια ηλεκτρονική βάση δεδομένων τόσο δημόσιων όσο και ιδιωτικών (private paid for) πακέτων. Τα αρχεία είναι προσβάσιμα από τον client και τα διαθέσιμα πακέτα μπορούν να αναζητηθούν μέσω της ιστοσελίδας του NPM (<https://www.npmjs.com/>), και υπεύθυνοι για αυτά είναι η NPM, Inc όπου εξαγοράστηκε από την Microsoft.

Επίσης, το NPM είναι γραμμένο εξ ολοκλήρου σε JavaScript και αναπτύχθηκε από τον Isaac Z. Schlueter με την έμπνευση από άλλα παρόμοια έργα όπως το PEAR (PHP). Ειδικότερα, είναι ένα online αποθετήριο για τη δημοσίευση πακέτων Node.js και ένα βοηθητικό πρόγραμμα γραμμής εντολών για αλληλεπίδραση με το εν λόγω αποθετήριο που βοηθά στην εγκατάσταση πακέτων, στη διαχείριση των εκδόσεων τους και στη διαχείριση εξάρτησης. Ένα πακέτο ουσιαστικά, είναι ένας

φάκελος όπου περιέχει ένα πρόγραμμα το οποίο περιγράφεται από ένα αρχείο που ονομάζεται “package.json”.

3.1.5 Angular (Web framework)

Η **Angular** (συνήθως αναφέρεται ως "Angular 2+" ή "Angular v2 και άνω") [6] είναι ένα πλαίσιο εφαρμογών ιστού δωρεάν και ανοιχτού κώδικα που βασίζεται σε TypeScript και το οποίο καθοδηγείται από την ομάδα Angular της Google και από μια κοινότητα ατόμων και εταιρειών.

Το Angular είναι μια πλήρης επανεγγραφή από την ίδια ομάδα που δημιούργησε το AngularJS και χρησιμοποιείται ως το προσκήνιο της στοίβας MEAN, αποτελούμενο από βάση δεδομένων MongoDB, πλαίσιο διακομιστή εφαρμογών ιστού Express.js, το ίδιο το Angular (ή AngularJS) και περιβάλλον εκτέλεσης διακομιστή Node.js.

3.1.6 Material Design / Angular Material

Το Material Design (Quantum Paper) [7] είναι μια γλώσσα σχεδιασμού που αναπτύχθηκε από την Google το 2014. Επεκτείνοντας τις "κάρτες" που πρωτοεμφανίστηκαν στο Google Now, το Material Design χρησιμοποιεί περισσότερες διατάξεις με βάση το πλέγμα, κινούμενες εικόνες και μεταβάσεις, πληρώσεις και βάθος εφέ, όπως ο φωτισμός και οι σκιές. Η Google ανακοίνωσε το Material Design στις 25 Ιουνίου 2014, στο συνέδριο Google I/O του 2014.

Ο κύριος σκοπός του Material Design είναι η δημιουργία μιας νέας σχεδιαστικής βιβλιοθήκης που συνδυάζει τις αρχές του καλού σχεδιασμού με την τεχνική και επιστημονική καινοτομία. Ο σχεδιαστής Ματίας Ντουάρτε (Matías Duarte) εξήγησε ότι, "σε αντίθεση με το πραγματικό χαρτί, το ψηφιακό υλικό μας μπορεί να επεκταθεί και να αναμορφωθεί έξυπνα. Το υλικό έχει φυσικές επιφάνειες και άκρα. Οι ραφές και οι σκιές παρέχουν νόημα σε αυτό που μπορείτε να αγγίξετε." Η Google δηλώνει ότι η νέα γλώσσα σχεδιασμού τους βασίζεται σε χαρτί και μελάνι, αλλά η εφαρμογή πραγματοποιείται με προηγμένο τρόπο.

Το 2018, η Google παρουσίασε λεπτομερώς μια ανανέωση της γλώσσας, με έμφαση στην παροχή περισσότερης ευελιξίας στους σχεδιαστές για τη δημιουργία προσαρμοσμένων "θεμάτων" με διαφορετική γεωμετρία, χρώματα και τυπογραφία. Η Google κυκλοφόρησε το Material Theme Editor αποκλειστικά για την εφαρμογή σχεδίασης macOS Sketch.

Η Material βιβλιοθήκη της Angular είναι μία open source βιβλιοθήκη με βασικά components για την ανάπτυξη του UI/UX μιάς ιστοσελίδας. Κάποια ενδεικτικά components είναι: buttons, menus, datepickers, icons, sliders, snack bars tables, tooltips κ.ά.

3.1.7 RxJS

Στην πληροφορική, ο reactive προγραμματισμός είναι ένα παράδειγμα δηλωτικού προγραμματισμού που αφορά τα ρεύματα δεδομένων (data streams) και τη διάδοση της αλλαγής. Με αυτό το παράδειγμα, είναι δυνατή η εύκολη έκφραση στατικών (π.χ. συστοιχιών) ή δυναμικών (π.χ. εκπομπών συμβάντων) ροών δεδομένων, και επίσης η επικοινωνία ότι υπάρχει μια συναγόμενη εξάρτηση στο σχετικό μοντέλο εκτέλεσης, η οποία διευκολύνει την αυτόματη διάδοση των αλλαγμένων δεδομένων ροή.

Το RxJS [8] είναι ένα library για τη σύνθεση ασύγχρονων και βασισμένων σε γεγονότα προγραμμάτων χρησιμοποιώντας παρατηρήσιμες ακολουθίες. Παρέχει έναν βασικό τύπο, το

Observable, δορυφορικούς τύπους (Observer, Schedulers, Subjects) και χειριστές εμπνευσμένους από τα (map, filter, κλπ) για να επιτρέπεται ο χειρισμός ασύγχρονων συμβάντων ως συλλογών.

Σκεφτείτε το RxJS ως Lodash για events.

Το ReactiveX συνδυάζει το μοτίβο Observer με το μοτίβο Iterator και τον λειτουργικό προγραμματισμό με συλλογές για να καλύψει την ανάγκη για έναν ιδανικό τρόπο διαχείρισης ακολουθιών γεγονότων.

Οι βασικές έννοιες στο RxJS που λύνουν τη διαχείριση ασύγχρονων συμβάντων είναι:

- **Observable:** αντιπροσωπεύει την ιδέα μιας αδιάσειστης συλλογής μελλοντικών αξιών ή γεγονότων.
- **Observer:** είναι μια συλλογή από κλήσεις που ξέρει πώς να ακούει τις τιμές που παραδίδονται από το Observable.
- **Subscription:** αντιπροσωπεύει την εκτέλεση ενός Observable, είναι κυρίως χρήσιμο για την ακύρωση της εκτέλεσης.
- **Operators:** είναι καθαρές συναρτήσεις που επιτρέπουν ένα λειτουργικό στυλ προγραμματισμού για την αντιμετώπιση συλλογών με λειτουργίες όπως map, filter, concat, reduce κ.α
- **Subject:** είναι ισοδύναμο με ένα EventEmitter και ο μόνος τρόπος πολλαπλής μετάδοσης μιας αξίας ή συμβάντος σε πολλούς παρατηρητές.
- **Schedulers:** είναι κεντρικοί αποστολείς για τον έλεγχο της ταυτόχρονης λειτουργίας, επιτρέποντάς μας να συντονιστούμε όταν συμβαίνει υπολογισμός π.χ. setTimeout ή requestAnimationFrame κ.ά.

3.1.8 JSON Web Token

Τα JSON Web Tokens [9] αποτελούν ένα standard μιας και η πληροφορία που φέρουν μεταδίδεται μέσω JSON. Λειτουργούν καλά ανάμεσα σε πολλές διαφορετικές γλώσσες όπως .NET, Python, Node.js, Java, PHP, Ruby, Go, JavaScript, και Haskell.

3.1.8.1 Χαρακτηριστικά

- Τα JWTs περιέχουν μέσα τους όλη την απαραίτητη πληροφορία. Αυτό σημαίνει ότι ένα JWT είναι σε θέση να μεταφέρει βασική πληροφορία για αυτό, το φορτίο του (που συνήθως είναι πληροφορίες που σχετίζονται με έναν χρήστη) και μια υπογραφή.
- Τα JWTs μπορούν να μεταφερθούν εύκολα. Από τη στιγμή που τα JWTs είναι αυτόνομα, μπορούν να χρησιμοποιηθούν μέσα σε HTTP headers όταν πιστοποιούν ένα API. Μπορούν επίσης να σταλούν μέσω του URL.

3.1.8.2 Μορφή

Είναι εύκολο να αναγνωρισθεί ένα JWT. Είναι 3 αλφαριθμητικά χωρισμένα με τελεία. Για παράδειγμα: aaaaaaaaaa.bbbbbbbbbbbbbb.cccccccccccccccc. Ας δούμε την σημασία του κάθε πεδίου.

3.1.8.3 Ανάλυση

Από τη στιγμή που έχουμε 3 πεδία, καθένα από αυτά δημιουργείται διαφορετικά. Τα πεδία αυτά είναι:

- Header
- Payload
- Signature

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.  
eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4  
gRG91IiwiaXNjb2NpYWwiOnRydWV9.  
4pcPyMD09o1PSyXnrXCjTwXyr4BsezdI1AVTmud2fU4
```

Ο header περιλαμβάνει 2 μέρη:

- αναφορά του τύπου, που είναι JWT
- ο hashing αλγόριθμος που χρησιμοποιείται

Ένα παράδειγμα φαίνεται παρακάτω: { "typ": "JWT", "alg": "HS256" }

Επειδή αυτό το κομμάτι κωδικοποιείται (base64encode), έχουμε το πρώτο μέρος του JWT που μοιάζει: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.

Το payload μεταφέρει το κυρίως φορτίο του JWT. Εδώ είναι που μπαίνει η πληροφορία που θέλουμε να μεταδώσουμε καθώς και άλλες πληροφορίες σχετικές με το token. Ένα παράδειγμα του payload φαίνεται παρακάτω:

Ένα παράδειγμα του payload φαίνεται παρακάτω:

```
{ "iss": "geokeloglou.com", "exp": 1300819380, "name": "George Keloglou", "admin": true }
```

Το οποίο κωδικοποιείται σε:

```
eyJJuYW1laWQiOiI2ZmFmMTFjZC04MmU0LTQyODYtOTI1ZS0zMGNmMTQ3ODIIZWYiLCJlb  
WFpbCI6Imdlb2tlbG9nbG91QGdtYWlsLmNvbSIsIm5iZiI6ImTYzMTY0NTg1MSwiZXhwIjozNjM  
xNzMyMjUxLCJpYXQiOiE2MzE2NDU4NTF9.
```

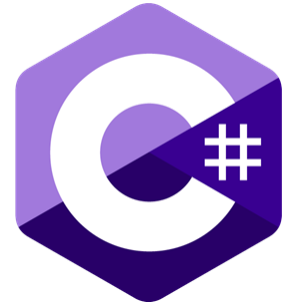
Το τρίτο και τελευταίο μέρος αποτελεί η υπογραφή. Αυτό το μέρος αποτελείται από τη hash τιμή των header, payload και ενός μυστικού κλειδιού. Το κλειδί αυτό βρίσκεται στο server. Έτσι ο server είναι σε θέση να πιστοποιήσει υπάρχοντα tokens και να δημιουργήσει νέα. Έτσι προκύπτει και το τελευταίο μέρος του token:

```
p7FEINjFy1vaaq59P2ZnFwLJlCc1cOyjMlVNlib1EwsRD_fp8Bkw1K8wx9ZK6eUBfG8zjDWSsx-  
zx0UCXoUDyw
```

3.2 Τεχνολογίες back-end

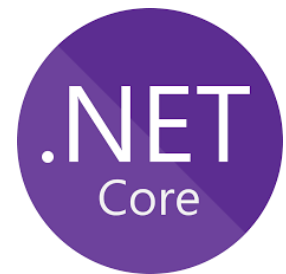
3.2.1 C#

Η C# (C Sharp) [10] είναι μια γλώσσα προγραμματισμού H/Y. Δημιουργήθηκε από την Microsoft μέσα από την πλατφόρμα .NET και αργότερα αναγνωρίστηκε επισήμως από την Ecma (ECMA-334) και την ISO (ISO/IEC 23270:2018). Ο βασικός σχεδιαστής της ομάδα που διαχειρίζεται την γλώσσα ονομάζεται Άντερς Χάιλσμπεργκ (Anders Hejlsberg). Στις 15-8-2012 κυκλοφόρησε η έκδοση 5.0 η οποία είναι η πιο πρόσφατη μέχρι σήμερα. Είναι μια από τις γλώσσες προγραμματισμού που δημιουργήθηκαν για την Κοινή Υποδομή Γλώσσας (αγγλικά: Common Language Infrastructure). Ο βασικός σκοπός της γλώσσας είναι να είναι απλή αντικειμενοστρεφής γλώσσα για γενική χρήση.



3.2.2 .NET Core

Το .NET Core [11] είναι μια πλατφόρμα ανάπτυξης ανοιχτού κώδικα γενικού σκοπού. Μπορείτε να δημιουργήσετε εφαρμογές .NET Core για Windows, macOS και Linux για επεξεργαστές x64, x86, ARM32 και ARM64 χρησιμοποιώντας πολλές γλώσσες προγραμματισμού. Τα πλαίσια και τα API παρέχονται για cloud, IoT, UI πελάτη και μηχανική εκμάθηση.



Πρόκειται για ένα Framework που λειτουργεί τόσο στο πλήρες .NET Framework, στα Windows όσο και στο .NET Core πολλαπλών πλατφορμών.

Ωστόσο, η έκδοση 3 του ASP.NET Core λειτουργεί μόνο σε .NET Core και σταμάτα να προσφέρει υποστήριξη για το .NET Framework. Το Framework είναι μια πλήρης επανεγγραφή που ενώνει το προηγούμενο ξεχωριστό ASP.NET MVC και ASP.NET Web API σε ένα μονό μοντέλο προγραμματισμού.

Παρόλο που είναι ένα νέο Framework, χτισμένο σε μια νέα στοίβα ιστού, έχει υψηλό βαθμό συμβατότητας με το ASP.NET. Οι εφαρμογές ASP.NET Core υποστηρίζουν side by side versioning , όπου διαφορετικές εφαρμογές, που εκτελούνται στον ίδιο υπολογιστή, μπορούν να στοχεύουν διαφορετικές εκδόσεις του ASP.NET Core. Αυτό δεν είναι δυνατό με προηγούμενες εκδόσεις του ASP.NET.

3.2.3 Entity Framework Core

Το Entity Framework [12][13] είναι μια ομάδα από τεχνολογίες στο ADO.NET που υποστηρίζει την ανάπτυξη των δεδομένων (data-oriented) σε εφαρμογές. Οι αρχιτέκτονες και οι προγραμματιστές των data-oriented εφαρμογών έχουν δυσκολευτεί να πετύχουν δυο διαφορετικούς στόχους. Πρέπει να μοντελοποιήσουν τις έννοιες (entities), τις σχέσεις (relationships), και την λογική από τα “business” προβλήματα που λύνουν και ακόμα πρέπει να δουλέψουν με data engines που χρησιμοποιούνται ώστε να αποκομίσουν δεδομένα. Τα δεδομένα μπορούν να έχουν πολλαπλά συστήματα αποθήκευσης, το κάθε ένα με το δικό του πρωτόκολλο, ακόμα εφαρμογές που λειτουργούν με ένα μέσο αποθήκευσης

πρέπει να ισορροπήσουν τα προαπαιτούμενα του ανυποθήκευτου συστήματος ενάντια τις απαιτήσεις του καλού και διατηρητέου κωδικά μια εφαρμογής.

Το Entity Framework προσφέρει στον προγραμματιστές να δουλεύουν με τα δεδομένα στη μορφή του domain-specific objects και ιδιότητες, όπως σαν πελάτες και διεύθυνση πελάτη, χωρίς να χρειάζεται να ανησυχεί για του πίνακες της βάσης δεδομένων και τις στήλες και το που αποθηκεύονται τα δεδομένα. Με το Entity Framework, οι προγραμματιστές μπορούν να δουλεύουν σε πιο υψηλό abstract επίπεδο όταν δουλεύουν με δεδομένα. Επειδή το Entity Framework είναι ένα στοιχείο του .NET Framework και του .NET Core, το Entity Framework μπορεί να λειτουργήσει σε όλους του υπολογιστές που έχουν το .NET Framework ή .NET Core εγκατεστημένο.

3.2.4 Dapper

Το Dapper [14] είναι ένας απλός αντιστοιχιστής αντικειμένων για το .NET και κατέχει τον τίτλο του King of Micro ORM όσον αφορά την ταχύτητα και είναι σχεδόν τόσο γρήγορος όσο η χρήση ενός ακατέργαστου προγράμματος ανάγνωσης δεδομένων ADO.NET. Το ORM είναι ένας αντικειμενικός σχεσιακός χαρτογράφος, ο οποίος είναι υπεύθυνος για τη χαρτογράφηση μεταξύ βάσης δεδομένων και γλώσσας προγραμματισμού. Το Dapper επεκτείνει το IDbConnection παρέχοντας χρήσιμες μεθόδους επέκτασης για να αναζητήσετε τη βάση δεδομένων σας.

3.2.5 SQL / PostgreSQL

Η SQL (αγγλ.αρκτ.από το Structured Query Language) [15] είναι μία γλώσσα υπολογιστών στις βάσεις δεδομένων, που σχεδιάστηκε για τη διαχείριση δεδομένων, σε ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (Relational Database Management System, RDBMS) και η οποία, αρχικά, βασίστηκε στη σχεσιακή άλγεβρα. Η γλώσσα περιλαμβάνει δυνατότητες ανάκτησης και ενημέρωσης δεδομένων, δημιουργίας και τροποποίησης σχημάτων και σχεσιακών πινάκων, αλλά και ελέγχου πρόσβασης στα δεδομένα. Η SQL4 ήταν μία από τις πρώτες γλώσσες για το σχεσιακό μοντέλο του Edgar F. Codd, στο σημαντικό άρθρο του το 1970, και έγινε η πιο ευρέως χρησιμοποιούμενη γλώσσα για τις σχεσιακές βάσεις δεδομένων.

Η PostgreSQL είναι ένα σύστημα διαχείρισης βάσης δεδομένων αντικειμενικής σχέσης (ORDBMS) βασισμένο στο POSTGRES, έκδοση 4.2, που αναπτύχθηκε στο Πανεπιστήμιο της Καλιφόρνιας στο Τμήμα Επιστήμης Υπολογιστών του Berkeley. Το POSTGRES πρωτοστάτησε σε πολλές έννοιες που έγιναν διαθέσιμες μόνο σε ορισμένα εμπορικά συστήματα βάσεων δεδομένων πολύ αργότερα. Η PostgreSQL είναι απόγονος ανοιχτού κώδικα αυτού του αρχικού κώδικα του Μπέρκλεϋ.

Υποστηρίζει ένα μεγάλο μέρος του προτύπου SQL και προσφέρει πολλά σύγχρονα χαρακτηριστικά:

- σύνθετα ερωτήματα
- ξένα κλειδιά
- ενεργοποιεί
- αναβαθμίσιμες προβολές
- συναλλακτική ακεραιότητα
- έλεγχος ταυτόχρονης πολυτροπίας

Επίσης, η PostgreSQL μπορεί να επεκταθεί από τον χρήστη με πολλούς τρόπους, για παράδειγμα με την προσθήκη νέων:

- τύπων δεδομένων
- λειτουργιών
- χειριστών
- συγκεντρωτικών συναρτήσεων
- μεθόδους ευρετηρίου
- διαδικαστικών γλωσσών

Και λόγω της φιλελεύθερης άδειας, το PostgreSQL μπορεί να χρησιμοποιηθεί, τροποποιηθεί και διανεμηθεί από οποιονδήποτε δωρεάν για οποιονδήποτε σκοπό, είτε πρόκειται για ιδιωτικό, είτε για εμπορικό είτε για ακαδημαϊκό επίπεδο.

3.2.6 SignalR

Το SignalR [16] είναι μια βιβλιοθήκη της Microsoft η οποία απλουστεύει τη διαδικασία της ενσωμάτωσης λειτουργικότητας σε πραγματικό χρόνο μέσα στις εφαρμογές μας. Η λειτουργικότητα πραγματικού χρόνου στην ουσία είναι η δυνατότητα του διακομιστή να δίνει δεδομένα στους συνδεδεμένους πελάτες αμέσως όταν τα δεδομένα είναι διαθέσιμα χωρίς να περιμένει να ζητήσουν οι πελάτες καινούργια δεδομένα. Μέσω του SignalR δίνεται η δυνατότητα να υπάρξουν εφαρμογές που χρειάζονται μεγάλης συχνότητας ενημερώσεων από το διακομιστή.

Το SignalR παρέχει τη δυνατότητα στιγμιαίας προώθησης περιεχομένου στους συνδεδεμένους client από τη στιγμή που αυτό γίνεται διαθέσιμο με χρήση λογισμικού από τη μεριά του server, αντί ο server να αναμένει τον client να ζητήσει νέα δεδομένα. Η σύνδεση μεταξύ του client και του server είναι μόνιμη, σε αντίθεση με μία κλασική HTTP σύνδεση η οποία επανεγκαθίσταται για κάθε επικοινωνία. Το SignalR χρησιμοποιεί τη μεταφορά τύπου WebSocket, όπου αυτή είναι διαθέσιμη (αυτού του είδους οι μεταφορές εξαρτώνται από την υποστήριξη σε HTML5), και όποτε είναι απαραίτητο επαναχρησιμοποιεί παλαιότερους τύπους μεταφοράς (αν ο browser του χρήστη δεν υποστηρίζει HTML5).

Η χρήση του SignalR σημαίνει πως πολλές επιπλέον λειτουργίες που υπό άλλες συνθήκες θα έπρεπε να ενσωματωθούν, είναι ήδη διαθέσιμες. Είναι σημαντικό να τονίσουμε ότι παρέχει την δυνατότητα εκμετάλλευσης του WebSocket, χωρίς να χρειάζεται να ανησυχούμε για τη δημιουργία λογισμικού ξεχωριστής διαδρομής για τους παλαιότερους clients. Το SignalR επίσης, απαλλάσσει από το άγχος των αναβαθμίσεων για το WebSocket, δεδομένου ότι το SignalR θα συνεχίσει να αναβαθμίζεται προκειμένου να υποστηρίζει αλλαγές στην υποκείμενη μεταφορά. Αυτό, σκοπό έχει να εξασφαλίσει ένα συνεκτικό interface για την εφαρμογή για όλες τις εκδόσεις του WebSocket. Το SignalR είναι ιδανικό για διαδικτυακές εφαρμογές που απαιτούν μεγάλη συχνότητα αναβαθμίσεων από τον server, πχ. παιχνίδια σε πραγματικό χρόνο.

3.2.6.1 SignalR API

Το SignalR API παρέχει τη δυνατότητα δημιουργίας απομακρυσμένων κλήσεων διαδικασίας (server-to-client remote procedure calls (RPC)) μεταξύ του server και του client, οι οποίες καλούν JavaScript λειτουργίες σε client browsers (και άλλες client πλατφόρμες) από το .NET λογισμικό του server.

Επιπλέον, το SignalR περιλαμβάνει API για τη διαχείριση των συνδέσεων, όπως για παράδειγμα, events σύνδεσης και αποσύνδεσης.

Το SignalR μπορεί να κάνει αυτόματη διαχείριση της σύνδεσης και επιτρέπει την εκπομπή μηνυμάτων προς όλους τους συνδεδεμένους clients ταυτόχρονα, όπως σε ένα chat room. Επιτρέπει την αποστολή μηνυμάτων σε συγκεκριμένους clients. Η σύνδεση μεταξύ client και server είναι μόνιμη, σε αντίθεση με μία κλασική HTTP σύνδεση η οποία επανεγκαθίσταται για κάθε επικοινωνία.

Το SignalR υποστηρίζει τη "server push" λειτουργία, κατά την οποία το λογισμικό του server μπορεί να καλέσει το λογισμικό του client σε ένα browser, χρησιμοποιώντας Remote Procedure Calls (RPC), αντί των καλεσμάτων που βασίζονται στο μοντέλο request-response και που είναι κυρίαρχα στο Διαδίκτυο σήμερα. Οι SignalR εφαρμογές μπορούν να κάνουν scale χιλιάδες clients χρησιμοποιώντας Service Bus, Redis ή RabbitMQ κ.α.

3.2.6.2 Connections και Hubs:

Ένα connection αναπαριστά ένα απλό endpoint για την αποστολή single-recipient, grouped, ή broadcast μηνυμάτων. Το Persistent Connection API δίνει στον developer απευθείας πρόσβαση στο low-level communication protocol που το SignalR κάνει expose. Ένα hub είναι ένα πιο πολύ high-level pipeline χτισμένο πάνω στο Connection API όπου επιτρέπει στον client και τον server να καλέσουν μεθόδους μεταξύ τους άμεσα. Το SignalR χειρίζεται την διεκπεραίωση στο μηχανήμα ως δια μαγείας, επιτρέποντας στους clients να καλέσουν μεθόδους στον server τόσο εύκολα όσο τις local μεθόδους, και αντίθετα.

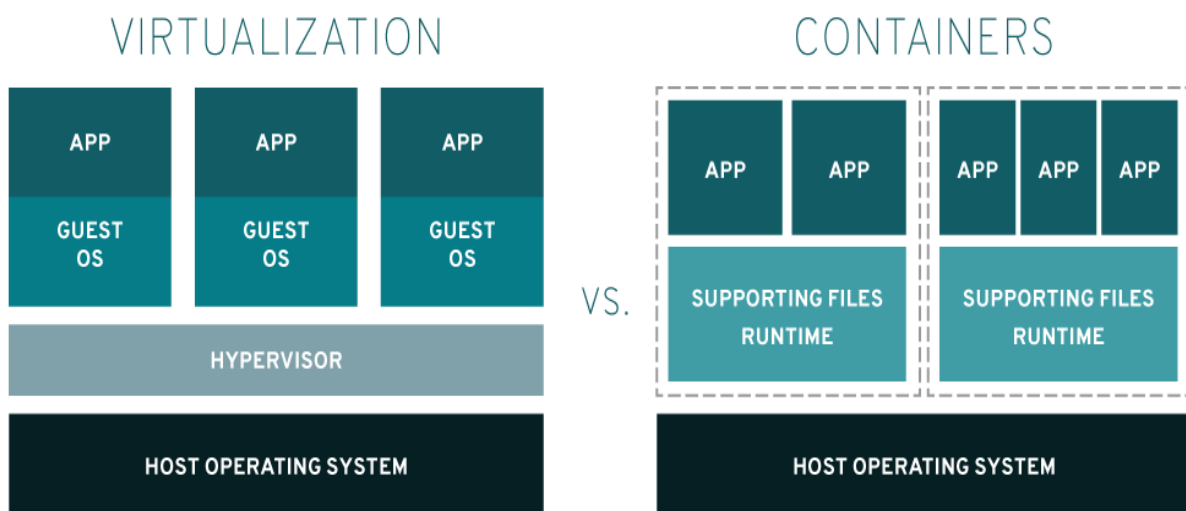
3.2.6.3 Πως λειτουργεί το Hub;

Πρέπει να σκεφτούμε το Hub ως μία κεντρική τοποθεσία για όλες τις συνδέσεις της εφαρμογής. Όταν ο server-side κώδικας καλεί μια μέθοδο στον client, , ένα πακέτο στέλνεται στην ενεργή μεταφορά που περιέχει το όνομα και τις παραμέτρους της μεθόδου που θέλουμε να κληθεί (όταν ένα αντικείμενο αποστέλλεται ως παράμετρος της μεθόδου, γίνεται serialized χρησιμοποιώντας JSON). Ο client ύστερα αντιστοιχεί το όνομα της μεθόδου στην μέθοδο που ορίστηκε στον client-side κώδικα. Αν υπάρχει αντιστοιχία, η client μέθοδος θα εκτελεστεί χρησιμοποιώντας τα deserialized parameters δεδομένα.

3.2.7 Docker

3.2.7.1 Εισαγωγή

Το Docker [17] είναι μια αρχιτεκτονική για containers. Άλλη αντίστοιχη αρχιτεκτονική είναι το OCI (Open Container Initiative). Τα containers ενσωματώνουν όλα τα κομμάτια που χρειάζεται μια εφαρμογή για να λειτουργήσει ξεχωριστά από τις υπόλοιπες σε ένα ενιαίο πακέτο, το οποίο απομονώνεται από τις υπόλοιπες εφαρμογές και λειτουργικό σύστημα με τρόπους που βασίζονται στη χρήση linux namespaces και έχουν τις απαρχές τους στο chroot jail. Θα μπορούσε κανείς να πει ότι τα containers υλοποιούν operating system virtualization κατά αντίστοιχο τρόπο που τα Virtual Machines υλοποιούν hardware virtualization. Κατά τον ίδιο τρόπο λοιπόν που τα Virtual Machines λειτουργούν απομονωμένα και μοιράζονται το ίδιο φυσικό hardware, έτσι και τα containers λειτουργούν απομονωμένα και μοιράζονται το ίδιο λειτουργικό σύστημα με απόλυτη απομόνωση σε επίπεδο λειτουργικού και processes.



Εικόνα 3.5: Virtualization vs Containers

3.2.7.2 Docker server

Γνωστός και ως docker host, είναι στο μηχάνημα (φυσικό ή VM) που φιλοξενεί τα containers και είναι το λογισμικό που διαμοιράζει τα resources, φροντίζει για την απομόνωση και την λειτουργία των containers (start, stop, κλπ) αλλά και την επικοινωνία τους είτε μεταξύ τους είτε με τον υπόλοιπο κόσμο.

3.2.7.3 Docker client

Το σύνολο των εντολών (linux) ή το ξεχωριστό λογισμικό (windows) που αποτελούν το περιβάλλον με το οποίο ο χρήστης ή μια εξωτερική εφαρμογή μπορεί να επικοινωνήσει με το Docker Server και να επηρεάσει τη λειτουργία των containers.

3.2.7.4 Docker Engine

Το σύνολο της client-server εφαρμογής που περιλαμβάνει

- Docker Server
- Docker Client

- Docker APIs

3.2.7.5 Docker images

Βασικά πακέτα πάνω στα οποία μπορεί κανείς να βασιστεί και να φτιάξει containers ή πιο σύνθετα images. Είναι δυνατό να φτιαχτούν πολύπλοκα σύνολα λογισμικού με αντίστοιχα πολύπλοκο τρόπο. Επίσης μπορεί ένα image να φτιαχτεί με layers. Ο τρόπος που φτιάχνεται ένα image από στοιχειώδη συστατικά μπορεί να προδιαγραφεί απόλυτα με την χρήση των Dockerfiles. Η διαδικασία της κατασκευής του image ονομάζεται build. Η δυνατότητα ορισμού των προδιαγραφών του build για ένα image δίνει τη δυνατότητα της χρήσης των containers ως Infrastructure as Code (περιγράψω απλώς πως θέλω να είναι το image και αυτό χτίζεται κατά τις προδιαγραφές μου αυτόματα). Έχει σημασία η προσοχή στις διαδικασίες που ορίζονται στα Dockerfile καθώς κάθε εντολή καταλήγει στο να προσθέσει ένα ακόμα layer στο image. Τα layers από τη στιγμή που θα προστεθούν είναι immutable. Γι αυτό και πολλές φορές οι εντολές σε κάθε βήμα ενός Dockerfile καταλήγουν να είναι αρκετά πολύπλοκες, καθώς δεν μπορεί κανείς να διαγράψει στην πράξη σε ένα επόμενο layer αρχεία που προστέθηκαν σε προηγούμενο layer. Επειδή τα images αποθηκεύονται σε κεντρικά σημεία - Docker registry - και γίνονται download από εκεί, το μέγεθος των images έχει σημασία για την αποδοτικότητα της υποδομής. [18]

3.2.7.5 Docker Hub

Η κεντρική αποθήκη της Docker για τα images. Είναι ο δημόσιος “κατάλογος” - Registry για τα images και μπορεί ο καθένας να δημοσιεύσει τα δικά του, είτε εταιρείες/ιδρύματα είτε ιδιώτες. Μπορεί φυσικά κανείς να διαθέτει και private registry που φιλοξενείται στη δική του ιδιωτική υποδομή. [19]

3.2.7.6 Docker containers

Τα αγαπημένα μας containers που φτιάχνονται πανομοιότυπα των images αλλά μπορούν να επηρεαστούν κατά τη δημιουργία τους ώστε να υπάρξουν τροποποιήσεις. Επίσης κατά τη δημιουργία των containers από κάποιο image, όπου το container είναι πλέον σαν ένα ξεχωριστό σύνολο από processes και εφαρμογές, μπορούν να οριστούν και διάφορες παράμετροι λειτουργίας, όπως μέρη του filesystem του host στα οποία έχει πρόσβαση το container με συγκεκριμένα δικαιώματα, τα network ports τα οποία ορίζονται ως σημεία πρόσβασης για το container είτε από το εξωτερικό δίκτυο είτε από το εσωτερικό “εικονικό” δίκτυο του docker host.

3.2.7.7 Docker networking

Γενικά μπορούμε να σκεφτούμε το τρόπο που επικοινωνούν τα containers μεταξύ τους αλλά και με το υπόλοιπο δίκτυο ως bridging. Δημιουργούνται εικονικά bridges στον Docker host που μπορούν να έχουν ή να μην έχουν επικοινωνία με το εξωτερικό δίκτυο. Ένα container μπορεί να επικοινωνεί με περισσότερα από ένα bridges. Αυτό δίνει τη δυνατότητα για πολύπλοκους συνδυασμούς ώστε κάποια containers π.χ. να επικοινωνούν μόνο μεταξύ τους και να χρησιμοποιούν τρίτο container για να δώσουν services προς το εξωτερικό δίκτυο. Φυσικά τέτοιες αρχιτεκτονικές είναι αρκετά πολύπλοκες

και η χρήση τους γίνεται αρκετά απλούστερη με τη χρήση εργαλείων για orchestration. Η επικοινωνία μεταξύ containers μπορεί αν χρειαστεί να ξεπεράσει τα όρια ενός docker host, κάτι το οποίο θεωρείται αρκετά πολύπλοκο σαν αρχιτεκτονική και πέρα από τη λογική ενός εισαγωγικού post.

3.2.7.8 Χρήση

Η χρήση των containers μπορεί να αφορά από την αυτοματοποίηση των διαδικασιών χτισίματος μιας πλατφόρμας ή μιας εφαρμογής μέχρι την πλήρη χρήση Infrastructure as Code. Μπορούν να θεωρηθούν, όταν οι εφαρμογές που υλοποιούν διαχωρίζονται απόλυτα ως προς τα services σε ξεχωριστά containers, ότι αποτελούν βασικό συστατικό του Microservices Architecture, ειδικά με το συνδυασμό με εργαλεία orchestration, όπου μπορεί κανείς να χρησιμοποιήσει υποδομές CI/CD για να ενημερώνει τον κώδικα των εφαρμογών, να χτίζει τα κατάλληλα images, να σηκώνει τα containers, να κάνει τις δοκιμές κλπ δημιουργώντας πλήρεις κύκλους ανάπτυξης και λειτουργίας λογισμικού και υπηρεσιών. Το orchestration γενικά μπορεί να διευκολύνει αρκετά τη χρήση των containers σε κάθε σενάριο. Εδώ και κάποια χρόνια η χρήση των containers εξετάζεται και σε περιπτώσεις edge computing στο δίκτυο, καθώς πολλές πλατφόρμες networking υποστηρίζουν containers όπως π.χ. τα Cisco Catalyst 9K κ.α

3.2.7.9 Docker-compose

Προδιαγραφές καταγεγραμμένες σε αρχεία ρυθμίσεων, μπορούν να χρησιμοποιηθούν από κατάλληλα εργαλεία για να επιτευχθεί η αυτοματοποίηση της δημιουργίας images και containers. Ένα τέτοιο εργαλείο για την πλατφόρμα Docker με απλή λειτουργία, κυρίως για λόγους test & development (χωρίς να αποκλείεται περιορισμένη χρήση σε production) είναι το docker-compose.

Στο docker-compose χρησιμοποιούνται αρχεία yaml για τον ορισμό services τα οποία χτίζονται με containers αλλά και για τον ορισμό της επικοινωνίας αυτών των services μεταξύ τους και με τον εξωτερικό κόσμο. Το docker-compose σαν λογισμικό βασίζεται σε ένα αριθμό από εντολών και είναι αντίστοιχο με το docker client runtime σε σχέση με το τί μπορεί να εκτελεστεί για να επηρεάσουμε τη λειτουργία των services βασισμένα στα docker containers ή να κάνουμε συγκεκριμένους ελέγχους. Η χρήση της πλατφόρμας Docker απλοποιείται εξαιρετικά με τη χρήση του docker-compose. Βασικό μειονέκτημα του docker-compose [20] είναι η έλλειψη υποστήριξης scale out δηλαδή να μπορεί κανείς να προσθέτει/αφαιρεί nodes οριζόντια για όποιο service χρειάζεται ώστε να ανταποκρίνεται στο φόρτο και τις ανάγκες για τις συγκεκριμένες εφαρμογές. Για τέτοιες περιπτώσεις είναι απαραίτητες διαφορετικά εργαλεία όπως το docker swarm ή το Kubernetes

3.2.7.10 Kubernetes

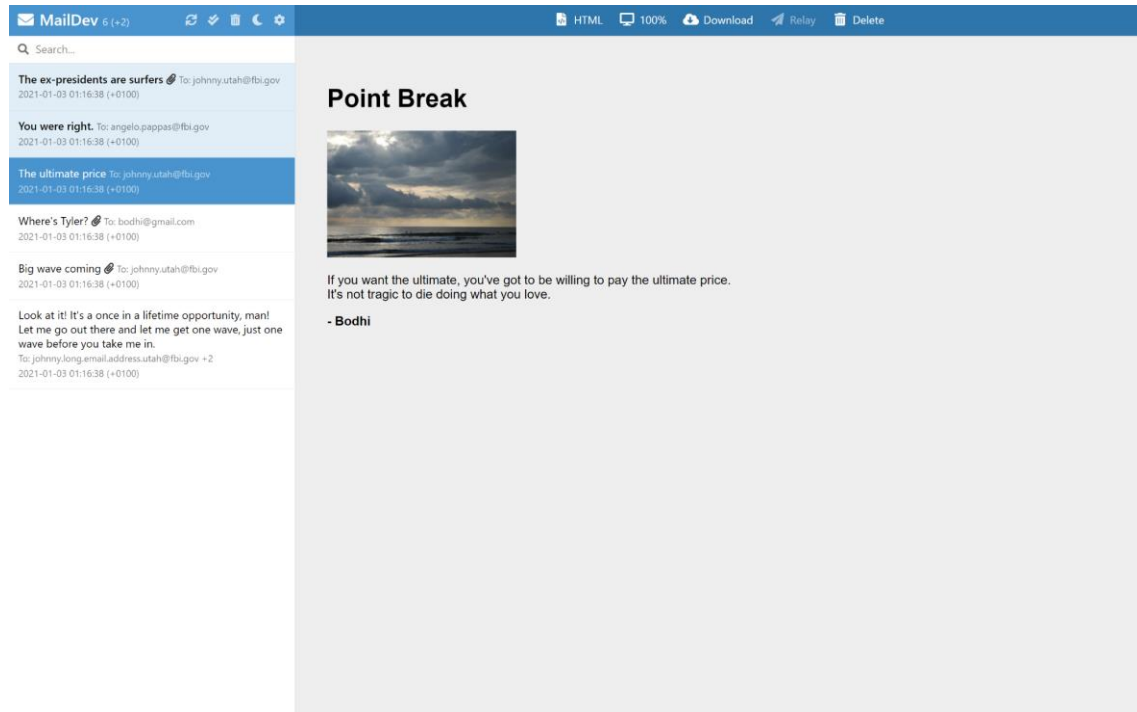
Το Kubernetes [21] είναι ένα πολύ ανταγωνιστικό εργαλείο στην αγορά για υποστήριξη microservices architecture και scalable υποδομών εφαρμογών σε περιβάλλοντα cloud. Βασικό του συστατικό είναι το pod το οποίο μπορεί να περιλαμβάνει ένα ή περισσότερα services. Η προσθήκη ή αφαίρεση pods είναι ο τρόπος που μια τέτοια υποδομή αντιμετωπίζει την αύξηση/μείωση του load για τις εφαρμογές που χτίζονται πάνω στα containers. Τέλος, πρόσφατα ανακοινώθηκε ότι το Kubernetes δεν θα υποστηρίζει πλέον docker containers.

3.2.7.11 Logging

Η παρατήρηση των logs είναι εξίσου σημαντική στην πλατφόρμα docker & docker-compose όσο και στο linux.

3.2.8 Maildev

Το Maildev [22] είναι ένα πολύ απλό open source εργαλείο, το οποίο δίνοντας μας ένα απλό web interface μπορούμε στην διαδικασία του development να πραγματοποιήσουμε testing στα emails που στέλνει η εφαρμογή μας. Είναι φτιαγμένο πάνω στην τεχνολογία Node.js.



Εικόνα 3.6: Maildev UI

3.2.9 Microsoft Azure και blob

Το Microsoft Azure [23], που συνήθως αναφέρεται ως Azure, είναι μια υπηρεσία υπολογιστικού νέφους που δημιουργήθηκε από τη Microsoft για τη δημιουργία, τον έλεγχο, την ανάπτυξη και τη διαχείριση εφαρμογών και υπηρεσιών μέσω κέντρων δεδομένων που διαχειρίζονται η Microsoft.

Το blob [24] είναι ένα δυαδικό, μεγάλο αντικείμενο και μια επιλογή αποθήκευσης για κάθε τύπο δεδομένων που θέλετε να αποθηκεύσετε σε δυαδική μορφή.

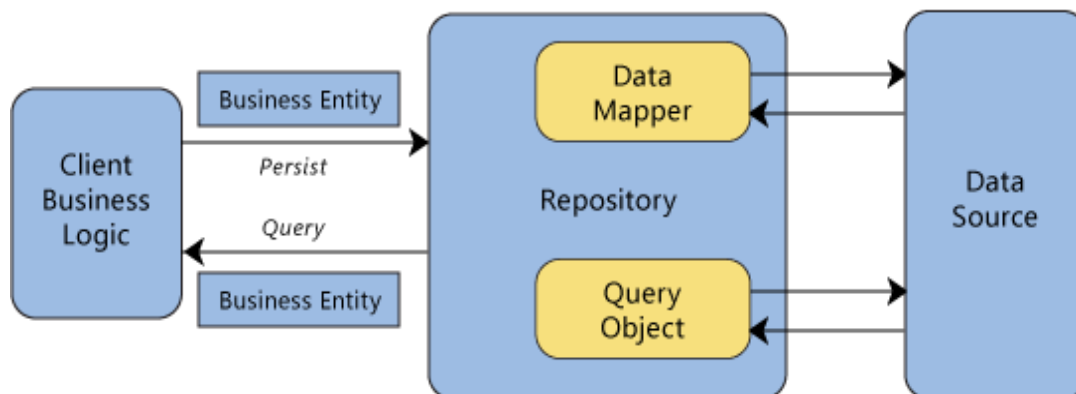
3.2.10 Repository Pattern

Ένα αποθετήριο σε C# [25] μεσολαβεί μεταξύ των επιπέδων χαρτογράφησης τομέα και δεδομένων (όπως το Entity Framework). Σας επιτρέπει να τραβήξετε μια εγγραφή ή αριθμό εγγραφών από σύνολα δεδομένων και, στη συνέχεια, να λειτουργήσουν αυτές οι εγγραφές ως μια συλλογή αντικειμένων εντός της μνήμης, και μπορείτε επίσης να ενημερώσετε ή να διαγράψετε εγγραφές εντός αυτού του συνόλου δεδομένων και τον κωδικό αντιστοίχισης που περιλαμβάνεται από το Αποθετήριο θα πραγματοποιήσει τις κατάλληλες επιχειρήσεις στο παρασκήνιο.

Το μοτίβο αποθήκευσης C# είναι ένας τρόπος υλοποίησης της πρόσβασης στα δεδομένα εγκλωβίζοντας το σύνολο των αντικειμένων που διατηρούνται σε μια αποθήκη δεδομένων και τις λειτουργίες που εκτελούνται πάνω από αυτά, παρέχοντας μια πιο αντικειμενοστραφή προβολή του επιπέδου επιμονής.

Το μοτίβο αποθήκευσης C# υποστηρίζει επίσης τον στόχο της επίτευξης καθαρού διαχωρισμού και μονόδρομης εξάρτησης μεταξύ των επιπέδων χαρτογράφησης τομέα και δεδομένων.

Το μοτίβο αποθήκευσης C# χρησιμοποιείται κυρίως εκεί που πρέπει να τροποποιήσουμε τα δεδομένα πριν περάσουμε στο επόμενο στάδιο. Ένα γράφημα που απεικονίζει την ιδέα:



Εικόνα 3.6: Repository Pattern

3.2.10.1 Γιατί να χρησιμοποιήσουμε Repository Pattern;

- Αυξάνει την ικανότητα και διαθεσιμότητα για Testing: Τα συστήματα αποθήκευσης είναι καλά για δοκιμές. Ένας λόγος είναι ότι μπορείτε να χρησιμοποιήσετε το Dependency Injection. Βασικά, δημιουργείτε μια διεπαφή για το αποθετήριο σας και αναφέρετε τη διεπαφή για αυτό όταν δημιουργείτε το αντικείμενο. Στη συνέχεια, μπορεί αργότερα να δημιουργήσετε ένα ψεύτικο αντικείμενο (χρησιμοποιώντας mock για παράδειγμα) το οποίο υλοποιεί αυτήν τη διεπαφή.
- Εύκολα εναλλάξιμο με πολλά data stores χωρίς να αλλαχθεί το API: Για παράδειγμα, σε μια περίπτωση, μπορεί να χρειαστεί να ανακτήσετε δεδομένα από τη βάση δεδομένων, σε άλλες περιπτώσεις μπορεί να χρειαστεί να ανακτήσετε κάτι από API τρίτου μέρους ή ίσως υπάρχει κάποιο άλλο μέρος από το οποίο πρέπει να ανακτήσετε δεδομένα. Ανεξάρτητα από αυτό, η ιδέα πίσω από το μοτίβο αποθήκευσης είναι ότι ό, τι βρίσκεται πίσω από αυτό δεν έχει σημασία όσο το API που παρέχει λειτουργεί για το επίπεδο της εφαρμογής που το καλεί.

Κεφάλαιο 4: Προγράμματα, οδηγίες, περιβάλλον και η εφαρμογή

4.1 Προγράμματα και οδηγίες

Για την υλοποίηση της εφαρμογής χρησιμοποιήθηκαν πολλά προγράμματα, όχι μόνο για την συγγραφή του κώδικα αλλά και για την επεξεργασία των δεδομένων - τεχνολογιών.

4.1.1 Προγράμματα που χρησιμοποιήθηκαν

Ο κώδικας του backend, αναπτύχθηκε στο εργαλείο ανάπτυξης κώδικα ονόματι Rider και αντίστοιχα ο κώδικας του front-end αναπτύχθηκε στο WebStorm. Και τα δύο προγράμματα είναι software της JetBrains. Για να τρέχει η εφαρμογή όλες οι διαχειριστικές εργασίες πραγματοποιήθηκαν στα τερματικά Konsole, και iTerm2. Για την επίβλεψη της βάσης PostgreSQL χρησιμοποιήθηκε το εργαλείο PgAdmin.

4.1.2 Πως τρέχουμε την εφαρμογή σε local development environment

Αρχικά, πρέπει να κατεβάσουμε την εφαρμογή στο τοπικό περιβάλλον εργασίας μας, οπότε η κλωνοποίηση του project είναι απαραίτητη.

Συνεπώς:

- Συνδεόμαστε στον λογαριασμό μας στο GitHub
- URL: <https://github.com/geokeloglou/symposium> του repository
- Κλωνοποιούμε στο terminal με HTTPS, SSH ή GitHub CLI το repository στο τοπικό μας περιβάλλον
- Μπαίνουμε στον φάκελο symposium και βλέπουμε τα υπο-projects (frontend, backend)
- Ανοίγουμε δεύτερο terminal και στο ένα είμαστε στο frontend και στο άλλο στο backend. Αντίστοιχα "symposium-frontend", "symposium-backend".
- Η εντολή για να τρέξει το backend στον φάκελο "symposium-backend" είναι:


```
$ docker-compose -f docker/local.yml up --build
```
- Η εντολή για να τρέξει το frontend είναι στον φάκελο "symposium-frontend":


```
$ ng serve
```
- Μόλις τρέξουν και τα δύο η πρόσβαση της εφαρμογής θα πραγματοποιηθεί από το localhost στο port που θα εμφανίσει το frontend όσον αφορά το development environment.
- Όλα τα emails της εφαρμογής θα εμφανιστούν και θα σταλούν στο Maildev, όπου η διεύθυνση πρόσβασης θα εμφανιστεί στο terminal με την ολοκλήρωση του backend.

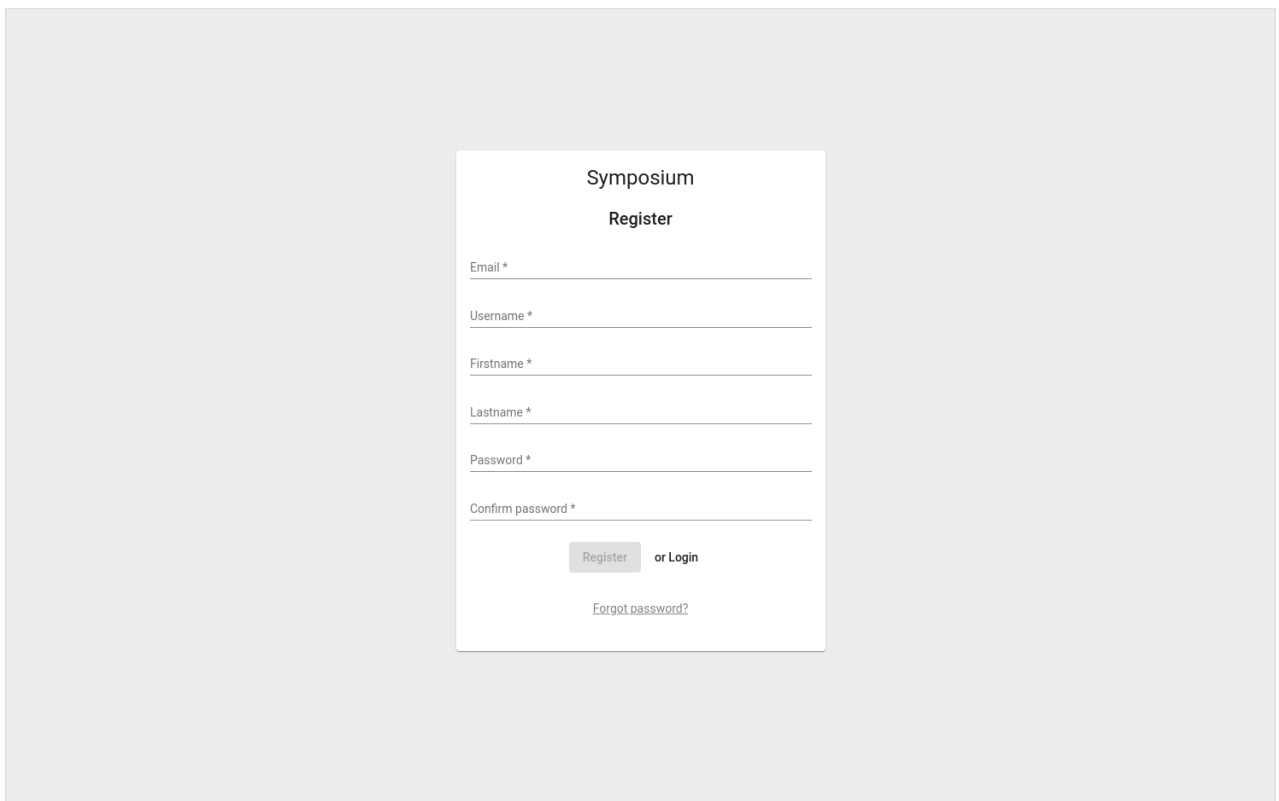
4.2 Η εφαρμογή - Symposium

Η εφαρμογή αποτελεί ένα κοινωνικό δίκτυο που στοχεύει στο microblogging και στην επικοινωνία των χρηστών σε chat.

Οι βασικές λειτουργίες που παρέχει είναι οι εξής:

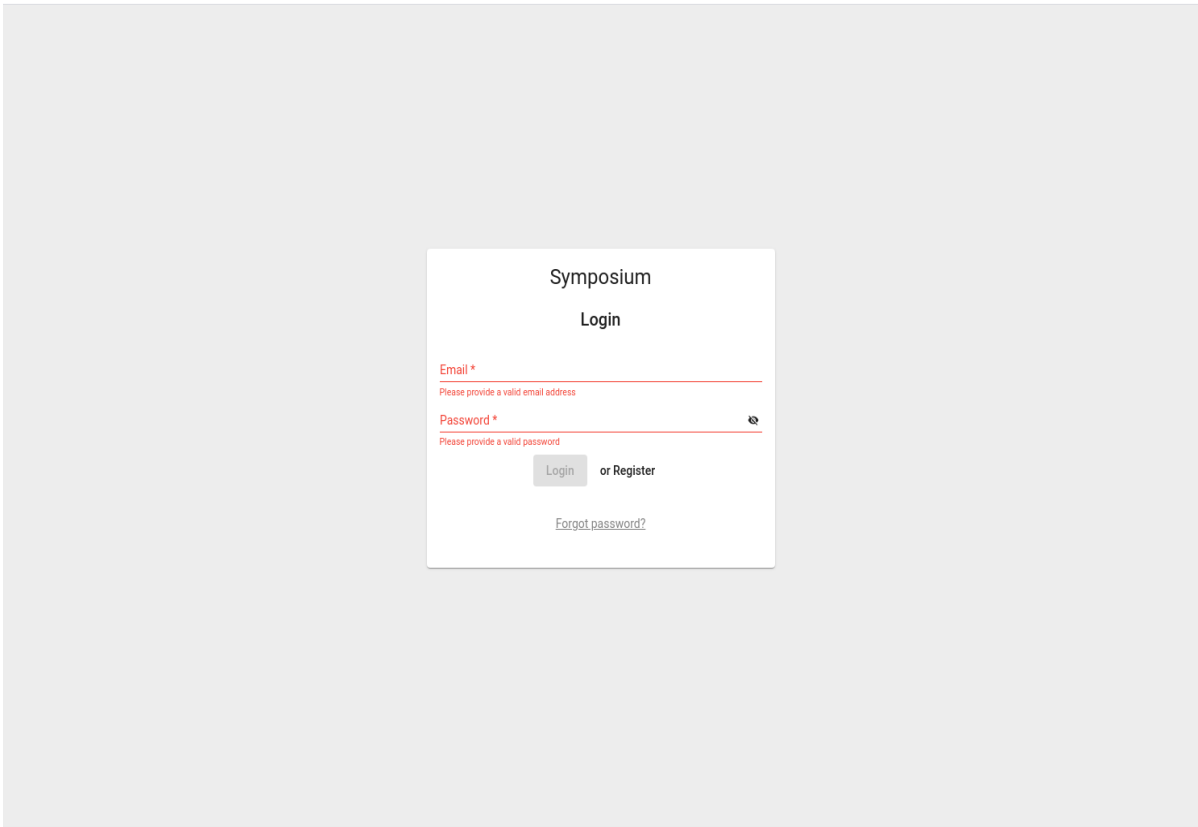
- Authentication system (login, register, forgot - reset password)
- Δημιουργία, επεξεργασία και διαγραφή ενός post
- Upload φωτογραφίας
- Upload φωτογραφίας profile
- Like σε post
- Chat μεταξύ των χρηστών

Με την είσοδο του νέου χρήστη στην πλατφόρμα πρέπει να πραγματοποιηθεί η εγγραφή του.

The image shows a registration form for the 'Symposium' application. The form is titled 'Symposium Register' and is centered on a light gray background. It contains several input fields: 'Email *', 'Username *', 'Firstname *', 'Lastname *', 'Password *', and 'Confirm password *'. Below the fields, there is a 'Register' button, a link 'or Login', and a link 'Forgot password?'. The form is enclosed in a white box with a thin border.

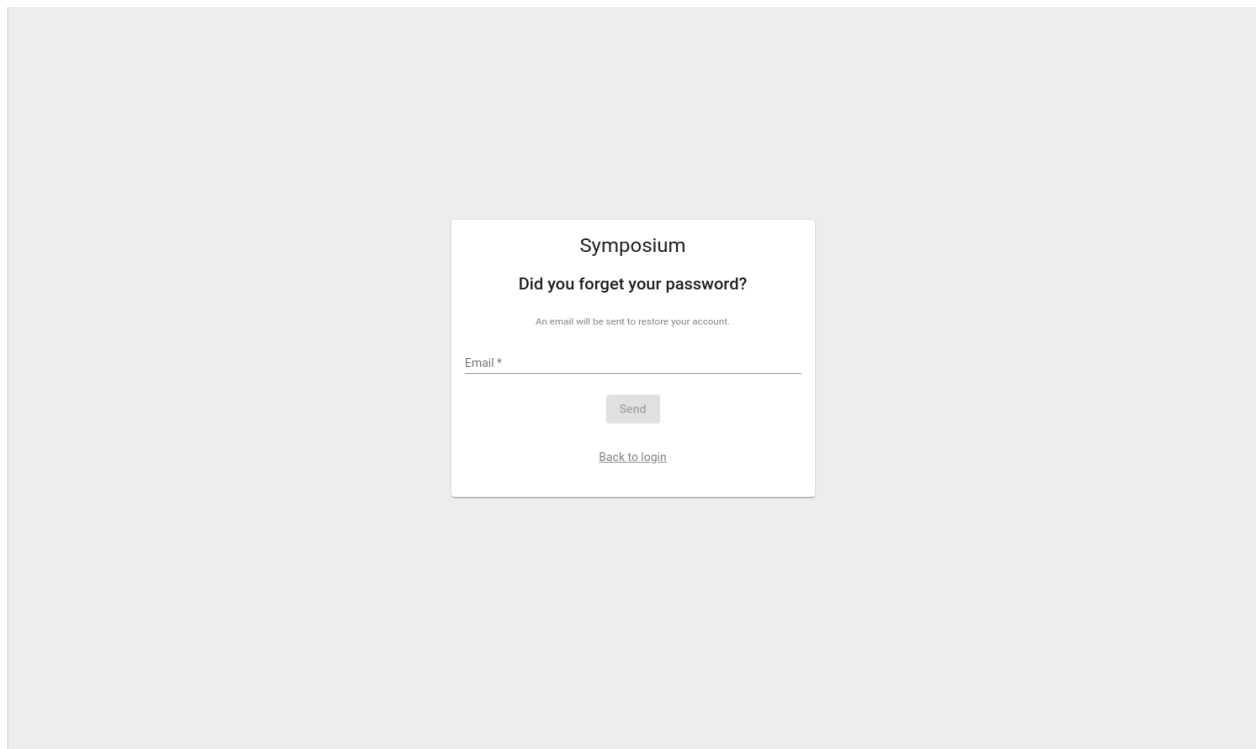
Εικόνα 4.1: Εγγραφή χρήστη

Ο εγγεγραμμένος χρήστης πλέον μπορεί να συνδεθεί και να αποκτήσει πρόσβαση στην πλατφόρμα.

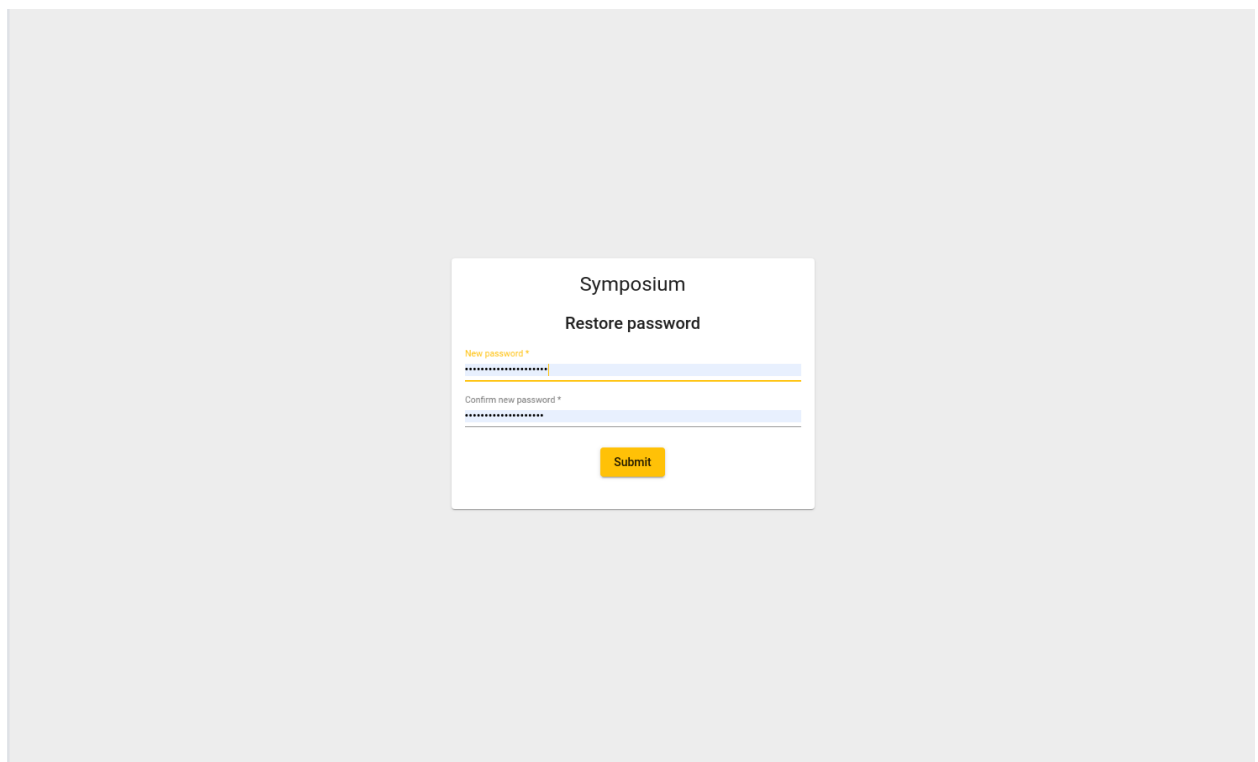


Εικόνα 4.2: Σύνδεση χρήστη

Εάν ο χρήστης έχει ξεχάσει τον κωδικό του, υπάρχει υλοποιημένη λειτουργικότητα, η οποία παρέχεται από την πλατφόρμα έτσι ώστε να επαναφέρει τον λογαριασμό του. Η διαδικασία που χρησιμοποιεί η πλατφόρμα είναι να στείλει ένα email στο email account του χρήστη με οδηγίες για να τοποθετήσει τον καινούργιο του κωδικό.

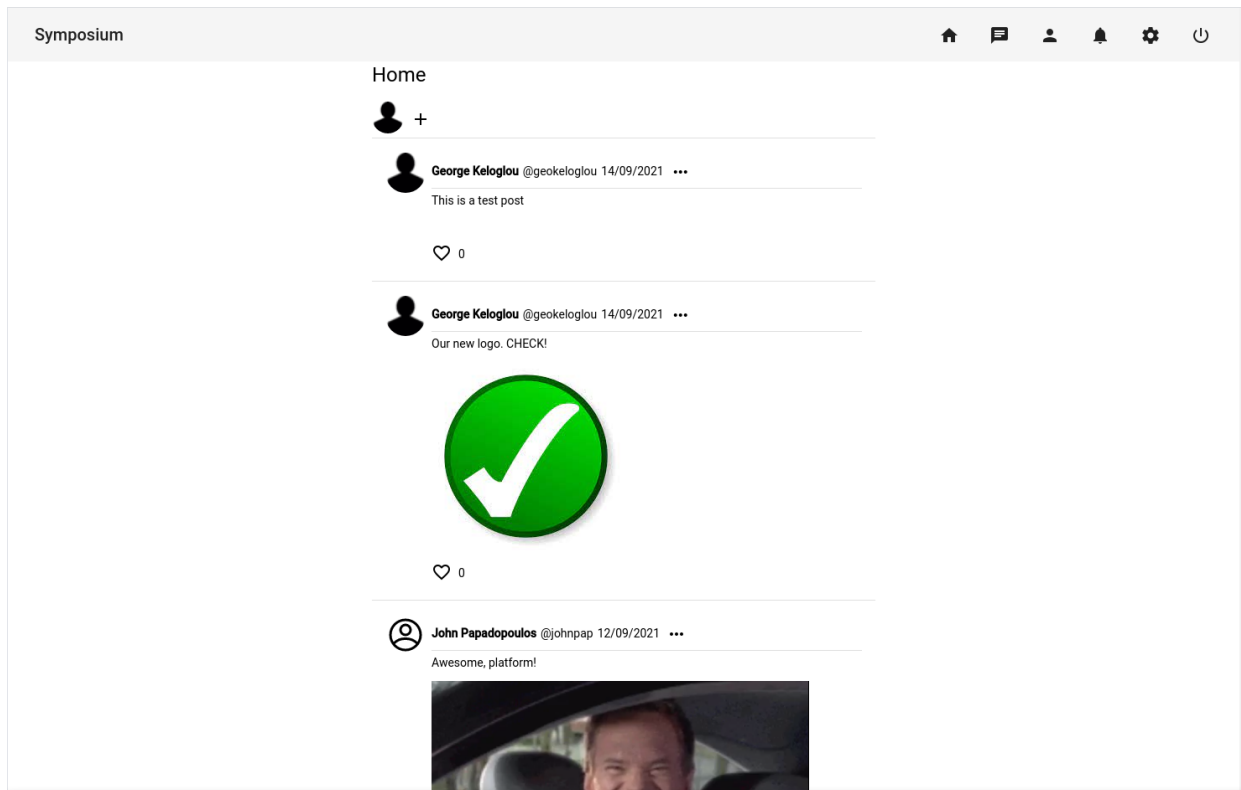


Εικόνα 4.3: Ξέχασες τον κωδικό σου;



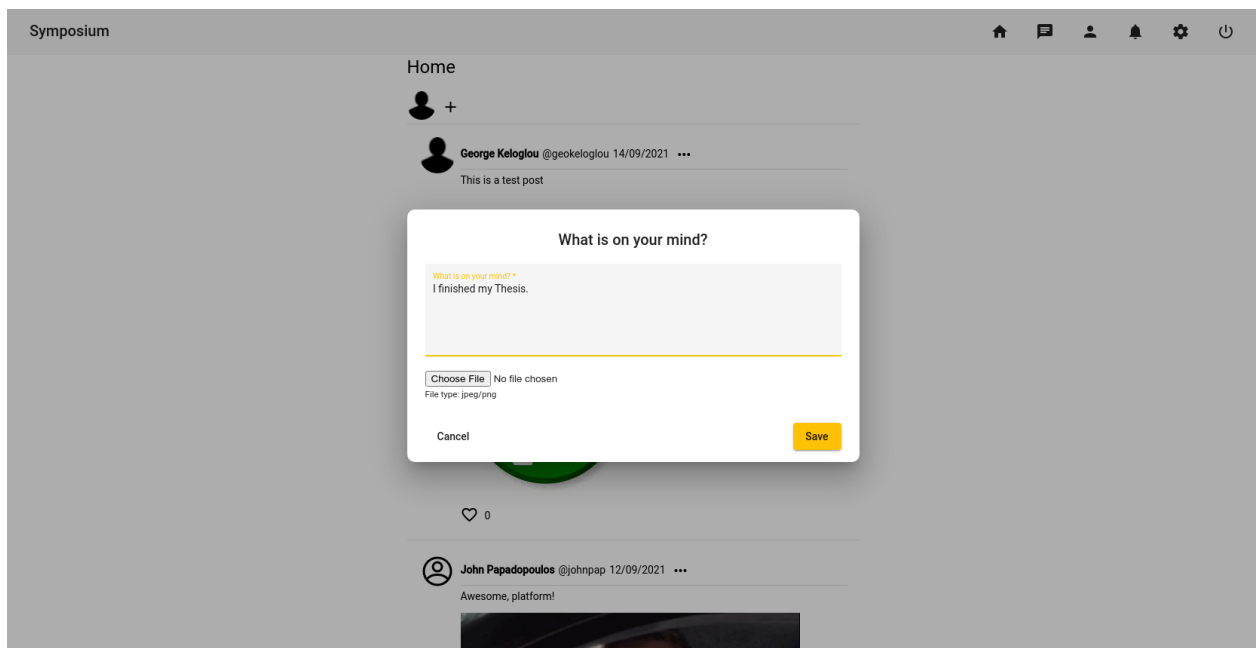
Εικόνα 4.4: Επαναφορά κωδικού

Ο χρήστης μόλις συνδεθεί στην πλατφόρμα του εμφανίζονται τα τελευταία posts των χρηστών.



Εικόνα 4.5: Newsfeed

Η δημιουργία ενός καινούργιου post πραγματοποιείται από το κουμπί του (+) δίπλα στην προσωπική φωτογραφία του χρήστη.

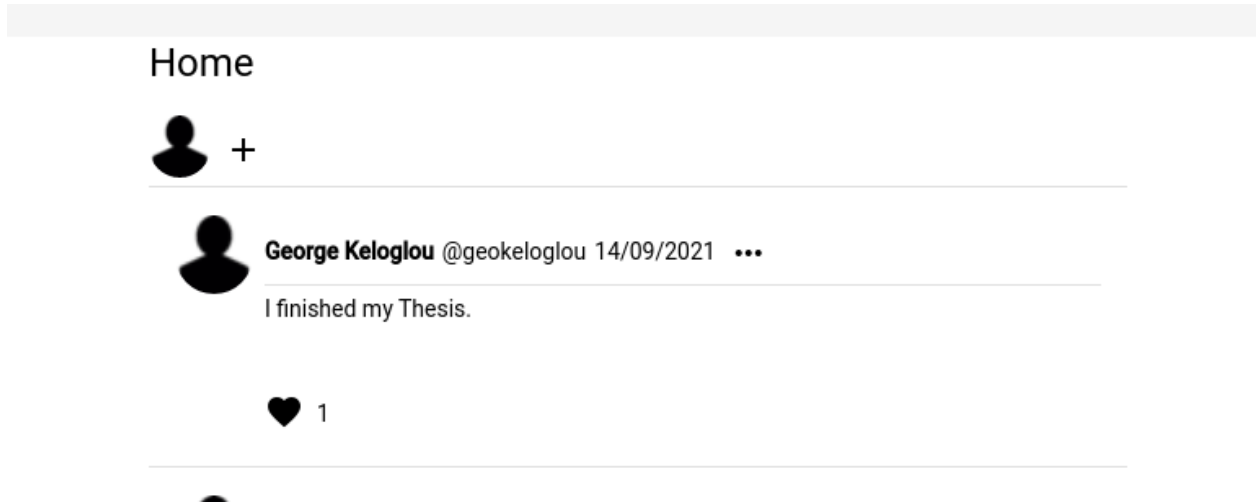


Εικόνα 4.6: Δημιουργία post

Κεφάλαιο 4

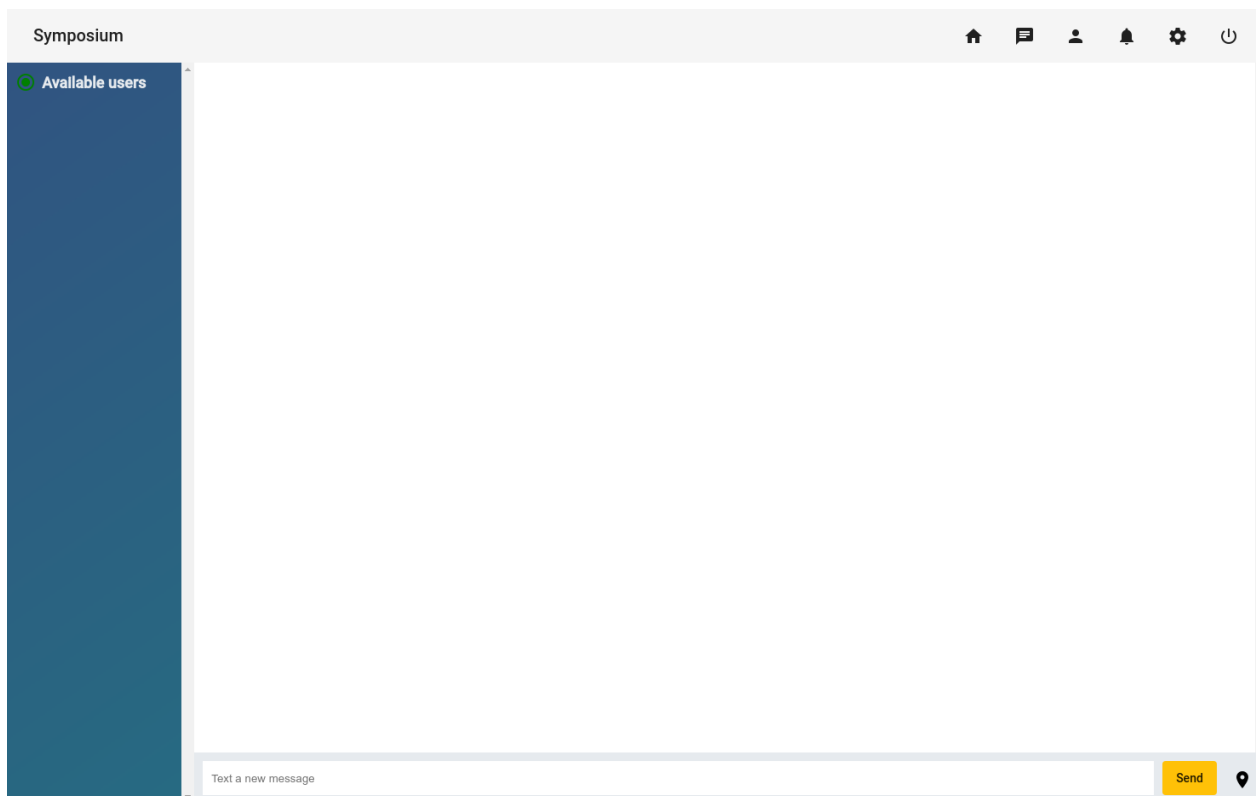
Μόλις ο χρήστης πατήσει το κουμπί του 'Create post' θα ανοίξει το dialog έτσι ώστε να συγγράψει και να δημοσιεύσει το επόμενο του post. Το post μπορεί να περιλαμβάνει μία φωτογραφία, μία φωτογραφία με μία περιγραφή ή μόνο κείμενο.

Μετά την αποθήκευση το καινούργιο post θα δημοσιευθεί στο 'Feed' όπου όλοι οι χρήστες έχουν πρόσβαση και μπορούν να πατήσουν το κουμπί του 'like' αν το επιθυμούν..



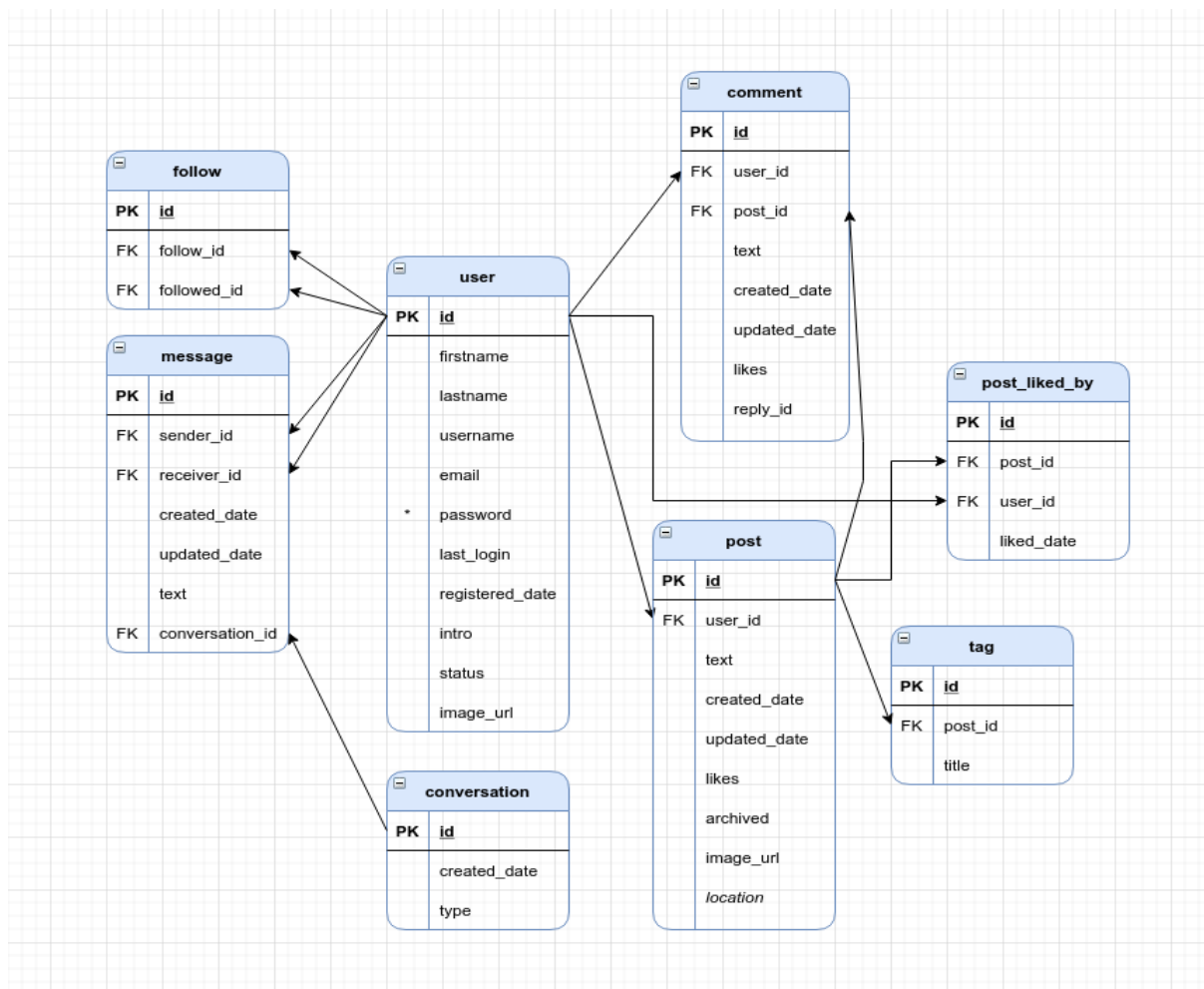
Εικόνα 4.7: Το Post

Τέλος, η πλατφόρμα προσφέρει την δυνατότητα του real-time chatting μεταξύ των χρηστών, όπου μπορούν πολλοί χρήστες να επικοινωνήσουν για διάφορα θέματα και απόψεις.



Εικόνα 4.8 Chat

4.2.1 Σχεδιάγραμμα βάσης δεδομένων - εφαρμογής



Σχήμα 4.1: Σχεδιάγραμμα βάσης δεδομένων Symposium

4.2.2 Αρχιτεκτονική εφαρμογής

Η αρχιτεκτονική μιας εφαρμογής είναι από τα πιο σημαντικά πράγματα για την υλοποίηση ενός έργου (project). Εκτός από το ότι προσφέρει μία καλή δομή στο έργο, μας δίνει και άλλα πλεονεκτήματα όπως την δυνατότητα για εύκολη συντήρηση, επεκτασιμότητα κ.α. Η εφαρμογή Symposium ακολουθεί διάφορες αρχιτεκτονικές στο frontend και στο backend. Μερικές από αυτές είναι το Repository Pattern στο backend όπου αναλύσαμε στο Κεφ.3 και φυσικά η εφαρμογή ακολουθεί τα S.O.L.I.D design principles στο frontend και στο backend.

4.2.3 Αναλυτική επεξήγηση front-end, back-end

Η εφαρμογή του front-end είναι δομημένη και χωρισμένη σε components, services, sandboxes, models, views, interceptors, guards και το shared.

- Τα components είναι τα σχεδιαστικά κομμάτια στην εφαρμογή τα οποία επαναλαμβάνονται. Τα services αποτελούν τα http requests ως προς τον server, καθώς τα sandboxes είναι αυτά τα οποία γίνονται provide για να πραγματοποιηθεί το request. Μόλις κάποιος component καταστραφεί, καταστρέφεται και το αντίστοιχο sandbox.
- Τα models είναι τα μοντέλα της εφαρμογής. Χρησιμοποιούμε κυρίως interfaces, types και enums. Με την χρήση των μοντέλων αξιοποιούμε τα types της TypeScript καθώς και του OOP.
- Τα views είναι οι η σύνθεση των components σε ένα συγκεκριμένο view.
- Οι interceptors χρησιμοποιούνται για να υλοποιήσουμε λειτουργίες και αρχικοποιήσεις στην εφαρμογή που αφορούν κυρίως τα requests. Για παράδειγμα, στην περίπτωση μας τοποθετούμε τον 'Authorization' header 'Bearer' εάν θέλουμε να στείλουμε ένα request στο API μας για να αντιληφθεί το backend εάν είμαστε authenticated.
- Η λειτουργία των guards είναι απλή. Μας δίνουν μία λειτουργικότητα πολύ σημαντική η οποία αφορά την πρόσβαση σε συγκεκριμένα components, container-components ή views. Ουσιαστικά, ένας guard επιστρέφει ένα boolean ή κάποια άλλα types (βλέπε UrlTree) και χρησιμοποιείται στο routing μέσω του CanActivate interface.
- Στο shared τοποθετούμε components, services, modules κ.α όταν θα χρησιμοποιηθούν σε πολλά σημεία οπότε αφορά πράγματα που χρησιμοποιούνται σε πολλά σημεία και έχουν μεγάλη επαναληψιμότητα. Επίσης, μπορούν να τοποθετηθούν components ή services ή modules τα οποία θα γίνουν provide σε όλη την εφαρμογή (βλέπε snackbar).

Η εφαρμογή του back-end είναι δομημένη και χωρισμένη σε Web, Data, Services, DTO και Tests.

- Το Web περιέχει ουσιαστικά το Startup.cs και το Program.cs όπου σε αυτά τα αρχεία γίνεται initialize και configure η εφαρμογή. Επίσης, περιέχει τους controllers οι οποίοι συντελούν το API. Επιπλέον, περιέχει το Dockerfile όπου περιέχει όλες τις οδηγίες για στηθεί η containerized εφαρμογή.
- Το DTO περιέχει όλα τα Data Transfer Objects.
- Το Data περιέχει τα μοντέλα, τα Migrations και το DbContext. Τα migrations είναι οι οδηγίες - εντολές του EF Core framework έτσι ώστε να δημιουργηθεί το DbContext δηλαδή το σχήμα της βάσης δεδομένων.
- Τα Services περιέχουν τα interfaces και τις implemented classes οι οποίες πραγματοποιούν όλες τις λειτουργίες της εφαρμογής.

Συγκεκριμένα:

Η API ακολουθεί το πρότυπο REST. Χρησιμοποιούμε το Design Pattern: Repository όπου υλοποιούμε τις βασικές λειτουργίες CRUD (Create, Read, Update, Delete).

```

4 usages 1 inheritor geokeloglou
public interface IPostService
{
    1 usage 1 implementation geokeloglou
    Task<ServiceResponse<Guid>> CreatePost(CreatePostDto postDto);
    1 usage 1 implementation geokeloglou
    Task<ServiceResponse<List<Post>>> GetPosts();
    1 usage 1 implementation geokeloglou
    Task<ServiceResponse<List<GetAllPostsDto>>> GetAllPosts();
    1 usage 1 implementation geokeloglou
    Task<ServiceResponse<Guid>> LikePost(LikePostDto postId);
    1 usage 1 implementation geokeloglou
    Task<ServiceResponse<List<PostLikedBy>>> GetAllLikedPosts();
    1 usage 1 implementation geokeloglou
    Task<ServiceResponse<Guid>> DeletePost(DeletePostDto deletePostDto);
}

```

Κώδικας 4.1: Interface IPostService (CRUD)

Όταν ο χρήστης στην εφαρμογή πατήσει να δημιουργήσει ένα Post θα γίνουν τα εξής πράγματα: Το front-end θα χτυπήσει το `‘/api/post/create’` endpoint, έχοντας στο Authorization header ένα valid Bearer Token και στο body τα keys `‘text’` ή `‘image’` (κείμενο ή φωτογραφία) και θα κληθεί η συνάρτηση `CreatePost` από τον Controller (Εικόνα 4.3).

```

[HttpPost(template: "create"),
Consumes(contentType: "multipart/form-data"),
RequestFormLimits(MultipartBodyLengthLimit = 20000000),
RequestSizeLimit(bytes: 20000000)]
geokeloglou *
public async Task<IActionResult> Create([FromForm] CreatePostDto createPostDto)
{
    var response = await _postService.CreatePost(createPostDto);
    if (!response.Success)
    {
        return BadRequest(response);
    }

    return Ok(response);
}

```

Κώδικας 4.2: Create post controller

Κεφάλαιο 4

Υστερα τα δεδομένα παραμετρικά θα περάσουν στην συνάρτηση που δημιουργεί το Post μέσα στο service. (Κώδικας 4.2)

Το service μέσω ελέγχων εξετάσει εάν έχουμε εικόνα για να την ανεβάσει στο Azure Blob μέσω του UploadAsync (IStorageService), έτσι ώστε να έχουμε πολύ χώρο για πολλές φωτογραφίες στο Cloud.

Εάν όλα πάνε καλά, θα δημιουργήσει το Post θα αποθηκεύσει νέα δεδομένα στην βάση και θα επιστρέψει το κατάλληλο response. (Κώδικας 4.3)

```
0+1 usages  geokeloglou
public async Task<ServiceResponse<Guid>> CreatePost(CreatePostDto postDto)
{
    var response = new ServiceResponse<Guid>();
    try
    {
        string imageUrl = null;
        if (postDto.PostImage != null)
        {
            imageUrl = _configuration.GetSection("Storage:ImageUrl").Value + postDto.PostImage.FileName;
            await _storageService.UploadAsync(postDto.PostImage);
        }

        var post = new Post
        {
            Text = postDto.Text,
            ImageUrl = imageUrl,
            CreatedDate = DateTimeOffset.Now,
            UserId = GetUserGuid()
        };

        await _context.Posts.AddAsync(post);
        await _context.SaveChangesAsync();

        response.Data = post.Id;
        response.Message = "Post has been created.";
    }
    catch (Exception e)
    {
        response.Success = false;
        response.Message = "Post has not been created.";
    }

    return response;
}
```

Κώδικας 4.3: Create post service

4.3 Προστιθέμενη αξία

Η προστιθέμενη αξία της εφαρμογής είναι το API που παρέχει. Ο σχεδιασμός του API ακολουθεί την αρχιτεκτονική του RESTful [26]. Για να υπάρξει πρόσβαση στην εφαρμογή από το API μετά το authentication θα πρέπει να τοποθετηθεί σαν “Authorization” header το Bearer token, όπου θα το λάβουμε στην διαδικασία του login.

4.4 Δημιουργία εσόδων (monetization)

Στην πλατφόρμα μπορούν να προστεθούν πολλοί τρόποι έτσι ώστε να είναι κερδοφόρα. Αρχικά, η πιο απλή και διαδεδομένη μέθοδος είναι οι διαφημίσεις, δηλαδή η παροχή υπηρεσιών σε λογαριασμούς όπου θέλουν να διαφημίσουν τον εαυτό τους, το προϊόν τους ή την υπηρεσία τους. Στην συνέχεια, μία ακόμη διαδεδομένη μέθοδος είναι η πληρωμένη ιδιότητα μέλους (paid membership) όπου η πλατφόρμα δίνει πρόσβαση σε χρήστες οι οποίοι θέλουν να αγοράσουν επιπλέον δυνατότητες στην εφαρμογή (π.χ αναλυτικά)

Κεφάλαιο 5: Προτάσεις για βελτίωση, συμπεράσματα και επίλογος

Στην παρούσα εργασία έγινε μία εκτενής αναφορά των τεχνολογιών της υλοποίησης μιας Διαδικτυακής εφαρμογής.

Η παρούσα υλοποίηση της εφαρμογής παρέχει βασική λειτουργικότητα κοινωνικής δικτύωσης. Ο χρήστης μπορεί να κάνει εγγραφή στην πλατφόρμα και μετά την είσοδό του να αποκτήσει πρόσβαση στο Newsfeed και να διαβάσει τα Posts των άλλων χρηστών, να κάνει like καθώς και να δημιουργήσει, να και επεξεργαστεί ένα Post εκ νέου. Το Post αποτελείται από κείμενο και φωτογραφία. Επίσης, μπορεί να επικοινωνήσει με τους υπόλοιπους χρήστες μέσω του chat και να μοιραστεί απόψεις και ιδέες.

Η εφαρμογή μπορεί να επεκταθεί σε πολλούς τομείς, μπορούν να προστεθούν features που μπορούν να δώσουν ουσιαστική λειτουργικότητα στους χρήστες και στα κέρδη της εφαρμογής.

Μερικά από αυτά είναι:

- Notification system
- Newsletter emails
- Analytics
- Server analytics
- Edit photos after upload
- Premium subscriptions
- Advertisements
- User tracking
- History logs
- Re-share
- Call, video call

Τέλος, έχοντας ολοκληρώσει την συγκεκριμένη εργασία με επιτυχία, προσθέτω στο προσωπικό μου πορτφόλιο, ένα ολοκληρωμένο πρότζεκτ, δομημένη από την αρχή έως το τέλος με προσωπική έρευνα. Η έρευνα αυτή μου έδωσε την δυνατότητα να δουλέψω και να μάθω νέες τεχνολογίες, ενώ ταυτόχρονα να δοκιμάσω τεχνικές και υλοποιήσεις που χρησιμοποιούνται από εταιρείες στον κλάδο.

Βιβλιογραφία

- [1] HTML, <https://developer.mozilla.org/en-US/docs/Web/HTML>, last accessed 2021-09-14
- [2] CSS, <https://developer.mozilla.org/en-US/docs/Web/CSS>, last accessed 2021-09-14
- [3] JavaScript, <https://developer.mozilla.org/en-US/docs/Web/JavaScript>, last accessed 2021-09-14
- [4] TypeScript, <https://www.typescriptlang.org/docs/>, last accessed 2021-09-14
- [5] NPM, <https://www.npmjs.com/>, last accessed 2021-09-14
- [6] Angular, <https://angular.io/docs>, last accessed 2021-09-14
- [7] Angular Material, <https://material.angular.io/>, last accessed 2021-09-14
- [8] RxJS, <https://rxjs.dev/guide/overview>, last accessed 2021-09-14
- [9] JWT, <https://jwt.io/introduction/>, last accessed 2021-09-14
- [10] C#, <https://docs.microsoft.com/en-us/dotnet/csharp/>, last accessed 2021-09-14
- [11] .NET Core, <https://docs.microsoft.com/en-us/dotnet/fundamentals/>, last accessed 2021-09-14
- [12] Entity Framework - Microsoft, <https://docs.microsoft.com/en-us/ef/>, last accessed 2021-09-14
- [13] Entity Framework, <https://entityframework.net/>, last accessed 2021-09-14
- [14] Dapper, <https://dapper-tutorial.net/>, last accessed 2021-09-14
- [15] PostgreSQL, <https://www.postgresql.org/docs/>, last accessed 2021-09-14
- [16] SignalR, <https://docs.microsoft.com/en-us/aspnet/core/signalr>, last accessed 2021-09-14
- [17] Docker, <https://docs.docker.com/>, last accessed 2021-09-14
- [18] Docker optimizations, <https://circleci.com/blog/tips-for-optimizing-docker-builds/>, last accessed 2021-09-14
- [19] Docker Hub, <https://hub.docker.com/>, last accessed 2021-09-14
- [20] Docker Compose, <https://docs.docker.com/compose/>, last accessed 2021-09-14
- [21] Kubernetes, <https://kubernetes.io/>, last accessed 2021-09-14
- [22] Maildev, <https://github.com/maildev/maildev>, last accessed 2021-09-14
- [23] Microsoft Azure, <https://azure.microsoft.com/en-us/>, last accessed 2021-09-14
- [24] Azure Blob, <https://docs.microsoft.com/en-us/azure/storage/blobs/>, last accessed 2021-09-14
- [25] Repository Pattern C#, <https://codewithshadman.com/repository-pattern-csharp>, last accessed 2021-09-14
- [26] REST, <https://www.w3.org/2001/sw/wiki/REST>, last accessed 2021-09-14
- [27] Πτυχιακή εργασία με θέμα: "Καταχώρηση βλαβών και συμβάντων σε επίπεδο δήμου μέσω διαδικτυακής εφαρμογής", <http://okeanis.lib.puas.gr/xmlui/handle/123456789/2263>, last accessed 2021-09-14
- [28] Διπλωματική εργασία με θέμα: “ΑΝΑΠΤΥΞΗ ΜΙΚΡΟΥΠΗΡΕΣΙΩΝ ΣΕ DOCKER”, <https://dspace.lib.uom.gr/bitstream/2159/20982/6/GivanoudisChristosMsc2017.pdf>, last accessed 2021-09-14
- [29] Διπλωματική Εργασία Σύστημα Ανάλυσης Συναισθήματος στο Twitter σε Πραγματικό Χρόνο με το Storm, https://dione.lib.unipi.gr/xmlui/bitstream/handle/unipi/9285/Ampla_Anneta.pdf, last accessed 2021-09-14
- [30] Πτυχιακή εργασία με θέμα: "Ανακάλυψη Πολιτιστικού Περιεχομένου βάσει Τοποθεσίας", <http://ikee.lib.auth.gr/record/282411/files/%CE%A0%CF%84%CF%85%CF%87%CE%B9%CE%B1%CE%BA%CE%AE.pdf>, last accessed 2021-09-14
- [31] S. Tanenbaum, “Δίκτυα Υπολογιστών”, 5η αμερικανική έκδοση, Κλειδάριθμος, 2010.
- [32] Α. Αλεξόπουλος και Γ. Λαγογιάννης, “Τηλεπικοινωνίες και Δίκτυα Υπολογιστών”, 8η έκδοση, 2012.

- [33] “Intelligent Cloud Computing”, Asma Al-Saidi / Rudolf Fleischer / Zakaria Maamar / Omer F. Rana,
- [34] “Cloud Computing”, N. B. Ruparelia, The MIT Press Essential Knowledge Series, 2016
- [35] “Cloud Computing: From Beginning to End”, Rafaels Ray J. CreateSpace Independent Publishing Platform, 2015