ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΕΦΑΡΜΟΓΗ ΔΙΑΔΙΚΤΥΟΥ ΚΟΙΝΩΝΙΚΗΣ ΔΙΚΤΥΩΣΗΣ ΠΟΥ ΔΙΝΕΙ ΤΗΝ ΔΥΝΑΤΟΤΗΤΑ ΚΟΙΝΟΠΟΙΗΣΗΣ ΦΩΤΟΓΡΑΦΙΩΝ ΚΑΙ ΒΙΝΤΕΟ

Διπλωματική Εργασία

Γκεβρέκης Νικόλαος

Επιβλέπων: Δασκαλοπούλου Ασπασία

Συν-επιβλέπων: Θάνος Γεώργιος

Συν-επιβλέπων: Τσουκαλάς Ελευθέριος

Βόλος 2020

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΕΦΑΡΜΟΓΗ ΔΙΑΔΙΚΤΥΟΥ ΚΟΙΝΩΝΙΚΗΣ ΔΙΚΤΥΩΣΗΣ ΠΟΥ ΔΙΝΕΙ ΤΗΝ ΔΥΝΑΤΟΤΗΤΑ ΚΟΙΝΟΠΟΙΗΣΗΣ ΦΩΤΟΓΡΑΦΙΩΝ ΚΑΙ ΒΙΝΤΕΟ

Διπλωματική Εργασία

Γκεβρέκης Νικόλαος

Επιβλέπων: Δασκαλοπούλου Ασπασία

Συν-επιβλέπων: Θάνος Γεώργιος

Συν-επιβλέπων: Τσουκαλάς Ελευθέριος

Βόλος 2020

UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

SOCIAL NETWORK WEB APPLICATION FOR SHARING PHOTOS AND VIDEOS IN THE WEB

Diploma Thesis

Gkevrekis Nikolaos

Supervisor: Daskalopoulou Aspasia

Co-Supervisor: Thanos Georgios

Co-Supervisor: Tsoukalas Eleutherios

Volos 2020

ΕΥΧΑΡΙΣΤΙΕΣ ή ΣΧΟΛΙΑ

ΥΠΕΥΘΥΝΗ ΔΗΛΩΣΗ ΠΕΡΙ ΑΚΑΔΗΜΑΪΚΗΣ ΔΕΟΝΤΟΛΟΓΙΑΣ ΚΑΙ ΠΝΕΥΜΑΤΙΚΩΝ ΔΙΚΑΙΩΜΑΤΩΝ

«Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ρητά ότι η παρούσα διπλωματική εργασία, καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας, αποτελεί αποκλειστικά προϊόν προσωπικής μου εργασίας, δεν προσβάλλει κάθε μορφής δικαιώματα διανοητικής ιδιοκτησίας, προσωπικότητας και προσωπικών δεδομένων τρίτων, δεν περιέχει έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον και πληρούν τους κανόνες της επιστημονικής παράθεσης. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή/και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής».

Ο Δηλών

(Υπογραφή)
Γκεβρέκης Νικόλαος
Ημερομηνία

ΠΕΡΙΛΗΨΗ

Η διπλωματική αυτή εργασία διαπραγματεύεται την κατασκευή μιας διαδικτυακής εφαρμογής κοινωνικής δικτύωσης. Η ανάπτυξη της εφαρμογής συμπεριλαμβάνει την υλοποίηση της διεπαφής του χρήστη με την εφαρμογή και την ανάπτυξη του διακομιστή της εφαρμογής. Γίνεται χρήση των React.js και Node.js, τεχνολογιών ανάπτυξης διαδικτυακών εφαρμογών, για την ανάπτυξη του περιβάλλοντος διεπαφής και του σέρβερ αντίστοιχα. Στόχος είναι η εκμάθηση των δύο αυτών προγραμματιστικών πλαισίων και η εξέλιξη των υπάρχοντων γνώσεων προγραμματισμού μέσω της ανάπτυξης της εφαρμογής. Στην εφαρμογή ο χρήστης έχει δικό του προφίλ που μπορεί να διαμορφώσει όπως ο ίδιος επιθυμεί αναρτώντας φωτογραφίες και βίντεο, να αναζητά άλλους χρήστες και να τους ακολουθάει ώστε να βλέπει τις αναρτήσεις τους και να σχολιάζει και να δηλώνει ότι του αρέσει οποιαδήποτε ανάρτηση. Τέλος, ο χρήστης έχει τη δυνατότητα να συνομιλήσει με οποιονδήποτε ακόλουθό του.

ABSTRACT

This thesis deals with the implementation of an online social networking web application. The development involves implementing the user interface with the application and deploying the application server. React.js and Node.js web development technologies are being used to develop the interface and server respectively. The aim is to learn these two programming frameworks and to develop existing programming knowledge through the application development. In the app, the user has their own profile that they can customize as they wish by posting photos and videos with description, searching for other users and follow them to view their posts and comment on their posts or declare that they like them. Finally, the user can chat using the instant messaging functionality with all his/her followers.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

1.1 Περιγραφή της Ιδέας

Τα τελευταία χρόνια οι εφαρμογές κοινωνικών δικτύων έχουν καταστεί αναπόσπαστο κομμάτι της καθημερινής μας ζωής. Καθημερινά ο αριθμός των χρηστών στις διάφορες εφαρμογές κοινωνικού δικτύου αυξάνεται και έρευνες δείχνουν πως δεν υπάρχουν σημάδια επιβράδυνσης σύντομα. Η κοινωνική δικτύωση έχει διεισδύσει στη ζωή μας τόσο πολύ που είναι απαραίτητο κομμάτι της καθημερινότητας, ειδικά για αυτή τη γενιά. Μερικές από τις πιο κοινές και ευρέως χρησιμοποιούμενες εφαρμογές είναι το Facebook, το Twitter και το Instagram. Η παρούσα διπλωματική θα επικεντρωθεί στο Instagram και θα προσπαθήσει να υλοποιήσει κάποιες από τις λειτουργίες του.

Βασικό χαρακτηριστικό των εφαρμογών κοινωνικής δικτύωσης είναι η δυνατότητα που παρέχουν στους χρήστες να γίνουν "φίλοι" με άλλους χρήστες, να τους "ακολουθούν" ή και να μοιράζονται μεταξύ τους κάποιο περιεχόμενο, όπως για παράδειγμα κάποια δημοσίευση που να αφορά μια εικόνα ή βίντεο, μαζί με κάποια σύντομη περιγραφή. Το Instagram σαν μέσο κοινωνικής δικτύωσης, έχει αλλάξει τη ζωή μας σε περισσότερες από μία πτυχές, με σημαντικότερα να είναι η φωτογραφία και τα μηνύματα [1].

- **Φωτογραφίες**

  Παρά τις όποιες αλλαγές περνάει το Instagram, βρίσκεται ακόμα στην κορυφή των προτιμήσεων των χρηστών για κοινή χρήση και δημοσίευση φωτογραφιών. Πολλοί ήδη ευρέως γνωστοί φωτογράφοι και ακόμη περισσότεροι νέοι επίδοξοι φωτογράφοι, άρχισαν να μοιράζονται το έργο τους σε αυτήν την πλατφόρμα. Το ποσοστό δημιουργικότητας σε συνδυασμό με τον τεράστιο αριθμό πιθανών συνεργατών παρέχουν ένα ιδανικό περιβάλλον για τους χρήστες.

  Το Instagram δεν αποτελεί όμως μόνο ένα είδος χαρτοφυλάκειου φωτογραφιών για επαγγελματίες. Τα μέσα κοινωνικής δικτύωσης είναι ένα μέρος για να παρουσιάσει κανείς τη δουλειά του, αλλά με έναν μη ανταγωνιστικό και προσωπικό τρόπο. Είναι ένας χώρος όχι μόνο για επαγγελματίες αλλά και για ερασιτέχνες, καθώς παρέχει φίλτρα για να κάνει τις φωτογραφίες πιο «επαγγελματικές».

- **Μηνύματα**

  Από την κοινή χρήση φωτογραφιών με φίλους και άλλους χρήστες έως τη συνομιλία με την εξυπηρέτηση πελατών, όλο και περισσότερα άτομα σε όλο τον κόσμο επιλέγουν και κατεβάζουν εφαρμογές ανταλλαγής μηνυμάτων για κινητές συσκευές και υπολογιστές για πολλούς λόγους. Σύμφωνα με έρευνες, οι περισσότεροι προτιμούν να στέλνουν μηνύματα όταν πρόκειται να συνομιλήσουν. Η ανταλλαγή μηνυμάτων κατατάσσεται πρώτη μεταξύ εννέα τρόπων επικοινωνίας. Περισσότερο από το 60% των νέων προτιμούν να στέλνουν μηνύματα κειμένου από το να καλούν τηλεφωνικά ή να στέλνουν email σε κάποιον, καθώς αποτελεί έναν απλούστερο τρόπο επικοινωνίας.

  Η ανταλλαγή μηνυμάτων έχει κάνει την ομαδική επικοινωνία πολύ πιο εύκολη. Τα άμεσα μηνύματα είναι απλώς ευκολότερα και απλούστερα. Όχι μόνο στις επιχειρήσεις αλλά και στην προσωπική ζωή των ανθρώπων. Η πλειοψηφία των χρηστών αλλάζει τον τρόπο που επικοινωνεί με τους άλλους προς το καλύτερο και σύμφωνα με στατιστικά το 50% αυτών ισχυρίζεται πως ο τρόπος αυτός βελτίωσε τις κοινωνικές του σχέσεις [2].

1.2 Στόχος της Διπλωματικής

Η παρούσα διπλωματική επικεντρώνεται στην ανάπτυξη μιας εφαρμογής διαδικτύου, όμοιας με αυτής του Instagram και στο σχεδιασμό ενός ανταποκρινόμενου REST API για τη μεταφόρτωση και κοινή χρήση αναρτήσεων που περιλαμβάνουν φωτογραφίες, βίντεο και σχόλια. Πιο συγκεκριμένα, ο στόχος είναι η εκμάθηση των γλωσσών προγραμματισμού και τα πλαίσια και εργαλεία που χρησιμοποιούνται για την ανάπτυξη μιας διαδικτυακής εφαρμογής, την αρχιτεκτονική του διακομιστή-πελάτη και τη δημιουργία ενός RESTful API. Η παρούσα εφαρμογή δεν παρέχει στο χρήστη φιλτράρισμα και επεξεργασία εικόνων και βίντεο.

1.3 Οργάνωση Κεφαλαίων

Στα ακόλουθα κεφάλαια θα αναλυθούν η χρησιμότητα και σημασία κάθε τεχνολογίας και

σχεδιασμού διεπαφής που χρησιμοποιήθηκαν στην εργασία αυτή.

Στο *Κεφάλαιο 1*, αναπτύσσεται η εισαγωγή, η οποία αναλύει την ιδέα και τον σκοπό της διπλωματικής.

Στο *Κεφάλαιο 2*, αναλύω το τεχνολογικό και θεωρητικό υπόβαθρο που πρέπει να έχει κάποιος για την κατανόηση της δομής και της ανάπτυξης μιας σύγχρονης εφαρμογής διαδικτύου.

Στο *Κεφάλαιο 3*, βρίσκεται η τεχνολογική επισκόπηση της εφαρμογής, συμπεριλαμβανομένης της γλώσσας προγραμματισμού που έχει χρησιμοποιηθεί, των πλαισίων backend και frontend και της βάσης δεδομένων.

Το *Κεφάλαιο 4* περιλαμβάνει όλες τις λειτουργίες που προσφέρει η εφαρμογή στους χρήστες. Εδώ εξετάζεται, επίσης, τόσο το περιβάλλον εργασίας του χρήστη όσο και οι παρεχόμενες υπηρεσίες.

Τέλος, στο *Κεφάλαιο 5* αναπτύσσονται ο επίλογος της διπλωματικής εργασίας και κάποιες πιθανές μελλοντικές επεκτάσεις.

ΚΕΦΑΛΑΙΟ 2

ΤΕΧΝΙΚΟ ΥΠΟΒΑΘΡΟ

2.1 Βασικοί Ορισμοί

Εδώ ακολουθούν ορισμένες λέξεις-κλειδιά μαζί με κάποιους ορισμούς που θα πρέπει να αναλυθούν για την κατανόηση της παρούσας διπλωματικής.

Στοίβα Λογισμικού - Software Stack.

Στην Επιστήμη των Υπολογιστών, μια στοίβα λύσεων ή στοίβα λογισμικού είναι ένα σύνολο υποσυστημάτων ή στοιχείων λογισμικού που απαιτούνται για τη δημιουργία μιας πλήρους πλατφόρμας, ώστε να μην χρειάζεται πρόσθετο λογισμικό για την υποστήριξη των εφαρμογών. Οι εφαρμογές λέγεται ότι εκτελούνται πάνω από την στοίβα λογισμικού που προκύπτει.

Για παράδειγμα, για την ανάπτυξη μιας εφαρμογής διαδικτύου, ο προγραμματιστής ορίζει τη στοίβα ως το λειτουργικό σύστημα προορισμού, τον διακομιστή ιστού, τη βάση δεδομένων και τη γλώσσα προγραμματισμού [3].

Πλήρης Ανάπτυξη Στοίβας - Full Stack Development.

Αναφέρεται στην ανάπτυξη των τμημάτων λογισμικού του πελάτη (frontend) και του διακομιστή (backend) οποιασδήποτε διαδικτυακής εφαρμογής [4].

Προγραμματιστές Εφαρμογών Διαδικτύου Πλήρους Στοίβας - Full stack Web Developers.

Οι προγραμματιστές πλήρους στοίβας διαδικτύου έχουν τη δυνατότητα να σχεδιάζουν μια πλήρη διαδικτυακή εφαρμογή και ιστότοπους. Δουλεύουν και υλοποιούν το frontend κομμάτι, το backend, τη βάση δεδομένων και τον εντοπισμό σφαλμάτων [4].

Διακομιστής - Backend.

Αναφέρεται στην ανάπτυξη διαδικτυακών εφαρμογών ή ιστότοπων από την πλευρά του διακομιστή με πρωταρχικό ρόλο την εστίαση στον τρόπο λειτουργίας του ιστότοπου. Είναι υπεύθυνο για τη διαχείριση της βάσης δεδομένων μέσω

ερωτημάτων και διάφορων αιτημάτων και API με εντολές από την πλευρά του πελάτη.

Διεπαφή - Frontend. (geeks)

Είναι το "ορατό"-"μπροστινό" μέρος του ιστότοπου ή της διαδικτυακής εφαρμογής, που είναι υπεύθυνο για την "εμπειρία" του χρήστη. Ο χρήστης αλληλεπιδρά άμεσα με το μπροστινό τμήμα της εφαρμογής ή του ιστότοπου.

API. (geeks & quora)

Ο όρος API (Application Programming Interface) σηματοδοτεί οποιαδήποτε προγραμματιστική διεπαφή (κύριος συμμετέχων σε όλη τη διαδραστικότητα). Λειτουργεί σαν ένας αγγελιοφόρος που μεταφέρει αιτήματα σε ένα σύστημα και επιστρέφει μια απάντηση στον αποστολέα μέσω απρόσκοπτης συνδεσιμότητας. Τα API χρησιμοποιούνται σε πολλές περιπτώσεις, όπως για παράδειγμα για τη λήψη δεδομένων για μια εφαρμογή ή για τη σύνδεση σε έναν απομακρυσμένο διακομιστή που διαθέτει δεδομένα ή για να επιτρέπουν σε δύο εφαρμογές να ανταλλάσσουν δεδομένα μεταξύ τους. Το API επιτρέπει την επαναχρησιμοποίηση κώδικα μέσα από την αρχή της Αφαίρεσης (Abstraction).

REST API. (wiki)

Το Representational State Transfer - REST είναι ένας τύπος αρχιτεκτονικής λογισμικού που περιλαμβάνει ένα σύνολο κανόνων που πρέπει να τηρούνται για τη δημιουργία υπηρεσιών Web. Οι υπηρεσίες Web που είναι συμβατές με το αρχιτεκτονικό στυλ REST, καλούνται RESTful Web services και παρέχουν διαλειτουργικότητα μεταξύ των συστημάτων υπολογιστών στο Διαδίκτυο. Επίσης, οι υπηρεσίες RESTful Web επιτρέπουν στα αιτούντα συστήματα να έχουν πρόσβαση και να χειρίζονται κείμενα από πόρους του διαδικτύου χρησιμοποιώντας ένα ομοιόμορφο και προκαθορισμένο σύνολο λειτουργιών χωρίς μεταβολές κατάστασης (state). Χρησιμοποιώντας ένα πρωτόκολλο χωρίς μεταβολές κατάστασης , τα συστήματα REST στοχεύουν στη γρήγορη ανάπτυξη διαδικτυακών υπηρεσιών, την διαλειτουργικότητα μεταξύ διάφορων συστημάτων που τις χρησιμοποιούν και την αξιοπιστία τους.

Πλαίσιο Λογισμικού - Software Framework. (wiki)

Στον προγραμματισμό υπολογιστών, ένα πλαίσιο λογισμικού είναι μια αφηρημένη έννοια κατά την οποία το λογισμικό που παρέχει γενική λειτουργικότητα μπορεί να συμπληρωθεί με πρόσθετο κώδικα από τον προγραμματιστή, παρέχοντας έτσι ειδικό λογισμικό για μια συγκεκριμένη εφαρμογή.

Τα Software frameworks μπορεί να περιλαμβάνουν προγράμματα υποστήριξης, μεταγλωττιστές, βιβλιοθήκες κώδικα, σύνολα εργαλείων και διεπαφές προγραμματισμού εφαρμογών (API), που συγκεντρώνουν όλα τα απαραίτητα στοιχεία για να υποστηρίξουν την ανάπτυξη μιας εφαρμογής ή συστήματος [4].

MVC.

Το model-view-controller, γνωστό ως MVC, είναι ένα σχέδιο σχεδιασμού λογισμικού που επικρατέστερα χρησιμοποιείται για την ανάπτυξη διεπαφών χρήστη, που διαιρεί τη σχετική λογική του προγράμματος σε τρία διασυνδεδεμένα στοιχεία. Αυτό γίνεται για να διαχωριστούν οι εσωτερικές αναπαραστάσεις πληροφοριών από τον τρόπο με τον οποίο οι πληροφορίες παρουσιάζονται και γίνονται αποδεκτές από τον χρήστη. Αυτό το είδος μοτίβου χρησιμοποιείται για το σχεδιασμό οποιασδήποτε διαδικτυακής εφαρμογής [5].

2.2 Εφαρμογή Διαδικτύου

Στην πληροφορική, μια εφαρμογή ιστού ή μια εφαρμογή διαδικτύου είναι ένα πρόγραμμα που εκτελεί κάποιος πελάτης (συμπεριλαμβανομένης της διεπαφής χρήστη και της λογικής του πελάτη) σε ένα πρόγραμμα περιήγησης ιστού ή φυλλομετρητή. Κάποιες από τις πιο κοινές εφαρμογές ιστού περιλαμβάνουν το ηλεκτρονικό ταχυδρομείο, διαδικτυακές εφαρμογές πώλησης προϊόντων, διαδικτυακές τραπεζικές συναλλαγές και διαδικτυακές δημοπρασίες.

Η γενική διάκριση μεταξύ μιας δυναμικής ιστοσελίδας οποιουδήποτε είδους και μιας «διαδικτυακής εφαρμογής» είναι ασαφής. Οι ιστότοποι είναι πιθανό να αναφέρονται ως "εφαρμογές ιστού" οι οποίες έχουν παρόμοια λειτουργικότητα με μια εφαρμογή λογισμικού υπολογιστή ή μια εφαρμογή για κινητά. Η HTML5 εισήγαγε συγκεκριμένη υποστήριξη γλώσσας για την πραγματοποίηση εφαρμογών που φορτώνονται ως ιστοσελίδες αλλά μπορούν να αποθηκεύουν δεδομένα τοπικά και να συνεχίζουν να

λειτουργούν ενώ είστε εκτός σύνδεσης. Οι εφαρμογές «μιας σελίδας» μοιάζουν περισσότερο με μια κοινή εφαρμογή, επειδή απορρίπτουν το πιο τυπικό παράδειγμα ιστού, δηλαδή της μετακίνησης μεταξύ διακριτών σελίδων με διαφορετικές διευθύνσεις URL. Διάφορα πλαίσια (frameworks) «μίας σελίδας» θα μπορούσαν να χρησιμοποιηθούν για την επιτάχυνση της ανάπτυξης μιας τέτοιας εφαρμογής ιστού ακόμα και για μια εφαρμογή για κινητά.

Η σύνταξη εφαρμογών ιστού απλοποιείται συχνά με τη χρήση πλαισίων εφαρμογών ιστού (frameworks). Αυτά τα πλαίσια διευκολύνουν την ταχεία ανάπτυξη εφαρμογών επιτρέποντας σε μια ομάδα ανάπτυξης να επικεντρωθεί σε πολύ συγκεκριμένα μέρη μιας εφαρμογής χωρίς να χρειάζεται να επιλύσουν κοινά θέματα ανάπτυξης, όπως η διαχείριση χρηστών. Πολλά από τα πλαίσια που χρησιμοποιούνται είναι λογισμικά ανοιχτού κώδικα (open source). Η χρήση πλαισίων εφαρμογών ιστού μπορεί συχνά να μειώσει τον αριθμό των σφαλμάτων σε ένα πρόγραμμα, κάνοντας τον κώδικα απλούστερο και επιτρέποντας σε μια ομάδα να επικεντρωθεί σε ένα πλαίσιο, ενώ μια άλλη επικεντρώνεται σε μια άλλη συγκεκριμένη λειτουργία ή ένα άλλο πλαίσιο [6].

Συνοπτικά μπορούμε να δούμε παρακάτω μια λίστα με κάποια από τα πλεονεκτήματα των εφαρμογών ιστού [7]:

- Οι εφαρμογές διαδικτύου εκτελούνται σε πολλές πλατφόρμες ανεξάρτητα από το λειτουργικό σύστημα ή τη συσκευή, εφόσον το πρόγραμμα περιήγησης είναι συμβατό.
- Όλοι οι χρήστες έχουν πρόσβαση στην ίδια έκδοση, εξαλείφοντας τυχόν προβλήματα συμβατότητας.
- Δεν είναι εγκατεστημένα στον σκληρό δίσκο, εξαλείφοντας έτσι τους περιορισμούς χώρου.
- Μειώνουν την πειρατεία λογισμικού σε εφαρμογές ιστού βάσει συνδρομών (δηλαδή SaaS).
- Μειώνουν το κόστος τόσο για την επιχείρηση όσο και για τον τελικό χρήστη, καθώς απαιτείται λιγότερη υποστήριξη και συντήρηση από την επιχείρηση και χαμηλότερες απαιτήσεις για τον υπολογιστή του τελικού χρήστη.

2.2.1 Web Application Architecture

Μία εφαρμογή χαρακτηρίζεται από την αρχιτεκτονική της. Η αρχιτεκτονική μιας εφαρμογής είναι ένα πλαίσιο που αποτελείται από τις σχέσεις και τις αλληλεπιδράσεις μεταξύ των συστατικών στοιχείων της εφαρμογής, όπως τα συστήματα ενδιάμεσου ιστού (middleware), τις διεπαφές χρήστη και τις βάσεις δεδομένων. Σε γενικές γραμμές, αυτό είναι ένα μοντέλο αλληλεπίδρασης μεταξύ των συστατικών εφαρμογών ιστού όπως ο διακομιστής ιστού, ο διακομιστής βάσης δεδομένων, το πρόγραμμα περιήγησης κλπ.

Ας το εξηγήσουμε με ένα απλό παράδειγμα ανοίγματος μιας ιστοσελίδας.

Μόλις ο χρήστης πατήσει το κουμπί μετά την πληκτρολόγηση μιας διεύθυνσης URL στη γραμμή διευθύνσεων ενός προγράμματος περιήγησης ιστού, ζητά τη συγκεκριμένη διεύθυνση ιστού. Ο διακομιστής στέλνει αρχεία στο πρόγραμμα περιήγησης ως απάντηση στο αίτημα που υποβλήθηκε. Στη συνέχεια, ο φυλλομετρητής εκτελεί αυτά τα αρχεία για να εμφανίσει τη σελίδα που ζητήθηκε. Αυτά τα αρχεία είναι συνήθως αρχεία τύπου json, αλλά θα αναλυθούν στην συνέχεια της διπλωματικής. Τέλος, ο χρήστης μπορεί να αλληλοεπιδράσει με τον ιστότοπο. Το αξιοσημείωτο εδώ είναι ο κώδικας που αναλύεται από το πρόγραμμα περιήγησης ιστού. Μια εφαρμογή ιστού λειτουργεί με πανομοιότυπο τρόπο.

Αυτός ο κώδικας μπορεί να έχει ή να μην έχει συγκεκριμένες οδηγίες που να λένε στο πρόγραμμα περιήγησης πώς να ανταποκρίνεται σε σχέση με τους διαφορετικούς τύπους εισόδων κάθε χρήστη.

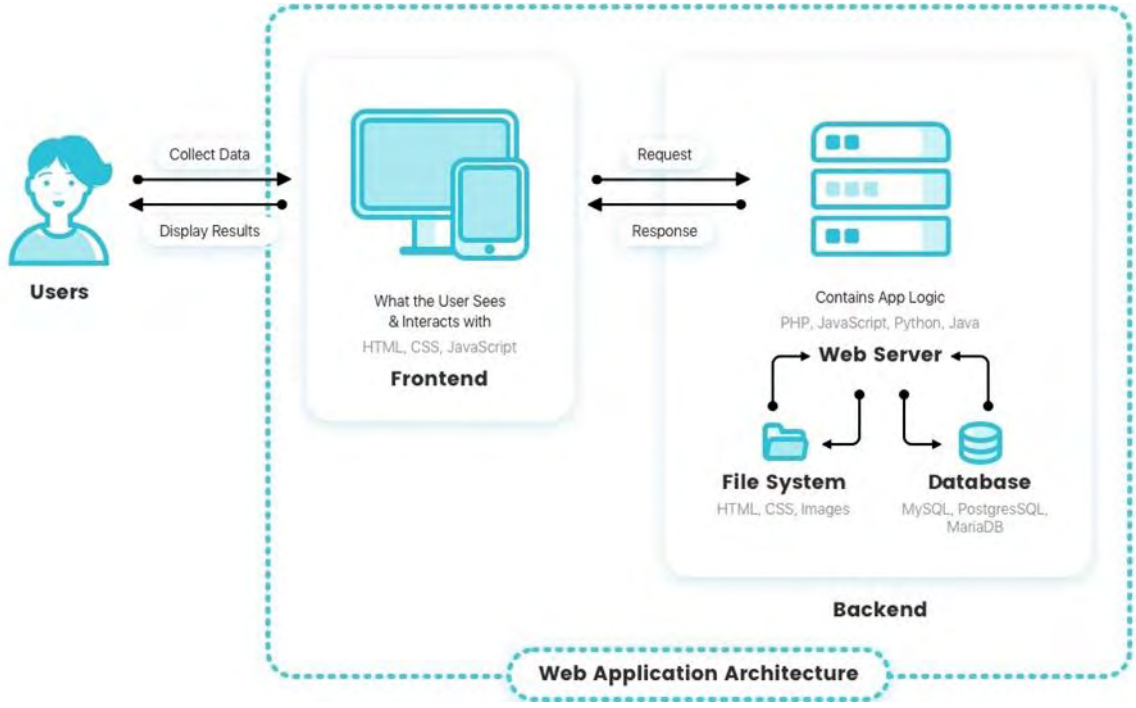
Ως εκ τούτου, μια αρχιτεκτονική εφαρμογών ιστού πρέπει να περιλαμβάνει όλα τα δευτερεύοντα συστατικά καθώς και τις εξωτερικές ανταλλαγές εφαρμογών για ολόκληρη την εφαρμογή λογισμικού, στην προαναφερθείσα περίπτωση, που είναι ιστότοπος.

Η αρχιτεκτονική εφαρμογών ιστού δεν πρέπει να ασχολείται μόνο με την αποτελεσματικότητα, αλλά και με την αξιοπιστία, την κλιμακούμενη ικανότητα, την ασφάλεια και την ευρωστία.

Στις εφαρμογές διαδικτύου, είναι ο διακομιστής έναντι του πελάτη. Στην ουσία, υπάρχουν δύο προγράμματα που εκτελούνται ταυτόχρονα:

- Ο κώδικας που βρίσκεται στο πρόγραμμα περιήγησης και ανταποκρίνεται στην είσοδο του χρήστη, εκτελείται στον διακομιστή frontend και αναφέρεται ως Κώδικας Πελάτη (Client Side Code).
- Ο κώδικας που βρίσκεται στο διακομιστή και ανταποκρίνεται σε αιτήματα HTTP, εκτελείται στο διακομιστή backend και αναφέρεται ως Κώδικας από την πλευρά του Διακομιστή (Server Side Code).

Στην *Εικόνα 2.1* παρουσιάζεται μια οπτική αναπαράσταση του τρόπου με τον οποίο ο χρήστης, το frontend και το backend αλληλοεπιδρούν μεταξύ τους.



Εικόνα 2.1: Αρχιτεκτονική Εφαρμογών Διαδικτύου.

Ένας προγραμματιστής ιστού (ομάδα ή άτομο) που αναπτύσσει μία εφαρμογή αποφασίζει για το τι θα κάνει ο κώδικας στον διακομιστή σε σχέση με τον κώδικα στο πρόγραμμα περιήγησης. Για τη σύνταξη κώδικα διακομιστή οι Java, JavaScript, Python, PHP και Ruby είναι μερικές από τις πιο συχνά χρησιμοποιούμενες γλώσσες προγραμματισμού.

Οποιοσδήποτε κώδικας που μπορεί να ανταποκριθεί σε αιτήματα HTTP έχει τη δυνατότητα να χρησιμοποιηθεί για την δημιουργία ενός διακομιστή. Ο κώδικας από την

πλευρά του διακομιστή είναι υπεύθυνος για τη δημιουργία της σελίδας που ζήτησε ο χρήστης, καθώς και για την αποθήκευση διαφορετικών τύπων δεδομένων, συμπεριλαμβανομένων των προφίλ χρήστη και άλλα. Δεν φαίνεται ποτέ στον τελικό χρήστη στον οποίο δεν επιτρέπεται η πρόσβαση στον διακομιστή. Ο χρήστης αλληλοεπιδρά αποκλειστικά μόνο με τον πρόγραμμα περιήγησης και το frontend.

Ένας συνδυασμός CSS, HTML και JavaScript χρησιμοποιείται για τη σύνταξη του κώδικα από την πλευρά του πελάτη. Αυτός ο κωδικός αναλύεται από το πρόγραμμα περιήγησης ιστού. Σε αντίθεση με ότι ισχύει για τον κώδικα από την πλευρά του διακομιστή, ο χρήστης μπορεί να δει και να τροποποιήσει τον κώδικα της πλευρά του πελάτη.

Ο κώδικας πελάτη επικοινωνεί μόνο μέσω αιτημάτων HTTP και δεν είναι σε θέση να διαβάζει απευθείας αρχεία από έναν διακομιστή. Ουσιαστικά, κάνει το διακομιστή backend σαν ένα API από το οποίο το frontend ανταλλάσσει πληροφορίες με τη μορφή αρχείων, συνήθως τύπου json [8] [9].

Η αλληλεπίδραση που φαίνεται στην *Εικόνα 2.1* έχει ως εξής:

1. Πρώτον, ο χρήστης στέλνει ένα αίτημα στο διακομιστή μέσω του Διαδικτύου.
2. Επειδή ο χρήστης μπορεί να αλληλοεπιδράσει μόνο με το frontend, αυτό είναι υπεύθυνο για την προώθηση αυτού του αιτήματος στο backend.
3. Μετά την αλληλεπίδραση με τη βάση δεδομένων, ο διακομιστής backend μπορεί να χειριστεί τα δεδομένα και να διαμορφώσει το αποτέλεσμα για το αίτημα.
4. Στη συνέχεια, αποκρίνεται πίσω στο frontend με τα αποτελέσματα.
5. Και, τέλος, το frontend παρέχει στον χρήστη τις ζητούμενες πληροφορίες εμφανίζοντας τα αποτελέσματα.

ΚΕΦΑΛΑΙΟ 3

ΤΕΧΝΟΛΟΓΙΚΗ ΕΠΙΣΚΟΠΙΣΗ

3.1 JavaScript

Η JavaScript, εν συντομία JS, είναι μια γλώσσα προγραμματισμού που βασίζεται στο πρότυπο τυποποίησης που ονομάζεται ECMAScript [10]. Η JavaScript είναι γλώσσα υψηλού επιπέδου. Συντακτικά, μοιάζει στην C και έχει αντιγράψει πολλά ονόματα και συμβάσεις της Java. Είναι βασισμένη σε πολλά διαφορετικά προγραμματιστικά παραδείγματα (multi-paradigm) και υποστηρίζει αντικειμενοστραφή προγραμματισμό αλλά και συναρτησιακό προγραμματισμό. Ένα από τα μεγαλύτερα πλεονεκτήματα της JavaScript είναι ότι δουλεύει ασύγχρονα, άρα και πιο γρήγορα σε συγκεκριμένες συνθήκες.

Μαζί με την HTML και την CSS, η JavaScript είναι μία από τις βασικές τεχνολογίες του παγκόσμιου ιστού και των φυλλομετρητών (web browser). Η JavaScript επιτρέπει τη δημιουργία διαδραστικών ιστοσελίδων και εφαρμογών και αποτελεί ένα από τα σημαντικότερα και πιο ευρέως χρησιμοποιούμενα μέσα δημιουργίας εφαρμογών ιστού. Η συντριπτική πλειονότητα των ιστότοπων και φυλλομετρητών την χρησιμοποιούν για τον προγραμματισμό της σελίδας από την πλευρά του χρήστη (client-side ή frontend server) και όλα τα μεγάλα προγράμματα περιήγησης στο διαδίκτυο (web browsers) διαθέτουν ειδικό περιβάλλον εκτέλεσης για την JavaScript.

Διαθέτει διεπαφές προγραμματισμού εφαρμογών (API) για προγραμματισμό με κείμενο, ημερομηνίες, κανονικές εκφράσεις (regular expressions), τυπικές δομές δεδομένων και το μοντέλο αντικειμένου εγγράφου (DOM) που θα μας απασχολήσει αργότερα στην ανάλυση του frontend server και της ReactJs. Ωστόσο, η ίδια η γλώσσα δεν περιλαμβάνει καμία λειτουργία για είσοδο / έξοδο (I / O), καθώς το περιβάλλον φιλοξενίας του χρήστη (συνήθως ένα πρόγραμμα περιήγησης ιστού) παρέχει αυτά τα API [11].

Αρχικά χρησιμοποιήθηκε μόνο σε προγράμματα περιήγησης ιστού, παρόλα αυτά περιβάλλοντα που μπορεί να τρέξει η JavaScript είναι πλέον ενσωματωμένα και σε σέρβερ

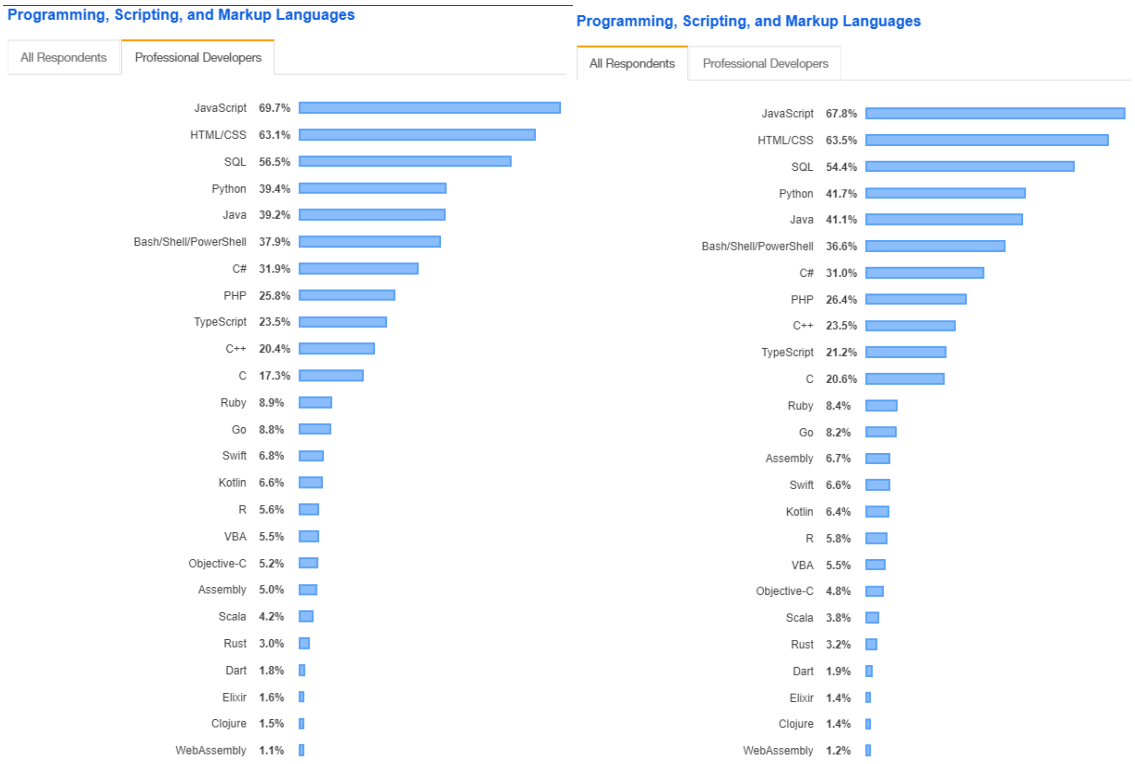
εφαρμογών διαδικτύου (server-side ή backend server) και σε άλλες εφαρμογές εκτός από εφαρμογές διαδικτύου.

Αναπτύχθηκε από τον Αμερικανό προγραμματιστή Brendan Eich της Netspace το 1995. Αρχικά μετονομάστηκε σε "Mocha" και "Livescript" και τον Δεκέμβριο του 1995 όταν κυκλοφόρησε το 3ο Beta τελικά πήρε το πλέον γνωστό όνομα, JavaScript [12].

Η JavaScript είναι επίσης γνωστή ως και η γλώσσα των φυλλομετρητών. Ο κώδικας εκτελείται από το πρόγραμμα περιήγησης και όχι από τον διακομιστή ιστού του προγράμματος περιήγησης (browser web server), περιορίζοντας έτσι την υπερφόρτωση του διακομιστή. Η μετάφρασή της είναι σχετικά απλή και ο μεταγλωττιστής της JavaScript δεν δημιουργεί σφάλματα τύπου/είδους μεταβλητών (type errors) και μπορεί να χειριστεί μεταβλητές οποιουδήποτε είδους χωρίς αρχικά να δηλωθούν, ένα χαρακτηριστικό που την καθιστά ακόμη πιο ευέλικτη ως γλώσσα προγραμματισμού. Έχει τη δυνατότητα να εκτελεί απλούς υπολογισμούς από την πλευρά του προγράμματος περιήγησης και να λαμβάνει και να στέλνει δεδομένα στον διακομιστή ασύγχρονα. Επίσης, ένα από τα σημαντικότερα χαρακτηριστικά του είναι η κληρονομικότητα. Ένα αντικείμενο μπορεί να κληρονομήσει ιδιότητες από άλλα αντικείμενα χωρίς την ανάγκη μιας κλάσης [11].

Σύμφωνα με το stackoverflow, για έβδομο συνεχόμενο έτος, η JavaScript είναι η πιο συχνά χρησιμοποιούμενη γλώσσα προγραμματισμού [13].

Επιπλέον, όπως φαίνεται στην ίδια έρευνα, η JavaScript είναι μια από τις λιγότερο πιθανές γλώσσες που θα σταματήσουν να χρησιμοποιούνται και μια από τις πιο επιθυμητές γλώσσες αυτή τη στιγμή. Συγκεκριμένα, το 67,8% όλων των ερωτηθέντων και το 69,7% των επαγγελματιών προγραμματιστών χρησιμοποιούν JavaScript. Μόνο το 33,2% από αυτούς εξέφρασαν την επιθυμία τους να σταματήσουν να εργάζονται με αυτήν και το 17,8% των προγραμματιστών που δεν προγραμματίζουν με αυτήν, έχουν εκδηλώσει ενδιαφέρον για ξεκινήσουν να δουλεύουν με την JavaScript.



Εικόνα 3.1: Στατιστικά στοιχεία της ετήσιας έρευνας stackoverflow για το 2019 για τις πιο χρησιμοποιημένες γλώσσες προγραμματισμού ανάμεσα σε επαγγελματίες (αριστερά) και όλων των ανταποκριτών της έρευνας (δεξιά).

3.1.1 Full Stack JavaScript

Αν και η JavaScript υπάρχει για πάνω από 20 χρόνια, δεν ήταν τόσο δημοφιλής λόγω της απόδοσής της και της έλλειψης συμβατότητας με τους προγραμματιστές της εποχής. Η βελτίωση της γλώσσας έκανε τη χρήση της απαραίτητη για τη δημιουργία του frontend κάθε εφαρμογής, ενώ το backend, δηλαδή ο διακομιστής, αναπτύσσοταν με τεχνολογίες όπως PHP, ASP.NET, JAVA και άλλες. Η χρήση δύο διαφορετικών τεχνολογιών, ωστόσο, καθιστά δύσκολη την ανάπτυξη μιας διαδικτυακής εφαρμογής. Αυτός είναι και ο λόγος για τον οποίο είχαν ξεκινήσει προσπάθειες για ένωση των δύο πλευρών. Αυτό ακριβώς προσπάθησε να πετύχει η JavaScript [14].

Τη λύση, λοιπόν, σε αυτό το πρόβλημα ήρθε να φέρει η ανάπτυξη του Node.js, το οποίο είναι ένα περιβάλλον ανοιχτού κώδικα JavaScript (open source JavaScript run-time environment) και επιτρέπει την χρήση της και από την πλευρά του διακομιστή (backend server). Η ανάπτυξη όλων των τμημάτων μιας εφαρμογής μόνο με JavaScript, δηλαδή μια

κοινή γλώσσα για το frontend και το backend, έχει επιτύχει καλύτερη αποτελεσματικότητα και κατανόηση του κώδικα από τους προγραμματιστές εφαρμογών διαδικτύου, χωρίς την ανάγκη δύο διαφορετικών εξειδικευμένων ομάδων για κάθε πλευρά της εφαρμογής. Επιπλέον, η χρήση του Node.js βασίζεται σε ένα μοντέλο προσανατολισμένο σε γεγονότα (event-oriented model), το οποίο καθιστά την εκτέλεση ταχύτερη από άλλες τεχνολογίες backend.

Σε αυτό το σημείο, αξίζει να αναφέρουμε εφαρμογές που χρησιμοποιούν το Node.js, που σημαίνει ότι έχουν αναπτυχθεί χρησιμοποιώντας JavaScript, είναι σχεδόν δύο φορές ταχύτερες και απαιτεί λιγότερους προγραμματιστές να συμμετάσχουν στην υλοποίηση μιας εφαρμογής από ότι η Java. Επίσης, η εφαρμογή, αναπτύχθηκε σε 33% λιγότερες γραμμές κώδικα και 40% λιγότερα αρχεία χρησιμοποιήθηκαν σε σχέση με την εφαρμογή της Java. Στην εφαρμογή Node.js είχαν ολοκληρωθεί δύο φορές περισσότερα αιτήματα ανά δευτερόλεπτο σε σχέση με την αντίστοιχη της Java και ο χρόνος απόκρισης για την ίδια σελίδα μειώθηκε κατά 5%, πράγμα που σημαίνει ότι είχε μειώσει τον χρόνο προβολής σελίδας κατά 200 δέκατα του δευτερολέπτου (milliseconds), κάτι που ενδιαφέρει κατά πολύ τον μέσο χρήστη εφαρμογών διαδικτύου και όχι μόνο [15].

3.1.2 JavaScript MVC

Τα τρία δομικά στοιχεία οποιουδήποτε συστήματος MVC είναι τα παρακάτω [16]:

- **Μοντέλο (Model)**

  Το βασικό συστατικό του προτύπου. Είναι η δυναμική δομή δεδομένων της εφαρμογής, ανεξάρτητη από τη διεπαφή χρήστη. Διαχειρίζεται άμεσα τα δεδομένα, τη λογική και τους κανόνες της εφαρμογής.

- **Προβολή (View)**

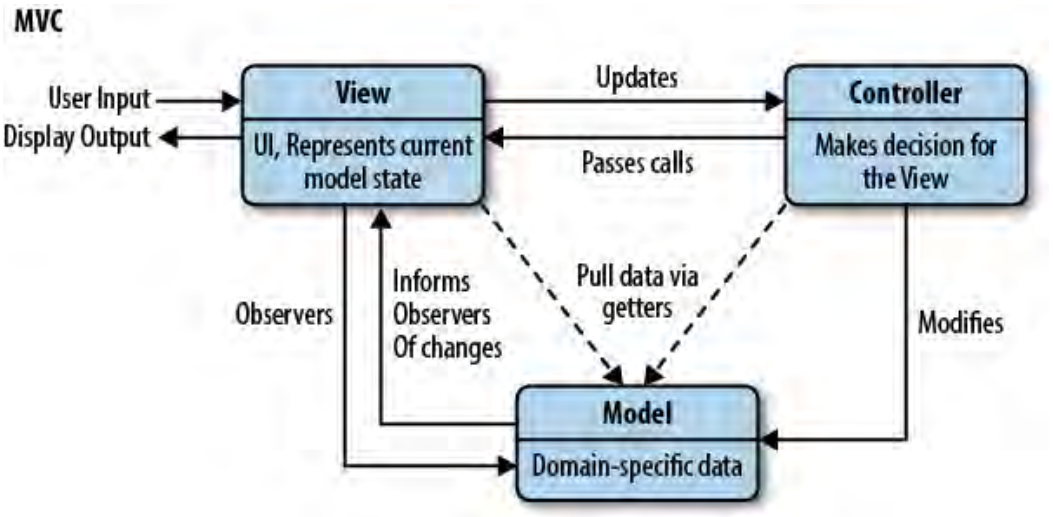
  Οποιαδήποτε αναπαράσταση πληροφοριών, όπως διαγράμματα ή πίνακες. Είναι δυνατές πολλαπλές προβολές σε διαφορετικά μέσα απεικόνισης για τις ίδιες πληροφορίες.

- **Ελεγκτής (Controller)**

  Δέχεται πληροφορίες σαν είσοδο και τις μετατρέπει σε εντολές για το μοντέλο (model) ή την προβολή (view).

Εκτός από τη διαίρεση της εφαρμογής σε αυτά τα τρία στοιχεία, ο σχεδιασμός MVC καθορίζει και τις αλληλεπιδράσεις μεταξύ τους. Το μοντέλο είναι υπεύθυνο για τη διαχείριση των δεδομένων της εφαρμογής και λαμβάνει είσοδο από τον ελεγκτή. Η προβολή αναλαμβάνει την παρουσίαση του μοντέλου σε συγκεκριμένη μορφή. Ο ελεγκτής αποκρίνεται στην είσοδο του χρήστη και εκτελεί αλληλεπιδράσεις μεταξύ των αντικειμένων του μοντέλου δεδομένων. Ο ελεγκτής, επίσης, λαμβάνει την είσοδο, την επικυρώνει (validate) προαιρετικά και μετά μεταβιβάζει την είσοδο στο μοντέλο.

Όπως και με άλλα πρότυπα λογισμικού, το MVC εκφράζει τον «πυρήνα της λύσης» σε ένα πρόβλημα, επιτρέποντάς του να προσαρμόζεται για κάθε σύστημα. Συγκεκριμένα σχέδια MVC μπορεί να διαφέρουν σημαντικά από την παραδοσιακή περιγραφή που δόθηκε προηγουμένως.



Εικόνα 3.2: Το MVC.

3.2 Ανάπτυξη του Διακομιστή - Backend Development – Node.js

Το Node.js είναι ένα περιβάλλον πραγματικού χρόνου εκτέλεσης ανοιχτού κώδικα, πολλαπλών πλατφορμών, JavaScript (open-source cross-platform runtime JavaScript environment) που εκτελεί κώδικα JavaScript εκτός προγράμματος περιήγησης ιστού. Το Node.js επιτρέπει στους προγραμματιστές να χρησιμοποιούν JavaScript για να γράφουν εργαλεία γραμμής εντολών και για δέσμες ενεργειών από διακομιστή (server-sire scripting) — εκτελώντας δέσμες ενεργειών από την πλευρά του διακομιστή για να

παράγουν δυναμικό περιεχόμενο ιστοσελίδας πριν από την αποστολή της σελίδας στο πρόγραμμα περιήγησης ιστού του χρήστη. Κατά συνέπεια, το Node.js αντιπροσωπεύει ένα παράδειγμα "JavaScript παντού", ενοποιώντας την ανάπτυξη εφαρμογών διαδικτύου γύρω από μία μόνο γλώσσα προγραμματισμού, αντί για διαφορετικές γλώσσες για τον διακομιστή και πελάτη (server- and client- side scripts). Το Node.js χρησιμοποιεί τη μηχανή V8, τη μηχανή που χρησιμοποιείται στο Google Chrome και στο Chromuim [17].

Το Node.js γράφτηκε αρχικά από τον Ryan Dahl το 2009, περίπου δεκατρία χρόνια μετά την εισαγωγή του πρώτου περιβάλλοντος JavaScript από τον διακομιστή, του LiveWire Pro Web του Netscape. Την αρχική κυκλοφορία υποστήριζε μόνο το Linux και το Mac OS X. Επικεφαλής της ανάπτυξης και συντήρησής του ήταν ο Dahl και αργότερα χρηματοδοτήθηκε από την Joyent. Τον Ιούνιο του 2011, η Microsoft και η Joyent εφάρμοσαν μια εγγενή έκδοση του Node.js. για τα Windows. Η πρώτη έκδοση Node.js που υποστηρίζει Windows κυκλοφόρησε τον Ιούλιο του 2011 [17] [18].

Τον Ιανουάριο του 2010, παρουσιάστηκε ένας διαχειριστής πακέτων για το περιβάλλον Node.js, που ονομάζεται npm [19]. Ο ρόλος του είναι να διευκολύνει τους προγραμματιστές να δημοσιεύουν και να μοιράζονται τον πηγαίο κώδικα των βιβλιοθηκών Node.js. Έχει σχεδιαστεί για να απλοποιεί την εγκατάσταση, την ενημέρωση και την απεγκατάσταση βιβλιοθηκών. Είναι, ίσως, το σημαντικότερο εργαλείο του Node.

Το Node.js επιτρέπει τη δημιουργία διακομιστών Web και εργαλείων δικτύωσης χρησιμοποιώντας JavaScript και μια συλλογή από "ενότητες" που χειρίζονται διάφορες βασικές λειτουργίες. Οι ενότητες παρέχονται για I / O συστήματος αρχείων, δικτύωση, δυαδικά δεδομένα (buffer), συναρτήσεις κρυπτογράφησης, ροές δεδομένων και άλλες βασικές λειτουργίες. Οι ενότητες του Node.js χρησιμοποιούν ένα API που έχει σχεδιαστεί για να μειώσει την πολυπλοκότητα της σύνταξης εφαρμογών διακομιστή.

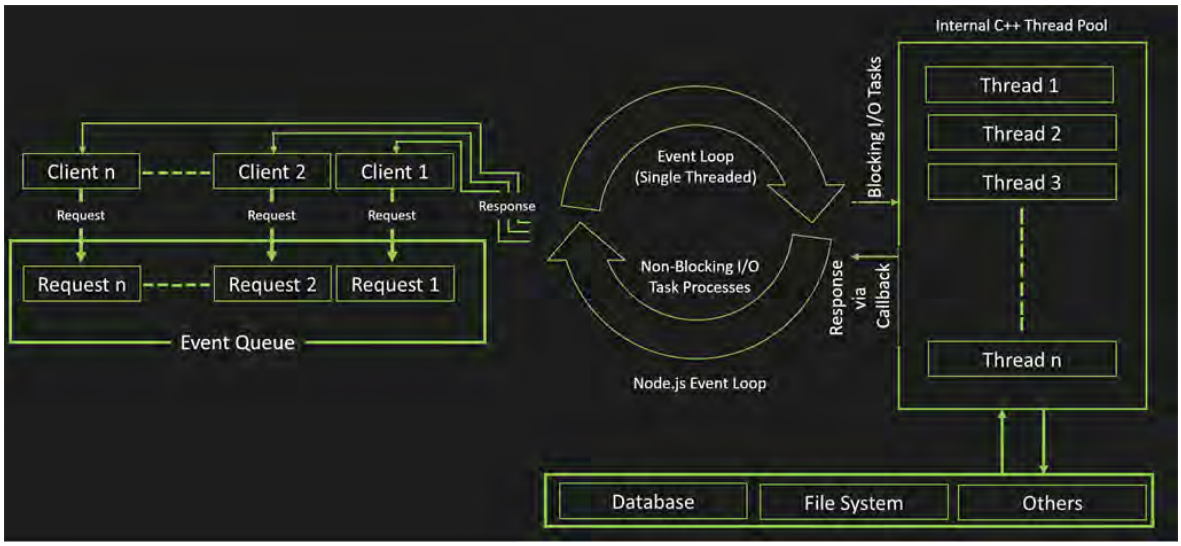
Η JavaScript είναι η μόνη γλώσσα που υποστηρίζει εγγενώς το Node.js, αλλά είναι διαθέσιμες πολλές γλώσσες που μπορούν να μεταφραστούν σε αυτήν (compile-to-JS). Το Node.js χρησιμοποιείται κυρίως για τη δημιουργία προγραμμάτων δικτύου, όπως διακομιστές Web. Η πιο σημαντική διαφορά μεταξύ Node.js και PHP είναι ότι οι περισσότερες λειτουργίες στο PHP μπλοκάρουν μέχρι την ολοκλήρωσή τους, που σημαίνει ότι οι εντολές εκτελούνται μόνο μετά την ολοκλήρωση προηγούμενων εντολών (σειριακά), ενώ οι λειτουργίες Node.js τρέχουν ασύγχρονα, πράγμα που σημαίνει ότι οι εντολές

εκτελούνται ταυτόχρονα ή ακόμη και παράλληλα και χρησιμοποιούν επιστροφές κλήσεων για ολοκλήρωση ή αστοχία σήματος (callback functions).

Το Node.js φέρνει τον προγραμματισμό βάσει συμβάντων (event-driven programming) σε διακομιστές ιστού, επιτρέποντας την ανάπτυξη γρήγορων διακομιστών ιστού σε JavaScript. Οι προγραμματιστές μπορούν να δημιουργήσουν επεκτάσιμους διακομιστές χωρίς να χρησιμοποιούν νήματα, χρησιμοποιώντας ένα απλοποιημένο μοντέλο προγραμματισμού βάσει συμβάντων που χρησιμοποιεί επιστροφές κλήσεων για να σηματοδοτήσει την ολοκλήρωση μιας εργασίας. Το Node.js συνδέει την ευκολία μιας γλώσσας δέσμης ενεργειών (script language) με τη δυναμική που προσφέρει ο προγραμματισμός δικτύων σε Unix.

Το Node.js λειτουργεί σε ένα νήμα, χρησιμοποιώντας κλήσεις εισόδου / εξόδου που δεν μπλοκάρουν, επιτρέποντάς του να υποστηρίζει δεκάδες χιλιάδες ταυτόχρονες συνδέσεις χωρίς να επιβαρύνεται με το κόστος της αλλαγής νήματος. Ο σχεδιασμός της κοινής χρήσης ενός νήματος μεταξύ όλων των αιτημάτων προορίζεται για τη δημιουργία πολύ ταυτόχρονων εφαρμογών, όπου οποιαδήποτε λειτουργία που εκτελεί I / O πρέπει να χρησιμοποιεί μια συνάρτηση σε μορφή επιστροφής κλήσης (callback functions). Για να φιλοξενήσει τον βρόχο συμβάντος με ένα νήμα, το Node.js χρησιμοποιεί τη βιβλιοθήκη libuv - η οποία, με τη σειρά της, χρησιμοποιεί μια ομάδα νημάτων σταθερού μεγέθους που χειρίζεται ορισμένες από τις μη μπλοκαρισμένες ασύγχρονες λειτουργίες εισόδου / εξόδου.

Μια ομάδα νημάτων χειρίζεται την εκτέλεση παράλληλων εργασιών στο Node.js. Η κύρια λειτουργία νήματος καλεί αναρτήσεις εργασιών στην ουρά κοινόχρηστων εργασιών, η οποία τραβάει και εκτελεί νήματα από την ομάδα νημάτων. Ενσωματωμένες λειτουργίες συστήματος που δεν μπλοκάρουν, όπως η δικτύωση, μεταφράζονται σε υποδοχές μη αποκλεισμού από πλευράς πυρήνα, ενώ εγγενώς οι λειτουργίες συστήματος μπλοκαρίσματος, όπως το αρχείο I / O, εκτελούνται με κατάλληλο τρόπο στα δικά τους νήματα. Όταν ένα νήμα της ομάδας νημάτων ολοκληρώνει μια εργασία, ενημερώνει το κύριο νήμα αυτού, το οποίο με τη σειρά του ξυπνά και εκτελεί την καταχωρημένη συνάρτηση από την επιστροφή κλήσης [17].



Εικόνα 3.3: Το οικοσύστημα του Node.js.

3.2.1 Node Package Manager

Το npm (αρχικά για το Node Package Manager) είναι ένας διαχειριστής πακέτων για τη γλώσσα προγραμματισμού JavaScript. Είναι ο προεπιλεγμένος διαχειριστής πακέτων για το περιβάλλον εκτέλεσης JavaScript Node.js [20]. Αποτελείται από μια γραμμή εντολών, που ονομάζεται επίσης npm, και μια ηλεκτρονική βάση δεδομένων δημόσιων και ιδιωτικών πακέτων, που ονομάζεται αρχείο πακέτων npm (npm registry). Η πρόσβαση στο αρχείο πακέτων γίνεται μέσω της γραμμής εντολών και τα διαθέσιμα πακέτα μπορούν να αναζητηθούν μέσω του ιστοτόπου του npm. Το npm είναι το μεγαλύτερο αρχείο λογισμικού (Software Registry) στον κόσμο με περισσότερα από 800.000 πακέτα κώδικα και είναι ανοιχτό για όλους. [21]

Περιλαμβάνεται ως προτεινόμενο αρχείο λογισμικού στο πρόγραμμα εγκατάστασης του Node.js. Το npm αποτελείται από μια γραμμής εντολών που αλληλεπιδρά με ένα απομακρυσμένο αρχείο. Επιτρέπει στους χρήστες να χρησιμοποιούν και να διανέμουν λειτουργικές μονάδες JavaScript που είναι διαθέσιμες στο αρχείο. Τα πακέτα στο αρχείο είναι σε μορφή CommonJS και περιλαμβάνουν ένα αρχείο μεταδεδομένων σε μορφή JSON. Το αρχείο δεν περιλαμβάνει κάποια διαδικασία ελέγχου για την υποβολή πακέτων, πράγμα που σημαίνει ότι τα πακέτα που βρέθηκαν μπορεί να είναι χαμηλής ποιότητας, ανασφαλή ή και κακόβουλα. Αντ 'αυτού, το npm βασίζεται σε αναφορές χρηστών για κατάργηση των πακέτων που παραβιάζουν πολιτικές. Εκθέτει στατιστικά στοιχεία που

περιλαμβάνουν τον αριθμό των λήψεων και τον αριθμό των πακέτων που εξαρτώνται για να βοηθήσουν τους προγραμματιστές να κρίνουν την ποιότητα των πακέτων [22].

Τέλος, το npm μπορεί να διαχειριστεί πακέτα που είναι τοπικές εξαρτήσεις ενός συγκεκριμένου έργου, καθώς και εργαλεία JavaScript που έχουν εγκατασταθεί στον υπολογιστή όχι μόνο τοπικά. Όταν χρησιμοποιείται ως διαχειριστής εξάρτησης για ένα τοπικό έργο, το npm μπορεί να εγκαταστήσει, με μία εντολή, όλες τις εξαρτήσεις ενός έργου μέσω του αρχείου package.json. Στο αρχείο package.json, κάθε εξάρτηση μπορεί να καθορίσει μια σειρά έγκυρων εκδόσεων κάθε εξάρτησης επιτρέποντας στους προγραμματιστές να ενημερώσουν αυτόματα τα πακέτα τους, ενώ ταυτόχρονα αποφεύγουν ανεπιθύμητες αλλαγές. Το npm παρέχει επίσης κάποια εργαλεία που επιτρέπουν στους προγραμματιστές να επισημάνουν τα πακέτα τους με μια συγκεκριμένη έκδοση. Παρέχει επίσης το αρχείο package-lock.json που έχει την καταχώριση της ακριβούς έκδοσης που χρησιμοποιείται από το έργο μετά την αξιολόγηση της σημασιολογικής έκδοσης στο package.json.

3.2.2 Express.js

Το Express.js, ή πιο απλά το Express, είναι ένα πλαίσιο εφαρμογής διαδικτύου (application framework) για το Node.js, το οποίο κυκλοφόρησε ως δωρεάν και ανοιχτού κώδικα λογισμικό βάσει της άδειας MIT. Έχει σχεδιαστεί για τη δημιουργία εφαρμογών διαδικτύου και API. Έχει χαρακτηριστεί ως το στάνταρ πλαίσιο διακομιστή για το Node.js [23].

Με μια πληθώρα μεθόδων που χρησιμοποιούνται στο HTTP πρωτόκολλο και συναρτήσεις middleware, η δημιουργία ενός ισχυρού API είναι γρήγορη και εύκολη με την χρήση του Express. Παρέχει ένα λεπτό επίπεδο βασικών δυνατοτήτων εφαρμογών διαδικτύου, χωρίς να αποκρύπτει τις δυνατότητες του Node.js.

Μερικά από τα βασικά χαρακτηριστικά του είναι η δυνατότητα και η χρήση των λειτουργιών συναρτήσεων middleware για την απόκριση σε αιτήματα HTTP, ο καθορισμός ενός πίνακα δρομολόγησης που χρησιμοποιείται για την εκτέλεση διαφορετικών ενεργειών με βάση τη μέθοδο HTTP και τη διεύθυνση URL και το γεγονός ότι επιτρέπει τη δυναμική απόδοση σελίδων HTML για τη μετάδοση ορισμάτων σε πρότυπα [24].

Το Middleware είναι μια συνάρτηση με τη μορφή συνάρτησης (req, res, next). Αναλύοντας τις τρεις μεταβλητές έχουμε το "req" που είναι το αίτημα HTTP που αποστέλλεται από το πρόγραμμα περιήγησης, το "res" που είναι η απόκριση από τον διακομιστή backend (ο διακομιστής Express) και το "next" που είναι η συνάρτηση επανάκλησης. Οι λειτουργίες Middleware καλούν την επόμενη συνάρτηση middleware, αφού πρώτα εκτελέσουν και ανταποκριθούν σε ένα αίτημα.

Κάθε εφαρμογή λαμβάνει αιτήματα HTTP με βάση τη διεύθυνση URL και πιθανώς άλλες πληροφορίες που περιέχονται στα δεδομένα που αποστέλλονται και καθορίζει ποια πρέπει να είναι η ενέργειά της. Ανάλογα με την απαιτούμενη ενέργεια, η εφαρμογή θα πρέπει να ικανοποιήσει το αίτημα είτε αλληλοεπιδρώντας με τη βάση δεδομένων ή εκτελώντας άλλες εργασίες. Η εφαρμογή θα επιστρέψει το αποτέλεσμα δημιουργώντας μια δυναμική σελίδα HTML ως απόκριση.

Το Express καταφέρνει να εκτελέσει όλη αυτή τη διαδικασία με έναν απλό τρόπο, δημιουργώντας πολύ απλοϊκές διαδρομές. Συνδυάζει ένα αίτημα HTTP με μία διαδρομή, είτε πρόκειται για διαδρομή GET, POST, PUT ή DELETE. Συνήθως έχει την ακόλουθη μορφή [24]:

app.method (διαδρομή, χειριστής);

app: είναι μια στιγμή του Express.

μέθοδος: είναι ένα αίτημα HTTP και μπορεί να είναι ένα από τα ακόλουθα: GET, POST, PUT ή DELETE.

διαδρομή: είναι η διαδρομή που προκύπτει από την διεύθυνση url.

χειριστής: είναι ο ελεγκτής που εκτελείται στη συγκεκριμένη διαδρομή (url).

3.2.3 Socket.io

Το Socket.IO είναι μια βιβλιοθήκη JavaScript για εφαρμογές διαδικτύου σε πραγματικό χρόνο [25]. Επιτρέπει σε πραγματικό χρόνο, αμφίδρομη επικοινωνία μεταξύ πελατών και διακομιστών Web. Διαθέτει δύο μέρη: μια βιβλιοθήκη από την πλευρά του πελάτη που εκτελείται στο πρόγραμμα περιήγησης και μια βιβλιοθήκη από την πλευρά του διακομιστή για το Node.js. Και τα δύο στοιχεία έχουν ένα σχεδόν ίδιο API. Όπως το Node.js, βασίζεται σε συμβάντα (event-driven).

Το Socket.IO χρησιμοποιεί κυρίως το πρωτόκολλο WebSocket. Αν και μπορεί να χρησιμοποιηθεί ως απλό περιτύλιγμα (wrapper) για το WebSocket, παρέχει πολλές ακόμη δυνατότητες, όπως μετάδοση σε πολλές υποδοχές (sockets) , αποθήκευση δεδομένων που σχετίζονται με κάθε ξεχωριστό πελάτη και ασύγχρονη είσοδο και έξοδο I / O [25].

Παρέχει τη δυνατότητα δημιουργίας εφαρμογών παρουσίασης αναλυτικών στοιχείων σε πραγματικό χρόνο, δυαδικής ροής, άμεσων μηνυμάτων και άλλα και διαχειρίζεται τη σύνδεση με διαφάνεια.

Το Socket.IO δεν είναι βιβλιοθήκη WebSocket με εναλλακτικές επιλογές σε άλλα πρωτόκολλα πραγματικού χρόνου. Είναι μια προσαρμοσμένη εφαρμογή πρωτοκόλλου μεταφοράς σε πραγματικό χρόνο πάνω από άλλα πρωτόκολλα πραγματικού χρόνου. Ένας διακομιστής υλοποίησης Socket.IO δεν μπορεί να συνδεθεί με έναν πελάτη που δεν χρησιμοποιεί Socket.IO WebSocket, δηλαδή απαιτείται η χρήση βιβλιοθηκών Socket.IO τόσο από πλευράς πελάτη όσο και από διακομιστή.

3.2.4 Multer

Το Multer είναι μία συνάρτηση middleware για το Node.js / Express για το χειρισμό πολλαπλών στοιχείων / φόρμας-δεδομένων, το οποίο χρησιμοποιείται κυρίως για τη μεταφόρτωση αρχείων. Το Multer δίνει την επιλογή αποθήκευσης αρχείων στο δίσκο. Αποδέχεται ένα αντικείμενο επιλογών, όπου το πιο βασικό από τις επιλογές αυτές είναι ο προορισμός (dest), που προσδιορίζει στο Multer πού να ανεβάσει τα αρχεία. Μπορεί να δεχτεί ένα μόνο αρχείο ή μια σειρά αρχείων [26] [27].

Κάθε αρχείο που ανεβαίνει / αποθηκεύεται μέσω του Multer περιέχει τις ακόλουθες πληροφορίες [27]:

Key (κλειδί)	Description (περιγραφή του κλειδιού)	Note (σημειώσεις)
fieldname	Το όνομα του πεδίου που καθορίζεται στη φόρμα	
originalname	Το όνομα του αρχείου στον υπολογιστή του χρήστη	
encoding	Ο τύπος κωδικοποίησης του αρχείου	
mimetype	Ο τύπος mime του αρχείου	
size	Το μέγεθος του αρχείου σε bytes	
destination	Ο φάκελος στον οποίο έχει αποθηκευτεί το αρχείο	DiskStorage
filename	Το όνομα του αρχείου εντός του φακέλου που ορίστηκε προηγουμένως	DiskStorage
path	Η πλήρης διαδρομή προς το αρχείο που μεταφορτώθηκε	DiskStorage
buffer	Ένα buffer ολόκληρου του αρχείου	MemoryStorage

Πίνακας 3.1: Τα κλειδιά ενός αρχείου που μεταφορτώνεται μέσω Multer και η περιγραφή τους.

3.3 Ανάπτυξη της Διεπαφής - Frontend Development – React.js

Η React (επίσης γνωστή ως React.js ή ReactJS) είναι μια βιβλιοθήκη JavaScript για τη δημιουργία διεπαφών χρήστη (user interface). Συντηρείται από το Facebook και μια κοινότητα μεμονωμένων προγραμματιστών και εταιρειών. Δημιουργήθηκε από τον Jordan Walke, έναν μηχανικό λογισμικού στο Facebook, ο οποίος κυκλοφόρησε ένα πρώιμο πρωτότυπο της React που ονομάζεται "FaxJS". Επηρεάστηκε από το XHP, μια βιβλιοθήκη HTML της PHP. Για πρώτη φορά αναπτύχθηκε στο News Feed του Facebook το 2011 και αργότερα στο Instagram το 2012. Ήταν ανοιχτού κώδικα μέχρι το συνέδριο JSConf US τον Μάιο του 2013 [28] [29].

Η React μπορεί να χρησιμοποιηθεί ως βάση για την ανάπτυξη μιας σελίδας ή εφαρμογών για κινητά. Ωστόσο, ασχολείται μόνο με την απόδοση δεδομένων στο DOM και επομένως η δημιουργία εφαρμογών React απαιτεί συνήθως τη χρήση πρόσθετων βιβλιοθηκών για διαχείριση της κατάστασης (state) και της δρομολόγησης (routing). Το React χρησιμοποιεί

το Virtual DOM, το οποίο θα αναλύσουμε περαιτέρω αργότερα στην διπλωματική, παράλληλα με τις μεθόδους Lifecycle και το JSX.

Η React αποτελείται από οντότητες που ονομάζονται συστατικά ή, καλύτερα, components. Τα components μπορούν να αποδοθούν σε ένα συγκεκριμένο στοιχείο στο DOM χρησιμοποιώντας τη βιβλιοθήκη React DOM. Κατά την απόδοση ενός component, κάποιος μπορεί να περάσει τιμές που είναι γνωστές ως "props". Οι δύο πρωταρχικοί τρόποι δήλωσης components στο React είναι είτε μέσω αρχείων δηλωμένων ως συναρτήσεις (functional component) ή αρχείων που βασίζονται σε κλάσεις (class-based components). Τα αρχεία δηλωμένα ως συναρτήσεις λειτουργούν ως κοινές συναρτήσεις (functions) και στο τέλος επιστρέφουν κώδικα JSX. Τα αντίστοιχα αρχεία που βασίζονται σε κλάσεις δηλώνονται χρησιμοποιώντας κλάσεις που ορίζονται στο ES6, οι οποίες είναι επίσης γνωστά ως "stateful" αρχεία και components, επειδή η κατάστασή τους μπορεί να διατηρεί τιμές σε ολόκληρο το στοιχείο και μπορεί να μεταδοθεί σε άλλα components μέσω των ορισμάτων props [29].

Ο κώδικας της React μπορεί να συνδυάσει και τις τρεις γνωστές τεχνολογίες frontend, HTML, CSS και JavaScript. Ένα άλλο χαρακτηριστικό της React είναι η ασύγχρονη λειτουργία μιας και χρησιμοποιεί JavaScript. Οι εφαρμογές των components του συνεχίζουν να λειτουργούν και να φορτώνονται ακόμη και αν ένα κομμάτι ενός component δεν έχει ακόμα φορτωθεί.

3.3.1 State

Τα στοιχεία (components) της React έχουν ενσωματωμένο ένα αντικείμενο που διαχειρίζεται την κατάσταση (state). Στο αντικείμενο κατάστασης αποθηκεύονται τιμές που ανήκουν στο component. Όταν αλλάζει το state, το component αποδίδεται ξανά προς εμφάνιση.

Το state αρχικοποιείται στον κατασκευαστή (contrsuctor) και περιέχει όσες ιδιότητες επιθυμεί ο προγραμματιστής. Αναφερόμενος στο state οπουδήποτε μέσα στο component με τη σύνταξη *this.state.property* και *this.setState()*, ο προγραμματιστής μπορεί να χειριστεί και να αλλάξει το state ανάλογα με τις περιστάσεις. Όπως έχω ήδη αναφέρει, αλλάζοντας την τιμή μιας μεταβλητής του state, η React θα αποδώσει εκ νέου το

component προς εμφάνιση, πράγμα που σημαίνει ότι η έξοδος θα αλλάξει σύμφωνα με τις νέες τιμές [30].

3.3.2 Virtual DOM

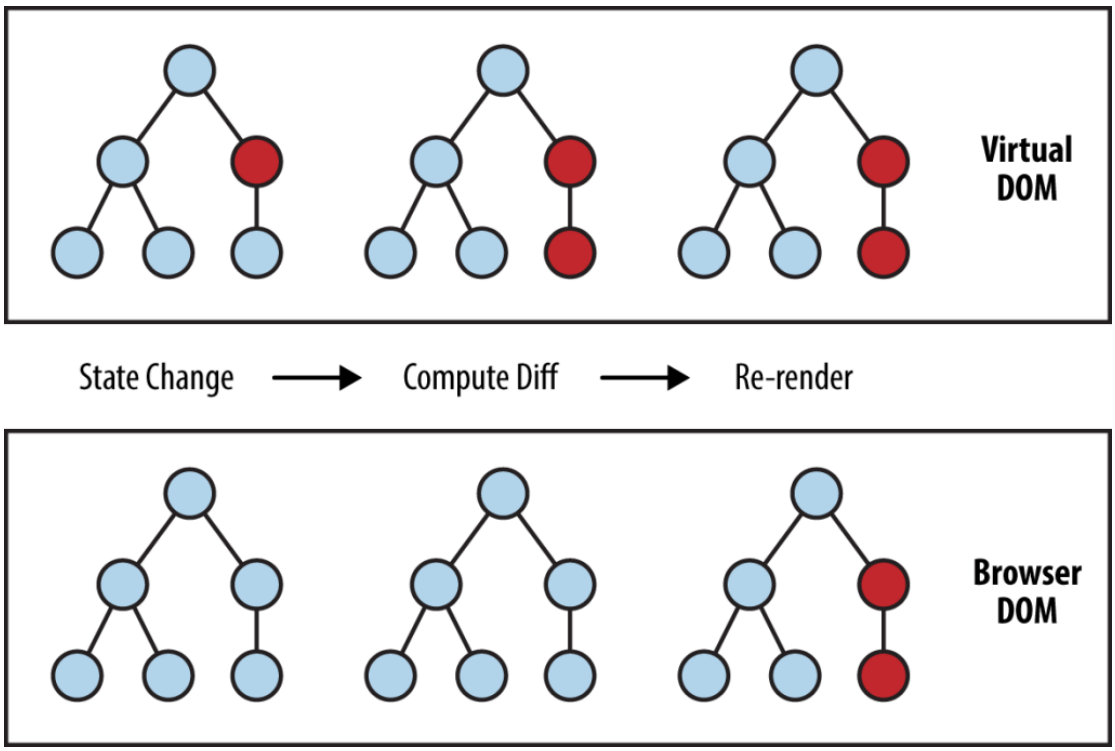
Πρώτα απ 'όλα ας δούμε τι είναι το Real DOM. Το DOM σημαίνει "Document Object Model" [31]. Το DOM με απλές λέξεις αντιπροσωπεύει τη διεπαφή του χρήστη μιας εφαρμογής. Κάθε φορά που υπάρχει αλλαγή στην κατάσταση διεπαφής του χρήστη της εφαρμογής, το DOM ενημερώνεται για να αντιπροσωπεύει αυτήν την αλλαγή. Η συνεχής αλλαγή του DOM επηρεάζει την απόδοση, προφανώς, καθιστώντας την πιο αργή.

Αλλά τι κάνει το DOM πιο αργό; Το DOM αντιπροσωπεύεται ως μία δομή δεδομένων δέντρου. Εξαιτίας αυτού, οι αλλαγές και οι ενημερώσεις στο DOM είναι γρήγορες. Ωστόσο, μετά την αλλαγή, το ενημερωμένο component και τα παιδιά του πρέπει να επανεμφανιστούν για να ενημερώσουν τη διεπαφή του χρήστη της εφαρμογής. Η εκ νέου απόδοση ή επανεμφάνιση του περιβάλλοντος είναι αυτό που το καθιστά αργό. Επομένως, όσο περισσότερα components χρησιμοποιούνται σε ένα UI (διεπαφή – user interface), τόσο πιο ακριβές θα είναι οι ενημερώσεις του DOM.

Εκεί έρχεται η έννοια του εικονικού DOM το οποίο έχει σημαντικά καλύτερη απόδοση από το πραγματικό DOM. Το εικονικό DOM είναι μόνο μια εικονική αναπαράσταση του DOM. Κάθε φορά που αλλάζει το state της εφαρμογής, το εικονικό DOM ενημερώνεται αντί του πραγματικού DOM. Όταν προστίθενται νέα στοιχεία στο UI, δημιουργείται ένα εικονικό DOM, το οποίο αντιπροσωπεύεται ως ένα δέντρο. Κάθε στοιχείο είναι ένας κόμβος σε αυτό το δέντρο. Εάν αλλάξει το state οποιουδήποτε από αυτά τα στοιχεία, δημιουργείται ένα νέο εικονικό δέντρο DOM. Αυτό το δέντρο συγκρίνεται με το προηγούμενο εικονικό δέντρο DOM. Μόλις γίνει αυτό, το εικονικό DOM υπολογίζει την καλύτερη δυνατή μέθοδο για να πραγματοποιήσει αυτές τις αλλαγές στο πραγματικό DOM. Ως εκ τούτου, υπάρχει μείωση του κόστους απόδοσης της ενημέρωσης του πραγματικού DOM.

Στην *Εικόνα 3.4*, οι κόκκινοι κύκλοι αντιπροσωπεύουν τους κόμβους που έχουν αλλάξει. Αφού αλλάξει ένα εικονικό DOM, στη συνέχεια υπολογίζεται η διαφορά μεταξύ της προηγούμενης έκδοσης του εικονικού δέντρου DOM και του τρέχοντος. Στη συνέχεια, ολόκληρο το γονικό υποδέντρο μεταδίδεται ξανά για να δώσει το ενημερωμένο

περιβάλλον στον χρήστη. Αυτό το ενημερωμένο δέντρο στη συνέχεια ενημερώνεται κατά παρτίδες στο πραγματικό DOM.



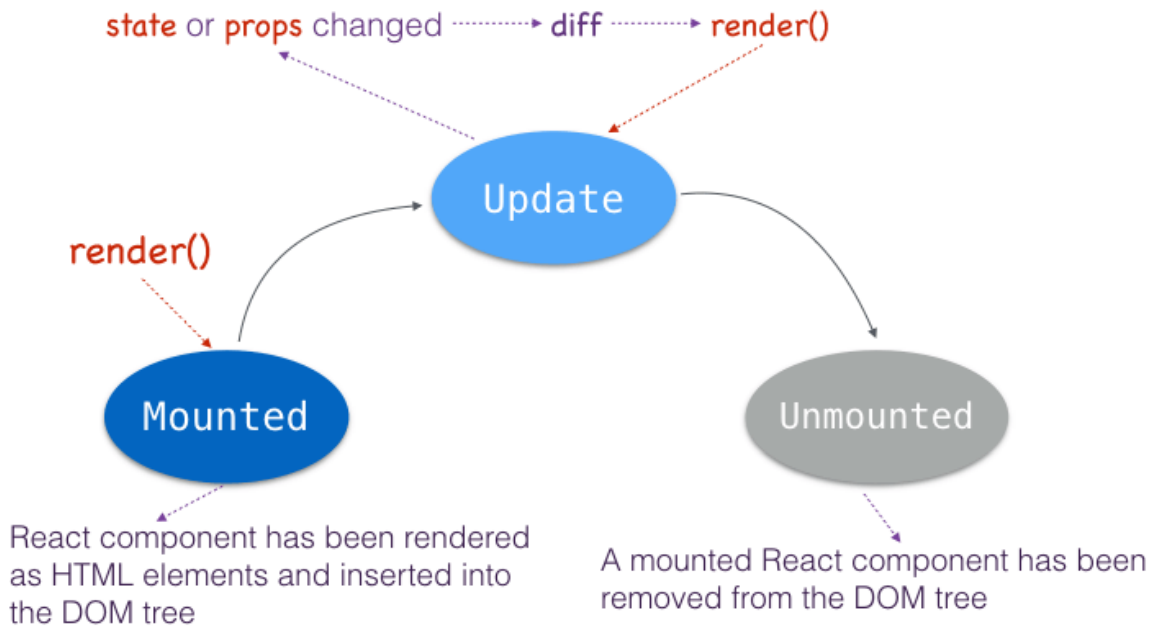
Εικόνα 3.4: Το εικονικό δέντρο DOM και η διαδικασία εύρεσης διαφορών σε περίπτωση αλλαγής του component.

Η διαδικασία, συνοπτικά, έχει ως εξής:

1. Στη React κάθε κομμάτι UI είναι ένα στοιχείο και κάθε στοιχείο έχει ένα state. Η React ακολουθεί το παρατηρήσιμο μοτίβο και ακολουθεί τις αλλαγές του state.
2. Όταν αλλάζει το state ενός component, η React ενημερώνει το εικονικό δέντρο DOM.
3. Μόλις ενημερωθεί το εικονικό DOM, η React στη συνέχεια συγκρίνει την τρέχουσα έκδοση του εικονικού DOM με την προηγούμενη έκδοση του εικονικού DOM.
4. Μόλις η React γνωρίσει ποια εικονικά αντικείμενα DOM έχουν αλλάξει, τότε ενημερώνει μόνο αυτά τα αντικείμενα, στο πραγματικό DOM.

3.3.3 Lifecycle Methods

Κάθε component της React περνά από έναν κύκλο ζωής εκδηλώσεων. Σκεφτείτε ότι τα components περνούν από έναν κύκλο γέννησης, ανάπτυξης και θανάτου που μπορούν να μεταφραστούν σε προσάρτηση, ενημέρωση και απομάκρυνση, αντίστοιχα.



Εικόνα 3.5: Ο κύκλος ζωής ενός component της React.

Παρακάτω ακολουθεί μια λίστα με τις πιο κοινές μεθόδους κύκλου ζωής της React [32].

- **render()**

  Η μέθοδος *render()* είναι η πιο χρησιμοποιημένη μέθοδος κύκλου ζωής και αυτό συμβαίνει επειδή το *render()* είναι η μόνη απαιτούμενη μέθοδος σε ένα component κλάσης στη React. Όπως υποδηλώνει το όνομα χειρίζεται την απόδοση του component στο περιβάλλον χρήστη. Συμβαίνει κατά την τοποθέτηση και ενημέρωση του component.

- **componentDidMount()**

  Μόλις το component έχει τοποθετηθεί και είναι έτοιμο, το *componentDidMount()* καλείται. Αυτό είναι πιθανώς το καλύτερο μέρος για να συμβούν κλήσεις API και να φορτώσουν δεδομένα από ένα απομακρυσμένο τελικό σημείο, τον διακομιστή backend.

  Σε αντίθεση με τη μέθοδο *render()*, το *componentDidMount()* επιτρέπει τη χρήση του *setState()*.

- **componentDidUpdate()**

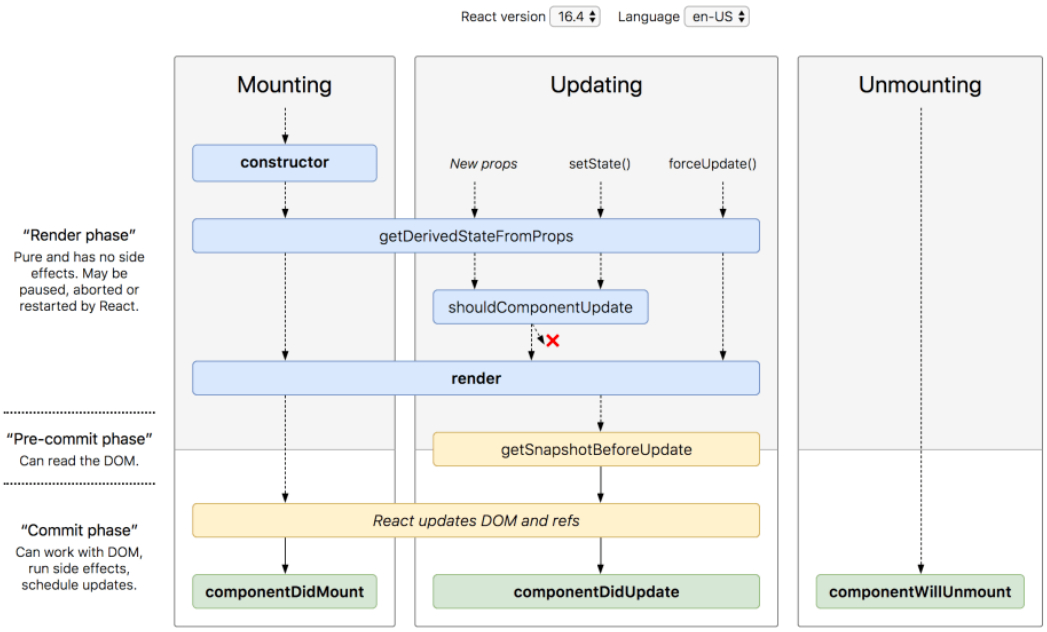
  Αυτή η μέθοδος επικαλείται, προφανώς, μετά την ενημέρωση. Η πιο συνηθισμένη περίπτωση χρήσης για τη μέθοδο *componentDidUpdate()* είναι η ενημέρωση του DOM όταν συμβούν αλλαγές στο state.

  Αυτή η μέθοδος επιτρέπει την κλήση του *setState()*, αλλά ο προγραμματιστής θα πρέπει να είναι προσεκτικός καθώς η εσφαλμένη χρήση του *setState()* μπορεί να δημιουργήσει έναν άπειρο βρόχο.

- **componentWillUnmount()**

  Όπως υποδηλώνει το όνομα, αυτή η μέθοδος κύκλου ζωής καλείται λίγο πριν το στοιχείο αποσυνδεθεί και καταστραφεί. Αυτό θα ήταν το σωστό σημείο για τυχόν ενέργειες καθαρισμού.

  Σε σύγκριση με τις δύο προηγούμενες μεθόδους, το *componentWIllUnmount()* δεν επιτρέπει τη χρήση του *setState()*.



Εικόνα 3.6: Διάγραμμα κύκλου ζωής των components της React.

3.3.4 JSX

Η JSX, ή JavaScript XML, είναι μια επέκταση στη σύνταξη γλώσσας JavaScript. Έχει παρόμοια εμφάνιση με το HTML. Αυτό προσφέρει στους προγραμματιστές έναν γνώριμο και εύκολο τρόπο να εμφανίσουν και να αποδώσουν components της React. Τα components της React γράφονται συνήθως σε JSX, αν και δεν είναι απαραίτητο, μπορούν επίσης να γραφτούν σε καθαρή JavaScript. Το JSX είναι παρόμοιο με μια άλλη σύνταξη επέκτασης που δημιουργήθηκε από το Facebook για PHP που ονομάζεται XHP [33].

```
function formatName(user) {
  return user.firstName + ' ' + user.lastName;
}

const user = {
  firstName: 'Harper',
  lastName: 'Perez'
};

const element = (
  <h1>
    Hello, {formatName(user)}!
  </h1>
);

ReactDOM.render(
  element,
  document.getElementById('root')
);
```

Εικόνα 3.7: Παράδειγμα κώδικα JSX.

3.4 Βάση Δεδομένων - NoSQL

Μια βάση δεδομένων NoSQL (αναφέρεται σε "μη SQL" ή "μη σχεσιακή") παρέχει έναν μηχανισμό αποθήκευσης και ανάκτησης δεδομένων που διαμορφώνεται με άλλα μέσα από τις σχέσεις πίνακα που χρησιμοποιούνται σε σχεσιακές βάσεις δεδομένων. Τέτοιες βάσεις δεδομένων υπάρχουν από τα τέλη της δεκαετίας του 1960, αλλά το όνομα "NoSQL" επινοήθηκε στις αρχές του 21ου αιώνα [34]. Οι βάσεις δεδομένων NoSQL χρησιμοποιούνται όλο και περισσότερο σε μεγάλες εφαρμογές δεδομένων και σε πραγματικού χρόνου διαδικτυακές εφαρμογές. Τα συστήματα NoSQL ονομάζονται επίσης μερικές φορές "Όχι μόνο SQL" για να τονίσουν ότι μπορούν να υποστηρίξουν γλώσσες ερωτήσεων τύπου SQL ή να παραμείνουν δίπλα σε βάσεις δεδομένων SQL.

Η NoSQL περιλαμβάνει μια μεγάλη ποικιλία διαφορετικών τεχνολογιών βάσης δεδομένων που αναπτύχθηκαν ως απάντηση στις απαιτήσεις που παρουσιάστηκαν για την κατασκευή σύγχρονων εφαρμογών. Οι προγραμματιστές αναπτύσσουν και χρησιμοποιούν εφαρμογές που δημιουργούν τεράστιους όγκους νέων, ταχέως μεταβαλλόμενων τύπων δεδομένων - δομημένα, ήμι-δομημένα, μη δομημένα και πολυμορφικά δεδομένα. Οι σχεσιακές βάσεις δεδομένων δεν είχαν σχεδιαστεί για να αντιμετωπίσουν τις προκλήσεις κλίμακας και ευελιξίας που αντιμετωπίζουν σύγχρονες εφαρμογές, ούτε κατασκευάστηκαν για να επωφεληθούν από τον διαθέσιμο αποθηκευτικό χώρο και επεξεργαστική ισχύ των εμπορευμάτων που διατίθενται σήμερα.

Έτσι, τα κίνητρα για αυτήν την προσέγγιση περιλαμβάνουν την απλότητα του σχεδιασμού, την απλούστερη «οριζόντια» κλιμάκωση σε πλήθος μηχανών και τον καλύτερο έλεγχο της διαθεσιμότητας. Οι δομές δεδομένων που χρησιμοποιούνται από βάσεις δεδομένων NoSQL είναι διαφορετικές από αυτές που χρησιμοποιούνται από προεπιλογή σε σχεσιακές βάσεις δεδομένων, κάνοντας κάποιες λειτουργίες γρηγορότερες στο NoSQL. Η καταλληλότητα μιας βάσης δεδομένων NoSQL εξαρτάται από το πρόβλημα που πρέπει να λύσει. Μερικές φορές οι δομές δεδομένων που χρησιμοποιούνται από βάσεις δεδομένων NoSQL θεωρούνται επίσης "πιο ευέλικτες" από τους σχεσιακούς πίνακες βάσεων δεδομένων [35].

Στον *Πίνακα 3.2* φαίνεται η ταξινόμηση των βάσεων δεδομένων NoSQL.

Type (τύπος της βάσης)	Notable examples of this type (παραδείγματα τέτοιων τύπων βάσεων δεδομένων)
Key-Value Cache	Apache Ignite, Couchbase, Coherence, eXtreme Scale, Hazelcast, Infinispan, Memcached, Redis, Velocity
Key-Value Store	ArangoDB, Aerospike, Couchbase, Redis
Key-Value Store (Eventually-Consistent)	Oracle NoSQL Database, Dynamo, Riak, Voldemort
Key-Value Store (Ordered)	FoundationDB, InfinityDB, LMDB, MemcacheDB
Tuple Store	Apache River, GigaSpaces
Object Database	Objectivity/DB, Perst, ZopeDB
Document Store	ArangoDB, BaseX, Clusterpoint, Couchbase, CouchDB, DocumentDB, eXist-db, IBM Domino, MarkLogic, MongoDB, Qizx, RethinkDB, Elasticsearch
Wide Column Store	Amazon DynamoDB, Bigtable, Cassandra, Scylla, HBase, Hypertable
Native Multi-model Database	ArangoDB, Cosmos DB, OrientDB, MarkLogic

Πίνακας 3.2: Τύποι και παραδείγματα βάσεων δεδομένων της NoSQL.

3.4.1 MongoDB

Το MongoDB είναι ένα πρόγραμμα βάσης δεδομένων που βασίζεται σε έγγραφα πολλαπλών πλατφορμών. Το MongoDB χαρακτηρίζεται ως πρόγραμμα βάσης δεδομένων NoSQL και χρησιμοποιεί έγγραφα τύπου JSON με ένα συγκεκριμένο σχήμα ή, καλύτερα, schema.

Η εταιρεία λογισμικού 10gen άρχισε να αναπτύσσει το MongoDB το 2007 ως συστατικό στοιχείο μιας προγραμματισμένης πλατφόρμας ως προϊόν υπηρεσιών. Το 2009, η εταιρεία στράφηκε σε ένα μοντέλο ανάπτυξης ανοιχτού κώδικα, με την εταιρεία να προσφέρει εμπορική υποστήριξη και άλλες υπηρεσίες. Το 2013, η 10gen άλλαξε το όνομά της σε MongoDB Inc [36].

Ένα από τα κύρια χαρακτηριστικά του είναι ότι το MongoDB υποστηρίζει αναζητήσεις με πεδίο, με ερωτήματα εύρους και με κανονικές εκφράσεις (field, range query and regular expressions). Τα ερωτήματα μπορούν να επιστρέψουν συγκεκριμένα πεδία εγγράφων και επίσης να περιλαμβάνουν λειτουργίες JavaScript καθορισμένες από το χρήστη. Τα ερωτήματα μπορούν επίσης να ρυθμιστούν ώστε να επιστρέφουν ένα τυχαίο δείγμα αποτελεσμάτων ενός δεδομένου μεγέθους. [37]

Το MongoDB υποστηρίζει περιορισμένες συλλογές. Αυτές είναι συλλογές σταθερού μεγέθους και διατηρούν τη σειρά εισαγωγής και, μόλις επιτευχθεί το καθορισμένο μέγεθος, συμπεριφέρονται σαν κυκλική ουρά.

Η βάση δεδομένων MongoDB παρέχει επίσης το MongoDB Atlas, μια παγκόσμια βάση δεδομένων τύπου cloud. Το MongoDB Atlas είναι η παγκόσμια βάση δεδομένων cloud για σύγχρονες εφαρμογές που διατίθεται ως πλήρως διαχειριζόμενη υπηρεσία στα AWS, Azure και Google Cloud. Παρέχει δημιουργία αντιγράφων ασφαλείας, παρακολούθηση και ειδοποιήσεις.

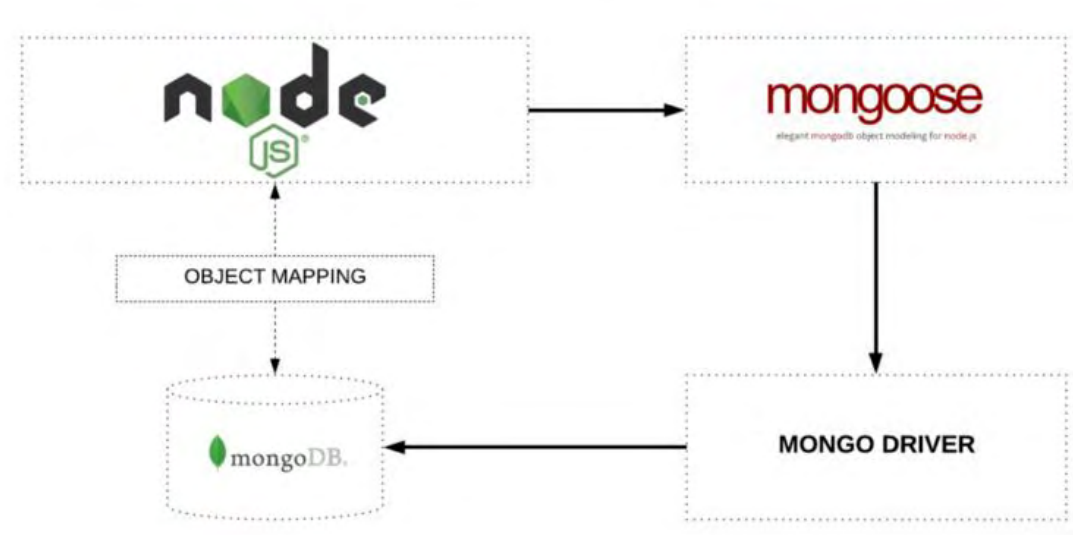
Τέλος, προσφέρει το MongoDB Compass. Με το Compass ο προγραμματιστής μπορεί να αναζητήσει, να οπτικοποιήσει και να εργαστεί με τα δεδομένα μέσω ενός GUI. Αυτό είναι ένα εξαιρετικά χρήσιμο εργαλείο για κάθε προγραμματιστή [38].

3.4.2 Mongoose

Το Mongoose είναι ένα εργαλείο μοντελοποίησης αντικειμένων MongoDB που έχει σχεδιαστεί για να λειτουργεί σε ασύγχρονο περιβάλλον. Υποστηρίζει επιστροφές τύπου promise και callback.

Το Mongoose παρέχει μια απλή, βασισμένη σε σχήμα λύση για τη μοντελοποίηση των δεδομένων της εφαρμογής (schema-based model). Περιλαμβάνει επικύρωση και δημιουργία ερωτημάτων στην βασική της έκδοση χωρίς να χρειάζεται επιπλέον εργαλεία ή βιβλιοθήκες.

Ένα μοντέλο Mongoose είναι ένα περιτύλιγμα (wrapper) στο σχήμα Mongoose, το οποίο καθορίζει τη δομή του εγγράφου, τις προεπιλεγμένες τιμές, τους επικυρωτές κ.λπ. Επίσης, ένα μοντέλο Mongoose παρέχει μια διεπαφή στη βάση δεδομένων για τη δημιουργία, την αναζήτηση, την ενημέρωση ή τη διαγραφή εγγραφών [39].



Σχήμα 3.8: Η σχέση μεταξύ των Mongoose, MongoDB και Node.Js.

3.4.3 Schema

Το σχήμα βάσης δεδομένων μιας βάσης δεδομένων είναι η δομή του που περιγράφεται σε μια επίσημη γλώσσα που υποστηρίζεται από το σύστημα διαχείρισης βάσεων δεδομένων (DBMS). Ο όρος "σχήμα" αναφέρεται στην οργάνωση των δεδομένων ως προσχέδιο του τρόπου κατασκευής της βάσης δεδομένων (χωρίζεται σε πίνακες βάσεων δεδομένων στην περίπτωση σχεσιακών βάσεων). Ο επίσημος ορισμός ενός σχήματος

βάσης δεδομένων είναι ένα σύνολο τύπων (προτάσεις) που ονομάζονται περιορισμοί ακεραιότητας που επιβάλλονται σε μια βάση δεδομένων. Αυτοί οι περιορισμοί ακεραιότητας διασφαλίζουν τη συμβατότητα μεταξύ τμημάτων του σχήματος. Όλοι οι περιορισμοί είναι εκφρασμένοι στην ίδια γλώσσα. Οι καταστάσεις ενός δημιουργημένου εννοιολογικού σχήματος μετατρέπονται σε μια σαφή χαρτογράφηση, το σχήμα βάσης δεδομένων. Αυτό περιγράφει πώς μοντελοποιούνται οντότητες στη βάση δεδομένων.

Μια βάση δεδομένων αποθηκεύει γενικά το σχήμα της σε ένα λεξικό δεδομένων. Αν και ένα σχήμα ορίζεται σε μία γλώσσα βάσης δεδομένων κειμένου, ο όρος χρησιμοποιείται συχνά για να αναφέρεται σε μια γραφική απεικόνιση της δομής της βάσης δεδομένων. Με άλλα λόγια, το σχήμα είναι η δομή της βάσης δεδομένων που καθορίζει τα αντικείμενα στη βάση αυτή [40].

3.5 Git

Το Git είναι ένα κατανεμημένο σύστημα ελέγχου εκδόσεων για την παρακολούθηση αλλαγών στον πηγαίο κώδικα κατά την ανάπτυξη λογισμικού και εφαρμογών. Έχει σχεδιαστεί για το συντονισμό της εργασίας μεταξύ προγραμματιστών, αλλά μπορεί να χρησιμοποιηθεί και για την παρακολούθηση αλλαγών σε οποιοδήποτε σύνολο αρχείων. Οι στόχοι του περιλαμβάνουν ταχύτητα, ακεραιότητα δεδομένων και υποστήριξη για κατανεμημένες, μη γραμμικές ροές εργασίας.

Το Git δημιουργήθηκε από τον Linus Torvalds το 2005 για την ανάπτυξη του πυρήνα του Linux, με άλλους προγραμματιστές του πυρήνα να συμβάλλουν στην αρχική του ανάπτυξη. Ο σημερινός συντηρητής του από το 2005 είναι ο Junio Hamano. Όπως συμβαίνει με τα περισσότερα άλλα κατανεμημένα συστήματα ελέγχου εκδόσεων και σε αντίθεση με τα περισσότερα συστήματα διακομιστή-πελάτη, κάθε κατάλογος Git σε κάθε υπολογιστή είναι ένα πλήρες αποθετήριο με πλήρες ιστορικό και πλήρεις δυνατότητες παρακολούθησης εκδόσεων, ανεξάρτητα από την πρόσβαση στο δίκτυο ή έναν κεντρικό διακομιστή. Το Git είναι δωρεάν και είναι λογισμικό ανοιχτού κώδικα που διανέμεται υπό τους όρους της έκδοσης 2 της άδειας GNU General Public License [41].

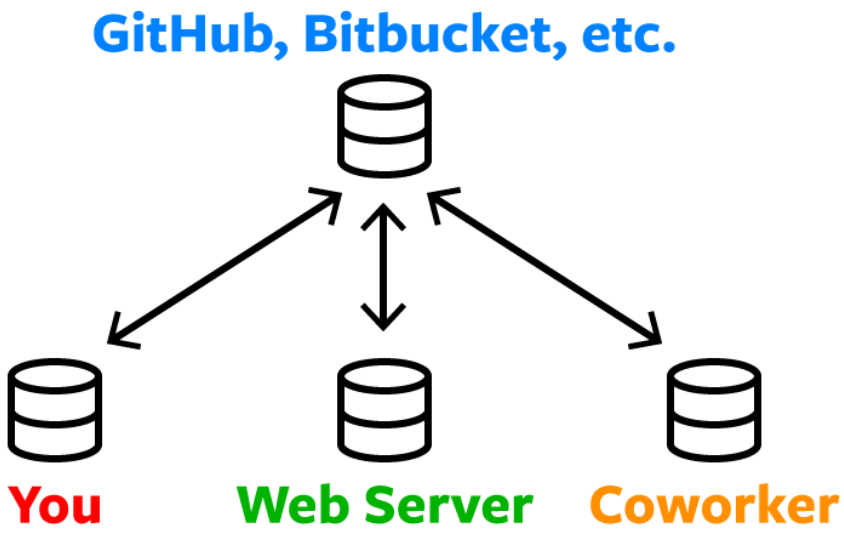
Ένα αποθετήριο Git (ή repo για συντομία) περιέχει όλα τα αρχεία ενός έργου και ολόκληρο το ιστορικό αναθεωρήσεων. Οποιοσδήποτε κοινός φάκελος αρχείων μπορεί να γίνει ένα

αποθετήριο Git. Αυτό δημιουργεί έναν υποφάκελο .git, ο οποίος περιέχει όλα τα μεταδεδομένα Git για την παρακολούθηση αλλαγών.

Πριν από το ανέβασμα των αρχείων, το Git πρέπει να γνωρίζει ποια αρχεία πρέπει να δεσμεύσει. Αυτό ονομάζεται σταδιοποίηση και χρησιμοποιεί μία εντολή προσθήκης. Η εντολή add του Git προσθέτει αρχεία σε μια περιοχή σταδιοποίησης και έπειτα ο προγραμματιστής μπορεί να δεσμεύσει τα αρχεία που έχουν σταδιαστεί για ανέβασμα.

Κάθε Git repo μπορεί να έχει πολλούς κλάδους που διαθέτουν διαφορετικές εκδόσεις ή λειτουργίες του κύριου κλάδου. Οι κλάδοι μπορούν να συγχωνευτούν στο κύριο για να ενημερώσουν την τρέχουσα έκδοση του έργου.

Η χρήση του Git είναι ζωτικής σημασίας για την ανάπτυξη οποιουδήποτε έργου, είτε το έργο έχει αναπτυχθεί από επαγγελματικές ομάδες ή μεμονωμένα άτομα.



Εικόνα 3.9: Η δομή του git.

3.6 Εργαλεία

Κατά την ανάπτυξη αυτής της εφαρμογής χρησιμοποίησα τα ακόλουθα εργαλεία:

Visual Studio Code - VSC.

Το Visual Studio Code είναι ένα πρόγραμμα επεξεργασίας πηγαίου κώδικα που αναπτύχθηκε από τη Microsoft για Windows, Linux και macOS. Περιλαμβάνει υποστήριξη για εντοπισμό σφαλμάτων, ενσωματωμένο έλεγχο Git και GitHub, επισήμανση σύνταξης, έξυπνη ολοκλήρωση κώδικα, αποσπάσματα και

αναδιαμόρφωση κώδικα. Είναι εξαιρετικά προσαρμόσιμο, επιτρέποντας στους χρήστες να αλλάξουν το θέμα, τις συντομεύσεις πληκτρολογίου, τις προτιμήσεις και να εγκαταστήσουν επεκτάσεις που προσθέτουν επιπλέον λειτουργικότητα. Ο πηγαίος κώδικας είναι δωρεάν και ανοιχτού κώδικα και κυκλοφορεί με την επιτρεπόμενη άδεια MIT [42].

Στην έρευνα προγραμματιστών που διεξήχθη στο Stack Overflow το 2019, το Visual Studio Code κατατάχθηκε ως το πιο δημοφιλές εργαλείο επεξεργασίας κώδικα, με το 50,7% των ερωτηθέντων να ισχυρίζονται ότι το χρησιμοποιούν [13].

MongoDB Compass.

Το MongoDB Compass χρησιμοποιήθηκε για την οπτική αναπαράσταση της βάσης δεδομένων αυτής της εφαρμογής.

GitHub.

Η GitHub, Inc. είναι μια εταιρεία που παρέχει έλεγχο έκδοσης ανάπτυξης λογισμικού χρησιμοποιώντας το Git. Είναι θυγατρική της Microsoft, η οποία εξαγόρασε την εταιρεία το 2018. Προσφέρει τη λειτουργικότητα του κατανεμημένου ελέγχου έκδοσης και διαχείρισης πηγαίου κώδικα (SCM) του Git, καθώς και τα δικά της χαρακτηριστικά. Παρέχει έλεγχο πρόσβασης και πολλές δυνατότητες συνεργασίας, όπως παρακολούθηση σφαλμάτων, αιτήματα λειτουργιών, διαχείριση εργασιών και wiki για κάθε έργο.

Από τον Ιανουάριο του 2019, το GitHub προσφέρει απεριόριστα ιδιωτικά αποθετήρια σε όλα τα προγράμματα, συμπεριλαμβανομένων των δωρεάν λογαριασμών. Από τον Ιανουάριο του 2020, το GitHub έχει το μεγαλύτερος πλήθος πηγαίου κώδικα στον κόσμο στα αποθετήρια του [43].

ΚΕΦΑΛΑΙΟ 4

ΛΕΙΤΟΥΡΓΙΕΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Αυτή η διπλωματική στοχεύει στην ανάπτυξη μιας διαδικτυακής εφαρμογής κοινωνικής δικτύωσης όπου ο χρήστης μπορεί να ανεβάσει μια ανάρτηση αποτελούμενη από φωτογραφίες και βίντεο. Η εφαρμογή αποτελείτε από δύο κύρια μέρη. Το πρώτο είναι το περιβάλλον διεπαφής του χρήστη και το δεύτερο είναι οι υπηρεσίες που προσφέρονται στον χρήστη.

Μόλις ένας χρήστης δημιουργήσει έναν λογαριασμό, δηλαδή αποκτήσει και προφίλ στην εφαρμογή, μπορεί να ανεβάζει αναρτήσεις και να ακολουθεί άλλους χρήστες. Η ανάρτηση αποτελείται από μία εικόνα ή ένα βίντεο και μια περιγραφή. Και τα δύο είναι σημαντικά για μια ανάρτηση αλλά μόνο η εικόνα/βίντεο είναι απαραίτητα για την δημιουργία μιας ανάρτησης . Μπορεί να υπάρξει μία ανάρτηση μόνο με εικόνα/βίντεο αλλά όχι μόνο με περιγραφή. Η σελίδα Feed εμφανίζει όλες τις αναρτήσεις που ανέβασε ο χρήστης και οι ακόλουθοί του και οποιοσδήποτε χρήστης που βλέπει αυτές τις αναρτήσεις μπορεί να κάνει like ή να τις σχολιάσει. Ο χρήστης μπορεί να ανανεώνει το status του είτε μέσω της σελίδας Feed ή μέσω της επεξεργασίας του προφίλ του. Με την γραμμή αναζήτησης, ο χρήστης μπορεί να αναζητήσει όλους τους χρήστες της εφαρμογής και να δει το προφίλ τους και σε κάθε προφίλ χρήστη υπάρχει κουμπί για να ξεκινήσεις ή να σταματήσεις την παρακολούθηση του συγκεκριμένου προφίλ. Τέλος, κάθε χρήστης μπορεί να συνομιλήσει με οποιοδήποτε ακόλουθό του μέσω της σελίδας Chat.

4.1 Διεπαφή του Χρήστη - User Interface

Το πρώτο σημαντικό μέρος της εφαρμογής είναι το περιβάλλον διεπαφής του χρήστη. Είναι οι διαθέσιμες διευθύνσεις στις οποίες μπορεί να κατευθυνθεί και έχει ο πρόσβαση ο χρήστης . Εν συντομία, υπάρχει η σελίδα σύνδεσης (Login page), η σελίδα εγγραφής (Signup page), η αρχική σελίδα των αναρτήσεων (Feed page), η σελίδα του προφίλ του χρήστη, η σελίδα επικοινωνίας με άλλους χρήστες (Chat page) και οι σελίδες για επαναφορά και αλλαγής email και κωδικού πρόσβασης. Μέρος του περιβάλλοντος διεπαφής αποτελεί και η κεφαλίδα (header) της εφαρμογής που περιέχει συνδέσμους προς τις άλλες σελίδες που είναι διαθέσιμες στον χρήστη και τη γραμμή αναζήτησης.

4.1.1 Σελίδα Σύνδεσης

Η σελίδα σύνδεσης είναι η πρώτη σελίδα που βλέπει ο χρήστης ανοίγοντας την εφαρμογή, όταν δεν είναι συνδεδεμένος. Απαραίτητη προϋπόθεση για τη χρήση της εφαρμογής είναι πρώτα να συνδεθεί. Αν ο επισκέπτης έχει ήδη λογαριασμό μπορεί να συνδεθεί στην εφαρμογή με τα στοιχεία του, διαφορετικά μπορεί να κατευθυνθεί στην σελίδα εγγραφής μέσω του κουμπιού Signup που βρίσκεται στην κεφαλίδα της σελίδας. Υπάρχει επίσης η επιλογή επαναφοράς του κωδικού πρόσβασης εάν ο χρήστης δεν έχει πλέον πρόσβαση στον υπάρχοντα κωδικό του.

Μετά τη συμπλήρωση της φόρμας σύνδεσης, τα στοιχεία ελέγχονται μέσω της βάσης δεδομένων. Το backend βρίσκει τον χρήστη με το ίδιο email από τη φόρμα σύνδεσης. Εάν ένα τέτοιο email δεν υπάρχει στη βάση δεδομένων εμφανίζεται ένα μήνυμα σφάλματος στον χρήστη. Εάν υπάρχει, τότε ελέγχει εάν ο κωδικός πρόσβασης είναι σωστός με τη βιβλιοθήκη bcrypt, ένα εργαλείο κρυπτογράφησης και αποκρυπτογράφησης κωδικών πρόσβασης, και εάν ο κωδικός πρόσβασης δεν ταιριάζει, ένα μήνυμα σφάλματος ενημερώνει τον χρήστη. Εάν ο χρήστης υπάρχει, συνδέεται στην εφαρμογή και ανακατευθύνεται στη αρχική σελίδα αναρτήσεων, το Feed page.

Επιπλέον, δημιουργείται ένα «διακριτικό» (token) μέσω της βιβλιοθήκης και του εργαλείου jwt. Αυτό το «διακριτικό» είναι μοναδικό για κάθε χρήστη που είναι συνδεδεμένος. Το backend στέλνει αυτό το «διακριτικό» στο frontend ως μέρος της απόκρισης στο αίτημα σύνδεσης. Το frontend γνωρίζει τώρα το «διακριτικό» αυτού του συγκεκριμένου χρήστη και μπορεί να ενεργήσει ανάλογα με ορισμένους κανόνες που παρέχονται από τον προγραμματιστή. Εάν το διακριτικό έχει λήξει, ο χρήστης αποσυνδέεται αυτόματα και ανακατευθύνεται πίσω στη σελίδα σύνδεσης. Στην εφαρμογή αυτή, ο χρήστης μπορεί να παραμείνει συνδεδεμένος για μία ώρα προτού αποσυνδεθεί αυτόματα και ανακατευθυνθεί στην σελίδα σύνδεσης (Login page).

Εικόνα 4.1: Η σελίδα σύνδεσης.

4.1.2 Σελίδα Εγγραφής

Στη σελίδα εγγραφής (Signup page) ο χρήστης δημιουργεί έναν λογαριασμό στην εφαρμογή, δηλαδή αποκτά και ένα προφίλ. Για τη δημιουργία του λογαριασμού απαιτείται μόνο μια διεύθυνση ηλεκτρονικού ταχυδρομείου (email), ένα ψευδώνυμο ή όνομα χρήστη και ένας κωδικός.

Ο λογαριασμός email πρέπει να είναι ένα έγκυρο email που θα πρέπει να είναι μοναδικό και κάθε προφίλ αντιστοιχεί σε ένα μόνο email. Προσπαθώντας να εγγραφείτε με το ίδιο email και διαφορετικό όνομα χρήστη, θα δημιουργηθεί και εμφανιστεί σφάλμα που θα ενημερώνει τον χρήστη ότι αυτό το email υπάρχει ήδη.

Το όνομα χρήστη δεν χρειάζεται να είναι μοναδικό και δεν έχει περιορισμούς χαρακτήρων ή αριθμών χαρακτήρων.

Ο κωδικός πρόσβασης πρέπει να έχει τουλάχιστον 5 χαρακτήρες, αλλά δεν υπάρχει περιορισμός στον τύπο των χαρακτήρων.

Μετά την ολοκλήρωση της διαδικασίας εγγραφής, ο χρήστης ανακατευθύνεται στη σελίδα σύνδεσης. Με την υποβολή της φόρμας, τα στοιχεία του χρήστη, συμπεριλαμβανομένου του email, του ονόματος χρήστη και του κωδικού πρόσβασης, αποθηκεύονται στη βάση δεδομένων. Αυτό σημαίνει ότι δημιουργείται ένα προφίλ για αυτό το email. Το προφίλ του

χρήστη αρχικοποιείται με την κατάσταση "I am new!" και με μια προεπιλεγμένη εικόνα προφίλ.

Για λόγους ασφαλείας, ο κωδικός πρόσβασης έχει κατακερματιστεί / κρυπτογραφηθεί πριν αποθηκευτεί στη βάση δεδομένων χρησιμοποιώντας τη βιβλιοθήκη bcrypt. Αφού δημιουργηθεί το μοντέλο χρήστη στο backend και πριν το μοντέλο αποθηκευτεί στη βάση δεδομένων, αποστέλλεται ένα email στη διεύθυνση email του χρήστη από το "welcome@insta.com" με θέμα "Welcome!" και κάποιο περιεχόμενο που καλωσορίζει τον χρήστη στην εφαρμογή. Προτού το backend ανταποκριθεί στο αίτημα του frontend, το εργαλείο της αναζήτησης χρηστών του frontend ενημερώνεται μέσω του socket.io για να συμπεριληφθεί ο νέος χρήστης στη λίστα αναζήτησης.

4.1.3 Σελίδα Αναρτήσεων – Αρχική Σελίδα

Η σελίδα αναρτήσεων, Feed page, είναι η αρχική και κεντρική σελίδα της εφαρμογής. Σχεδόν όλες οι ενέργειες και υπηρεσίες που προσφέρονται στον χρήστη υπάρχουν σε αυτήν την σελίδα.

Εδώ εμφανίζονται όλες οι αναρτήσεις που ανέβασε ο χρήστης και οι ακόλουθοί του με φθίνουσα σειρά ως προς την ημερομηνία ανάρτησης, που σημαίνει ότι οι πιο πρόσφατες αναρτήσεις βρίσκονται στην κορυφή.

Από πάνω προς τα κάτω της σελίδας αναρτήσεων μπορούμε να δούμε την περιοχή ενημέρωσης κατάστασης (status update), ένα κουμπί δημιουργίας νέας ανάρτησης και έπειτα όλες τις διαθέσιμες αναρτήσεις.

Η περιοχή του κειμένου κατάστασης δείχνει την τρέχουσα κατάσταση του χρήστη. Κάνοντας κλικ στην περιοχή εισαγωγής κειμένου (text area input), ο χρήστης μπορεί να αλλάξει την κατάστασή του. Κάνοντας κλικ στο κουμπί ενημέρωσης, αποστέλλεται ένα αίτημα στον διακομιστή backend για ενημέρωση της κατάστασης του χρήστη.

Κάνοντας κλικ στο κουμπί νέας ανάρτησης, εμφανίζεται ένα αναδυόμενο παράθυρο (modal). Αυτό το παράθυρο αποτελείται από ένα πεδίο μεταφόρτωσης αρχείων, μια περιοχή κειμένου περιγραφής και δύο κουμπιά, Αποδοχή και Ακύρωση. Το πεδίο επιλογής αρχείων χειρίζεται τα μέσα που ο χρήστης επιθυμεί να ανεβάσει. Η περιοχή κειμένου είναι

η περιγραφή ή το περιεχόμενο που ο χρήστης επιθυμεί να συνοδεύσει την εικόνα ή το βίντεό του. Τα δύο κουμπιά κάνουν ακριβώς όπως υποδεικνύεται από τα ονόματά τους. Το κουμπί ακύρωσης ακυρώνει τη διαδικασία και κλείνει το αναδυόμενο παράθυρο ενώ το κουμπί αποδοχής στέλνει ένα αίτημα στο backend για τη δημιουργία μιας νέας ανάρτησης και κλείνει και αυτό με τη σειρά του το αναδυόμενο παράθυρο. Περισσότερες λεπτομέρειες σχετικά με την δημιουργία των αναρτήσεων εξηγούνται αργότερα στην διπλωματική.

Η ανάρτηση αποτελείται από μια **κεφαλίδα**, μια **ενότητα μέσων** και περιγραφής, μια **ενότητα σχολίων** και ένα **υποσέλιδο**.

Η **κεφαλίδα** διατηρεί πληροφορίες σχετικά με τον χρήστη που ανέβασε την ανάρτηση. Από αριστερά προς τα δεξιά μπορούμε να δούμε την εικόνα και το όνομα χρήστη, το οποίο λειτουργεί ως σύνδεσμος για το προφίλ του χρήστη, και στην άκρη δεξιά υπάρχει ένα κουμπί που ανοίγει ένα αναδυόμενο παράθυρο με τις διαθέσιμες ενέργειες για κάποιον χρήστη. Εάν η ανάρτηση ανήκει στον χρήστη που είναι συνδεδεμένος αυτήν τη στιγμή, οι επιλογές είναι είτε "προβολή" των λεπτομερειών της ανάρτησης, "επεξεργασία" της ανάρτησης, "διαγραφή" της ανάρτησης ή κλείσιμο του αναδυόμενου παραθύρου. Εάν η ανάρτηση δεν ανήκει στον χρήστη, οι μόνες διαθέσιμες επιλογές είναι η "προβολή" των λεπτομερειών της ανάρτησης ή να κλείσει το αναδυόμενο παράθυρο.

Στην **ενότητα πολυμέσων** έχουμε την εικόνα ή το βίντεο που ανέβασε ο χρήστης και ακριβώς κάτω από αυτό το μέσο, μπορούμε να δούμε το περιεχόμενο ή την περιγραφή της ανάρτησης, ακριβώς δίπλα στο όνομα χρήστη που δημιούργησε την ανάρτηση. Ακριβώς κάτω από την περιγραφή έχουμε ένα κουμπί με αντίχειρα emoji και δίπλα του ο συνολικός αριθμός επισημάνσεων "μου αρέσει" και των σχολίων αυτής της ανάρτησης. Κάνοντας κλικ τον αντίχειρα emoji, ο χρήστης μπορεί να δηλώσει "μου αρέσει" (like) στην ανάρτηση ή να πάρει πίσω το like που δήλωσε προηγουμένως.

Κάτω από την ενότητα πολυμέσων μπορούμε να βρούμε την **ενότητα σχολίων**. Εδώ οι χρήστες μπορούν να σχολιάσουν την ανάρτηση. Στις αναρτήσεις που εμφανίζονται στη σελίδα Feed, ο χρήστης μπορεί να δει μόνο τα δύο πιο δημοφιλή σχόλια της ανάρτησης, εάν η ανάρτηση έχει σχόλια, να δηλώσει αν του αρέσει ή δεν του αρέσει κάποιο από αυτά τα σχόλια και, τέλος, να επεξεργάζεται ή διαγράφει ένα σχόλιο εάν μπορεί να εγκριθεί από το σύστημα, δηλαδή μόνο εάν είναι δικό του αυτό το σχόλιο. Ο χρήστης μπορεί επίσης να

σχολιάσει είτε μέσω μιας περιοχής κειμένου είτε ενός κουμπιού σχολίου που εμφανίζει ένα αναδυόμενο παράθυρο (modal) με μια περιοχή κειμένου και δύο κουμπιά, για την ακύρωση αυτής της ενέργειας ή την αποδοχή της.

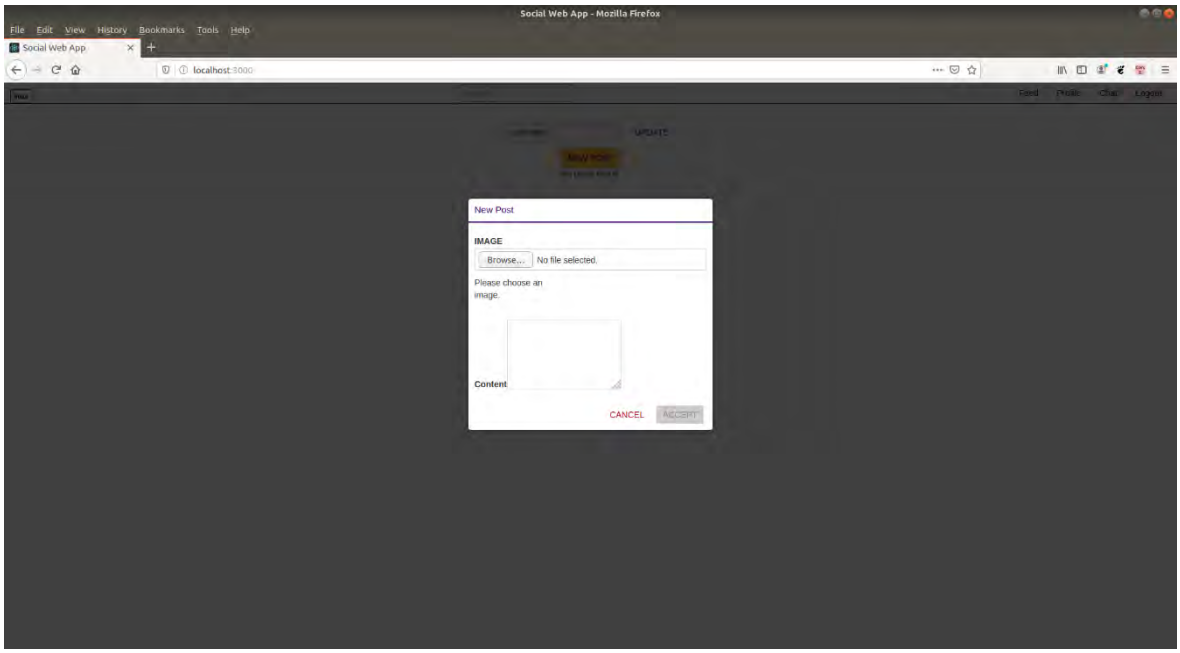
Τέλος, στο **υποσέλιδο**, υπάρχουν πληροφορίες σχετικά με την ημερομηνία μεταφόρτωσης αυτής της ανάρτησης. Η ημερομηνία εμφανίζεται σε χρόνο που έχει περάσει από τη στιγμή δημιουργίας της ανάρτησης.

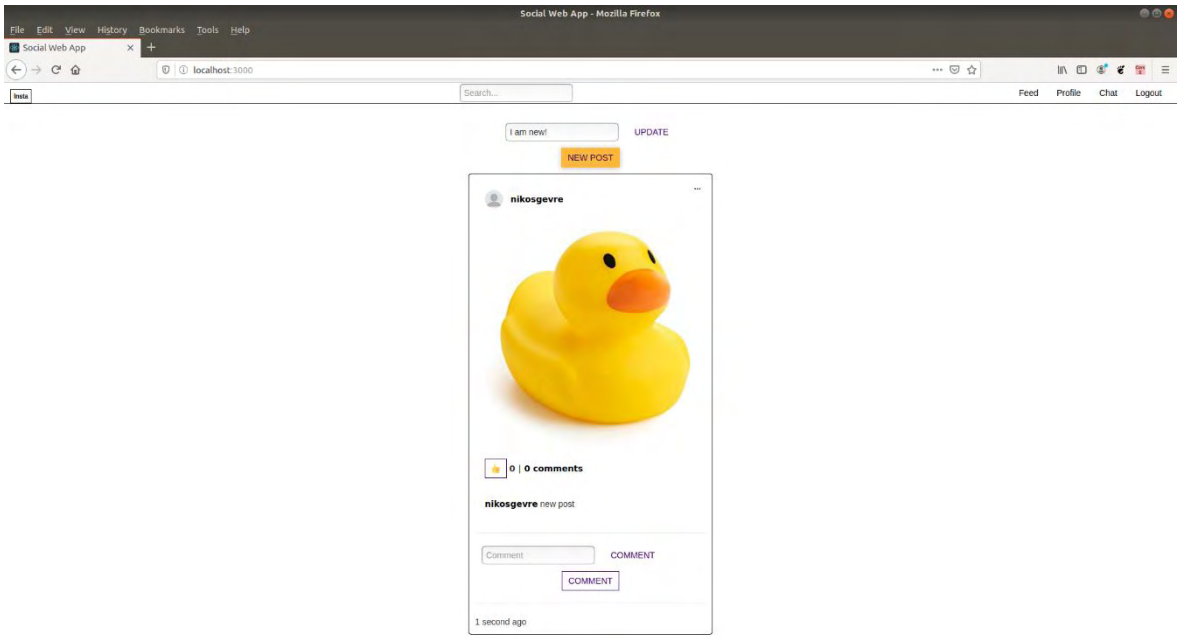
Μόλις ο χρήστης κατευθυνθεί σε αυτή τη σελίδα, συμβαίνουν τα ακόλουθα. Το frontend στέλνει ένα αίτημα στο backend για να απαντήσει με όλες τις αναρτήσεις που μπορεί να δει ο χρήστης. Εάν ο χρήστης δεν έχει ανεβάσει καμία ανάρτηση και δεν ακολουθεί κανένα χρήστη που έχει ανεβάσει αναρτήσεις στο παρελθόν, ο χρήστης βλέπει μόνο μια περιγραφή που λέει, "Δεν βρέθηκαν αναρτήσεις." αντί των δημοσιεύσεων. Όσο το backend χειρίζεται το αίτημα, το frontend εμφανίζει ένα spinner (μία γραμμή που γυρνάει περιστροφικά) στην οθόνη δηλώνοντας ότι περιμένει κάποια απάντηση από το backend.

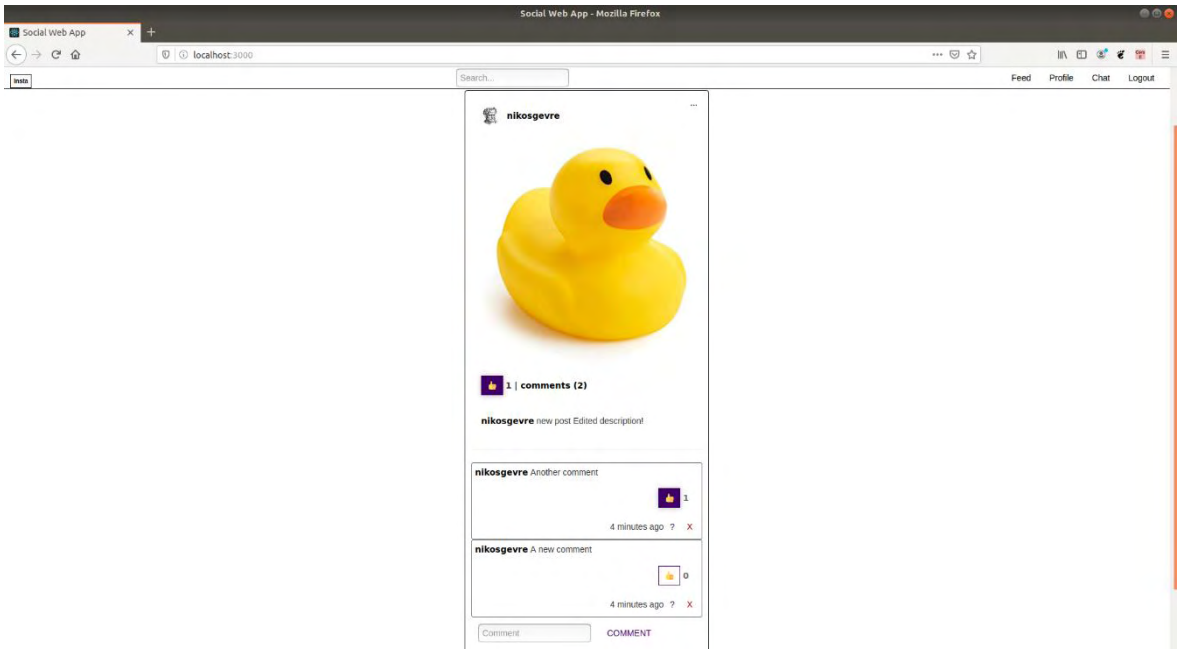
Στο στοιχείο της react αυτής της σελίδα (Feed page component), υπάρχουν χειριστές για τη δημιουργία αναρτήσεων, την επεξεργασία τους και τη διαγραφή τους, εάν ο χρήστης είναι εξουσιοδοτημένος να το κάνει, για τη δήλωση "μου αρέσει" και την διαχείριση σχολίων μιας ανάρτησης και του ανοίγματος του αναδυόμενου παραθύρου με τις διαθέσιμες ενέργειες του χρήστη. Οι χειριστές ενημέρωσης κατάστασης αποτελούν επίσης μέρος αυτού του component. Όλοι οι χειριστές σχετικά με τις δημοσιεύσεις και την κατάσταση χρησιμοποιούν socket.io για άμεση ενημέρωση.

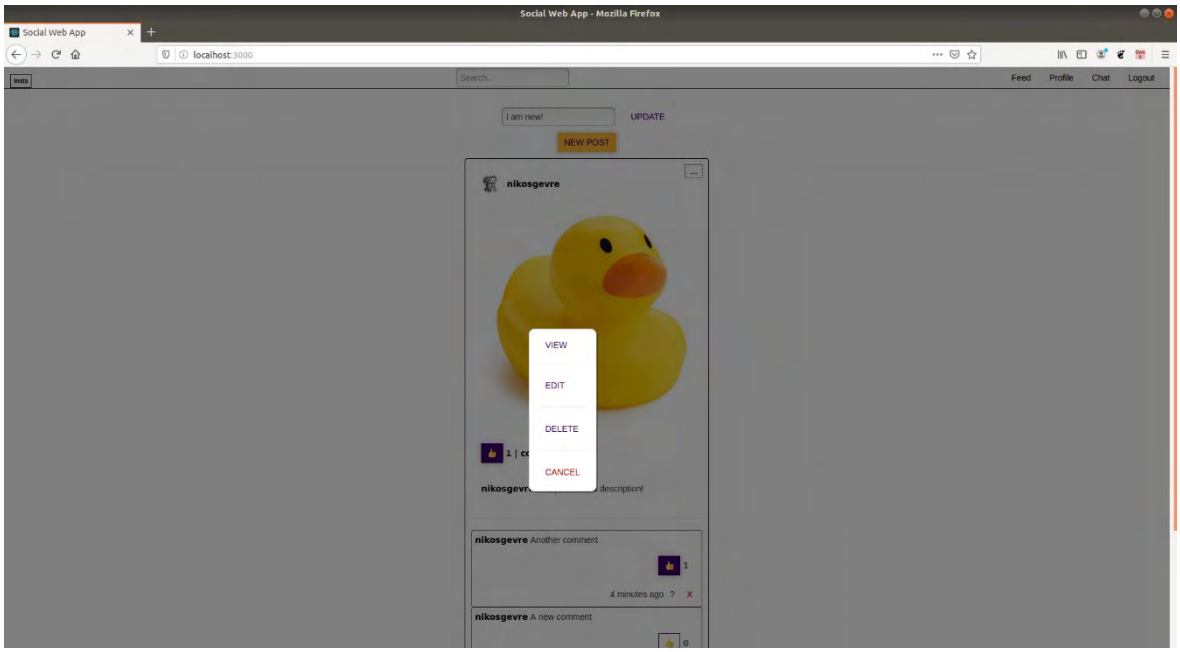
Ανάλογα με τον αριθμό των αναρτήσεων που είναι διαθέσιμες στον χρήστη, η σελίδα αυτή μπορεί να είναι πολύ μικρή ή πολύ μεγάλη. Ο χρήστης μπορεί να πραγματοποιήσει κύλιση προς τα κάτω έως ότου φτάσει στην τελευταία ανάρτηση.


Social Web App - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Social Web App
localhost:3000
Insta
Search
Feed Profile Chat Logout
I am new!
UPDATE
NEW POST
No posts found.


Social Web App - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Social Web App
localhost:3000
New Post
IMAGE
Browse...
No file selected.
Please choose an image.
Content
CANCEL
ACCEPT


Social Web App - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Social Web App
localhost:3000
Search...
Feed Profile Chat Logout
I am new!
UPDATE
NEW POST
nikosgevre
0 | 0 comments
nikosgevre new post
Comment
COMMENT
COMMENT
1 second ago


Social Web App - Mozilla Firefox
Social Web App
localhost:3000
Search...
Feed Profile Chat Logout
nikosgevre
1 | comments (2)
nikosgevre new post Edited description!
nikosgevre Another comment
1
4 minutes ago ? X
nikosgevre A new comment
0
4 minutes ago ? X
Comment
COMMENT



Εικόνα 4.2: Από πάνω προς τα κάτω: 4.2.1: Η αρχική σελίδα χωρίς αναρτήσεις, 4.2.2: Το παράθυρο δημιουργίας νέας ανάρτησης, 4.2.3: Η αρχική σελίδα με αναρτήσεις, 4.2.4: Η ανάρτηση με σχόλια και 4.2.5: Το παράθυρο ενεργειών μιας ανάρτησης.

4.1.4 Σελίδα Λεπτομερειών Ανάρτησης

Αυτή είναι η σελίδα μιας ανάρτησης, όπου ο καθένας μπορεί να δει σε βάθος τις λεπτομέρειες για την ανάρτηση που έχει επιλεχθεί.

Από πάνω προς τα κάτω μπορούμε να δούμε μια **κεφαλίδα**, την **ενότητα πολυμέσων**, **την ενότητα σχολίων** και ένα **υποσέλιδο**.

Στην **κεφαλίδα**, για άλλη μια φορά, έχουμε πληροφορίες σχετικά με τον χρήστη που ανέβασε αυτήν την ανάρτηση. Το όνομα χρήστη είναι ένας σύνδεσμος προς το προφίλ αυτού του χρήστη και η ημερομηνία εμφανίζεται ως η ακριβής ώρα δημιουργίας της ανάρτησης.

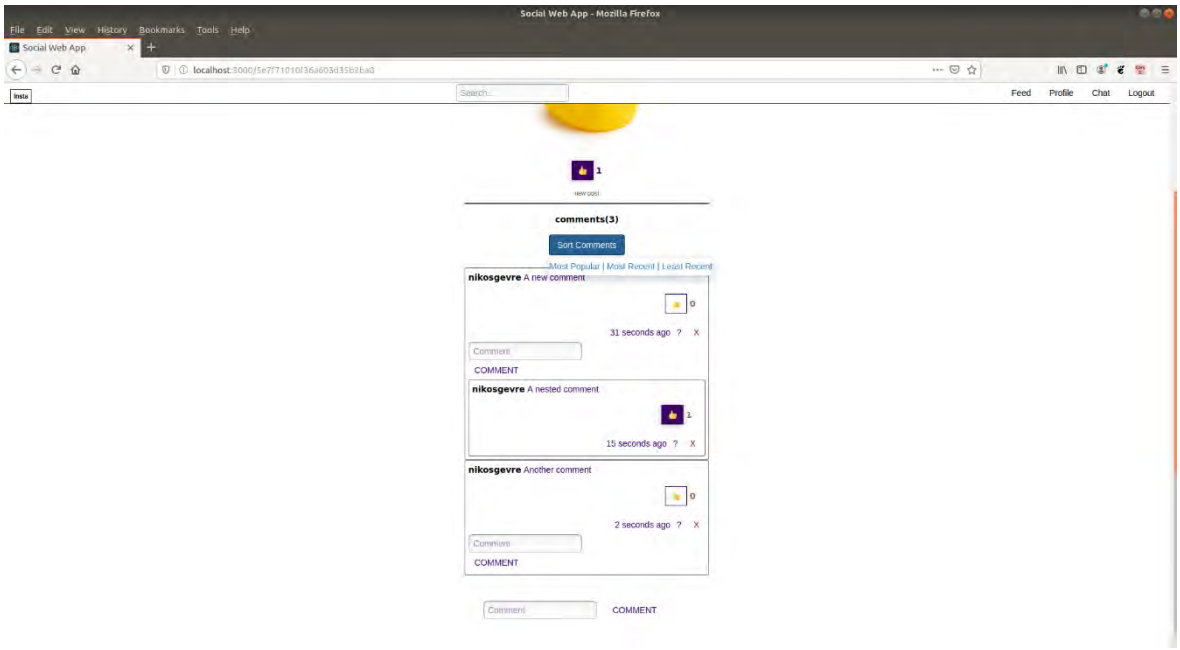
Μεταβαίνοντας στην **ενότητα πολυμέσων** μπορούμε να δούμε την εικόνα ή το βίντεο που μεταφορτώθηκε. Ακριβώς από κάτω, έχουμε τους ίδιους αντίχειρες emoji για την δήλωση like/unlike στην ανάρτηση. Τέλος, έχουμε και την περιγραφή της ανάρτησης.

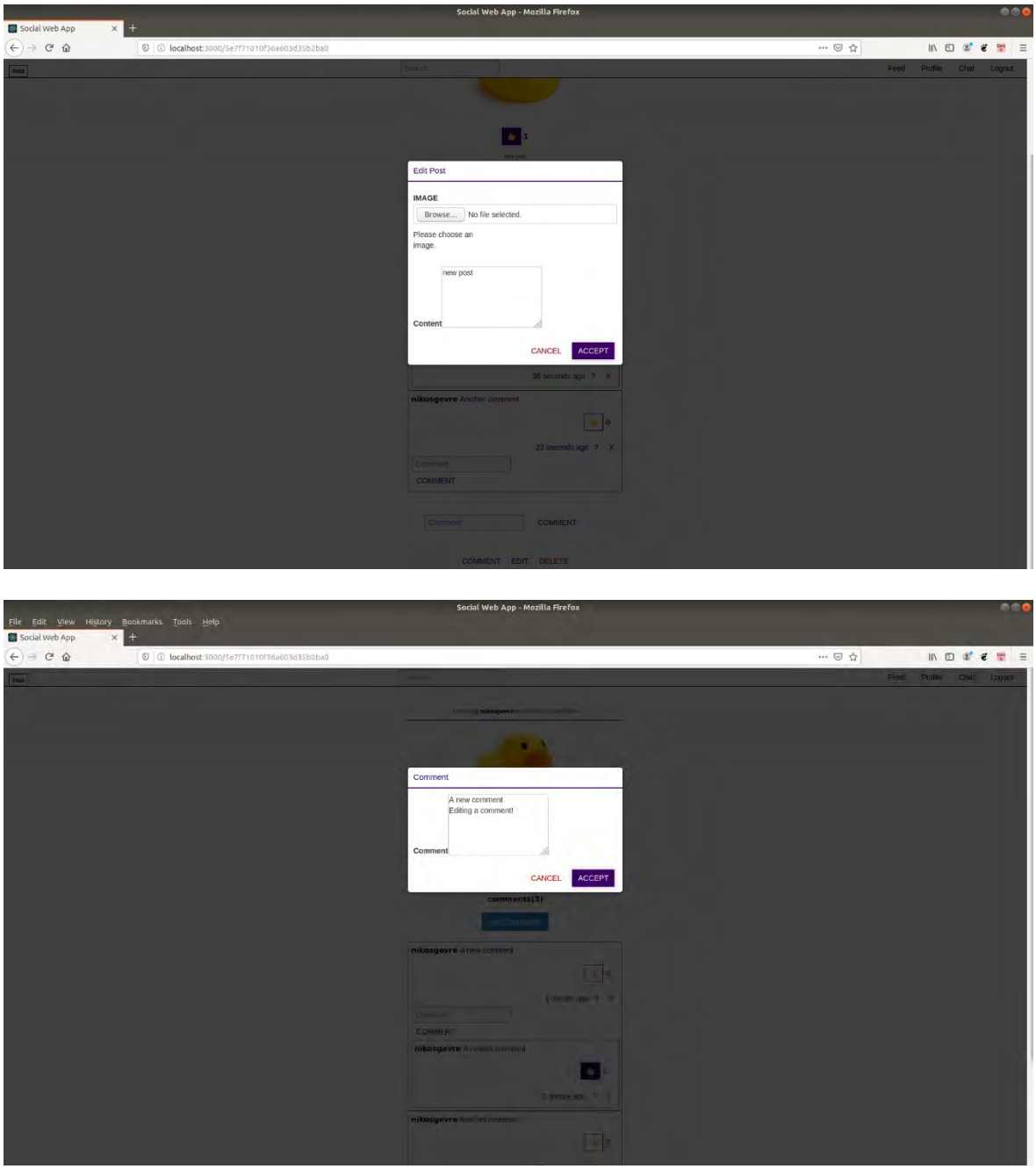
Στην **ενότητα σχολίων** μπορούμε να δούμε τα σχόλια αυτής της ανάρτησης σε αύξουσα σειρά από τη στιγμή που δημιουργήθηκαν, που σημαίνει ότι το παλαιότερο σχόλιο είναι στην κορυφή και το νεότερο στο κάτω μέρος. Σε κάθε σχόλιο μπορεί κάποιος να δηλώσει

αν του αρέσει ή όχι. Επίσης, ο χρήστης μπορεί να σχολιάσει ένα σχόλιο και να του αρέσει ή όχι ένα ένθετο σχόλιο, αλλά αυτή η λειτουργικότητα θα αναλυθεί περαιτέρω στην υπηρεσία σχολίων αργότερα. Κάθε σχόλιο έχει μια περιοχή κειμένου για σχολιασμό μέσα σε αυτό το σχόλιο και ακριβώς κάτω από την ενότητα σχολίων έχουμε την περιοχή κειμένου για τη δημιουργία σχολίων στην αρχική ανάρτηση. Τα σχόλια συνδέονται με το backend με το socket.io.

Στην περιοχή του **υποσέλιδου**, υπάρχουν ορισμένα κουμπιά ανάλογα με το αν ο χρήστης είναι εξουσιοδοτημένος ή όχι (αν είναι δικιά του η ανάρτηση η όχι) και ο χρήστης μπορεί να σχολιάσει την αρχική ανάρτηση μέσω ενός αναδυόμενου παραθύρου modal, να επεξεργαστεί ή να διαγράψει την ανάρτηση.

Στο στοιχείο react της σελίδας μιας ανάρτησης, υπάρχουν χειριστές υπεύθυνοι για τα like, τα σχόλια και επεξεργασία ή διαγραφή μιας ανάρτησης ή σχολίου. Κάθε χειριστής χρησιμοποιεί το socket.io για άμεση ενημέρωση.





Εικόνα 4.3: Από πάνω προς τα κάτω: 4.3.1: Η σελίδα μιας ανάρτησης, 4.3.2: Ο τρόπος επεξεργασίας ανάρτησης μέσο του modal και 4.3.3: Το modal για επεξεργασία σχολίων.

4.1.5 Σελίδα Προφίλ

Αυτή είναι η σελίδα που δείχνει το προφίλ κάθε χρήστη. Μπορεί να διαφέρει ανάλογα με το αν αυτό είναι το προφίλ του χρήστη που είναι συνδεδεμένος αυτήν τη στιγμή ή το προφίλ κάποιου άλλου χρήστη.

Κάποιες λειτουργίες αυτής της σελίδας είναι ακριβώς ίδιες και ορισμένες ενότητες και κουμπιά εξαρτώνται από το αν βρισκόμαστε στο προφίλ του χρήστη που είναι συνδεδεμένος αυτήν τη στιγμή είτε κάποιου άλλου χρήστη. Ας εξετάσουμε λοιπόν αυτήν τη σελίδα σε δύο περιπτώσεις.

Το προφίλ του χρήστη που είναι συνδεδεμένος.

Ως συνήθως, ξεκινώντας από πάνω προς τα κάτω, μπορούμε να δούμε την εικόνα προφίλ του χρήστη, ένα κουμπί για την επεξεργασία του προφίλ, το όνομα χρήστη, το email, την ημερομηνία που ο χρήστης δημιούργησε λογαριασμό στην εφαρμογή, ορισμένα στατιστικά στοιχεία σχετικά με τις αναρτήσεις και τους ακολούθους, την κατάσταση του χρήστη και τις αναρτήσεις του χρήστη.

Πατώντας το κουμπί επεξεργασίας προφίλ, εμφανίζεται ένα αναδυόμενο παράθυρο (modal) στην οθόνη. Από εκεί, ο χρήστης μπορεί να αλλάξει την εικόνα του προφίλ του από έναν επιλογέα αρχείων, να αλλάξει το όνομα χρήστη και την κατάστασή του. Επιπλέον, υπάρχει ένα κουμπί για την επαναφορά του email. Όπως έχει ήδη αναφερθεί, δεν υπάρχουν περιορισμοί στο όνομα χρήστη ή την κατάσταση, εφόσον έχουν τουλάχιστον έναν χαρακτήρα. Υπάρχουν επίσης δύο κουμπιά, ένα για ακύρωση της ενέργειας και ένα για αποδοχή και αποστολή αιτήματος στο backend.

Τα στατιστικά περιλαμβάνουν τον αριθμό των άλλων χρηστών που ο χρήστης ακολουθά, τον αριθμό των χρηστών που ακολουθούν τον χρήστη και τον αριθμό των δημοσιεύσεων που δημιούργησε αυτός ο χρήστης.

Στην ενότητα αναρτήσεων, το στοιχείο ανάρτησης είναι λίγο διαφορετικό από αυτό της αρχικής σελίδας. Η διαφορά είναι το γεγονός ότι οι αναρτήσεις ταξινομούνται σε αύξουσα σειρά από τη στιγμή που δημιουργήθηκαν. Εδώ η κεφαλίδα δεν περιλαμβάνει πληροφορίες χρήστη και περιλαμβάνει μόνο το κουμπί ενεργειών που δίνονται στον χρήστη. Οι ενέργειες περιορίζονται μόνο στην προβολή λεπτομερειών ή διαγραφή της ανάρτησης. Ο χρήστης δεν μπορεί να σχολιάσει τις αναρτήσεις αλλά μόνο να δηλώσει «μου αρέσει». Για άλλη μια φορά, ο χρήστης μπορεί να μετακινηθεί προς τα κάτω στη σελίδα επ 'αόριστον μέχρι να φτάσει στην τελευταία ανάρτηση.

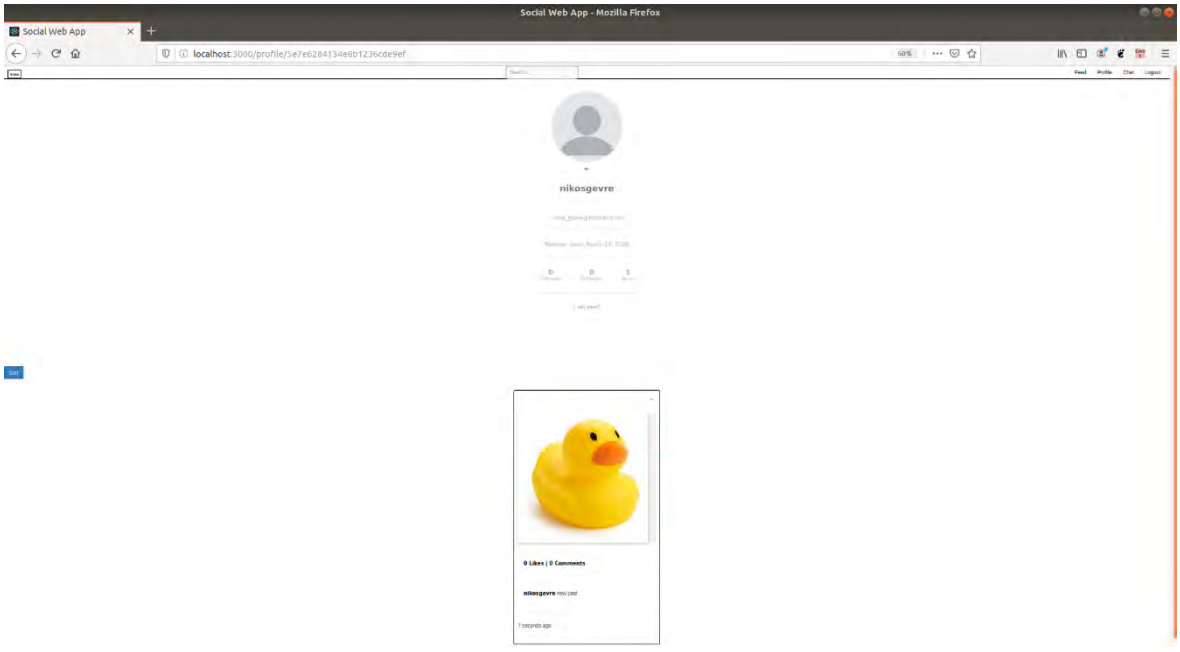
Το προφίλ άλλων χρηστών.

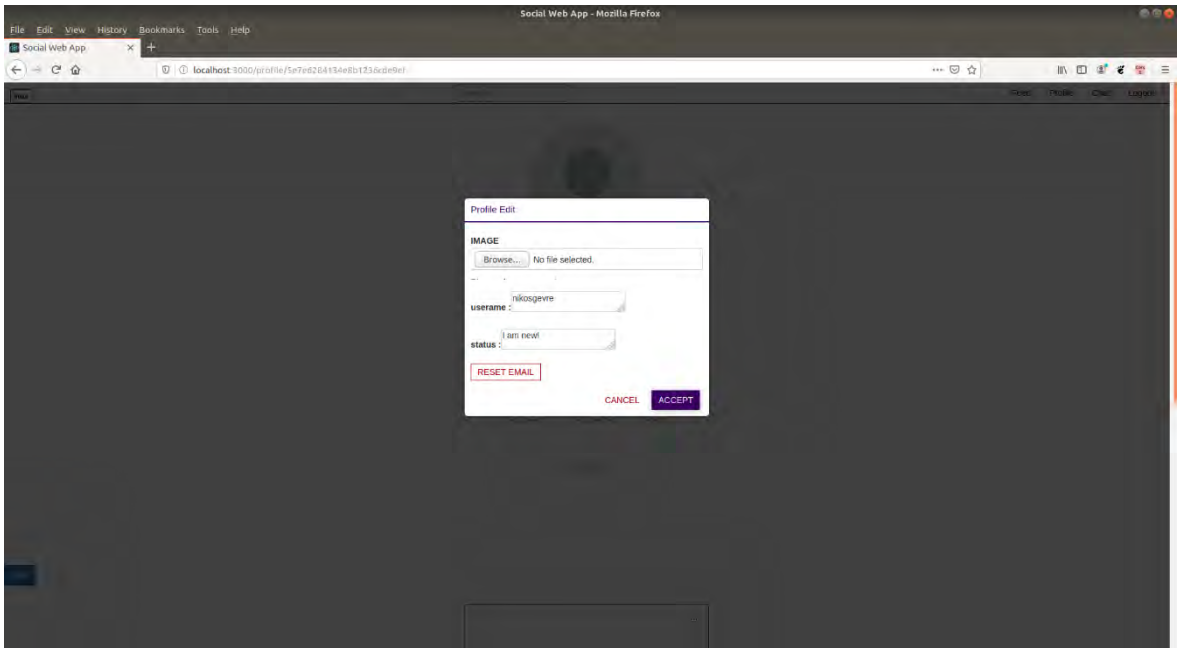
Ξεκινώντας από πάνω προς τα κάτω, μπορούμε να δούμε την εικόνα προφίλ του χρήστη, το όνομα χρήστη, το email, την ημερομηνία που ο χρήστης δημιούργησε λογαριασμό στην εφαρμογή, ορισμένα στατιστικά στοιχεία σχετικά με τις αναρτήσεις και τους ακολούθους, την κατάστασή του ένα κουμπί από το οποίο μπορεί να τον ακολουθήσει ο συνδεδεμένος χρήστης ή να σταματήσει να τον ακολουθάει χρήστη και τις αναρτήσεις του χρήστη.

Πατώντας το κουμπί follow / unfollow ο χειριστής της react ενεργεί ανάλογα.

Η ενότητα αναρτήσεων έχει και πάλι κάποιους περιορισμούς. Ο χρήστης που είναι συνδεδεμένος αυτήν τη στιγμή μπορεί να δει μόνο την ανάρτηση πατώντας το κουμπί προβολής από το παράθυρο ενεργειών. Δεδομένου ότι ο χρήστης που είναι συνδεδεμένος δεν είναι εξουσιοδοτημένος, δεν μπορεί να διαγράψει καμία ανάρτηση. Για άλλη μια φορά, ο χρήστης μπορεί να μετακινηθεί προς τα κάτω στη σελίδα επ 'αόριστον μέχρι να φτάσει στην τελευταία ανάρτηση.

Τέλος, η σελίδα προφίλ διαθέτει τους χειριστές για την επεξεργασία του προφίλ, παρακολούθηση ή κατάργηση παρακολούθησης κάποιου χρήστη και προβολή ή διαγραφή μιας ανάρτησης. Όλοι οι χειριστές χρησιμοποιούν το socket.io για άμεση ενημέρωση.





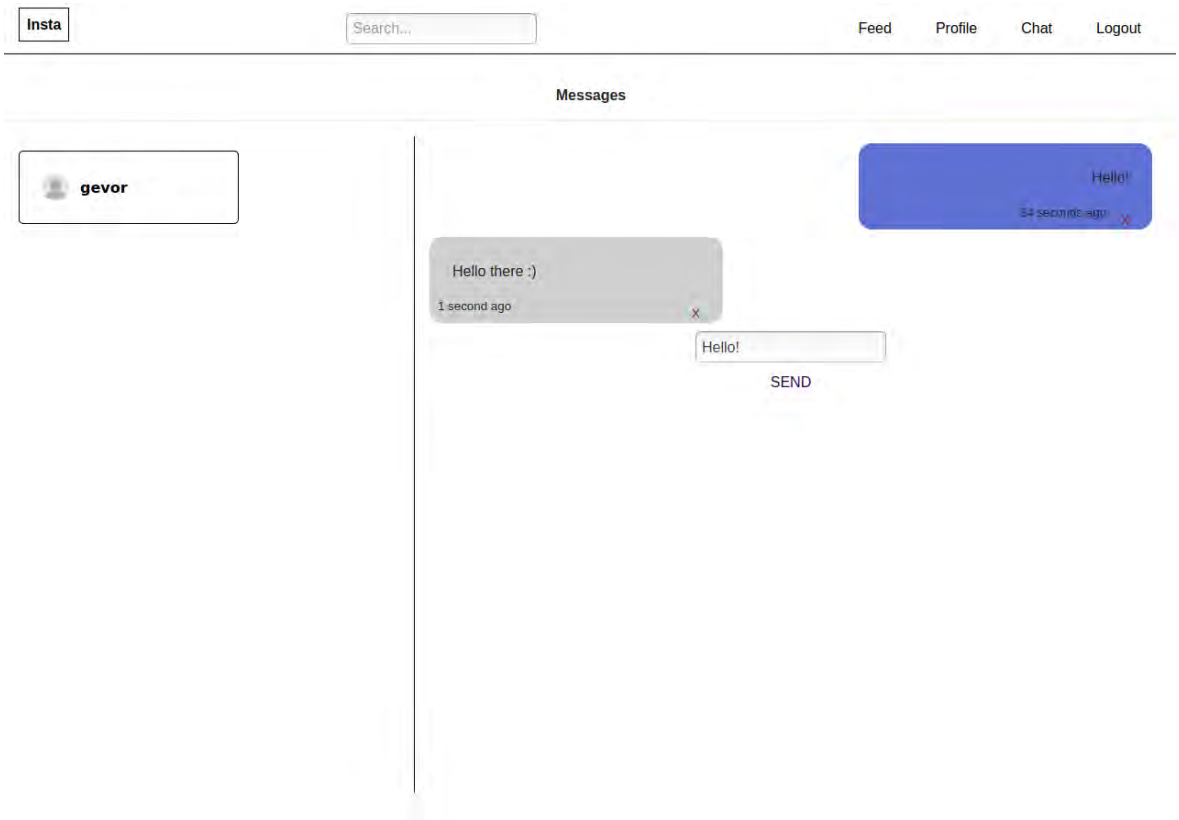
Εικόνα 4.4: Από πάνω προς τα κάτω: 4.4.1: Το προφίλ του συνδεδεμένου χρήστη και 4.4.2: Το αναδυόμενο παράθυρο επεξεργασίας του προφίλ

4.1.6 Σελίδα Μυνημάτων

Η σελίδα συνομιλίας είναι αρκετά απλή και αποτελείται από δύο στοιχεία. Στην αριστερή πλευρά ο χρήστης μπορεί να δει όλους τους χρήστες που ακολουθεί. Κάνοντας κλικ σε μια κεφαλή συνομιλίας (chat head), εμφανίζεται στη δεξιά πλευρά το περιβάλλον των μηνυμάτων (chat interface). Από εκεί ο χρήστης μπορεί να δει τα μηνύματα που έχει ανταλλάξει με αυτόν τον συγκεκριμένο χρήστη και να στείλει νέα μηνύματα. Κάνοντας κλικ σε ένα άλλο chat head, το chat interface στα δεξιά αλλάζει ανάλογα.

Κάθε στοιχείο μηνύματος αποτελείται από το μήνυμα, τον χρόνο που πέρασε από τη στιγμή της αποστολής και ένα σύμβολο «Χ». Κάνοντας κλικ στο «Χ», ο χρήστης μπορεί να διαγράψει ένα μήνυμα. Προφανώς, ένας χρήστης μπορεί να διαγράψει μόνο τα δικά του μηνύματα. Τα μηνύματα εμφανίζονται σε δύο διαφορετικά χρώματα και σε δύο διαφορετικά μέρη. Τα μηνύματα του συνδεδεμένου χρήστη εμφανίζονται στη δεξιά πλευρά με μπλε χρώμα και τα μηνύματα από τον χρήστη με τον οποίο συνομιλεί εμφανίζονται στην αριστερή πλευρά με γκρι χρώμα.

Η υπηρεσία της συνομιλίας έχει αναπτυχθεί χρησιμοποιώντας websockets για ανταλλαγή άμεσων μηνυμάτων. Κάθε χρήστης μπορεί να δει το μήνυμα αμέσως μετά την αποστολή του.



Εικόνα 4.5: Η σελίδα συνομιλίας.

4.1.7 Σελίδες Επαναφοράς Κωδικού Και Email

Αυτές οι σελίδες είναι υπεύθυνες για την επαναφορά του κωδικού πρόσβασης ή του email ενός χρήστη. Είναι πολύ παρόμοιες μεταξύ τους. Η σελίδα επαναφοράς κωδικού πρόσβασης είναι προσβάσιμη μόνο εάν ο χρήστης αποσυνδεθεί, ενώ η σελίδα επαναφοράς email είναι προσβάσιμη μόνο όταν ο χρήστης είναι συνδεδεμένος και βρίσκεται στο αναδυόμενο παράθυρο επεξεργασίας προφίλ.

Η διαδικασία επαναφοράς ακολουθεί ορισμένα μέτρα ασφαλείας. Σε ορισμένες περιπτώσεις, η διαδικασία είναι παρόμοια και για τους δύο ελεγκτές επαναφοράς. Ας τα δούμε όμως αναλυτικά.

Επαναφορά κωδικού πρόσβασης

Αυτή η σελίδα περιλαμβάνει μόνο μια φόρμα με περιοχή κειμένου που ο χρήστης πρέπει να δώσει το email του. Μόλις ο χρήστης υποβάλει τη φόρμα που περιλαμβάνει ένα έγκυρο email, ο διακομιστής backend εφαρμόζει ένα διακριτικό

επαναφοράς (reset token) στον χρήστη και αποθηκεύει αυτό το διακριτικό και μια ημερομηνία λήξης στη βάση δεδομένων. Προτού στείλει την απάντηση πίσω στο frontend, το backend στέλνει επίσης ένα email στη διεύθυνση που παρέχεται από τη φόρμα με έναν σύνδεσμο που ο χρήστης πρέπει να ακολουθήσει το οποίο ανακατευθύνει τον χρήστη στην επόμενη σελίδα επαναφοράς του κωδικού πρόσβασης.

Στη δεύτερη σελίδα επαναφοράς κωδικού πρόσβασης υπάρχει και πάλι μια απλή φόρμα, μόνο με μία περιοχή εισαγωγής του νέου κωδικού πρόσβασης. Αυτή η σελίδα ονομάζεται νέα σελίδα στοιχείων, καθώς ο χρήστης ορίζει τα νέα στοιχεία του και τα αποθηκεύει στη βάση δεδομένων.

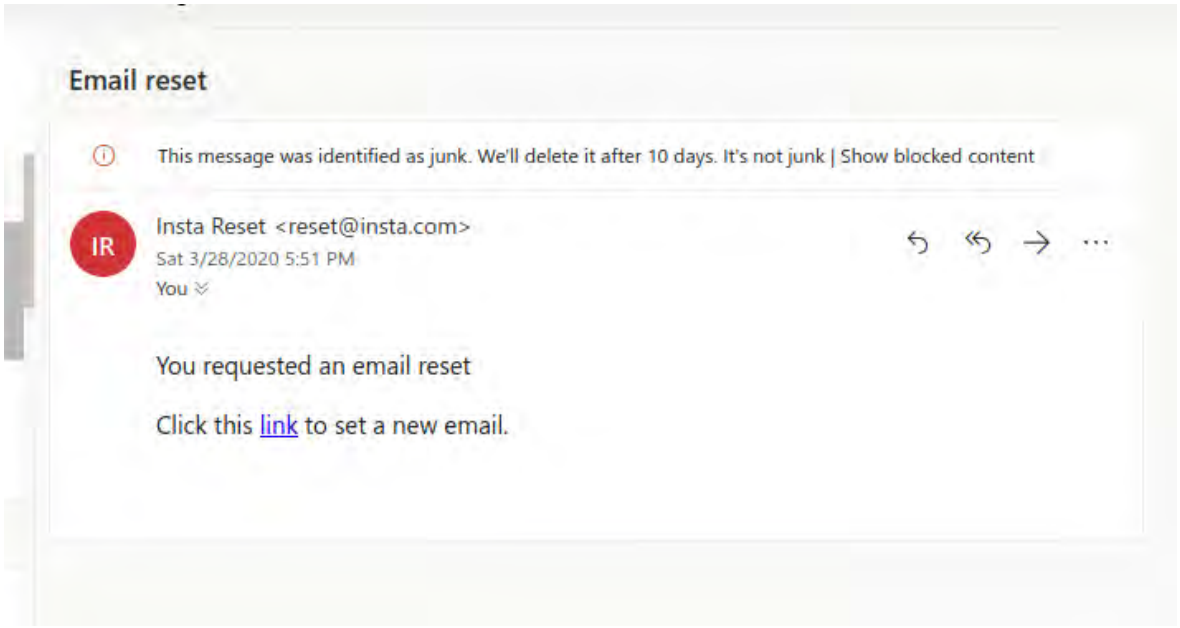
Στη συνέχεια, ο χρήστης ανακατευθύνεται στη σελίδα σύνδεσης όπου μπορεί να συνδεθεί στην εφαρμογή με τον καινούργιο κωδικό που όρισε προηγουμένως.

Επαναφορά διεύθυνσης ηλεκτρονικού ταχυδρομείου (email)

Αυτή η σελίδα περιλαμβάνει μόνο μια φόρμα με περιοχή κειμένου για το νέο email. Σε αυτήν την περιοχή κειμένου email, ο χρήστης πρέπει να παρέχει ένα έγκυρο email στο οποίο έχει πρόσβαση. Το backend ενεργεί όπως πριν. Ένα διακριτικό (reset token) και μια ημερομηνία λήξης για αυτό το διακριτικό αποθηκεύονται στη βάση δεδομένων του χρήστη και ένα email αποστέλλεται στη διεύθυνση που παρέχεται από τον χρήστη μέσω της φόρμας. Σε αυτήν την περίπτωση και σε αντίθεση με την προηγούμενη λειτουργικότητα, ο χρήστης δεν καθορίζεται από το email που παρείχε, αλλά από το userId του, καθώς το email που παρείχε ο χρήστης πιθανώς δεν είναι αυτό που είναι αποθηκευμένο στη βάση δεδομένων του και το userId είναι διαθέσιμο στο backend από τη στιγμή που ο χρήστης είναι συνδεδεμένος. Αυτή η διαδικασία διασφαλίζει ότι ο χρήστης έχει πρόσβαση στο νέο του email. Ακολουθώντας το σύνδεσμο από το email, ο χρήστης ανακατευθύνεται στη δεύτερη σελίδα επαναφοράς του email του.

Στη νέα σελίδα στοιχείων, παρόμοια με τη δεύτερη σελίδα επαναφοράς κωδικού πρόσβασης, ο χρήστης πρέπει να παράσχει ένα έγκυρο email που επιθυμεί να χρησιμοποιεί στο εξής. Το backend ενημερώνει τη βάση δεδομένων με τα καινούργια στοιχεία του χρήστη.

Στη συνέχεια, ο χρήστης ανακατευθύνεται στη αρχική σελίδα (‘/’), η οποία είναι πιθανώς η σελίδα αναρτήσεων, καθώς ο χρήστης είναι ήδη συνδεδεμένος.



Εικόνα 4.5: Από πάνω προς τα κάτω: 4.5.1: Η σελίδα επαναφοράς email, 4.5.2: Το μήνυμα για επαναφορά της ηλεκτρονικής διεύθυνσης και 4.5.2: Η σελίδα νέων στοιχείων του χρήστη.

4.2 Υπηρεσίες της Εφαρμογής

Το δεύτερο μέρος της εργασίας είναι οι Υπηρεσίες που προσφέρονται στον χρήστη. Εν ολίγοις, ο χρήστης μπορεί να δημιουργήσει αναρτήσεις, να επεξεργαστεί αυτές τις αναρτήσεις, να αρέσει ή / και να σχολιάσει μια ανάρτηση και να αρέσει ή / και να σχολιάσει ένα υπάρχον σχόλιο, να επεξεργαστεί το προφίλ του, να αλλάξει την κατάστασή του, να αναζητήσει άλλους χρήστες και να ακολουθήσει αυτούς τους χρήστες.

4.2.1 Δημιουργία Ανάρτησης

Ο χρήστης μπορεί να δημιουργήσει και να ανεβάσει αναρτήσεις.

Η ανάρτηση πρέπει να αποτελείται από ένα πολυμέσο, είτε μια εικόνα είτε ένα βίντεο, και μια περιγραφή. Η περιγραφή δεν πρέπει να είναι κενή ή, με άλλα λόγια, πρέπει να είναι μεγαλύτερη από 1 χαρακτήρα.

Το backend είναι πολύ συγκεκριμένο σε αυτήν την υπηρεσία. Παίρνει τη φόρμα από το αίτημα του frontend και δημιουργεί μια καταχώριση στη βάση δεδομένων των δημοσιεύσεων.

4.2.2 Επεξεργασία Ανάρτησης

Ο χρήστης μπορεί να επεξεργαστεί κάθε ανάρτηση που έχει ανεβάσει. Η επεξεργασία γίνεται προσβάσιμη από τη σελίδα αναρτήσεων μέσω του μενού επιλογών της ανάρτησης ή από τη σελίδα μίας ανάρτησης μέσω του κουμπιού επεξεργασίας.

Ο χρήστης μπορεί να αλλάξει την φωτογραφία ή βίντεο της ανάρτησης καθώς και το περιεχόμενό της. Και πάλι, το περιεχόμενο πρέπει να έχει τουλάχιστον 1 χαρακτήρα.

4.2.3 Like Και Σχόλια Ανάρτησης

Ο χρήστης μπορεί να δηλώσει ότι «του αρέσει» ή όχι οποιαδήποτε δημοσίευση και να σχολιάσει οποιαδήποτε ανάρτηση και κάποια σχόλια. Υπάρχει περιορισμός για να σχολιάσετε μόνο ένα σχόλιο πρώτου επιπέδου. Ο χρήστης δεν μπορεί να σχολιάσει ένα ένθετο σχόλιο.

Προτιμώντας μια ανάρτηση, ο χρήστης προστίθεται στον πίνακα των επισημάνσεων «μου αρέσει» σε μια ανάρτηση και η ανάρτηση προστίθεται στη λίστα των αναρτήσεων του χρήστη που «του αρέσουν». Με αυτόν τον τρόπο, τόσο ο χρήστης όσο και η ανάρτηση γνωρίζουν ότι ο συγκεκριμένος χρήστης δήλωσε ότι «του αρέσει» αυτή η ανάρτηση.

Τα σχόλια είναι μια συλλογή στη βάση δεδομένων, σε αντίθεση με τα likes. Με αυτόν τον τρόπο έχουμε μια αναφορά που είτε δείχνει μια ανάρτηση, επισημαίνοντας αυτό το σχόλιο ως σχόλιο πρώτου επιπέδου, ή δείχνει ένα σχόλιο, επισημαίνοντάς το ως σχόλιο δεύτερου επιπέδου ή ένθετο σχόλιο. Ο χρήστης είναι ελεύθερος να αρέσει και να επεξεργάζεται σχόλια, αλλά περιορίζεται στο να σχολιάζει μόνο ένα σχόλιο πρώτου επιπέδου.

4.2.4 Επεξεργασία Προφίλ

Ο χρήστης μπορεί να επεξεργαστεί το προφίλ του. Όπως ήδη αναφέρθηκε, ο χρήστης μπορεί να αλλάξει το όνομα χρήστη, την κατάσταση και την εικόνα του προφίλ του. Δεν υπάρχουν περιορισμοί εκτός από το γεγονός ότι το όνομα χρήστη και η κατάσταση πρέπει να έχουν τουλάχιστον 1 χαρακτήρα.

Ο χρήστης, επίσης, έχει τη δυνατότητα να αλλάξει τον κωδικό πρόσβασης και το email του μέσω της διαδικασίας επαναφοράς που εξηγείται νωρίτερα στη διπλωματική.

4.2.5 Κατάσταση Χρήστη

Ο χρήστης μπορεί να ορίσει την κατάσταση του. Από προεπιλογή, όταν δημιουργείται ένα προφίλ, η κατάσταση του χρήστη αρχικοποιείται με την τιμή "I am new!". Ο χρήστης μπορεί να αλλάξει αυτήν την τιμή είτε από τη σελίδα αναρτήσεων μέσω της περιοχής κατάστασης είτε από το προφίλ του μέσω της υπηρεσίας της επεξεργασίας του προφίλ.

4.2.6 Αναζήτηση Χρηστών

Ο χρήστης μπορεί να αναζητήσει άλλους χρήστες μέσω της γραμμής αναζήτησης στην κύρια γραμμή πλοήγησης της εφαρμογής. Η γραμμή αναζήτησης είναι πάντα διαθέσιμη στον χρήστη, εφόσον έχει γίνει έλεγχος ταυτότητας, που σημαίνει ότι είναι συνδεδεμένος. Όταν δημιουργείται ένας νέος χρήστης, η γραμμή αναζήτησης ενημερώνεται αυτόματα με τον νέο χρήστη.

Ωστόσο, η λειτουργία της αναζήτησης είναι αμφιλεγόμενη. Υπάρχουν δύο διαφορετικοί τρόποι προσέγγισης μιας αναζήτησης στη βάση δεδομένων. Το πρώτο είναι να εκτελεστεί η αναζήτηση στο backend και να σταλθούν τα τελικά αποτελέσματα φιλτραρισμένα στο frontend. Ο δεύτερος τρόπος είναι να σταλθούν όλα τα αποτελέσματα στο frontend και να φιλτραριστούν τα δεδομένα εκεί. Και οι δύο τρόποι έχουν κάποια οφέλη και μειονεκτήματα.

Εκτελώντας την αναζήτηση και το φιλτράρισμα στο backend, όλο το φορτίο φιλτραρίσματος των δεδομένων πέφτει στον διακομιστή/σέρβερ. Εάν ένας μεγάλος αριθμός χρηστών εκτελέσει την αναζήτηση ταυτόχρονα, ο διακομιστής θα υπερφορτώσει και ενδεχομένως να έχει αντίκτυπο σε όλους όσους χρησιμοποιούν την εφαρμογή ταυτόχρονα, αφήνοντάς όλους τους χρήστες δυσαρεστημένους με τα αποτελέσματα.

Εκτελώντας την αναζήτηση και το φιλτράρισμα στο frontend, κάθε χρήστης ίσως να υπερφορτώσει το πρόγραμμα περιήγησής του, αλλά κάθε άλλος χρήστης είναι ασφαλής. Ακόμα κι αν ένας μεγάλος αριθμός ατόμων ζητήσει αναζήτηση την ίδια στιγμή, ο διακομιστής backend δεν θα υπερφορτωθεί. Οι χρήστες που ζήτησαν μια αναζήτηση πιθανώς θα υπερφορτώσουν το frontend σέρβερ τους ξεχωριστά, πράγμα που σημαίνει ότι εάν το αρχείο δεδομένων είναι πολύ μεγάλο, μόνο οι χρήστες που ζήτησαν την αναζήτηση θα έχουν κακή εμπειρία.

Σε μικρότερες εφαρμογές όπου οι χρήστες δεν είναι τόσοι πολλοί, προτιμάται το φιλτράρισμα των δεδομένων στο backend. Σε μεγαλύτερες εφαρμογές όπου τα αρχεία δεδομένων είναι πολύ μεγάλα για ταυτόχρονο χειρισμό, είναι καλύτερο να χειριστεί το φιλτράρισμα των δεδομένων το frontend. Τα τελευταία χρόνια, δεδομένου ότι οι προσωπικοί υπολογιστές έχουν γίνει καλύτεροι και πιο ισχυροί, είναι πιο συνηθισμένο να φιλτράρονται τα δεδομένα στον frontend σέρβερ.

Στην εφαρμογή μου, για τους λόγους που έχουν ήδη αναφερθεί, έχω φιλτράρει τα δεδομένα στο frontend. Αυτό, επίσης, βοηθά στην καλύτερη οπτικοποίηση των αποτελεσμάτων.

4.2.7 Παρακολούθηση / Follow Χρηστών

Ο χρήστης μπορεί να ακολουθήσει ή να σταματήσει να ακολουθά άλλους χρήστες οποιαδήποτε στιγμή. Με τη δημιουργία λογαριασμού, ο χρήστης δεν έχει ούτε ακόλουθους ούτε ακολουθεί κανέναν. Ο χρήστης μπορεί να βλέπει μόνο τις δικές του αναρτήσεις, έως ότου αρχίσει αρχίσει να ακολουθεί άλλους χρήστες.

Επιλέγοντας κάποιον από τα αποτελέσματα αναζήτησης, ο χρήστης ανακατευθύνεται στο προφίλ του χρήστη που επέλεξε από την λίστα αποτελεσμάτων και από εκεί μπορεί να ακολουθήσει αυτόν τον χρήστη.

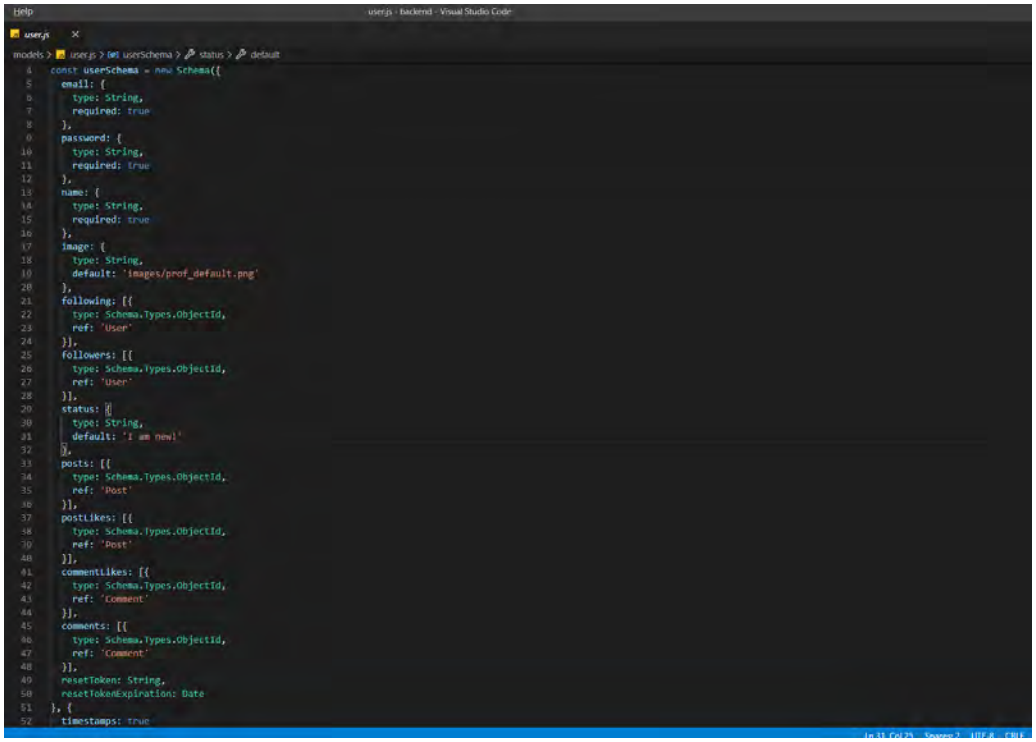
Μετά το πάτημα του κουμπιού Follow, οι αναρτήσεις του χρήστη αυτού εμφανίζονται στη σελίδα αναρτήσεων. Ο συνδεδεμένος χρήστης προσθέτει τον άλλο χρήστη στον πίνακα «following» (χρήστες που ακολουθάει) και ο άλλος χρήστης προσθέτει τον συνδεδεμένο χρήστη στους «followers» του (χρήστες που τον ακολουθούν).

4.3 Μοντέλα Βάσης Δεδομένων

Για αυτήν την εφαρμογή δημιουργήθηκε μια βάση δεδομένων MongoDB με τρεις συλλογές (collection). Κάθε συλλογή έχει ένα διαφορετικό μοντέλο που δημιουργείται και διαχειρίζεται μέσω του mongoose.

Στην *Εικόνα 4.6*, έχουμε το μοντέλο του χρήστη. Πρώτον, έχουμε πεδία email, κωδικού πρόσβασης και ονομάτων που απαιτούνται για να υπάρχει ο χρήστης. Αυτά είναι τα μόνα πεδία στα οποία έχει πρόσβαση ο χρήστης μέσω της διαδικασίας εγγραφής. Αργότερα ο χρήστης μπορεί να αλλάξει όλα τα άλλα πεδία.

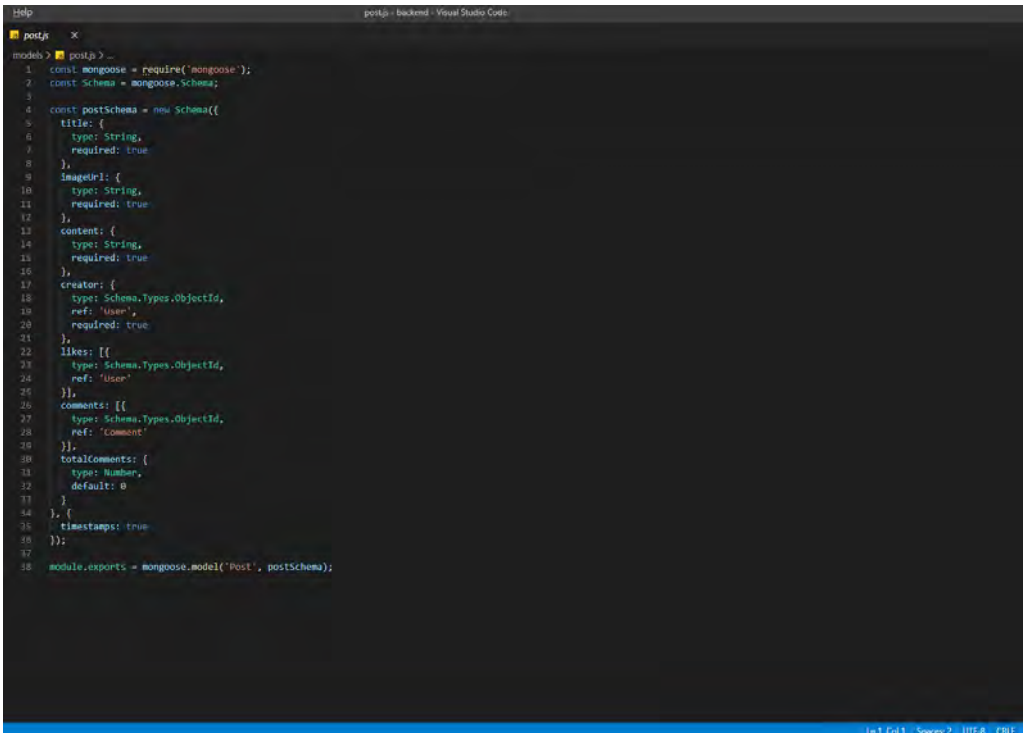
Η εικόνα (image) προφίλ αρχικοποιείται με την προεπιλεγμένη εικόνα προφίλ και η κατάσταση (status) αρχικοποιείται με "I am new!". Τα following, followers, posts, postLikes, commentLikes και σχόλια αναφέρονται (reference) σε άλλες συλλογές (collections) και ενημερώνονται ενώ ο χρήστης χρησιμοποιεί την εφαρμογή. Η χρονική σήμανση (timestamp) σηματοδοτεί την ακριβή ώρα δημιουργίας του χρήστη.



Εικόνα 4.6: Το μοντέλο χρήστη.

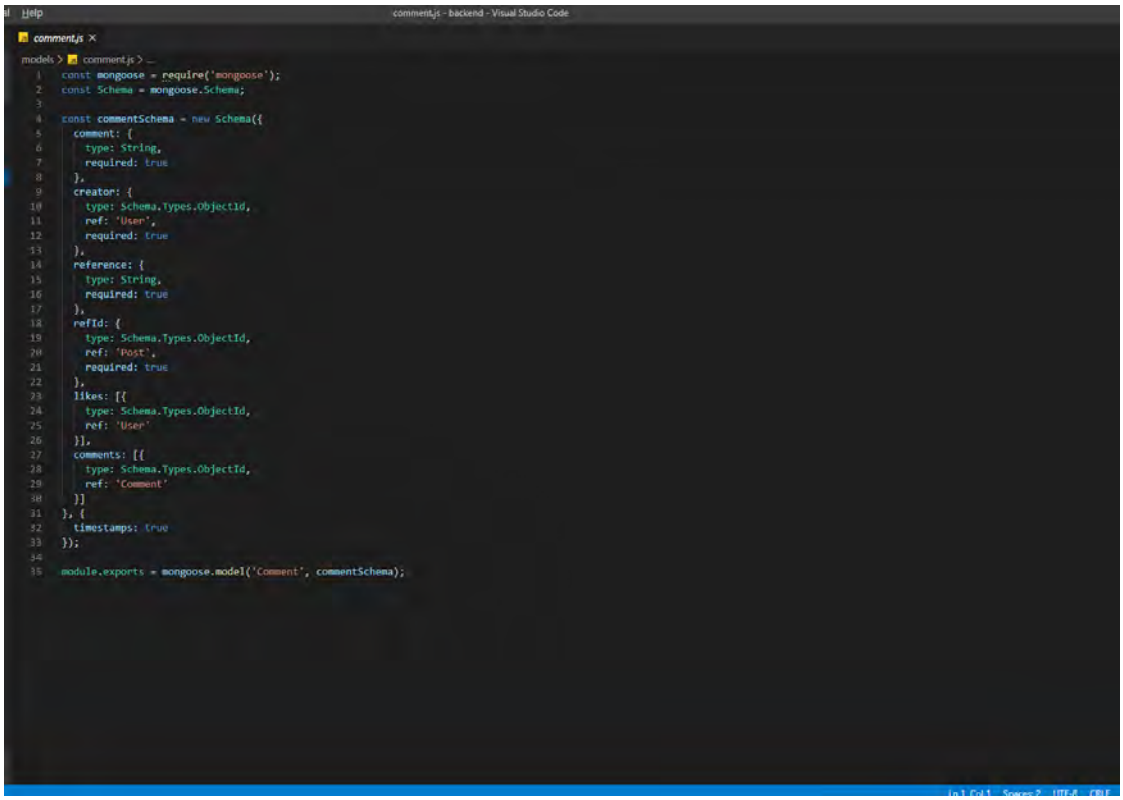
Στην *Εικόνα 4.7*, έχουμε το μοντέλο αναρτήσεων. Αρχικά, το imageUrl κρατά τη διαδρομή του πολυμέσου που ανέβασε ο χρήστης, το content διατηρεί την περιγραφή, το creator κρατά την αναφορά στο μοντέλο χρήστη του χρήστη που δημιούργησε την ανάρτηση και όλα αυτά απαιτούνται για την ύπαρξη μιας καταχώρησης ανάρτησης.

Οι επισημάνσεις "μου αρέσει" (likes) και τα σχόλια (comments) περιέχουν αναφορές (references) σε χρήστες και σχόλια αντίστοιχα και το πεδίο totalComments, επίσης, διατηρεί τον αριθμό των συνολικών σχολίων σε αυτήν την ανάρτηση. Η χρονική σήμανση (timestamps) σηματοδοτεί τον ακριβή χρόνο δημιουργίας της ανάρτησης.



Εικόνα 4.7: Το μοντέλο ανάρτησης.

Στην *Εικόνα 4.8*, μπορούμε να δούμε το μοντέλο σχολίων. Απαιτείται το περιεχόμενο του σχολίου, ο δημιουργός, ο οποίος διατηρεί μια αναφορά στον χρήστη, το reference και το refId, τα οποία διατηρούν τον τύπο αναφοράς και το αναγνωριστικό της αναφοράς αντίστοιχα. Οι επισημάνσεις "μου αρέσει" (likes) και τα σχόλια (comments) είναι πίνακες με αναφορές στους χρήστες που τους άρεσαν το σχόλιο και τα ένθετα σχόλια εάν υπάρχουν. Η χρονική σήμανση (timestamp) σηματοδοτεί την ακριβή ώρα δημιουργίας του σχολίου.



Εικόνα 4.8: Το μοντέλο σχολίων.

ΚΕΦΑΛΑΙΟ 5

ΣΥΜΠΕΡΑΣΜΑΤΑ - ΒΙΒΛΙΟΓΡΑΦΙΑ

5.1 Συμπεράσματα

Αυτή η εργασία επικεντρώνεται στις πιο πρόσφατες και πιο βελτιωμένες τεχνολογίες στον προγραμματισμό εφαρμογών ιστού και συγκεκριμένα σε αυτές που βασίζονται σε JavaScript. Ο στόχος ήταν η εκμάθηση σε βάθος τη JavaScript και ειδικότερα των τεχνολογιών Node.js και React.js και ο εντοπισμός των πλεονεκτημάτων και τα μειονεκτημάτων τους.

Συμπληρώνοντας τόσο το θεωρητικό όσο και το πρακτικό μέρος αυτής της διπλωματικής μπορούμε να συμπεράνουμε ότι οι νέες τεχνολογίες δίνουν στον προγραμματιστή τεράστιες δυνατότητες και εργαλεία ανάπτυξης εφαρμογών που μπορούν να ικανοποιήσουν όλες τις απαιτήσεις και να διευκολύνουν τη διαδικασία. Η χρήση της JavaScript τόσο στο frontend όσο και στο backend μιας εφαρμογής ιστού, συμβάλλει στην εύκολη σύνδεση των frontend και backend και διευκολύνει την ανάπτυξη ολόκληρης της εφαρμογής όχι απαραίτητα από μία ομάδα προγραμματιστών αλλά ακόμη και από έναν μόνο προγραμματιστή. Έτσι, η χρήση της JavaScript ως γλώσσα κωδικοποίησης της εφαρμογής ελαχιστοποίησε τις τεχνολογίες και τις γλώσσες που έπρεπε να μάθω, και προφανώς και τον χρόνο, για να επιτύχω το τελικό αποτέλεσμα.

Συμπερασματικά, η παρούσα διπλωματική εργασία ήταν προγραμματιστικά ιδιαίτερα εκτενής και απαιτούσε σημαντικό χρόνο εκμάθησης και προετοιμασίας. Η εφαρμογή μπορεί εύκολα να επεκταθεί περαιτέρω με νέες υπηρεσίες που θα παρέχονται στο χρήστη και αναλύονται παρακάτω.

5.2 Μελλοντικές Επεκτάσεις Εφαρμογής

Υπάρχουν πολλά διαφορετικά πράγματα που μπορούν να προστεθούν για να κάνουν αυτήν την εφαρμογή ακόμη καλύτερη και να επεκτείνουν και να βελτιώσουν τη λειτουργικότητά της.

Όσον αφορά το backend, κάποιες επεκτάσεις θα μπορούσαν να είναι οι ακόλουθες:

- Μία από τις πρώτες και ευκολότερες λειτουργίες, είναι ο χρήστης να μπορεί να ανεβάζει περισσότερα από ένα αρχεία σε μία ανάρτηση. Αυτήν τη στιγμή, ο χρήστης μπορεί να ανεβάσει μόνο μία μεμονωμένη εικόνα ή βίντεο, λόγω του frontend και συγκεκριμένα των απαιτήσεων CSS.
- Επιπλέον, μια πρόσθετη λειτουργικότητα της παραπάνω επέκτασης είναι ο χρήστης να μπορεί να ανεβάζει και εικόνες και βίντεο σε μία μόνο ανάρτηση.

Το Frontend μπορεί να επεκταθεί ως εξής:

- Μία από τις πιο κοινές και ευρέως χρησιμοποιούμενες βιβλιοθήκες στο React.js είναι η Redux. Το Redux είναι μια βιβλιοθήκη JavaScript για τη διαχείριση της κατάστασης εφαρμογής. Μια άλλη ευρέως χρησιμοποιούμενη λειτουργικότητα, ήδη μέρος του React.js, είναι τα hooks. Τα React Hooks επιτρέπουν στον προγραμματιστή να χρησιμοποιεί το state και άλλες δυνατότητες της React χωρίς να γράφει σε στοιχεία κλάσης (class components) αλλά σε λειτουργικά στοιχεία (functional components). Και οι δύο βιβλιοθήκες θα μπορούν να ενταχθούν σε αυτήν την εφαρμογή.
- Συνεχίζοντας την αναζήτηση μελλοντικών επεκτάσεων, πολλές αν όχι όλες οι υπάρχουσες εφαρμογές κοινωνικών μέσων, είναι επίσης διαθέσιμες σε Android ή και iOS. Το React Native είναι μια βιβλιοθήκη για τον προγραμματισμό εφαρμογών ιστού για να λειτουργούν σε συσκευές Android, ενώ η πλευρά του διακομιστή της εφαρμογής (backend) μπορεί να είναι ακριβώς η ίδια με αυτήν που χρησιμοποιεί μια εφαρμογή ιστού.
- Οι πιο συνηθισμένες και ευρέως χρησιμοποιούμενες εφαρμογές συνομιλίας έχουν ενεργές ειδοποιήσεις (push notifications) για να υποδείξουν στον χρήστη ότι έλαβε ένα νέο μήνυμα τη στιγμή που ακριβώς που ελήφθη το μήνυμα. Αυτό θα ήταν μια ωραία αναβάθμιση στη λειτουργικότητα της συνομιλίας.
- Τέλος, ο σχεδιασμός της εφαρμογής μπορεί να βελτιωθεί σε μεγάλο βαθμό. Η παρούσα διπλωματική εργασία εστίασε περισσότερο στην πλευρά του backend και λιγότερο στο frontend και πιο συγκεκριμένα στα σχεδιαστικά και CSS τμήματα του frontend τα οποία χρίζουν περαιτέρω βελτίωσης.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Appu Srva, "www.groupdiscussionideas.com," 02 June 2019. [Online]. Available: https://www.groupdiscussionideas.com/social-networking-in-our-lives/. [Accessed 20 January 2020].

[2] *Facebook Insights,* 2020.

[3] M. S. Mimoso, "Red Hat: Linux served at vertical data center dinner tables," 24 February 2003. [Online]. Available: https://searchdatacenter.techtarget.com/news/880604/Red-Hat-Linux-served-at-vertical-data-center-dinner-tables. [Accessed 22 January 2020].

[4] AbhiDoshi, "geeksforgeeks," [Online]. Available: https://www.geeksforgeeks.org/what-is-full-stack-development/. [Accessed 22 January 2020].

[5] T. Reenskaug and J. O. Coplien, "The DCI Architecture: A New Vision of Object-Oriented Programming," 20 March 2009.

[6] A. Chaffee, "jguru," 04 May 2012. [Online]. Available: http://www.jguru.com/faq/view.jsp?EID=129328. [Accessed 02 February 2020].

[7] R. Gibb, "stackpath," 31 May 2016. [Online]. Available: https://blog.stackpath.com/web-application/. [Accessed 02 February 2020].

[8] "stackify," stackify, [Online]. Available: https://stackify.com/web-application-architecture/. [Accessed 25 January 2020].

[9] S. Banga, "hackr," 12 January 2020. [Online]. Available: https://hackr.io/blog/web-application-architecture-definition-models-types-and-more. [Accessed 25 January 2020].

[10] J. Harband, S.-y. Guo, M. Ficarra and K. Gibbons, "Draft ECMA-262," *ECMAScript® 2021 Language Specification,* 2020.

[11] B. Eich, "Brendan Eich: An Introduction to JavaScript," in *JSConf*, 2010.

[12] *NETSCAPE AND SUN ANNOUNCE JAVASCRIPT, THE OPEN, CROSS-PLATFORM OBJECT SCRIPTING LANGUAGE FOR ENTERPRISE NETWORKS AND THE INTERNET.* [Interview]. 04 December 1995.

[13] stackoverflow, "Developer Survey Results," stackoverflow, 2019.

[14] "w3techs," 23 July 2020. [Online]. Available: https://w3techs.com/technologies/details/cp-javascript/. [Accessed 23 July 2020].

[15] P. Engineering, "medium," 22 November 2013. [Online]. Available: https://medium.com/paypal-engineering/node-js-at-paypal-4e2d1d08ce4f. [Accessed 03 February 2020].

[16] S. Burbeck, Applications Programming in Smalltalk-80(TM): How to use Model-View-Controller (MVC), 1997.

[17] L. Orsini, "readwrite," 07 November 2013. [Online]. Available: https://readwrite.com/2013/11/07/what-you-need-to-know-about-nodejs/. [Accessed 05 February 2020].

[18] CNET News staff, "cnet," 15 Octomber 1996. [Online].
Available: https://www.cnet.com/news/netscape-opens-intranet-attack/.
[Accessed February 2020].

[19] *npm early releases,* 2010.

[20] S. Yegulalp, "InfoWorld," 23 May 2016. [Online].
Available: https://www.infoworld.com/article/3047177/how-one-yanked-javascript-package-wreaked-havoc.html.
[Accessed February 2020].

[21] "npmjs," [Online]. Available: https://www.npmjs.com/.

[22] npmjs, "npmjs," [Online].
Available: https://www.npmjs.com/policies/conduct#acceptable-package-content.
[Accessed February 2020].

[23] P. Serby, "venturebeat," 07 January 2012. [Online].
Available: https://venturebeat.com/2012/01/07/building-consumer-apps-with-node/.
[Accessed 10 February 2020].

[24] "expressjs," [Online]. Available: https://expressjs.com/. [Accessed 10 February 2020].

[25] "npmjs," [Online]. Available: https://www.npmjs.com/package/socket.io.
[Accessed 10 February 2020].

[26] E. Vaati, "tusplus," 09 November 2018. [Online].
Available: https://code.tutsplus.com/tutorials/file-upload-with-multer-in-node--cms-32088.
[Accessed 11 February 2020].

[27] "github," [Online]. Available: https://github.com/expressjs/multer. [Accessed 11 February 2020].

[28] F. Hámori, "risingstack," 04 April 2018. [Online].
Available: https://blog.risingstack.com/the-history-of-react-js-on-a-timeline/.
[Accessed 12 February 2020].

[29] "reactjs," [Online]. Available: https://reactjs.org/. [Accessed 12 February 2020].

[30] "w3schools," [Online]. Available: https://www.w3schools.com/react/react_state.asp.
[Accessed 13 February 2020].

[31] M. Hamedani, "programmingwithmosh," 03 December 2018. [Online].
Available: https://programmingwithmosh.com/react/react-virtual-dom-explained/.
[Accessed 13 February 2020].

[32] M. Hamedani, "programmingwithmosh," 19 November 2018. [Online].
Available: https://programmingwithmosh.com/javascript/react-lifecycle-methods/.
[Accessed 13 February 2020].

[33] "reactjs," [Online].
Available: https://reactjs.org/docs/introducing-jsx.html. [Accessed 11 February 2020].

[34] N. Leavitt, "Will NoSQL Databases Live Up to Their Promise?," *Technology News,* p. 3, 2010.

[35] I. Fernandez, "No! to SQL and No! to NoSQL," *So Many Oracle Manuals, So Little Time,* 2013.

[36] D. Harris, "gigaom," 27 August 2013. [Online].
Available: https://gigaom.com/2013/08/27/10gen-embraces-what-it-created-becomes-mongodb-inc/.
[Accessed 15 February 2020].

[37] D. Kerby, "Why MongoDB is the Way to Go," *Database Zone,* 2015.

[38] "mongodb," [Online]. Available: https://www.mongodb.com/. [Accessed 15 February 2020].

[39] "npmjs," [Online]. Available: https://www.npmjs.com/package/mongoose. [Accessed 15 February 2020].

[40] D. Sevilla, "modeling-languages," 25 May 2018. [Online]. Available: https://modeling-languages.com/discovery-and-visualization-of-nosql-database-schemas/ [Accessed 18 February 2020].

[41] "git-scm," [Online]. Available: https://git-scm.com/book/en/v2/Getting-Started-A-Short-History-of-Git. [Accessed 18 February 2020].

[42] "visualstudio," [Online]. Available: https://code.visualstudio.com/. [Accessed 18 February 2020].

[43] F. Lardinois, "techcrunch," 14 April 2020. [Online]. Available: https://techcrunch.com/2020/04/14/github-is-now-free-for-all-teams/?guccounter=1&guce_referrer=aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnLw&guce_referrer_sig=AQAAAJ62XY-1l37umi9A8f8VE0MK1F2W7i6UQpU6iKN_li. [Accessed 23 July 2020].

ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

Αναφορές μερικών συντομογραφιών που χρησιμοποιήθηκαν κατά τη σύνταξη της διπλωματικής:

API	Application Programming Interface
HTML	Hypertext Markup Language
CSS	Cascading Style Sheet
JSON	JavaScript Object Notation
NPM	Node Package Manager
URL	Uniform Resource Locator
HTTP	Hypertext Transfer Protocol
DOM	Document Object Model
JSX	JavaScript XML
App	Application
MVC	Model-View-Controller
Repo	Repository