



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

**ΑΝΑΠΤΥΞΗ ΔΙΑΔΙΚΤΥΑΚΗΣ ΕΦΑΡΜΟΓΗΣ ΓΙΑ
ΕΠΕΞΕΡΓΑΣΙΑ ΠΗΓΙΑΙΟΥ ΚΩΔΙΚΑ**

Διπλωματική Εργασία

Σταυρινός Νικόλαος

Επιβλέπων: Θάνος Γεώργιος

Σεπτέμβριος 2022



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

**ΑΝΑΠΤΥΞΗ ΔΙΑΔΙΚΤΥΑΚΗΣ ΕΦΑΡΜΟΓΗΣ ΓΙΑ
ΕΠΕΞΕΡΓΑΣΙΑ ΠΗΓΙΑΙΟΥ ΚΩΔΙΚΑ**

Διπλωματική Εργασία

Σταυρινός Νικόλαος

Επιβλέπων: Θάνος Γεώργιος

Σεπτέμβριος 2022



UNIVERSITY OF THESSALY
SCHOOL OF ENGINEERING
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

**DEVELOPMENT OF A WEB APP FOR SOURCE CODE
EDITING**

Diploma Thesis

Stavrinos Nikolaos

Supervisor: Thanos Georgios

September 2022

Εγκρίνεται από την Επιτροπή Εξέτασης:

Επιβλέπων **Θάνος Γεώργιος**

Ε.ΔΙ.Π., Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

Μέλος **Τσαλαπάτα Χαρίκλεια**

Ε.ΔΙ.Π., Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

Μέλος **Φεύγας Αθανάσιος**

Ε.ΔΙ.Π., Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Θάνο Γεώργιο για την καθοδήγηση, την βοήθεια αλλά και τις συμβουλές που μου παρείχε κατά την διάρκεια της διπλωματικής μου εργασίας. Επίσης, θα ήθελα να ευχαριστήσω την κ. Τσαλαπάτα και τον κ. Φεύγα που δέχτηκαν να συμμετέχουν στην επιτροπή. Τέλος, ένα μεγάλο ευχαριστώ στην οικογένεια μου και στους φίλους μου που ήταν πάντα εκεί για μένα με την στήριξη και την παρουσία τους όλα αυτά τα χρόνια.

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. Δηλώνω επίσης ότι τα αποτελέσματα της εργασίας δεν έχουν χρησιμοποιηθεί για την απόκτηση άλλου πτυχίου. 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

Σταυρινός Νικόλαος

Διπλωματική Εργασία

ΑΝΑΠΤΥΞΗ ΔΙΑΔΙΚΤΥΑΚΗΣ ΕΦΑΡΜΟΓΗΣ ΓΙΑ ΕΠΕΞΕΡΓΑΣΙΑ ΠΗΓΙΑΙΟΥ ΚΩΔΙΚΑ

Σταυρινός Νικόλαος

Περίληψη

Το βασικό εργαλείο ενός προγραμματιστή είναι ο Editor στον οποίον μετατρέπει τις ιδέες του σε κώδικα. Με τον προγραμματισμό να γίνεται όλο και πιο δημοφιλής, τα τελευταία χρόνια υπάρχει μεγάλη ανάπτυξη και εξέλιξη των Online code editors. Σκοπός της παρούσας διπλωματικής εργασίας είναι η ανάπτυξη ενός Online code editor με την χρήση της τεχνολογίας MERN Stack (MongoDB, Express, React, Node). Τα κύρια χαρακτηριστικά της εφαρμογής είναι η δημιουργία απεριόριστων πρότζεκτ με ιεραρχία αρχείων/καταλόγων και η διαμοίραση αυτών μεταξύ των χρηστών, οι οποίοι θα μπορούν να τα επεξεργαστούν ταυτόχρονα. Ακόμα, η εφαρμογή συγχρονίζεται με το λογαριασμό του χρήστη στο Github για την εκτέλεση εντολών Git, υποστηρίζει πολλές γλώσσες προγραμματισμού και προσφέρει αποθήκευση των πρότζεκτ στον server της. Στην εργασία παρουσιάζεται η λειτουργική περιγραφή της εφαρμογής και αναλύονται οι τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη της, όπως και οι τεχνικές λεπτομέρειες της. Τέλος, αναφέρονται τα συμπεράσματα και οι μελλοντικές επεκτάσεις που σχεδιάζονται για την εφαρμογή.

Λέξεις-κλειδιά:

MERN Stack, MongoDB, Express, React, Node, Axios, Socket.IO, Git

Diploma Thesis

DEVELOPMENT OF A WEB APP FOR SOURCE CODE EDITING

Stavrinos Nikolaos

Abstract

The basic tool of a programmer is the Editor in which he turns his ideas into code. With programming becoming increasingly popular, in recent years there has been a great development and evolution of the Online code editors. The purpose of this thesis is the development of an Online code editor using the MERN Stack technology (MongoDB, Express, React, Node). The main features of the application are the creation of unlimited projects with a file/directory hierarchy and sharing them among users, who will be able to edit them simultaneously. Also, the application syncs with the user's Github account to run Git commands, supports multiple programming languages and offers storage for the projects on its server. The paper presents the functional description of the application and analyzes the technologies used for its development, as well as its technical details. Finally, conclusions and future extensions planned for the application are reported.

Keywords:

MERN Stack, MongoDB, Express, React, Node, Axios, Socket.IO, Git

Πίνακας περιεχομένων

Ευχαριστίες	ix
Περίληψη	xii
Abstract	xiii
Πίνακας περιεχομένων	xv
Κατάλογος σχημάτων	xix
Κατάλογος πινάκων	xxi
Συνομογραφίες	xxiii
1 Εισαγωγή	1
1.1 Αντικείμενο της διπλωματικής	1
1.2 Οργάνωση του τόμου	2
2 Λειτουργική περιγραφή της εφαρμογής	3
2.1 Εισαγωγή	3
2.2 Σύνδεση χρήστη	3
2.3 Δημιουργία λογαριασμού	4
2.4 Επιλογή Πρότζεκτ	5
2.5 Κεντρική σελίδα	6
2.5.1 File Explorer	7
2.5.2 Editor	8
2.5.3 Γραμμή πλοήγησης	9
2.6 Συνεργασία	10

2.7	Github	12
2.8	Ρυθμίσεις	14
2.9	Πρότζεκτ	15
2.10	Πληροφορίες	15
2.11	Επικοινωνία	16
2.12	Όνομα χρήστη	17
3	Τεχνολογίες που χρησιμοποιήθηκαν	19
3.1	Εισαγωγή	19
3.2	Node.js	19
3.3	Express	20
3.4	React	20
3.5	Redux	21
3.6	MongoDB	22
3.7	Axios	22
3.8	Socket.IO	23
3.9	Monaco Editor	23
3.10	Simple Git	24
4	Τεχνική περιγραφή	25
4.1	Εισαγωγή	25
4.2	Back End	25
4.2.1	Βάση δεδομένων	26
4.2.2	API του server	27
4.2.3	Δημιουργία λογαριασμού	29
4.2.4	Σύνδεση μέσω Github	30
4.2.5	Git	31
4.2.6	Συνεργασία χρηστών	32
4.2.7	Συμπίεση και αποσυμπίεση Zip αρχείων	32
4.3	Front End	33
4.3.1	Redux Store	33
4.3.2	Editor	34
4.3.3	Ιεραρχία φακέλων και αρχείων του πρότζεκτ	35

4.3.4	Επικοινωνία με Emailjs	35
5	Συμπεράσματα	37
5.1	Σύνοψη και συμπεράσματα	37
5.2	Μελλοντικές επεκτάσεις	37
	Βιβλιογραφία	39

Κατάλογος σχημάτων

2.1	Sign In	3
2.2	Sign Up	5
2.3	Projects	5
2.4	Choose one of your Projects	6
2.5	Main Page	7
2.6	File Explorer	7
2.7	Editor	9
2.8	Navbar	10
2.9	Collab	10
2.10	Share	11
2.11	Κεντρική σελίδα κατά την διάρκεια της συνεργασίας	12
2.12	Γραμμή πλοήγησης κατά την συνεργασία σε διαμοίραση	12
2.13	Clone	13
2.14	Push	14
2.15	Settings	15
2.16	Information	16
2.17	Contact	16
2.18	SignOut	17
3.1	Αρχιτεκτονική React Redux	21
3.2	Παράδειγμα MongoDB εγγράφου	22
3.3	Socket.IO Rooms	23
4.1	Αρχιτεκτονική MERN Stack	25
4.2	Npm πακέτα	26
4.3	Mongoose Schema	27

4.4	Οι μεταβλητές του Redux Store	34
-----	---	----

Κατάλογος πινάκων

4.1	API του server	28
4.2	Συνέχεια του API του server	29

Συντομογραφίες

κ.λπ.	και λοιπά
MERN	Mongo Express React Node
API	Application Programming Interface
NPM	Node Package Manager
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure

Κεφάλαιο 1

Εισαγωγή

Τα τελευταία χρόνια οι διαδικτυακές υπηρεσίες επεξεργασίας πηγαίου κώδικα (Online Code Editors, εν συντομία OCE) έχουν γίνει ιδιαίτερα δημοφιλείς ανάμεσα στους προγραμματιστές και η εξέλιξη τους είναι ραγδαία. Η ευκολία στη χρήση και τα εργαλεία που προσφέρουν τους κάνουν να ξεχωρίζουν. Πιο συγκεκριμένα προσφέρουν λειτουργίες που δεν υπάρχουν στους παραδοσιακούς επεξεργαστές, όπως η συνεργασία με άλλους προγραμματιστές, η ασφαλή αποθήκευση των αρχείων και η πρόσβαση σε αυτούς από οποιαδήποτε γεωγραφικό σημείο ή συσκευή. Αυτοί είναι μερικοί από τους λόγους που τους καθιστούν πιο ελκυστικούς στους προγραμματιστές, με αποτέλεσμα να προτιμούνται για την ανάπτυξη κώδικα.

1.1 Αντικείμενο της διπλωματικής

Σκοπός της παρούσας διπλωματικής είναι η δημιουργία ενός Online Code Editor που υποστηρίζει διαφορετικές γλώσσες προγραμματισμού, έχει τη δυνατότητα να συγχρονισμού με την υπηρεσία Github και δε χρειάζεται κάποια ρύθμιση πριν την χρήση του. Ο κάθε χρήστης μπορεί να δημιουργήσει απεριόριστα πρότζεκτ με ιεραρχία αρχείων/καταλόγων και να τα αποθηκεύσει στον εξυπηρετητή (server) της εφαρμογής. Επιπλέον, έχει την δυνατότητα να διαμοιραστεί τα πρότζεκτ του με άλλους χρήστες, οι οποίοι θα μπορούν να τα επεξεργαστούν σε συνεργασία μαζί του σε πραγματικό χρόνο. Η παρούσα εργασία κάνει χρήση της τεχνολογίας MERN Stack που χρησιμοποιείται εκτενώς για την ανάπτυξη διαδικτυακών εφαρμογών. Στο Κεφάλαιο 3 υπάρχει αναλυτική περιγραφή των τεχνολογιών που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής.

1.2 Οργάνωση του τόμου

Η συγκεκριμένη διπλωματική εργασία αποτελείται από πέντε κεφάλαια. Στο Κεφάλαιο 1 γίνεται μία εισαγωγή στο θέμα της διπλωματικής. Στο Κεφάλαιο 2 παρουσιάζεται η λειτουργική περιγραφή της εφαρμογής. Στο Κεφάλαιο 3 αναφέρονται οι τεχνολογίες που χρησιμοποιήθηκαν. Στο Κεφάλαιο 4 περιγράφονται τεχνικές της υλοποίησης. Τέλος, στο Κεφάλαιο 5 παραθέτονται τα συμπεράσματα και οι μελλοντικές επεκτάσεις που σχεδιάζονται για την εφαρμογή.

Κεφάλαιο 2

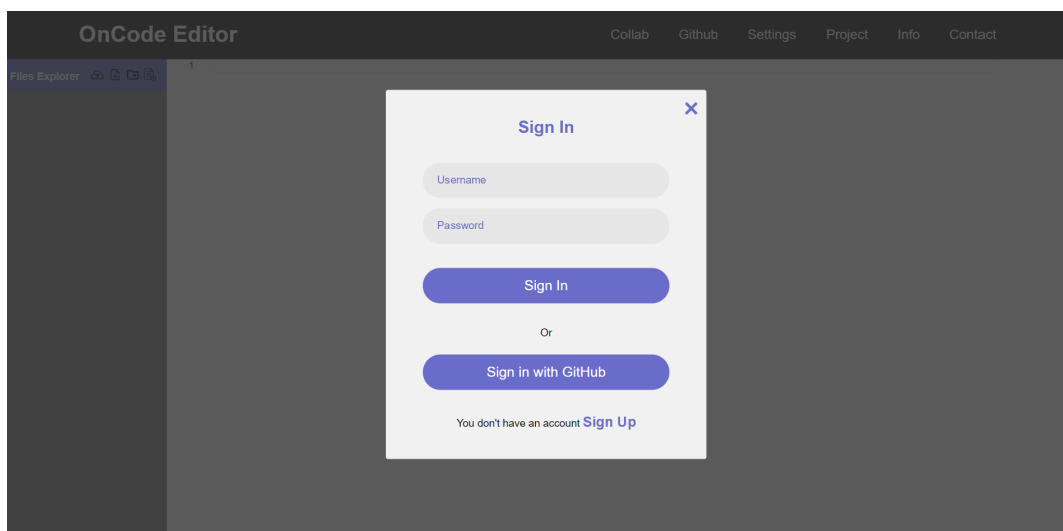
Λειτουργική περιγραφή της εφαρμογής

2.1 Εισαγωγή

Στο κεφάλαιο αυτό παρατίθεται μία λεπτομερή λειτουργική περιγραφή της εφαρμογής με την βοήθεια στιγμιοτύπων.

2.2 Σύνδεση χρήστη

Όταν ένας χρήστης εισέρχεται στην ιστοσελίδα της εφαρμογής αντικρίζει το αναδυόμενο παράθυρο της εικόνας 2.1 που του ζητάει να συνδεθεί. Για να συνεχίσει την περιήγηση του στην σελίδα είναι αναγκαίο να ολοκληρώσει αυτήν την διαδικασία.



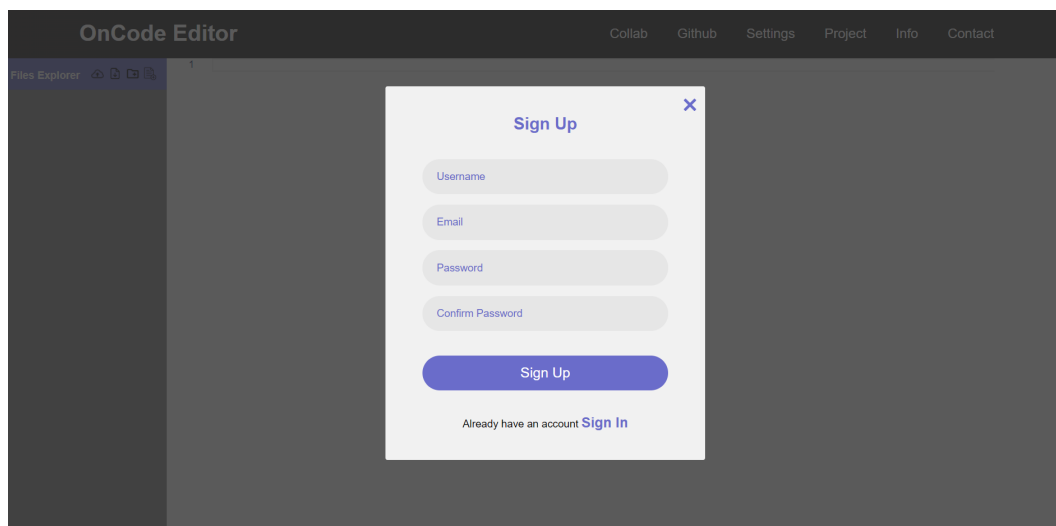
Σχήμα 2.1: Sign In

Για να συνδεθεί ο χρήστης στην εφαρμογή έχει δύο τρόπους, ο πρώτος είναι να συμπληρώσει το όνομα χρήστη και τον κωδικό του και να πατήσει το κουμπί Sign In. Ο δεύτερος τρόπος είναι να επιλέξει το κουμπί που αναγράφει Sign in with Github. Όταν το πατήσει θα εμφανιστεί ένα καινούργιο παράθυρο, το οποίο του ζητά να συνδεθεί στην υπηρεσία Github και εφόσον το κάνει, στη συνέχεια, του ζητάει να εξουσιοδοτήσει την εφαρμογή να έχει πρόσβαση στα στοιχεία του Github λογαριασμού του. Εφόσον απαντήσει θετικά, τότε ανδρομολογείται απευθείας στην εφαρμογή όπου είναι πια συνδεδεμένος. Αυτή την διαδικασία απαιτείται μόνο την πρώτη φορά που συνδέεται με αυτόν τον τρόπο. Τις επόμενες φορές η διαδικασία ταυτοποίησης είναι πολύ συντομότερη εφόσον είναι συνδεδεμένος στο Github.

Στην περίπτωση που ο χρήστης δεν έχει λογαριασμό στην εφαρμογή και επιθυμεί να δημιουργήσει ένα τοπικό λογαριασμό, τότε χρειάζεται να πατήσει στο έγχρωμο κείμενο Sign Up στο κάτω μέρος του παραθύρου.

2.3 Δημιουργία λογαριασμού

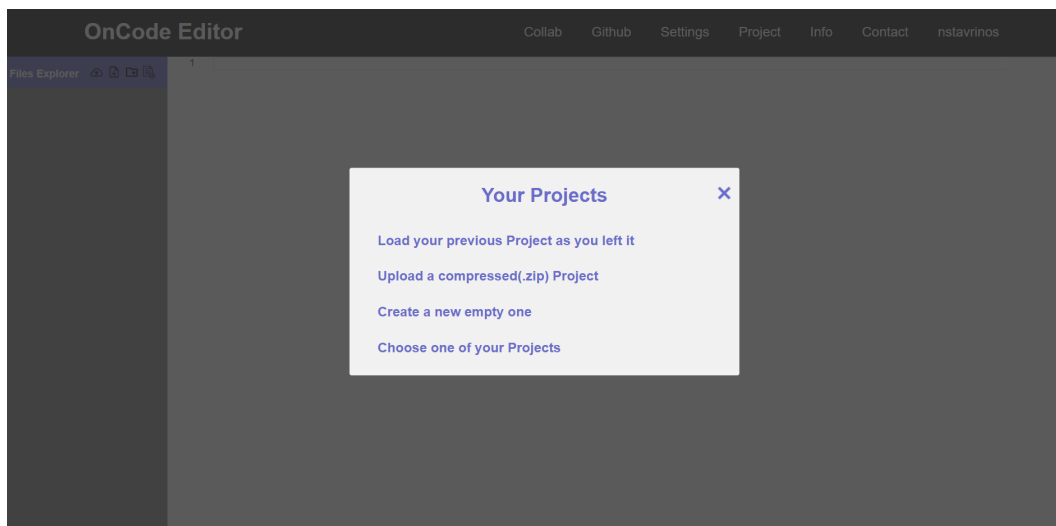
Εφόσον ο χρήστης έχει πατήσει το Sign Up του εμφανίζεται το παράθυρο που φαίνεται στην παρακάτω εικόνα 2.2. Για να δημιουργήσει ένα καινούργιο λογαριασμό χρειάζεται να συμπληρώσει ένα όνομα χρήστη, το οποίο πρέπει να είναι τουλάχιστον 4 χαρακτήρες αλλά και μοναδικό, εάν το όνομα χρήστη που επέλεξε υπάρχει ήδη τότε η εφαρμογή θα τον ειδοποιήσει ότι αυτό χρησιμοποιείται και να διαλέξει ένα διαφορετικό. Ακόμα, πρέπει να συμπληρώσει ένα έγκυρο email, διότι με την ολοκλήρωση της εγγραφής θα του αποσταλεί ένα email με ένα link, στο οποίο πρέπει να πατήσει για να επιβεβαιώσει το email του και μόνο τότε θα του επιτραπεί η είσοδος στην εφαρμογή. Επιπλέον, χρειάζεται να εισάγει έναν ασφαλή κωδικό, ο οποίος θα αποτελείται από τουλάχιστον ένα πεζό και ένα κεφαλαίο γράμμα της λατινικής αλφαβήτου, έναν αριθμό και τέλος έναν ειδικό χαρακτήρα (π.χ. @,!). Στο τελευταίο πεδίο ο χρήστης πρέπει να επαναλάβει τον κωδικό που εισήγαγε από πάνω, ώστε να βεβαιωθεί ότι τον συμπλήρωσε σωστά. Εάν έχει συμπληρώσει όλα τα πεδία σωστά τότε η εφαρμογή θα τον αφήσει να πατήσει το κουμπί Sign Up. Τέλος, η εφαρμογή θα τον δρομολογήσει απευθείας πίσω στο παράθυρο Sign In, αλλά πριν προσπαθήσει να συνδεθεί πρέπει πρώτα να επιβεβαιώσει το email του από το email που του έχει σταλθεί.



Σχήμα 2.2: Sign Up

2.4 Επιλογή Πρότζεκτ

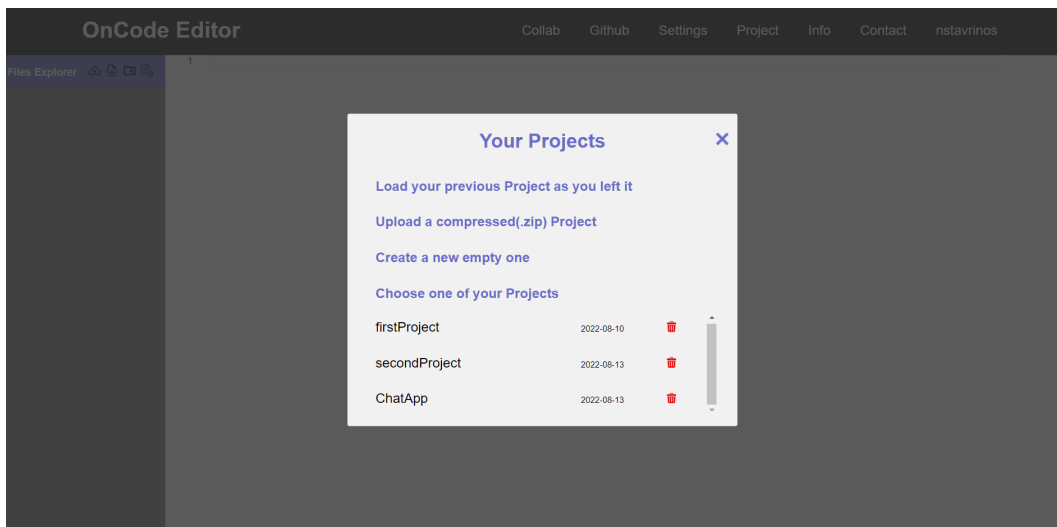
Αφού ο χρήστης συνδεθεί του εμφανίζεται ένα καινούργιο παράθυρο, όπως φαίνεται στην εικόνα 2.3, στο οποίο του δίνεται η επιλογή να διαλέξει ποιο πρότζεκτ θα ανοίξει στην εφαρμογή.



Σχήμα 2.3: Projects

Όπως φαίνεται στην παραπάνω εικόνα 2.3 έχει τέσσερις επιλογές:

- Η **Load your previous Project as you left it**, με την οποία η εφαρμογή ανοίγει το πρότζεκτ που επεξεργαζόταν την προηγούμενη φορά που χρησιμοποιούσε την εφαρμογή.
- Η **Upload a compressed(.zip) Project**, με την οποία ο χρήστης μπορεί να ανεβάσει ένα συμπιεσμένο πρότζεκτ τύπου .zip από τον υπολογιστή του, το οποίο αυτόματα θα αποσυμπιεστεί και θα ανοιχτεί στην εφαρμογή
- Η **Create a new empty one**, με την οποία δημιουργεί ένα καινούργιο άδειο πρότζεκτ με όνομα emptyProject που μπορεί να μετονομάσει στην συνέχεια.
- Η **Choose one of your Projects**, με την οποία ανοίγει ένα υπομενού με όλα τα πρότζεκτ που έχει αποθηκευμένα στην εφαρμογή. Στο κάθε πρότζεκτ εμφανίζεται το όνομα του, η ημερομηνία τελευταίας τροποποίησης και ένα εικονίδιο με το οποίο μπορεί να το διαγράψει. Ακόμα εάν πατήσει πάνω σε ένα από αυτά τότε ανοίγει το πρότζεκτ στην εφαρμογή και αυτή η επιλογή απεικονίζεται στην παρακάτω εικόνα 2.4.

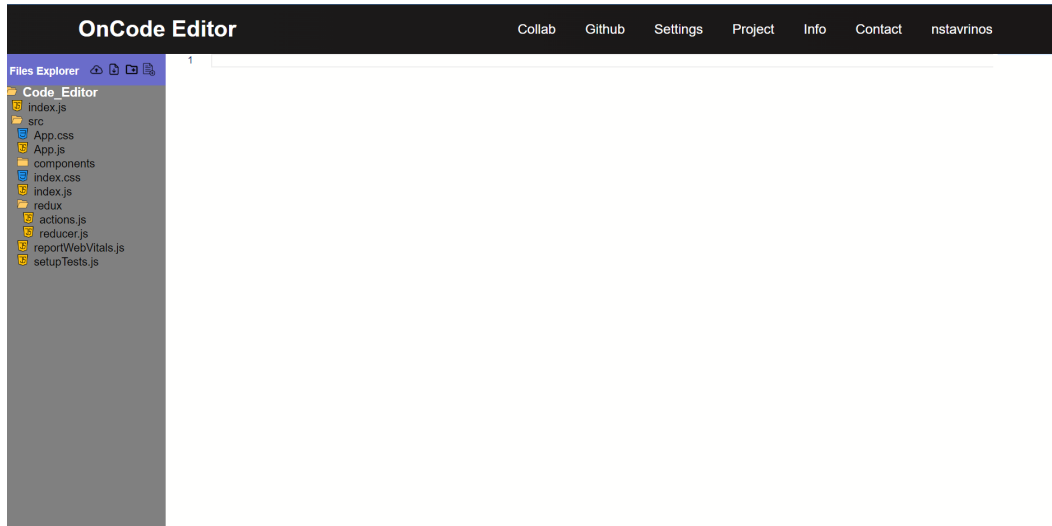


Σχήμα 2.4: Choose one of your Projects

2.5 Κεντρική σελίδα

Σε αυτό το σημείο για να γίνει περαιτέρω επεξήγηση της εφαρμογής χρησιμοποιείται ο χρήστης nstavrinος με ανοιγμένο το πρότζεκτ Code_Editor.

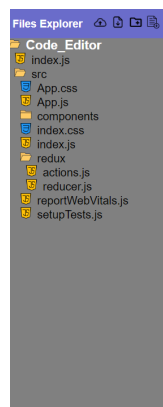
Εφόσον ο χρήστης έχει συνδεθεί και έχει επιλέξει κάποιο πρότζεκτ αντικρίζει την κεντρική σελίδα της εικόνας 2.5 που αποτελείται από τη γραμμή πλοήγησης, τον File Explorer και τον Editor, τα οποία αναλύονται το καθένα ξεχωριστά στις παρακάτω υποπαραγράφους.



Σχήμα 2.5: Main Page

2.5.1 File Explorer

Ο File Explorer, όπως φαίνεται και στην παρακάτω εικόνα 2.6, αποτελείται από δύο κομμάτια που είναι η γραμμή με τα βοηθητικά εικονίδια και η ιεραρχία αρχείων του πρότζεκτ.



Σχήμα 2.6: File Explorer

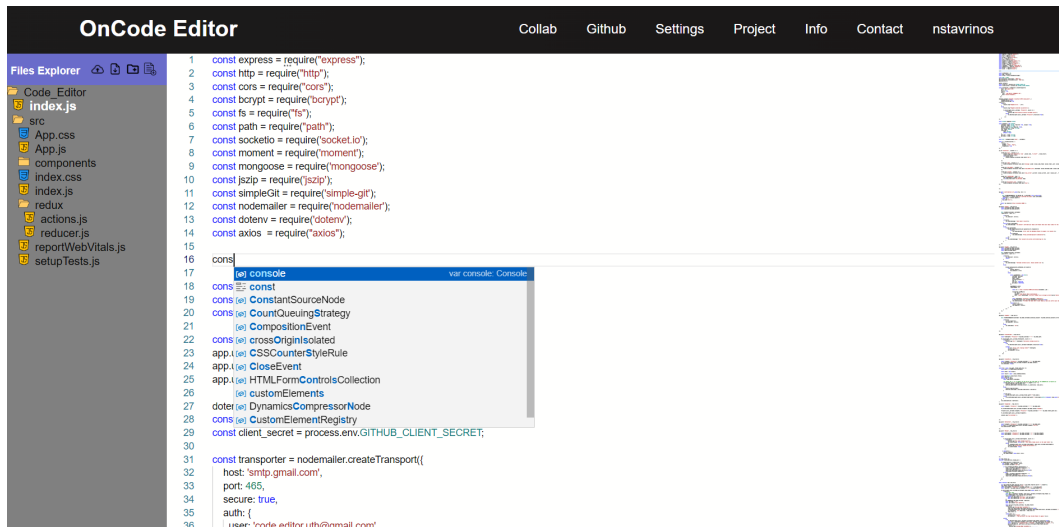
Στην γραμμή με τα βοηθητικά εικονίδια το πρώτο από τα αριστερά χρησιμοποιείται για να ανεβάσει ο χρήστης αρχεία από τον υπολογιστή του και έχει την δυνατότητα να ανεβάσει συμπιεσμένα αρχεία, τα οποία τα αποσυμπιέζει αυτόματα η εφαρμογή. Το δεύτερο από τα

αριστερά χρησιμοποιείται για να κατεβάσει ο χρήστης αρχεία από την εφαρμογή στον υπολογιστή του. Εάν έχει επιλέξει ένα αρχείο τότε το κατεβάζει απευθείας, ενώ αν έχει επιλέξει ένα φάκελο τότε η εφαρμογή συμπιέζει τα περιεχόμενα του και κατεβάζει το συμπιεσμένο αρχείο που δημιούργησε. Το τρίτο από αριστερά δίνει την δυνατότητα στο χρήστη να δημιουργήσει ένα καινούργιο φάκελο. Το τέταρτο και τελευταίο από αριστερά χρησιμοποιείται για την δημιουργία ενός καινούργιου αρχείου. Όλα τα εικονίδια εκτός αυτού που κατεβάζει αρχεία πριν τα πατήσει ο χρήστης πρέπει να επιλέξει έναν φάκελο από την δομή του πρότζεκτ, ώστε να ενεργοποιηθούν οι λειτουργίες τους. Τέλος, εάν ο χρήστης περάσει το ποντίκι του πάνω από αυτά τα εικονίδια τότε εμφανίζεται ένα μικρό παράθυρο με την επεξήγηση της λειτουργίας του.

Η ιεραρχία αρχείων του πρότζεκτ είναι όπως φανερώνει και το όνομα το μέρος όπου ο χρήστης μπορεί να διαχειριστεί τα αρχεία του πρότζεκτ του, τα οποία οργανώνονται σε μια δενδροειδή μορφή. Πατώντας πάνω σε ένα φάκελο ή ένα αρχείο αλλάζει το χρώμα με το οποίο αναγράφεται, ώστε να κάνει ευδιάκριτο ότι είναι επιλεγμένο. Επίσης, όταν ο χρήστης πατήσει σε ένα αρχείο τότε εμφανίζεται το περιεχόμενο του στον Editor και κάθε φορά που επιλέγει διαφορετικό αρχείο, η εφαρμογή αυτόματα αποθηκεύει τις αλλαγές που έχουν γίνει στο αρχείο που αποεπιλέχθηκε. Στην περίπτωση που ο χρήστης πατήσει δεξί κλικ σε έναν φάκελο τότε εμφανίζεται ένα μικρό μενού που του δίνει την δυνατότητα να δημιουργήσει ένα καινούργιο αρχείο ή φάκελο, να αντιγράψει το φάκελο, να κάνει επικόλληση μέσα σε αυτόν εάν έχει κάνει ήδη αντιγραφή κάποιο αρχείο, να κάνει αντιγραφή το path του, να τον μετονομάσει και τέλος να τον διαγράψει. Εναλλακτικά, αν πατήσει δεξί κλικ σε έναν αρχείο τότε το μενού του δίνει την δυνατότητα να το αντιγράψει, να κάνει αντιγραφή το path του, να το μετονομάσει και τέλος να το διαγράψει.

2.5.2 Editor

Ο Editor είναι η βασική λειτουργία της εφαρμογής και γι' αυτό καταλαμβάνει το μεγαλύτερο μέρος της οθόνης. Για την διευκόλυνση του χρήστη και την ευχρηστία της εφαρμογής, ο Editor υποστηρίζει κάποιες λειτουργίες χωρίς να χρειάζεται να ρυθμίσει οτιδήποτε ο χρήστης.



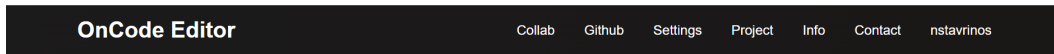
Σχήμα 2.7: Editor

Μια από τις υποστηριζόμενες λειτουργίες είναι ότι ο Editor υποστηρίζει διαφορετικές γλώσσες προγραμματισμού, όπως C, C++, C#, Javascript, Java, Json, Python, Php, Html, Css, Xml. Κάθε φορά που ο χρήστης μεταβάλλει το αρχείο που θέλει να επεξεργαστεί ο Editor αλλάζει αυτόματα στην κατάλληλη γλώσσα, την οποία την αντιλαμβάνεται από την κατάληξη του αρχείου. Αν ανοιχτεί κάποιο αρχείο που η γλώσσα του δεν υποστηρίζεται τότε χρησιμοποιείται η προκαθορισμένη γλώσσα που είναι η Java. Ακόμα, ο Editor βοηθάει τον χρήστη όταν γράφει κώδικα υπογραμμίζοντας με κόκκινο τα συντακτικά λάθη και δείχνοντας του γρήγορες προτάσεις για να ολοκληρώσει αυτό που γράφει. Άλλη μία λειτουργία είναι ότι στο δεξί μέρος του Editor βρίσκεται ένας μικρός χάρτης για προεπισκόπηση του κώδικα και γρήγορη μεταβίβαση σε διαφορετικά σημεία μέσα στο αρχείο. Επιπλέον, υποστηρίζει συντομεύσεις πληκτρολογίου, όπως CTRL + C για αντιγραφή, CTRL + V για επικόλληση, CTRL + A για επιλογή όλων, CTRL + Z για να αποκαταστήσει κάποια αλλαγή, CTRL + Y για να ξανακάνει κάποια αλλαγή, CTRL + F για γρήγορη αναζήτηση και CTRL + S για αποθήκευση του αρχείου. Τέλος, ο χρήστης μπορεί να εξατομικεύει τον Editor από τις ρυθμίσεις, όπως φαίνεται στην υποπαράγραφο 2.8.

2.5.3 Γραμμή πλοήγησης

Η γραμμή πλοήγησης που είναι η εικόνα 2.8 βρίσκεται στο πάνω μέρος της σελίδας και αποτελείται από το όνομα της εφαρμογής και επτά "κουμπιά" που το καθένα έχει από μία διαφορετική λειτουργία που επεξηγείται παρακάτω. Όταν το παράθυρο μικραίνει σε μέγεθος τότε στην γραμμή πλοήγησης εμφανίζεται μόνο το όνομα της εφαρμογής και ένα εικονίδιο

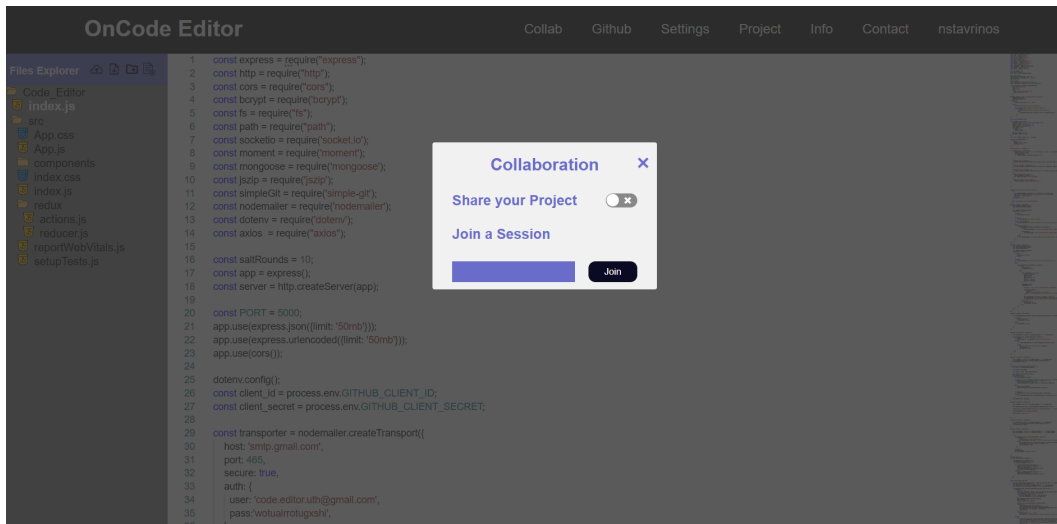
με τρεις γραμμές που όταν πατηθεί εμφανίζει ένα κάθετο μενού με τις εφτά λειτουργίες.



Σχήμα 2.8: Navbar

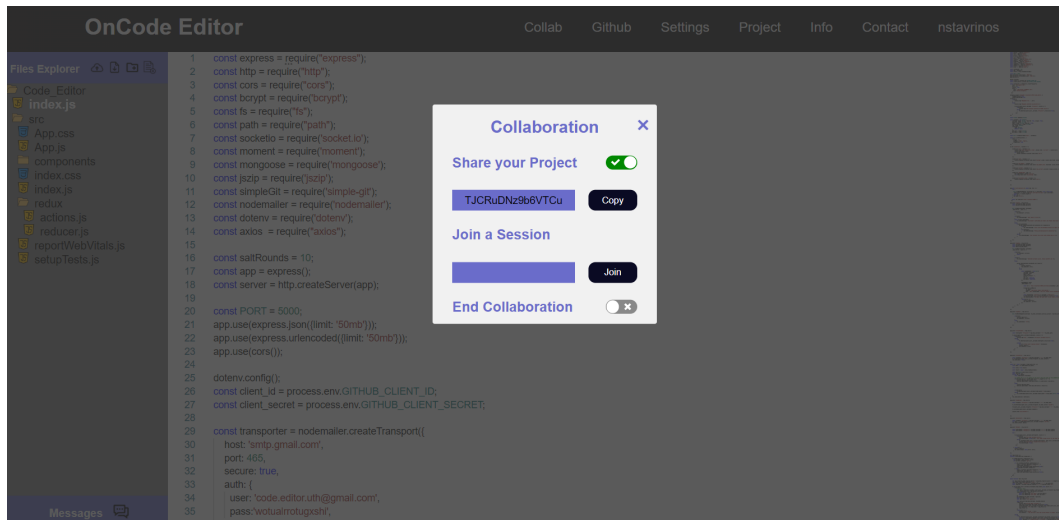
2.6 Συνεργασία

Η εφαρμογή υποστηρίζει συνεργασία μεταξύ χρηστών, οπότε εάν ο χρήστης επιθυμεί να διαμοιραστεί το πρότζεκτ του ή θέλει να συνεργαστεί με κάποιον άλλο χρήστη που έχει κάνει διαμοίραση τότε πρέπει να πατήσει στο "κουμπί" Collab και το αναδυόμενο παράθυρο της εικόνας 2.9 θα εμφανιστεί.



Σχήμα 2.9: Collab

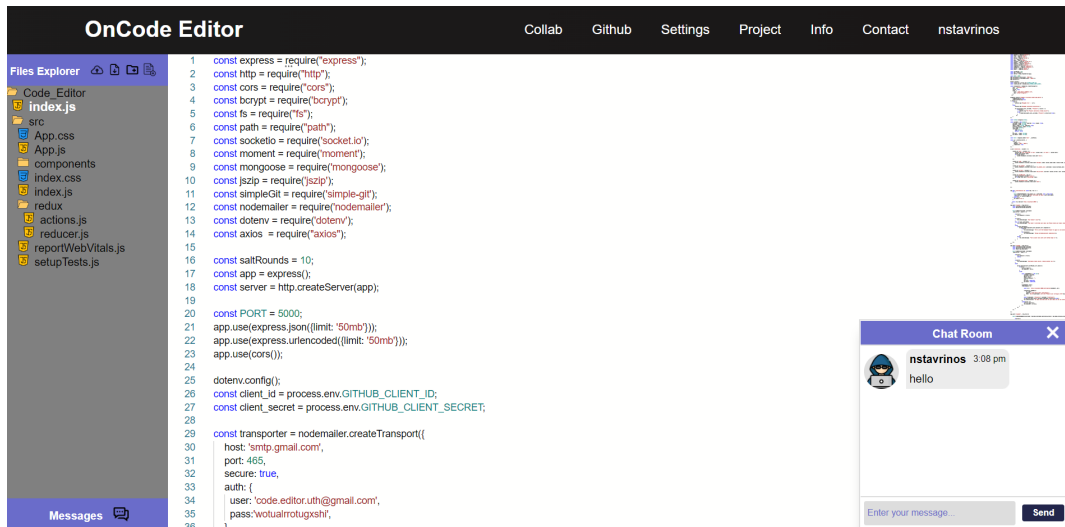
Όταν ο χρήστης θέλει να διαμοιραστεί το πρότζεκτ του τότε πατάει πάνω στο διακόπτη δίπλα στο Share your Project και εμφανίζεται από κάτω ένα αλφαριθμητικό που αποτελείται από δεκαπέντε χαρακτήρες, όπως φαίνεται στην εικόνα 2.10. Αυτό χρειάζεται να το αποστείλει σε όποιον θέλει να προσκαλέσει στην διαμοίραση και αφού πατήσει το κουμπί Cory το παράθυρο κλείνει απευθείας.



Σχήμα 2.10: Share

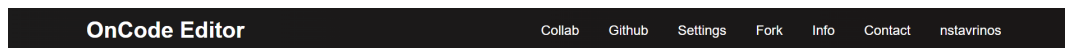
Όταν ο χρήστης θέλει να συνεργαστεί με κάποιον άλλον που έχει κάνει διαμοίραση, τότε πρέπει να πάρει το αλφαριθμητικό που του έστειλε και να το εισάγει στο μοβ χώρο που φαίνεται στην εικόνα 2.9. Αφού πατήσει το κουμπί Join θα συνδέσει στην διαμοίραση και το παράθυρο θα κλείσει απευθείας. Όταν συνδεθεί, θα του αποσταλούν αυτομάτως όλη η δομή του πρότζεκτ και το περιεχόμενο του Editor, έτσι ώστε να μπορεί και αυτός να συμμετάσχει στην συνεργασία του πρότζεκτ, στο οποίο έχει δικαιώματα επεξεργασίας.

Όταν ο χρηστής διαμοιράζει το πρότζεκτ του ή συνεργάζεται με κάποιον άλλο χρήστη που έχει κάνει διαμοίραση, τότε η κεντρική του σελίδα αλλάζει λίγο. Μία από αυτές τις αλλαγές είναι ότι εμφανίζεται στο κάτω αριστερά μέρος της σελίδας ένα κουμπί με όνομα Messages που αν πατηθεί ανοίγει στο κάτω δεξί μέρος της σελίδας ένα αναδυόμενο παράθυρο, μέσω του οποίου πραγματοποιείται συνομιλία μεταξύ των χρηστών που συνεργάζονται εκείνη την στιγμή, όπως φαίνεται στην εικόνα 2.11.



Σχήμα 2.11: Κεντρική σελίδα κατά την διάρκεια της συνεργασίας

Επιπλέον αλλάζει η γραμμή πλοήγησης, στην οποία αντικαθίσταται η επιλογή Project με την επιλογή Fork, όπως φαίνεται στην εικόνα 2.12 και αυτή πραγματοποιείται μόνο στον χρήστη που συνεργάζεται με κάποιον άλλο. Ο λόγος που γίνεται αυτό είναι για λόγους ιδιωτικότητας του διαμοιραστή και ώστε ο χρήστης να έχει την ικανότητα αν πατήσει το κουμπί Fork να μπορεί να αντιγράψει στον προσωπικό του χώρο στον server το πρότζεκτ που διαμοιράζεται.



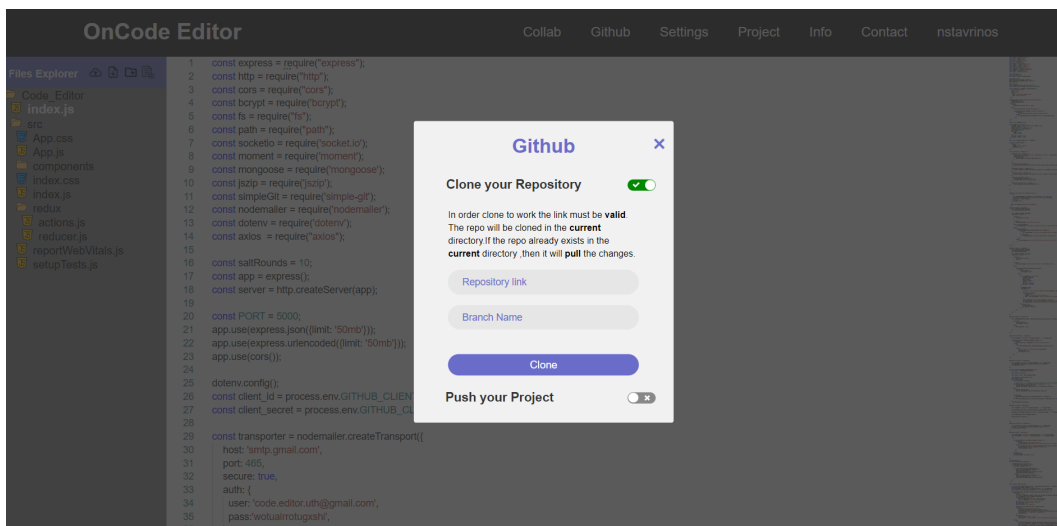
Σχήμα 2.12: Γραμμή πλοήγησης κατά την συνεργασία σε διαμοίραση

Όταν ο χρήστης θέλει να αποσυνδεθεί από την διαμοίραση ή να την σταματήσει τότε το μόνο που έχει να κάνει είναι να πατήσει το Collab στην γραμμή πλοήγησης και να πατήσει πάνω στο διακόπτη δίπλα στο End Collaboration. Αν ο χρήστης κάνει διαμοίραση τότε αποσυνδέει αυτόματα όλους τους χρήστες που είναι συνδεδεμένοι, ενώ αν είναι απλά συνδεδεμένος τότε καθαρίζει τον Editor και τον File Explorer από τα αρχεία που είχε και ανοίγει το παράθυρο από την εικόνα 2.3 για να επιλέξει ο χρήστης πιο πρότζεκτ θα ανοίξει. Τέλος, η κεντρική σελίδα γυρίζει στην αρχική της κατάσταση.

2.7 Github

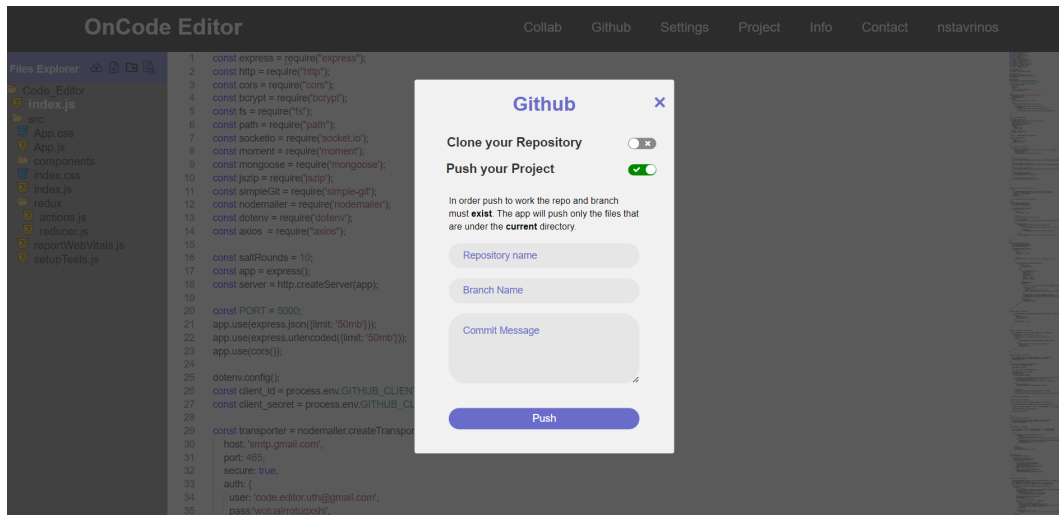
Η εφαρμογή έχει ενσωματωμένες τις εντολές Clone, Pull και Push του Git, στις οποίες ο χρήστης έχει πρόσβαση πατώντας το κουμπί Github και μόνο αν έχει συνδεθεί μέσω Github

ή αν έχει συνδέσει τον λογαριασμό του που έχει στην εφαρμογή με τον αυτόν στο Github. Αφού, πληρεί μία από τις προηγούμενες περιπτώσεις τότε του ανοίγει το παράθυρο της εικόνας 2.13, στο οποίο ο χρήστης μπορεί να αντιγράψει(Clone) ένα συγκεκριμένο branch ενός δημόσιου ή ιδιωτικού repository. Το μόνο που χρειάζεται είναι το link του repository και το όνομα του branch, τα οποία πρέπει να υπάρχουν. Πριν ανοίξει το παράθυρο πρέπει να έχει επιλέξει έναν φάκελο από το πρότζεκτ, στον οποίο μέσα θα αντιγραφεί το repository. Αν υπάρχει ήδη το όνομα του repository στο φάκελο που προσπαθεί να το αντιγράψει τότε η εφαρμογή αυτόματα τραβάει(Pull) όλες τις αλλαγές που έχουν γίνει στο repository και τις εφαρμόζει στον ήδη υπάρχοντα φάκελο. Μόλις τελειώσει μία από τις δύο διαδικασίες το παράθυρο θα κλείσει μόνο του.



Σχήμα 2.13: Clone

Ο χρήστης μπορεί ακόμα να στείλει και να αποθηκεύσει(Push) ένα repository στο Github. Αυτό μπορεί να το κάνει αφού στο παράθυρο της εικόνας 2.13 πατήσει τον διακόπτη δίπλα στο Push your Project, ο οποίος αλλάζει το περιεχόμενο του παραθύρου και μετατρέπεται σε αυτό στην εικόνα 2.14. Εφόσον το έχει πατήσει τότε μπορεί να στείλει και να αποθηκεύσει τον φάκελο του πρότζεκτ που έχει επιλέξει πριν ανοίξει το παράθυρο σε ένα repository, του οποίου είναι αυτός ο δημιουργός. Στην φόρμα που έχει το παράθυρο πρέπει να συμπληρώσει μόνο το όνομα του repository, το όνομα του branch, τα οποία πρέπει να υπάρχουν και ένα μήνυμα ώστε να ξεχωρίζει η αποθήκευση. Μόλις τελειώσει η διαδικασία το παράθυρο θα κλείσει μόνο του.

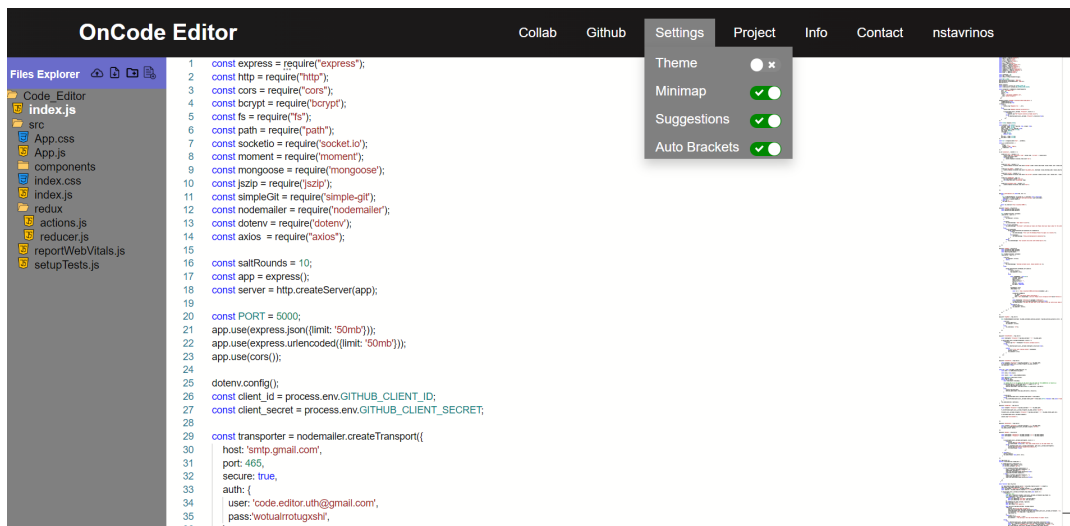


Σχήμα 2.14: Push

2.8 Ρυθμίσεις

Ο χρήστης έχει την δυνατότητα να εξατομικεύσει τον Editor πατώντας ή περνώντας τον κέρσορα πάνω από το κουμπί Settings, το οποίο φαίνεται στην εικόνα 2.15 και έχει τέσσερις επιλογές:

- Η **Theme**, με την οποία αλλάζει το θέμα του Editor από λευκό σε μαύρο και αντίστροφα.
- Η **Minimap**, με την οποία απενεργοποιείται ή ενεργοποιείται ο μικρός χάρτης του Editor.
- Η **Suggestions**, με την οποία απενεργοποιούνται ή ενεργοποιούνται οι γρήγορες προτάσεις του Editor.
- Η **Auto Brackets**, με την οποία απενεργοποιείται ή ενεργοποιείται το αυτόματο κλείσιμο των (, {, [στον Editor.



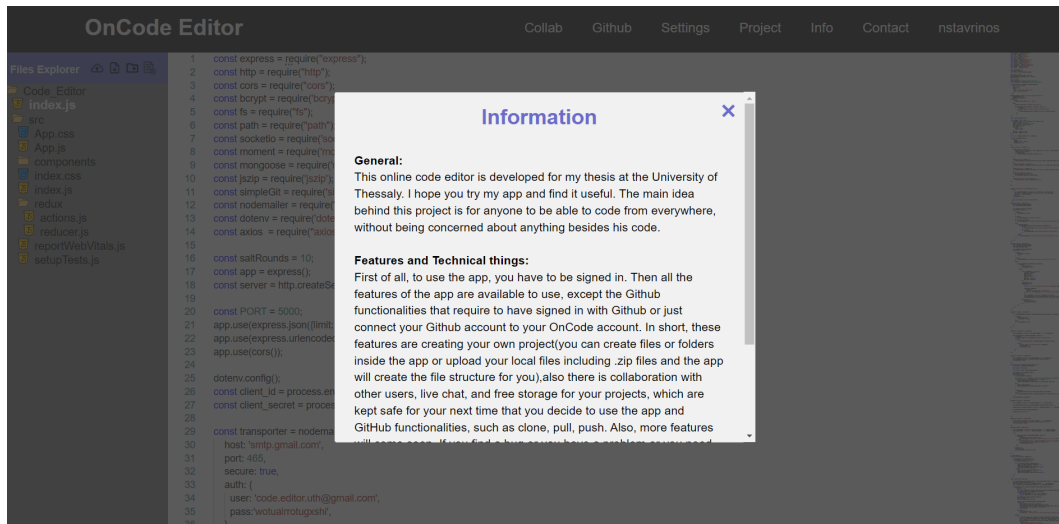
Σχήμα 2.15: Settings

2.9 Πρότζεκτ

Η επιλογή Project ανοίγει ένα παράθυρο που δίνει την δυνατότητα στον χρήστη να αλλάξει πρότζεκτ ενώ έχει κάποιο άλλο ανοιχτό και έχει ακριβώς τις ίδιες λειτουργίες με αυτό που παρουσιάστηκε στην υποπαράγραφο 2.4.

2.10 Πληροφορίες

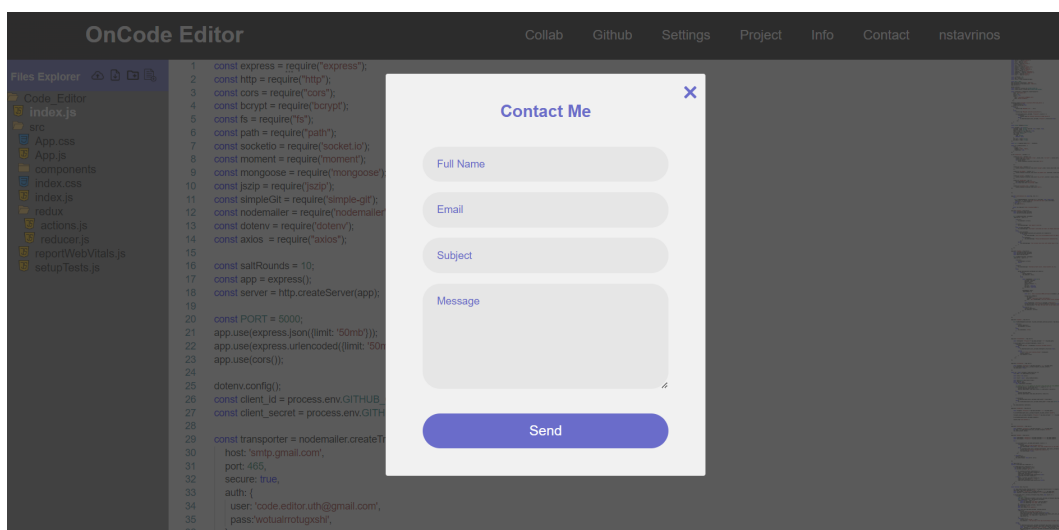
Το κουμπί Info ανοίγει το παράθυρο της εικόνας 2.16, το οποίο περιέχει διάφορες πληροφορίες για την εφαρμογή και τις λειτουργίες της και βοηθάει τον χρήστη να ενημερωθεί για αυτές.



Σχήμα 2.16: Information

2.11 Επικοινωνία

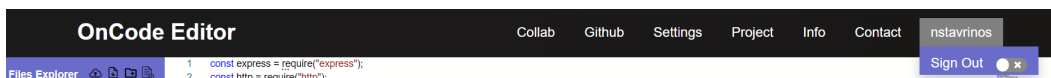
Ο χρήστης όταν αντιμετωπίζει κάποιο πρόβλημα ή για κάποιο λόγο θέλει να επικοινωνήσει με τον διαχειριστή της εφαρμογής, τότε πατάει το κουμπί Contact και ανοίγει το παράθυρο της εικόνας 2.17. Σε αυτό πρέπει να συμπληρώσει όλα τα πεδία της φόρμας, τα οποία είναι το όνομα του, το email στο οποίο θέλει να λάβει απάντηση, το θέμα και το μήνυμα που θέλει να αποστείλει.



Σχήμα 2.17: Contact

2.12 Όνομα χρήστη

Στην γραμμή πλοήγησης φαίνεται το όνομα του χρήστη το οποίο πατώντας το ή περνώντας τον κέρσορα πάνω, εμφανίζεται η επιλογή Sign Out, όπως στην εικόνα 2.18, που αν πατηθεί ο διακόπτης τότε ο χρήστης αποσυνδέεται από την εφαρμογή. Αυτός είναι και ο προτεινόμενος τρόπος αποσύνδεσης, ώστε να αποθηκευτεί η κατάσταση του πρότζεκτ.



Σχήμα 2.18: SignOut

Κεφάλαιο 3

Τεχνολογίες που χρησιμοποιήθηκαν

3.1 Εισαγωγή

Στην ανάπτυξη της εφαρμογής χρησιμοποιήθηκαν πολλές τεχνολογίες και εργαλεία, γι' αυτό παρακάτω αναλύεται το τεχνολογικό υπόβαθρο της εφαρμογής.

3.2 Node.js

Το Node.js [1] είναι ένα περιβάλλον ανοιχτού κώδικα για την εκτέλεση κώδικα JavaScript στην πλευρά του διακομιστή. Το συγκεκριμένο εργαλείο τρέχει σε πολλές πλατφόρμες, όπως Windows, Linux, Unix, Mac OS με την χρήση της V8 JavaScript μηχανής εκτέλεσης χρησιμοποιώντας ένα μόνο νήμα για την εκτέλεση εργασιών, το οποίο το καθιστά ιδανικό για την υποστήριξη εκατοντάδων συνδέσεων και αιτημάτων. Επίσης, είναι γραμμένο σε C, C++, και JavaScript, με αποτέλεσμα οι διαδικτυακές εφαρμογές που αναπτύσσονται με τη χρήση αυτού να είναι αποδοτικές και γρήγορες.

Ένα από τα πράγματα που κάνουν το Node.js να ξεχωρίζει είναι η ενσωματωμένη υποστήριξη του Npm. Το προαναφερόμενο είναι ένα πακέτο διαχείρισης Node.js πακέτων, τα οποία είναι βιβλιοθήκες ανοιχτού κώδικα που διευκολύνουν πολύ τον προγραμματιστή στην ανάπτυξη των εφαρμογών.

3.3 Express

Το Express [2] είναι ένα Node.js framework για την ανάπτυξη του back-end τόσο των διαδικτυακών όσο και των κινητών εφαρμογών. Το συγκεκριμένο ακόμα παρέχει ταυτόχρονα απλούς και ισχυρούς τρόπους για δημιουργία APIs για εξυπηρέτηση διάφορων αιτημάτων. Άλλο ένα χαρακτηριστικό του framework είναι ότι προσθέτει λειτουργίες χωρίς να επηρεάζει αυτά που κάνουν το Node.js να ξεχωρίζει. Σημαντικό είναι να αναφερθεί ότι είναι εύκολο στη χρήση και στην εκμάθηση. Αυτοί είναι μερικοί από τους λόγους που το καθιστούν το πιο δημοφιλές Node.js framework και πολλά άλλα δημοφιλή framework έχουν βασιστεί πάνω σε αυτό.

3.4 React

Το React [3] είναι μια JavaScript βιβλιοθήκη που χρησιμοποιείται για την κατασκευή user interfaces βασιζόμενη σε user interfaces components. Η υλοποίηση της έχει γίνει με τον συνδυασμό JavaScript και XML από την Facebook, αλλά είναι ανοιχτού κώδικα οπότε στην διατήρηση της βοηθάει και η προγραμματιστική κοινότητα. Τα βασικά χαρακτηριστικά που τη κάνουν να είναι τόσο διάσημη και πολυχρησιμοποιημένη είναι:

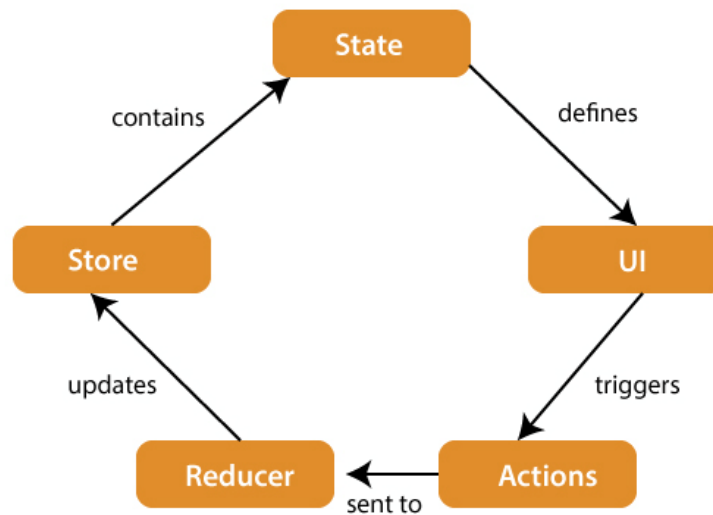
- Η εύκολη δημιουργία δυναμικών εφαρμογών.
- Η ανάπτυξη τόσο διαδικτυακών όσο και κινητών εφαρμογών.
- Η χρήση εικονικού DOM ώστε να γίνει σύγκριση των καταστάσεων των components και να αντικατασταθούν μόνο εκείνα που έχουν αλλάξει στο πραγματικό DOM. Αυτό βοηθάει στην βελτίωση της απόδοσης της εφαρμογής.
- Η μείωση της ώρα ανάπτυξης λόγω της επαναχρησιμοποίησης των components.
- Η εύκολη στην εκμάθηση.
- Τα ειδικά εργαλεία για αποσφαλμάτωση του κώδικα.

Για την ανάπτυξη React εφαρμογών συνήθως χρειάζονται και άλλες βιβλιοθήκες για την υλοποίηση διάφορων λειτουργιών καθώς το React κάνει μόνο διαχείριση καταστάσεων και αλλαγές στο DOM.

3.5 Redux

Το Redux [4] είναι ένας χώρος για αποθήκευση της κατάστασης των μεταβλητών της εφαρμογής. Επίσης, είναι μία JavaScript βιβλιοθήκη που μπορεί να χρησιμοποιηθεί από πολλά frameworks, αλλά για τον σκοπό της εργασίας αυτής χρησιμοποιήθηκε το React Redux. Η συγκεκριμένη βιβλιοθήκη δίνει την δυνατότητα στα React components να διαβάσουν, αλλά και να αλλάξουν την τιμή συγκεκριμένων μεταβλητών από το Store, το οποίο καθιστά αρκετά εύκολο την διαχείριση της κατάσταση των μεταβλητών με κέρδος ότι δεν υπάρχει κατεύθυνση ροής δεδομένων.

Η αρχιτεκτονική του Redux, όπου και φαίνεται στην παρακάτω εικόνα 3.1, αποτελείται από το Store που είναι ο χώρος όπου αποθηκεύεται η κατάσταση των μεταβλητών, το Action το οποίο είναι ένα φορτίο με την πληροφορία του συμβάντος που το δημιούργησε και στη συνέχεια αποστέλνεται από το user interface στον Reducer. Τέλος, τον Reducer που διαβάζει το περιεχόμενο του Action και ενημερώνει ανάλογα το Store.



Σχήμα 3.1: Αρχιτεκτονική React Redux

3.6 MongoDB

Το MongoDB [5] είναι μια ανοιχτού κώδικα, μη σχεσιακή (NoSQL) βάση δεδομένων για την αποθήκευση και διαχείριση μεγάλου όγκου δεδομένων. Σε αντίθεση με τις παραδοσιακές σχεσιακές βάσεις το MongoDB είναι προσανατολισμένο στα έγγραφα, το οποίο επιτρέπει ένα πιο ελαστικό μοντέλο αποθήκευσης δεδομένων. Αυτό σημαίνει ότι τα δεδομένα αποθηκεύονται σε συλλογές που αποτελούνται από JSON έγγραφα, τα οποία μέσα τους έχουν αποθηκευμένα σχεσιακά δεδομένα, όπως φαίνεται στην εικόνα 3.2. Το πλεονέκτημα αυτού του μοντέλου είναι ότι είναι εύκολο στη χρήση και επιτρέπει την αποθήκευση σύνθετων δεδομένων. Επιπλέον, το MongoDB υποστηρίζεται από όλα τα γνωστά λειτουργικά συστήματα και έχει υλοποίηση για πάνω από δέκα γλώσσες προγραμματισμού. Αυτοί είναι μερικοί από τους λόγους που το καθιστούν την πιο διάσημη NoSQL βάση δεδομένων.

```
{
  _id : "5373aadadcac133aad5b6660",
  name : "kaushal patel",
  email : "kp@example.com",
  courses : {
    course_name : "java",
    fees : "5000",
    duration : "3",
    professor : "g.r."
  }
}
```

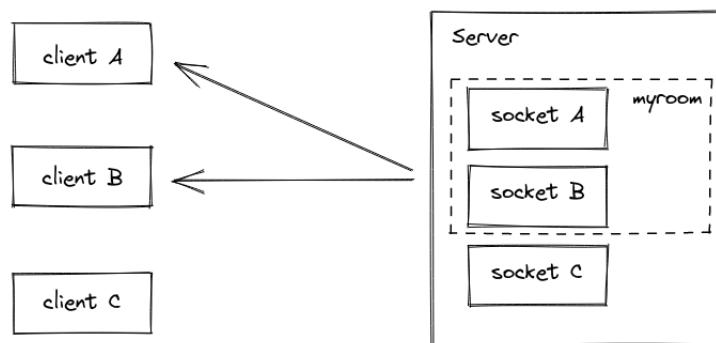
Σχήμα 3.2: Παράδειγμα MongoDB εγγράφου

3.7 Axios

Το Axios [6] είναι μία JavaScript βιβλιοθήκη που υλοποιεί ένα HTTP Client, το οποίο από την πλευρά του server κάνει HTTP αιτήματα ενώ από του περιηγητή κάνει XMLHttpRequests. Η βιβλιοθήκη αυτή είναι promise-based, οπότε μπορεί να χρησιμοποιηθεί ασύγχρονα. Άλλο ένα χαρακτηριστικό είναι η αυτόματη μετατροπή των δεδομένων του αιτήματος ή της απάντησης σε μορφή JSON, το οποίο μπορεί να αναιρεθεί και να μετατραπεί αν το επιθυμεί ο προγραμματιστής. Τέλος, το Axios υποστηρίζει όλες τις γνωστές HTTP μεθόδους όπως GET, POST, DELETE, PUT, PATCH.

3.8 Socket.IO

Το Socket.IO [7] είναι μια βιβλιοθήκη που είναι χτισμένη πάνω στο πρωτόκολλο Web-Socket και επιτυγχάνει την αμφίδρομη επικοινωνία μεταξύ client και server έχοντας πάντα χαμηλή καθυστέρηση. Αξιοσημείωτο είναι ότι εγγυάται HTTP long-polling, το οποίο είναι μία τεχνική όπου ο server στέλνει απευθείας πληροφορίες όταν είναι δυνατόν στον client, ή αυτόματη επανασύνδεση σε περίπτωση που χαθεί η σύνδεση. Για να επιτευχθεί η επικοινωνία πρέπει και ο client όπως και ο server να έχουν εγκαταστήσει την αντίστοιχη υλοποίηση της βιβλιοθήκης. Τέλος, το Socket.IO έχει την λειτουργία δωματίων, όπως και φαίνεται στην παρακάτω εικόνα 3.3, όπου ο server μπορεί να βάλει στο ίδιο δωμάτιο πολλούς clients με αποτέλεσμα να δημιουργήσει μία ομάδα που όλοι διαμερίζονται τα ίδια δεδομένα, αυτό μπορεί να βοηθήσει πολύ σε εφαρμογές με ανταλλαγή μηνυμάτων και συνεργασία μεταξύ χρηστών.



Σχήμα 3.3: Socket.IO Rooms

3.9 Monaco Editor

Ο Monaco Editor [8] έχει αναπτυχθεί από την Microsoft και είναι ο editor που χρησιμοποιείται στο Visual Studio Code. Ο συγκεκριμένος editor υποστηρίζεται από όλους τους γνωστούς περιηγητές και έχει ατελείωτες δυνατότητες και λειτουργίες. Μερικές από αυτές είναι:

- Υποστήριξη πάρα πολλών γλωσσών προγραμματισμού.
- Χρωματισμό των keywords κάθε γλώσσας.
- Αλλαγή θέματος του editor.

- Χρήση μικρού χάρτη για γρήγορη και εύκολη περιήγηση στο αρχείο.
- Γρήγορες προτάσεις για ολοκλήρωση λέξεων.
- Δυνατότητα χρήσης επιπλέον όψης για σύγκριση κώδικα
- Αυτόματο κλείσιμο των (, {.

3.10 Simple Git

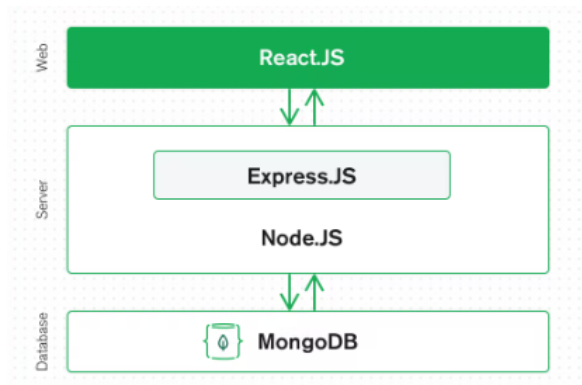
Το Simple Git [9] είναι μία μικρή βιβλιοθήκη για την εκτέλεση git εντολών μέσα από οποιαδήποτε Node.js εφαρμογή. Για να λειτουργήσει η βιβλιοθήκη χρειάζεται να είναι εγκαταστημένο και το git στον υπολογιστή-server που θα εκτελείται. Το Simple Git έχει για μεθόδους όλες τις εντολές git που υπάρχουν, όπως add, commit, clone, pull, push, κ.λ.π. , και ο προγραμματιστής τις καλεί με την σειρά που θα τις εκτελούσε σε ένα τερματικό. Τέλος, όλες οι βασικοί μέθοδοι είναι promise-based, οπότε μπορούν να χρησιμοποιηθούν και ασύγχρονα.

Κεφάλαιο 4

Τεχνική περιγραφή

4.1 Εισαγωγή

Η εφαρμογή υλοποιήθηκε με τη χρήση της αρχιτεκτονικής MERN Stack (MongoDB, Express, React, Node) [10], όπου καθιστά εύκολη την κατασκευή εφαρμογών που έχουν front end, back end και βάση δεδομένων. Ο κώδικας της διπλωματικής εργασίας βρίσκεται σε αυτό το δημόσιο Github repository [11].



Σχήμα 4.1: Αρχιτεκτονική MERN Stack

4.2 Back End

Το back end ή αλλιώς και server της εφαρμογής είναι γραμμένο σε Express.js και με τη χρήση του Npm του Node.js έχουν εγκατασταθεί 12 πακέτα που φαίνονται στην εικόνα 4.2, τα οποία βοηθάνε στην υλοποίηση κάποιων λειτουργιών. Στις παρακάτω ενότητες επεξηγούνται πως υλοποιήθηκαν οι βασικές λειτουργίες του server.

```
"dependencies": {  
  "axios": "^0.26.1",  
  "bcrypt": "^5.0.1",  
  "cors": "^2.8.5",  
  "dotenv": "^16.0.1",  
  "express": "^4.17.3",  
  "fs": "^0.0.1-security",  
  "fs-extra": "^10.1.0",  
  "jszip": "^3.10.0",  
  "mongoose": "^6.4.6",  
  "nodemailer": "^6.7.7",  
  "simple-git": "^3.10.0",  
  "socket.io": "^4.5.0"  
}
```

Σχήμα 4.2: Npm πακέτα

4.2.1 Βάση δεδομένων

Τα δεδομένα λογαριασμού κάθε χρήστη αποθηκεύονται στη MongoDB βάση που τρέχει στον τοπικό χώρο του server. Ο server για να συνδεθεί, αλλά και για να διαχειριστεί τη MongoDB χρησιμοποιεί την βιβλιοθήκη mongoose [12]. Αυτή η βιβλιοθήκη, μιας και η MongoDB είναι NoSQL, έχει την δυνατότητα να δημιουργήσει διάφορα Models, τα οποία αντιπροσωπεύουν την μορφή ενός αντικειμένου της MongoDB και βοηθούν στην ευκολότερη διαχείριση της βάσης. Για την αντιπροσώπευση κάθε χρήστη στην εφαρμογή έχει δημιουργηθεί ένα τέτοιο Model που ονομάζεται User και βασίζεται στο Schema που φαίνεται στην εικόνα 4.3, το οποίο ονομάζεται userData και αποτελείται από οχτώ πεδία. Αυτά τα πεδία είναι το όνομα χρήστη που είναι απαραίτητο και μοναδικό για κάθε χρήστη, ο κωδικός πρόσβασης, το email που είναι απαραίτητο, το όνομα του τελευταίου πρότζεκτ που επεξεργάστηκε, τον αριθμό τον πρότζεκτ που έχει δημιουργήσει, το αν έχει επιβεβαιωμένο email, το όνομα χρήστη και email που έχει στο Github.

```
const userData = new Schema({
  username: {type: String, required: true, unique: true},
  password: {type: String},
  email: {type: String, required: true},
  previous_project: {type: String},
  num: {type: Number},
  confirmed: {type: Boolean, default: false},
  git_user : {type: String},
  git_email: {type: String},
});
```

Σχήμα 4.3: Mongoose Schema

4.2.2 API του server

Όπως προαναφέρθηκε το Express είναι πολύ καλό στη δημιουργία APIs για την εξυπηρέτηση διάφορων αιτημάτων και αυτός είναι ο βασικός λόγος που επιλέχθηκε ως framework του server. Ο server, επειδή δέχεται συνέχεια αιτήματα από τους χρήστες που χρησιμοποιούν την εφαρμογή είναι σημαντικό τα δεδομένα που ανταλλάσσουν μεταξύ τους να είναι ασφαλή και ιδιωτικά. Αποτέλεσμα αυτού είναι η εφαρμογή να χρησιμοποιεί μόνο το HTTPS πρωτόκολλο για την επικοινωνία του client και server, το οποίο κρυπτογραφεί τα δεδομένα πριν την μεταφορά τους και τα αποκρυπτογραφεί όταν επαληθεύσει ότι έφθασαν στον σωστό παραλήπτη. Στους πίνακες 4.1, 4.2 φαίνονται όλα τα διαφορετικά Routes που αποτελούν το API και δίπλα στο καθένα δίνεται μία μικρή περιγραφή για το τι κάνει.

Πίνακας 4.1: API του server

Server API	
Routes	Λειτουργία
/api/SignIn	Συνδέει τον χρήστη στην εφαρμογή αν έχει δώσει σωστό όνομα χρήστη, κωδικό και έχει επιβεβαιωμένο email
/api/SignUp	Δημιουργεί λογαριασμό για τον νέο χρήστη και στέλνει email για να επιβεβαιώσει το email του
/api/confirmation/:id	Επιβεβαιώνει το email του χρήστη
/api/GitSignIn	Συνδέει τον χρήστη στην εφαρμογή μέσω Github και αν δεν έχει ήδη λογαριασμό στην εφαρμογή τότε του δημιουργεί έναν
/api/SignOut	Αποσυνδέει τον χρήστη από την εφαρμογή
/api/CreateFolder	Δημιουργεί ένα καινούργιο φάκελο
/api/CreateFile	Δημιουργεί ένα καινούργιο αρχείο ή ενημερώνει το περιεχόμενο του αν υπάρχει ήδη
/api/GetContent	Επιστρέφει το περιεχόμενο ενός αρχείου
/api/Rename	Μετονομάζει ένα αρχείο
/api/Paste	Επικολλεί ένα αρχείο ή ένα φάκελο με τα περιεχόμενα του στον φάκελο που έχει επιλέξει ο χρήστης
/api/DeleteFile	Διαγράφει ένα φάκελο με τα περιεχόμενα του
/api/DeleteFolder	Διαγράφει ένα αρχείο
/api/ZipUpload	Αποσυμπιέζει το .zip αρχείο που ανέβασε ο χρήστης
/api/ZipDownload	Συμπιέζει έναν φάκελο σε .zip αρχείο και το στέλνει στον χρήστη για να τον κατεβάσει στον υπολογιστή του

Πίνακας 4.2: Συνέχεια του API του server

Server API	
Routes	Λειτουργία
/api/ConnectGit	Προσθέτει στον λογαριασμό του χρήστη και τα στοιχεία του Github λογαριασμό του
/api/CloneRepo	Κάνει Clone ή Pull ένα Github repository
/api/PushRepo	Κάνει Push ένα τοπικό repository σε ένα Github repository
/api/NewEmptyProject	Δημιουργεί ένα καινούργιο πρότζεκτ
/api/AllProject	Επιστρέφει τα ονόματα όλων των πρότζεκτ του χρήστη
/api/SaveCurrentProject	Αποθηκεύει ένα πρότζεκτ ως τελευταίο χρησιμοποιημένο
/api/GetPreviousProject	Επιστρέφει την ιεραρχία του πρότζεκτ που χρησιμοποίησε την προηγούμενη φορά ο χρήστης
/api/OneProject	Επιστρέφει την ιεραρχία ενός συγκεκριμένου πρότζεκτ
/api/DeleteProject	Διαγράφει ένα πρότζεκτ
/api/Fork	Δημιουργεί ένα αντίγραφο του πρότζεκτ που διαμοιράζεται στον χώρο του συνδεδεμένου χρήστη

4.2.3 Δημιουργία λογαριασμού

Αφού ένας χρήστης συμπληρώσει την φόρμα με τα στοιχεία εγγραφής τότε αποστέλλονται στον server, ο οποίος πρώτα ελέγχει αν το όνομα χρήστη που επέλεξε χρησιμοποιείται ήδη. Εάν δεν χρησιμοποιείται τότε κρυπτογραφεί τον κωδικό πρόσβασης με την χρήση της βιβλιοθήκης bcrypt [13] για να είναι ασφαλής, αλλιώς αποστέλλει ανάλογο μήνυμα λάθους πίσω στον χρήστη. Στη συνέχεια, δημιουργεί ένα καινούργιο αντικείμενο στην βάση με δεδομένα το όνομα χρήστη, το email, τον κρυπτογραφημένο κωδικό και βάζοντας λάθος στο πεδίο επιβεβαιωμένο email.

Αφού τελειώσει με την ενημέρωση της βάσης τότε στέλνει ένα email με την χρήση της βιβλιοθήκης nodemailer [14] σε αυτό που συμπλήρωσε ο χρήστης με περιεχόμενο ένα μικρό μήνυμα και ένα link στο οποίο πρέπει να πατήσει, ώστε να μπορέσει να επιβεβαιώσει το email του και να του επιτραπεί η είσοδος στην εφαρμογή. Αυτό το link στην αρχή έχει το Route του server και στο τέλος του περιέχει το μοναδικό `_id` από το αντικείμενο της MongoDB που μόλις δημιούργησε. Όταν ο χρήστης το πατήσει τότε με τη μέθοδο HTTP GET ο server λαμβάνει αυτό το `_id` και μπορεί έτσι να ταυτοποιήσει στη βάση τον χρήστη και να ενημερώσει σε σωστό το πεδίο επιβεβαιωμένο email.

Τέλος, δημιουργεί έναν φάκελο με όνομα το όνομα χρήστη όπου μέσα σε αυτόν θα αποθηκεύονται όλα τα αρχεία των πρότζεκτ που θα δημιουργήσει ο χρήστης στην εφαρμογή. Τέλος, στέλνει απάντηση στο αίτημα του χρήστη, η οποία περιέχει ένα μήνυμα που εμφανίζεται στην σελίδα της εφαρμογής και αναφέρει ότι πρέπει να επιβεβαιώσει το email του για να μπορέσει να εισέλθει στην εφαρμογή.

4.2.4 Σύνδεση μέσω Github

Το Github υποστηρίζει το OAuth 2.0 πρωτόκολλο, το οποίο επιτρέπει σε εξωτερικές εφαρμογές να ζητούν εξουσιοδότηση για προσωπικά στοιχεία στον GitHub λογαριασμό ενός χρήστη χωρίς να λάβουν τον κωδικό πρόσβασής του. Για να υποστηρίξει αυτή την λειτουργία μία εφαρμογή πρέπει ο προγραμματιστής να καταχωρήσει την εφαρμογή του στην σελίδα με τις Github εφαρμογές, όπου δηλώνει το όνομα, το link της εφαρμογής και ένα callback link που το Github θα δρομολογήσει τον χρήστη όταν τελειώσει με όλη την διαδικασία.

Η διαδικασία για να συνδεθεί ένας χρήστης μέσω Github είναι απλή και εύκολη. Με το πάτημα του κουμπιού Sign with Github ένα νέο παράθυρο εμφανίζεται στην οθόνη του χρήστη, όπου ο χρήστης πρέπει να αποδεχτεί εάν εξουσιοδοτεί την εφαρμογή να έχει πρόσβαση στα προσωπικά του δεδομένα. Εφόσον δεχτεί, το Github του επιστρέφει έναν προσωρινό κωδικό τον οποίο τον στέλνει απευθείας στον server. Τη στιγμή που τον λάβει ο server στέλνει ένα αίτημα στο Github με την χρήση αυτού του κωδικού για να ζητήσει ένα access token. Αφού του το παραχωρήσει τότε στέλνει ένα τελευταίο αίτημα στο Github με την χρήση του access token αυτή την φορά, ώστε να του επιστρέψει τα προσωπικά στοιχεία του GitHub λογαριασμού του χρήστη. Στη συνέχεια, ο server ελέγχει αν το όνομα χρήστη και email που του επέστρεψε το Github υπάρχουν ήδη στη βάση δεδομένων. Αν υπάρχουν, τότε ο server επιστρέφει στον client το όνομα χρήστη που χρησιμοποιεί στην εφαρμογή και το access token

του Github. Αλλιώς δημιουργεί ένα νέο λογαριασμό με τα στοιχεία που του επέστρεψε το Github και στέλνει στον client τα δεδομένα που προαναφέρθηκαν. Όταν δημιουργείτε έτσι ο λογαριασμός του χρήστη, τότε δεν χρειάζεται να επιβεβαιώσει το email του, γιατί είναι ήδη επιβεβαιωμένο από το Github. Επίσης, από εδώ και πέρα ο χρήστης θα μπορεί να συνδεθεί στην εφαρμογή μόνο μέσω του Github, διότι δεν έχει θέσει κωδικό πρόσβασης κατά την δημιουργία του λογαριασμού του.

4.2.5 Git

Η εφαρμογή υποστηρίζει τις Git εντολές Clone, Pull, Push μέσω των μεθόδων της βιβλιοθήκης simple-git, οι οποίοι είναι promise-based οπότε μπορούμε να εκτελέσουμε τις εντολές με τη σωστή σειρά.

Όταν ο χρήστης αποφασίζει να αντιγράψει ένα remote Github repository, τότε ο server ελέγχει εάν υπάρχει ήδη στον τοπικό χώρο αποθήκευσης του και ανάλογα αποφασίζει αν θα εκτελέσει Clone ή Pull. Αν δεν υπάρχει εκτελεί Clone με επιλογές [”-single-branch”, ”-branch”, branch_name,”-depth”, ”1”] που σημαίνουν ότι θα φέρει μόνο το branch που ζήτησε ο χρήστης και μόνο την τελευταία έκδοση του. Αφού η Clone αρχικοποιεί από μόνη της το τοπικό repository δεν χρειάζεται να κάνουμε τίποτα άλλο εκτός από το να προσθέσουμε το remote repository, ώστε να μπορούμε να εκτελέσουμε και άλλες μεθόδους με αυτό στο μέλλον. Αν υπάρχει, τότε εκτελεί Pull με επιλογές [branch_name,’-verbose’,’-rebase’,’-autostash’] που σημαίνουν ότι θα φέρει το συγκεκριμένο branch που ζήτησε ο χρήστης και θα ενσωματώσει απευθείας τις αλλαγές στο τοπικό branch, το οποίο θα χρησιμοποιείται σαν βασικό branch από εδώ και πέρα.

Το Push είναι πιο περίπλοκο από τις άλλες δύο εντολές. Αρχικά ο server ελέγχει αν υπάρχει το remote repository στην λίστα του Git. Αν υπάρχει τότε προσθέτει όλα τα αρχεία που βρίσκονται μέσα στον φάκελο που διάλεξε ο χρήστης στο stash και κάνει ένα commit με το μήνυμα που έγραψε ο χρήστης. Τέλος, εκτελεί το Push με επιλογές [branch_name,”-set-upstream”,”-force”] που σημαίνουν ότι θα στείλει το συγκεκριμένο branch που ζήτησε ο χρήστης στο remote και για μελλοντικές εντολές θα χρησιμοποιείται αυτό το branch, αν δεν αναφερθεί κάποιο συγκεκριμένο. Αν δεν υπάρχει το remote, τότε πριν εκτελέσει τις προηγούμενες εντολές ο server πρέπει να αρχικοποιήσει το τοπικό repository και να προσθέσει το remote στην λίστα του Git.

4.2.6 Συνεργασία χρηστών

Η συνεργασία μεταξύ των χρηστών επιτυγχάνεται μέσω των socket της βιβλιοθήκης Socket.IO, τα οποία έχουν ρυθμιστεί να κρυπτογραφούν τα μηνύματα που ανταλλάσσουν, ώστε να μην μπορεί κανείς να υποκλέψει τα δεδομένα των χρηστών.

Κάθε φορά που κάποιος χρήστης επιλέγει να διαμοιραστεί το πρότζεκτ του τότε συνδέεται μέσω socket με τον server και αυτός δημιουργεί ένα δωμάτιο με όνομα αυτό που του έστειλε ο χρήστης. Στη συνέχεια βάζει μέσα σε αυτό το δωμάτιο τον χρήστη και όποιον άλλον θέλει να συνεργαστεί μαζί του. Ο server κρατάει μια λίστα με τα ενεργά δωμάτια, ώστε να μην φτιάξει πάνω από μία φορά το ίδιο δωμάτιο και δημιουργήσει πρόβλημα στην επικοινωνία των χρηστών.

Όταν ζητήσει να μπει κάποιος χρήστης σε ένα δωμάτιο τότε ο server στέλνει μήνυμα στους υπόλοιπους που είναι ήδη μέσα ότι κάποιος νέος μπήκε στο δωμάτιο. Σε αυτό το μήνυμα απαντάει μόνο αυτός που κάνει την διαμοίραση στέλνοντας πίσω όλα τα δεδομένα που χρειάζεται ο χρήστης ώστε να συμμετάσχει και αυτός στην συνεργασία. Αφού τα λάβει ο server, τα κάνει προώθηση στους υπόλοιπους χρήστες που είναι μέσα σε αυτό το δωμάτιο και έτσι αυτός που μπήκε τελευταίος ενημερώνει την κατάσταση των μεταβλητών του για να βλέπει την ίδια "εικόνα" του πρότζεκτ με τους υπόλοιπους.

Όταν γίνει κάποια αλλαγή στο περιεχόμενο του editor ή στην ιεραρχία αρχείων και φακέλων του πρότζεκτ ή επιλεγεί κάποιο διαφορετικό αρχείο ή φάκελος τότε αυτόματα ο client στέλνει αυτήν την αλλαγή στον server, ο οποίος με τη σειρά του προωθεί αυτή την αλλαγή στους υπόλοιπους του δωματίου, ώστε όλοι να έχουν την ίδια "εικόνα" του πρότζεκτ και να μπορούν να συνεργαστούν σωστά μεταξύ τους. Στην περίπτωση που αποσυνδεθεί ο χρήστης που κάνει την διαμοίραση, τότε ο server κλείνει την επικοινωνία με τους υπόλοιπους του δωματίου και διαγράφει το δωμάτιο.

4.2.7 Συμπίεση και αποσυμπίεση Zip αρχείων

Όταν ο χρήστης ανεβάζει ένα συμπιεσμένο .zip αρχείο τότε ο client διαβάζει και στέλνει στον server το όνομα του αρχείου και τα δεδομένα του σε τύπο base64. Ο server τότε δημιουργεί προσωρινά το συμπιεσμένο αρχείο στον τοπικό του χώρο αποθήκευσης και με την χρήση της βιβλιοθήκης jszip [15] αποσυμπιέζει ένα ένα τα αρχεία που βρίσκονται μέσα σε αυτό. Αφού τελειώσει με την αποσυμπίεση τότε διαγράφει το .zip αρχείο που δημιούργησε στην αρχή και στέλνει πίσω στον client μία λίστα με τα αρχεία που είχε το συμπιεσμένο

αρχείο, ώστε ο χρήστης να έχει πρόσβαση σε αυτά.

Ο χρήστης έχει την δυνατότητα να κατεβάσει στον προσωπικό του υπολογιστή ένα ολόκληρο πρότζεκτ ή ένα μόνο φάκελο με τα περιεχόμενα του. Η διαδικασία για να επιτευχθεί αυτή η λειτουργία είναι σχεδόν αντίστροφη από αυτή που αναφέρεται στην προηγούμενη παράγραφο. Αρχικά με μία αναδρομική συνάρτηση διαβάζει τα περιεχόμενα όλου του φακέλου και τα συμπιέζει, αφού τελειώσει με αυτή την διαδικασία τότε μετατρέπει τα δεδομένα σε τύπο base64 και τα στέλνει στον client. Από την πλευρά του ο client λαμβάνει τα δεδομένα και τα μετατρέπει σε αντικείμενο Blob και με την χρήση της βιβλιοθήκης file-saver κατεβάζει και αποθηκεύει το .zip αρχείο στον υπολογιστή του χρήστη.

4.3 Front End

Το front end ή αλλιώς και client της εφαρμογής είναι γραμμένο σε React, το οποίο είναι ιδανικό για την κατασκευή user interfaces. Στο Κεφάλαιο 2 παρουσιάζεται η λειτουργική περιγραφή του front end, ενώ σε αυτή την ενότητα αναλύεται η τεχνική πίσω από κάποιες λειτουργίες του.

4.3.1 Redux Store

Το Redux Store χρησιμοποιείται για την αποθήκευση της κατάστασης σημαντικών μεταβλητών της εφαρμογής, στις οποίες χρειάζεται να έχουν πρόσβαση όλα τα components. Στην παρακάτω εικόνα 4.4 φαίνονται ποιες είναι αυτές οι μεταβλητές και η αρχική τιμή που έχει η καθεμία, οι οποίες έχουν κύριο ρόλο για την σωστή λειτουργία της εφαρμογής. Για την επεξεργασία της κάθε μεταβλητής υπάρχει και ένα ξεχωριστό Action που στέλνεται στον Reducer που αλλάζει την κατάσταση των μεταβλητών που επηρεάζει το συγκεκριμένο Action.

```
const INITIAL_STATE = {
  language: "java",
  code: "",
  theme: "light",
  minimap: true,
  quickSuggestion: true,
  autoClosingBrackets: "always",
  files: undefined,
  current: undefined,
  username: undefined,
  room: undefined,
  socket: undefined,
  writting: false,
  host: true,
  loggedIn: false,
  messages: [],
  enableChat: false,
  hostname: undefined,
  fcode: undefined,
  token: undefined,
  git_username: undefined,
  rename_flag: false,
  copy_filepath: ''
};
```

Σχήμα 4.4: Οι μεταβλητές του Redux Store

4.3.2 Editor

Ο editor που χρησιμοποιείται στην εφαρμογή είναι ο Monaco Editor που είναι και ο καλύτερος δωρεάν editor που υπάρχει. Η χρήση του γίνεται μέσω της βιβλιοθήκης `@monaco-editor/react`, η οποία καθιστά την ρύθμιση του editor πολύ εύκολη. Οι αρχικές ρυθμίσεις του μπορούν να τροποποιηθούν από τον χρήστη μέσω των ρυθμίσεων που βρίσκονται στην γραμμή περιήγησης. Η γλώσσα με την οποία κάνει έλεγχο σύνταξης ο editor καθορίζεται αυτόματα κάθε φορά από τον τύπο του αρχείου που είναι ανοιχτό. Ακόμα, ο editor έχει από την βιβλιοθήκη υλοποιημένη την αντιγραφή, επικόλληση, αποκοπή, αναζήτηση και αντικατάσταση λέξεων. Μια επιπλέον λειτουργία που του έχει προστεθεί είναι η αποθήκευση των αλλαγών που γίνονται στο αρχείο που επεξεργάζεται ο χρήστης. Το συγκεκριμένο υλοποιείται μέσω ενός `EventListener` που ελέγχει αν πατήθηκε στο πληκτρολόγιο ταυτόχρονα το CTRL και S πλήκτρο. Όταν συμβεί αυτό τότε ο client στέλνει στον server το όνομα του

αρχείου που είναι ανοιχτό και το περιεχόμενο του editor, ώστε ο server να αποθηκεύσει τις αλλαγές του αρχείου.

4.3.3 Ιεραρχία φακέλων και αρχείων του πρότζεκτ

Ένα από τα βασικά components του front end είναι η ιεραρχία φακέλων και αρχείων του πρότζεκτ. Αυτή η ιεραρχία υλοποιείται μέσα από μία δένδροειδή δομή που διατηρείται στο Redux Store και αποτελείται από κόμβους που ανάλογα αν είναι φάκελος ή αρχείο έχουν διαφορετική μορφή. Οι διαφορές μεταξύ αυτών των δύο βρίσκονται κυρίως στις μεθόδους που υλοποιεί το καθένα. Για παράδειγμα ο φάκελος από το αρχείο έχει επιπλέον τις μεθόδους δημιουργίας νέου αρχείου ή φακέλου και επικόλλησης κάποιου φακέλου ή αρχείου στα περιεχόμενα του. Από την άλλη κάθε φορά που ο χρήστης επιλέγει κάποιο αρχείο τότε το περιεχόμενο του προβάλλεται στον editor.

Στο παρασκήνιο της εφαρμογής κάθε φορά που ο χρήστης αλλάζει αρχείο που θέλει να επεξεργαστεί τότε η εφαρμογή στέλνει αυτόματα το περιεχόμενο του editor στον server, ώστε να αποθηκεύσει τις αλλαγές που έχουν γίνει. Τέλος, έχει υλοποιηθεί ένα μικρό μενού με χρήσιμες λειτουργίες που ανοίγει όταν ο χρήστης πατήσει δεξί κλικ πάνω σε ένα κόμβο. Αυτές οι λειτουργίες αν έχει πατηθεί πάνω σε αρχείο είναι αντιγραφή, αντιγραφή του path, μετονομασία και διαγραφή του αρχείου, ενώ αν είναι φάκελος έχει τα ίδια με το αρχείο και επιπλέον την δημιουργία αρχείου ή φακέλου και την επικόλληση.

4.3.4 Επικοινωνία με Emailjs

Ο χρήστης μέσω της εφαρμογής έχει την δυνατότητα να επικοινωνήσει με τον διαχειριστή της εφαρμογής. Αυτό γίνεται με την χρήση της βιβλιοθήκης emailjs-com που είναι ικανή να στείλει email απευθείας από την πλευρά του client. Πιο αναλυτικά ο προγραμματιστής πρέπει να εγγραφεί στην σελίδα του Emailjs και να δηλώσει ένα email από το οποίο θα στέλνονται τα email. Στη συνέχεια, χρειάζεται να δημιουργήσει ένα προκαθορισμένο email, το οποίο θα έχει για παραμέτρους τα πεδία της φόρμας που συμπληρώνει ο χρήστης. Για να λειτουργήσει σωστά η συμπλήρωση του προκαθορισμένου email πρέπει να έχουν όμοια ονόματα τα πεδία της φόρμας και οι παράμετροι.

Σημαντικό είναι να αναφερθεί ότι αυτή η υπηρεσία προσφέρει δωρεάν 200 email τον μήνα και αν ξεπεραστεί αυτός ο αριθμός τότε χρειάζεται πληρωμή. Τέλος, η χρήση της

emailjs-com είναι πολύ εύκολη καθώς το μόνο που χρειάζεται για να σταλθεί ένα email είναι η μέθοδος `emailjs.sendForm(serviceID, templateID, templateParams, publicKey)`, όπου τα `serviceID`, `templateID`, `publicKey` βρίσκονται στον λογαριασμό Emailjs και σαν `templateParams` θέτονται οι τιμές των πεδίων της φόρμας.

Κεφάλαιο 5

Συμπεράσματα

5.1 Σύνοψη και συμπεράσματα

Ο στόχος της παρούσας εργασίας ήταν η δημιουργία ενός Online code editor με την χρήση της τεχνολογίας MERN Stack. Η τεχνολογία MERN Stack αποδείχθηκε εξαιρετικά ευέλικτη στη φάση της ανάπτυξης του κώδικα της εφαρμογής. Επιπλέον, παρέχονται στον προγραμματιστή πολλά συμπληρωματικά εργαλεία για την προσθήκη επιπλέον λειτουργιών και δυνατοτήτων. Δεν πρέπει να ξεχνάμε, ότι η ανάπτυξη μίας εφαρμογής τέτοιου μεγέθους χρειάζεται προσεκτικό σχεδιασμό, ώστε μετά την ανάπτυξη του κώδικα να είναι εύκολη η διαχείριση και η αποσφαλμάτωση της.

5.2 Μελλοντικές επεκτάσεις

Με την ολοκλήρωση της διπλωματικής εργασίας, η εφαρμογή βρίσκεται σε ένα ικανοποιητικό επίπεδο. Η εφαρμογή όμως παραμένει ανοικτή σε βελτιώσεις και προσθήκες που θα την κάνουν πιο ελκυστική στους τελικούς χρήστες. Μια μελλοντική επέκταση της εφαρμογής είναι όταν ο χρήστης επιλέγει ποιο πρότζεκτ θα ανοίξει στην εφαρμογή να έχει την δυνατότητα να επιλέξει από μία λίστα έτοιμων και γνωστών templates. Μία λειτουργία που αυτή τη στιγμή δεν παρέχεται είναι η εφαρμογή να υποστηρίζει μέθοδο για ανάκτηση του κωδικού πρόσβασης. Τέλος, σημαντική επέκταση της εφαρμογής θα ήταν να υποστηρίζει shell, στο οποίο οι χρήστες θα έχουν την δυνατότητα να μεταγλωττίζουν και να εκτελέσουν τον κώδικα που γράφουν.

Βιβλιογραφία

- [1] M. Casciaro and L. Mammino. *Node.js Design Patterns*. Packt Publishing, Birmingham, United Kingdom, 3 edition, 2020.
- [2] E. Brown. *Web Development with Node and Express : Leveraging the JavaScript Stack*. O'Reilly Media, Sebastopol, United States, 2 edition, 2019.
- [3] A. Banks. *Learning React: Functional Web Development with React and Redux*. O'Reilly Media, Sebastopol, United States, 1 edition, 2017.
- [4] React redux. <https://www.javatpoint.com/react-redux>. Ημερομηνία πρόσβασης: 15-03-2022.
- [5] Mongodb. <https://www.mongodb.com>. Ημερομηνία πρόσβασης: 01-04-2022.
- [6] Axios. <https://axios-http.com>. Ημερομηνία πρόσβασης: 30-03-2022.
- [7] T. Cadenhead. *Socket.IO Cookbook*. Packt Publishing, Birmingham, United Kingdom, 1 edition, 2015.
- [8] Monaco editor. <https://microsoft.github.io/monaco-editor>. Ημερομηνία πρόσβασης: 15-03-2022.
- [9] Simple-git. <https://www.npmjs.com/package/simple-git>. Ημερομηνία πρόσβασης: 10-07-2022.
- [10] G. Lim. *Beginning MERN Stack: Build and Deploy a Full Stack MongoDB, Express, React, Node.js App*. Independently, 1 edition, 2021.
- [11] Κώδικας Εφαρμογής. <https://github.com/nstavrinis/OnCode-Editor>.

-
- [12] Mongoose. <https://www.npmjs.com/package/mongoose>. Ημερομηνία πρόσβασης: 01-04-2022.
- [13] Bcrypt. <https://www.npmjs.com/package/bcrypt>. Ημερομηνία πρόσβασης: 15-04-2022.
- [14] Nodemailer. <https://nodemailer.com/about/>. Ημερομηνία πρόσβασης: 13-06-2022.
- [15] Jzip. <https://www.npmjs.com/package/jzip>. Ημερομηνία πρόσβασης: 25-06-2022.