

Πληροφοριακά Συστήματα Παγκόσμιου Ιστού

Τίτλος: FunC

VM: <http://83.212.77.175/>

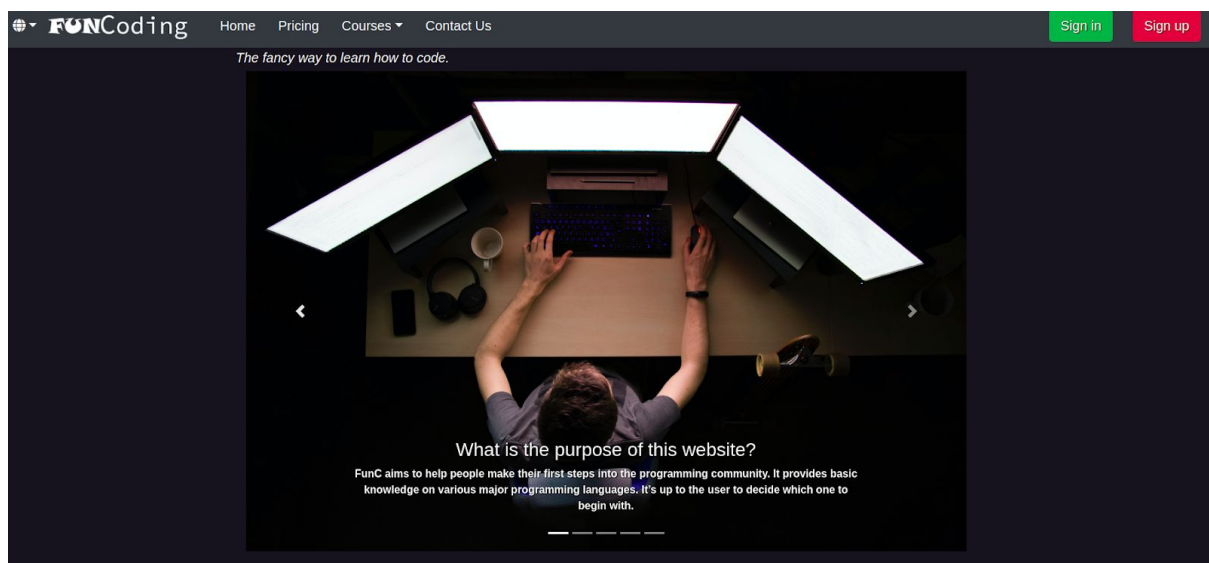
Admin Dashboard: <http://83.212.77.175/admin>

SDGs: GOAL 4- Quality Education

FUNCoding

Μέλη ομάδας

Εσκιόγλου Μαρία	3237
Κυνηγόπουλος Γεώργιος	3093
Λυσσούδη Πασχαλίνα	3116
Νίκας Μάριος	3065



Contents

1.Introduction.....	3
2.Fundamentals.....	4
3.Architecture	
1.Αρχική σελίδα.....	5
2.Μενού.....	5
3.Σύνδεση - Εγγραφή - Προφίλ Χρήστη.....	6
4.Ταμπλό διαχειριστή.....	7
5.Μαθήματα.....	8
4.Technologies	
1.Σύνδεση στην βάση δεδομένων.....	9
2.Εγγραφή/Σύνδεση Χρηστών.....	10
3.Εγγραφή/Σύνδεση Διαχειριστών.....	13
4.Φόρμα επικοινωνίας.....	14
5.Μαθήματα.....	15
6.Tests.....	16
5.Showcase-Presentation-Scenarios	
1.Main Page.....	16
2.Nav Menu.....	17
3.User Profile.....	18
4.Admin's Dashboard.....	19
5.Contact Page.....	20
6.Lesson Structure.....	21
7.Social Bar.....	22
8.Responsiveness for mobile screen.....	22
9.Virtual Machine.....	22
Bonus	
1.Search Bar.....	23
2.Edit/Add/Delete Lesson.....	23
3.React, Reactstrap, Bootstrap.....	24
4.MongoDB Express React Node.js.....	24
Copyrights.....	24
Comparisons and Experimentations.....	24
6.Future Ideas.....	27

1.Introduction

Το θέμα που επιλέξαμε να υλοποιήσουμε στο project είναι η δημιουργία ιστοσελίδας που θα έχει ως σκοπό την εκμάθηση γλωσσών προγραμματισμού. Το θέμα αυτό προέκυψε λαμβάνοντας υπόψη τους 17 Στόχους του Ο.Η.Ε. Από αυτούς αποφασίσαμε να επικεντρωθούμε στον 4ο, που πρεσβεύει την Ποιοτική Εκπαίδευση.

Όπως όλοι ξέρουμε, στην σύγχρονη πραγματικότητα ο προγραμματισμός είναι μια πολύ σημαντική δεξιότητα που μπορεί να αποκτήσει κάποιος για να βγει στην αγορά εργασίας. Εκεί ακριβώς προσπαθεί να βοηθήσει η ιστοσελίδα μας, διδάσκοντας βασικές γλώσσες προγραμματισμού σε οποιονδήποτε έχει ανάγκη. Η FunC απευθύνεται σε άτομα όλων των ηλικιών και αρκετών εθνικοτήτων, ανεξαρτήτως γεωγραφικού πλάτους, προσφέροντάς τους εξ'αποστάσεως εκπαίδευση.

Ένα από τα θετικά της σελίδας είναι ότι δεν προϋποθέτει την ύπαρξη κάποιου γνωστικού υποβάθρου, καθώς όλα τα μαθήματα ξεκινούν από πολύ βασικές ενότητες. Αυτό έχει ως αποτέλεσμα να απευθύνονται σε ένα ευρύ κοινό, που περιλαμβάνει άπειρους και έμπειρους χρήστες. Οι πρώτοι μπορούν να ξεκινήσουν όποια γλώσσα επιθυμούν από την αρχή, με αναλυτικές επεξηγήσεις και βοηθητικά παραδείγματα, και οι δεύτεροι να εξελίσουν τις ήδη υπάρχουσες γνώσεις τους εντρυφώντας σε πιο προχωρημένα και εξειδικευμένα μαθήματα.

Όλη η απαραίτητη γνώση είναι συγκεντρωμένη σε μια πλατφόρμα, συμπεριλαμβάνοντας εκτός από το θεωρητικό κομμάτι την δυνατότητα εξάσκησης της νέας γνώσης επιτόπου, χωρίς να χρειάζεται ανακατεύθυνση σε κάποιον άλλο ιστοχώρο.

Για τα μαθήματα δεν απαιτείται ιδιαίτερος εξοπλισμός, η ιστοσελίδα ανταποκρίνεται σε οποιαδήποτε συσκευή, με μοναδική προϋπόθεση την ύπαρξη ίντερνετ.

Όπως έχει προαναφερθεί, η γνώση οποιασδήποτε προγραμματιστικής γλώσσας αποτελεί προσόν για αυτόν που την κατέχει. Σε μία πολύ απαιτητική αγορά εργασίας, ο εμπλουτισμός του βιογραφικού με περιζήτητες γλώσσες δίνει σημαντικό πλεονέκτημα για την αντιμετώπιση του ανταγωνισμού. Η FunC δεν απευθύνεται μόνο σε προγραμματιστές, αλλά δίνει την δυνατότητα σε όλους να εξελίσουν τις γνώσεις τους και να διευκολύνουν την καθημερινότητα τους σε επαγγελματικό ή ερασιτεχνικό επίπεδο.

Τέλος, εκσυγχρονίζοντας τις απαιτήσεις που δημιουργούνται στην εξ' αποστάσεως εκπαίδευση με την εμφάνιση της πανδημίας COVID-19, η FunC αποτελεί εργαλείο που μπορεί να χρησιμοποιηθεί με πολλαπλούς τρόπους στην καταπολέμηση της υποεκπαίδευσης. Λειτουργεί ως βοηθημα στο οποίο μπορούν να παραπεμφθούν έπειτα από υπόδειξη των εκπαιδευτικών, μαθητές και φοιτητές. Παράλληλα σε ατομικό επίπεδο συνεισφέρει στην παραγωγική αξιοποίηση του ελεύθερου χρόνου, κάνοντας πιο αποδοτική την δύσκολη καθημερινότητα που βιώνει η παγκόσμια κοινότητα.

2.Fundamentals

Η FunC είναι μια ιστοσελίδα που απευθύνεται σε όλους. Άτομα όλων των ηλικιών, ανεξαρτήτως επαγγέλματος και γνωστικού υποβάθρου, μπορούν γίνουν χρήστες της σελίδας και να χρησιμοποιήσουν την γνώση που προσφέρεται για να εξελίξουν τις προγραμματιστικές τους ικανότητες.

Ωστόσο δεν είναι απαραίτητο να γίνει δημιουργία λογαριασμού για να έχει κάποιος πρόσβαση στα μαθήματα. Μπορεί απλώς να εισέλθει στην σελίδα, να επιλέξει την γλώσσα προγραμματισμού που θέλει και να ξεκινήσει, με ελαφρώς περιορισμένες δυνατότητες σε σχέση με τους χρήστες, όπως την δυνατότητα πρόσβασης σε τεστ προόδου στο τέλος πολλών μαθημάτων όλων των γλωσσών.

Το στήσιμο της ιστοσελίδας έχει ως επίκεντρο την εύκολη πρόσβαση στα μαθήματα και την ευχάριστη πλοήγηση μεταξύ αυτών. Έτσι, δεν χρειάζεται κάποια εμπειρία σε διαδικτυακά μαθήματα από την πλευρά του χρήστη. Η σελίδα τον κατευθύνει στα εισαγωγικά μαθήματα, και από εκεί μπορεί με την ίδια ευκολία να επιλέξει είτε να ξεκινήσει κάποια γλώσσα από την αρχή, είτε να μεταβεί σε κάποιο συγκεκριμένο μάθημα της γλώσσας μέσω του μενού μαθημάτων.

Οι θέσεις των διαχειριστών μοιράζονται εξίσου απλά, με την προϋπόθεση να έχει προηγηθεί επικοινωνία μεταξύ του ατόμου που αιτείται την θέση και των δημιουργών της σελίδας, για την εξακρίβωση των προγραμματιστικών ικανοτήτων του ατόμου. Πέρα από αυτό, η δημιουργία λογαριασμού διαχειριστή και η επεξεργασία των μαθημάτων μέσω του dashboard είναι μια τόσο απλοποιημένη διαδικασία, που μπορεί να την πραγματοποιήσει οποιοσδήποτε.

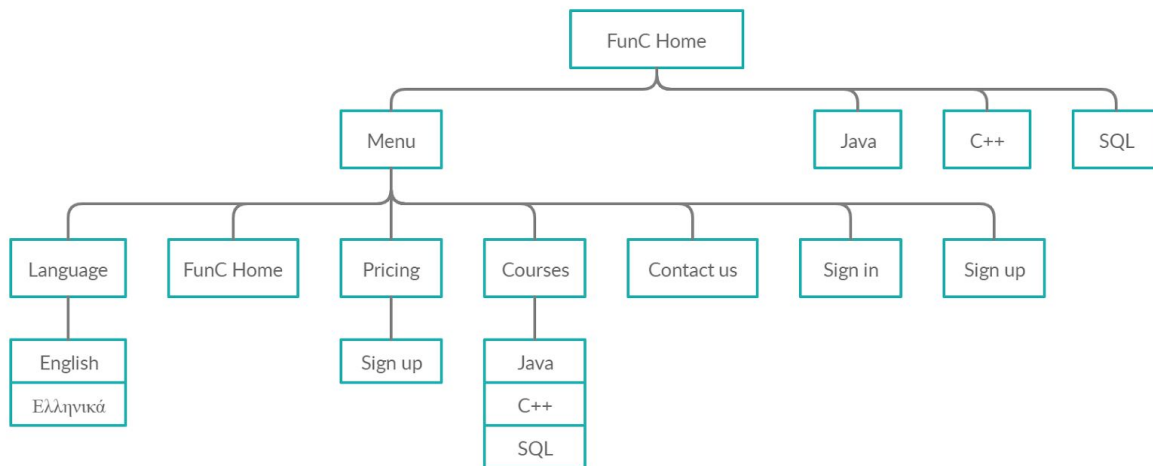
Σκοπός αυτής της σχεδίασης είναι η ανάδειξη της ευκολίας με την οποία μπορεί κάποιος να ξεκινήσει να ασχολείται με τον προγραμματισμό, και να φτάσει σε ένα αξιόλογο επίπεδο. Όλοι από κάπου ξεκινάνε, και η FunC επιθυμεί να προσφέρει μια αρκετά αποτελεσματική και διασκεδαστική αφετηρία.

Οι ίδιοι οι δημιουργοί της ιστοσελίδας, όντας προγραμματιστές, έφτασαν σε επίπεδο να μπορούν να συντάξουν κείμενα για την εκμάθηση γλωσσών προγραμματισμού με τις οποίες έχουν ασχοληθεί, σε μία απόπειρα να μεταφέρουν την γνώση τους σε όποιον επιθυμεί να την αποκτήσει. Φυσικά, τα κείμενα που έχουν χρησιμοποιηθεί έχουν εμπνευστεί από επίσημες ιστοσελίδες επιστημονικού περιεχομένου που προσφέρουν πληροφορίες για διάφορες γλώσσες προγραμματισμού, όπως οι :

www.cplusplus.com , www.tutorialspoint.com , www.programiz.com ,
www.w3schools.com , www.geeksforgeeks.org .

3.Architecture

Η συγκεκριμένη ενότητα περιγράφει κυρίως το στήσιμο της ιστοσελίδας σε επίπεδο front end, και αναλύει τον τρόπο και τον λόγο για τον οποίο έχει οργανωθεί με τον συγκεκριμένο λόγο η σελίδα.



Σχήμα 1: Αρχική Σελίδα

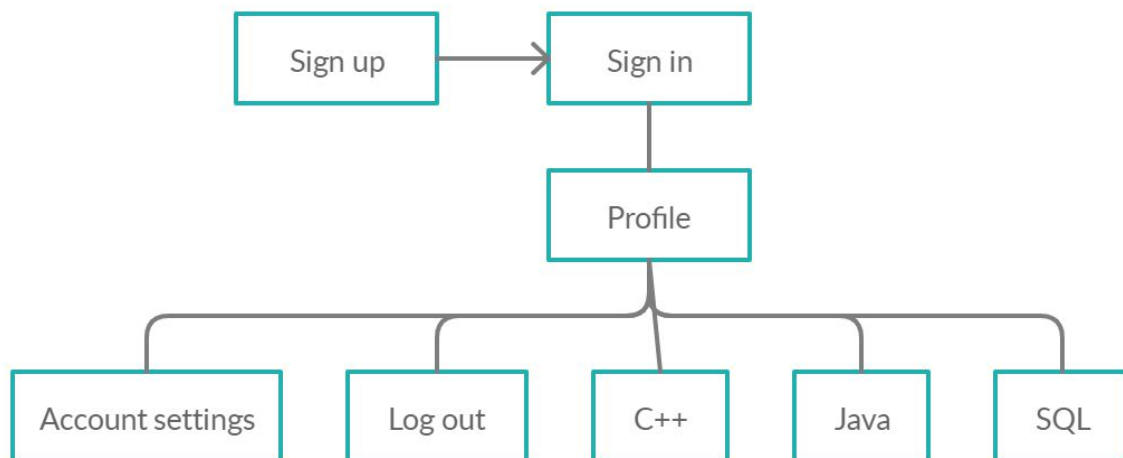
1.Αρχική Σελίδα:

Οι διασυνδέσεις που προσφέρονται από την Αρχική σελίδα, όπως φαίνεται στο Σχήμα 1, είναι μέσω της ενσωματωμένης μπάρας μενού που υπάρχει σε όλες τις σελίδες πλην του Ταμπλό, και επιπλέον τα μαθήματα σε Java, C++ και SQL. Τα μαθήματα αυτά φαίνονται κάτω από μερικές στατικές πληροφορίες για την σελίδα. Οι πολλαπλές διαδρομές που έχουμε ορίσει να κατευθύνουν τον χρήστη στα μαθήματα, προκύπτουν από το γεγονός ότι αυτά είναι ο λόγος ύπαρξης της ιστοσελίδας, οπότε θέλουμε να είναι εύκολη η πρόσβαση σε αυτά από πολλά και διαφορετικά σημεία της Αρχικής σελίδας.

2.Μενού:

Το Μενού βρίσκεται καρφιτσωμένο στο πάνω μέρος της σελίδας και δεν μεταβάλλεται αν ο χρήστης επιλέξει οποιαδήποτε από τα αντικείμενα του. Αυτό εξυπηρετεί την πλοήγηση του χρήστη αλλά και την λειτουργικότητα της ιστοσελίδας καθώς επιτρέπει την μετακίνηση σε όλες τις πιθανές μελλοντικές κινήσεις του χρήστη από ένα αμετάβλητο μενού. Παράλληλα, ο σχεδιασμός του το καθιστά εύκολα κατανοητό και δεν αποσπά τον χρήστη με άσκοπες πληροφορίες. Το Μενού περιέχει τις γλώσσες συστήματος, οι οποίες εμφανίζονται σε ένα drop-down menu και έχουν ως λογότυπο μία υδρόγειο σφαίρα, ένα εικονίδιο κατανοητό ανεξαρτήτως εθνικότητας. Επιπλέον, περιέχει πλοήγηση προς την Αρχική σελίδα, την Τιμολόγηση, τα Μαθήματα καθώς και τις φόρμες επικοινωνίας, σύνδεσης και εγγραφής. Τέλος,

αξίζει να σημειωθεί ότι κατά την είσοδο ενός χρήστη το περιεχόμενο του μενού αλλάζει και στα δεξιά αντί για σύνδεση ή εγγραφή εμφανίζεται η επιλογή της αποσύνδεσης και ένα επιπλέον εικονίδιο που οδηγεί στο προφίλ του χρήστη.

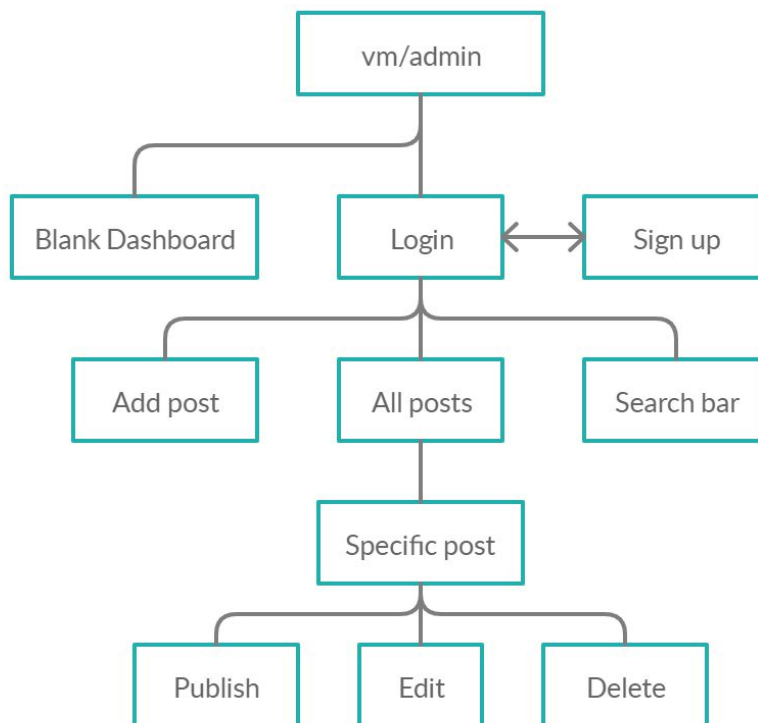


Σχήμα 2: Σύνδεση - Εγγραφή - Προφίλ Χρήστη

3.Σύνδεση - Εγγραφή - Προφίλ Χρήστη:

Η διαδικασία μετάβασης του χρήστη από την Αρχική σελίδα στο προφίλ του αναπαρίσταται στο Σχήμα 2. Αν έχει ήδη ενεργό λογαριασμό, θα συνδεθεί κατευθείαν από την σελίδα Sign in, εναλλακτικά θα κάνει πρώτα εγγραφή από το Sign up και θα ανακατευθυνθεί αυτόματα στην σελίδα Sign in για σύνδεση. Μετά την σύνδεση, κατευθύνεται αυτόματα στην σελίδα του προφίλ του, από όπου έχει πρόσβαση και στις Ρυθμίσεις Λογαριασμού που είναι απαραίτητη για την αλλαγή των στοιχείων του χρήστη σε περίπτωση λανθασμένης υποβολής. Η χρησιμότητα της, ωστόσο, δεν περιορίζεται εκεί καθώς περιέχει πληροφορίες για τον έλεγχο προόδου και τον αριθμό των ολοκληρωμένων test, καθώς και να μεταβεί απευθείας στα μαθήματα οποιασδήποτε γλώσσας επιθυμεί να συνεχίσει. Η σχεδίαση της έχει βασιστεί στο πρότυπο της υπόλοιπης ιστοσελίδας, ώστε να ενσωματώνεται πλήρως.

Ένας συνδεδεμένος χρήστης μπορεί να μεταβεί στην καρτέλα Pricing μέσω του μενού, και να επιλέξει αναβάθμιση του λογαριασμού του σε Premium, που αλλάζει κάποια στοιχεία στην εμφάνιση της σελίδας. Αν προσπαθήσει να κάνει αναβάθμιση του λογαριασμού χωρίς να έχει κάνει σύνδεση, η σελίδα του προτείνει με μήνυμα να κάνει πρώτα σύνδεση και να ξαναπροσπαθήσει.



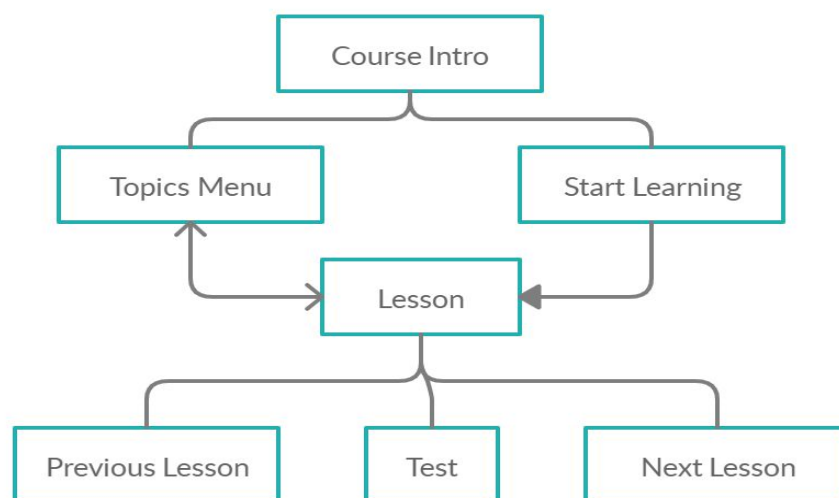
Σχήμα 3 : Ταμπλό Διαχειριστή

4. Ταμπλό Διαχειριστή:

Η σύνδεση για τους διαχειριστές έχει αποφασιστεί να είναι ανεξάρτητη από την σελίδα της FunC, για να μην μπορεί οποιοσδήποτε να αυτοπροσδιοριστεί διαχειριστής. Δεν υπάρχει, δηλαδή, κάποιο κουμπί που να οδηγεί από την αρχική σελίδα στο ταμπλό των διαχειριστών.

Ένας διαχειριστής θα πρέπει να γνωρίσει να μπει στον σύνδεσμο `vm/admin` (αναγράφεται και στην αρχή της εργασίας) ώστε να έχει πρόσβαση σε ένα κενό μέχρι να κάνει σύνδεση ταμπλό, και στο κουμπί για σύνδεση. Σε περίπτωση που δεν έχει ενεργό λογαριασμό, από την Σύνδεση μπορεί να μεταβεί στην Εγγραφή, και πάλι πίσω στην Σύνδεση. Μόλις εισάγει έγκυρα στοιχεία σύνδεσης, το ταμπλό θα γεμίσει με όλα τα `posts` που έχουν γίνει από όλους τους διαχειριστές. Έχει, επίσης, την δυνατότητα να δημιουργήσει ο ίδιος ένα ποστ με το κουμπί (+) πάνω αριστερά. Ένα ποστ που έχει δημιουργηθεί, δεν ανεβαίνει αυτόματα σαν μάθημα στην σελίδα της FunC. Πρέπει πρώτα κάποιος διαχειριστής να ανοίξει το συγκεκριμένο ποστ, και να πατήσει Δημοσίευση στο κάτω μέρος του ποστ. Εναλλακτικά, μπορεί να το ενημερώσει ή να το διαγράψει με αντίστοιχα κουμπιά.

Τέλος, έχει υλοποιηθεί δυνατότητα αναζήτησης μεταξύ των ποστ, με μπάρα αναζήτησης που βρίσκεται πάνω δεξιά στο ταμπλό των διαχειριστών. Η αναζήτηση γίνεται κατευθείαν, με την εισαγωγή ενός γράμματος στην μπάρα, και απορά γλώσσες προγραμματισμού και τους τίτλους των μαθημάτων.



Σχήμα 4 : Μαθήματα

5.Μαθήματα:

Η πρόσβαση στα μαθήματα θα επιτυγχάνεται με πολλούς τρόπους μέσα στην ιστοσελίδα, καθώς, όπως έχει αναφερθεί, είναι το πιο βασικό περιεχόμενό της. Για παράδειγμα, κάποιος μπορεί να μεταβεί σε μάθημα οποιασδήποτε γλώσσας από το μενού, από την αρχική σελίδα, από το προφίλ χρήστη μετά την σύνδεση, από το ταμπλό των διαχειριστών. Με την επιλογή μιας γλώσσας προγραμματισμού, προκαλείται ανακατεύθυνση στο εισαγωγικό κομμάτι της γλώσσας αυτής. Από εκεί υπάρχει δυνατότητα επιλογής ενός συγκεκριμένου μαθήματος από πλάγιο μενού που περιέχει τους τίτλους όλων των μαθημάτων της γλώσσας, ή μετακίνηση στο πρώτο μάθημα της γλώσσας με το κουμπί Ξεκινήστε τα μαθήματα. Σε όποιο μάθημα κι αν πραγματοποιηθεί μετάβαση, υπάρχει πάντα το πλάγιο μενού με τους τίτλους, και ορισμένα κουμπιά πλοήγησης σε άλλα μαθήματα. Αν το τρέχον μάθημα δεν είναι το πρώτο της γλώσσας, υπάρχει κουμπί Προηγούμενο μάθημα. Αν το τρέχον μάθημα δεν είναι το τελευταίο της γλώσσας, υπάρχει κουμπί Επόμενο μάθημα. Αν υπάρχει κάποιο τεστ που να εξετάσει τις γνώσεις που αποκτήθηκαν από το τρέχον μάθημα, υπάρχει κουμπί Κάνε το τεστ.

Η σειρά των μαθημάτων και της ύπαρξης ή όχι τεστ καθορίζεται ρητά από τους διαχειριστές.

4. Technologies

Το συγκεκριμένο κομμάτι της αναφοράς απευθύνεται κυρίως στο back end, για το οποίο έχουν χρησιμοποιηθεί οι περισσότερες τεχνολογίες.

Η ανάπτυξη της ιστοσελίδας βασίστηκε στην χρήση της τεχνολογίας MERN.

Το πρώτο μέρος της εργασίας (front end) είχε υλοποιηθεί σε React, η οποία βασίζεται στην Javascript με πιο εμπλουτισμένες δυνατότητες. Η μορφοποίηση έγινε με CSS και SASS, και υπήρχαν ελάχιστα κομμάτια HTML. Πέρα από αυτά, χρησιμοποιήθηκαν βιβλιοθήκες όπως η bootstrap, που προσέφεραν περισσότερες επιλογές για την μορφοποίηση, και την δυνατότητα προσαρμογής της ιστοσελίδας σε διάφορους τύπους και μεγέθη οθόνης.

Μετά από ψάξιμο καταλήξαμε στην MongoDB ως καλύτερη λύση για την βάση δεδομένων, καθώς το περιεχόμενο της σελίδας μας αποτελείται από documents. Η σύνδεση μεταξύ τους γίνεται με το NodeJS και την Express.

1. Σύνδεση στην βάση δεδομένων

```
api
| config
| models
| routes
| controllers
| executables
| middleware
| uservalidation
| validation
.env
server.js
```

Η δομή του φακέλου api αναγράφεται στο παραπάνω σχήμα και συγκεκριμένα η σύνδεση με την βάση δεδομένων γίνεται στο server.js. Το αρχείο .env περιλαμβάνει το mongo uri για την ένωση με το cluster από το mongo atlas και το secret.

Ο server.js πραγματοποιεί την σύνδεση με την βάση δεδομένων, χωρίς να είναι ορατά τα στοιχεία του cluster σε πρώτη ανάγνωση και υπάρχουν μηνύματα επιβεβαίωσης ή λάθους, τα οποία εμφανίζονται στο console log ώστε να γίνει κατανοητό αν η σύνδεση με την βάση δεδομένων ήταν επιτυχής. Επίσης, εμφανίζεται και το port στο οποίο τρέχει ο server. Τέλος, στον server υπάρχουν και τα routes για τις διάφορες λειτουργίες που προσφέρονται στην ιστοσελίδα.

Πρέπει να αναφερθεί κάπου εδώ ότι η διαδικασία σύνδεσης της βάσης δεδομένων που υπάρχει στο mongoDB ήταν αρκετά επίπονη. Όλα τα tutorials που υπήρχαν στο διαδίκτυο είναι προηγούμενων ετών, και το string του mongo uri έχει πλέον διαφορετική μορφοποίηση από ότι είχε παλαιότερα. Αυτό είχε σαν αποτέλεσμα να μην το αναγνωρίζει το πρόγραμμα. Για μικρό διάστημα πραγματοποιούνταν τοπική σύνδεση ανά υπολογιστή με την βάση δεδομένων, με την χρήση του Mongo Compass. Η λύση φάνηκε μετά από πολλές μέρες έρευνας, όπου σε σχόλια μιας δημοσίευσης σε ιστοσελίδα για βοήθεια προγραμματιστών ανακαλύψαμε ότι έπρεπε να ενημερώσουμε τις εκδόσεις κάποιων βιβλιοθηκών όπως η mongo και η mongoose σε συγκεκριμένες εκδόσεις, όπως και την έκδοση που χρησιμοποιούμε στο Node.js . Μόνο τότε το πρόγραμμα αναγνώρισε την μορφή του uri και επιτεύχθηκε σύνδεση με την βάση δεδομένων.

Γενικά, για την υλοποίηση των διαφόρων λειτουργιών της ιστοσελίδας έχουν χρησιμοποιηθεί πολλές βιβλιοθήκες για την υποστήριξη του backend. Μία από τις πιο σημαντικές είναι η bcrypt, ώστε οι κωδικοί που βάζουν οι χρήστες να γίνονται hash πριν αποθηκευτούν στην βάση δεδομένων. Έτσι, όσοι έχουν πρόσβαση στο mongo atlas δεν θα μπορούν να δουν τους προσωπικούς κωδικούς των χρηστών και ούτε να υποκλέψουν λογαριασμούς. Με αυτόν τον τρόπο επιτυγχάνεται η ασφάλεια των στοιχείων των λογαριασμών.

Επίσης, χρησιμοποιήθηκε η βιβλιοθήκη concurrently ώστε να μπορούμε να τρέξουμε παράλληλα το frontend και το backend, η βιβλιοθήκη mongoose για την αλληλεπίδραση με την mongoDB και η validator για την επιβεβαίωση διαφόρων inputs, όπως αν δύο κωδικοί ταιριάζουν μεταξύ τους, αν τα email έχουν την σωστή μορφή, κτλ.

2. Εγγραφή/Σύνδεση Χρηστών

```
api
| config
  passport.js
| models
  UserModel.js
| routes
  |api
    usermodel.js
  | uservalidation
    login.js
    register.js
```

Το σχήμα για τους χρήστες βρίσκεται στον φάκελο models και αποτελείται από το username, το email και τον κωδικό, τα οποία είναι όλα σε string και υποχρεωτικά. Επίσης αποθηκεύονται η ημερομηνία εγγραφής, μια Boolean μεταβλητή που δείχνει αν ο λογαριασμός είναι premium και ένα Object όπου κρατά τα ονόματα των τεστ που έχουν λυθεί. Κάνουμε export το model για να μπορούμε να το χρησιμοποιήσουμε και εκτός φακέλου.

Στην συνέχεια στο φάκελο `uservalidation`, υπάρχουν τα αρχεία για την εγγραφή και την σύνδεση των χρηστών. Ξεκινώντας από την εγγραφή, μετατρέπουμε όλα τα πεδία σε άδεια strings γιατί η validator λειτουργεί μόνο σε strings. Οι έλεγχοι που γίνονται είναι για άδεια πεδία, σωστή μορφή του email, επιβεβαίωση ότι οι δύο κωδικοί είναι ίδιοι μεταξύ τους και αν οι απαιτήσεις για τον κωδικό ικανοποιούνται. Ενδεικτικά, βάλαμε ότι το ελάχιστο μήκος κωδικού πρέπει να είναι 6 και το μέγιστο 30.

Σε παρόμοια λογική έχει φτιαχτεί και το login απλά με διαφορετικά πεδία. Εδώ μας ενδιαφέρει μόνο το email και ο κωδικός και οι έλεγχοι πάνω σε αυτά τα δύο στοιχεία.

Και στα δύο αρχεία επιστρέφουμε τα errors που έχουν εντοπιστεί και με μία μεταβλητή τύπου boolean ελέγχουμε για το αν έχουμε καθόλου errors.

Στο φάκελο routes και μέσα στο `arī` έχουν προστεθεί τα validations μαζί με το σχήμα και δημιουργήθηκαν τα endpoints για την εγγραφή και την σύνδεση. Η διαδικασία για το register endpoint είναι αρχικά να γίνει ο έλεγχος στην βάση αν υπάρχει ήδη ο χρήστης με την συνάρτηση `User.findOne()`. Αν υπάρχει επιστρέφει μήνυμα λάθους, αλλιώς τα πεδία name, email και password θα πάρουν τις τιμές που έδωσε ο χρήστης.

Παρόμοια είναι και η διαδικασία για το login endpoint, όπου ελέγχουμε αν ο χρήστης υπάρχει πρώτα. Αν υπάρχει, χρησιμοποιούμε την `bcrpyt` για να ελεγχουμε αν ο κωδικός που εισήγαγε ο χρήστης είναι ο ίδιος με τον καταχωρημένο hashed κωδικό στην βάση δεδομένων για τον συγκεκριμένο λογαριασμό. Αν οι κωδικοί ταιριάζουν τότε δημιουργείται το JWT Payload και το υπογράφουμε με το secret που έχουμε θέσει στο `keys.js`. Αν είναι επιτυχής η σύνδεση τότε επιστρέφει success και το Bearer, αλλιώς μήνυμα λάθους.

POST localhost:5000/api/usermodel/register

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies Code

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE	DESCRIPTION
email	3237@gmail.com	
password	123456	
password2	123456	
name	Maria	
Key	Value	Description

Body Cookies Headers (6) Test Results Status: 200 OK Time: 1193 ms Size: 403 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": "5ef075670d31311de45b08ae",
3   "name": "Maria",
4   "email": "3237@gmail.com",
5   "password": "$2a$10$1K1truP8kQ3rwTFPcqqhXeJ0v/xIaMs554uywtvcVhh66T8Me1Hba",
6   "date": "2020-06-22T09:09:59.022Z",
7   "_v": 0
8 }
```

POST localhost:5000/api/usermodel/login Send Save

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies Code

☐ none ☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> email	3237@gmail.com			
<input checked="" type="checkbox"/> password	123456			
<input type="checkbox"/> password2	123456			
<input type="checkbox"/> name	Maria			
Key	Value	Description		

Body Cookies Headers (6) Test Results Status: 200 OK Time: 371 ms Size: 438 B Save Response

Pretty Raw Preview Visualize JSON 🔍

```

1 {
2   "success": true,
3   "token": "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjVlZjA3NTY3MGQzMGMxMWRlNDViMDhhZSIsIm5hbWUiOiJNYXJpYSIsImh0dCI6MTU5Mjg4ZjY3bG9hZG93dGVzIiwiaWF0IjE0MzcwOTgxfQ.dJbA6Bxy9Q1CNx84eACLGQSTRweYE3Jb6gRHqGLLdvE"
4 }
```

3. Εγγραφή/Σύνδεση Διαχειριστών

```

| api
| middleware
|   passport.js
| models
|   Post.js
|   User.js
| routes
|   | api
|   |   post.js
|   |   user.js
| validation
|   login.js
|   signup.js
|   post.js
| .env
| server.js

```

Στο πρώτο μέρος της εργασίας, το dashboard αποτελούνταν από ένα έτοιμο template, το οποίο περιείχε πολλές περιττές πληροφορίες σε σχέση με αυτές που χρειαζόμασταν οπότε ξεκινήσαμε την υλοποίηση του εκ νέου και αυτή την φορά ξεκίνησε από το back και μεταφέρθηκε στο front.

Μία από τις αρχικές και απαραίτητες λειτουργίες είναι η σύνδεση και εγγραφή των διαχειριστών. Το σχήμα τους απαρτίζεται από το username, email και password τα οποία είναι σε μορφή string και υποχρεωτικά. Οι απαραίτητοι έλεγχοι που πρέπει να εφαρμοστούν στις δύο φόρμες, βρίσκονται στον φάκελο validation και είναι αντίστοιχοι με αυτούς που χρησιμοποιήθηκαν για την σύνδεση και εγγραφή των απλών χρηστών και ελέγχουν την εγκυρότητα των πεδίων.

Αντίστοιχα, έχει υλοποιηθεί και το σχήμα του post με υποχρεωτικά πεδία title, lesson, language, author που είναι τύπου String και μία μεταβλητή τύπου Boolean, την hasTest ενώ υπάρχουν και τα προαιρετικά στοιχεία που είναι τα test, initialCode και answer, όλα String. Οι απαραίτητοι έλεγχοι που πρέπει να εφαρμοστούν στα posts, βρίσκονται στο φάκελο validation και ελέγχουν την εγκυρότητα των υποχρεωτικών πεδίων καθώς και αυτών που επιλέχθηκαν.

Στο φάκελο routes και μέσα στο api έχουν προστεθεί τα validations μαζί με το σχήμα και δημιουργήθηκαν τα endpoints για την εγγραφή και την σύνδεση. Η διαδικασία για το signup endpoint είναι παρόμοια με την αντίστοιχη των χρηστών, δηλαδή γίνεται πρώτα η ταυτοποίηση για σωστή εισαγωγή των πεδίων. Αν είναι λάθος εμφανίζει μηνύματα λάθους αλλιώς ψάχνει στην βάση δεδομένων για διαχειριστές με τα ίδια στοιχεία. Αν υπάρχουν, εμφανίζει μήνυμα λάθους ενώ αν δεν υπάρχουν δημιουργεί τους διαχειριστές, χρησιμοποιεί συνάρτηση κατακερματισμού για τους κωδικούς και πραγματοποιείται η αποθήκευση στην βάση δεδομένων.

Σε παρόμοια λογική έχει δημιουργηθεί και το login endpoint στο οποίο μετά την εφαρμογή των validations ελέγχουμε αν ο διαχειριστής υπάρχει. Αν υπάρχει, χρησιμοποιούμε την bcrypt για να συγκρίνουμε τον κωδικό που έχει δώσει με αυτόν που είναι αποθηκευμένο στην βάση δεδομένων. Αν είναι διαφορετικοί επιστρέφουμε μήνυμα λάθους αλλιώς δημιουργείται ένα payload, το υπογράφουμε με το secret και στέλνουμε την απάντηση στον διαχειριστή.

Στο φάκελο routes και μέσα στο api υπάρχει το αρχείο post.js το οποίο διαχειρίζεται όλα τις λειτουργίες που επιτελούν τα posts. Υπάρχουν λειτουργίες για την εύρεση posts με get requests ανά author και id, μία post για την δημιουργία, patch και delete για την ενημέρωση και διαγραφή αντίστοιχα.

POST localhost:5000/api/users/signup

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies Code

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE	DESCRIPTION
email	3237@gmail.com	
password	123456	
username	Maria	
Key	Value	Description

Body Cookies Headers (6) Test Results Status: 200 OK Time: 861 ms Size: 373 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "_id": "5ef22dd60c006c285370c2cc",
3   "username": "Maria",
4   "email": "3237@gmail.com",
5   "password": "$2a$10$qpJzv3uJ2tVzkmzzkpGQV0DS1L1LcQFBZ3lVkyBKmmLTnIPtHZmjC",
6   "__v": 0
7 }
```

POST localhost:5000/api/users/login

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies Code

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> email	3237@gmail.com			
<input checked="" type="checkbox"/> password	123456			
<input type="checkbox"/> username	Maria			
Key	Value	Description		

Body Cookies Headers (6) Test Results Status: 200 OK Time: 399 ms Size: 443 B Save Response

Pretty Raw Preview Visualize JSON

```

1 {
2   "success": true,
3   "token": "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjVlZjIyZGQ2MGZjMjg1MzZjYyYyIsInVzZXJuYV11IjoiaWFyaWEiLCJpYXQiOi0jE10TI5Mjk4MjEsImV4cCI6MTU5Mjk2NTgyMjE0Lm0qO9R0AKMNdVXbCbyIE8Wt0LDPSIP8F6Chd0RNmyfc"
4 }

```

Στην συνέχεια θέτουμε το authorization token ίσο με αυτό που δόθηκε ώστε να έχουμε πρόσβαση στις λειτουργίες του post.

4. Φόρμα Επικοινωνίας

(a) *Πρώτη Υλοποίηση*

```

api
| controllers
|   Contact.js
| models
|   contact.js
| routes
|   contact.js

```

Για την αποστολή των μηνυμάτων μέσω της φόρμας επικοινωνίας υπάρχουν δύο εναλλακτικές υλοποιήσεις στο πρόγραμμα. Η πρώτη δουλεύει μόνο στο backend οπότε δεν μπορεί να εξεταστεί πρακτικά, αλλά τρέχει σωστά μέσω PostMan ή αντίστοιχων λογισμικών. Το σχήμα του contact που βρίσκεται στον φάκελο models απαρτίζεται από 4 στοιχεία, δηλαδή το όνομα, το επίθετο, το email και το μήνυμα που επιθυμεί να στείλει ο αποστολέας. Καθένα από αυτά είναι υποχρεωτικό και στον φάκελο Controllers γίνεται ο αντίστοιχος έλεγχος και η εμφάνιση των μηνυμάτων λάθους, ανάλογα με το αν λείπει ή όχι κάποιο από τα υποχρεωτικά αυτά στοιχεία και επιστρέφει τον αντίστοιχο κωδικό status. Συγκεκριμένα υπάρχουν έλεγχοι για το αν κάποιο από τα πεδία δεν είναι συμπληρωμένο αλλά και εξετάζει αν το email έχει την επιτρεπτή μορφή.

POST localhost:5000/api/contact/submit

Send Save

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies Code

none form-data **x-www-form-urlencoded** raw binary GraphQL

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> firstName	Maria			
<input checked="" type="checkbox"/> email	78@gmail.com			
<input checked="" type="checkbox"/> message	This is a message.			
<input checked="" type="checkbox"/> lastName	Eskioglou			
Key	Value	Description		

Body Cookies Headers (4) Test Results Status: 201 Created Time: 385 ms Size: 127 B Save Response

Όπως φαίνεται και στην εικόνα, το μήνυμα έχει δημιουργηθεί και αποθηκεύεται στο collection “contact forms”. Το μειονέκτημα αυτής της υλοποίησης είναι ότι δεν το καταφέραμε να στέλνεται το μήνυμα και στο email μας.

(b) Δεύτερη Υλοποίηση

Η δεύτερη υλοποίηση πραγματοποιήθηκε μέσω του formspree.io, όπου δημιουργήσαμε έναν λογαριασμό και αντιγράψαμε τον κώδικα μας για την φόρμα στο αρχείο myform.js. Έτσι, κάθε φορά που γίνεται μια υποβολή, έρχεται ειδοποίηση στο email μας με τα στοιχεία του αποστολέα και το μήνυμα.

5. Μαθήματα

api

| models

Lesson.js

| routes

lessonTest.js

Το σχήμα για τα μαθήματα βρίσκεται στον φάκελο models και περιέχει αναγκαίες τιμές τίτλο (ελληνικά και αγγλικά), κείμενο σε string όπου περιέχει html tags για ομορφότερη εμφάνιση (ελληνικά και αγγλικά), γλώσσα προγραμματισμού, boolean που ισχύει όταν το μάθημα έχει αντίστοιχο τεστ και τέλος ένα κλειδί που υποδηλώνει σειρά σχετικά με τα υπόλοιπα μαθήματα της συγκεκριμένης γλώσσας. Στο lessonTest υπάρχουν διάφορες λειτουργίες για ανάκτηση, δημιουργία, ενημέρωση και διαγραφή. Ειδικότερα υπάρχει post request που δέχεται όλα τα στοιχεία του μοντέλου Lesson και δημιουργεί ένα μάθημα, patch request ή delete request τα οποία γίνονται δίνοντας είτε το id του μαθήματος, είτε τον τίτλο του με την γλώσσα προγραμματισμού του υπάρχοντος μαθήματος και ύστερα ενημερώνονται/διαγράφονται. Υπάρχουν ακόμα 4 get request. Όταν δεν δίνεται τίποτα, επιστρέφονται όλα, όταν δίνεται γλώσσα και τίτλος επιστρέφονται όλες οι πληροφορίες εκείνου του μαθήματος ή το url του επόμενου μαθήματος (ξεχωρίζεται ανάλογα με την κλήση του) και τέλος εάν δοθεί μόνο γλώσσα τότε επιστρέφεται η λίστα με όλοι οι τίτλοι μαθημάτων αυτής της γλώσσας.

6. Tests

api

| **executables**

- cpp.bat
- java.bat
- Ubuntucpp.bat
- Ubuntujava.bat

| **models**

- Test.js

| **routes**

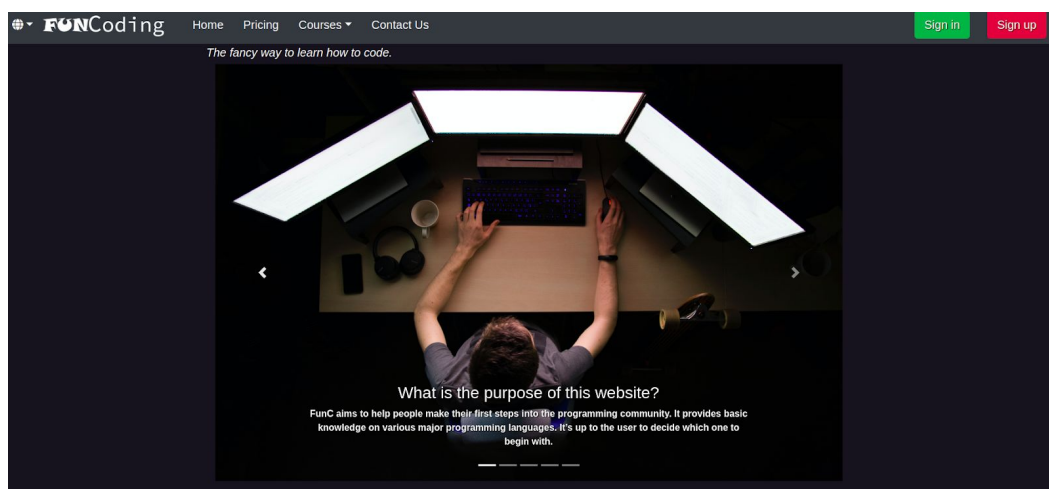
- test.js
- compileCode.js

Το σχήμα για τα τεστ βρίσκεται στον φάκελο models και παρέχει απαραίτητα πεδία όπως γλώσσα και τίτλος για την αντιστοίχιση του με το ανάλογο μάθημα και ύστερα τις πληροφορίες του τεστ όπως μια μικρή σύνοψη του μαθήματος, την εκφώνηση του τεστ, ένας αρχικός κώδικας για να συνεχίσει ο χρήστης και τέλος την σωστή απάντηση όπου είναι case sensitive. Υπάρχουν οι κλασικές λειτουργίες με requests όπως δημιουργία (post), ανάκτηση (get), ενημέρωση (patch) και διαγραφή (delete) τεστ. Το compileCode.js έχει 2 post request οι οποίες δέχονται ένα string με τον κώδικα και επιστρέφουν το αποτέλεσμα του compile. Η μια είναι για την c++ και η άλλη για την Java. Πιο συγκεκριμένα η nodejs ξεκινά μια καινούρια διαδικασία και εκτελεί εντολές που βρίσκονται στο cpp.bat για την C++ και java.bat για την Java (με λειτουργικό τα Windows). Επιλέγονται τα αντίστοιχα Ubuntu bat αν το λειτουργικό σύστημα είναι Linux (χρειάστηκαν για το VM). Τέλος στέλνεται πίσω ένα string όπου έχει τον αριθμό 0 μπροστά σε περίπτωση επιτυχημένου compile. Αλλιώς, έχει το 1.

5.Showcase-Presentation-Scenarios

Η δημιουργία της ιστοσελίδας FunC στηρήχθηκε, αρχικά, στα παρακάτω δομικά στοιχεία:

1.Main Page



Η αρχική σελίδα είναι αυτή που εμφανίζεται όταν κάποιος επισκέπτης μπει για πρώτη φορά στην FunC. Από πάνω προς τα κάτω, εντοπίζει κανείς μία οριζόντια μπάρα μενού. Από κάτω βρίσκεται το λογότυπο της σελίδας και μια χαρακτηριστική φράση. και ακολούθως μια δομή της Bootstrap εν ονόματι “Carousel”, που περιλαμβάνει εικόνες και κάποια εισαγωγικά λόγια για την ιστοσελίδα μας. Η αλλαγή των διαφανειών του Carousel γίνεται αυτόματα, μπορεί ωστόσο να γίνει και χειροκίνητη μετακίνηση. Ο χρήστης συναντάει, μετέπειτα, λογότυπα των γλωσσών που προσφέρονται με μια έξυπνη περιγραφή και κουμπί ανακατεύθυνσης στην καθεμία ξεχωριστά. Στο κάτω μέρος της σελίδας υπάρχουν εικονίδια για τα κοινωνικά μέσα της σελίδας, και από κάτω αναφέρονται τα δικαιώματα ολόκληρης της ιστοσελίδας.

- Στο δεύτερο μέρος της εργασίας, προστέθηκαν, λογαριασμοί για τα κοινωνικά δίκτυα, περισσότερες μεταφράσεις και οι καρτέλες με τα μαθήματα απέκτησαν λειτουργικότητα.

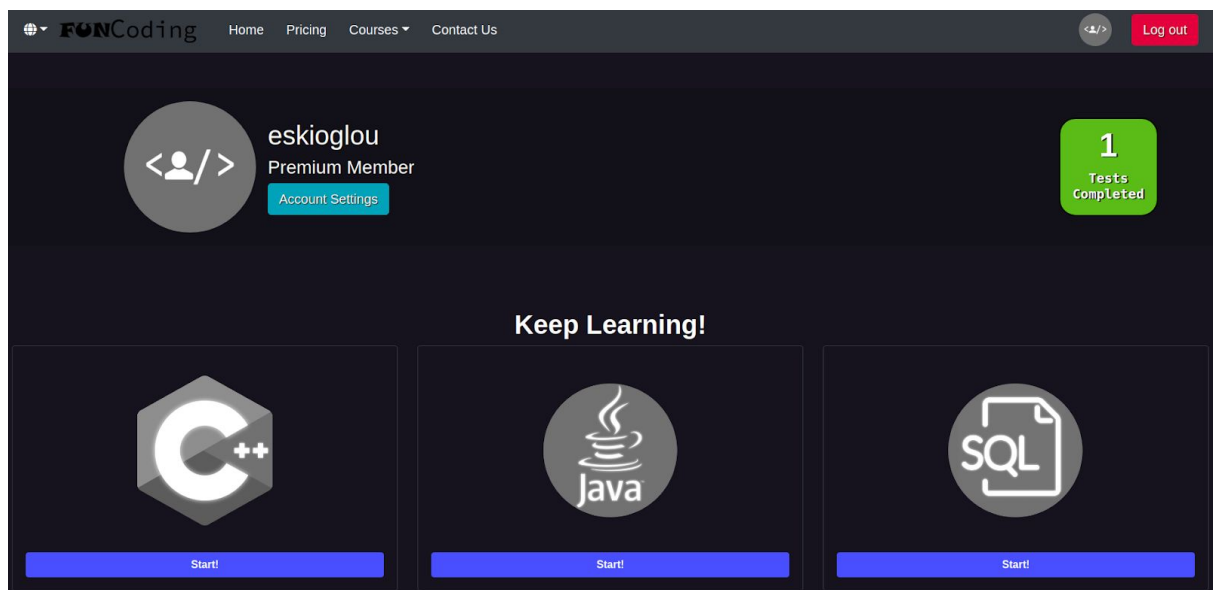
2.Navigation menu



Το μενού πλοήγησης όπως προαναφέρθηκε είναι μια οριζόντια μπάρα που βρίσκεται στην αρχική και όλες τις άλλες σελίδες εκτός από το dashboard του διαχειριστή, και μένει σταθερό και ανεξάρτητο κύλισης της ιστοσελίδας. Δίνει την δυνατότητα αλλαγής γλώσσας από την υδρόγειο σφαίρα που βρίσκεται στην αριστερή άκρη, και προσφέρει την δυνατότητα ανακατεύθυνσης μεταξύ των σελίδων “Αρχική”, “Τιμολόγηση”, “Dashboard” (βρίσκεται εδώ για την παρουσίαση, στο δεύτερο μέρος θα ανακατευθύνονται εκεί οι διαχειριστές μετά την είσοδο στο σύστημα), “Courses” με λίστα όλων των διαθέσιμων μαθημάτων και “Επικοινωνία”. Στην δεξιά άκρη έχει κουμπιά “Σύνδεση” και “Εγγραφή”, που οδηγούν στις σελίδες για τις αντίστοιχες λειτουργίες.

- Οι αλλαγές που πραγματοποιήθηκαν στο δεύτερο μέρος της εργασίας σχετικά με το κομμάτι του μενού πλοήγησης είναι ότι αφαιρέθηκε το Dashboard και πλέον η πρόσβαση γίνεται μέσω του /admin ώστε να μην είναι δημόσιο και μπορεί ο κάθε χρήστης να φτιάξει λογαριασμό. Επίσης, προστέθηκε λειτουργικότητα στην “Σύνδεση” και “Εγγραφή”. Τα “Courses” έχουν αποκτήσει περιεχόμενο και tests τα οποία έχουν συνδεθεί με την βάση δεδομένων και μπορούν είτε να εισαχθούν κατευθείαν από εκεί είτε μέσω του “Dashboard”. Τέλος, η “Επικοινωνία” έχει αποκτήσει λειτουργικότητα και στέλνει τα μηνύματα σε ένα λογαριασμό email.

3. User Profile



Account Settings

Change Username

Change Email

Change Password

Delete Your Account

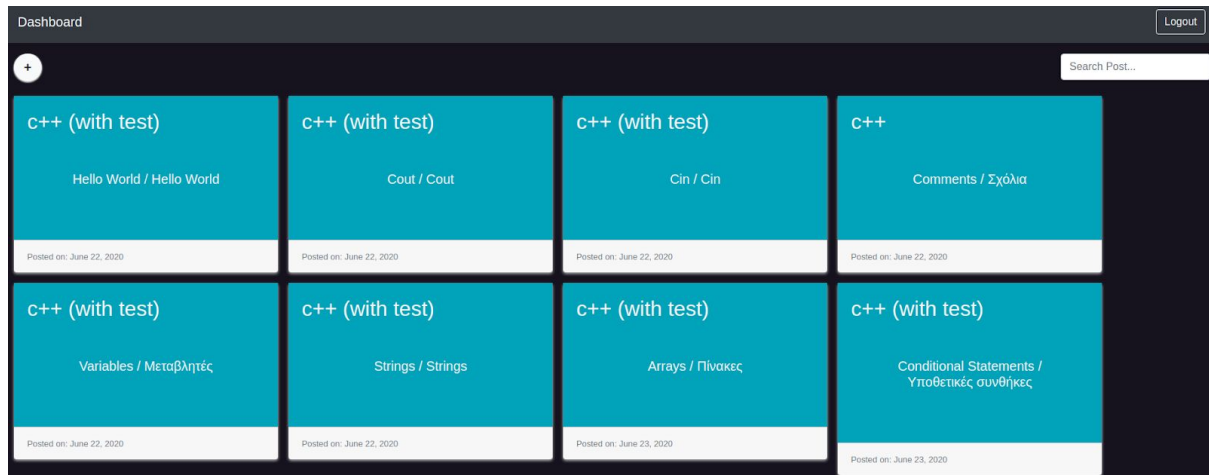
You can't undo this option!

Αφού ο χρήστης επιλέξει “Σύνδεση” και δώσει τα κατάλληλα στοιχεία ώστε να επιβεβαιωθούν με την βάση δεδομένων, θα ανακατευθύνεται στο προφίλ του. Εκεί, εμφανίζεται η εικόνα που έχει επιλέξει, το όνομα, η ιδιότητα του ως “Free Member” ή “Premium Member” και στα δεξιά το σύνολο των test που έχει ολοκληρώσει. Στο

κάτω μέρος της σελίδας εμφανίζονται υπό την μορφή καρτέλας τα μαθήματα που έχει δηλώσει, και ένα λινκ ανακατεύθυνσης για να συνεχίσει.

- Στο δεύτερο μέρος της εργασίας και μετά την επιτυχή σύνδεση, αλλάζει και η μπάρα πλοήγησης και αντί για Εγγραφή/Σύνδεση εμφανίζεται το κουμπί Αποσύνδεσης. Επίσης, στο προφίλ του Χρήστη υπάρχει και το κουμπί με τις Ρυθμίσεις στο οποίο μπορεί ο Χρήστης να αλλάξει τα στοιχεία του ή να διαγράψει τον λογαριασμό του.

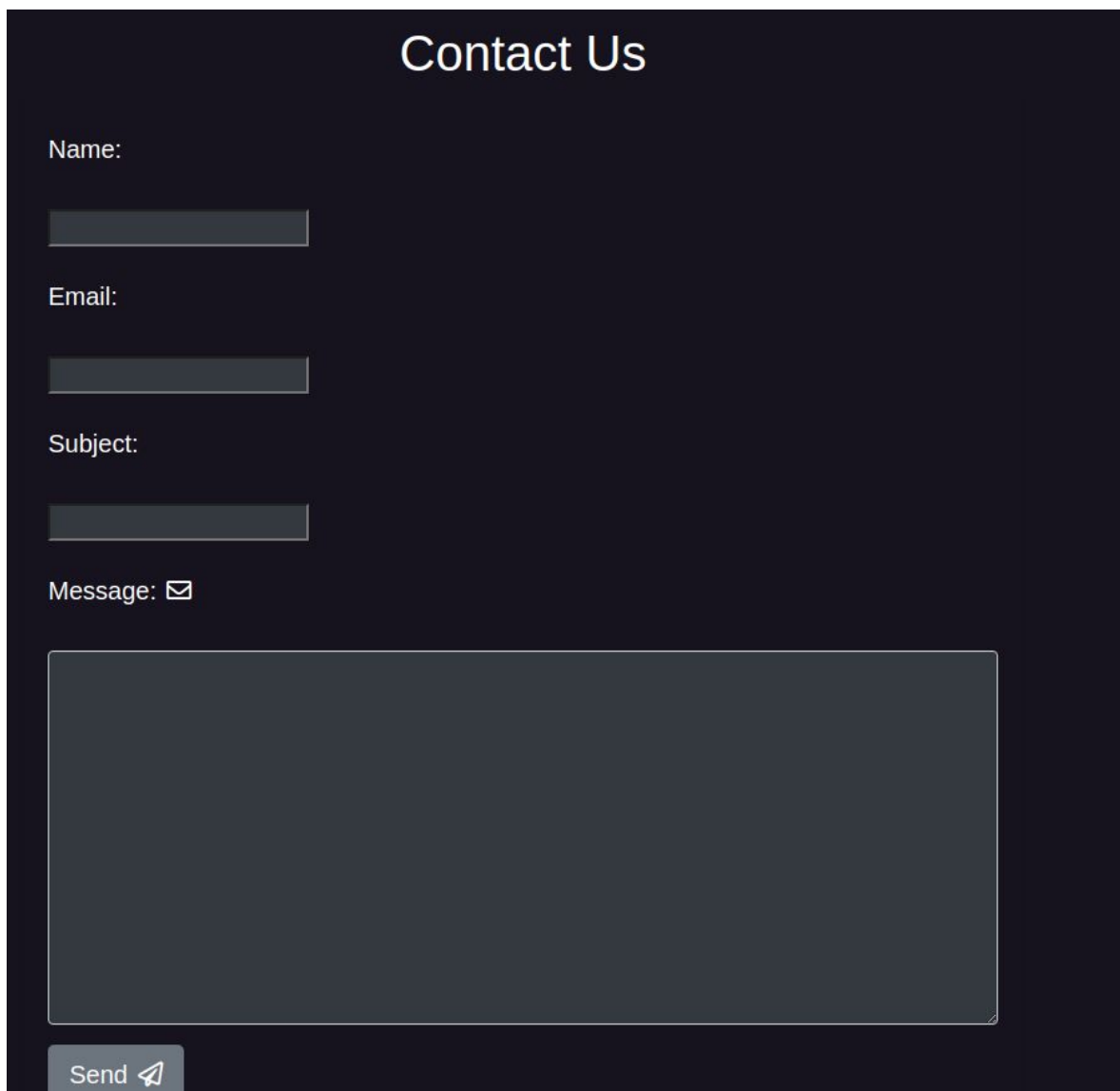
4. Admin's Dashboard



Η πρόσβαση στο dashboard θα πραγματοποιείται έπειτα από σύνδεση του διαχειριστή με τα κατάλληλα στοιχεία και επικύρωση με την βάση δεδομένων που υλοποιήθηκε στο Β Μέρος. Η δομή του περιλαμβάνει ένα πλάγιο μενού πλοήγησης με υπό-σελίδες για επεξεργασία προφίλ, εμφάνιση στατιστικών, φόρμες επικοινωνίας και επεξεργασίας μαθημάτων, σελίδες για μελλοντικά μαθήματα που θα προστεθούν. Επιπλέον, υπάρχει μία οριζόντια μπάρα στην οποία αριστερά υπάρχει ένα κουμπί για ελαχιστοποίηση του dashboard και αριστερά ένα search bar για την διευκόλυνση του διαχειριστή. Το περιεχόμενο της αρχικής σελίδας του dashboard αναφέρει την επωνυμία, την χαρακτηριστική φράση και επιγραμματικά τις γλώσσες που προσφέρονται ή βρίσκονται υπό κατασκευή.

- Στο δεύτερο μέρος της εργασίας αποφασίσαμε να διαγράψουμε την υλοποίηση που εφαρμόσαμε στο πρώτο και ξεκινήσαμε την υλοποίηση του dashboard από το backend και σταδιακά χτίσαμε το front. Η πρόσβαση στο dashboard πραγματοποιείται με την προσθήκη του /admin στο url και οδηγεί σε ένα landing page από όπου μπορεί να κάνει εγγραφή ή σύνδεση. Η λογική που ακολουθήσαμε είναι να μην υπάρχει κουμπί από οποιοδήποτε μέρος της σελίδας το οποίο να οδηγεί στο dashboard ώστε να μην μπορεί ο καθένας να δημιουργήσει λογαριασμό και να αλλοιώσει το περιεχόμενο των μαθημάτων. Με αυτόν τον τρόπο, πολύ συγκεκριμένα άτομα θα γνωρίζουν πως να έχουν πρόσβαση στο ταμπλό και θα διασφαλιστεί η ακεραιότητα του περιεχομένου.

5. Contact page

A dark-themed contact form titled "Contact Us". It features four input fields: "Name:", "Email:", "Subject:", and "Message:". The "Message:" field is preceded by an envelope icon. Below the "Message:" field is a large, empty text area. At the bottom left, there is a "Send" button with a paper plane icon.

Η σελίδα επικοινωνίας οδηγεί σε μια νέα σελίδα που ζητάει προσωπικά στοιχεία. Ο χρήστης εισάγει “Όνομα”, “Email”, “Θέμα” και ακολούθως το γραπτό του μήνυμα. Κάτω από το πεδίο συγγραφής του μηνύματος, υπάρχει ένα κουμπί “Αποστολή”, το οποίο θα μπορεί να πατήσει ο χρήστης αν συμπληρώσει όλα τα πεδία.

- Στο δεύτερο μέρος της εργασίας, τα μηνύματα συνδέονται με βάση δεδομένων στην οποία θα αποθηκεύονται και από όπου θα προσπελαύνονται για ανάγνωση. Όπως έχει προαναφερθεί υπάρχουν δύο υλοποιήσεις για το συγκεκριμένο component.

6. Lesson Structure / Test Layout

C++/Hello World

Congratulations ✓
You have passed this test. Click on "Next Lesson" to continue.

Lesson Summary
This is the test area. After each lesson, you will be asked to write a small piece of code in order to prove that you understood it. The procedure is as follows:

- Read the test exercise below carefully. There is also a summary of the lesson.
- Write your code in the form on the right.
- When you're done, submit your code by pressing the green play button.
- On bottom right, you will see the output of your program.
- If your code is correct, you can proceed to the next lesson. Otherwise, try again, we always learn from our mistakes! 😊

Extra tip: always print your texts as the exercise's ones.

Now for the theory:
Remember that we use "std::cout" followed by "<<" and the text we want to print in order to show it on screen. Example:
std::cout << "test";
Also don't forget to end your statement with a semicolon! (;)

Test Exercise
Use std::cout to print "Hello World!" on your screen.
Be careful! "Hello World!" is correct while "Hello world!" and "Hello World" are incorrect.
("Hello world!" has a lowercase w and "Hello World" misses the exclamation mark.)

[Back to Lesson](#) [Next Lesson](#)

```
1 #include <iostream>
2
3 int main() {
4     std::cout << "Hello world!";
5     return 0;
6 }
```

OUTPUT
Hello World!

To component Lesson Structure αφού πάρει από την βάση δεδομένων τα απαραίτητα στοιχεία για το συγκεκριμένο μάθημα που του ζητηθεί το εμφανίζει σε html μορφή για να είναι πιο όμορφο, επίσης πάνω αριστερά υπάρχει ένα κουμπί όπου εμφανίζει όλα τα μαθήματα της συγκεκριμένης γλώσσας που βλέπει ο χρήστης. Ύστερα ο χρήστης (αφού έχει συνδεθεί) έχει την επιλογή να ανακατευθυνθεί σε τεστ εάν υπάρχει για το συγκεκριμένο μάθημα. Αυτή την λειτουργία την διαχειρίζεται το Test Layout, το οποίο παίρνει από την βάση το αντίστοιχο τεστ του μαθήματος και εμφανίζει εν συντομία το κείμενο του μαθήματος και την εκφώνηση του τεστ. Ο χρήστης ύστερα γράφει κώδικα και μέσω το πράσινου κουμπιού έχει την δυνατότητα να τον τρέξει. Εάν η απάντηση είναι σωστή, ένα μήνυμα επιτυχίας εμφανίζεται και μπορεί να συνεχίσει στο επόμενο μάθημα. Αλλιώς, εάν έχει υπάρξει συντακτικό λάθος με τον κώδικα ζητείται να το διορθώσει. Σε λογικό λάθος, απλά ενημερώνεται πως πρέπει να ξαναπροσπαθήσει. Κάτω από τον editor εμφανίζεται η έξοδος του κώδικα οπότε πάντα μπορεί να δει τι έκανε για να συνεχίσει αναλόγως.

Cin

In most program environments, the standard input by default is the keyboard, and the C++ stream object defined to access it is cin.

For formatted input operations, cin is used together with the extraction operator, which is written as >> (i.e., two "greater than" signs). This operator is then followed by the variable where the extracted data is stored.

For example:

```
int number;
cin >> number;
```

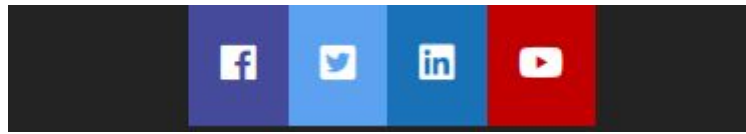
The first statement declares a variable of type int called age, and the second extracts from cin a value to be stored in it. This operation makes the program wait for input from cin; generally, this means that the program will wait for the user to enter some sequence with the keyboard. Once the statement with the extraction operation on cin is reached, the program will wait for as long as needed until some input is introduced.

Tip: Note that the characters introduced using the keyboard are only transmitted to the program when the ENTER (or RETURN) key is pressed.

[Take a Test!](#)

[Previous Lesson](#) [Next Lesson](#)

7. Social Bar



Στο κάτω μέρος της αρχικής σελίδας και όλων των άλλων σελίδων, το “footer”, υπάρχουν εικονίδια με τα λογότυπα των μέσων κοινωνικής δικτύωσης της σελίδας μας. Πιο συγκεκριμένα, οι επισκέπτες μπορούν να ανακατευθυνθούν σε Facebook, Twitter, Google, LinkedIn και Youtube, με αυτή την σειρά.

Ακριβώς κάτω από τα εικονίδια αυτά, υπάρχει το χαρακτηριστικό σήμα των Copyrights, και τα δικαιώματα της ιστοσελίδας.

- Στο δεύτερο μέρος της εργασίας έχουν προστεθεί διασυνδέσεις σε καθένα από αυτά τα εικονίδια. Το εικονίδιο του Facebook έχει οριστεί να κάνει ανακατεύθυνση σε σελίδα που έχουμε δημιουργήσει στον αντίστοιχο ιστότοπο. Τα υπόλοιπα τρία εικονίδια παραπέμπουν σε σελίδα λάθους που πληροφορεί ότι η πληροφορία θα προστεθεί σύντομα.

8. Responsiveness for mobile screen

Αρκετές δοκιμές έγιναν για να υπάρχει ένα όμορφο responsiveness σε κάθε μέρος της ιστοσελίδας, έχοντας μεγάλη βοήθεια και από την react-bootstrap. Μεγάλο παράδειγμα για την λειτουργία του, είναι το header το οποίο για συσκευές με πλάτος $\leq 992px$ προσφέρει ένα “hamburger menu” το οποίο ανοίγει/κλείνει το μενού. Έστερα τα στοιχεία του μενού πιάνουν όλη την γραμμή οπότε μπαίνουν στο κέντρο για μια πιο καλή αξιοποίηση του κενού χώρου. Άλλο ένα καλό παράδειγμα είναι το profile menu το οποίο έχει δικό του layout αλλά καθώς μικραίνει η οθόνη τα στοιχεία καταλαμβάνουν όλη την γραμμή και κεντραρίζονται. Αυτό ισχύει και για το σχεδιασμό των test τα οποία χωρίζονται σε 2 στήλες εκφώνηση και editor αλλά σε μικρές συσκευές είναι το ένα κάτω από το άλλο.

9. Virtual Machine

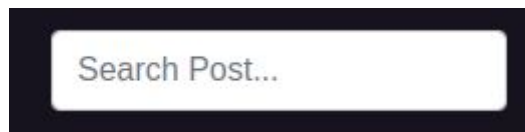
Το site είναι ανεβασμένο στο <http://83.212.77.175/>

Η λειτουργία του virtual machine γίνεται με το nginx. Τα αρχεία του front-end ανεβαίνουν αφού γίνουν static μέσω της εντολής build. Το back-end τρέχει με το pm2 και η βάση δεδομένων παραμένει online στο cloud της mongoDB.

Bonus:

Μετά την ολοκλήρωση των βασικών στοιχείων της ιστοσελίδας, αποφασίσαμε ως ομάδα να ασχοληθούμε και την υλοποίηση κάποιων προαιρετικών τμημάτων όπως αναφέρονται στο έγγραφο με την ανακοίνωση της εργασίας.

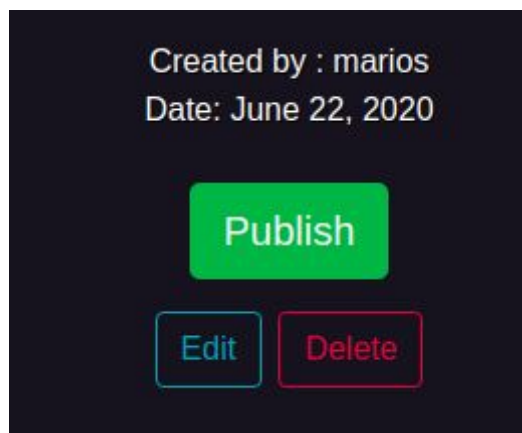
1. Search Bar



Αρχικά, υλοποιήσαμε search bars τόσο στην αρχική σελίδα όσο και στο ταμπλό διαχείρισης για μεγαλύτερη ευκολία και καλύτερη λειτουργικότητα της ιστοσελίδας. Για το Α Μέρος, η μπάρα αναζήτησης υπάρχει στατικά αλλά υπάρχει ως σκέψη για το Β Μέρος να υλοποιηθεί η σύνδεση της με μία βάση δεδομένων η οποία θα κρατάει διάφορα tags για κάθε μάθημα και ανάλογα με την αναζήτηση θα εμφανίζει στον χρήστη τα αποτελέσματα που ανταποκρίνονται καλύτερα στην λέξη που έδωσε.

- Επειδή η ιδέα της search bar αποδείχτηκε δύσκολα υλοποιήσιμη, την έχουμε εφαρμόσει σε ένα περιορισμένο τμήμα της ιστοσελίδας και συγκεκριμένα στο ταμπλό του διαχειριστή όπου είναι χρήσιμη και η υλοποίηση ενός τέτοιου αρχείου για τον εντοπισμό ενός συγκεκριμένου μαθήματος μέσα στην πληθώρα που υπάρχουν.

2. Edit/Add/Delete A Lesson



Επίσης, σαν προαιρετικό κομμάτι αναφέρεται η ύπαρξη φόρμας συμπλήρωσης με παράδειγμα την δημιουργία προϊόντος. Καθώς κάτι τέτοιο δεν καθίσταται εφικτό και δεν θα είχε λειτουργικό ρόλο με το περιεχόμενο της ιστοσελίδας που επιλέξαμε, αποφασίσαμε να δημιουργήσουμε φόρμες για τους διαχειριστές. Συγκεκριμένα, προσαρμόστηκαν στις ανάγκες της FunC και έχοντας υπόψη μας το δεύτερο μέρος της εργασίας, θα υλοποιηθούν φόρμες για την προσθήκη, επεξεργασία και διαγραφή ενός μαθήματος από την βάση δεδομένων. Έτσι, αυτές οι φόρμες διαφοροποιημένες ελάχιστα για να προσαρμοστούν στην λειτουργία της εκάστοτε θα παρέχουν την δυνατότητα στον διαχειριστή να συμπληρώνει τον τίτλο, μάθημα, περιεχόμενο και ενδεικτικά tags και να επιλέγει “Δημοσίευση”, “Ενημέρωση”, ή “Διαγραφή” ανάλογα με την λειτουργία της φόρμας.

- Όντως στο δεύτερο κομμάτι της εργασίας πραγματοποιήθηκε αυτή η λειτουργία μέσα στο Dashboard. Παρόλο που η δομή του έχει αλλάξει, ο διαχειριστής έχει την δυνατότητα να φτιάξει, να ενημερώσει, να δημοσιεύσει και να διαγράψει μαθήματα. Επίσης, μπορεί να επιλέξει για το κάθε μάθημα σε ποια γλώσσα ανήκει και αν θα περιλαμβάνει τεστ.

3. React, Reactstrap, Bootstrap

Η αρχική προσέγγιση για τον προγραμματισμό του front-end ήταν η χρήση html και css. Καθώς κανένας από εμάς δεν είχε κάποια προγενέστερη γνώση σε άλλες τεχνολογίες αποφασίσαμε να ξεκινήσουμε με τις πιο απλές που διδασκόμασταν παράλληλα στα μαθήματα. Αντιληφθήκαμε όμως ότι η χρήση της react θα βοηθούσε την διαδικασία και παράλληλα το αποτέλεσμα θα ήταν πιο αισθητικά άρτιο αλλά και η τήρηση της ανταποκρισιμότητας σε διαφόρους τύπους συσκευής θα μπορούσε να επιτευχθεί με ευκολότερο τρόπο. Έτσι, όλη η ομάδα ξεκίνησε να μαθαίνει react από video tutorials και πολύ πειραματισμό πάνω στον κώδικα.

4. MongoDB Express React NodeJs

Για την υλοποίηση του back-end έχει αποφασιστεί να γίνει χρήση του framework Express το οποίο θα συνδέεται με μια βάση δεδομένων του Mongo.

- Πράγματι, στο δεύτερο μέρος η υλοποίηση των λειτουργιών του backend ολοκληρώθηκε με την τεχνολογία MERN.

Copyrights

Εικόνες: Οι εικόνες που χρησιμοποιήθηκαν για να πλαισιώσουν το carousel έχουν κατεβαστεί από το www.unsplash.com και το www.pixabay.com. Το favicon.ico έχει σχεδιαστεί από τον Μάριο Νίκα, ενώ το logo της σελίδας καθώς και οι εικόνες για τις καρτέλες των μαθημάτων έχουν δημιουργηθεί από τον Γεώργιο Κυνηγόπουλο.

Κείμενα: Τα κείμενα επεξεργάστηκαν από την Λίνα Λυσσούδη με την κατάλληλη επιλογή πληροφοριών από έγκυρες επιστημονικές ιστοσελίδες όπως το cplusplus.com, www.tutorialspoint.com, www.programiz.com, www.w3schools.com, www.geeksforgeeks.org. Οι αντίστοιχες μεταφράσεις στα ελληνικά υλοποιήθηκαν από την Μαρία Εσκιόγλου.

Comparisons and Experimentations

Είμαστε γνώστες ότι η ιδέα της δημιουργίας ιστοσελίδων που θα προσφέρει μαθήματα προγραμματισμού δεν αποτελεί πρωτοπορία, ωστόσο έχουμε διαφοροποιήσει την δικιά μας ώστε να ξεχωρίζει από τις αντίστοιχες που υπάρχουν. Όπως έχει προαναφερθεί, ο προγραμματισμός αποτελεί μεγάλο προσόν για όποιον

το κατέχει και έτσι δεν αποτελεί έκπληξη το γεγονός ότι έχουν δημιουργηθεί πολλές ιστοσελίδες με παρόμοιο άξονα.

Το βασικό πλεονέκτημα που προσφέρει η ιστοσελίδα μας είναι σε σχέση με τον τρόπο διεξαγωγής των μαθημάτων. Πιο συγκεκριμένα, τα μαθήματα κάθε γλώσσας προγραμματισμού είναι αριθμημένα και τοποθετημένα με σειρά που θεωρούμε, από προσωπική εμπειρία, πως θα γίνονται ευκολότερα κατανοητή. Ξεκινώντας από μια εισαγωγή με πληροφορίες για την τρέχουσα γλώσσα, τα μαθήματα περνούν στην επεξήγηση ενός πολύ εύκολου προγράμματος, και ακολούθως αρχίζουν να αναλύουν τις πιο χρήσιμες εντολές, που αποτελούν την βάση όλων των προγραμμάτων. Κατόπιν, όσο προχωρούν τα μαθήματα, ασχολούνται με πιο εξειδικευμένα και συνδυαστικά θέματα, αξιοποιώντας και τις πρόσφατα αποκτηθείσες γλώσσες.

Άλλη μια σημαντική πρωτοπορία της FunC είναι το κομμάτι της πρακτικής εξάσκησης. Η πρακτική ακολουθεί την θεωρητική γνώση, και στόχο της έχει να εξοικειώσει τον χρήστη με το γραμματικό φαινόμενο που μόλις διδάχτηκε, συμπληρώνοντας τις κατάλληλες λέξεις που χρειάζονται στα παραδείγματα ώστε να τρέξει σωστά το πρόγραμμα. Η διαδικασία αυτή λαμβάνει χώρα ακριβώς μετά από το πέρας του θεωρητικού τμήματος, για την ολοκληρωμένη εμπειρία του μαθήματος. Αυτή η λειτουργία θα ολοκληρωθεί αρτιότερα στο Β Μέρος, αλλά αυτή την στιγμή υπάρχει στατικά υλοποιημένη.

Όπως προκύπτει από τα προηγούμενα, η ιστοσελίδα μας αποσκοπεί στην προσφορά μιας λειτουργικής και εύκολης πλοήγησης. Αποφεύγει, δηλαδή, την αναφορά ενοτήτων που πιθανώς ενδιαφέρουν τον χρήστη στο τέλος του μαθήματος, όπως κάνουν πολλές ιστοσελίδες διαδικτυακών μαθημάτων. Αντίθετα, καθορίζει εκ των προτέρων την ροή των μαθημάτων, η πλοήγηση μεταξύ των οποίων προτείνεται να γίνεται με την χρήση του κουμπιού “Επόμενο μάθημα”. Αυτό, φυσικά, δεν στερεί από τον χρήστη την δυνατότητα να μεταβεί σε οποιοδήποτε άλλο μάθημα τον ενδιαφέρει.

Πρωτοτυπία σε σχέση με ανάλογες ιστοσελίδες παρουσιάζεται και στο τρόπο χρέωσης των μαθημάτων. Η FunC πρεσβεύει ότι η γνώση είναι δωρεάν, συνεπώς η θεωρία των μαθημάτων μαζί με παραδείγματα και διευκρινίσεις είναι προσιτή από όλους, χρήστες ή απλούς επισκέπτες της σελίδας. Για την απόκτηση πρόσβασης στα τεστ και τα ατομικά στατιστικά στοιχεία προόδου, το μόνο που χρειάζεται είναι η δημιουργία ενός λογαριασμού, και η παρακολούθηση των μαθημάτων με σύνδεση στον λογαριασμό αυτό. Υπάρχουν, ωστόσο, και χρεώσιμα πακέτα, που θα προσφέρουν στους χρήστες-κατόχους τους δυνατότητα εκμάθησης περισσότερων γλωσσών και ξεκλείδωμα πιο εξειδικευμένων κεφαλαίων θεωρίας και επιπλέον τεστ πρακτικής.

Ένα ακόμα μέρος στο οποίο διαφοροποιείται η ιστοσελίδα μας, είναι η πληθώρα γλωσσών που θα υποστηρίξει. Συγκεκριμένα, υπάρχει η σκέψη σε μελλοντική επέκταση της να προστεθούν επιπλέον γλώσσες, δίνοντας έμφαση σε αυτές που ομιλούνται σε χώρες του τρίτου κόσμου. Έτσι, η FunC θα υπηρετεί και τον

4ο Στόχο του Ο.Η.Ε που είναι η παροχή ποιοτικής εκπαίδευσης σε όλα τα μέρη του κόσμου, χωρίς να υπάρχουν γεωγραφικοί περιορισμοί. Με αυτόν τον τρόπο, η γνώση δεν θα είναι προσβάσιμη από συγκεκριμένα τμήματα του πληθυσμού, όπως συμβαίνει στις περισσότερες ιστοσελίδες εκμάθησης προγραμματισμού. Εκεί έγκειται και η διαφορετικότητα της σελίδας, η οποία στοχεύει στην παροχή γνώσης στην μητρική γλώσσα του χρήστη, για ευκολότερη κατανόηση των εννοιών.

Τέλος, ιδιαιτερότητα της FunC συγκριτικά με άλλες ιστοσελίδες που διδάσκουν γλώσσες προγραμματισμού διαδικτυακά, είναι η ενσωμάτωση video tutorials με την περαιτέρω επεξήγηση δύσκολων εννοιών ή ακόμα και ασκήσεων για την ευκολότερη κατανόηση τους από τους χρήστες. Συγκεκριμένα, μελλοντικός στόχος της ιστοσελίδας είναι η πρόσληψη ειδικών για την δημιουργία βιντεοσκοπημένων διαλέξεων με περιεχόμενο κατάλληλο ανάλογα με τον εκάστοτε θεματική ενότητα.

Τα μαθήματα που προσφέρει η FunC κάνουν έντονη την παρουσία ενός τμήματος πειραματισμού. Στην περίπτωση του συνδεδεμένου χρήστη, ακολουθούν την λογική πραγματικών μαθημάτων, όπου πρώτα διδάσκεται η θεωρία μιας θεματικής ενότητας, και στην συνέχεια ο χρήστης υποβάλλεται σε τεστ αξιολόγησης. Πειραματίζεται, στην κυριολεξία, κάνοντας χρήση των νέων του γνώσεων και προσπαθώντας να συμπληρώσει τις σωστές λέξεις, που θα του κερδίσουν την πρόκριση στο επόμενο μάθημα.

Όταν ολοκληρωθεί με επιτυχία κάποιο τεστ, ενημερώνονται αυτόματα τα στατιστικά στοιχεία προόδου του χρήστη, που φαίνονται στο προφίλ του. Εκεί είναι ευδιάκριτος, με μορφή παρασήμου, ο αριθμός των τεστ που έχει ολοκληρώσει συνολικά, που θα αποθηκεύεται μαζί με τα στοιχεία του χρήστη. Έτσι θα μπορεί την επόμενη φορά να συνεχίσει από το σημείο που είχε σταματήσει χωρίς να χάνεται η πρόοδος του.

Για την διευκόλυνση τόσο των χρηστών όσο και των διαχειριστών της ιστοσελίδας, προστέθηκαν progress bars στην υλοποίηση για την επίδειξη της προόδου με γραφικό χαρακτήρα. Τα progress bars έχουν τοποθετηθεί στο dashboard των διαχειριστών και στα profile των χρηστών ώστε να ανταποκρίνονται στις ανάγκες της εκάστοτε κατηγορίας.

Για τους χρήστες, έχει τοποθετηθεί στο κάτω μέρος του προφίλ τους ένα εξατομικευμένο progress bar για κάθε μάθημα που επέλεξαν να παρακολουθήσουν και ένα κουμπί με την ένδειξη “Αρχή” ή “Συνέχεια”. Το πρώτο κουμπί αντιστοιχεί σε μάθημα που έχει επιλέξει ο χρήστης αλλά δεν έχει ξεκινήσει ακόμα, ενώ το δεύτερο σε μαθήματα στα οποία έχει ολοκληρωθεί ένα τμήμα τους. Ως αποτέλεσμα, οι χρήστες μπορούν να διαχειριστούν την πρόοδο τους, χωρίς να χρειάζεται να επαναλάβουν μέρη της ύλης που έχουν ήδη μελετήσει. Ωστόσο, κανένας δεν τους αποτρέπει από το να ξανά επισκεφτούν προηγούμενα τμήματα, καθώς το progress bar λειτουργεί σαν υπενθύμιση και όχι σαν αποτρεπτικός παράγοντας.

Αντίστοιχα, για τους διαχειριστές της ιστοσελίδας, τα progress bar έχουν βοηθητικό χαρακτήρα και είναι τοποθετημένα στο dashboard που εμφανίζεται έπειτα

από login με στοιχεία admin, τα οποία θα είναι αποθηκευμένα σε ειδική βάση δεδομένων. Τα στατιστικά στοιχεία των admin περιλαμβάνουν, προβολές σελίδων, διαγράμματα με παραμέτρους τους χρήστες που αποφάσισαν να αγοράσουν κάποιο πρόγραμμα, χρήστες που ολοκλήρωσαν κάποιο πρόγραμμα, ή αυτούς που αποφάσισαν να ξεκινήσουν μία καινούρια προγραμματιστική γλώσσα. Όλα αυτά θα βοηθήσουν τον εκάστοτε διαχειριστή να προσαρμόσει τα μαθήματα ανάλογα με την ζήτηση των χρηστών, εξατομικεύοντας τα σύμφωνα με τις ανάγκες τους.

6.Future ideas

Για την μελλοντική επέκταση της ιστοσελίδας υπάρχουν ήδη αρκετές ιδέες για την πιο άρτια και ολοκληρωμένη διεκπεραίωση των μαθημάτων. Καταρχάς, θα προστεθούν επιπλέον γλώσσες προγραμματισμού για τις οποίες θα προσφέρονται μαθήματα, η πρόσβαση στις οποίες θα περιορίζεται ανάλογα με το πρόγραμμα χρέωσης που έχει επιλέξει ο χρήστης. Σε όλες τις γλώσσες θα προστεθούν επιπλέον, πιο προχωρημένα μαθήματα, που θα ενισχυθούν και με video tutorials δικής μας παραγωγής.

Θα ήταν επίσης συνετό να γίνει μια καλύτερη συντήρηση του σάιτ όπως πιο καλογραμμένος και κατανοητός κώδικας αλλά πιο σημαντικό θα ήταν να υπάρχει μια ορισμένη ασφάλεια για τα τεστ που κάνουν compile κώδικα, καθώς μιας και δεν υπάρχουν αρκετοί έλεγχοι για τους “κακόβουλους” χρήστες.

Μια πρωτότυπη ιδέα είναι να δημιουργηθούν αναγνωριστικά τεστ, τα οποία θα αξιολογούν την εμπειρία του χρήστη και θα του προτείνουν απο ποιο σημείο θα μπορούσε να ξεκινήσει τα μαθήματα κάποιας γλώσσας. Η τελική απόφαση θα είναι πάντα δική του, σε περίπτωση που νιώθει αβέβαιος για τις γνώσεις του σε κάποια προηγούμενη ενότητα.

Αναμφίβολη θεωρείται η μετάφραση της FunC σε περισσότερες γλώσσες, ώστε να είναι πιο προσιτή. Είναι, άλλωστε, μια ιστοσελίδα που στοχεύει στην παροχή γνώσης ανεξαρτήτου γεωγραφικού μήκους και πλάτους, οπότε η πολυγλωσσία της είναι απαραίτητη για να θεωρείται η εκπαίδευση που προσφέρεται ποιοτική. Αυτό θα βοηθήσει και στην καταπολέμηση της γνωστικής ανισότητας μεταξύ αναπτυγμένων και αναπτυσσόμενων χωρών.

Για την ενίσχυση της αλληλεπίδρασης μεταξύ των χρηστών και την βαθύτερη κατανόηση των αποκτηθέντων γνώσεων, θα δημιουργηθούν real - time διαγωνισμοί με έπαθλα ως κίνητρο για την συμμετοχή. Έτσι, η ιστοσελίδα θα προσφέρει μία πιο hands-on εμπειρία σε θέματα προγραμματισμού, προσφέροντάς τους μια ρεαλιστική εικόνα του ανταγωνισμού που επικρατεί σε πραγματικές συνθήκες.