ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ ΣΤΑ ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ

ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ ΔΙΑΧΕΙΡΙΣΗΣ ΧΡΟΝΟΥ: ΑΞΙΟΠΟΙΗΣΗ ΤΟΥ ANGULAR JAVASCRIPT FRAMEWORK ΚΑΙ ΣΥΓΚΡΙΣΗ ΤΟΥ ΜΕ ΤΗ ΒΙΒΛΙΟΘΗΚΗ JQUERY

Διπλωματική Εργασία

Του

ΑΝΑΣΤΑΣΗ ΜΗΤΡΟΝΑΤΣΙΟΥ ΤΟΥ ΣΤΕΦΑΝΟΥ

Επιβλέπων Καθηγητής: ΝΙΚΟΛΑΟΣ ΠΡΩΤΟΓΕΡΟΣ

Υποβλήθηκε ως απαιτούμενο για την απόκτηση του Μεταπτυχιακού Διπλώματος Ειδίκευσης στα Πληροφοριακά Συστήματα

Μάρτιος 2018

Ευχαριστίες

Θα ήθελα να εκφράσω τις ευχαριστίες μου στον κ. Νικόλαο Πρωτόγερο, επιβλέποντα καθηγητή μου, για την εμπιστοσύνη του και την καθοδήγηση, που μου παρείχε καθ' όλη τη διάρκεια της εκπόνησης της διπλωματικής μου εργασίας.

Ευχαριστώ θερμά την οικογένειά μου για την αμέριστη συμπαράσταση, την οικονομική βοήθεια καθώς και την κατανόηση που μου έδειξαν όλο αυτό το διάστημα.

Επιπρόσθετα, θα ήθελα να ευχαριστήσω όλους τους φίλους μου, που με βοήθησαν και με στήριξαν, είτε με το χιούμορ τους, είτε με συμβουλές καθ' όλη τη διάρκεια των μεταπτυχιακών μου σπουδών.

“Design is not just what it looks like and feels like.

Design is how it works.”

Steve Jobs, 1955-2011

Περίληψη

Σε μια εποχή που η τεχνολογία εξελίσσεται με ραγδαίες ταχύτητες ο τομέας των γλωσσών προγραμματισμού δε θα μπορούσε παρά να αποτελεί το επίκεντρο. Με δεδομένο ότι μια εφαρμογή μπορεί να αναπτυχθεί με πολλούς και διάφορους τρόπους και ότι στον προγραμματισμό ο όρος «σωστό» και «λάθος» δεν υφίσταται, είναι πολύ σημαντικό να αποφασιστεί η σωστή μεθοδολογία και τεχνολογία που θα ακολουθηθεί κατά την δημιουργία της.

Η παρούσα διπλωματική εργασία έχει ως στόχο τη δημιουργία μιας πλατφόρμας που θα διευκολύνει τη συνεργασία φοιτητών και καθηγητών κατά την εκπόνηση των εργασιών και βασίστηκε στη μελέτη και τη σύγκριση βιβλιοθηκών Javascript και Javascript Frameworks. Με τη νέα πλατφόρμα έχουμε μια πιο σαφή εικόνα για το πώς τα δυο παραπάνω μοντέλα λειτουργούν. Η εκπόνηση της ήταν αποτέλεσμα προσωπικής έρευνας και πρακτικής εφαρμογής σε συνδυασμό με την ανάλυση της εγχώριας και διεθνούς βιβλιογραφίας. Αρχικά έγινε η συλλογή των δεδομένων και η απαραίτητη βιβλιογραφική έρευνα αποτέλεσμα της οποίας ήταν η ανάπτυξη μιας πρωτότυπης εφαρμογής με δύο διαφορετικούς τρόπους τεχνολογίας. Στη συνέχεια, και κατόπιν σύγκρισης των αποτελεσμάτων που έδωσαν οι δυο αρχικές εφαρμογές, προέκυψε η ανάγκη για τη δημιουργία μιας τρίτης εφαρμογής. Η νέα και τελική εφαρμογή, η οποία βασίζεται στα αποτελέσματα έρευνας του Πανεπιστημίου Μακεδονίας, έχει ως αρχικό στόχο την παρουσίαση συναντήσεων φοιτητή - καθηγητή για την εκπόνηση διπλωματικής εργασίας και κατ΄ επέκταση την ανάπτυξη μιας πιο αποτελεσματικής σχέσης μεταξύ τους, σε μία νέα πιο οργανωτική και φιλική βάση.

Τέλος, στηριζόμενοι, στις πληροφορίες που μας παρέχει το παραπάνω πρόγραμμα είμαστε σε θέση να καταγράψουμε τα συμπεράσματά μας με βάση τα αποτελέσματα από την πρακτική της εφαρμογή. Επίσης παραθέτουμε κάποιες ενδεικτικές προτάσεις σχετικά με την βελτιστοποίηση της εφαρμογής «LetMeKnow» ώστε αυτή να γίνει ακόμα πιο αποτελεσματική και φιλική προς το χρήστη.

Λέξεις Κλειδιά: Javascript, Angular, JQuery, Framework, Φοιτητής

Abstract

In an era that technology is developing rapidly, the section of programming languages couldn't be less than the focal point. Granted that an application can be evolved in many and various ways and that in programming the word «correct» and «wrong» does not exist, it is very important to decide the right methodology and technology that will be followed during its formation.

The present diplomatic project aims at creating a new platform that enables the cooperation between professors and their students during the preparation of their thesis and it count on studying and comparing the Javascript libraries and Javascript Frameworks. This new platform gave us a clearer picture about how the two above models function. Its elaboration was a result of personal research and practical application in combination with the analysis of the local and international bibliography. Initially there was the collection of data and the necessary bibliographical research which resulted in the development of an original application with two different ways of technology. Then, and after comparing the results of the two initial applications, a need for the creation of a third application arose. The new and final application, which is based on the results of a survey by the university of Macedonia, haw aw initial target the presentation of meetings between students – teachers for the elaboration of a diplomatic project and moreover the development of a more effective relation between them on a new more organized and friendlier basis.

Finally, based on the information that is mentioned in the above program, we are able to write down our conclusions due to the results by its practical application. Moreover we cite some indicative proposals relevant to the improvement of the «Let me know» application in order to be more effective and friendly towards the user.

Keywords: Javascript, Angular, JQuery, Framework, Student

Κατάλογος εικόνων

Εικόνα 1: Βασική σύνταξη της JQuery

Εικόνα 2: Βασική σύνταξη της React

Εικόνα 3: Αποτέλεσμα του κώδικα, της εικόνας 2

Εικόνα 4: Βασική σύνταξη της Angular

Εικόνα 5: Output του κώδικα του Angular

Εικόνα 6: Διαφορές JQuery - Angular

Εικόνα 7: Διαφορές Κώδικα JQuery - Angular

Εικόνα 8: Two-way Binding

Εικόνα 9: One way & Two-way Data Binding

Εικόνα 10: Αρχική σελίδα της εφαρμογής

Εικόνα 11: Σελίδα Student

Εικόνα 12: Δομή αρχείων εφαρμογής «Library»

Εικόνα 13: Δομή αρχείων «Framework»

Εικόνα 14: Κλήση Request/Response

Εικόνα 15: Κώδικας της σελίδας home

Εικόνα 16: Κώδικας της σελίδας student

Εικόνα 17: Κώδικας της σελίδας index

Εικόνα 18: Κώδικας Render εγγραφής «Library»

Εικόνα 19: Κώδικας Render εγγραφής «Framework»

Εικόνα 20: Καταμέτρηση γραμμών

Εικόνα 21: Χωρητικότητα εφαρμογής «Library» στον server.

Εικόνα 22: Χωρητικότητα εφαρμογής «Framework» στον server

Εικόνα 23: Σύγκρισηχώρου στον διακομιστή

Εικόνα 24: Χρόνος φόρτωσης σελίδας «home»

Εικόνα 25: Χρόνος φόρτωσης σελίδας «Student»

Εικόνα 26: Χρόνος φόρτωσης εφαρμογής «Framework»

Εικόνα 27: Σύγκριση χρόνου φόρτωσης ιστοσελίδας

Εικόνα 28: Σχήμα βάσης δεδομένων

Εικόνα 29: Welcome dashboard

Εικόνα 30: Student page

Εικόνα 31: Εισαγωγή νέας επαφής

Εικόνα 32: Επιτυχία εισαγωγή επαφής

Εικόνα 33: Αναδυόμενο μήνυμα διαγραφής

Εικόνα 34: Προσωπικά στοιχεία φοιτητή

Εικόνα 35: Ιστορικό συναντήσεων φοιτητή με τον επιβλέποντα καθηγητή

Εικόνα 36: Δημιουργία νέας συνάντησης

Εικόνα 37: Απεικόνιση καρτέλας φοιτητή σε συνδυασμό με το ιστορικό συναντήσεων.

Εικόνα 38: Δημιουργία νέας υπενθύμισης

Εικόνα 39: Εμφάνιση της υπενθύμισης

Κατάλογος πινάκων

Πίνακας 1: Πληροφορίες ίδρυσης και επίσημη σελίδα του κάθε Framework-library

Πίνακας 2: Ιστοσελίδα των bugs και άδεια χρήσης

Περιεχόμενα

Ευχαριστίες ii

Περίληψη v

Abstract vi

Κατάλογος εικόνων viii

Κατάλογος πινάκων x

Κεφάλαιο 1 1

1.1 Εισαγωγή 1

1.2 Περιεχόμενο διπλωματικής εργασίας 2

Κεφάλαιο 2 Βιβλιογραφική Επισκόπηση 3

2.1 Εισαγωγή 3

2.2. Javascript 3

2.3 Ανάλυση και Σύγκριση των Javascript Frameworks και Libraries 5

2.3.1 JQuery 5

2.3.2 React 7

2.3.3 AngularJS 9

2.4 Σύγκριση της βιβλιοθήκης JQuery και Framework Angular 14

2.4.1 Μέγεθος του κώδικα 15

2.4.2 Χρόνος φόρτωσης Ιστοσελίδας 16

2.4.3 Ασφάλεια 16

2.4.4 Two-wayBinding 16

Κεφάλαιο 3 : Μεθοδολογία 18

3.1 Εισαγωγή 18

3.2 Μέτρα σύγκρισης εφαρμογής 18

3.3 Έρευνα και μελέτη 19

3.4 Διαδικασία αξιολόγησης 20

Κεφάλαιο 4 : Αποτελέσματα υλοποίησης 21

4.1 Εισαγωγή 21

4.2 Παρουσίαση εφαρμογής 21
4.2.1 Δομή εφαρμογής «Library» 22
4.2.2 Δομή εφαρμογής «Framework» 24
4.2.3 Σύγκριση «Library» με «Framework» 25
4.2.4 Δημιουργία νέα σελίδας 25
4.2.5 Render εγγραφής 28
4.2.6 Χώρος κατανάλωσης στον server 29
4.2.7 Χρόνος φόρτωσης σελίδας 31
4.3 LetMeKnow 33
4.3.1 Αποθήκευση πληροφορίας 34
4.3.2 Welcome 35
4.3.3 Καρτέλα «Student» 37
4.3.4 Νέα επαφή 38
4.3.5 Διαγραφή φοιτητή 39
4.3.7 Καταγραφή Συνάντησης 40
4.3.8 Καρτέλα ToDo 42
4.4 Αξιολόγηση εφαρμογής 43
Κεφάλαιο 5 : Συμπεράσματα και προτάσεις 44
5.1 Συμπεράσματα 44
5.2 Προτάσεις 45
Παράρτημα κώδικα 46

Κεφάλαιο 1

1.1 Εισαγωγή

Στόχος της παρούσας διπλωματικής εργασίας είναι η δημιουργία μιας εφαρμογής διαχείρισης χρόνου με την χρήση του Angular Javascript Framework. Στη συνέχεια και κατά την πρακτική εφαρμογή της θα χρησιμοποιηθεί πιλοτικά στη διαχείριση χρόνου των καθηγητών – φοιτητών με στόχο να διευκολυνθεί η συνεργασία τους με τους κατά την εκπόνηση των εργασιών. Για τη δημιουργία της βασιστήκαμε στη μελέτη και τη σύγκριση βιβλιοθηκών Javascript και Javascript Frameworks και πιο συγκεκριμένα, συγκρίναμε τη βιβλιοθήκη Javascript JQuery με ένα Javascript Framework, μετρώντας διάφορες παραμέτρους ώστε να εντοπίσουμε τις διαφορές τους ακολουθώντας την παρακάτω μεθοδολογία.

Αφού βρούμε τα μετρήσιμα και συγκρίσιμα χαρακτηριστικά θα δημιουργήσουμε μία πρωτότυπη εφαρμογή με δύο διαφορετικούς τρόπους. Στη μία περίπτωση θα την υλοποιήσουμε με την χρήση της βιβλιοθήκης της JQuery και στην άλλη με το Framework Angular. Αφού ολοκληρώσουμε τις εφαρμογές θα συγκρίνουμε τις μετρήσεις που κάναμε ώστε να απαντήσουμε στα βασικά ερωτήματα που έχουμε θέσει όπως το μέγεθος του κώδικα, το χρόνο φόρτωσης της ιστοσελίδας και την ασφάλεια του ιστότοπου. Χρησιμοποιώντας στη συνέχεια την καλύτερη από τις δύο τεχνολογίες θα δημιουργήσουμε μία νέα πλήρη εφαρμογή που θα είναι η διαχείριση των διπλωματικών εργασιών ενός καθηγητή με στόχο να απαντήσουμε σε ερωτήματα για την τεχνολογία του τύπου: πόσο είναι εύχρηστη – προσβάσιμη , επεκτάσιμη και λειτουργική στον χρήστη είναι μια εφαρμογή που παράγεται μ' αυτή την τεχνολογία.Μια Javascript βιβλιοθήκη είναι ένα σύνολο γενικών ή ειδικών συναρτήσεων /κλάσεων, οι οποίες μας παρέχουν γενικευμένες (abstracted) ή ειδικές λύσεις (opinionated) πάνω σε ένα σύνολο προβλημάτων.

Το Javascript Framework αποτελεί ένα παγκόσμιο, επαναχρησιμοποιήσιμο μοντέλο προγραμματισμού που παρέχει έτοιμες συναρτήσεις και λύσεις, μέρος ενός μεγαλύτερου προγράμματος – γλώσσας προγραμματισμού, για την διευκόλυνση των λογισμικών εφαρμογών και προϊόντων. Ένα Framework είναι χρήσιμο στην γρήγορη δημιουργία κώδικα, στην επαναχρησιμοποίηση και στην επεκτασιμότητα του από άλλους

προγραμματιστές. Εν' ολίγοις, είναι ένας σκελετός με γενικές λειτουργίες που σε συνδυασμό με επιπρόσθετο κώδικα από τον προγραμματιστή, δημιουργείται μια εφαρμογή.

Η βασική διαφορά των βιβλιοθηκών Javascript από τα Frameworks Javascript είναι ότι στις βιβλιοθήκες ο προγραμματιστής έχει περισσότερο έλεγχο. Όταν χρησιμοποιούμε μια παραδοσιακή βιβλιοθήκη, έχουμε όλο τον έλεγχο: καλούμε τις μεθόδους της βιβλιοθήκης και χρησιμοποιούμε το αποτέλεσμα. Ένα Framework αντιστρέφει τη ροή του έλεγχου. Εάν καλέσουμε μια μέθοδο ενός Framework θα συμβούν πολλά πράγματα στην εφαρμογή μας, τα οποία δεν έχουμε δηλώσει στον κώδικα μας. Ωστόσο, υπάρχουν και χρησιμοποιούνται εσωτερικά από το Framework.

1.2 Περιεχόμενο διπλωματικής εργασίας

Η διπλωματική εργασία αποτελείται από τέσσερα κεφάλαια. Μετά το πρώτο κεφάλαιο, όπου είναι η εισαγωγή, ακολουθεί το δεύτερο, στο οποίο παρουσιάζεται η βιβλιογραφική επισκόπηση. Πιο συγκεκριμένα, στο κεφάλαιο αυτό γίνεται ανάλυση των δύο πιο δημοφιλών Javascript Framework (React και Angular) και της βιβλιοθήκης JQuery. Τέλος, γίνεται η σύγκριση της τελευταίας με το Framework Angular. Στην συνέχεια στο τρίτο κεφάλαιο παρουσιάζεται η μεθοδολογία σύγκρισης των δύο βιβλιοθηκών. Το τέταρτο κεφάλαιο παρουσιάζει τα αποτελέσματα της πρωτότυπης εφαρμογής που αναπτύχθηκε με δύο τρόπους. Στην συνέχεια, μελετώντας και συγκρίνοντας τις εφαρμογές αυτές, οδηγούμαστε στην ανάπτυξη της νέας εφαρμογής «LetMeKnow». Η εργασία τελειώνει με το πέμπτο κεφάλαιο και την καταγραφή των συμπερασμάτων που προέκυψαν με την ολοκλήρωση της μελέτης των εφαρμογών και της διαδικτυακής εφαρμογής που υλοποιήθηκε. Επιπρόσθετα, γίνεται αναφορά σε κάποιες ενδεικτικές προτάσεις που θα συμβάλουν στην μελλοντική βελτίωση της εφαρμογής αυτής.

Κεφάλαιο 2 Βιβλιογραφική Επισκόπηση

2.1 Εισαγωγή

Το κεφάλαιο αυτό περιλαμβάνει την βιβλιογραφική επισκόπηση της γλώσσας προγραμματισμού Javascriptκαι τωνJavascript Frameworks.

Σύμφωνα με τον Douglas Crockford (2008) η Javascript (JS) θεωρείται μία από τις πιο σημαντικές γλώσσες προγραμματισμού, κυρίως λόγο της στενής της σύνδεσης με τους φυλλομετρητές του Παγκόσμιου Ιστού (WebBrowsers). Σήμερα η συγκεκριμένη γλώσσα προγραμματισμού είναι μια από τις πιο διαδεδομένες, καθώς δίνει την δυνατότητα στον χρήστη να αλληλεπιδρά με την εφαρμογή και το περιβάλλον της. Εισήγαγε ουσιαστικά την έννοια της διαδραστικότητας και δόθηκε η ευκαιρία στον χρήστη να ελέγχει δυναμικά τον τρόπο παρουσίασης την πληροφορίας και να συμμετέχει στην εξέλιξη της εφαρμογής. Εξαιτίας αυτής της ιδιότητα, ο χρήστης μπορεί να καθορίσει τη μορφή, τη σειρά και την ταχύτητα με βάση την οποία παρουσιάζεται η πληροφορία.

2.2. Javascript

Η γλώσσα προγραμματισμού Javascript είναι μια δυναμική , τυποποιημένη και ερμηνευτική γλώσσα. Είναι βασισμένη και τυποποιημένη πάνω στη γλωσσική προδιαγραφή του ECMA Script. Θεωρείται μια από τις σημαντικότερες τεχνολογίες παραγωγής του παγκόσμιου ιστού μαζί με την HTML και την CSS. Ως γλώσσα διαθέτει μια Διεπαφή Προγραμματισμού Εφαρμογών (Application Programming Interface - API) για την εργασία με ημερομηνίες, κείμενο, κανονικές εκφράσεις και πίνακες. Χρησιμοποιείται επιπλέον σε περιβάλλοντα που δεν βασίζονται στο δίκτυο, όπως είναι οι εξειδικευμένοι φυλλομετρητές, τα έγγραφα PDF και εφαρμογές της επιφάνειας εργασίας.

Παρά το όνομα της, η Javascript δεν σχετίζεται με την Java. Υπάρχουν κάποιες ονοματικές, προδιαγραφικές και συντακτικές ομοιότητες αλλά είναι πολύ διαφορετικές σημασιολογίες. Το συντακτικό της για την ακρίβεια προέρχεται από την γλώσσα προγραμματισμού C, ενώ από τις Self και Scheme επηρεάζεται στην σημασιολογία της

και τον σχεδιασμό της. Είναι μια γλώσσα βασισμένη σε πρώτης τάξης λειτουργίες και πρωτότυπη που την καθιστά ως μια γλώσσα παραδειγματική και με την δυνατότητα υποστήριξη αντικειμενοστρέφειας.

Η Javascript ως γλώσσα προγραμματισμού δημιουργήθηκε αρχικά από τον Brendan Eich με την επωνυμία Mocha, μέσα σε 10 μέρες τον Μάιο του 1995 καθώς αυτός εργαζόταν για την εταιρία επικοινωνιών Netscape. Ενώ η εταιρία αυτή ανταγωνίζονταν την εταιρία Microsoft για την χρήσης διαδικτυακών τεχνολογιών αλλά και πλατφόρμων, είχε ως προτέρημα να λαμβάνει υπόψη της το διακομιστή της από την πλευρά του πελάτη, προσφέροντας έτσι ένα κατανεμημένο λειτουργικό σύστημα με μια φορητή έκδοση της SunMicrosystems της Java. Αποτέλεσμα αυτού είναι η παροχή ενός περιβάλλοντος, στο οποίο θα μπορούσαν να «τρέξουν» μικροεφαρμογές. Η Java ήταν ανταγωνιστής της C++ και είχε ως στόχο τους επαγγελματίες προγραμματιστές, ενώ η Netscape ήθελε μία ελαφριά ερμηνευτική γλώσσα που θα συμπλήρωνε την Java και να μπορούσε να χρησιμοποιείται και από μη επαγγελματίες προγραμματιστές, όπως η VisualBasic της Microsoft .

Αργότερα, μετονομάστηκε σε LiveScript και τον Σεπτέμβριο του 1995 κυκλοφόρησε στην αγορά σε δοκιμαστική έκδοση (beta) για πρώτη φορά με ένα πρόγραμμα περιήγησης για τον Παγκόσμιο Ιστό (Web), το Netscape Navigator. Τέλος, μετονομάστηκε σε Javascript (JS) τον Δεκέμβριο του 1995. Ονομάστηκε έτσι κυρίως επειδή η ανάπτυξη της βασίστηκε περισσότερο πάνω στην γλώσσα προγραμματισμού Java. Γρήγορα απέκτησε επιτυχία, καθώς εκτελεί και παράγει στον browser του επισκέπτη (πελάτη) το αποτέλεσμα και περιλήφθηκε σε πολλά προγράμματα περιήγησης στο web.

Η εταιρία Microsoft ονόμασε την εφαρμογή της Jscript και πρόσθεσε νέες μεθόδους για την διόρθωση κάποιων προβλημάτων. Τον Αύγουστο του 1996 κυκλοφόρησε ο browserInternetExplorer 3.0 που περιελάμβανε την JScript. Η Netscape τον Νοέμβριο του 1996 ανακοίνωσε ότι υπέβαλε την γλώσσα για εξέταση ως βιομηχανικό πρότυπο στην οργάνωση EcmaInternational (οργάνωση της τυποποίησης γλωσσών προγραμματισμού), με αποτέλεσμα την τυποποιημένη μορφή ECMAScript.

Σήμερα η Javascript είναι μια από τις δημοφιλέστερες γλώσσες προγραμματισμού ηλεκτρονικών υπολογιστών στον Παγκόσμιο Ιστό αν και αρχικά πολλοί προγραμματιστές την υποτίμησαν. Η Javascript με την χρήση της τεχνολογίας Ajax κέντρισε το ενδιαφέρων των προγραμματιστών, με αποτέλεσμα την εξάπλωση χρήσης

την γλώσσα εντός και εκτός των προγραμμάτων περιήγησης του Web. Το έργο CommonJS ιδρύθηκε τον Ιανουάριο του 2009 με στόχο τον καθορισμό ενός κοινού προτύπου βιβλιοθήκης για την ανάπτυξη της γλώσσας και σε άλλες τεχνολογίες.

2.3 Ανάλυση και Σύγκριση των Javascript Frameworks και Libraries

Εν συνεχεία θα αναλυθούν οι λειτουργίες τριών ευρέως χρησιμοποιούμενων πλαισίων της Javascript για το web: AngularJS, ReactJS και JQuery. Στο τέλος θα γίνει η σύγκριση της βιβλιοθήκης της Javascript JQuery και το Framework Angular

2.3.1 JQuery

Η JQuery είναι μια γρήγορη, μικρή και πλούσια σε χαρακτηριστικά βιβλιοθήκη της Javascript η οποία κυκλοφόρησε τον Ιανουάριο του 2006 από τον John Resig στο Bar Camp της Νέας Υόρκης Επηρεάστηκε από μια παλαιότερη βιβλιοθήκη του Dean Edwards με όνομα css Query. Έχει δυνατότητες όπως η επεξεργασία και η μετακίνηση του DOM (Document Object Model), την διαχείριση των events, χρήση του AJAX με ένα απλούστερο API το οποίο δουλεύει σε ένα μεγάλο αριθμό των φυλλομετρητών. Επί του παρόντος διατηρείται από μια ομάδα προγραμματιστών, όπου επικεφαλής τους είναι ο TimmyWillison.

Το Μοντέλο Αντικειμένου Εγγράφου (DOM – Document Object Model), είναι μια διεπαφή προγραμματισμού εφαρμογών πλατφόρμας και γλώσσας, η οποία αντιμετωπίζει ένα έγγραφο HTML, XHTML ή XML ως μια δομή δένδρου, όπου κάθε κόμβος είναι ένα αντικείμενο που αντιπροσωπεύει ένα τμήμα του εγγράφου. Απλούστερα είναι μια δομική αναπαράσταση δένδρου που περιέχει όλα τα στοιχεία μιας ιστοσελίδας και η JQuery βοηθάει στην απλοποίηση της σύνταξης για επιλογή, εύρεση και χειρισμό αυτών των στοιχείων. Τα αντικείμενα μπορούν να επεξεργαστούν μέσω προγραμματισμού και οι αλλαγές που προκύπτουν εμφανίζονται στο έγγραφο (https://en.wikipedia.org/wiki/Document_Object_Model).

Με τον συνδυασμό της ευελιξίας που προσφέρει καθώς και της επεκτασιμότητας της, η JQuery έχει αλλάξει τον τρόπο με τον οποίο εκατομμύρια άνθρωποι γράφουν

Javascript (https://JQuery.com/). Είναι σχεδιασμένη για να απλοποιεί την υλοποίηση σεναρίου HTML από την πλευρά του πελάτη και χρησιμοποιείται σε πάνω από το 65% των 10.000.000 ιστότοπων με την μεγαλύτερη επισκεψιμότητα (https://el.wikipedia.org/wiki/JQuery). Σημαντικές ιστοσελίδες που χρησιμοποιούν την JQuery είναι το Twitter, το Linkedin και το eBay.

Η βιβλιοθήκη αυτή διατίθεται με την άδεια χρήσης GNU (GeneralPublicLicence) MIT και (LicenceX11), οι οποίες δίνουν στον χρήστη απεριόριστα δικαιώματα χρήσης, αντιγραφής, τροποποίησης και διανομής (συμπεριλαμβανομένου και της πώλησης) του αρχικού ή του τροποποιημένου προγράμματος (Krol&Szomorova, 2015).

Σύμφωνα με τον Venkat (2009), η JQuery είναι μια μικρή βιβλιοθήκη που βασίζεται σε μερικές πολύ απλές και διαισθητικές αρχές. Παρέχει αρκετά βοηθητικά προγράμματα για την επίτευξη εργασίας σε περιβάλλοντα περιήγησης. Επιτυγχάνει την σωστή ισορροπία ανάμεσα στο μέγεθος, το σύνολο χαρακτηριστικών και την ευκολία χρήσης. Κατά τον Venkat (2009), έχει την δυνατότητα να επιφέρει αλλαγή στον προγραμματισμό στην πλευρά του πελάτη (client-side) με ένα σύνθετο UI.

Κατά τους Krol και Szomorova (2015), οι David (2011), McFarland (2012) και Li&Li (2013) πιστεύουν ότι με την χρήση της συγκεκριμένης βιβλιοθήκης μπορεί κανείς να βελτιώσει την διαδικασία δημιουργίας ιστότοπων. Μπορεί επιπλέον να βελτιωθεί ο τρόπος που φαίνεται η ιστοσελίδα με την ενίσχυση της διαδραστικότητας και της δυναμικής μορφής που το περιεχόμενο παρουσιάζεται.

Πλεονεκτήματα της συγκεκριμένης βιβλιοθήκης είναι ότι λόγο της παλαιότητας της, έχει την μεγαλύτερη υποστήριξη από την προγραμματιστική κοινότητα. Ένα βασικό χαρακτηριστικό της είναι η επιλογή στοιχείων αλλά και η ευκολία διαχείρισης του DOM σε διάφορους browsers, το οποίο βασίζεται στην σχεδίαση της. Επιπλέον μειώνει τον χρόνο και τις γραμμές του κώδικα για την χρήση του AJAX. Βασικό πλεονέκτημα επίσης είναι οι ενσωματωμένες μεθόδους που περιέχει για την χρήση βασικών animations.

Προτρέπει στον διαχωρισμό μεταξύ της Javascript και της HTML για τους προγραμματιστές. Είναι εύκολα επεκτάσιμη καθώς νέα στοιχεία, μέθοδοι και γεγονότα μπορούν εύκολα να προστεθούν και να ξαναχρησιμοποιηθούν. Προωθεί την σαφήνεια και την συντομία με χαρακτηριστικά όπως είναι οι αλυσιδωτές λειτουργίες. Τέλος, διαχειρίζεται τις ασυμβατότητες μεταξύ των προγραμμάτων περιήγησης παρέχοντας έτσι μια διεπαφή συνεπής που λειτουργεί σε διαφορετικά προγράμματα.

Από την άλλη πλευρά ένα βασικό μειονέκτημα είναι η δυσκολία επέκτασης του κώδικα, κυρίως όταν δεν ακολουθείται κάποιο συγκεκριμένο design pattern. Η Jquery χρειάζεται αρκετά εξελιγμένα μοτίβα για να μπορέσει να χρησιμοποιηθεί σε εφαρμογές μεγάλου μεγέθους.

Ακολουθεί ένα παράδειγμα της βασικής σύνταξης μια συνάρτησης της βιβλιοθήκης. Πάνω αριστερά φαίνεται ο κώδικας της HTML, ακριβώς από κάτω παρατηρείται μια συνάρτηση της JQuery, όπου εμφανίζει την οθόνη του χρήστης το μήνυμα “HelloWorld!”, το οποίο και παρουσιάζεται στην Εικόνα 1 κάτω δεξιά.

Εικόνα 1: Βασική σύνταξη της JQuery.

2.3.2 React

Σύμφωνα με τον Cory Gackenheimer (2015), η «React» δημιουργήθηκε από τον JordanWalke έναν μηχανικό λογισμικού του Facebook. Επηρεάστηκε από ένα πλαίσιο της γλώσσας προγραμματισμού PHP. Χρησιμοποιήθηκε αρχικά στο Facebook το 2011 και αργότερα στο instagram.com το 2012. Είναι μια βιβλιοθήκη ανοιχτού λογισμικού, η οποία ακολουθεί το παράδειγμα του δηλωτικού προγραμματισμού (declarativeprogramming), κατά τον οποίο το επιθυμητό αποτέλεσμα ενός προγράμματος υπολογίζεται μέσω της περιγραφής των επιθυμητών ιδιοτήτων του (https://en.wikipedia.org/wiki/React_(Javascript_library)). Συντηρείται από το Facebook, το Instagram και μια κοινότητα από προγραμματιστές. Από τον Μάρτιο του

2016 τα δυο κορυφαία ανοιχτού λογισμικού projects του Facebook είναι η React και η React Native.

Στην React δηλώνεται ο επιθυμητός τρόπος εμφάνισης της εφαρμογής. Η βιβλιοθήκη είναι υπεύθυνη για να την ανανέωση και το σχεδιασμό της εφαρμογής μας από την αρχή, στο DOM καθώς τα δεδομένα αλλάζουν. Χρησιμοποιώντας συναρτήσεις, οι οποίες περιγράφουν μικρά κομμάτια της εφαρμογή, κερδίζεται την δυνατότητα της σύνθεσης (composition), η οποία βοηθάει στην επαναχρησιμοποίηση και ελάττωση των γραμμών του κώδικα που χρειάζεται να γραφεί, καθώς παράλληλα και στην ελάττωση των προβλημάτων που θα μπορούσαν να προκύψουν.

Η απλότητα καθώς και η ταχύτητα της React είναι δυο από τους πιο σημαντικούς παράγοντες που την έχουν καταστήσει ως μια από τις πιο δημοφιλείς βιβλιοθήκες της Javascript, με μεγαλύτερο όφελος όμως να είναι η επαναχρησιμοποίηση των components καθώς και η δυνατότητα της χρήσης της ίδιας βιβλιοθήκης για την δημιουργία εφαρμογών Android, iOS, macOs, linux, Windows (https://facebook.github.io/react/).

Τα components της React είναι συνήθως γραμμένα σε JSX, η οποία στην σύνταξή της μοιάζει με την HTML. Η JSX είναι μια επέκταση της Javascript που επιτρέπει την παραπομπή της HTML καθώς και την χρήσης σύνταξης ετικετών HTML για την απόδοση των υποσυστημάτων (https://facebook.github.io/react/docs/jsx-in-depth.html). Η σύνταξη της HTML μετατρέπεται σε κλήσεις της Javascript από το Framework της React, επίσης οι προγραμματιστές μπορούν μα γράψουν σε καθαρή Javascript. Παρέχει για τα δεδομένα που παρέχονται ως HTML ένα οπτικό πεδίο, το οποίο αποδίδεται τυπικά μέσω της χρήσης στοιχείων που περιέχουν επιπλέον στοιχεία τα οποία ορίζονται ως ειδικές ετικέτες HTML.

Πλεονεκτήματα αυτής της βιβλιοθήκης είναι η εύκολη σχεδίαση της διεπαφής αλλά και η εκμάθηση του API. Υπάρχουν σημαντικά κέρδη απόδοσης έναντι των άλλων JavascriptFrameworks καθώς και η ενημέρωση της βιβλιοθήκης γίνεται ταχύτερα. Γίνεται αποδοτικότερη επαναχρησιμοποίηση του κώδικα καθώς και αποσφαλμάτωση.

Μολαταύτα αρνητικά της React είναι ότι σαν βιβλιοθήκη έχει αρκετά περίπλοκο επίπεδο παρουσίασης (viewlayer). Επιπλέον αρκετοί προγραμματιστές είναι δυσαρεστημένη με την JSX. Τέλος, η ενσωμάτωση της React σε ένα παραδοσιακό MVCFramework απαιτεί αρκετές διαμορφώσεις στον κώδικα για να επιτευχτεί ο σκοπός αυτός.

Εν τέλει, ακολουθεί στην Εικόνα 2 όπου και παρουσιάζει την βασική σύνταξη μιας κλάσης του συγκεκριμένου Framework. Το παράδειγμα δείχνει το πώς η React μπορεί να χρησιμοποιηθεί στην HTML με την χρήση JSX. Η κλάση “Greeter” είναι ένα component με μια ιδιότητα, η μέθοδος “ReactDOM.render” δημιουργεί ένα στιγμιότυπο της κλάσης θέτοντας την ιδιότητα της σε “HelloWorld” και εισάγοντας το component σαν στοιχείο-παιδί του DOM με id “myReactApp”. Το αποτέλεσμα του κώδικα αυτού όταν εμφανίζεται στην ιστοσελίδα θα είναι η Εικόνα 3.

```
<div id="myReactApp"></div>

<script type="text/babel">
  class Greeter extends React.Component {
    render() {
      return <h1>{this.props.greeting}</h1>
    }
  }

  ReactDOM.render(<Greeter greeting="Hello World" />, document.getElementById('myReactApp'));
</script>
```

Εικόνα 2: Βασική σύνταξη της React

```
<div id="myReactApp">
  <h1>Hello World</h1>
</div>
```

Εικόνα 3: Αποτέλεσμα του κώδικα, της εικόνας 2

2.3.3 AngularJS

Αρχικά το AngularJS αναπτύχθηκε το 2009 από τον MiskoHevery ως λογισμικό πίσω από μια διαδικτυακή υπηρεσία αποθήκευσης JSON η οποία θα χρεωνόταν από τα megabytesγια την εύκολη δημιουργία εφαρμογών σε μια επιχείρηση. Αρχικά αυτό το εγχείρημα βρισκόταν στον ιστότοπο “GetAngular.com” και είχε λίγους συνδρομητές

προτού παρθεί η απόφαση να εγκαταλειφτεί αυτή η επιχειρηματική ιδέα και να δοθεί η Angular ως ανοιχτό λογισμικό (https://en.wikipedia.org/wiki/AngularJS).

Ένα σημαντικό στοιχείο για την AngularJS το οποίο χρήζει ιδιαίτερης σημασίας είναι ότι έχει δημιουργηθεί αλλά και συντηρείται από έμπειρους μηχανικούς της Google σε σχέση με άλλα Frameworks τα οποία έχουν δημιουργηθεί από προγραμματιστές ανοιχτού λογισμικού κατά τους Jainetal. Αποτέλεσμα αυτού είναι η ύπαρξη μεγάλης κοινότητας μέσω της οποίας μπορεί να μάθει κάποιος αλλά και να λάβει απαντήσεις σε απορίες σχετικά με την Angular από έμπειρους μηχανικούς λογισμικού.

Σύμφωνα με τους Jainetal, η χρήση της AngularJS επιτρέπει την επέκταση του λεξιλογίου της HTML σε μια πιο εκφραστική και ευανάγνωστη μορφή βοηθώντας τον προγραμματιστή να γράψει πιο καθαρά και αποτελεσματικά κώδικα. Το περιβάλλον που προκύπτει είναι εξαιρετικά εκφραστικό, ευανάγνωστο και γρήγορο για ανάπτυξη.

Πολλά πλαίσια και βιβλιοθήκες προσπαθούν να αντιμετωπίσουν τις αδυναμίες της HTML είτε μέσω κάποιας αφαιρετικής τεχνικής, η οποία απαλείφει την HTML, την CSS ή την Javascript, είτε παρέχοντας έναν επιτακτικό τρόπο χειρισμού του DOM. Καμία από αυτές όμως δεν αντιμετωπίζει το ριζικό πρόβλημα της HTML η οποία δεν σχεδιάστηκε για δυναμικές προβολές αλλά για στατικές ιστοσελίδες με κύριο περιεχόμενο το απλό κείμενο.

Το AngularJS είναι ένα σύνολο από εργαλεία για την κατασκευή του πιο κατάλληλου πλαισίου ή μοτίβου για την ανάπτυξη των εφαρμογών. Είναι πλήρως επεκτάσιμο και μπορεί να συνδυαστεί και να συνεργαστεί εύκολα με άλλες βιβλιοθήκες. Κάθε δυνατότητα που παρέχει μπορεί να τροποποιηθεί ή να αντικατασταθεί ώστε να ταιριάζει στις μοναδικές ανάγκες της κάθε εφαρμογής.

Είναι ένα πλαίσιο εφαρμογών ανοιχτού κώδικα με την αρχιτεκτονική Model-View-Controller (MVC) (Angular 1) και την αρχιτεκτονική Model-View-ViewModel (MVVM) (Angular 2) που υποστηρίζεται από την Google. Η αρχιτεκτονική MVC παρουσιάστηκε το 1970 και έγινε γρήγορα δημοφιλής σε εφαρμογές ηλεκτρονικών υπολογιστών κατά τους Brad Green και Shyam Seshadri. Η βασική ιδέα πίσω από το MVC είναι ο διαχωρισμός του κώδικα για την διαχείριση των δεδομένων (datamodel), την λογική της εφαρμογής (controller) και την παρουσίαση των δεδομένων στον χρήστη (view). Ο χρήστης κατά την αλληλεπίδρασή του με την εφαρμογή βλέπει τα δεδομένα μέσω των controllers.

Το Framework αυτό λύνει το πρόβλημα της ανάπτυξης εφαρμογών μιας σελίδα (Single Page Applications - SPA), επεκτείνοντας τη λειτουργικότητα της HTML με οδηγίες δίνοντας έμφαση στην γρήγορη εκκίνηση της εφαρμογής. Επιπλέον, κατά την δημιουργία πεδίων εισαγωγής σε ένα έγγραφο HTML, μια ξεχωριστή σύνδεση δεδομένων δημιουργείται για κάθε πεδίο που παρέχεται. Η Angular τείνει να ελέγχει κάθε δεσμευμένο πεδίο στη σελίδα για τυχόν αλλαγές πριν από την εκ νέου απόδοση του UI.

Ένα ακόμα στοιχείο άξιο αναφοράς του συγκεκριμένου Framework είναι ότι παρέχει ισχυρές λύσεις για τη δημιουργία προτύπων (template). Χρησιμοποιεί δεσμευτικές εκφράσεις σε HTML attributes για να οδηγήσει τη λειτουργικότητα του προτύπου. Η Angular έχει μια βαθιά κατανόηση του DOM και τα καλά δομημένα πρότυπα του μειώνουν το συνολικό ποσό του κώδικα που απαιτείται για τη δημιουργία της σελίδας που προκύπτει.

HAngularJS είναι βασισμένη στην πεποίθηση ότι ο δηλωτικός προγραμματισμός θα ήταν καλό να χρησιμοποιείται για την δημιουργία των διεπαφών του χρήστη αλλά και την σύνδεση εξαρτημάτων λογισμικού. Αντίθετα, ο επιτακτικής ανάγκης προγραμματισμός είναι καλύτερα να χρησιμοποιείται για τον καθορισμό της επιχειρηματικής λογικής σε μια εφαρμογή. Επεκτείνει και προσαρμόζει την HTML για να παρουσιάσει το περιεχόμενο δυναμικά μέσω μίας σύνδεσης δεδομένων, η οποία είναι αμφίδρομη και επιτρέπει τον αυτόματο συγχρονισμό των μοντέλων (models) και των οπτικών πεδίων (views).

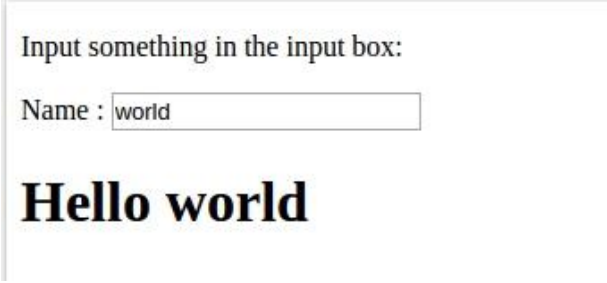
Πλεονεκτήματα αυτού του Framework είναι η δημιουργία παραμετροποιήσεων μοντέλων για το DOM (Document Object Model) καθώς και ο ευθύς σχεδιασμός και η μεταβολή του γραφικού περιβάλλοντος. Ο έλεγχος του κώδικα γίνεται εύκολα όπως και η επαναχρησιμοποίηση του. Το Framework διευκολύνει την επέκταση της σύνταξης HTML και δημιουργεί επαναχρησιμοποιήσιμα components. Η μοντελοποίηση δεδομένων περιορίζεται στη χρήση μικρών μοντέλων δεδομένων προκειμένου να γίνει ο κώδικας απλός και εύκολος στη δοκιμή.

Από την άλλη πλευρά όπως κάθε Framework ή βιβλιοθήκη, έχει και τα μειονεκτήματα του. Προκύπτουν ανησυχίες για την απόδοση μιας εφαρμογής λόγω των πολλών στοιχείων DOM. Για σελίδες με πολλά διαδραστικά στοιχεία, η Angular μπορεί να γίνει αργή. Εντοπίζεται μια πολυπλοκότητα των οδηγιών του API. Επιπλέον, η ενσωμάτωση ξένων βιβλιοθηκών είναι μια πολύπλοκη διαδικασία. Τέλος, η χρήση της ενώ είναι εύκολη, υπάρχει δυσκολία στον εντοπισμό των σφαλμάτων.

Στην Εικόνα 4, φαίνεται ένα παράδειγμα από το συγκεκριμένο Framework. Παρατηρούμε την συγχώνευση της εισαγωγής του πεδίου “Name” με την λέξη “Hello”, ενώ στην Εικόνα 5, φαίνεται το αποτέλεσμα του κώδικα.

```
<!DOCTYPE html>
<html>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.4/angular.min.js"></script>
  <body>

  <div ng-app="">

    <p>Input something in the input box:</p>
    <p>Name : <input type="text" ng-model="name" placeholder="Enter name here"></p>
    <h1>Hello {{name}}</h1>

  </div>

  </body>
</html>
```

Εικόνα 4: Βασική σύνταξη της Angular



Εικόνα 5: Output του κώδικα του Angular

Στη συνέχεια ακολουθούν δυο πίνακες με τα στοιχεία των Frameworks που αναλύθηκαν πρωτύτερα. Ο πρώτος πίνακας περιλαμβάνει πληροφορίες για το έτος δημιουργίας, καθώς και το όνομα του δημιουργού για το κάθε Framework-library, την επίσημη σελίδα του, καθώς και η ιστοσελίδα στην οποία βρίσκετε διαθέσιμο στο github (repository). Ο δεύτερος πίνακας περιέχει τις ιστοσελίδες των bugs, τα οποία αναφέρονται στο κάθε Framework-library και ταυτόχρονα προτείνεται η λύση αυτών. Επιπλέον, αναφέρεται η άδεια με την οποία το Framework-library έχει γίνει ανοιχτού λογισμικού.

Framework	Founded	Official Page	Repository
Angular	Founded in 2009 by Miško Hevery	https://Angularjs.org/	https://github.com/Angular/Angular.js
React	Created by Jordan 2006	https://reactjs.net/	https://github.com/facebook/react
JQuery	Authored by John Risig in 2006	https://JQuery.com/	github.com/JQuery/JQuery

Πίνακας 1: Πληροφορίες ίδρυσης και επίσημη σελίδα του κάθε Framework-library

Framework	Bug report	Licenses
Angular	https://github.com/Angular/Angular.js/issues	MIT
React	https://github.com/facebook/react/issues	BSD
JQuery	https://github.com/JQuery/JQuery/issues	MIT

Πίνακας 2: Ιστοσελίδα των bugs και άδεια χρήσης.

2.4 Σύγκριση της βιβλιοθήκης JQuery και Framework Angular

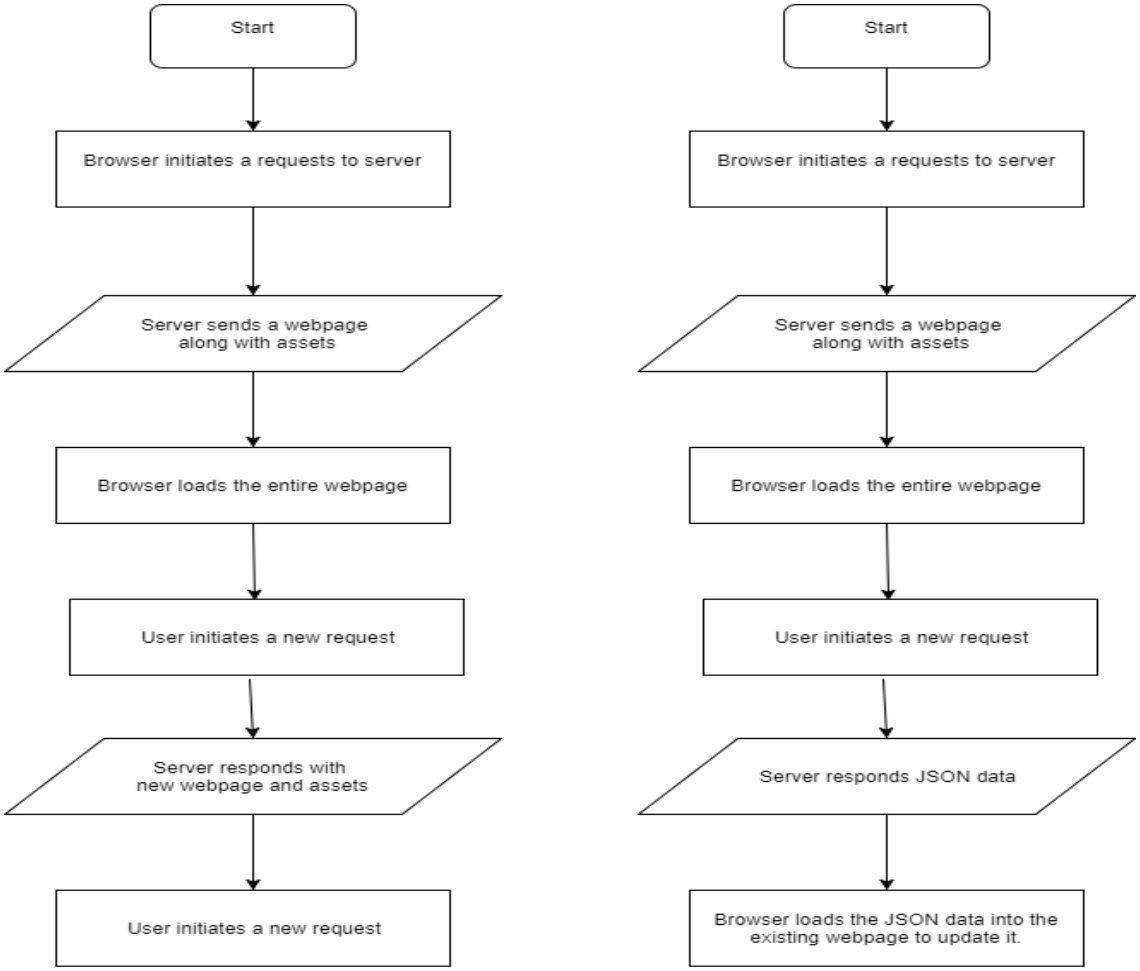
Σύμφωνα με τον NavnathN. Chavare και A. V. Nadargi (2016), το Framework AngularJS είναι πιο ασφαλές, πιο αποδοτικό και συγχρόνως πιο γρήγορο σε σχέση με την βιβλιοθήκη της JQuery.

Η δημιουργία μιας web εφαρμογής, με χρήση της βιβλιοθήκης JQuery διαφέρει κατά πολύ στον τρόπο υλοποίησής της, καθώς επίσης και στον τρόπο επικοινωνίας με τον διακομιστή, σε σύγκριση με το Framework Angular. Η επικοινωνία της βιβλιοθήκη JQuery στον server γίνεται ως εξής:

- Ο Φυλλομετρητής εκτελεί ένα request.
- Ο διακομιστής στέλνει μια ιστοσελίδα μαζί με τα στοιχεία (assets) (HTML και Javascript).
- Ο περιηγητής φορτώνει ολόκληρη την ιστοσελίδα.
- Ο χρήστης κάνει κλικ σε έναν σύνδεσμο (εκκινεί ένα νέο request).
- Ο διακομιστής ανταποκρίνεται με νέα ιστοσελίδα και στοιχεία (assets) απο την αρχή.
- Ο περιηγητής φορτώνει ξανά ολόκληρη την ιστοσελίδα.

Επειδή η παραπάνω διαδικασία είναι χρονοβόρα και συγχρόνως μη αποδοτική, εισήχθη η έννοια SPA(Single Page Application) μέσω από το Framework Angular. Η επικοινωνία με χρήση το Framework Angular γίνεται με την παρακάτω σειρά:

- Ο Φυλλομετρητής εκτελεί ένα request.
- Ο διακομιστής στέλνει μια ιστοσελίδα μαζί με τα στοιχεία (assets) (HTML και Javascript).
- Ο περιηγητής φορτώνει ολόκληρη την ιστοσελίδα.
- Ο χρήστης κάνοντας «κλικ» σε ένα σύνδεσμο, στέλνει ένα νέο request και ο server απαντάει με ένα json αρχείο. Η διαδικασία αυτή γίνεται χωρίς να χρειάζεται ξανά φόρτωση των στοιχείων assets, με αποτέλεσμα ο χρήστης να εξοικονομεί πολύτιμο χρόνο.



Εικόνα 6: Διαφορές Jquery -Angular

2.4.1 Μέγεθος του κώδικα

Στο στιγμιότυπο (εικόνα 6) παρατηρείται η υλοποίηση δύο όμοιων εφαρμογών αλλά με διαφορετικό τρόπο κατασκευής. Αριστερά, η εφαρμογή σχεδιάστηκε με την χρήση της βιβλιοθήκης της JQuery, ενώ δεξιά με το Fremawork Angular. Το μέγεθος του κώδικα που δημιουργήθηκε χρησιμοποιώντας το Framework Angular είναι αρκετά μικρότερο σε σύγκριση με την βιβλιοθήκη της Javascript JQuery. Αυτό οφείλεται στις προηγμένες μεθόδους που ενσωματώνει το Framework AngularJS για να αποφευχθεί η πολυπλοκότητα DOM.

Εικόνα 7: Διαφορές Κώδικα Jquery - Angular

2.4.2 Χρόνος φόρτωσης Ιστοσελίδας

Ο χρόνος φόρτωσης μια ιστοσελίδας χρησιμοποιώντας το Framework Angular μειώνεται κατά πολύ. Ο επισκέπτης φορτώνει την ιστοσελίδα μία φορά και στην συνέχεια μπορεί να περιηγηθεί σε όλη την ιστοσελίδα δίχως να ανανεώσει ο browjser.

2.4.3 Ασφάλεια

Το AngularJS αποκρύπτει την πραγματική δομή της διεύθυνσης URL της σελίδας στο browsersε σχέση με την βιβλιοθήκη της JQuery, έτσι ώστε ο επισκέπτης δεν μπορεί να εισβάλει στο σύστημα. Συνεπώς, αυξάνεται η ασφάλεια της ιστοσελίδας.

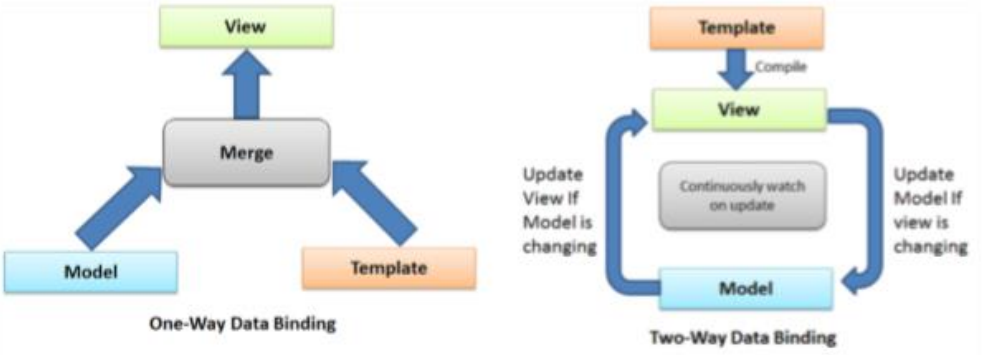
2.4.4 Two-wayBinding

Η σύνδεση δεδομένων στο AngularJS είναι ο συγχρονισμός μεταξύ του model και view. Όταν αλλάζουν δεδομένα στο model, η προβολή «αντανακλά» την αλλαγή και όταν τα δεδομένα στην προβολή αλλάζουν, το μοντέλο ενημερώνεται επίσης. Αυτό συμβαίνει αμέσως και αυτόματα, γεγονός που διασφαλίζει ότι το μοντέλο και η προβολή ενημερώνονται ανά πάσα στιγμή.

Name: Foo bar

Foo bar

Εικόνα 8: Two-way Binding



Εικόνα 9: One way &Two-way Data Binding

Κεφάλαιο 3 : Μεθοδολογία

3.1 Εισαγωγή

Στα πλαίσια των αναγκών της έρευνας θα υλοποιηθεί μια πρωτότυπη εφαρμογή με δύο τρόπους. Αρχικά, η εφαρμογή αυτή θα υλοποιηθεί με χρήση της βιβλιοθήκης της Javascript «JQuery» και στην συνεχεία με το Framework Angular.Τα ονόματα που θα δοθούν στις εφαρμογές θα είναι «Framework» και «library» αντίστοιχα. Ακολούθως, θα γίνει η σύγκριση των δύο εφαρμογών αυτών. Ως μέτρο σύγκρισης θα χρησιμοποιηθεί η καταμέτρηση γραμμών της κάθε εφαρμογής, ο χρόνος φόρτισης ιστοσελίδας, η ασφάλεια της εφαρμογής καθώς επίσης και η κατανάλωση χώρου στον διακομιστή.

3.2 Μέτρα σύγκρισης εφαρμογής

Αριθμός γραμμών κώδικα: Η καταμέτρηση των γραμμών του κώδικα της κάθε εφαρμογής θα γίνει με την βοήθεια του προγράμματος atom.

Χρόνος φόρτωσης ιστοσελίδας: Ο χρόνος φόρτωσης της ιστοσελίδας θα πραγματοποιηθεί με την βοήθεια ενός extension, του φυλλομετρητή Chrome.Το όνομα του extension είναι pageloadtime και βρίσκεται δωρεάν στο διαδίκτυο.

Ασφάλεια: Κατά το πόσο είναι ασφαλής μια εφαρμογή θα προκύψει από το url που θα δημιουργηθεί κατά την εκτέλεση ενός request.

Χώρος κατανάλωσης: Η κατανάλωση του χώρου της κάθε εφαρμογής μπορεί να ελεγχθεί μέσα από τις ιδιότητες του φακέλου στον διακομιστή.

3.3 Έρευνα και μελέτη

Στο Κεφάλαιο 2 έγινε αναφορά για το τι είναι Framework και τι βιβλιοθήκη. Γίνεται ανάλυση των δύο πιο δημοφιλώνJavascriptFramework (React και Angular)και της βιβλιοθήκης Javascript. Παρουσιάστηκαν και διατυπώθηκαν τα πλεονεκτήματα και τα μειονεκτήματα της κάθε βιβλιοθήκης και του κάθε Framework. Στο τέλος, έγινε σύγκριση της βιβλιοθήκης JQuery με το FrameworkAngular.

Σύμφωνα με μία πρόχειρη έρευνα που έγινε με την συμμετοχή του επιβλέποντα καθηγητή κ. Πρωτόγερο Νικόλαο του πανεπιστημίου Μακεδονίας και συγκεκριμένα του τμήματος Πληροφοριακών Συστημάτων, κάθε καθηγητής, κάθε χρόνο έχει κατά μέσο όρο περίπου 6 φοιτητές υπό την εποπτεία του σχετικά με την εκπόνηση της διπλωματικής εργασίας. Αυτό αναλύεται σε 12 επισκέψεις τον μήνα ενώ μπορεί η επίσκεψη να πραγματοποιηθεί μέσω ηλεκτρονικού ταχυδρομείου (email) ή τηλεδιασκέψεων, οι οποίες είναι περισσότερες από τις συναντήσεις. Εξαιτίας του φόρτου εργασίας τόσο στο πρόγραμμα του φοιτητή (απαιτήσεις-υποχρεώσεις με χρονικό περιορισμό) όσο και του επιβλέποντα καθηγητή, πολλές φορές οι συναντήσεις δεν πραγματοποιούνται λόγω έλλειψης υπενθύμισης.

Επίσης, λόγω των μη οργανωμένων συναντήσεων, παρουσιάζεται ως συχνό φαινόμενο η μη εποικοδομητική συνάντηση με μια συνεχή και άσκοπη επανάληψη και ανάλυση ίδιων θεμάτων , που έχει ως αποτέλεσμα η εξέλιξη της διπλωματικής εργασίας να παραμένει στάσιμη. Έχοντας αυτά ως βάση, θα δημιουργηθεί μια πλήρης τρίτη εφαρμογή, με τεχνολογία σχεδίασης που θα προκύψει στηριζόμενη στο αποτέλεσμα της σύγκρισης μεταξύ των δυο εφαρμογών «Framework» και «library». Στόχος της εφαρμογής αυτής είναι να καλύψει τις προαναφερόμενες ανάγκες, καθώς επίσης να αναπτύξει μία νέα σχέση μεταξύ καθηγητή και φοιτητή, η οποία θα είναι πιο οργανωτική πιο φιλική και συνεπώς πιο αποδοτική.

3.4 Διαδικασία αξιολόγησης

Η αξιολόγηση της εφαρμογής θα γίνει μέσω κάποιων βασικών κριτηρίων. Τα κριτήρια στα οποία θα βασιστεί η εφαρμογή είναι κατά πόσο είναι εύχρηστη – προσβάσιμη , επεκτάσιμη και λειτουργική στον χρήστη. Αναλυτικότερα:

Εύχρηστη - προσβάσιμη :Θα δίνεται στον χρήστη η δυνατότητα για περιήγηση στην εφαρμογή μέσω οποιαδήποτε ηλεκτρονικής συσκευής (κινητό τηλέφωνο, τάμπλετ ή ηλεκτρονικό υπολογιστή)**,** καθώς θα έχει σχεδιαστεί με τεχνολογία responsive.

Επεκτάσιμη: Αν η εφαρμογή έχει σχεδιαστεί σωστά αρχιτεκτονικά, δεν θα αντιμετωπίσει ενδεχόμενα προβλήματα σχετικά με την επέκταση και την βελτιστοποίησή της.

Λειτουργική: Θα είναι λειτουργική καθώς δημιουργεί μία άμεση επικοινωνία μεταξύ καθηγητή – φοιτητή.

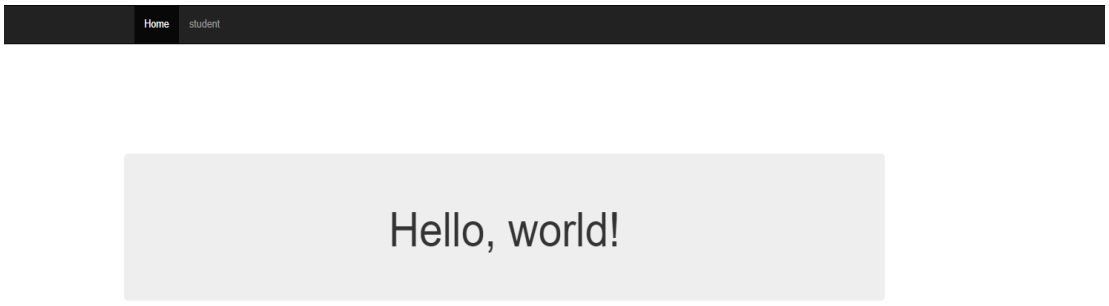
Κεφάλαιο 4 : Αποτελέσματα υλοποίησης

4.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα γίνει η παρουσίαση των δύο εφαρμογών που αναφέρθηκαν στο κεφάλαιο τρία. Ακολούθως, μετά την σύγκριση μεταξύ αυτών και με βάση τα αποτελέσματα που προέκυψαν, δημιουργείται μία νέα εφαρμογή που θα καλύψει τις ανάγκες φοιτητή - καθηγητή. Οι εφαρμογές θα είναι διαθέσιμες στο διαδίκτυο στους συνδέσμους http://mitronatsios.gr/library και http://mitronatsios.gr/framework. Τα ονόματα των εφαρμογών είναι «Library» και «Framework» αντίστοιχα. Για τις ανάγκες της έρευνας, θα χρησιμοποιηθεί το extension του browserchrome μέτρισης χρόνου φόρτωση ιστότοπου «pageloadtime»της google.

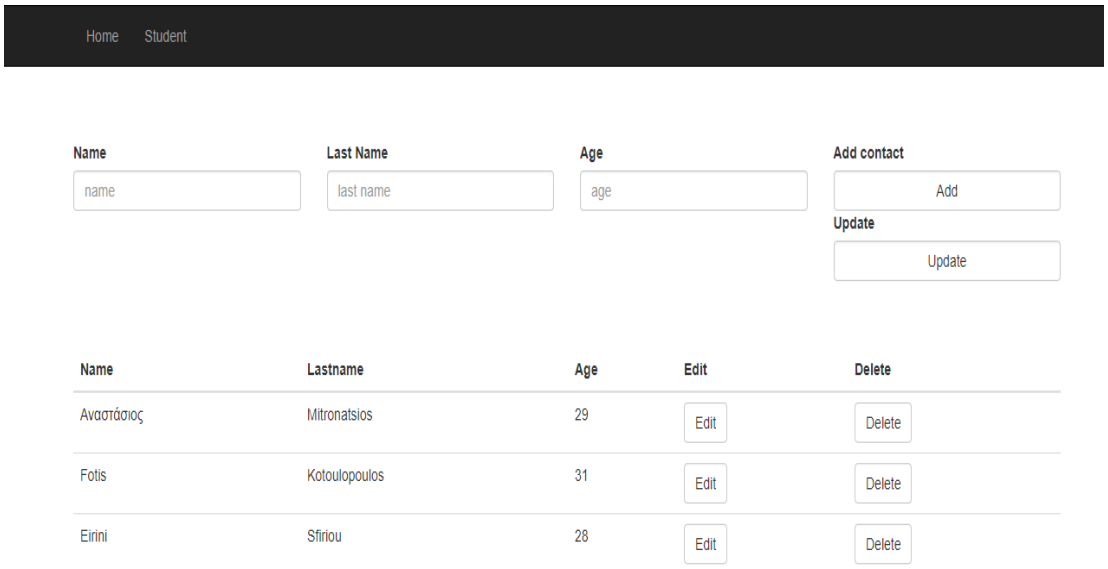
4.2 Παρουσίαση εφαρμογής

Για τις ανάγκες δημιουργίας της εφαρμογής αυτής χρησιμοποιήθηκε για front-end το Framework Bootstrap και η βιβλιοθήκη της Javascript «JQuery» όσο αφορά την εφαρμογή «Library», ενώ για την εφαρμογή «Framework» το Framework το «Angular». Για back-end χρησιμοποιήθηκε η php7 σε συνδυασμό με mysql για την σύνδεση της βάσης. Το αποτέλεσμα όλων των παραπάνω διατυπώνεται στην εικόνα 10.



Εικόνα 10: Αρχική σελίδα της εφαρμογής.

Παρατηρώντας προσεχτικά στο header παρατηρούμε ότι υπάρχει ακόμη μία σελίδα. Η σελίδα «Student», η οποία φαίνεται στην εικόνα 11.

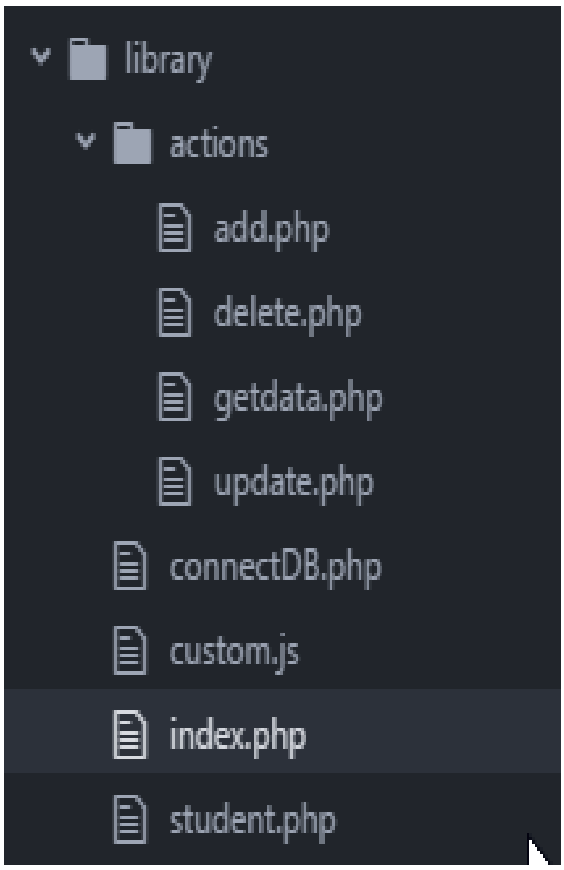


Εικόνα 11: Σελίδα Student

Η σελίδα «Student» είναι υπεύθυνη για την διαχείριση «CRUD» (Create , Read, Update, Delete) μιας εγγραφής. Η εισαγωγή μια εγγραφής γίνεται συμπληρώνοντας τα στοιχεία της φόρμας και πατώντας το κουμπί «Add». Μόλις γίνει καταχώρηση εγγραφής στο σύστημα, αυτόματα γίνεται και ενημέρωση του πίνακα με τα νέα στοιχεία. Σε περίπτωση καταχώρησης λανθασμένων στοιχείων εγγραφής δίνεται η δυνατότητα τροποποίησης της εγγραφής πατώντας το κουμπί «Edit». Αφού έγινε επεξεργασία της εγγραφής, για να ολοκληρωθεί επιτυχώς η ενημέρωσή της με τα νέα δεδομένα, θα πρέπει να πατηθεί το κουμπί «update». Τέλος, σε περίπτωση που δεν χρειαζόμαστε την επαφή αυτή, υπάρχει η δυνατότητα οριστικής διαγραφής πατώντας το κουμπί «delete».

4.2.1 Δομή εφαρμογής «Library»

Σε αυτή την παράγραφο θα αναλυθεί η αρχιτεκτονική δομή που χρησιμοποιεί η εφαρμογή «Library», καθώς επίσης θα αναφερθεί ο τρόπος διαχείρισης των δεδομένων «CRUD».

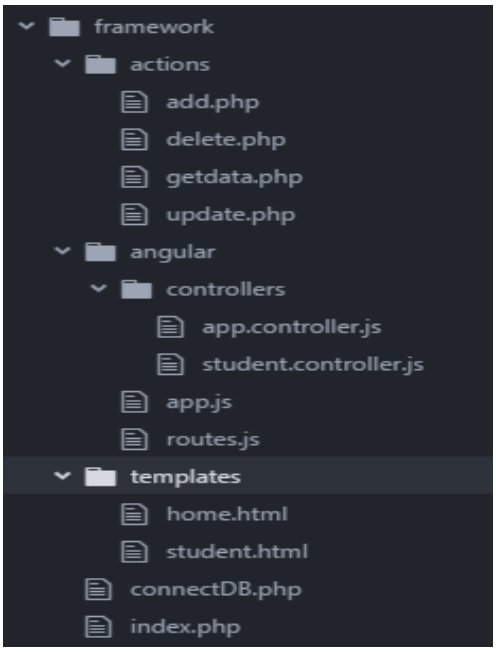


Εικόνα 12: Δομή αρχείων εφαρμογής «Library»

Όπως βλέπουμε στην εικόνα 12, η δομή του αποτελείται από τον φάκελο «actions», στον οποίο περιέχονται κάποια αρχεία. Μπορούμε να κατανοήσουμε τον λόγο χρήσης των αρχείων αυτών, διαβάζοντας μόνο το όνομα του αρχείου. Τα αρχεία αυτά είναι αναγκαία για την σωστή διαχείριση «CRUD». Στην συνέχεια υπάρχει το αρχείο «connectDB.php» που είναι υπεύθυνο για την σύνδεση της βάσης με το σύστημα και ακολουθούν τα αρχεία «custom.js», «index.php» και «student.php». Το αρχείο «custom.js» είναι υπεύθυνο για τα requests που γίνονται στο σύστημα. Τα υπόλοιπα δύο αρχεία είναι οι δύο σελίδες της εφαρμογής, οι οποίες παρουσιάστηκαν στην προηγούμενη παράγραφο. Όσο αφορά την αρχιτεκτονική δομή δεν ακολουθεί κάποιο συγκεκριμένο μοτίβο.

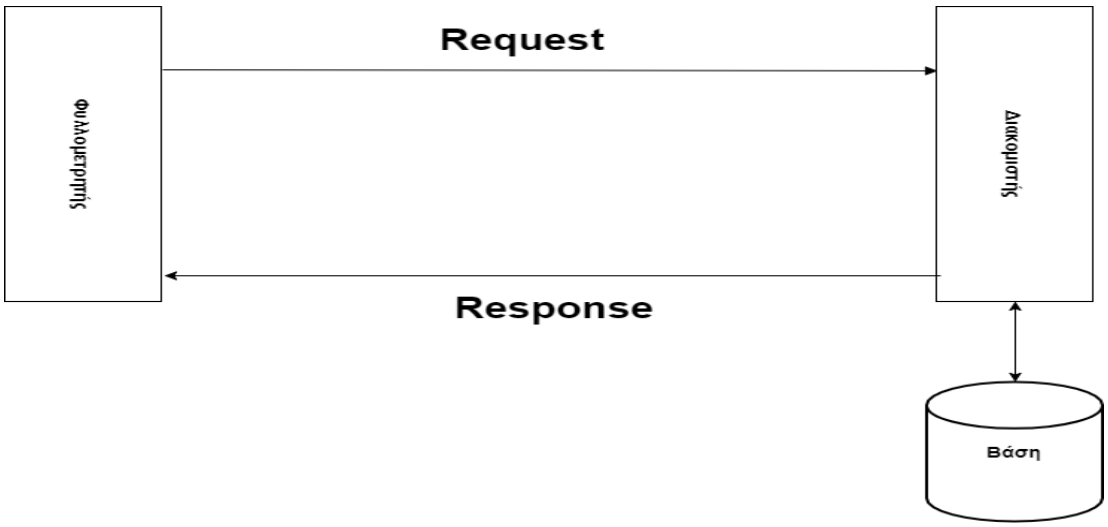
4.2.2 Δομή εφαρμογής «Framework»

Σε αντίθεση με την δομή της εφαρμογής «Library», η δομή σε αυτή την περίπτωση είναι λίγο πιο περίπλοκη. Η συγκεκριμένη εφαρμογή ακολουθεί την αρχιτεκτονική δομή του MVC, η οποία είναι πιο οργανωτική και ταυτόχρονα πιο αποτελεσματική (εικόνα 13).



Εικόνα 13: Δομή αρχείων «Framework»

Η συγκεκριμένη εφαρμογή υλοποιήθηκε με τη χρήση Framework «Angular». Δηλαδή υπάρχει μια καθορισμένη σειρά αλληλουχίας ενεργειών, όπου εκτελούνται για την διαχείριση «CRUD». Για κάθε καρτέλα που υπάρχει, υπάρχει και ένας «Controller». Οπότε όταν ο φυλλομετρητής στείλει κάποιο request, torequest επικοινωνεί με τον «Controller», από εκεί στέλνεται στον server, ο server με την σειρά του κάνει call στην βάση. Το response που προκύπτει το στέλνει πίσω στον φυλλομετρητή με την ίδια σειρά αλλά αντίστροφα όμως τώρα. Το στιγμιότυπο που φαίνεται παρακάτω αναπαριστά όλη την διαδικασία που προαναφέρθηκε προηγουμένως.



Εικόνα 14: Κλήση Request/Response

4.2.3 Σύγκριση «Library» με «Framework»

Στις παραγράφους 4.2.1 και 4.2.2 αναλύθηκε η αρχιτεκτονική δομή της κάθε εφαρμογής. Στην συγκεκριμένη θα γίνει η ανάλυση του κώδικα και ταυτόχρονα η μέτρηση όγκου της εφαρμογής, ο χρόνος φόρτωσης κάθε σελίδας και της μνήμης.

4.2.4 Δημιουργία νέα σελίδας

Οι δύο εφαρμογές ως προς τον τρόπο δημιουργίας μια νέας σελίδα, εμφανίζουν ορισμένες σημαντικές διαφορές. Στην πρώτη εφαρμογή γίνεται επανάληψη του κώδικα σε κάθε σελίδα. Πιο συγκεκριμένα, ο προγραμματιστής θα πρέπει να δηλώνει ξανά τις απαραίτητες βιβλιοθήκες που είναι αναγκαίες για την λειτουργία της νέας σελίδας. Αυτό έχει ως συνέπεια, την κατανάλωση του χώρου τόσο στον server όσο και την καθυστέρηση χρόνου για τον επισκέπτη. Δηλαδή ο φυλλομετρητής πρέπει κάθε φορά να φορτώνει την σελίδα από την αρχή. Στα παρακάτω στιγμιότυπα (εικόνα 15, εικόνα 16) φαίνεται η δήλωση και η επανάληψη του κώδικα στις σελίδες «home» και «student».

```
<!DOCTYPE html>
<html>
<head>
<title>Home</title>
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" >
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
</head>
<body>
  <nav class="navbar navbar-inverse">
    <div class="container">
        <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-9">
          <ul class="nav navbar-nav">
            <li class="active"><a href='http://localhost/application/'>Home</a></li>
            <li><a href='http://localhost/application/student.php'>student</a></li>
          </ul>
        </div>
      </div>
    </nav>
    <div class="container">
</body>
</html>
```

Εικόνα15: Κώδικας της σελίδας Home

```
<!DOCTYPE html>
<html>
<head>
<title>Student</title>
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" >
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
    <script src="../custom.js"></script>
</head>

<body>
  <nav class="navbar navbar-inverse">
    <div class="container">
        <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-9">
          <ul class="nav navbar-nav">
            <li class="active"><a href='http://localhost/application/'>Home</a></li>
            <li><a href='http://localhost/application/studentt.php'>student</a></li>
          </ul>
        </div>
      </div>
    </nav>
  <div class="container">
</body>
</html>
```

Εικόνα 16: Κώδικας της σελίδας Student.

Στην εφαρμογή «Framework» πετύχαμε την αποφυγή επανάληψη κώδικα με την χρήση Framework Angular. Αυτό έχει ως αποτέλεσμα, την μείωση κατανάλωσης χώρου στον server. Η εφαρμογή αυτή μπορεί να μεταφραστεί ως μία σελίδα SPA(Single Page Application). Δηλαδή γίνεται φόρτωση του «header» και του «Footer» μια φορά και το μόνο που αλλάζει είναι το περιεχόμενο της καρτέλας. Στο παρακάτω στιγμιότυπο φαίνεται η δήλωση του κώδικα της σελίδας «Index».

```
<!DOCTYPE html>
<html>
<head>
  <title>Infos</title>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.4/angular.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/angular.js/1.6.4/angular-route.min.js"></script>

  <script src="angular/app.js"></script>
  <script src="angular/routes.js"></script>
  <script src="angular/controllers/app.controller.js"></script>
  <script src="angular/controllers/student.controller.js"></script>
</head>

<body ng-app="diplomaApp" ng-controller="AppController">
  <nav class="navbar navbar-inverse">
    <div class="container">
        <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-9">
          <ul class="nav navbar-nav">
            <li><a href="#!/home">Home</a></li>
            <li><a href="#!/student">Student</a></li>
          </ul>
        </div>
      </div>
    </nav>

    <div class="container">
      <ng-view></ng-view>
    </div>

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" >
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
</body>
</html>
```

Εικόνα 17: Κώδικας της σελίδας Index

Στο παραπάνω στιγμιότυπο (εικόνα 17), στην γραμμή 27 φορτώνεται το περιεχόμενο των καρτελών «Home» και «Student» αντίστοιχα.

4.2.5 Render εγγραφής

Στην εφαρμογή «Library» για να γίνει render μία εγγραφή, θα πρέπει να «τρέξει» το σύστημα τον παρακάτω κώδικα που φαίνεται στο στιγμιότυπο (εικόνα 18).

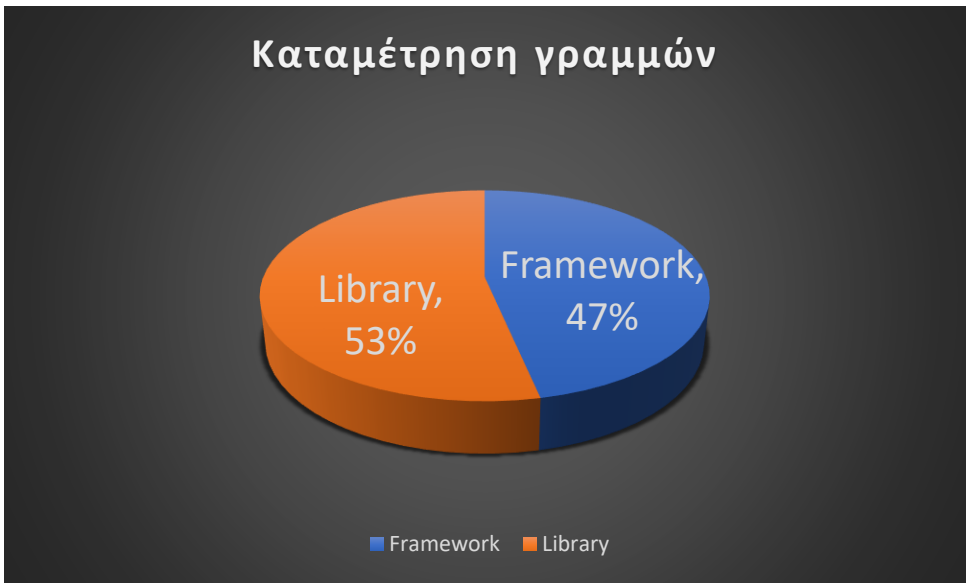
```
$("#list").append("<tr><td class='hidden'>"
+item['id']+"</td><td>"
+item['name']+"</td><td>"
+item['lastname']+"</td><td>"
+item['age']+"</td><td>"
+"<button class='btn btn-default btnSelect'>Select</button>"
+"</td><td>"+"<button onclick=deleterecord("+item['id']+"
 type='button' class='btn btn-default'>Delete</button>"+"</td></tr>");
```

Εικόνα 18: Κώδικας Render εγγραφής «Library»

Σε αντίθεση όμως η εφαρμογή «Framework» για την ίδια ακριβώς απαίτηση «τρέχει» τον παρακάτω κώδικα που φαίνεται στο στιγμιότυπο (εικόνα 19).

```
<tr id="infos" ng-repeat="x in names">
  <td>{{ x.Name }}</td>
  <td>{{ x.LastName }}</td>
  <td>{{ x.age }}</td>
  <td><button ng-click="selectRecord(x)"; type='button' class='btn btn-default'>Edit</button></td>
  <td><button ng-click="deleteRecord(x.id);"; type='button' class='btn btn-default'>Delete</button></td>
</tr>
```

Εικόνα 19: Κώδικας Render εγγραφής «Framework»



Εικόνα 20: Καταμέτρηση γραμμών

Με βάση το παραπάνω γράφημα (εικόνα 20) διαπιστώνουμε ότι, στην καταμέτρηση γραμμών κώδικα ανάμεσα στις δύο εφαρμογές για την συγκεκριμένη ενέργεια, η εφαρμογή «Framework» είναι κατά 1.23 φορές μικρότερη σε σχέση με την εφαρμογή «Library». Επίσης, κοιτάζοντας τον κώδικα ανάμεσα στις δύο εφαρμογές, παρατηρούμε ότι ο κώδικας της εφαρμογής «Framework» εκτός από ότι είναι μικρότερος σε αριθμών γραμμών είναι και πιο κατανοητός στον προγραμματιστή. Αυτό συνεπάγεται με λιγότερο χώρο κατανάλωσης στον server, λιγότερο χρόνο φόρτωσης της σελίδας και συγχρόνως λιγότερο χρόνο στην περίπτωση επέκτασης της εφαρμογής .

4.2.6 Χώρος κατανάλωσης στον server

Ο χώρος που καταλαμβάνει η εφαρμογή «Library» στο server φαίνεται στο παρακάτω στιγμιότυπο (εικόνα 21).

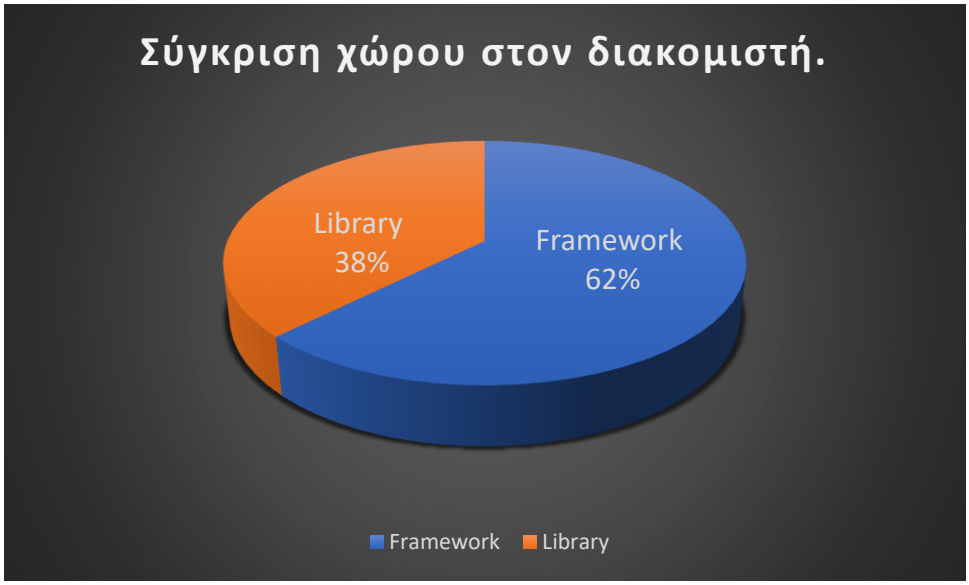
Size:	8.63 KB (8,846 bytes)
Size on disk:	12.0 KB (12,288 bytes)
Contains:	8 Files, 1 Folders

Εικόνα 21: Χωρητικότητα εφαρμογής «Library» στον server.

Ο χώρος που καταλαμβάνει η εφαρμογή «Framework» στο server φαίνεται στο παρακάτω στιγμιότυπο (εικόνα 22).

Size:	8.63 KB (8,843 bytes)
Size on disk:	20.0 KB (20,480 bytes)
Contains:	12 Files, 4 Folders

Εικόνα 22: Χωρητικότητα εφαρμογής «Framework» στον server.

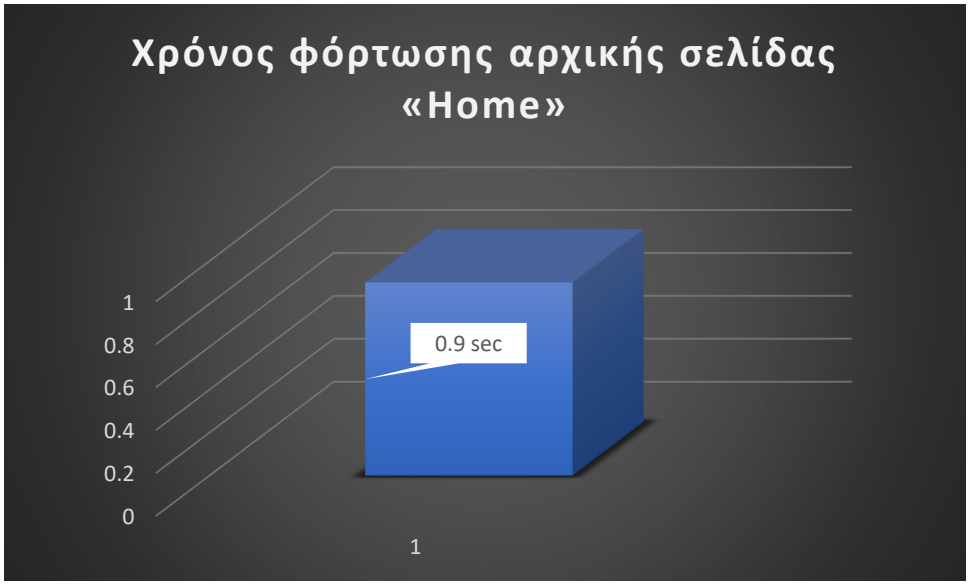


Εικόνα 23: Σύγκριση χώρου στον διακομιστή.

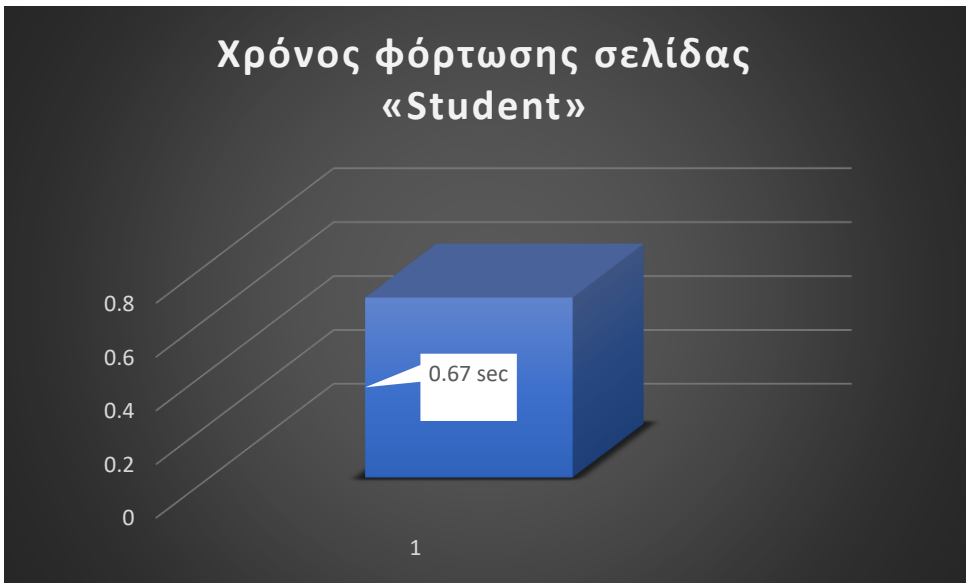
Από το παραπάνω γράφημα (εικόνα 23) προκύπτει, ότι η εφαρμογή «Framework» καταλαμβάνει σχεδόν το διπλάσιο σε όγκο δεδομένων έναντι της εφαρμογής «Library» στον διακομιστή. Αυτό όμως συμβαίνει διότι έχουμε μόνο δύο σελίδες. Σε περίπτωση που ο αριθμός των σελίδων αυξηθεί, στην εφαρμογή «Library» θα έχουμε επανάληψη κώδικα τόσες φορές όσες είναι και ο αριθμός των σελίδων πράγμα που καθιστά αύξηση όγκου δεδομένων στο server. Συνεπώς, το αποτέλεσμα τότε θα είναι αντιστρόφως ανάλογο.

4.2.7 Χρόνος φόρτωσης σελίδας

Ο φυλλομετρητής «φόρτωσε» τις δύο σελίδες σε ξεχωριστό «Tab» στον browser Chrome και κατέγραψε τους παρακάτω χρόνους. Για την εφαρμογή «Library» ο χρόνος που χρειάστηκε για να φορτώσει η σελίδα «Home» και αντίστοιχα για την σελίδα «Student» καταγράφηκαν στα παρακάτω στιγμιότυπα (εικόνα 24, εικόνα 25).

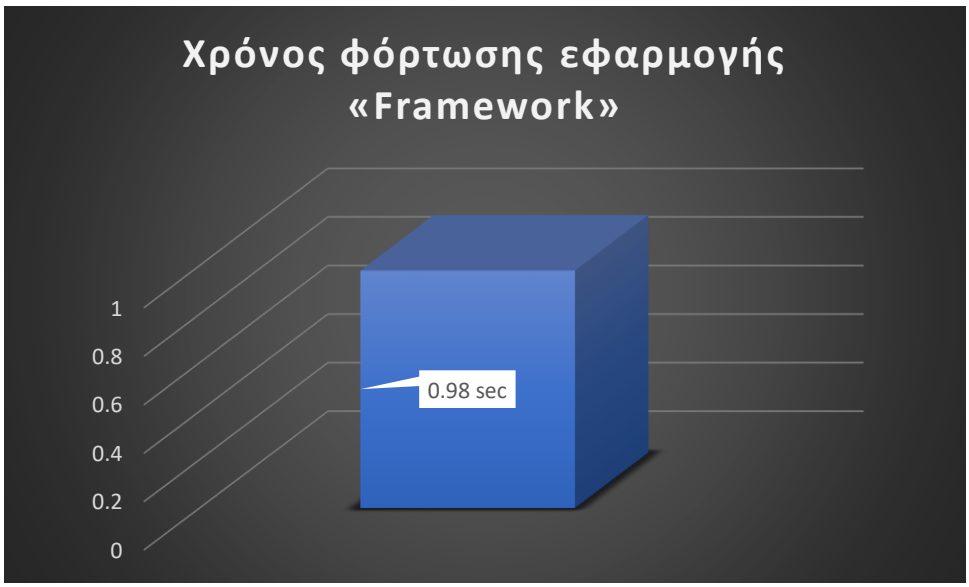


Εικόνα 24: Χρόνος φόρτωσης αρχικής σελίδας «Home» .



Εικόνα 25: Χρόνος φόρτωσης σελίδας «Student»

Αντίστοιχα για την εφαρμογή «Framework» ο χρόνος που χρειάστηκε για να «φορτώσει» η σελίδα home και η σελίδα student είναι ο ίδιος χρόνος διότι στον ίδιο χρόνο «φορτώνονται» παράλληλα και οι υπόλοιπες σελίδες. Ο χρόνος καταγράφηκε στο παρακάτω στιγμιότυπο (εικόνα 26).



Εικόνα 26: Χρόνος φόρτωσης εφαρμογής «Framework»

Εικόνα 27: Σύγκριση χρόνου φόρτωσης Ιστοσελίδας

Στο παραπάνω γράφημα η (εικόνα 27) αναπαριστά τον συνολικό χρόνο φόρτισης ιστοσελίδας της κάθε εφαρμογής. Παρατηρούμε ότι η εφαρμογή «Library» απαιτεί τον διπλάσιο χρόνο φόρτωσης σε σχέση με την εφαρμογή «Framework».

4.3 LetMeKnow

Αφού έγινε ανάλυση και σύγκριση των εφαρμογών Library και Framework αντίστοιχα, επιλέχτηκε ο τρόπος σχεδιασμού της τελικής εφαρμογής. Πιο αναλυτικά, έχοντας ως βάση τα στατιστικά στοιχεία που προέκυψαν στην προηγούμενη ενότητα και πιο συγκεκριμένα στα γραφήματα της καταμέτρησης γραμμών (εικόνα 20), κατανάλωση χώρου στον διακομιστή (εικόνα 23), όπως επίσης και στο γράφημα χρόνο φόρτισης ιστοσελίδας (εικόνα 27), για την σχεδίαση και ανάπτυξη της νέας εφαρμογής LetMeKnow θα χρησιμοποιηθεί το FrameworkAngular.

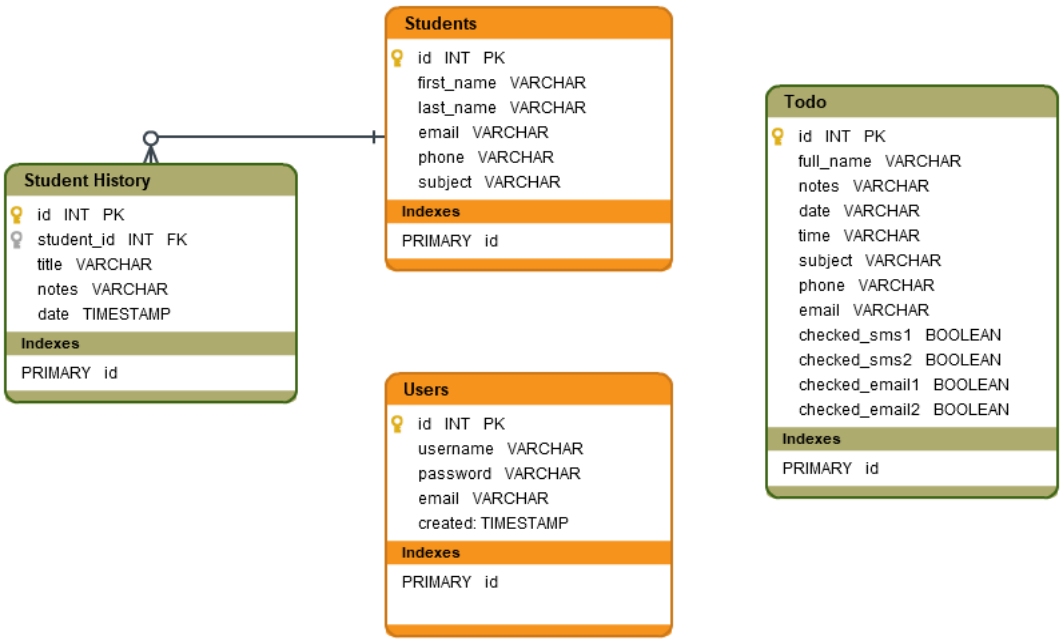
Καθημερινά ο καθηγητής του Πανεπιστημίου δέχεται κλήσεις, emails, επαγγελματικά ραντεβού, ραντεβού φοιτητών και συγχρόνως έχει πολλές ενέργειες σε εκκρεμότητα που πρέπει να γίνουν άμεσα ή μακροπρόθεσμα. Εξαιτίας του φόρτου εργασίας μερικές φορές οι υποχρεώσεις αυτές παραλείπονται με αποτέλεσμα να μην πραγματοποιούνται ποτέ. Ακόμη, υπάρχει το ενδεχόμενο να χαθεί ή να κλαπεί η ατζέντα όπου είναι σημειωμένες οι ημερομηνίες συνάντησης και χρήσιμες πληροφορίες.

Για το λόγο αυτό δημιουργήθηκε η ανάγκη της εφαρμογής LetMeKnow. Η LetMeKnow δημιουργήθηκε ώστε να υπάρχει συνεχής επικοινωνία του διδάσκοντα με τον φοιτητή και το αντίστροφο.

Η έξυπνη διαδικτυακή εφαρμογή υποστηρίζει την ηλεκτρονική καταχώριση των στοιχείων του φοιτητή, το ιστορικό των συναντήσεών του, καθώς και τα “toDO” που αφορούν είτε κάποια συνάντηση είτε κάποια άλλη σημαντική ενέργεια που έχει προγραμματιστεί. Η υπενθύμιση των ενεργειών που είναι προγραμματισμένες να γίνουν στο μέλλον γίνεται μέσω email και ταυτόχρονα με μήνυμα στο κινητό τηλέφωνο και των δύο άμεσα ενδιαφερόμενων. Η εφαρμογή βρίσκεται διαθέσιμη στον ιστότοπο http://mitronatsios.gr/thesis .

4.3.1 Αποθήκευση πληροφορίας

Για την αποθήκευση της πληροφορίας χρησιμοποιήθηκε η βάση της MySQL. Οι πίνακες που χρησιμοποιήθηκαν παρουσιάζονται στην εικόνα 28.



Εικόνα 28: Σχήμα βάσης δεδομένων

- Students: Στον πίνακα αυτόν θα αποθηκεύονται τα προσωπικά στοιχεία του κάθε φοιτητή. Τα πεδία είναι υποχρεωτικά και είναι id, first_name , last_name , email, phone, subject. Πρωτεύον κλειδί του πίνακα αυτού είναι το πεδίο «id» τύπου autoincrement και λαμβάνει σειριακή τιμή κατά την δημιουργίας εγγραφής.

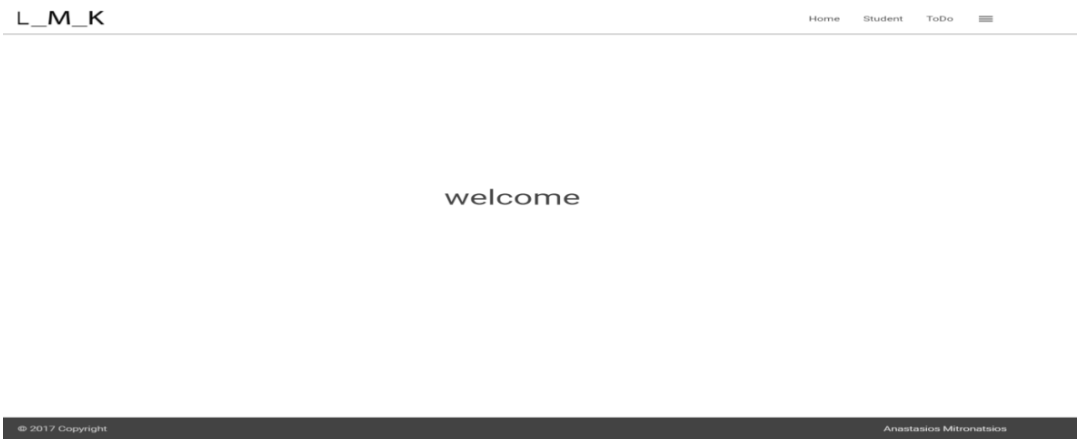
- Students_history: Στον πίνακα αυτόν θα αποθηκεύονται τα στοιχεία της συνάντησης του φοιτητη. Ο πίνακας αυτός θα συσχετίζεται με τον πίνακα «students» βάση του πεδίου student_id το οποίο είναι και ξένο κλειδί. Τα υπόλοιπα πεδία του πίνακα είναι το id, title, notes, date. Πρωτεύον κλειδί του πίνακα αυτού είναι το πεδίο «id» τύπου autoincrement και λαμβάνει σειριακή τιμή κατά την δημιουργίας εγγραφής.

- Todo: Στον πίνακα αυτόν θα γίνεται η αποθήκευση μιας ενέργειας που είναι προγραμματισμένη να γίνει στο άμεσο μέλλον. Για την ενέργεια αυτή είναι απαραίτητη η εισαγωγή των πεδίων full_name, notes, date, time, subject, phone, email. Πρωτεύον κλειδί του πίνακα αυτού είναι το πεδίο «id» τύπου autoincrement και λαμβάνει σειριακή τιμή κατά την δημιουργίας εγγραφής.

- Users: Ο πίνακας αυτός δημιουργήθηκε για να υπάρχει μια υποδομή σε περίπτωση αναβάθμισης της εφαρμογής με μηχανισμό δημιουργίας λογαριασμού χρήστη στο σύστημα. Τα πεδία που θεωρήθηκαν σημαντικά είναι τα τα εξής: id , username, password, email, created. Πρωτεύον κλειδί του πίνακα αυτού είναι το πεδίο «id» τύπου autoincrement και λαμβάνει σειριακή τιμή κατά την δημιουργίας εγγραφής.

4.3.2 Welcome

Ο χρήστης αφού εισάγει τα κατάλληλα διαπιστευτήρια οδηγείται όπως φαίνεται και στην (εικόνα 29) στην σελίδα dashboard.



Εικόνα 29: welcomedashboar

Εξαιτίας της νέας τεχνολογίας που χρησιμοποιήσαμε για την υλοποίηση της εφαρμογής θα παρατηρήσουμε, ότι όταν ο φυλλομετρητής αλλάζει σελίδα στον φυλλομετρητή, ο φυλλομετρητής δεν κάνει ανανέωση. Δηλαδή, ο χρήστης μπορεί να περιηγηθεί σε όλη την εφαρμογή με μία μόνο ανανέωση. Με αυτό τον τρόπο εξοικονομεί χρόνο και γίνεται η εργασία του πιο εύκολη και ταυτόχρονα πιο εύχρηστη. Η εφαρμογή LetMeKnow αποτελείται από 3 σελίδες.

- Home
- Student
- ToDo

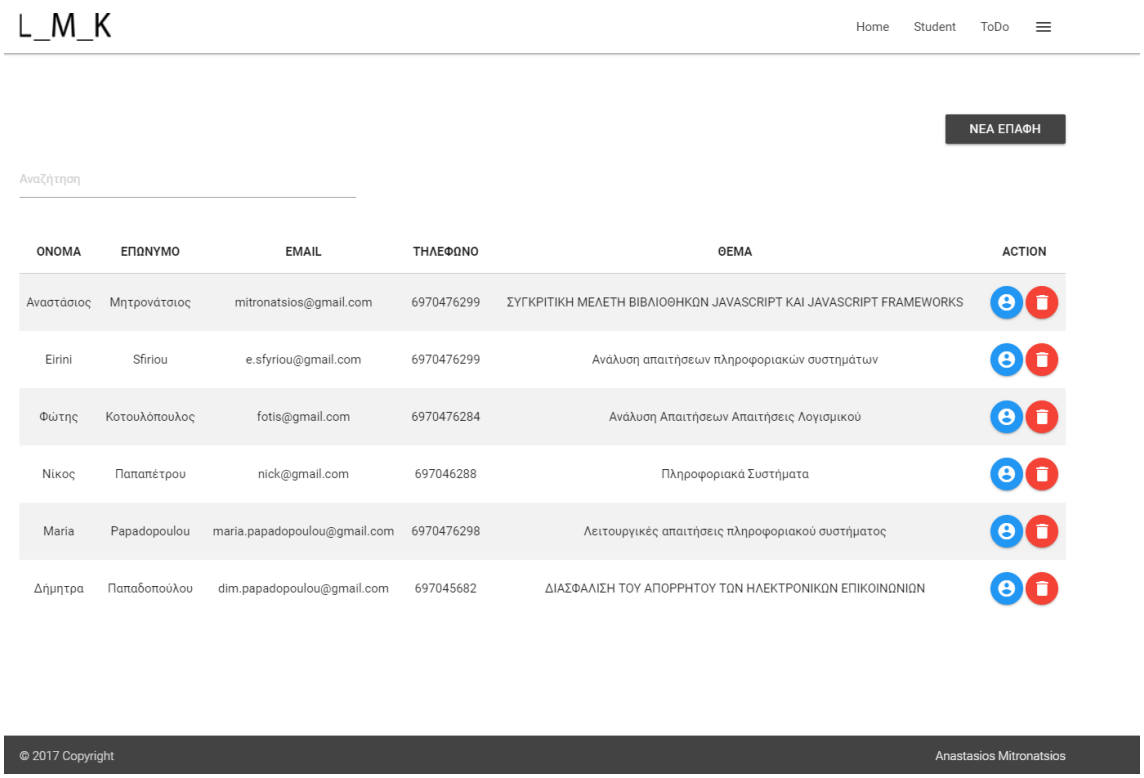
Η σελίδα «Home» είναι η σελίδα καλωσορίσματος, η οποία φαίνεται στο προηγούμενο στιγμιότυπο (εικόνα 24). Το ενδιαφέρον αυτής της εφαρμογής επικεντρώνεται στις επόμενες σελίδες, τις οποίες θα αναλύσουμε βήμα βήμα στην συνέχεια. Αρχικά θα αναλύσουμε την καρτέλα «Student», τον τρόπο λειτουργιά της

καθώς επίσης την σκοπιμότητά της στον χρήστη. Με τον ίδιο ακριβώς τρόπο θα συνεχίσουμε και στην καρτέλα «ToDo».

4.3.3 Καρτέλα «Student»

Η καρτέλα «Student» όπως παρατηρούμε στο παρακάτω στιγμιότυπο (εικόνα 29), αποτελείται, πάνω δεξιά από ένα κουμπί με την ονομασία «ΝΕΑ ΕΠΑΦΗ», στην από κάτω σειρά υπάρχει ένα πεδίο εισαγωγής «Αναζήτηση» και ακριβώς από κάτω υπάρχουν κάποιες εγγραφές φοιτητών.

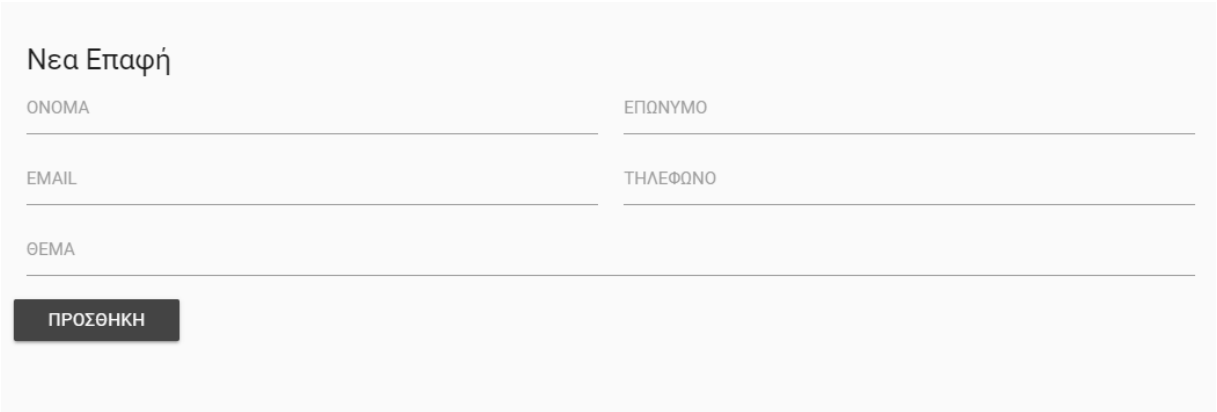
Επιπλέον, σε κάθε εγγραφή υπάρχουν τα πεδία όνομα, επώνυμο, email, τηλέφωνο, θέμα καθώς επίσης και δύο ενέργειες. Και οι δύο ενέργειες εκτελούνται με κουμπί που έχει προγραμματιστεί κατάλληλα. Το μπλε κουμπί μας ανακατευθύνει στην καρτέλα φοιτητή ενώ το κόκκινο στην οριστική διαγραφή της εγγραφής.



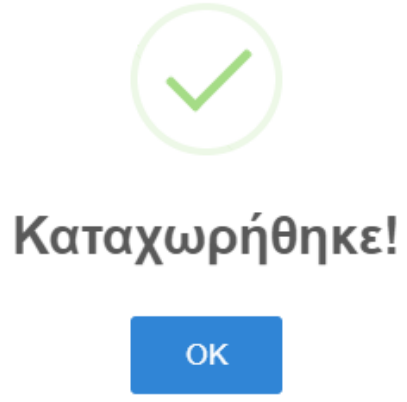
Εικόνα 30: Studentpage.

4.3.4 Νέα επαφή

Σε αυτήν την παράγραφο θα αναφερθούμε πως κάνουμε εισαγωγή μιας νέας επαφής. Πατώντας το κουμπί «ΝΕΑ ΕΠΑΦΗ» εμφανίζεται ένα αναδυόμενο παράθυρο με κάποια υποχρεωτικά πεδία συμπλήρωσης (εικόνα 30). Για να γίνει επιτυχής μια εγγραφή θα πρέπει να συμπληρωθούν όλα τα στοιχεία. Μόλις ο χρήστης συμπληρώσει τα απαραίτητα πεδία θα πρέπει να πατήσει το κουμπί «ΠΡΟΣΘΗΚΗ», αν τα στοιχεία είναι σωστά θα εμφανιστεί ένα επιτυχές μήνυμα (εικόνα 31). Τα στιγμιότυπα των ενεργειών αυτών φαίνονται στις παρακάτω απεικονίσεις.



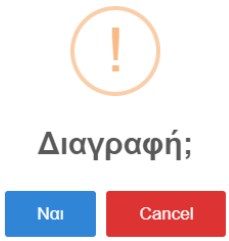
Εικόνα 31: Εισαγωγή νέας επαφής.



Εικόνα 32: Επιτυχία εισαγωγή επαφής.

4.3.5 Διαγραφή φοιτητή

Για την διαγραφή του φοιτητή θα πρέπει να εκτελέσουμε την ενέργεια που είναι προγραμματισμένη στο κόκκινο κουμπί. Πατώντας το κουμπί αναδύεται το παρακάτω μήνυμα (εικόνα 33),το οποίο μας δίνει δύο επιλογές: είτε να ακυρώσουμε την διαγραφή, είτε να προχωρήσουμε στην οριστική διαγραφή.

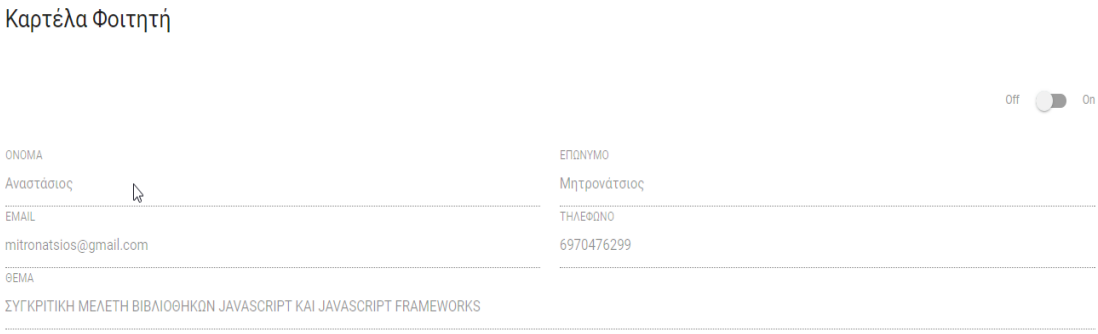


Εικόνα 33: Αναδυόμενο μήνυμα διαγραφής

4.3.6 Καρτέλα Φοιτητή – Ιστορικό

Στην καρτέλα αυτή παρουσιάζονται τα προσωπικά στοιχεία του φοιτητή, καθώς επίσης και το ιστορικό των συναντήσεων που έχουν λάβει μέρος στο παρελθόν σε σχέση με τον επιβλέπων καθηγητή. Μέσα από το ιστορικό αυτό, ο καθηγητής και ο φοιτητής μπορεί να έχει ανά πάσα στιγμή ένα feedback σχετικά με το πόσο έχει προχωρήσει την διπλωματική εργασία, καθώς επίσης με το τι ειπώθηκε σε κάθε συνάντηση.

Μέχρι τώρα έχουμε αναφερθεί στην δημιουργία καρτέλας φοιτητή, στην ανάκτηση στοιχείων του φοιτητή, όπως επίσης στη οριστική διαγραφή καθώς και στην αναζήτηση αυτού. Παρακάτω θα δείξουμε πώς γίνεται η ενημέρωση των προσωπικών στοιχείων του φοιτητή(εικόνα 33).

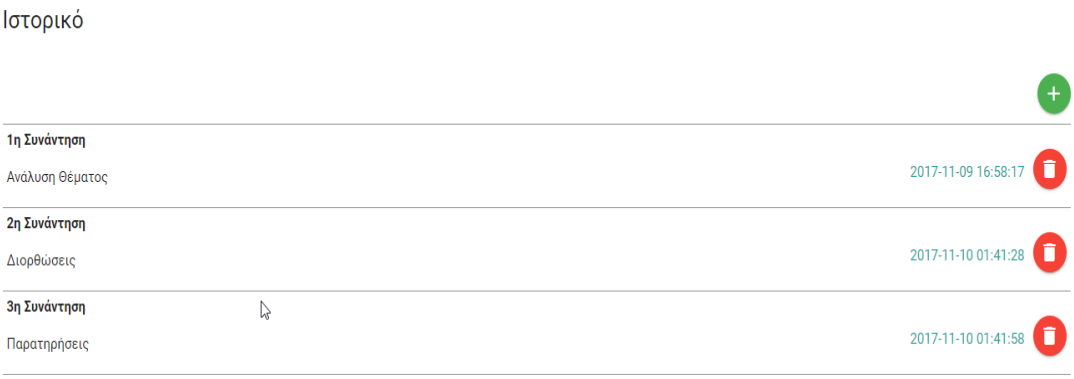


Εικόνα 34: Προσωπικά στοιχεία φοιτητή

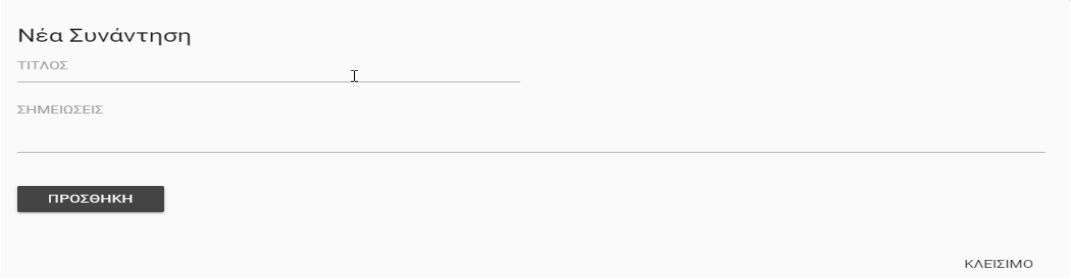
Στο παραπάνω στιγμιότυπο (εικόνα 34) απεικονίζεται μια ενδεικτική φόρμα με τα προσωπικά στοιχεία του φοιτητή. Για την επεξεργασία των στοιχείων αυτών αρκεί να μεταβληθεί ο διακόπτης (πάνω δεξιά) από το «off» στο «on».

4.3.7 Καταγραφή Συνάντησης

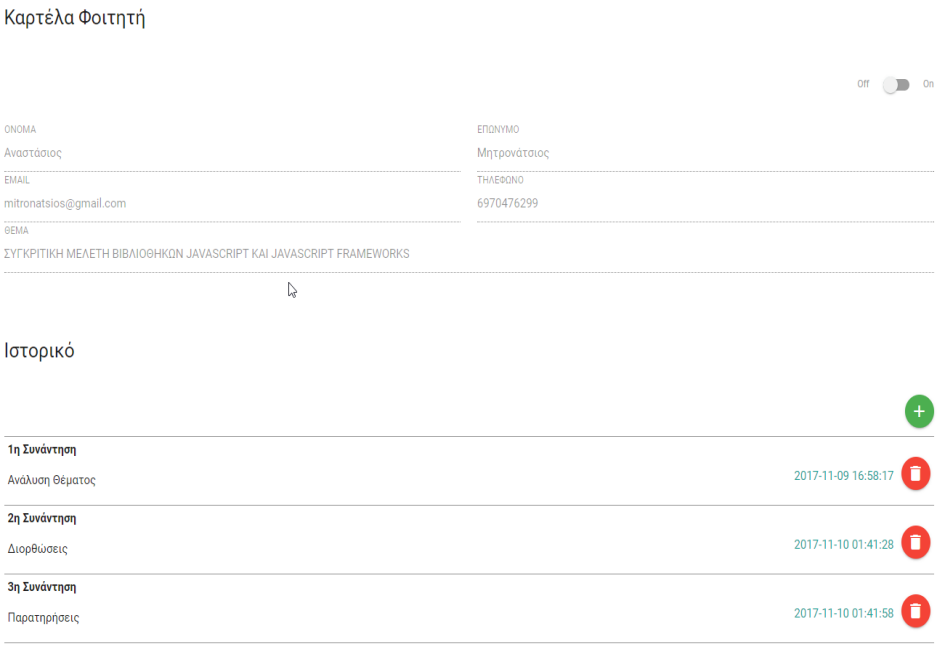
Μόλις τελειώσει η συνάντηση του επιβλέποντα καθηγητή με τον φοιτητή, ο καθηγητής θα πρέπει να δημιουργήσει μία νέα εγγραφή στην εφαρμογή. Αυτό πετυχαίνετε πατώντας το πράσινο κουμπί, με αναγραφόμενο ειδικό χαρακτήρα «+», (εικόνα 35). Πατώντας το κουμπί αναδύεται ένα νέο παράθυρο με υποχρεωτικά πεδία συμπλήρωσης «Τίτλος» και «Σημειώσεις», (εικόνα 36). Αφού συμπληρωθούν τα στοιχεία αυτά, τότε μόνο γίνεται η καταγραφή της συνάντησης. Παρακάτω φαίνονται τα στιγμιότυπα την δημιουργίας «Νέας συνάντησης» καθώς επίσης και την ανάκτηση αυτών στην οθόνη του χρήστη.



Εικόνα 35: Ιστορικό συναντήσεων φοιτητή με τον επιβλέποντα καθηγητή



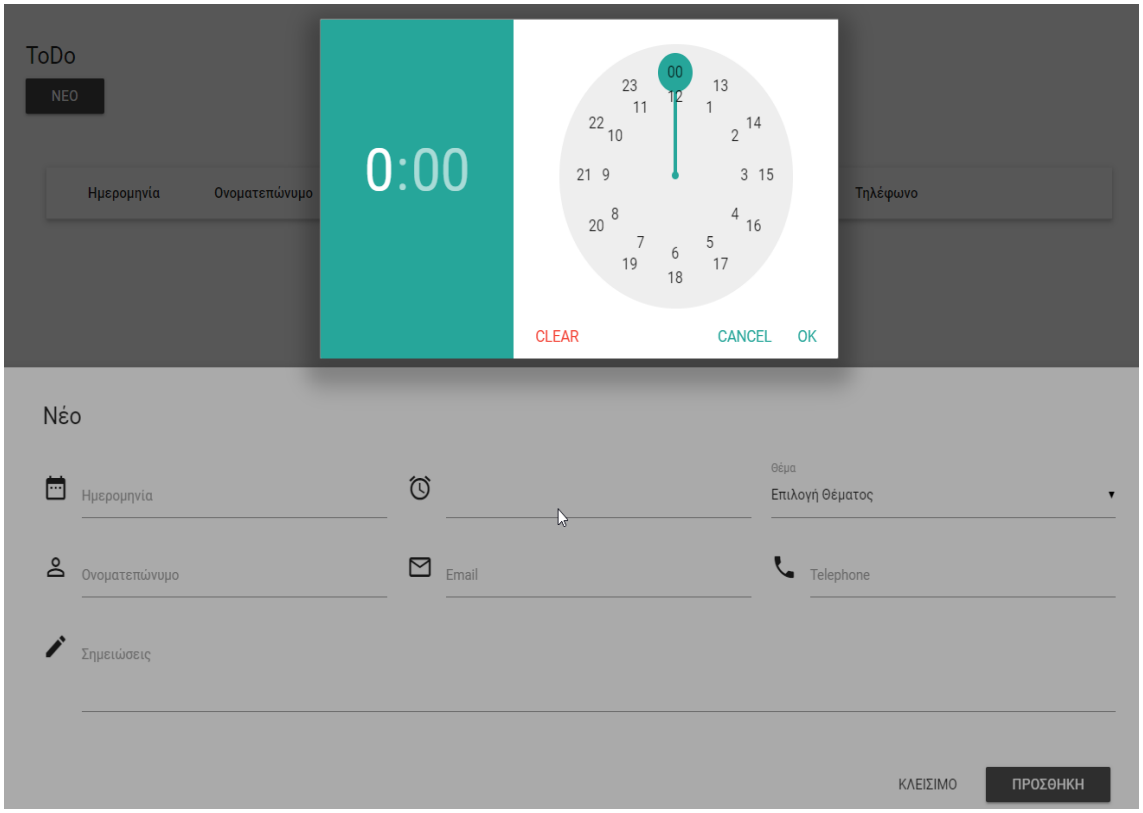
Εικόνα 36: Δημιουργία νέας συνάντησης



Εικόνα 37: Απεικόνιση καρτέλας φοιτητή σε συνδυασμό με το ιστορικό συναντήσεων.

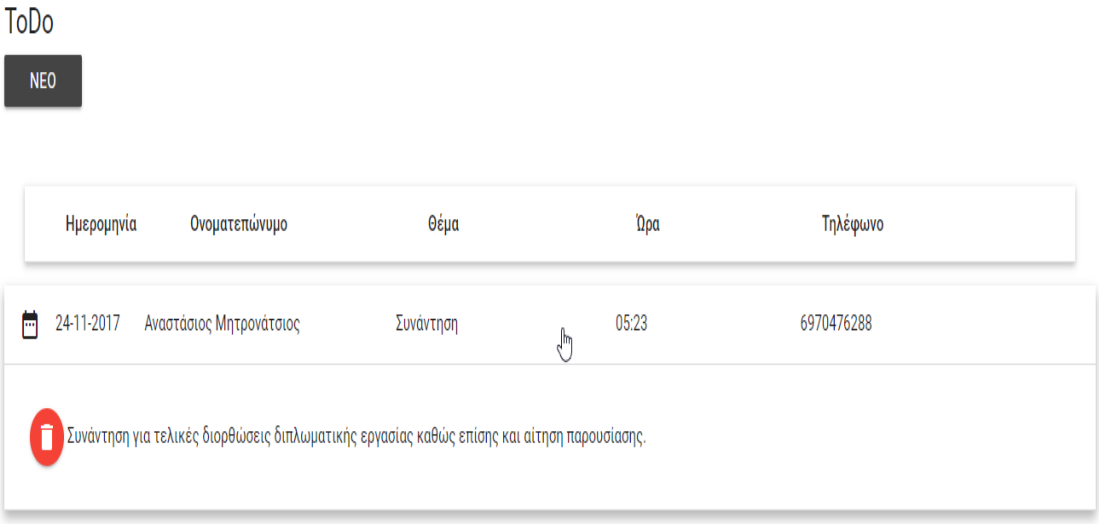
4.3.8 Καρτέλα ToDo

Η καρτέλα ToDo χρησιμοποιείται αποκλειστικά από τον καθηγητή και μόνο. Έχει δημιουργηθεί για την υπενθύμιση των ραντεβού που είναι προγραμματισμένα να γίνουν στο άμεσο ή έμμεσο μέλλον. Η χρήση της βοηθάει τόσο στην υπενθύμιση του καθηγητή που είναι υπεύθυνος για την δημιουργία του συμβάντος αυτού, όσο και του μαθητή ή συναδέλφου που θα λάβει μέρος στην συνάντησης αυτή. Η υπενθύμιση γίνεται μέσω ηλεκτρονικής αλληλογραφίας, καθώς επίσης και αποστολή μηνύματος τόσο στον «πομπό» δηλαδή στον καθηγητή, όσο και στον «δέκτη» δηλαδή κάποιον σπουδαστή ή κάποιον άλλον συνάδελφο (εικόνα 38). Η ειδοποίηση γίνεται 24 ώρες πριν και μια ώρα πριν το συμβάν.



Εικόνα 38:Δημιουργία νέας υπενθύμισης

Παρακάτω βλέπουμε τα στιγμιότυπα δημιουργίας μιας υπενθύμισης, καθώς επίσης και την εμφάνισή τους στην οθόνη του χρήστη.



Εικόνα 39: Εμφάνιση της υπενθύμισης

4.4 Αξιολόγηση εφαρμογής

Καθ' όλη τη διάρκεια της έρευνας, της μελέτης και της υλοποίησης του στόχου, σκοπός της παρούσας διπλωματικής εργασίας ήταν η συγκριτική μελέτη βιβλιοθηκών Javascript και JavascriptFrameworks. Κατόπιν της διαδικασίας έρευνας και μεθοδολογίας τέθηκαν κάποια σημαντικά κριτήρια αξιολόγησης.

Βασικό μέλημα ήταν από την αρχή να παρέχουμε στον χρήστη να διαβάζει όσο το δυνατόν γρηγορότερα τον ιστότοπο που επισκέπτεται. Εκτός όμως από τις δυνατότητες που δύνονται στο χρήστη, εξίσου σημαντικό κομμάτι ήταν η ανάπτυξη εφαρμογής μέσω μιας συγκεκριμένης αρχιτεκτονικής δομής (MVC), η οποία καθοδηγεί τον προγραμματιστή μια καθορισμένη σειρά αλληλουχίας ενεργειών. Η μη επανάληψη κώδικα έχει ως συνέπεια την μείωση όγκου της εφαρμογής στο δίσκο και συγχρόνως την μείωση χρόνου φόρτωσης της εφαρμογής στον χρήστη.

Τέλος, η εφαρμογή έχει δομηθεί έτσι ώστε να υπάρξουν στο μέλλον αρκετές πρόσθετες λειτουργίες , χωρίς κανένα περιορισμό ως προς τη δυνατότητα επέκτασής της και βελτιστοποίησής της.

Κεφάλαιο 5 : Συμπεράσματα και προτάσεις

5.1 Συμπεράσματα

Ολοκληρώνοντας την παρούσα διπλωματική εργασία και έχοντας μελετήσει διεξοδικά τα ζητήματα που πραγματεύτηκε, μπορούμε να οδηγηθούμε στην εξαγωγή ορισμένων συμπερασμάτων.

Όπως αναφέρθηκε και στην εισαγωγή της συγκεκριμένης διπλωματικής εργασίας δεν υπάρχει ένας και μοναδικός προκαθορισμένος τρόπος δημιουργίας μιας εφαρμογής αλλά πρέπει να ακολουθούμε την ιδανικότερη μεθοδολογία που θα οδηγήσει στην σωστή ανάπτυξη αυτής. Αυτό επίσης θα βοηθήσει και στην βελτιστοποίηση της εφαρμογής.

Βασιζόμενοι, λοιπόν, στο διάγραμμα σύγκρισης «Καταμέτρηση γραμμών» διαπιστώνουμε, ότι ανάμεσα στις δύο εφαρμογές για την συγκεκριμένη ενέργεια, η χρήση κώδικα που απαιτείται στην εφαρμογή «Framework» είναι μικρότερη σε σχέση με την εφαρμογή «Library».

Αναλύοντας το διάγραμμα «Σύγκριση χώρου στον διακομιστή» συμπεραίνουμε, ότι η εφαρμογή που κατασκευάστηκε με την χρήση της τεχνολογίας «Angular», δεσμεύει περισσότερο χώρο σε σχέση με την εφαρμογή που υλοποιήθηκε με την βιβλιοθήκη της «JQuery».

Επιπρόσθετα, όσο άφορα το διάγραμμα «Σύγκριση χρόνου φόρτωσης Ιστοσελίδας» ανάμεσα στις δυο εφαρμογές προκύπτει μια διαφορά σχετικά με το χρόνο εκτέλεσης. Η εφαρμογή «Framework» απαιτεί περισσότερο χρόνο φόρτωσης σε σύγκριση με την εφαρμογή «Library».

Συγκρίνοντας τα παραπάνω, οδηγούμαστε αρχικά στο εξής συμπέρασμα: η εφαρμογή «Library» υπερτερεί σε σχέση με την εφαρμογή «Framework». Με μία βαθύτερη ανάλυση διαπιστώνουμε ότι το αποτέλεσμα αυτό είναι αντιστρόφως ανάλογο, και αυτό λόγω της επανάληψης κώδικα στην εφαρμογή «Library» σε περίπτωση δημιουργίας νέων ιστοσελίδων. Η επανάληψη αυτή δημιουργεί πρόβλημα πολυπλοκότητας, με αποτέλεσμα, η αύξηση χώρου κατανάλωσης στον διακομιστή να είναι μεγαλύτερη σε σύγκριση με την εφαρμογή «Framework». Τέλος, στο διάγραμμα σύγκρισης «χρόνου φόρτισης ιστοσελίδας», η εφαρμογή «Library» μπορεί να φορτώνει

την ιστοσελίδα «home» πιο γρήγορα, αλλά η δεύτερη στον ίδιο χρόνο περίπου φορτώνει ολόκληρη την εφαρμογή. Βάση των νέων δεδομένων που προέκυψαν, συμπεραίνουμε ότι η τελευταία υπερτερεί της «Library».

Έχοντας ως γνώμονα όλα τα παραπάνω, η εφαρμογή «Framework» είναι πιο εύχρηστη, πιο αποδοτική και ακολούθως πιο επεκτάσιμη σε σχέση με την εφαρμογή «library». Για το λόγο αυτό, η εφαρμογή «LetMeknow» αναπτύχθηκε χρησιμοποιώντας το Framework «Angular».

5.2 Προτάσεις

Σκοπός της εφαρμογής αυτής είναι κάλυψη αναγκών στην επικοινωνία μεταξύ σπουδαστή και καθηγητή. Έχει σχεδιαστεί με τέτοιο τρόπο, ώστε να είναι εφικτή σε ενημερώσεις, ανάλογα με τις ανάγκες που προκύπτουν μέσα από την καθημερινή συναναστροφή διδάσκοντα και φοιτητή. Παρακάτω προτείνονται κάποιες ενδεικτικές προτάσεις σχετικά με την βελτίωση της εφαρμογής «LetMeKnow».

- Διασύνδεση με το API της Google «Google Calendar» για την καταγραφή συναντήσεων.
- Δημιουργία blog για τις αναρτήσεις ανακοινώσεων του καθηγητή.
- Δημιουργία chatroom για όλους τους φοιτητές που παρακολουθούν τον εκάστωτε καθηγητή σε κάποιο μάθημά του.
- Δημιουργία λογαριασμού για τον σπουδαστή ώστε να παρακολουθεί τα αποτελέσματα των εξετάσεών του και επιπλέον να συμμετέχει στις συζητήσεις που γίνονται ανοιχτά.
- Δημιουργία on-line σύστημα καταγραφής συνάντησης σπουδαστή - καθηγητή.

Εν κατακλείδι, οι προτάσεις που προτάθηκαν αποσκοπούν στις καθημερινές ανάγκες που προκύπτουν στην επικοινωνία σπουδαστή – καθηγητή. Υλοποιώντας όλες τις παραπάνω προτάσεις που επισημάνθηκαν θα δημιουργηθεί μία νέα σχέση σπουδαστή – καθηγητή που θα είναι πιο ευχάριστη, πιο αποδοτική και κατά συνέπεια πιο φιλική.

Παράρτημα κώδικα

Σύνδεση με τη βάση «connectDB.php»

Ο παρακάτω κώδικας αποσκοπεί στην σύνδεση της βάσης δεδομένων με την εφαρμογή. Μόλις γίνει η σύνδεση στη βάση, τότε μόνο έχουμε το δικαίωμα νέας εγγραφής , ανάκτηση δεδομένων, όπως επίσης την επεξεργασία και την ενημέρωση αυτών.

```
<?php
header("Access-Control-Allow-Origin: *");
header("Content-Type: application/json; charset=UTF-8");
$servername = "localhost";
$username = "root";
$password = "";
$db_name = "diplomaDB";

// Create connection
$conn = new mysqli($servername, $username, $password , $db_name);

// Check connection
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}
//echo "Connectedsuccessfully";
?>
```

Δημιουργίαφόρμαςεγγραφής

```
<h5>ΝεαΕπαφή</h5>
<div class="row"  ng-controller="StudentController">
<form  class="col s12"  method="POST" name="addItem" role="form" >
<div class="row">
<div class="input-field col s6">
<input id="name" type="text" ng-model="form.first_name" class="validate">
<label for="name">ONOMA</label>
</div>
<div class="input-field col s6">
<input id="lastName" type="text" ng-model="form.last_name" class="validate">
<label for="lastname">ΕΠΩΝΥΜΟ</label>
</div>
<div class="input-field col s6">
<input id="email" ng-model="form.email" type="text" class="validate">
<label for="email">EMAIL</label>
</div>
<div class="input-field col s6">
<input id="phone" ng-model="form.phone" type="tel" class="validate">
<label for="phone">ΤΗΛΕΦΩΝΟ</label>
</div>
<div class="input-field col s12">
<input id="subject" ng-model="form.subject" type="text" class="validate">
<label for="subject">ΘΕΜΑ</label>
```

<button type="submit" ng-click="saveAdd()" class="waves-teal waves-light btn">ΠΡΟΣΘΗΚΗ</button>

</form>

Παρατηρούμε ότι σε κάθε πεδίο εισαγωγής περιέχεται ένα attribute με το όνομα ng-model="form.____". Στο attribute αυτό αποθηκεύεται η τιμή που δίνουμε. Στο τέλος της φόρμας υπάρχει το elementbutton ένα attribute, το οποίο καλεί μια συνάρτηση την saveAdd() και στέλνει τις τιμές που καθορίσαμε.

Δημιουργία εγγραφής

Ο παρακάτω κώδικα είναι υπεύθυνος για την δημιουργία μια νέας εγγραφής στην βάση.

```
$scope.saveAdd = function() {
   $http({
      url: '/thesis/actions/student.php',
      method: 'POST',
      data: {
        action:  'insert',
        name : $scope.form.first_name,
        lastName : $scope.form.last_name,
```

```
        email: $scope.form.email,
        phone: $scope.form.phone,
        subject: $scope.form.subject
      },
      headers: {'Content-Type': 'application/x-www-form-urlencoded'}
    }).then(function(response) {
     $scope.getData();
     swal(
      'Καταχωρήθηκε!',
      '',
      'success'
     ).then(function() {
      $route.reload();
     });
     $('#modal1').modal('close');
}, function(response) {
});
  }
```

Ανάκτηση εγγραφής

Για την ανάκτηση δεδομένων χρησιμοποιείται ο παρακάτω κώδικας.

```
$scope.getData = function() {
```

```
    $http.get("/thesis/actions/student.php")
     .then(function (response) {
       $scope.names = [];
       response.data.records.forEach(function(record) {
$scope.names.push(record);
       });
     });
  }
  $scope.getData();
```

Παρατηρούμε ότι η συνάρτηση getData καλεί το αρχείο student.php, το οποίο με την σειρά του εκτελεί ένα ερώτημα στην βάση, ώστε να γίνει ανάκτηση των δεδομένων που επιθυμούμε. Το «query» που εκτελεί φαίνεται στον παρακάτω κώδικα.

```
  $result = $conn->query("SELECT * FROMstudents");
$outp = "";
  while($rs = $result->fetch_array(MYSQLI_ASSOC)) {
    if ($outp != "") {$outp .= ",";}
    $outp .= '{"Name":"'  . $rs["first_name"] . '",';
      $outp .= '"LastName":"'   . $rs["last_name"]       . '",';
      $outp .= '"Id":"'   . $rs["id"]       . '",';
      $outp .= '"Email":"'   . $rs["email"]       . '",';
      $outp .= '"Phone":"'   . $rs["phone"]       . '",';
      $outp .= '"Subject":"'. $rs["subject"]     . '"}';
    }
    $outp ='{"records":['.$outp.']}';
```

```
$conn->close();

echo($outp);

  }
```

Renderεγγραφής

Η εμφάνιση του αποτελέσματος στην οθόνη του χρήστη απεικονίζεται στον παρακάτω κώδικα.

```
<table class="striped centered responsive-table">

<thead>

<tr>

<th>ONOMA</th>

<th>ΕΠΩΝΥΜΟ</th>

<th>EMAIL</th>

<th>ΤΗΛΕΦΩΝΟ</th>

<th>ΘΕΜΑ</th>

<th>ACTION</th>

</tr>

</thead>

<tbody ng-controller="StudentController">

<tr id="infos" ng-repeat="x in names | filter:search" ng-animate="animate">

<td>{{ x.Name }}</td>

<td>{{ x.LastName }}</td>

<td>{{ x.Email }}</td>

<td>{{ x.Phone }}</td>
```

<td>{{ x.Subject }}</td>

<td>

account_circle

delete

</td>

</tr>

</tbody>

</table>

Στον παραπάνω κώδικα υπάρχει το attributeng-repeat="xinnames | filter:search , το οποίο μας δείχνει πώς γίνεται η εμφάνιση των στοιχείων για κάθε εγγραφή και μας επισημαίνει ταυτόχρονα, ότι έχουμε την δυνατότητα να κάνουμε αναζήτηση των δεδομένων αυτών, βάση όλων των στοιχείων.

Διαγραφή εγγραφής

<aclass="btn-floatingbtn-smallred" ng-click="removeStudent(x.Id)">

delete

Παρατηρούμε ότι και σε αυτόν τον κώδικα υπάρχει το attribute, το οποίο καλεί μία συνάρτηση removeStudent(x.Id) με το id της εγγραφής. Ο κώδικας της συνάρτησης αυτής φαίνεται παρακάτω.

```
 $scope.removeStudent = function(studentId){
swal({
    title: 'Διαγραφή;',
    text: "",
    type: 'warning',
    showCancelButton: true,
    confirmButtonColor: '#3085d6',
cancelButtonColor: '#d33',
    confirmButtonText: 'Ναι'
  })
  .then(function () {
    $http({
      url: '/thesis/actions/student.php',
      method: 'POST',
      data: {
        id :studentId,
        action:  'delete',
      },
      headers: {'Content-Type': 'application/x-www-form-urlencoded'}
    }).then(function(response) {
        swal('Διαγράφηκε!',
            '',
```

```
            'success'
          );
        $scope.getData();
      }, function(response) {
console.log(response)
});
  });
 };
```

Σε περίπτωση που επιθυμούμε την οριστική διαγραφή και πατήσουμε «ΝΑΙ», η συνάρτηση καλεί το αρχείο student.php και ταυτόχρονα στέλνει και το «id» της εγγραφής που επιθυμούμε να διαγράψουμε. Στην συνεχεία το αρχείο αυτό εκτελεί το ερώτημα στην βάση και το διαγράφει. Ο κώδικα του ερωτήματος αυτού φαίνεται παρακάτω.

```
$id = $data->id;
$sql= $conn->query("SELECT student_id FROM student_history where student_id = $id
");
  if(mysqli_num_rows($sql) > 0){
   $sql = "DELETE students , student_history FROM students INNER JOIN
student_history on students.id = student_history.student_id where students.id =$id ";
    if ($conn->query($sql) === TRUE) {
     //echo "Record deleted successfully";
    } else {
     echo "Error deleting record: " . $conn->error;
    }
  }else{
```

```
    $sql = "DELETE FROM students WHERE id=$id";
    if ($conn->query($sql) === TRUE) {
      //echo "Record deleted successfully";
    } else {
      echo "Error deleting record: " . $conn->error;
}
   }
  }
```

Ενημέρωσηεγγραφής

```
$scope.updateInfos = function() {
    var Name = $("#name").val();
    var lastName = $("#lastName").val();
    var email = $("#email").val();
    var phone = $("#phone").val();
    var subject = $("#subject").val();
     $http({
         url: '/thesis/actions/studentprofile.php',
         method: 'POST',
        data: {
          id:$scope.StudentId,
          action:  'updated',
          name : Name,
          lastName : lastName,
```

```
            email: email,
            phone: phone,
            subject: subject
          },
          headers: {'Content-Type': 'application/x-www-form-urlencoded'}
      }).then(function(response) {
        //console.log(response);
       }, function(response) {
console.log(response)
});
    }
```

Η συνάρτηση καλεί το αρχείο studentprofile.php και στέλνει το jsonData με τις νέες τιμές που έχουμε δώσει. Το αρχείο studentprofile.php συνδέεται με την βάση και εκτελεί το ερώτημα στην βάση που είναι υπεύθυνο για την ενημέρωση των στοιχείων. Ο κώδικας που εκτελεί το ερώτημα φαίνεται παρακάτω.

```
 $first_name =  $data->name;
$last_name =  $data->lastName;
   $email =  $data->email;
   $phone =  $data->phone;
   $subject =  $data->subject;
   $id =  $data->id;;
   $sql = "UPDATE students
         SET first_name='".$first_name. "' ,last_name='".$last_name. "' ,email='".$email.
"',phone='".$phone. "',subject='".$subject. "'
        WHERE id='".$id."'";
```

```
if ($conn->query($sql) === TRUE) {

//  echo "Record updated successfully";

} else {

  echo "Error" . $conn->error;

}
```

ΒΙΒΛΙΟΓΡΑΦΙΑ

Α. Άρθρα - Βιβλία

1. Douglas Crockford , 2008 , “Javascript: The Good Parts”

2. KOWALK., 2009, “CommonJS effort sets Javascript on path for world domination”, Ars Technica, Condé Nast Publications.

3.HAMILTON N., 2008, “The A-Z of Programming Languages: Javascript”, computerworld.com.au

4. CROCKFORD D., 2008, “Javascript: The Good Parts”, O’Reilly Media

5. Cory Gackenheimer , 2015 , “ Introduction to React”

6. "Πολυμέσα - Δίκτυα", Γ' Ενιαίου Λυκείου, ΥΠ.Ε.Π.Θ - Π.Ι., Αθήνα 1999

7. GREEN B. & SESHADRI S., 2013, “AngularJS”, O’Reilly Media

8. JAIN N., MANGAL P. & MEHTA D., 2014, “AngularJS: A Modern MVC Framework in Javascript”, Journal of Global Research in computer science, Volume 5.

9. VENKAT G., 2009, “Why JQuery”, VishwakJ. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, pp. 68–73.

10. DAVID M., 2011, “Developing websites with JQuery mobile.”, Focal Press.

11. LI J., & LI H., 2013, “ Management System of SanBao Village Peace District Based on JQuery.”, In Computational and Information Sciences (ICCIS), 5th International Conference on (pp. 336-339). IEEE.

12. McFARLAND D. S., 2012, “Writing Your First Javascript Program.”, Javascript& JQuery, The Missing Manual, O’Reilly, Sebastopol, CA: 21-40.

13. KROL K., & SZOMOROVA L., 2015. “The possibilities of using chosen JQuery Javascript components in creating interactive maps.”, Geomatics, Landmanagement and Landscape, (2), 45-54.

14.Navnath N. Chavare 1, A. V. Nadargi, 2016 “Performance Comparison and Evaluation of JQuery with AngularJS”,International Research Journal of Engineering and Technology (IRJET)

Β. Σημειώσεις

1. "Netscape and Sun announce Javascript", PR Newswire, 1995

2. ECMAScript 3rd Edition specification, 2017

Γ. Ιστοσελίδες

https://el.wikipedia.org/wiki/Javascript

https://en.wikipedia.org/wiki/Javascript_Framework

https://el.wikipedia.org/wiki/%CE%94%CE%B9%CE%B5%CF%81%CE%BC%CE%B7%CE%BD%CE%AD%CE%B1%CF%82_(%CF%85%CF%80%CE%BF%CE%BB%CE%BF%CE%B3%CE%B9%CF%83%CF%84%CE%AD%CF%82)

http://www.dofactory.com/Javascript/design-patterns

https://facebook.github.io/react/

https://en.wikipedia.org/wiki/React_(Javascript_library)

https://el.wikipedia.org/wiki/JQuery

https://en.wikipedia.org/wiki/AngularJS

https://JQuery.com/

http://studentguru.gr/w/tutorials/a0-Javascript-Javascript-html

https://en.wikipedia.org/wiki/List_of_Javascript_libraries

https://en.wikipedia.org/wiki/Comparison_of_Javascript_Frameworks

http://dide.flo.sch.gr/Plinet/Tutorials/Tutorials-JavascriptTerminology.html

https://shapebootstrap.net/blog/96-most-popular-Javascript-libraries-to-learn-in-2017

https://facebook.github.io/react/docs/jsx-in-depth.html