



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Προηγμένα Συστήματα Πληροφορικής»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Ανάπτυξη διαδικτυακής εφαρμογής για τη δημιουργία και ψηφοφορία ομάδων μπάσκετ Web application for the creation and voting of basket teams
Όνοματεπώνυμο Φοιτητή	Γολεγού Γεωργία
Πατρώνυμο	Εμμανουήλ
Αριθμός Μητρώου	ΜΠΣΠ/14017
Επιβλέπων	Ευθύμιος Αλέπης, Αναπληρωτής Καθηγητής

Ημερομηνία Παράδοσης Οκτώβριος 2021

Τριμελής Εξεταστική Επιτροπή

Βίρβου Μαρία
Καθηγήτρια

Αλέπης Ευθύμιος
Αναπληρωτής Καθηγητής

Πατσάκης Κωνσταντίνος
Αναπληρωτής Καθηγητής

ΠΕΡΙΛΗΨΗ

Η παρούσα διπλωματική εργασία έχει ως κύριο στόχο την σχεδίαση και ανάπτυξη μιας Web εφαρμογής αθλητικού περιεχομένου, για χρήση σε οποιονδήποτε browser καθώς και για κινητές συσκευές.

Στόχος της εφαρμογής είναι ο χρήστης, μέσα από μια λίστα αθλητών, να δημιουργεί και να ψηφίζει 2 ιδανικές για εκείνον αντίπαλες ομάδες μπάσκετ. Μετά το πέρας της ψηφοφορίας, από τα δεδομένα που έχουν συλλεχθεί, θα παρουσιάζεται η "στατιστικά" καλύτερη δεκάδα.

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ:

web, application, browser, component, Vue.js, CSS, Javascript

ABSTRACT

Main target of this current thesis is designing and developing a sport-content WEB application, used by any kind of browser as well as any kind of mobile devices.

Purpose of this application is, the user to be able to choose at his will, from an athlete list, to create and vote 2 ideal rival basketball teams for him. After voting, from the collected data, the "statistically" best ten will be presented.

KEYWORDS:

web, application, browser, component, Vue.js, CSS, Javascript

ΕΥΧΑΡΙΣΤΙΕΣ

Για τη διεκπεραίωση της παρούσας Πτυχιακής Εργασίας, θα θέλαμε να ευχαριστήσω τον επιβλέπων καθηγητή κ.Ευθύμιο Αλέπη,για την βοήθεια που προσέφερε,και την οικογένεια μου για την υπομονή που έδειξε κατά την διάρκεια των σπουδών μου, καθώς και τον φίλο μου Ανδρέα Γεωργίου, για τη συνεργασία και την πολύτιμη συμβολή του στην ολοκλήρωση της.

ΕΙΣΑΓΩΓΗ

Τα τελευταία χρόνια, είναι έντονη η αύξηση ολοένα και περισσότερων ιστοσελίδων που εγκαταλείπουν την κλασική χρήση βάσεων δεδομένων, στοχεύοντας σε NoSQL λύσεις βάσεων. Αίτια αυτού, αποτελεί η ραγδαία αύξηση και μεταβολή του όγκου των δεδομένων που ως αποτέλεσμα έχουν, οι κλασικές λύσεις βάσεων δεδομένων να μην μπορούν να ανταποκριθούν στα ζητούμενα.

Επιπλέον, αισθητή είναι και η χρήση των Javascript framework. Τα περισσότερα frameworks υλοποιούν συγκεκριμένες προγραμματιστικές τεχνικές που μισούν τον χρήστη τους σ' έναν τρόπο σκεψής που βοηθά να οργανωθεί καλύτερα ο κώδικας του. Έχοντας υλοποιημένες κάποιες βασικές λειτουργίες, επιτυγχάνεται πιο δομημένος και συγκροτημένος κώδικας. Τα frameworks ασχολούνται με τη συμβατότητα μεταξύ των browsers - cross-browser compatibility.

Τα frameworks είναι εδώ για να επιταχύνουν την υλοποίηση της ιστοσελίδας. Έχουν ενσωματωμένα στοιχεία HTML, CSS και JavaScript. Μέσω αυτών των πλαισίων, οι προγραμματιστές έχουν πρόσβαση σε έτοιμα Τυπογραφικά, Μορφές, Κουμπιά, Πίνακες, Πλοηγίες, Πινακίδες, Ειδοποιήσεις, Modals, Tabs, Accordion, Carousel κλπ. Και πολλά άλλα στοιχεία JavaScript, HTML και CSS. Συνοπτικά, τα πλαίσια Front-end είναι μια συλλογή λειτουργιών HTML, CSS και JavaScript.

Επίσης ένα καλογραμμένο framework βοηθά στην εκμάθηση νέων τεχνικών και συνεπάγεται σ' έναν καλύτερο προγραμματιστή που θα μπορεί να γραφεί και να διαχειρίζεται κώδικα πιο αποδοτικά.

Από την άλλη πλευρά βέβαια, αν ο χρήστης δεν γνωρίζει τα βασικά στοιχεία της γλώσσας στην οποία βασίζεται το framework, δεν θα υπάρξει ποτέ ουσιαστική γνώση και χρήση της.

Συνοψίζοντας, χρησιμοποιούμε έτοιμα frameworks, επειδή μας παρέχουν:

- έτοιμα χαρακτηριστικά που ελαχιστοποιούν τον χρόνο ανάπτυξης
- σχεδιασμό ιστοσελίδων προσαρμοζόμενο σε συσκευές responsive web design
- συμβατότητα μεταξύ των προγραμμάτων περιήγησης - cross-browser compatibility

ΠΡΟΛΟΓΟΣ

Αντικείμενο της παρούσας διπλωματικής εργασίας αποτελεί ο σχεδιασμός, η ανάπτυξη και η υλοποίησης μια διαδικτυακής πλατφόρμας δημιουργίας 2 "εικονικών" ομάδων μπάσκετ μέσα από μια λιστα διαθέσιμων επιλογών.

Η δομή της εργασίας συνοψίζεται ως εξής:

- Στο πρώτο κεφάλαιο γίνεται θεωρητική αναφορά στα εργαλεία και τα javascript frameworks που μελετήθηκαν και χρησιμοποιήθηκαν.
- Στο δεύτερο κεφάλαιο πραγματοποιείται εκτενής ανάλυση της διαδικασίας σχεδιασμού και υλοποίησης της πλατφόρμας με την χρήση εικόνων και επεξηγήσεων.
- Στο τελευταίο κεφάλαιο συνοψίζονται και παρουσιάζονται προβλήματα και περιορισμοί που αντιμετωπίστηκαν καθώς και πιθανή μελλοντική επίλυση τους.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ	7
ABSTRACT	8
ΕΙΣΑΓΩΓΗ	10
ΠΡΟΛΟΓΟΣ	11
ΚΕΦΑΛΑΙΟ 1.	15
ΕΙΣΑΓΩΓΗ	15
1.1 Διαδικτυακή Εφαρμογή (Web Application)	15
1.2 HTML 5	16
1.3 JAVASCRIPT	16
1.3.1 Εισαγωγή	16
1.3.2 Τι είναι η javascript	16
1.3.3 Βασικά χαρακτηριστικά	17
1.3.4 Βασικά πλεονεκτήματα	17
1.3.5 Χρήση της Javascript	19
1.4 AJAX	19
1.5 SASS	20
1.6 JAVASCRIPT FRAMEWORKS	20
1.6.1 Front-End Programming	24
1.6.2 NODE JS	25
1.6.3 Vue JS	26
1.7 JSON Objects	30
1.8 PHPstorm	30
ΚΕΦΑΛΑΙΟ 2	32
ΑΝΑΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ ΚΑΙ ΣΧΕΔΙΑΣΜΟΣ	32
2.1 Συνοπτικά	32
2.2 Αναλυση	33
2.3 Αναλυση των Components	41
2.3.1 Stadium Component	41
2.3.2 Player Component	41
2.3.3 Lineup Component	41
2.3.4 Player Modal Component	41
ΚΕΦΑΛΑΙΟ 3	42
Συμπεράσματα	42
ΚΕΦΑΛΑΙΟ 4	42
Επεκτάσεις για το μέλλον	42
Βιβλιογραφικές Αναφορές	43

Λίστα Εικόνων

Εικόνα 1. Σύγκριση δημοτικότητας των πιο διαδεδομένων js frameworks	14
Εικόνα 2. Σύγκριση ενδιαφέροντος των πιο διαδεδομένων js frameworks	14
Εικόνα 3. Rankings	15
Εικόνα 4. Πληροφορίες frameworks	19
Εικόνα 5. User input model	20
Εικόνα 6. Component Life circle	21
Εικόνα 7. Project	25
Εικόνα 8. Project JSON Object	26
Εικόνα 9. Project	27
Εικόνα 10. Project	28
Εικόνα 11. Project	29
Εικόνα 12. Project	30
Εικόνα 13. Project	30
Εικόνα 14. Project	31
Εικόνα 15. Cookies set	31
Εικόνα 16. Project - Mobile	32

ΚΕΦΑΛΑΙΟ 1.

ΕΙΣΑΓΩΓΗ

Πολύ περιληπτικά παρουσιάζονται τα βασικά εργαλεία που χρησιμοποιήθηκαν για την διεκπεραίωση της πλατφόρμας και γίνεται εκτενέστερη περιγραφή της javascript και του αντιστοιχου framework που χρησιμοποιήθηκε καθώς αποτέλεσαν τα σημαντικότερα και πιο εξειδικευμένα εργαλεία.

1.1 Διαδικτυακή Εφαρμογή (**Web Application**)

Μια διαδικτυακή εφαρμογή είναι ένα πρόγραμμα υπολογιστή που χρησιμοποιεί προγράμματα περιήγησης ιστού και τεχνολογίες ιστού για την εκτέλεση εργασιών μέσω του Διαδικτύου. Εκατομμύρια χρήστες χρησιμοποιούν το Διαδίκτυο ως δίαυλο επικοινωνίας, για αγορές, για εκπαιδευτικούς σκοπούς, για συναλλαγές κλπ. Τους επιτρέπει να ανταλλάσσουν πληροφορίες γρήγορα και αποτελεσματικά.

Οι διαδικτυακές εφαρμογές χρησιμοποιούν έναν συνδυασμό scripts από την πλευρά του server με PHP και ASP, καθώς και οι πιο σύγχρονες εφαρμογές με Node.js για να διαχειριστούν την αποθήκευση και την ανάκτηση των πληροφοριών, και από την πλευρά του πελάτη (JavaScript και HTML) για την παρουσίαση πληροφοριών στους χρήστες. Αυτό επιτρέπει στους χρήστες να αλληλεπιδρούν με την εφαρμογή χρησιμοποιώντας ηλεκτρονικές φόρμες, συστήματα διαχείρισης περιεχομένου, καλάθια αγορών κλπ. Επιπλέον, οι εφαρμογές επιτρέπουν στους χρήστες να δημιουργούν έγγραφα, να μοιράζονται πληροφορίες, να συνεργάζονται σε έργα και να εργάζονται σε κοινά έγγραφα ανεξάρτητα από την τοποθεσία ή τη συσκευή.

Οι εφαρμογές διαδικτύου γράφονται σε κώδικα (γλώσσα προγραμματισμού) που υποστηρίζεται από προγράμματα περιήγησης, όπως JavaScript και HTML, καθώς αυτές οι γλώσσες βασίζονται στο πρόγραμμα περιήγησης για να καταστήσουν εκτελέσιμο το πρόγραμμα.

Η εφαρμογή διαδικτύου απαιτεί έναν server ιστού να διαχειρίζεται αιτήματα από τον χρήστη, έναν server εφαρμογών για την εκτέλεση των ζητούμενων εργασιών και, ορισμένες φορές, μια βάση δεδομένων για την αποθήκευση των πληροφοριών.

Οι διαδικτυακές εφαρμογές εκτελούνται σε πολλαπλές πλατφόρμες ανεξάρτητα από το λειτουργικό σύστημα ή την συσκευή, εφόσον το πρόγραμμα περιήγησης είναι συμβατό. Όλοι οι χρήστες έχουν πρόσβαση στην ίδια έκδοση, εξαλείφοντας τυχόν προβλήματα συμβατότητας. Δεν είναι εγκατεστημένα στον σκληρό δίσκο, εξαλείφοντας έτσι τους περιορισμούς χώρου. Μειώνουν το κόστος τόσο για την επιχείρηση όσο και για τον τελικό χρήστη, καθώς απαιτείται λιγότερη υποστήριξη και συντήρηση από την επιχείρηση και χαμηλότερες απαιτήσεις για τον υπολογιστή του τελικού χρήστη. Ταυτόχρονα όμως υπάρχει και η άλλη πλευρά του νομίσματος. Αδυναμία χρήσης της εφαρμογής χωρίς σύνδεση στο διαδίκτυο δηλαδή δεν είναι δυνατόν να χρησιμοποιηθούν αν δεν υπάρχει σύνδεση με το διαδίκτυο. Μη πλήρης συμβατότητα των περιηγητών δηλαδή αρκετοί από τους περιηγητές δεν είναι ακόμα πλήρως συμβατοί με την τελευταία έκδοση HTML. Έτσι δεν γίνεται πλήρης χρήση των δυνατοτήτων αυτών, πράγμα που περιορίζει τους προγραμματιστές που έχουν αναλάβει ένα έργο. Επίσης, σε περίπτωση που δεν έχει προβλεφθεί η μη λειτουργία κάποιου χαρακτηριστικού της εφαρμογής σε κάποιον περιηγητή, αυτό μπορεί να δημιουργήσει προβλήματα στην εφαρμογή με αποτέλεσμα να μην λειτουργεί σωστά ή να μην λειτουργεί καθόλου.

1.2 HTML 5

Η HTML5 είναι μια εξέλιξη της HTML με πληθώρα νέων δυνατοτήτων που καθιστούν την κατασκευή ιστοσελίδων ευκολότερη με αποτέλεσμα μια ευχάριστη εμπειρία για τον χρήστη. Πλέον, δίνεται η επιλογή

διαδραστικότητας (σε περιορισμένο βαθμό φυσικά). Επιπλέον έχουν προστεθεί νέα δομικά στοιχεία ανάπτυξης.

1.3 JAVASCRIPT

1.3.1 Εισαγωγή

Η JavaScript είναι μια δυναμική γλώσσα προγραμματισμού. Στην αρχή αποτέλεσε μέρος της υλοποίησης των περιηγητών, ώστε τα σενάρια από την πλευρά του πελάτη να μπορούν να επικοινωνούν με τον χρήστη, να ανταλλάσσουν δεδομένα ασύγχρονα και να αλλάζουν δυναμικά το περιεχόμενο του εγγράφου που εμφανίζεται. Η JavaScript είναι η πιο δημοφιλής scripting γλώσσα στον κόσμο. Με τον όρο scripting, εννοείται μια ελαφριά γλώσσα προγραμματισμού που υποστηρίζει τη συγγραφή σεναρίων. Τα σενάρια είναι γραμμές κώδικα που μπορούν να ερμηνεύονται και να εκτελούνται χωρίς μεταγλώττιση. Με αυτήν την scripting γλώσσα, ο χρήστης μπορεί να αλληλεπιδράσει σε μια ιστοσελίδα. Αποτελεί μια client-side γλώσσα που σημαίνει ότι εκτελείται και παράγεται το αποτέλεσμα στον browser του χρήστη. Δεν είναι κατάλληλη στο να υλοποιηθεί εξ' ολοκλήρου μια δυναμική ιστοσελίδα, αλλά λειτουργεί συμπληρωματικά ώστε να παραχθεί ένα πιο αλληλεπιδραστικό αποτέλεσμα. Επιτρέπει στον χρήστη να πραγματοποιεί κάποιες ενέργειες χωρίς να επικοινωνεί η πλατφόρμα με τον server για να εκτελέσει το αίτημα του. Συνήθως, ακολουθεί συνεργατικά την HTML.

1.3.2 Τι είναι η javascript

Η Javascript είναι μια γλώσσα προγραμματισμού και δημιουργήθηκε αρχικά από τον Brendan Eich της εταιρείας Netscape με την επωνυμία Mocha. Αργότερα, Mocha μετονομάστηκε σε LiveScript, και τελικά σε JavaScript, κυρίως επειδή η ανάπτυξή της επηρεάστηκε περισσότερο από τη γλώσσα προγραμματισμού Java. Η JavaScript απέκτησε μεγάλη επιτυχία ως γλώσσα στην πλευρά του πελάτη (client-side) για εκτέλεση κώδικα σε ιστοσελίδες, και περιλήφθηκε σε διάφορα προγράμματα περιήγησης στο Web. Η JavaScript έχει γίνει μία από τις πιο δημοφιλείς γλώσσες προγραμματισμού ηλεκτρονικών υπολογιστών στον Παγκόσμιο Ιστό (Web). Αρχικά, όμως, πολλοί επαγγελματίες προγραμματιστές υποτίμησαν τη γλώσσα διότι το κοινό της ήταν ερασιτέχνες συγγραφείς ιστοσελίδων και όχι επαγγελματίες προγραμματιστές (και μεταξύ άλλων λόγων). Με τη χρήση της τεχνολογίας Ajax, η JavaScript γλώσσα επέστρεψε στο προσκήνιο και έφερε πιο επαγγελματική προσοχή προγραμματισμού. Το αποτέλεσμα ήταν ένα καινοτόμο αντίκτυπο στην εξάπλωση των πλαισίων και των βιβλιοθηκών, τη βελτίωση προγραμματισμού με JavaScript, καθώς και αυξημένη χρήση της JavaScript έξω από τα προγράμματα περιήγησης στο Web.

Η JavaScript δεν ήταν πάντα μια δημοφιλής γλώσσα. Απορρίφθηκε από τους προγραμματιστές ως ερασιτεχνική γλώσσα. Έγινε δημοφιλής επειδή πλέον κυριαρχεί στο διαδίκτυο και στις εφαρμογές κινητού, ανταγωνίζεται επιτυχώς τις υπόλοιπες γλώσσες προγραμματισμού καθώς και γλώσσες του διακομιστή και δεν παραμένει πίσω ούτε στο Διαδίκτυο των πραγμάτων (Internet of Things).

Η JavaScript είναι μια εξέχουσα τεχνολογία ιστού που κυριαρχεί στον τομέα. Ως μια ευέλικτη γλώσσα προγραμματισμού μπορεί να υλοποιηθεί στις εφαρμογές του πελάτη (client), του διακομιστή (server), των κινητών και των επιτραπέζιων υπολογιστών και να χρησιμοποιηθεί για να σχεδιάσει τις πρώτες διεπαφές, να εμπλουτίσει εφαρμογές ιστού με πολλές λειτουργίες και δυνατότητες, να τροποποιήσει ιστοσελίδες σε πραγματικό χρόνο και πολλά άλλα. Η τεχνολογία είναι πολύ ισχυρή, αποτελεσματική και μεγάλη για την κατασκευή γρήγορων και υψηλής απόδοσης εφαρμογών ιστού και εφαρμογών για κινητά, επεκτάσεων προγραμμάτων περιήγησης, γι' αυτό έχει μεγάλη υποστήριξη από εκατομμύρια προγραμματιστές. Τα τελευταία χρόνια προτιμάται γιατί κάνει τα πάντα με τον πιο αποτελεσματικό και ευκολότερο τρόπο.

Κατά συνέπεια, η εταιρεία Microsoft ονόμασε την εφαρμογή της σε JScript για να αποφύγει δύσκολα θέματα εμπορικών σημάτων. Η JScript πρόσθεσε νέους μεθόδους για να διορθώσει τα Y2K-προβλήματα στην JavaScript, οι οποίοι βασίστηκαν στην java.util.Date τάξη της Java. Η JScript περιλήφθηκε στο Internet Explorer version 3.0, το οποίο κυκλοφόρησε τον Αύγουστο του 1996. Τον Νοέμβριο του 1996, η Netscape

ανακοίνωσε ότι είχε υποβάλει τη γλώσσα JavaScript στο Ecma International (μια οργάνωση της τυποποίησης των γλωσσών προγραμματισμού) για εξέταση ως βιομηχανικό πρότυπο, και στη συνέχεια το έργο είχε ως αποτέλεσμα την τυποποιημένη μορφή που ονομάζεται ECMAScript.

1.3.3 Βασικά χαρακτηριστικά

- Η JavaScript είναι δομημένη γλώσσα χαρακτηριστικό που κληρονόμησε από τη C.
- Είναι δυναμική τόσο στη σύνταξή της, όσο και στην εκτέλεσή της που είναι run-time.
- Είναι ταυτόχρονα και αντικειμενοστραφής, και συναρτησιακή – κάθε συνάρτηση είναι ένα αντικείμενο, έτσι μπορεί να έχει ιδιότητες και μεθόδους.
- Υποστηρίζει εμφωλευμένες συναρτήσεις όπως επίσης και ανώνυμες συναρτήσεις.

Η πιο συνηθισμένη χρήση της JavaScript είναι να προσθέσει συμπεριφορά στην HTML στην πλευρά του χρήστη.

1.3.4 Βασικά πλεονεκτήματα

Μερικά από τα πλεονεκτήματα της JavaScript που την καθιστούν τόσο δημοφιλή στους προγραμματιστές είναι τα ακόλουθα:

- **Επεξεργασία από την πλευρά του πελάτη (client-side):** Αυτό σημαίνει ότι ο κώδικας εκτελείται στον επεξεργαστή του χρήστη αντί του διακομιστή ιστού, εξοικονομώντας έτσι εύρος ζώνης και μειώνοντας το επιπλέον φορτίο του διακομιστή.
- **Απλή στην εκμάθηση:** Η σύνταξη της είναι παρόμοια με τα αγγλικά (σαν γλώσσα ομιλίας) με αποτέλεσμα την διευκόλυνση των προγραμματιστών.
- **Απλή στην εφαρμογή:** Η δυνατότητα χρήσης της ίδιας γλώσσας front-end και back-end διευκολύνει τους προγραμματιστές.
- **Δεν απαιτείται μεταγλώττιση:** Το πρόγραμμα περιήγησης ερμηνεύει τη JavaScript ως ετικέτες HTML.
- **Πιο γρήγορη και αποτελεσματική στον χρήστη:** κάθε πιθανό πεδίο ή δομικό στοιχείο μπορεί να επαληθεύεται καθώς συμπληρώνεται από τους χρήστες, γεγονός που παρέχει άμεση ανατροφοδότηση, όταν αυτοί κάνουν κάποιο λάθος.

1.3.5 Χρήση της Javascript

Κάποιες από τις βασικές χρήσεις της είναι οι εξής:

Έλεγχος πεδίων: Η JavaScript μπορεί να χρησιμοποιηθεί για την επικύρωση των πεδίων (για παράδειγμα σε φόρμες εισόδου). Τα δεδομένα που καταχωρούνται μπορούν να επικυρωθούν πριν από την επεξεργασία τους.

Αναδυόμενα Παράθυρα: Η JavaScript μπορεί να χρησιμοποιηθεί για τη δημιουργία αναδυόμενων παραθύρων. Αυτά τα παράθυρα χρησιμοποιούνται συνήθως για την προβολή σημαντικών ανακοινώσεων, προσφορών και ειδήσεων κλπ.

Δυναμικά περιεχόμενα: Η JavaScript μπορεί να χρησιμοποιηθεί για τη δημιουργία δυναμικών περιεχομένων σε μία ιστοσελίδα.

Αλληλεπίδραση χρηστών: Η JavaScript μπορεί να χρησιμοποιηθεί για το συμφέρον του χρήστη. Το πεδίο που εισάγεται από το χρήστη μπορεί να επεξεργαστεί και να εμφανιστεί το σωστό μήνυμα στον χρήστη. Οι διαδραστικές δυνατότητες μιας ιστοσελίδας την καθιστούν πιο ενδιαφέρουσα και παραγωγική για τους χρήστες.

1.4 AJAX

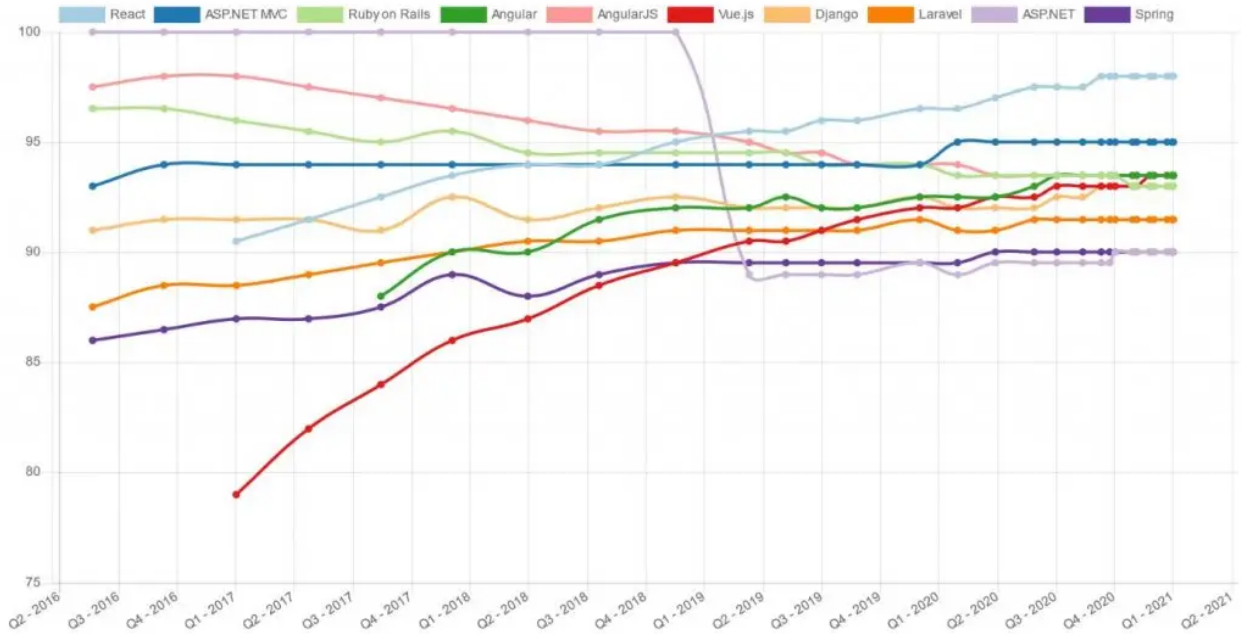
Η AJAX (Asynchronous JavaScript and XML) είναι ένα σύνολο τεχνικών ανάπτυξης ασύγχρονων εφαρμογών στο Διαδίκτυο που χρησιμοποιεί πολλές τεχνολογίες μαζί από την πλευρά του πελάτη (client). Αποτελεί τον δίαυλο επικοινωνίας μεταξύ server και client. Είναι μία τεχνολογία που χρησιμοποιεί και την javascript προκυμένου να πραγματοποιηθεί η προαναφερθείσα επικοινωνία. Ουσιαστικά, μόλις δωθεί η εντολή να πραγματοποιηθεί κάποια λειτουργία, το κύκλωμα αυτό θα επικοινωνήσει με τον server στο παρασκήνιο, θα ληφθούν τα πιθανά δεδομένα και θα παρουσιαστούν στον χρήστη, χωρίς να πραγματοποιηθεί επαναφόρτωση της ιστοσελίδας. Η AJAX δεν είναι από μόνο του μια τεχνολογία αλλά ένα σύνολο άλλων τεχνολογιών. Χρησιμοποιεί HTML και CSS για την σήμανση και την παρουσίαση των ιστοσελίδων και χρησιμοποιεί JavaScript και XMLHttpRequest για να κινεί δεδομένα ασύγχρονα ανάμεσα στον πελάτη και τον διακομιστή (client - server). Τα δεδομένα δύνανται να μετακινούνται και σε μορφή JSON ή XML. Ο διαχωρισμός της ανταλλαγής δεδομένων με την προβολή τους επιτρέπει στην ιστοσελίδα να αλλάζει το περιεχόμενο της δυναμικά χωρίς να χρειάζεται ολοκληρωτική ανανέωση της. Αφορά μια τεχνολογία που μας παρέχει μια ταχύτερη και εξυπηρετικότερη ιστοσελίδα.

1.5 SASS

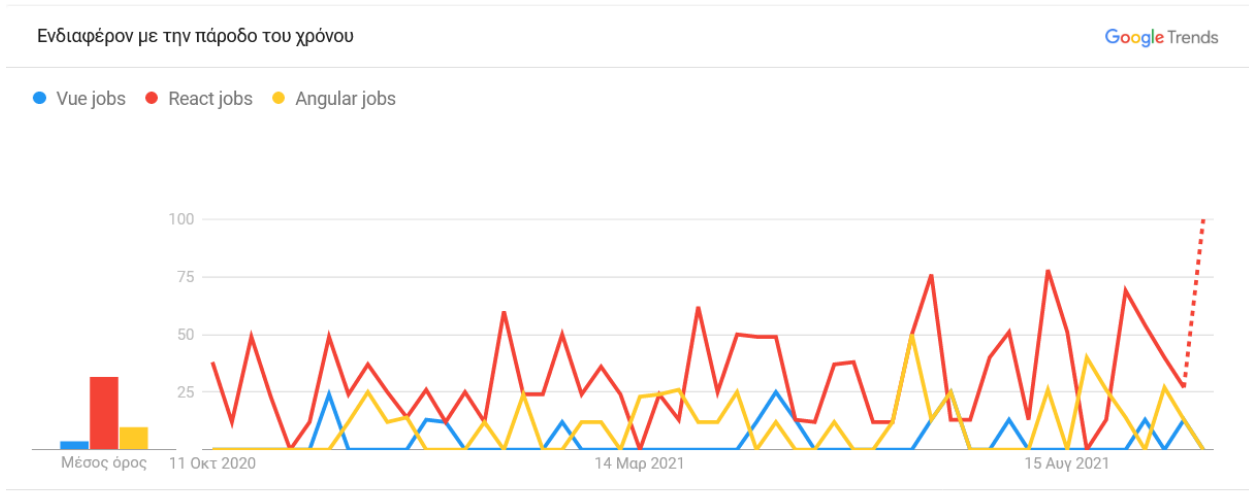
Η Sass αποτελεί έναν CSS preprocessor, ο οποίος προσθέτει ειδικά χαρακτηριστικά όπως μεταβλητές, εμφωλευμένους κανόνες και διάφορες "προγραμματιστικές" δυνατότητες στην κλασική CSS, με σκοπό έναν απλούστερο και αποτελεσματικότερο κώδικα. Το ακρονύμια προέρχεται από τις λέξεις Syntactically awesome style sheets. Το γραψίμο με sass γίνεται με πιο scripting τρόπο, όπου ο αντιστοιχος compiler το μετατρέπει στο κλασικό CSS.

1.6 JAVASCRIPT FRAMEWORKS

Καταλαβαίνουμε πως η JavaScript είναι η γλώσσα προγραμματισμού η οποία κυριαρχεί στο κόσμο του διαδικτύου τα τελευταία χρόνια και για αυτό γίνεται λόγος για τα frameworks. Αναπτύχθηκαν για να εξομαλύνουν τη διαδικασία και να παρέχουν ένα σύνολο μεθόδων λειτουργιών και ελέγχων. Τα πρώτα δημοφιλή frameworks της JavaScript ήταν Prototype, YUI, ExtJS, MooTools και jQuery. Καθώς περνούσαν τα χρόνια δημιουργήθηκαν frameworks όπως η Angular το 2016, η Backbone.js (2010), η React(2013), Ember.js (2015) και το Vue.js (2013). Τα JavaScript frameworks είναι εργαλεία για την κωδικοποίηση JavaScript πιο γρήγορα και πιο αποτελεσματικά.



Εικόνα 1. Σύγκριση δημοτικότητας των πιο διαδεδομένων js frameworks



Παγκοσμίως. Τελευταίοι 12 μήνες. Αναζήτηση στον Ιστό.

Εικόνα 2. Σύγκριση ενδιαφέροντος των πιο διαδεδομένων js frameworks

Rankings

Framework	Github Score	Stack Overflow Score	Overall Score
React	99	97	98
ASP.NET MVC		95	95
Angular	91	96	93
Ruby on Rails	87	99	93
AngularJS	90	97	93
Vue.js	100	87	93
Django	89	97	93
Laravel	90	93	91
ASP.NET	80	100	90
Spring	86	94	90

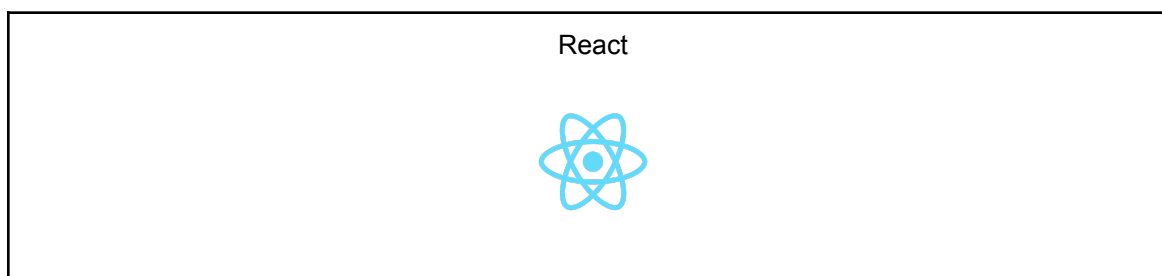
Εικόνα 3. Rankings

Τα Javascript Frameworks είναι συνήθως βασισμένα στο αρχιτεκτονικό πρότυπο MVC αρχιτεκτονικής (Model-view-controller), επειδή έχουν σχεδιαστεί για να διαχωρίζουν τις διάφορες πτυχές μιας διαδικτυακής εφαρμογής για τη βελτίωση της ποιότητας του κώδικα και για να κάνουν την ανάπτυξη του ευκολότερη. Είναι τυποποιημένα πακέτα που περιέχουν στοιχεία javascript, html και css και έχουν πρόσβαση σε έτοιμα Τυπογραφικά, Μορφές, Κουμπιά, Πίνακες, Πλοηγίες, Πινακίδες, Ειδοποιήσεις, Modals, Tabs, Accordion, Carousel κλπ. Τα περισσότερα από τα JavaScript Frameworks είναι ανοιχτού κώδικα και δωρεάν, μειώνοντας έτσι το συνολικό κόστος και είναι ελεύθερα διαθέσιμα. Επιπλέον, υποστηρίζεται από μεγάλες κοινότητες, και παρέχουν υποστήριξη και ασφάλεια.


Κύριος στόχος τους είναι να δώσουν μία κατεύθυνση στον προγραμματιστή, αποτελώντας μία αρχή πάνω στην οποία θα υλοποιηθεί μια εφαρμογή. Αυτό το καταφέρνουν μέσω των παρακάτω χαρακτηριστικών:


- Προωθώντας την επαναχρησιμοποίηση του κώδικα και παρέχοντας έτοιμες υλοποιήσεις για διάφορες εργασίες.
- Προβλέποντας για ενέργειες όπως ο έλεγχος ταυτότητας χρήστη (user authentication), αιτήματα AJAX (AJAX requests), η δρομολόγηση της εφαρμογής (routing) και η αποθήκευση μέρους της εφαρμογής στη μνήμη του υπολογιστή του χρήστη (caching) με σκοπό την εξοικονόμηση πόρων και τη γρηγορότερη φόρτωση της σε μετέπειτα χρόνο.

Ακολουθεί μια πολύ μικρή σύγκριση με τα άλλα 2 πιο δημοφιλή frameworks.



Pros	Μεγάλη απόδοση από πλευράς πελάτη. Καλός και σωστός έλεγχος δεδομένων. Ελαφριά αρχιτεκτονική. Επεκτασιμότητα. Μεγάλο community support
Cons	Δυσκολία εκμάθησης. Λειτουργία μόνο με χρήση βιβλιοθηκών Η ελευθερία δομής κώδικα οδηγεί σε αδόμητο κώδικα.

Angular 	
Pros	TypeScript Google Support Υψηλή απόδοση Σύσταση ενημερώσεων Προσαρμοστικότητα
Cons	Περιπλοκότητα Ελλιπής εκδόσεις Βαριά αρχιτεκτονική

Vue JS 	
Pros	Φιλικό προς τους προγραμματιστές. Εξαιρετικές επιδόσεις. Καλά δομημένος κώδικας. Κομψή σύνταξη γλώσσας. Εξαιρετική τεκμηρίωση χρήστη. Συμβατικότητα Ικανοποιητικό μέγεθος από πλευράς αρχιτεκτονικής Λειτουργικές επεκτάσεις

Cons	Έλλειψη έμπειρων προγραμματιστών. Μικρή κοινότητα. Ένα μεγάλο μέρος της τεκμηρίωσης είναι γραμμένο μόνο στα κινέζικα. Αντιδραστικότητα κώδικα. Έλλειψής υποστήριξη. Κίνδυνος ευελιξίας. Περιορισμός πόρων.
-------------	---

React vs Vue.js

Κοινά:

- Χρησιμοποιούν εικονικό DOM
- Λειτουργούν με την χρήση components
- Γρήγορα
- Συμβατότητα βιβλιοθηκών

Διαφορές

- Διαφορετικά templates για την δομή ενός project. Το React χρησιμοποιεί κατά μέγιστο βαθμό javascript, ενώ το vue.js από javascript, html και css ταυτόχρονα στο ίδιο component.
- Το vue παρέχει μεγαλύτερη παραμετροποίηση.
- Το vue διαθέτει δικό του css framework.

Η εφαρμογή της παρούσας εργασίας υλοποιήθηκε με την χρήση του Vue JS framework το οποίο θα αναλυθεί παρακάτω.

1.6.1 Front-End Programming

Με τον όρο "front-end" στον προγραμματισμό εννοούμε εκείνο το κομμάτι στο οποίο αλληλεπιδρούν οι χρήστες, δηλαδή αυτό που βλέπουν. Από γραμματοσειρές, χρώματα, εικόνες, μενού, αναδυόμενα παράθυρα μέχρι οτιδήποτε άλλο βλέπουν όταν σερφάρουν σε μια ιστοσελίδα. Όλα τα παραπάνω είναι μια σύνθεση από HTML, CSS και JavaScript τα οποία εκτελούνται στον browser. Το "front-end" κομμάτι, είναι αυτό που εκτελείται στο περιβάλλον του χρήστη, δηλαδή στον browser.

Τα πρώτα χρόνια αυτό περιελάμβανε τον προγραμματισμό μόνο σε HTML, CSS και JavaScript, όμως γρήγορα ξεκίνησαν να εμφανίζονται βιβλιοθήκες που βοηθούσαν στην εύκολη δημιουργία διαδραστικών προγραμμάτων. Τα τελευταία χρόνια έχουν δημιουργηθεί πολλά, ολοκληρωμένα JavaScript frameworks, που προσφέρουν τρομερές λειτουργίες στον χρήστη και πολλά εργαλεία για τη δημιουργία εφαρμογών υπολογιστή και κινητού. Η χρήση ενός framework έχει πάρα πολλά προτερήματα, αφού δεν χρειάζεται κάθε φορά που ξεκινάει ένα project να δημιουργούνται τα πάντα από την αρχή, ο προγραμματιστής μπορεί να δημιουργεί πράγματα γράφοντας λιγότερο κώδικα, γλιτώνοντας χρόνο και μπορεί να βασιστεί πάνω σε κώδικα που έχει γραφτεί και δοκιμαστεί από πάρα πολύ κόσμο, άρα και αρκετά έμπιστο.

1.6.2 NODE JS

Η αυξανόμενη δημοτικότητα της JavaScript έχει φέρει μαζί της πολλές αλλαγές, και η ανάπτυξη ιστοσελίδων σήμερα είναι εντυπωσιακά διαφορετική. Γι' αυτό το λόγο οι χρήστες απαιτούν ολοένα και πιο αποδοτικές και

γρήγορε εφαρμογές. Η καθυστέρηση και η απόδοση είναι βασικοί δείκτες απόδοσης για τους διακομιστές στο διαδίκτυο. Η διατήρηση της χαμηλής καθυστέρησης και η υψηλή απόδοση όσο αυξάνονται οι ανάγκες μιας ιστοσελίδας ή μιας εφαρμογής δεν είναι εύκολη.

Το Node.js είναι λοιπόν βασισμένο σε ένα runtime JavaScript περιβάλλον που επιτυγχάνει χαμηλή λανθάνουσα κατάσταση και υψηλή απόδοση. Δηλαδή, δεν σπαταλάει χρόνο ή πόρους για αναμονή των αιτημάτων εισόδου / εξόδου (I / O) για επιστροφή. Για κάθε εισερχόμενη εντολή ή σύνδεση ο server δημιουργεί ένα νέο αίτημα εκτέλεσης για την διαχείριση του και την αποστολή μιας απάντησης. Αυτό δημιουργεί μεγάλη επιβάρυνση.

<p>Πώς η PHP ή η ASP χειρίζονται ένα αίτημα αρχείου:</p> <ul style="list-style-type: none"> • Στέλνει την εργασία στο σύστημα αρχείων του υπολογιστή. Περιμένει ενώ ανοίγει το σύστημα αρχείων και διαβάζει το αρχείο. • Επιστρέφει το περιεχόμενο στο χρήστη. • Έτοιμο να χειριστεί το επόμενο αίτημα. 	<p>Πώς το Node.js χειρίζεται ένα αίτημα αρχείου:</p> <p>Στέλνει την εργασία στο σύστημα αρχείων του υπολογιστή.</p> <ul style="list-style-type: none"> • Έτοιμο να χειριστεί το επόμενο αίτημα. • Όταν το σύστημα αρχείων ανοίξει και διαβάσει το αρχείο, ο server επιστρέφει το περιεχόμενο στον χρήστη.
--	--

Το Node.js εξαλείφει την αναμονή και απλώς συνεχίζει με το επόμενο αίτημα. Εκτελεί έναν μονόκλωνο, μη αποκλειστικό, ασύγχρονο προγραμματισμό, ο οποίος είναι αποδοτικότερος για την μνήμη.

1.6.3 Vue JS

Η Vue υλοποιήθηκε από τον Evan You, πρώην υπάλληλο της Google το 2014, και τα τελευταία χρόνια έχει γνωρίσει μεγάλη δημοσιότητα, παρόλο που δεν έχει την στήριξη εταιρειών-κολοσσών όπως άλλα frameworks (Angular-Google, React-Facebook). Τελευταία σταθερή έκδοση είναι η 3.2 που κυκλοφόρησε τον Αυγουστο του 2021.

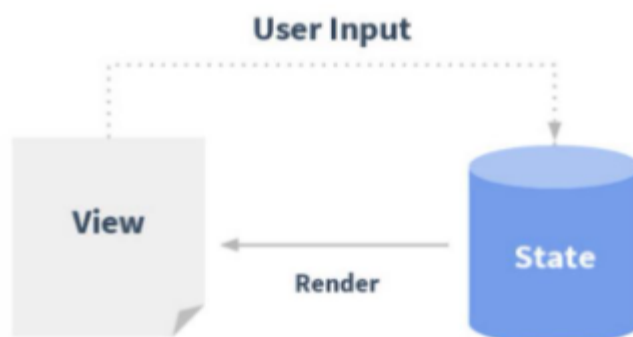
	Angular	React	Vue
Initial release	2010	2013	2014
Official site	angular.io	reactjs.org	vuejs.org
Current version	11	17.x	3.0.x
Used by	Google, Wix	Facebook, Uber	Alibaba, GitLab

Εικόνα 4. Πληροφορίες frameworks

Στα θετικά χαρακτηριστικά που αναφέρθηκαν νωρίτερα επισημαίνονται συμπληρωματικά και τα εξής για την vue.js:

- test utilities - παρέχονται εργαλεία που αφορούν το testing του κώδικα
- vue CLI - επιλογή configurations για την ανάπτυξη εφαρμογών
- vue dev-tools - plugin των browsers για την προβολή των δεδομένων, μεταβλητών, κλπ

Το vue.js εστιάζει στο κομμάτι της προβολής μιας εφαρμογής. Για να πραγματοποιηθεί αυτό, το framework κάνει χρήση του μοτίβου αρχιτεκτονικής MVVM (Model-view-viewmodel)



Εικόνα 5. User input model

Βασικό χαρακτηριστικό του Vue.js είναι ότι το σύστημα μπορεί να απεικονίσει τα δεδομένα στο DOM με την χρήση της κλασικής HTML. Η βασική δομική βιβλιοθήκη εστιάζει στο view layer.

Στον πυρήνα του Vue.js υπάρχει ένα σύστημα που μας δίνει τη δυνατότητα να αποδώσουμε στον DOM δηλωτικά δεδομένα.

Με αυτόν τον τρόπο δημιουργείται ένας σύνδεσμος μεταξύ DOM και δεδομένων, δηλαδή διαδραστικότητα. Αυτό σημαίνει ότι κάθε μεταβολή στα δεδομένα, απεικονίζονται άμεσα στο template. Η δήλωση της vue εφαρμογής(#app), λειτουργεί σαν ένα βασικό DOM στοιχείο, κάτω από το οποίο χτίζεται και διαχειρίζεται η εκάστοτε εφαρμογή.

Το Vue.js κάνει χρήση της λογικής των components, δηλαδή μικρά, αυτοτελή, ανεξάρτητα και συνήθως επαναχρησιμοποιούμενα στοιχεία και σ'ένα project περιέχονται τουλάχιστον 1 ή και περισσότερα components. Χρησιμοποιείται τουλάχιστον 1 component καθώς αποτελεί το root στοιχείο της εφαρμογής.

Κάθε component που δημιουργείται, μπαίνει αυτόματα σ' έναν κύκλο ζωής που παρουσιάζεται στο παρακάτω, άκρως αναλυτικό, σχήμα.

Στάδια κύκλου ζωής ενός component

1. Before create

Αποτελεί το πρώτο hook, αμέσως μετά την κλήση του Vue και πριν δημιουργηθεί το component. Δομικά στοιχεία όπως computed properties, watchers, events, data κλπ δεν έχουν δημιουργηθεί ακόμα.

2. Created

Το αμέσως επόμενο hook, στο οποίο αρχικοποιούνται, ενεργοποιούνται και εμφανίζονται τα πρώτα δομικά στοιχεία του component. Είναι εμφανή τα στοιχεία μόνο στο τμήμα View-Model.

3. Before mount

Η φάση κατά την οποία γίνονται compile τα templates και styles στοιχεία, χωρίς όμως να είναι δυνατή η διαχείριση τους.

4. Mounted

Το DOM στοιχεία έχει φορτωθεί και μπορεί να γίνει διαχείριση του.

5. Before update

Αφορά οποιαδήποτε αλλαγή γίνει στο DOM και πρέπει να ανανεωθεί.

6. Updated

Η κατάσταση αμέσως μετά από οποιοδήποτε update.

7. Before destroy

Η κατάσταση αμέσως πριν την πιθανή καταστροφή(διαγραφή).Σ' αυτή την φάση όλες οι λειτουργικότητες είναι ακόμα ενεργές. Στην παρούσα φάση μπορεί να γίνει διαχείριση των πόρων, διαγραφή μεταβλητών και δεδομένων, κλπ.

8. Destroyed

Το τελευταίο στάδιο του κύκλου ζωής, όπου όλα τα αντικείμενα έχουν καταστραφεί. Αντικείμενα εννοούμε τα events και τα vue directives.

Τα δομικά στοιχεία που απαρτίζουν ένα component είναι τα ακόλουθα:

➤ template - παράγει την markup (html κώδικας) ενός project και γίνεται χρήση των vue directives, δηλαδή attribute για να κάνεις conditional rendering σε elements. Κάποια από τα directives που χρησιμοποιούνται είναι:

- v-if
- v-else
- v-for
- v-on-{{ events}} -->events = click/load/change/select/etc..
- v-bind
- v-model
- v-html

- script - παράγεται η javascript του project με τα εξής αντίστοιχα (υπο)δομικά στοιχεία
- ονομα του vue component
- data - δεδομένα και μεταβλητές που χρησιμοποιούνται στο project
- component - δήλωση εξωτερικών components αν και εφόσον εισάγονται και χρησιμοποιούνται.
- props - attributes που εισάγονται στο component.
- computed - ιδιότητα όπου γίνεται εκ νέου υπολογισμός δεδομένων ή και μεταβλητών που υπάρχει εξάρτηση από άλλες τιμές. Όταν χρησιμοποιείται μια computed ιδιότητα μέσα στο template, ο DOM κάνει update, όταν μεταβάλλεται η computed τιμή του data που χρησιμοποιείται.
- mounted - καλείται όταν η φόρτωση του DOM έχει ολοκληρωθεί, ώστε να υπάρχει πρόσβαση στο αλληλεπιδραστικό κομμάτι των component, template, DOM στοιχείων καθώς και να είναι εφικτή η διαχείριση τους.
- watcher - ειδική λειτουργία που επιτρέπει σε κάποιον να παρακολουθεί ένα στοιχείο και να εκτελεί συγκεκριμένες ενέργειες όταν αλλάζει η τιμή του στοιχείου. Είναι ένας πιο γενικός τρόπος παρατήρησης και αλληλεπίδρασης στις αλλαγές δεδομένων. Είναι πιο χρήσιμα όταν χρησιμοποιούνται για την εκτέλεση ασύγχρονων λειτουργιών.
- created - Αρχικοποίηση του component. Σ' αυτή την φάση δεν μπορούμε να διαχειριστούμε τίποτα. Αποτελεί μια στατική έκδοση του component.
- methods - αποτελεί μια "λίστα" με όλες τις μεθόδους και την ανάπτυξη τους που χρησιμοποιούνται μέσα στο component.
- style - μορφοποίηση του project. Έχουμε 2 επιλογές.
- scoped - style που αφορά μόνο το component
- global - style που αφορά ολόκληρη την html μας

1.7 JSON Objects

Τα JSON (Javascript Object Notation) είναι μια μορφή δεδομένων που συχνά χρησιμοποιείται για την ανταλλαγή δεδομένων λόγω του μικρού μεγέθους της. Είναι ευανάγνωστα, εύκολα να συνταχθούν και ευκολα να αναλυθεί και να επεξεργαστεί από τις μηχανές. Είναι μια μορφή κειμένου πλήρως ανεξάρτητη από την γλώσσα προγραμματισμού που χρησιμοποιείται. Παρουσιάζεται ως μια συλλογή ζεύγους ονόματος-τιμής.

1.8 PHPstorm



Για την υλοποίηση και ανάπτυξη της πλατφόρμας μας χρησιμοποιήθηκε το PhpStorm το οποίο αποτελεί ένα Javascript IDE με πολλές δυνατότητες, όπως:

- Γρήγορη και εύκολη πλοήγηση
- Γρήγορη και εύκολη αναζήτηση και χρήση συναρτήσεων και κώδικα
- Εξυπηρετική γραφή με την λειτουργία του autocomplete στο κώδικα και με συντομέυσεις
- Διατίθεται για όλες τις πλατφόρμες λογισμικών
- Plugins για την ανάπτυξη κώδικα και την προσαρμογή του editor στα δεδομένα του χρήστη
- Υποστηρίζει Node.js
- Υποστηρίζει και τα 3 βασικότερα javascript frameworks (Angular, React, Vue)
- Άμεση πρόσβαση σε βάσεις δεδομένων απευθείας από το PhpStorm

- Ευκολο και αναλυτικό debugging

ΚΕΦΑΛΑΙΟ 2

ΑΝΑΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ ΚΑΙ ΣΧΕΔΙΑΣΜΟΣ

2.1 Συνοπτικά

Το web κομμάτι της πλατφόρμας είναι βασισμένο στο javascript framework Vue.js και παρέχει τις παρακάτω δυνατότητες στον χρήστη:

- Θα μπορεί να ενημερώνεται μέσα από μια λίστα παικτών σχετικά με τις βασικές αθλητικές πληροφορίες του καθενός.
- Θα επιλέγει τον παίκτη της κατάλληλης θέσης(forward, guard, center) που επιθυμεί για την αντίστοιχη θέση που έχει οριστεί στο γηπεδάκι που εμφανίζεται.
- Σε περίπτωση που δεν έχει επιλεγεί συγκεκριμένη θέση στο γήπεδο, θα συμπληρώνεται η πρώτη διαθέσιμη κενή θέση ανάλογα του επιλεγμένου παίκτη.
- Αν και εφόσον έχουν συμπληρωθεί όλα τα δυνατά σημεία του γηπέδου τότε ο χρήστης μπορεί να ολοκληρώσει και να δημοσιεύσει τις 2 επιλεγμένες αντίπαλες ομάδες.

2.2 Αναλυση

Το project παρουσιάζει ένα γήπεδο μπάσκετ, στο οποίο έχουν τοποθετηθεί pins που αφορούν τις βασικές θέσεις παικτών.



Εικόνα 7. Project

Για κάθε πλευρά ομάδας, οι θέσεις των παικτών είναι οι ακόλουθες:

- guards - οι δυάδες που είναι πιο κοντά στο κέντρο του γηπέδου
- forward - οι δυάδες δεξιά και αριστερά από την μπάσκέτα
- center - η θέση στην μπάσκέτα

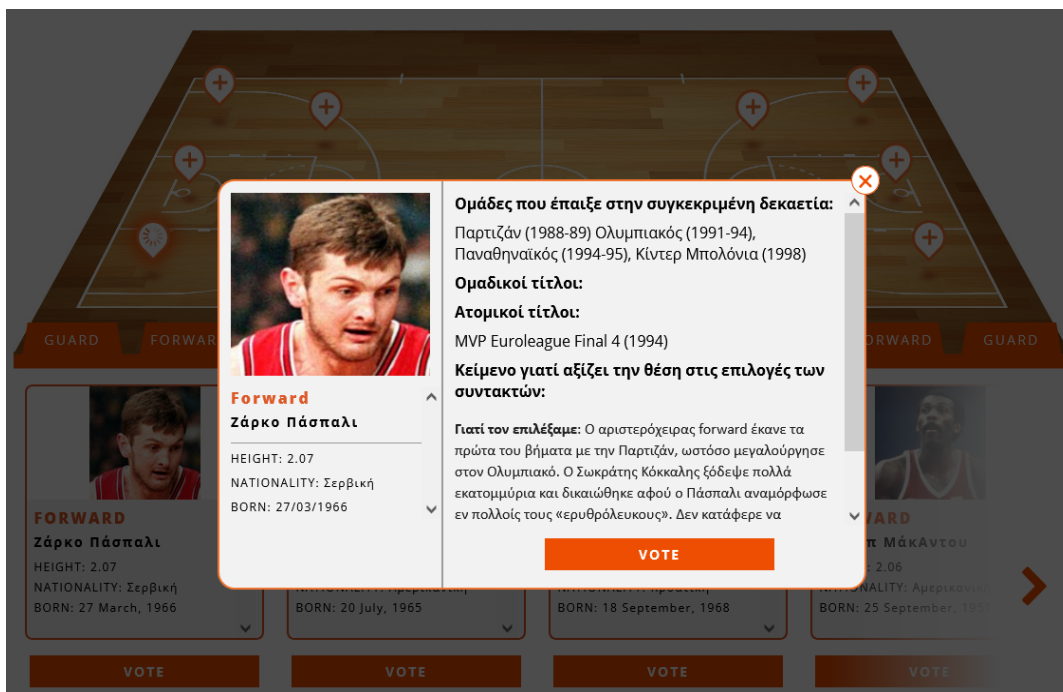
Το Json Object που αφορά τους παίκτες, είναι ομαδοποιημένο αναλόγως την θέση στην οποία παίζει ο εκάστοτε παίκτης. Δηλαδή, για την "Center" θέση, κλειδί-id της οποίας είναι το 103 (βλ. object - position), αντιστοιχούν παίκτες (βλ. object - players) όπως "Άρβιντας Σαμπόνις". Επίσης το επιλεγμένο rip στο γήπεδο είναι συνεχώς ορατό στον χρήστη.

```

▼ positions:
  101: "Guard"
  102: "Forward"
  103: "Center"
▶ lineups: [...]
end_vote: 1604131800
▼ players:
  ▶ 101: [...]
  ▶ 102: [...]
  ▼ 103:
    ▶ 0: {...}
    ▼ 1:
      player_id: "46"
      name: "Άρβιντας Σαμπόνις"
      nationality: "Λιθουανική"
      birthday: "19/12/1964"
      height: "2.21"
      teams: "Ζαλγκίρις (1981-89), Ρεάλ (1992-95)"
      currentTeam: null
      trophies: "1 Euroleague (1995)"
      achievements: "MVP Final 4 (1995), 1ος σκόρερ (1995)"
      ▼ description: "<p><font face=\"Calibri, sans-serif\">
        <b>Γιατί τον επιλέξαμε</b></font><font
        face=\"Calibri, sans-serif\">: Ο
        Σαμπόνις δεν θα μπορούσε να λείπει από
        τη λίστα, αφού είναι ένας από τους πιο
        επιδραστικούς και κυρίαρχους ψηλούς της
        ιστορίας. Έκανε μανικά ποάνυατα το 1995.
      "
```

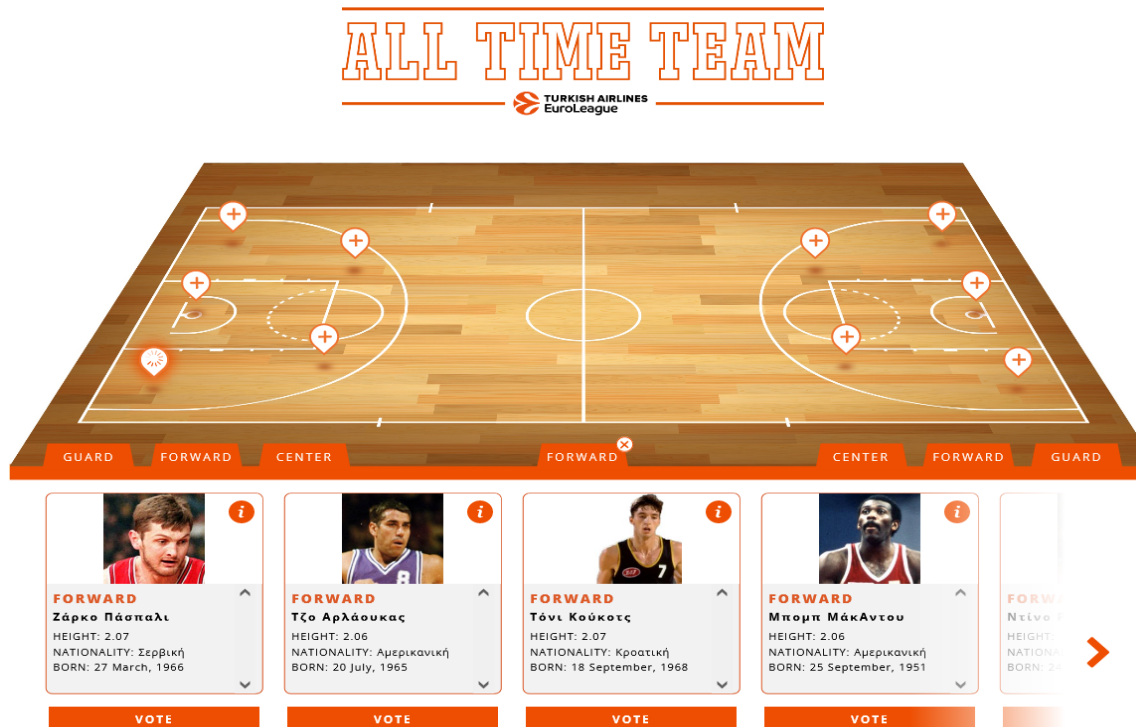
Εικόνα 8. Project JSON Object

Εφόσον ο χρήστης το θελήσει, μπορεί να ενημερωθεί σχετικά με αθλητικές πληροφορίες για τον παίκτη που θέλει, μέσω ενός modal-popup παραθύρου.



Εικόνα 9. Project

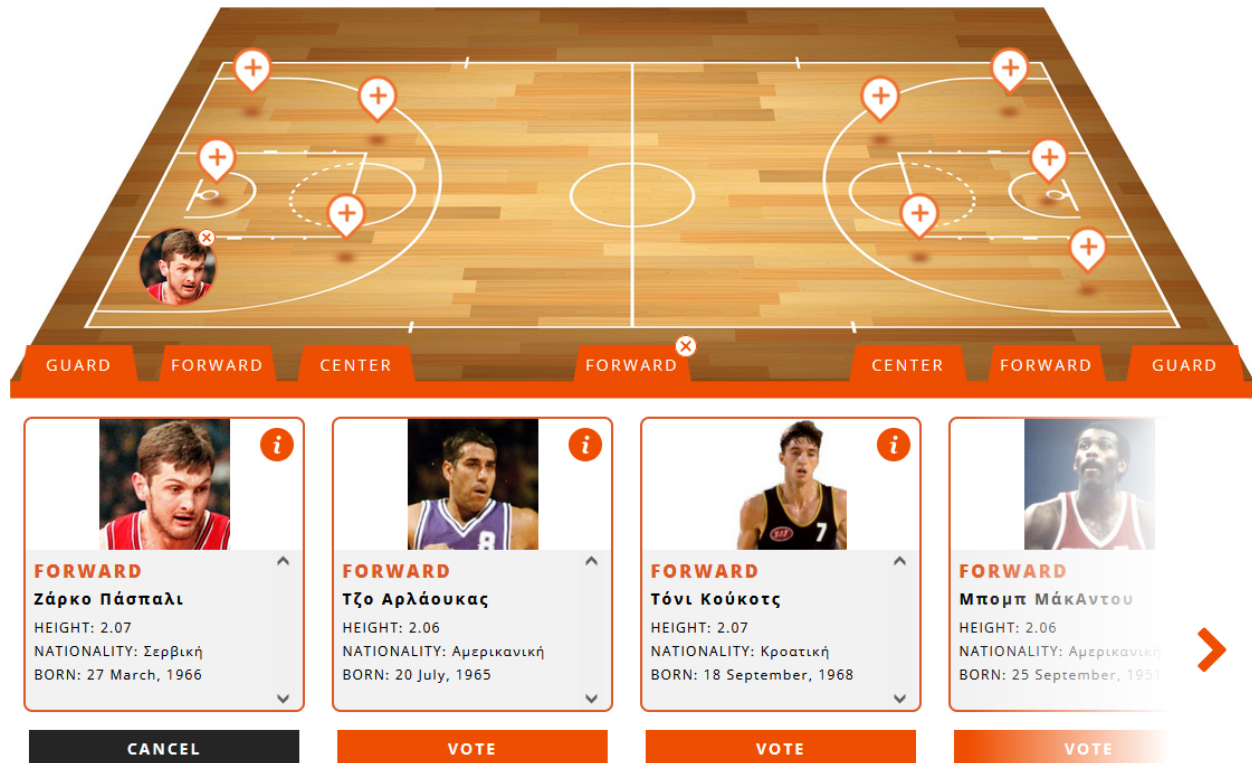
Πρακτικά σημαίνει ότι, εφόσον ο χρήστης επιλέξει μια συγκεκριμένη θέση στο γήπεδο, θα εμφανιστούν μόνο οι παίκτες της αντίστοιχης θέσης, καθώς και η ίδια η θέση (κεντρικά και πάνω από την λίστα των εμφανιζόμενων παικτών), ώστε να γνωρίζει κάθε φορά ο χρήστης την επιλογή που έχει κάνει.



Εικόνα 10. Project

Όταν ο χρήστης επιλέξει τον παίκτη που επιθυμεί, εμφανίζεται στο αντίστοιχο pin η μικρογραφία του επιλεγμένου παίκτη. Η επιλεγμένη θέση (forward) παραμένει ενεργή, για την περίπτωση που ο χρήστης επιθυμεί να αλλάξει την επιλογή του, μέχρις ότου επιλέξει κάποιο διαφορετικό pin.

Εάν ο χρήστης δεν επιλέξει συγκεκριμένο pin, αλλά θέλει γενικότερα κάποιον παίκτη στην γενική θέση πχ του forward, τότε ο επιλεγμένος παίκτης τοποθετείται στην πρώτη διαθέσιμη και ελεύθερη θέση που υπάρχει γενικά στο γήπεδο.



Εικόνα 11. Project

Εφόσον, ολοκληρωθούν όλες οι θέσεις, τότε ο χρήστης είναι έτοιμος να υποβάλει την ψηφοφορία ΤΟΥ.



Εικόνα 12. Project

Μέχρι να πατηθεί το κουμπί της υποβολής, ο χρήστης έχει το δικαίωμα να αλλάξει τις επιλογές του.



Εικόνα 13. Project

Εφόσον υποβληθεί η ψηφοφορία, αποστέλλεται ενά POST request, με τα ids των παικτών στις αντίστοιχες θέσεις και επιπλέον, δημιουργούνται και cookies, όπου δεν επιτρέπει στον χρήστη να ξαναψηφίσει.



Λυπούμαστε, αλλά έχεις ήδη ψηφίσει.

Ευχαριστούμε για την συμμετοχή.

Εικόνα 14. Project

Name	Value	Domain	Path	Expires / Max-Age	Data
Phpstorm...	56810431-f0c4...	localhost	/	Thu, 06 Mar 2031 ...	
gzt_eurol...	{["101":"42"],["1...	localhost	/	Wed, 08 Oct 6279 ...	▼ gzt_euroleague_vote4 : "[["101":"42"),("1..."),("103":"56")]" Created: "Sun, 10 Oct 2021 17:23:36 GMT" Domain: "localhost" Expires / Max-Age: "Wed, 08 Oct 6279 17:23:36 GMT" HostOnly: true HttpOnly: false Last Accessed: "Sun, 10 Oct 2021 17:28:12 GMT" Path: "/" SameSite: "None" Secure: false Size: 151
Phpstorm...	ad5a05dd-a5a...	localhost	/	Thu, 21 Aug 2031 ...	
Phpstorm...	d0caaba2-f9af-...	localhost	/	Fri, 04 Jul 2031 18:...	
Phpstorm...	f14bd579-946...	localhost	/	Thu, 28 Nov 2030 ...	
fblo_157...	y	localhost	/	Sat, 18 Jun 2022 0...	

▼ **gzt_euroleague_vote4**: Array

- ▶ 0: Object
- ▶ 1: Object
- ▶ 2: Object
- ▶ 3: Object
- ▶ 4: Object
- ▶ 5: Object
- ▶ 6: Object
- ▶ 7: Object
- ▶ 8: Object
- ▶ 9: Object
- length: 10
- ▶ __proto__: Array

Εικόνα 15. Cookies set

Όλη η υλοποίηση είναι mobile friendly, ώστε να χρησιμοποιείται εξίσου αποδοτικά από πλευράς front και σε κινητές συσκευές.



Εικόνα 16. Project - mobile

2.3 Αναλυση των **Components**

2.3.1 Stadium Component

Στο κεντρικό component, γίνεται το αρχικό και γενικό fetch των δεδομένων.Επιπλέον γίνεται ο διαχωρισμός των δεδομένων αναλόγως των θέσεων, μέσω του επόμενου component που αφορά την λίστα των παικτών. Στο παρόν component, υλοποιούνται μέθοδοι που αφορούν την τοποθέτηση παίκτη σε συγκεκριμένο pin η γενικά σε οποιαδήποτε θέση. Υπάρχει η λίστα με τις 10 τελικές θέσεις, που συμπληρώνονται κατοπιν χρήσης των υπολοίπων component. Γίνεται η σύνδεση του επόμενου component, που αφορά τον συνδιασμό παίκτη-θέσης. Γίνεται ο έλεγχος αν έχει συμπληρωθεί η ψηφοφορία, πραγματοποιείται η αποστολή του Post Request και γίνεται η επεξεργασία των cookies(έλεγχος αν υπάρχει και δημιουργία).

2.3.2 Player Component

Δημιουργείται η λίστα των παικτών και το animation της πλοήγησης της λίστας και τα αντίστοιχα κουμπα. Δίνεται η εντολή για το modal του κάθε παίκτη με την χρήση του επόμενου component. Με την επιλογή του επιθυμητού παίκτη, ενημερωνεται το βασικό component ώστε να ενημερωθεί η λίστα. Αντιστοιχα γίνεται και σε περίπτωση ακύρωσης της ψήφου.

2.3.3 Lineup Component

Το ενδιάμεσο component.Δημιουργείται 10 φορές(όσες και οι συνολικές θέσεις της ψηφοφορίας) και λειτουργεί αυτόνομα για κάθε θέση επιλογής.Χαρακτηριζεται από το id του κάθε ζευγους, και επικοινωνεί με το βασικό component. Αλλάζει και στην επιλογή και στην ακύρωση.

2.3.4 Player Modal Component

Ανεξάρτητο component, που δέχεται το json object ενός επιλεγμένου παίκτη, και δημιουργεί το modal(propur) των πληροφοριών.

ΚΕΦΑΛΑΙΟ 3

Συμπεράσματα

Από άποψη development, μετά το πέρας του συγκεκριμένου project, καταλήγουμε ότι η vue.js ήταν η καλύτερη επιλογή, καθώς είναι πιο γρήγορη και πιο απλή σε σχέση με άλλα javascript framework που δοκιμάσαμε και αναλύσαμε. Βέβαια, πρέπει να γίνει κατανοητό ότι κάθε framework, έχει τις δικές του δυνατότητες και εξαρτήσεις. Η επιλογή του framework που χρησιμοποιήσουμε, εξαρτάται από τις απαιτήσεις της εφαρμογής που πρόκειται να υλοποιηθεί. Το πρόβλημα που αντιμετωπίστηκε στο συγκεκριμένο project, αφορούσε την αποτύπωση του εικαστικού, που δημιουργούσε δυσκολίες στο responsiveness της εφαρμογής.

ΚΕΦΑΛΑΙΟ 4

Επεκτάσεις για το μέλλον

Η παρούσα εφαρμογή, μπορεί να επεκταθεί και για άλλα αθλήματα με ψυχαγωγικό σκοπό. Μπορούν να γίνουν βελτιώσεις από πλευράς κώδικα, όσον αφορά τις συνδέσεις και τις επικοινωνίες των components.

Βιβλιογραφικές Αναφορές

<https://www.amarinfotech.com/best-javascript-frameworks.html>

<https://da-14.com/blog/2018-comparison-angular-vs-react-js-framework-make-your-project-succeed>

<https://hotframeworks.com/#top-frameworks>

<https://brocoders.com/blog/react-vs-vue-comparison-2021/>

<https://podcasts.google.com/feed/aHR0cHM6Ly9hbmNob3luZm0vcy80NjY3ZDcyMC9wb2RjYXN0L3Jzcmw/episode/MWE0NzYwODAtZjYyNi00MzI5LWE0MjEtZTBmNmU4YjhhMWM1?ep=14>

<https://www.codeinwp.com/blog/angular-vs-vue-vs-react/>

<https://www.simform.com/blog/angular-vs-vue/>

<https://vuejs.org/>

<https://www.jesuisundev.com/en/understand-vuejs-in-5-minutes/>

<https://blog.logrocket.com/introduction-to-vue-lifecycle-hooks/>