ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Μελέτη σύγχρονων JavaScript Frameworks και σύγκριση αυτών στον τομέα των φορμών

Γεώργιος Καλλίνικος Α.Μ.0993

Επιβλέπων καθηγητής: Μιχαήλ Παρασκευάς

Αντίρριο – Δεκέμβριος 2018

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

Αντίρριο, 07/12/2018

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. Εισηγητής: Μιχαήλ Παρασκευάς

2. Σωτήριος Χριστοδούλου

3. Ιωάννης Τζήμας

Αφιέρωση

Στην οικογένεια μου και τους φίλους για τη χρόνια στήριξη τους κατά τη διάρκεια των σπουδών μου.

Ευχαριστίες

Για τη διεκπεραίωση της παρούσας Πτυχιακής Εργασίας, θα ήθελα να ευχαριστήσω τον επιβλέπων καθηγητή, κ. Μιχαήλ Παρασκευά για τη συνεργασία και την πολύτιμη συμβολή του στην ολοκλήρωση της, για την εμπιστοσύνη και τη στήριξη που μου έδειξε.

Επίσης τον κ. Σωτήρη Χριστοδούλου και τον κ. Ιωάννη Τζήμα για τη βοήθεια που μου έδωσαν όλα αυτά τα χρόνια της φοίτησής μου.

Τέλος, θέλω να ευχαριστήσω την οικογένειά μου για την στήριξή τους όλα αυτά τα χρόνια.

Περιεχόμενα

Λίστα Εικόνων

Πρόλογος

Το κίνητρο για τη συγκεκριμένη πτυχιακή εργασία προήλθε από το ενδιαφέρον που παρέχουν οι καινούριες τάσεις στο χώρο του προγραμματισμού στο Διαδίκτυο. Τα τελευταία χρόνια έχουν αλλάξει οι ανάγκες των χρηστών του Διαδικτύου, αντίστοιχα και στο χώρο των προγραμματιστών οι οποίοι προσπαθούν με κάθε τρόπο να βρουν καλύτερα εργαλεία και να κάνουν τα πράγματα πιο εύκολα. Η JavaScript τα τελευταία χρόνια έχει επικρατήσει στο χώρο του διαδικτύου, και αυτό διότι είναι ταχύτατη και μπορεί εφαρμόσει λειτουργικότητες από τη πλευρά του διακομιστή (server) πράγμα που σημαίνει λιγότερος χρόνος αναμονής για το χρήστη. Για αυτό το λόγο δημιουργήθηκαν τα JavaScript frameworks τα οποία έδωσαν απίστευτη δύναμη στους προγραμματιστές να υλοποιούν διαδικτυακές εφαρμογές καθώς και εφαρμογές κινητών με τρομερές δυνατότητες και απίστευτες ταχύτητες. Σκοπός λοιπόν της συγκεκριμένης πτυχιακής εργασίας είναι η ανάλυση αυτών των JavaScript frameworks και η σύγκριση τους, αναφέροντας τα πλεονεκτήματα και τα μειονεκτήματα τους. Βέβαια όπως θα δείτε και παρακάτω όλα τα σύγχρονα frameworks βρίσκονται στο ίδιο επίπεδο. Οπότε η επιλογή κάποιου γίνεται ανάλογα τη περίπτωση και τις ανάγκες της κάθε διαδικτυακής εφαρμογής.

Περίληψη

Η παρούσα πτυχιακή εργασία έχει ως θέμα τη μελέτη των σύγχρονων JavaScript frameworks, τα οποία μονοπωλούν το ενδιαφέρον των προγραμματιστών τα τελευταία χρόνια καθώς η εξέλιξη του διαδικτύου μεγαλώνει με ραγδαίους ρυθμούς. Οι ανάγκες τόσο των χρηστών όσο και των προγραμματιστών για εύρεση εργαλείων τα οποία θα τους βοηθήσουν στη καλύτερη χρήση και πιο εύκολη και καθαρή υλοποίηση αντίστοιχα αυξήθηκε με την πάροδο των χρόνων. Οι άνθρωποι που είναι πίσω από την υλοποίηση αυτών των framewokrs είχαν στόχο να δημιουργήσουν κάποια εργαλεία τα οποία θα έχουν ευανάγνωστο κώδικα, θα μπορούν να τα χρησιμοποιήσουν ξανά στο μέλλον και θα μπορούν να έχουν την ίδια λειτουργικότητα με άλλες εφαρμογές σε άλλες γλώσσες προγραμματισμού με λιγότερο κώδικά και μικρότερο χρόνο αναμονής για το χρήστη.

Κίνητρο για την διεξαγωγή της εργασίας

Με την πάροδο των χρόνων, οι εφαρμογές διαδικτύου γνωρίζουν μεγάλες αλλαγές τόσο ως προς τις νέες τεχνολογίες που συνεχώς εισέρχονται όσο και ως προς τις απαιτήσεις που υπάρχουν από τους χρήστες του διαδικτύου. Για αυτό το λόγο ο κλάδος των προγραμματιστών καλείται να βρει τη βέλτιστη λύση για αυτόν, και αυτή η οποία θα εξυπηρετεί με το καλύτερο τρόπο το χρήστη.

Σκοπός και Στόχοι της Εργασίας

Σκοπός της συγκεκριμένης πτυχιακής εργασίας είναι η μελέτη των σύγχρονων JavaScript Frameworks και η σύγκριση πάνω στις δυνάμεις και στις αδυναμίες τους. Συγκεκριμένα μελετήθηκαν τα frameworks Angular, React και Vue τα οποία είναι τα μεγαλύτερα και πιο δημοφιλή frameworks αυτή τη στιγμή στο διαδίκτυο. Καθώς είναι κατάλληλα για εφαρμογές στις οποίες έχει επίδραση ο χρήστης.

Λέξεις κλειδί: JavaScript Framework, JavaScript, Front-End, Διαδίκτυο, Διαδικτυακή εφαρμογή, Angular, React, Vue

Abstract

This diploma thesis deals with the study of modern JavaScript frameworks, which monopolize the interest of developers in the late years as the development of the internet grows at a rapid pace. The needs of both users and developers to find tools that will help them make better use and more effortless and cleaner implementation respectively have increased over the years. The people behind the implementation of these frameworks were aiming to create some tools that would have easy-to-read code, be able to use them again in the future, and be able to have the same functionality as other applications in other programming languages with less code and shorter waiting times for the user.

Key words: JavaScript Frameworks, JavaScript, Front-End, Web, Web Application, Angular, React, Vue

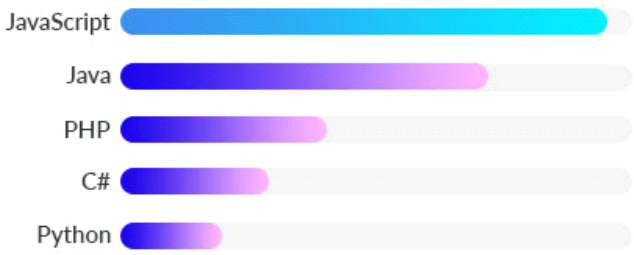
1 ΕΙΣΑΓΩΓΗ

1.1 Τι είναι η JavaScript

Η JavaScript είναι μια δυναμική γλώσσα προγραμματισμού ηλεκτρονικών υπολογιστών. 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. Μια γλώσσα scripting είναι μια ελαφριά γλώσσα προγραμματισμού που υποστηρίζει τη συγγραφή σεναρίων. Σενάρια είναι γραμμές κώδικα που μπορούν να ερμηνεύονται και να εκτελούνται χωρίς μεταγλώττιση.

Ενώ το σύμπαν τεχνολογίας συνεχώς επεκτείνεται, η JavaScript είναι μια εξέχουσα τεχνολογία ιστού που κυριαρχεί στον τομέα. Η ευέλικτη γλώσσα προγραμματισμού μπορεί να εφαρμοστεί στις εφαρμογές του πελάτη (client), του διακομιστή (server), των κινητών και των επιτραπέζιων υπολογιστών και χρησιμοποιείται για να σχεδιάσει-πρώτες διεπαφές, να εμπλουτίσει εφαρμογές ιστού με πολλές λειτουργίες και δυνατότητες, να τροποποιήσει ιστοσελίδες σε πραγματικό χρόνο και πολλά άλλα.

Στην παρακάτω εικόνα θα δούμε τη κατάταξη από γλώσσες προγραμματισμού με τις περισσότερες ερωτήσεις στο StackOverflow (το StackOverflow είναι μία ιστοσελίδα που διαθέτει ερωτήσεις και απαντήσεις σε ένα ευρύ φάσμα θεμάτων στον προγραμματισμό των υπολογιστών).



Εικόνα 1.1 Δημοτικότητα της JavasScript 1

Καταλαβαίνουμε πως η JavaScript είναι η γλώσσα προγραμματισμού η οποία κυριαρχεί στο κόσμο του διαδικτύου τα τελευταία χρόνια και για αυτό το λόγο οι προγραμματιστές ανακαλύπτουν καινούρια Frameworks. 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.

1.2 Χαρακτηριστικά της JavaScript

- Η JavaScript είναι δομημένη γλώσσα χαρακτηριστικό που κληρονόμησε από τη C.
- Είναι δυναμική τόσο στη σύνταξή της, για παράδειγμα: μια μεταβλητή που περιέχει έναν αριθμό μπορεί αργότερα να αποθηκεύσει κείμενο, όσο και στην εκτέλεσή της που είναι run-time.
- Είναι αντικειμενοστραφής, επίσης είναι 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, που είναι γνωστό και ως Δυναμική HTML. Τα script είναι ενσωματωμένα στις HTML σελίδες και αλληλεπιδρούν με το Μοντέλο Αντικειμένου Έγγραφου (DOM – Document Object Model).

1.3 Ιστορία της 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.

LiveScript ήταν το επίσημο όνομα της γλώσσας όταν για πρώτη φορά κυκλοφόρησε στην αγορά σε βήτα εκδόσεις με το πρόγραμμα περιήγησης στο Web, Netscape Navigator εκδοχή 2.0 τον Σεπτέμβριο του 1995.LiveScript μετονομάστηκε σε JavaScript σε μια κοινή ανακοίνωση με την εταιρεία Sun Microsystems στις 4 Δεκεμβρίου, 1995, όταν επεκτάθηκε στην έκδοση του προγράμματος περιήγησης στο Web, Netscape εκδοχή 2.0B3. 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. Κατά συνέπεια, η εταιρεία Microsoft ονόμασε την εφαρμογή της σε JScript για να αποφύγει δύσκολα θέματα εμπορικών σημάτων. JScript πρόσθεσε νέους μεθόδους για να διορθώσει τα Y2K-προβλήματα στην JavaScript, οι οποίοι βασίστηκαν στην java.util.Date τάξη της Java. JScript περιλήφθηκε στο πρόγραμμα Internet Explorer εκδοχή 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 .

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 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 AJAX (Asynchronous JavaScript and XML) είναι ένα σύνολο τεχνικών ανάπτυξης ασύγχρονων εφαρμογών στο Διαδίκτυο που χρησιμοποιεί πολλές τεχνολογίες μαζί από την πλευρά του προγράμματος πελάτη (client). Με το AJAX οι εφαρμογές ιστού μπορούν να στέλνουν και να λαμβάνουν δεδομένα στο παρασκήνιο χωρίς να τους απασχολεί τι προβάλλεται στην σελίδα. Ο διαχωρισμός της ανταλλαγής δεδομένων με την προβολή τους επιτρέπει στην ιστοσελίδα να αλλάζει το περιεχόμενο της δυναμικά χωρίς να χρειάζεται ολοκληρωτική ανανέωση της.

Το 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 δύναται να μετακινούνται και σε μορφή JSON ή XML.

Τον Ιανουάριο του 2009, το έργο CommonJS ιδρύθηκε με στόχο τον καθορισμό ενός κοινού προτύπου βιβλιοθήκης κυρίως για την ανάπτυξη της JavaScript έξω από το πρόγραμμα περιήγησης και μέσα σε άλλες τεχνολογίες.

1.4 Γιατί η JavaScript είναι η κυρίαρχη γλώσσα του Ιστού

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 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.

Ας δούμε μερικά από τα πλεονεκτήματα της JavaScript που την καθιστούν τόσο δημοφιλή στους προγραμματιστές:

- **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 να τη μάθουν:** Η σύνταξη αυτής της γλώσσας είναι παρόμοια με απλά αγγλικά που διευκολύνουν τους προγραμματιστές να μάθουν.
- **Απλή να εφαρμοστεί:** Η δυνατότητα χρήσης της ίδιας γλώσσας front-end και back-end διευκολύνει τους προγραμματιστές.
- **Δεν απαιτεί διαδικασία μεταγλώττισης**: Το πρόγραμμα περιήγησης ερμηνεύει τη JavaScript ως ετικέτες HTML.
- **Σχετικά γρήγορη για τον τελικό χρήστη:** Δεν χρειάζεται, πλέον, οι επισκέπτες να συμπληρώσουν μία ολόκληρη φόρμα και να την υποβάλλουν, για να μάθουν πως υπάρχει κάποιο τυπογραφικό λάθος στο πρώτο πεδίο και ότι θα πρέπει να συμπληρώσουν ολόκληρη τη φόρμα ξανά. Με τη JavaScript, κάθε πεδίο μπορεί να επαληθεύεται καθώς συμπληρώνεται από τους χρήστες, γεγονός που παρέχει άμεση ανατροφοδότηση, όταν αυτοί κάνουν κάποιο λάθος.

1.5 Χρήση της JavaScript

Ορισμένες σημαντικές χρήσεις της JavaScript είναι οι εξής:

Έλεγχος πεδίων: Η JavaScript μπορεί να χρησιμοποιηθεί για την επικύρωση των πεδίων (για παράδειγμα σε μια φόρμα εισόδου το πεδίο όνομα χρήστη). Τα δεδομένα που καταχωρήθηκαν 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. Διαφορετικές ετικέτες HTML μπορούν να δημιουργηθούν με βάση την είσοδο του χρήστη κ.λπ.

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.

1.6 Σύνταξη

```
1  var x;  // ορίζουμε μεταβλητή
2  var y = 7 // ορίζουμε στη μεταβλητή y την τιμή 7
3
```

Εικόνα 1.2 Παράδειγμα Μεταβλητών στη JavaScript

```
1  function sayHello(){
2      alert("Hello, I am inside JavaScript function");
3  }
```

Εικόνα 1.3 Παράδειγμα Συνάρτησης στη JavaScript

1.7 Διαδικτυακή Εφαρμογή (Web Applications)

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 κανάλι επικοινωνίας, αγορών, για εκπαιδευτικούς σκοπούς, συναλλαγές κλπ. Τους επιτρέπει να ανταλλάσσουν πληροφορίες με γρήγορα και αποτελεσματικά.

Οι διαδικτυακές εφαρμογές χρησιμοποιούν έναν συνδυασμό scripts από την πλευρά του server με PHP και ASP καθώς και οι πιο σύγχρονες 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 και άλλα. 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. Ορισμένες από τις εφαρμογές είναι δυναμικές, απαιτώντας επεξεργασία από την πλευρά του server. Άλλες είναι εντελώς στατικές χωρίς επεξεργασία που απαιτείται στο server.

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. Η τεχνολογία server εφαρμογών κυμαίνεται από ASP.NET, ASP έως PHP και Node.js.

Μια κλασσική ροή μιας διαδικτυακής εφαρμογής:

- Ο χρήστης ενεργοποιεί ένα αίτημα στο server ιστού μέσω του Διαδικτύου, είτε μέσω ενός προγράμματος περιήγησης ιστού είτε μέσω εφαρμογής

- Ο server ιστού διαβιβάζει αυτό το αίτημα στον κατάλληλο server εφαρμογών ιστού

- Ο server διαδικτυακής εφαρμογής εκτελεί την απαιτούμενη εργασία – με τα ανάλογα query στη βάση δεδομένων ή επεξεργασία δεδομένων - στη συνέχεια, δημιουργεί τα αποτελέσματα των ζητούμενων δεδομένων.

- Ο server διαδικτυακής εφαρμογής στέλνει αποτελέσματα στον διακομιστή ιστού με τις ζητούμενες πληροφορίες ή τα επεξεργασμένα δεδομένα

- Ο server ανταποκρίνεται πίσω στον πελάτη με τις ζητούμενες πληροφορίες που εμφανίζονται στην οθόνη του χρήστη.

Πλεονεκτήματα των διαδικτυακών εφαρμογών

- Οι διαδικτυακές εφαρμογές εκτελούνται σε πολλαπλές πλατφόρμες ανεξάρτητα από λειτουργικό σύστημα ή συσκευή, εφόσον το πρόγραμμα περιήγησης είναι συμβατό

- Όλοι οι χρήστες έχουν πρόσβαση στην ίδια έκδοση, εξαλείφοντας τυχόν προβλήματα συμβατότητας

- Δεν είναι εγκατεστημένα στον σκληρό δίσκο, εξαλείφοντας έτσι τους περιορισμούς χώρου

- Μειώνουν την πειρατεία λογισμικού σε εφαρμογές ιστού

- 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

Πέρα από τα πλεονεκτήματα όμως οι διαδικτυακές εφαρμογές έχουν και μειονεκτήματα:

- 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.

2 ΑΝΑΛΥΣΗ ΤΩΝ JAVASCRIPT FRAMEWORKS

2.1 Front-End Programming

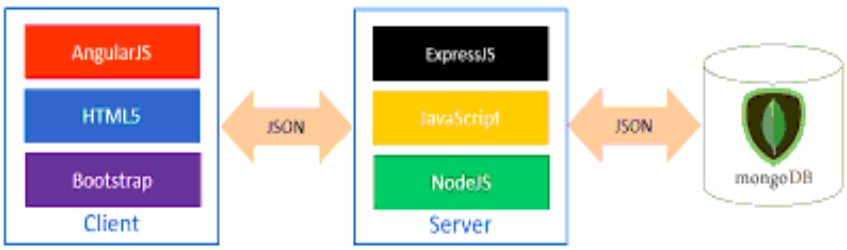
Με τον όρο "front-end" στον προγραμματισμό εννοούμε εκείνο το κομμάτι στο οποίο αλληλεπιδρούν οι χρήστες, δηλαδή αυτό που βλέπουν. Από Γραμματοσειρές, χρώματα εικόνες, μενού, αναδυόμενα παράθυρα μέχρι οτιδήποτε άλλο βλέπουν όταν σερφάρουν σε μια ιστοσελίδα. Όλα τα παραπάνω είναι μια σύνθεση από HTML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

2.2 Node.js

Η αυξανόμενη δημοτικότητα της JavaScript έχει φέρει μαζί της πολλές αλλαγές, και η ανάπτυξη ιστοσελίδων σήμερα είναι εντυπωσιακά διαφορετική. Τα πράγματα που μπορούμε να κάνουμε στο internet σήμερα με τη JavaScript που εκτελείται στον server, όπως και στο πρόγραμμα περιήγησης, ήταν δύσκολο να φανταστούν πριν από μερικά χρόνια.



Εικόνα 2.1 Χρήση Node.js

Στις μέρες μας, 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 Input/Output (I/O) αιτημάτων. Με άλλα λόγια, το Node.js δεν αποβάλλει χρόνο ή πόρους για την αναμονή των αιτήσεων εισόδου / εξόδου (I / O) για επιστροφή. Δηλαδή για κάθε εισερχόμενο αίτημα ή σύνδεση ο server δημιουργεί ένα νέο αίτημα εκτέλεσης ή ακόμα και πιέζει μια νέα διαδικασία για να χειριστεί το αίτημα και να στείλει μια απάντηση. Πράγμα το οποίο προκαλεί μεγάλη επιβάρυνση.

Το Node.js υιοθετεί μια διαφορετική προσέγγιση για την υλοποίηση των εισερχόμενων και εξερχόμενων νημάτων από τον διακομιστή. Εκτελεί έναν βρόχο συμβάντων με μια συγκεκριμένη διαδικασία που έχει καταχωρηθεί με το σύστημα για την διαχείριση συνδέσεων και κάθε νέα σύνδεση προκαλεί την πυροδότηση μιας λειτουργίας επανάκλησης JavaScript. Αυτή μπορεί να χειριστεί I/O αιτήματα με μη αποκλειστικές κλήσεις I/O και αν είναι απαραίτητο μπορεί να κρίνει ώστε να εκτελέσει λειτουργίες αποκλεισμού ή εντατικής χρήσης CPU και να ισορροπήσει φορτία μεταξύ πυρήνων της CPU όπου είναι αναγκαίο. Η προσέγγιση αυτή απαιτεί λιγότερη μνήμη για να χειριστεί περισσότερες συνδέσεις σε σχέση με τις περισσότερες ανταγωνιστικές αρχιτεκτονικές που κλιμακώνονται με νήματα.

Μια κοινή εργασία για έναν server μπορεί να είναι να ανοίξει ένα αρχείο στο server και να επιστρέψει το περιεχόμενο στο χρήστη.

Πώς η PHP ή η ASP χειρίζονται ένα αίτημα αρχείου:

- 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.

Πώς το Node.js χειρίζεται ένα αίτημα αρχείου:

- Στέλνει την εργασία στο σύστημα αρχείων του υπολογιστή.
- Έτοιμο να χειριστεί το επόμενο αίτημα.

- Όταν το σύστημα αρχείων έχει ανοίξει και έχει διαβάσει το αρχείο, ο server επιστρέφει το περιεχόμενο στον χρήστη.

Συμπέρασμα:

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

2.3 JavaScript Frameworks

Καθώς η ανάγκη για συμβατότητα των browsers αύξησε και την επιθυμία της επεξεργασίας στοιχείων της σελίδας από το πρόγραμμα περιήγησης σε πραγματικό χρόνο έγινε μεγάλη ζήτηση η JavaScript και άρχισαν να δημιουργούνται τα frameworks. Τα οποία 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. Αυτά τα frameworks επέτρεπαν τότε στους προγραμματιστές να δημιουργούν συντομεύσεις για να δουλεύουν με στοιχεία στο Document Object Model (DOM).

Καθώς περνούσαν τα χρόνια δημιουργήθηκαν frameworks όπως η Angular.js (2010) η οποία ξαναγράφτηκε από την αρχή και ονομάστηκε Angular το 2016), 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. 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 εφαρμογή για τη βελτίωση της ποιότητας του κώδικα και για να κάνουν την ανάπτυξη του ευκολότερη.

Πολλές λύσεις ανοιχτού κώδικα ήρθαν για να βοηθήσουν τους προγραμματιστές στη δημιουργία αυτών των εφαρμογών μιας σελίδας. Κατά τη διάρκεια μερικών ετών, αυτά τα frameworks έχουν οδηγήσει σε αυξημένο ενδιαφέρον για εφαρμογές μιας σελίδας και γενικά για τη JavaScript.

Αξίζει να σημειωθεί ότι ενώ πολλά από αυτά τα frameworks και βιβλιοθήκες αναπτύχθηκαν ανεξάρτητα, μεγάλες εταιρίες με έδρα στο διαδίκτυο έχουν συμμετάσχει στην προώθηση αυτών των εφαρμογών μιας σελίδας μέσω της κοινότητας ανοιχτού κώδικα, με την Google να αναπτύσσει την Angular και το Facebook και το Instagram να δημιουργούν τη React.

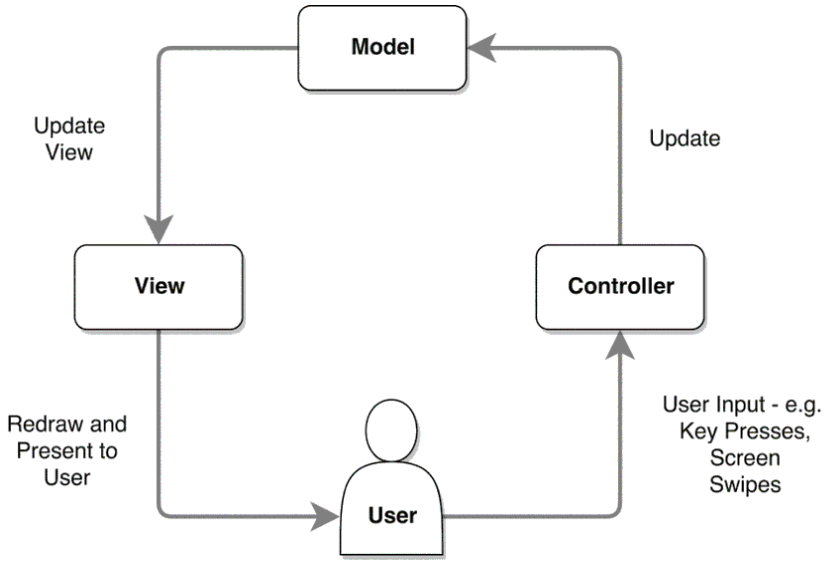
Ο κύριος στόχος τους είναι να δώσουν μία κατεύθυνση στον προγραμματιστή, αποτελώντας τη βάση, μία αρχή πάνω στην οποία θα υλοποιηθεί η εφαρμογή. Αυτό το καταφέρνουν μέσω των εξής χαρακτηριστικών:

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

2.3.1 Model - View - Controller (MVC)

Το αρχιτεκτονικό πρότυπο MVC (Model-view-controller) είναι ένα πολυχρησιμοποιημένο και αποδεδειγμένα επιτυχημένο μοντέλο για να οργανώσεις σωστά ένα project και με μία δομή που οι αλλαγές, τροποποιήσεις, εξελίξεις που μπορεί να χρειαστούν να γίνουν στο μέλλον να είναι εύκολα υλοποιήσιμες. Αναλυτικά:

- **Model** : Περιλαμβάνει όλη τη λογική των δεδομένων και της συσχέτισης με το API
- **View** : Περιλαμβάνει όλη τη λογική της παρουσίασης των δεδομένων
- **Controller** : Περιλαμβάνει όλη τη λογική της αλληλεπίδρασης μεταξύ του προγράμματος και του χρήστη

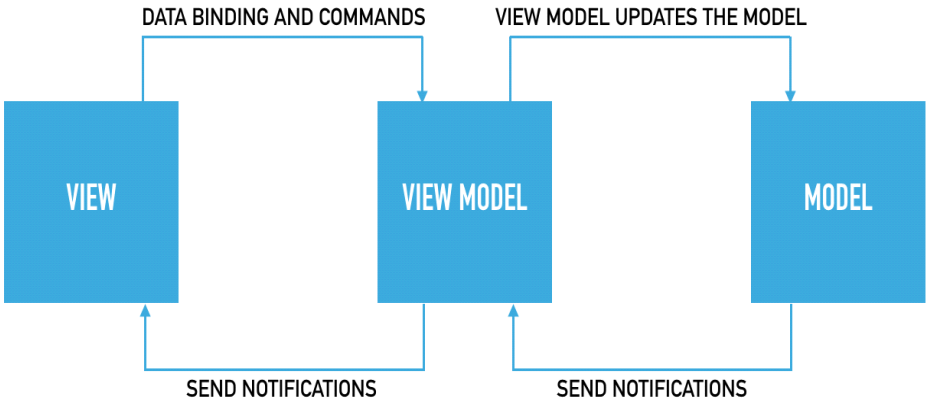


Εικόνα 2.2 Μοντέλο MVC

2.3.2 Model - View - ViewMode (MVVM)

Το Model-View-ViewModel ή MVVM αποτελεί ένα μοτίβο αρχιτεκτονικής λογισμικού, που έχει ως στόχο το διαχωρισμό της ανάπτυξης του γραφικού περιβάλλοντος χρήστη (GUI) από την ανάπτυξη της λειτουργικότητας . Περισσότερο αναλυτικά, το κομμάτι του viewmodel είναι υπεύθυνο για την μετατροπή του περιεχομένου από το model σε μορφή κατάλληλη για τη διαχείριση και παρουσίασή του, και επιβλέπει ολόκληρη ή σχεδόν ολόκληρη τη λογική παρουσίασης του view. Το viewmodel είναι δυνατόν να υλοποιεί ένα μοτίβο «μεσολαβητή» (mediator pattern), οργανώνοντας έτσι την πρόσβαση στο business logic γύρω από ένα σύνολο σεναρίων χρήσης που υποστηρίζονται από το view.

- **Model :** περιλαμβάνει είτε το μοντέλο πεδίου, που αναπαριστά τα πραγματικά δεδομένα σε μία αντικειμενοστραφή προσέγγιση, είτε το στρώμα που έχει πρόσβαση στα δεδομένα
- **View :** περιλαμβάνει την εμφάνιση του περιεχομένου που βλέπει ο χρήστης στην οθόνη.
- **ViewModel :** Περιλαμβάνει μία γενικευμένη διεπαφή του view που αποκαλύπτει τις δημόσιες ιδιότητες και εντολές του. Αντί για τον controller του MVC, το MVVM έχει έναν binder. Ο binder μεσολαβεί για να γίνει η επικοινωνία ανάμεσα στο view και στον data binder, που είναι υπεύθυνος για την αντιστοίχιση και τον συγχρονισμό μεταξύ των πηγών των δεδομένων και των καταναλωτών τους.

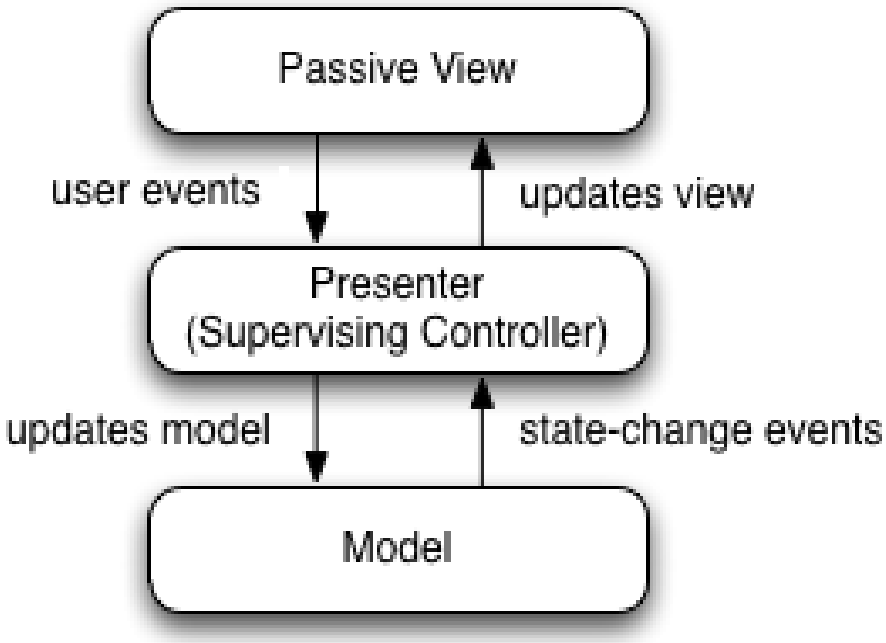


Εικόνα 2.3 Μοντέλο MVVM 1

2.3.3 Model-View-Presenter (MVP)

Το Model-View-Presenter ή MVP αποτελεί ένα μοτίβο αρχιτεκτονικής λογισμικού, εμπνευσμένο από το MVC, το οποίο χρησιμοποιείται κυρίως για την κατασκευή γραφικού περιβάλλοντος χρήστη. Στο MVP, ο presenter παίζει το ρόλο του «διαμεσολαβητή» και όλη η λογική της παρουσίασης των δεδομένων παραχωρείται σε αυτόν.

- **Model:** Περιλαμβάνει τα δεδομένα προς παρουσίαση, δηλαδή το business logic της εφαρμογής.
- **View:** Περιλαμβάνει τα δεδομένα και στέλνει τις εντολές του χρήστη στον presenter μέσω γεγονότων (events
- **Presenter:** Περιλαμβάνει τα δεδομένα από το model και τα επεξεργάζεται κατάλληλα για παρουσίαση στο view.

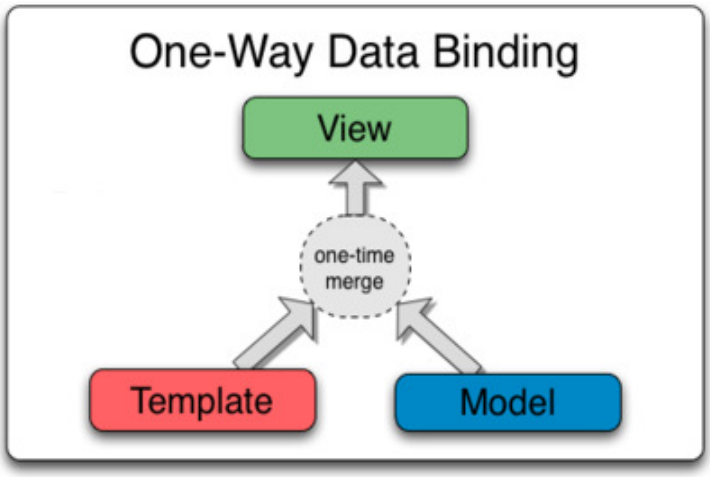


Εικόνα 2.4 Μοντέλο MVP

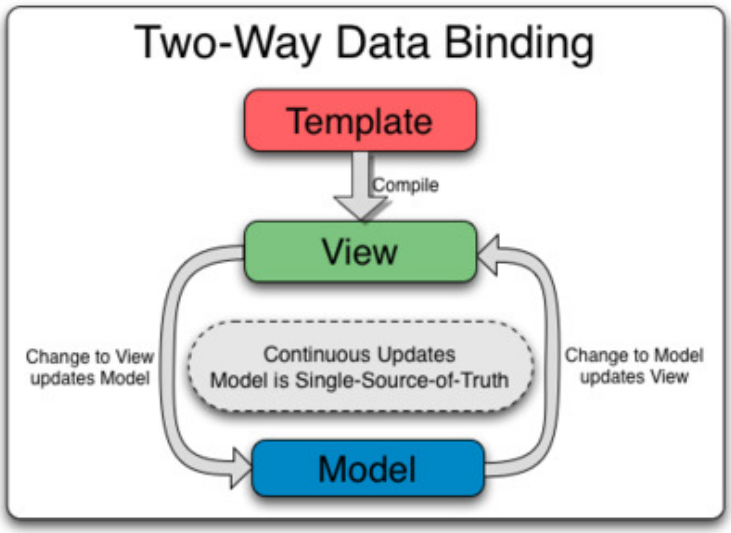
2.4 Data Binding

Στον προγραμματισμό υπολογιστών, η δέσμευση δεδομένων είναι μια γενική τεχνική που συνδέει πηγές δεδομένων από τον πάροχο και τον καταναλωτή μαζί και τις συγχρονίζει. Αυτό γίνεται συνήθως με δύο πηγές δεδομένων / πληροφοριών με διαφορετικές γλώσσες όπως στις δεσμεύσεις δεδομένων XML. Σε δέσμευση δεδομένων UI, αντικείμενα δεδομένων και πληροφοριών της ίδιας γλώσσας αλλά διαφορετικής λογικής λειτουργίας συνδέονται μαζί .

Σε μια διαδικασία σύνδεσης δεδομένων, κάθε αλλαγή δεδομένων αντανακλάται αυτόματα από τα στοιχεία που δεσμεύονται στα δεδομένα. Ο όρος δέσμευση δεδομένων χρησιμοποιείται επίσης σε περιπτώσεις όπου μια εξωτερική αναπαράσταση δεδομένων σε ένα στοιχείο αλλάζει και τα υποκείμενα δεδομένα ενημερώνονται αυτόματα για να αντικατοπτρίζουν αυτήν την αλλαγή.



Εικόνα 2.5 One-Way Data Binding



Εικόνα 2.6 Two-Way Data Binding

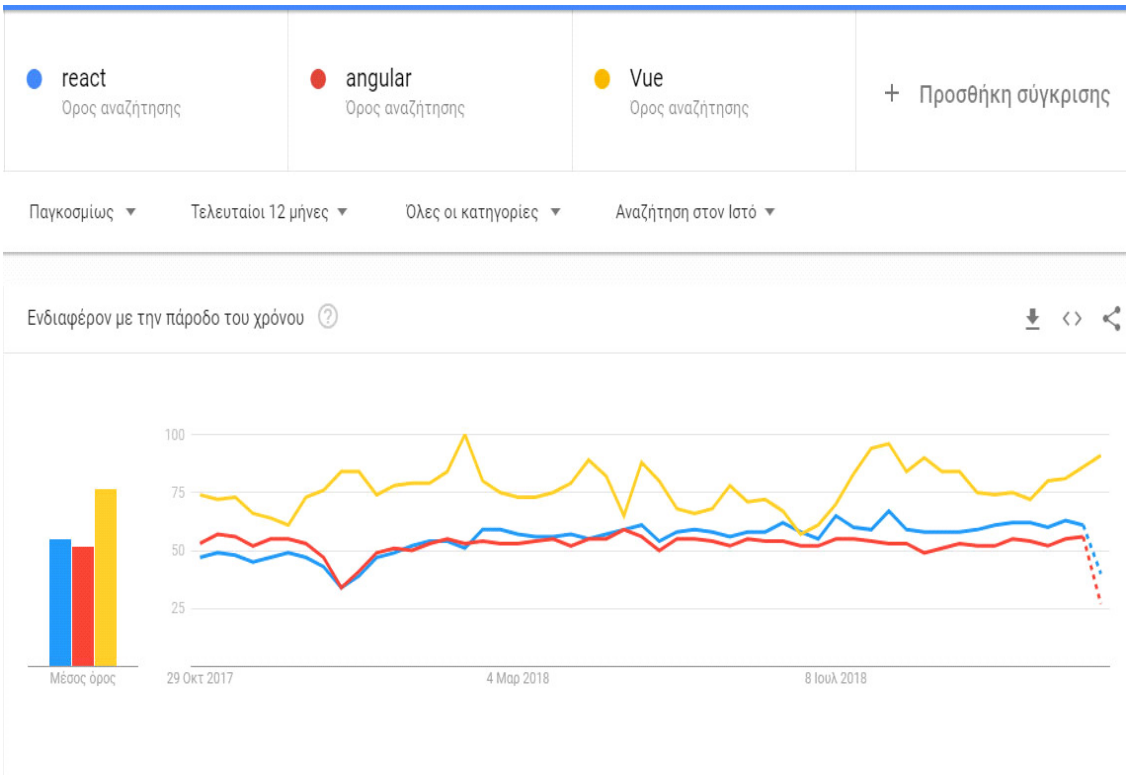
Όπως θα δούμε και παρακάτω μία σημαντική διαφορά ανάμεσα στα frameworks όπως η Angular και η React είναι η αλλαγή δεδομένων ανάμεσα στο model και στο view. Με την Angular να χρησιμοποιεί αμφίδρομο data binding το λεγόμενο two-way-data-binding στην ουσία η διαφορά από το μονόδρομο data binding δηλαδή το one-way-data-binding είναι ότι στο μονόδρομο γίνεται ανανέωση μόνο στο view όταν υπάρξει αλλαγή στο model.

2.5 Δημοφιλή σύγχρονα JavaScript Frameworks

Όπως αναφέρθηκε παραπάνω τα τελευταία χρόνια οι προγραμματιστές ανακαλύπτουν συνεχώς σύγχρονα frameworks τα οποία προσφέρουν τρομερές λειτουργίες και εργαλεία για το χρήστη. Μεγάλες εταιρείες έχουν αφιερώσει αρκετό χρόνο για την υλοποίηση των frameworks και αυτό που βλέπουμε στο χώρο του διαδικτύου και συγκεκριμένα στο κόσμο της JavaScript είναι ότι συνεχώς εξελίσσεται και μέρα με τη μέρα οι προγραμματιστές υλοποιούν κάτι καινούριο που βοηθά τους υπόλοιπους προγραμματιστές. Έτσι τα τελευταία χρόνια η εξέλιξη των JavaScript frameworks είναι ραγδαία, για αυτό το λόγο είναι αδύνατη η ανάλυση όλων αυτών των frameworks. Η συγκεκριμένη μελέτη θα εστιάσει στα περισσότερο δημοφιλή.

Αυτά είναι:

- Angular
- React
- Vue.js



Εικόνα 2.5 JavaScript Frameoworks στις μηχανές αναζήτησης

Στην παραπάνω εικόνα βλέπουμε σύμφωνα με το εργαλείο Google Trends, ότι οι χρήστες τους τελευταίους δώδεκα μήνες κάνουν περισσότερες αναζητήσεις για τη Vue σε σχέση με τα υπόλοιπα δύο framework.

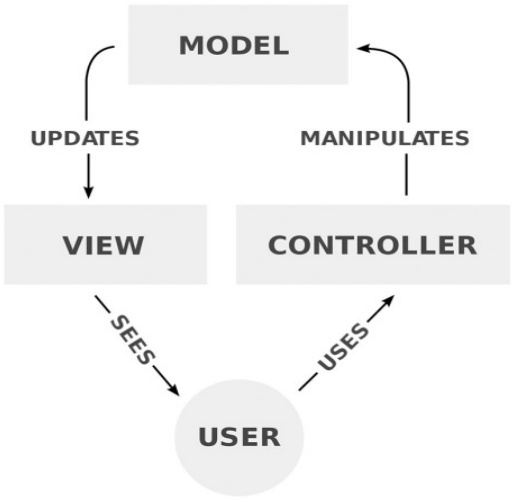
Επίσης στην ιστοσελίδα Github η οποία είναι ένα τεράστιο αποθετήριο, εκεί όπου οι προγραμματιστές αποθηκεύουν τη δουλειά τους η Vue πλέον είναι η γλώσσα με τα περισσότερα downloads με τη React βέβαια να την ακολουθεί από κοντά. Αυτό που αξίζει να σημειωθεί είναι ότι η Angular σε σχέση με τα άλλα δύο framework είναι αρκετά πίσω και αυτό οφείλεται στις τεράστιες αλλαγές που έχουν γίνει στο πυρήνα του framework, με τους προγραμματιστές να μην μπορούν να ακολουθήσουν τις τόσες ενημερώσεις και να στρέφονται στα άλλα framework που η βάση τους είναι σταθερή παρά τις ενημερώσεις που γίνονται.

Framework	Έκδοση	Downloads	Μέγεθος Πακέτων
AngularJS	7.02	42,380	18 MB
React	16.6	114,896	355 kB
Vue	2.5.17	118,312	2.46 MB

2.5.1 AngularJS

Η AngularJS (που συνήθως αναφέρεται ως «Angular» ή «Angular.js») είναι ένα ανοιχτού κώδικα διαδικτυακό framework που διατηρείται κυρίως από την Google και από μία κοινότητα μεμονωμένων προγραμματιστών και εταιρειών για να διευθετήσουν πολλές από τις προκλήσεις που αντιμετωπίζουν κατά την ανάπτυξη εφαρμογών μονής σελίδας (single-page-app).

Έχει ως στόχο να απλοποιήσει τόσο την ανάπτυξη και τον έλεγχο τέτοιων εφαρμογών για αρχιτεκτονικές από την πλευρά του client, model-view-controller /MVC και model-view-view model/MVVM, μαζί με συνιστώσες που χρησιμοποιούνται συνήθως σε πλούσιες εφαρμογές του Διαδικτύου.



Εικόνα 2.6 Μοντέλο λειτουργίας Angular 1

Η AngularJS έγινε αμέσως δημοφιλής στους προγραμματιστές για τους εξής λόγους:

- **Μοντέλο MVC**
- **User Interface**
- **Λιγότερος κώδικας**

Πίσω στο 2010, δημιουργήθηκε η Angular.js (ή αλλιώς Angular1), ήταν ένα εξαιρετικό framework που βοήθησε τους προγραμματιστές να οικοδομήσουν και να διατηρήσουν τεράστιες εφαρμογές (και μικρές), αλλά η Angular1 είχε μια (αποκαλούμενη) ανίατη ασθένεια: καθώς η σελίδα γίνεται μεγαλύτερη (εμφανίζει δηλαδή περισσότερες πληροφορίες) γίνεται πιο αργή και πιο αργή.

Το 2012, όταν οι βιβλιοθήκες JavaScript και τα JavaScript framework μόλις άρχιζαν να ωριμάζουν, η AngularJS έγινε γρήγορα το πιο περιζήτητο framework, πετώντας ψηλά στην εύκολη αμφίδρομη σύνδεση δεδομένων, στην αρχιτεκτονική MVC, σε ένα ενσωματωμένο σύστημα modules, και το πακέτο δρομολόγησης (routing). Το γεγονός ότι το framework διατηρήθηκε από την Google έδωσε μια πρόσθετη ώθηση για να γίνει γρήγορα δημοφιλές.

Ωστόσο, μετά την είσοδο της React, η ομάδα πίσω από την AngularJS το 2014 αποφάσισε να ανανεώσει πλήρως το framework για να παραμείνει ενημερωμένο με τα επερχόμενα framework. Ωστόσο, το νέο framework, που χαρακτηρίζεται ως το Angular 2, δεν ήταν συμβατό με το υπάρχον framework AngularJS. Λόγω αυτού, οι προγραμματιστές χρειάστηκε να ξαναγράψουν τις εφαρμογές τους για να υποστηρίξουν τη νέα έκδοση του framework. Έτσι πολλοί προγραμματιστές έσβησαν την πλατφόρμα και την άλλαξαν σε άλλες πλατφόρμες. Παρόλο που ορισμένοι φανατικοί οπαδοί αποφάσισαν να συνεχίσουν να βλέπουν τα υποσχεθέντα οφέλη του νέου framework, αυτή η καταστροφή ήταν υπεύθυνη για την ξαφνική πτώση του AngularJS και την εμφάνιση της React και άλλων frameworks.

Λόγω της υποστήριξης από την Google και της αφοσίωσης από ένα ευρύ φόρουμ, το frameworks είναι πάντα ενημερωμένο. Επίσης, ενσωματώνει πάντα τις τελευταίες τάσεις της αγοράς.

Ας δούμε πως λειτουργεί η AngularJS:

Η AngularJS λειτουργεί αφού πρώτα διαβαστεί η σελίδα HTML, στην οποία έχουν ενσωματωθεί καινούριες ετικέτες (html tags). Η AngularJS ερμηνεύει αυτά τις ετικέτες ως οδηγίες για να ενώσει τα κομμάτια εισόδου και εξόδου της σελίδας σε ένα μοντέλο που αντιπροσωπεύεται από πρότυπες μεταβλητές JavaScript. Οι τιμές για αυτές τις μεταβλητές JavaScript μπορούν να ρυθμιστούν χειροκίνητα μέσα στον κώδικα ή να ανακτηθούν από αρχεία JSON.

Επίσης, χρησιμοποιεί έναν ειδικό αντικείμενο, το οποίο ονομάζεται scope, και αντιστοιχεί σε ένα model. Οι μεταβλητές που ορίζονται στο scope έχουν πρόσβαση και από το view και από τον controller, είναι δηλαδή ο συνδετικός κρίκος ανάμεσα σ το view με τον controller.

Τα πιο συχνά χρησιμοποιούμενα directives:

- **ng-app**

Υποδηλώνει τη ρίζα (root) της Angular εφαρμογής, κάτω από την οποία τα directives μπορούν να χρησιμοποιηθούν για να δηλώσουν δεσίματα μεταβλητών και συμπεριφοράς.

- **ng-bind**

Θέτει το κείμενο ενός DOM element ως την αποτίμηση μίας έκφρασης.
**
Οποιαδήποτε αλλαγή στην τιμή της έκφρασης αμέσως αντανακλάται στο DOM.

- **ng-model**

Ίδια χρήση με το ng-bind, με τη διαφορά ότι χρησιμοποιεί two-way data binding μεταξύ view και scope.

- **ng-class**

Εφαρμόζει μία κλάση, ανάλογα αν η τιμή αληθείας μίας λογικής έκφρασης (boolean) είναι αληθής ή ψευδής.

- **ng-controller**

Υποδηλώνει έναν JavaScript controller που αποτιμά HTML εκφράσεις.

- **ng-repeat**

Δημιουργεί ένα DOM element για κάθε ένα αντικείμενο μίας συλλογής

- **ng-show & ng-hide**

Αναλόγως αν η τιμή μίας λογικής έκφρασης είναι αληθής ή ψευδής εμφανίζει ή κρύβει το συγκεκριμένο DOM element.

- **ng-view**

Το βασικό directive για το χειρισμό των routes της εφαρμογής που επιστρέφουν δεδομένα σε μορφή JSON.

- **ng-if**

Εάν η if - συνθήκη είναι αληθής (boolean) τότε το DOM element δημιουργείται, διαφορετικά καταστρέφεται.

Πλεονεκτήματα *Angular*

Δέσμευση δεδομένων διπλής κατεύθυνσης:
Ένα από τα βασικά χαρακτηριστικά της Angular είναι η υποστήριξη για ταχύτερη και ευκολότερη σύνδεση των δεδομένων, απαιτώντας έτσι ελάχιστη παρέμβαση από τον προγραμματιστή. Η αμφίδρομη δέσμευση δεδομένων επαναλαμβάνει τις αλλαγές που έγιναν για άμεση προβολή στο μοντέλο και αντίστροφα.

Χειρισμός του DOM:
Σε αντίθεση με άλλα δημοφιλή JavaScript frameworks, η Angular ανακουφίζει άνετα τον προγραμματιστή του ενεργού χειρισμού του DOM (Document Object Model), χάρη στην προσέγγιση αμφίδρομης δέσμευσης δεδομένων. Εξασφαλίζοντας ότι ο προγραμματιστής εξοικονομεί χρόνο και προσπάθειες για κωδικοποίηση ενώ μεταφράζει και ενημερώνει τα στοιχεία DOM.

Two way data binding:
Δεσμευτική δέσμευση με την διεπαφή προγραμματισμού DOM (αυτόματα συγχρονίζει την προβολή και το μοντέλο)

Αρνητικά *Angular*

Διαμοιρασμένη κοινότητα:

Παρά το γεγονός ότι είναι η δεύτερη πιο διαδεδομένη τεχνολογία, η Angular έχει βιώσει μια μεγάλη αναταραχή, με τους προγραμματιστές να στραφούν σε διαφορετικές τεχνολογίες.

Ενημερώσεις:

Οι συνεχείς ενημερώσεις είναι ένα ακόμη σημαντικό πρόβλημα που ωθεί τους προγραμματιστές μακριά από την Angular. Ενώ οι συχνές ενημερώσεις οδηγούν στην αύξηση της συμμετοχής της κοινότητας, οι περισσότερες ενημερώσεις έχουν εισαγάγει σημαντικές αλλαγές στα παραδείγματα που καθιστούν περιττό πρόβλημα για τους προγραμματιστές.

Προβλήματα με τις μηχανές αναζήτησης (SEO):

Προβλήματα με την μηχανές αναζήτησης σε εφαρμογές μιας σελίδας , οι ανιχνευτές δεν μπορούν να διαβάσουν JavaScript και να προσεγγίσουν μόνο στατικές σελίδες.

Αρχικός χρόνος φόρτωσης:

Ένα από τα μειονεκτήματα περιλαμβάνει επίσης το χρόνο που απαιτείται για την απόδοση σελίδων και εφαρμογών που έχουν σχεδιαστεί με τη χρήση Angular, καθώς θα επιβαρυνόταν με

πρόσθετα καθήκοντα όπως ο χειρισμός DOM. Ωστόσο, αυτό είναι πολύ σπάνιο τώρα και εξαιρετικά περιορισμένο στους παλιούς υπολογιστές και συσκευές.

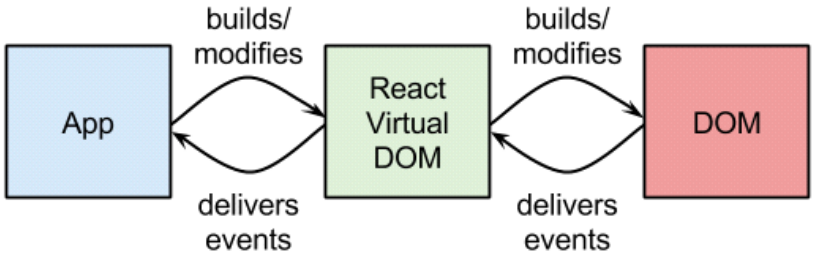
Απότομη καμπύλη εκμάθησης:

Οι νέοι προγραμματιστές στην Angular αντιμετωπίζουν μεγάλη δυσκολία να προσαρμοστούν στο framework. Επιπλέον, η περιορισμένη επίσημη τεκμηρίωση καθιστά ακόμη πιο δύσκολη την εκμάθηση. Ωστόσο, η κοινότητα εργάζεται συνεχώς για τη δημιουργία τεκμηρίωσης για το πλαίσιο.

2.5.2 React

Η React παρουσιάστηκε ως έργο ανοιχτού κώδικα τον Μάιο του 2013. Ο αρχικός προγραμματιστής ήταν ο Jordan Walke, μηχανικός στο Facebook. Η React μπορεί να χρησιμοποιηθεί ως βάση για την ανάπτυξη εφαρμογών μιας σελίδας ή εφαρμογές κινητού. Ένα από τα βασικά του πλεονεκτήματα είναι ότι μπορεί επίσης να λειτουργήσει από πλευρά του διακομιστή (server) μαζί με την πλευρά του χρήστη (client) και μπορούν να συνεργαστούν διαλειτουργικά.

Χρησιμοποιεί επίσης μια έννοια που ονομάζεται εικονικός DOM που εκλεκτικά κάνει subtrees των κόμβων με βάση αλλαγές κατάσταση. Κάνει πολύ λίγα χειρισμούς στο DOM για να διατηρήσει τα στοιχεία ενημερωμένα.



Εικόνα 2.7 Μοντέλο λειτουργίας React

Η React στην ουσία δημιουργεί το δικό του DOM (εικονικό) όπου λειτουργούν τα components. Αυτό δίνει στους προγραμματιστές μεγάλη ευελιξία και εκπληκτικά κέρδη απόδοσης, επειδή υπολογίζει τη μεταβολή που απαιτείται για να γίνει στο DOM εκ των προτέρων . Έτσι αποφεύγει τις δαπανηρές λειτουργίες του DOM και ενημερώνει αποτελεσματικά.

Η React χρησιμοποιεί σύνταξη JSX, ένα συνονθύλευμα JavaScript και HTML. Το JSX απλοποιεί ολόκληρη τη διαδικασία σύνταξης components για τις ιστοσελίδες και η πτυχή HTML επιτρέπει στους προγραμματιστές να εκτελούν λειτουργίες χωρίς να συνδυάζουν συμβολοσειρές.

Μία από τις μεγαλύτερες προκλήσεις με τα frameworks της JavaScript είναι ότι δεν είναι καθόλου φιλικά προς τις μηχανές αναζήτησης. Παρόλο που πρόσφατα έγιναν ορισμένες βελτιώσεις στον τομέα αυτό, δεν ήταν βοήθησε και τόσο. Παραδόξως, η React ξεχωρίζει, καθώς μπορεί να εκτελεστεί στο server και το εικονικό DOM θα επιστραφεί και θα αποτυπωθεί στο πρόγραμμα περιήγησης ως κανονική ιστοσελίδα.

Αυτό που είναι αξιοσημείωτο στη React είναι ότι προσφέρει τη δυνατότητα επαναχρησιμοποίησης των components οποιαδήποτε στιγμή. Αυτό είναι ένα πολύ σημαντικό αποτέλεσμα εξοικονόμησης χρόνου. Τα components στη React είναι απομονωμένα και η αλλαγή σε ένα δεν επηρεάζει τα υπόλοιπα. Αυτό επιτρέπει στους προγραμματιστές να επαναχρησιμοποιήσουν τα components που δεν παράγουν αλλαγές και καθιστούν τον προγραμματισμό πιο ακριβή, εργονομικό και άνετο για αυτούς.

Πλεονεκτήματα *React*

JSX:

Το JSX σημαίνει JavaScript XML. Είναι μια συνταγή XML / HTML που χρησιμοποιεί η React. Επεκτείνει το ECMAScript έτσι ώστε το κείμενο ως XML / HTML να μπορεί να συνυπάρχει μαζί με τον κώδικα αντίδρασης JavaScript. Αυτή η σύνταξη χρησιμοποιείται από τους προεπεξεργαστές όπως η Babel για να μετατρέψει το HTML όπως το κείμενο που βρίσκεται στα αρχεία JavaScript στα πρότυπα αντικείμενα JavaScript.

Virtual DOM:

Ένα εικονικό DOM, όπως και το πραγματικό DOM, είναι επίσης ένα δέντρο κόμβων που αναγράφει στοιχεία μαζί με τα χαρακτηριστικά και το περιεχόμενό τους ως αντικείμενα. Η λειτουργία "Ανταπόκριση απόδοσης" θα δημιουργήσει ένα δέντρο κόμβων έξω από τα στοιχεία της React.

Testability:

Η React επιτρέπει στους προγραμματιστές να χρησιμοποιήσουν τις λειτουργίες προβολής της κατάστασης (μια κατάσταση είναι ένα αντικείμενο που καθορίζει τον τρόπο με τον οποίο το στοιχείο θα εκφραστεί και θα συμπεριφερθεί). Ως εκ τούτου, ο προγραμματιστής μπορεί εύκολα να χειριστεί με την κατάσταση των components που έχουν περάσει στην άποψη του React.

Χρησιμοποιώντας αυτή τη μέθοδο, είναι πολύ απλό να δοκιμάσετε και να εντοπίσετε σφάλματα στις εφαρμογές React.

Εκτέλεση διακομιστή (SSR):

Η εκτέλεση σε πλευρά διακομιστή επιτρέπει να προ-αναπαράγεται η αρχική κατάσταση των αντιδραστικών στοιχείων σας μόνο στην πλευρά του διακομιστή. Με την SSR, η απάντηση του διακομιστή στο πρόγραμμα περιήγησης γίνεται μόνο το HTML της σελίδας που είναι τώρα έτοιμο να αποτυπωθεί. Έτσι, το πρόγραμμα περιήγησης μπορεί τώρα να ξεκινήσει την απόδοση χωρίς να χρειαστεί να περιμένετε να φορτωθεί και να εκτελεστεί ολόκληρη η JavaScript.

Σύνδεση δεδομένων μονής κατεύθυνσης:

Το React ακολουθεί μια ροή δεδομένων μονής κατεύθυνσης, γνωστή και ως σύνδεση δεδομένων ενός τρόπου. Το πλεονέκτημα της σύνδεσης One-Way-Data είναι ότι τα δεδομένα ρέουν σε μία μόνο κατεύθυνση σε όλη την εφαρμογή, γεγονός που δίνει στον χρήστη καλύτερο έλεγχο.

Native εφαρμογές:

Οι Native βιβλιοθήκες κυκλοφόρησαν το 2015 και έδωσαν στη React ένα σημαντικό πλεονέκτημα σε σχέση με τις εφαρμογές για κινητά (Android, iOS)

Μειονεκτήματα *React*

JSX:

Παρόλο που το JavaScript XML είναι ένα αξιοσημείωτο χαρακτηριστικό της React, ορισμένοι προγραμματιστές τείνουν να θεωρούν το JSX ένα σοβαρό μειονέκτημα. Οι προγραμματιστές και οι σχεδιαστές διαμαρτύρονται για την πολυπλοκότητα του JSX και την επακόλουθη απότομη καμπύλη μάθησης.

Έλλειψη καλού οδηγού εκμάθησης:

Η React δεν έχει καλά σχεδιασμένο οδηγό , πράγμα που κάνει τους αρχάριους για αυτή προγραμματιστές να δυσκολεύονται στην εκμάθηση της.

Χρειάζεται απαιτούμενες βιβλιοθήκες:

Το πλαίσιο δεν είναι αρκετά πλήρες και ένας προγραμματιστής χρειάζεται κάποια εμπειρία για να είναι σε θέση να επιλέξει επιπλέον τις απαιτούμενες βιβλιοθήκες.

Δεν είναι MVC:

Η React ασχολείται μόνο με το View του MVC. Για να διατηρήσουμε την κατάσταση και το μοντέλο, πρέπει να χρησιμοποιήσουμε επιπλέον βιβλιοθήκες όπως το Redux.

2.5.3 Vue

Η Vue.js εισήχθη το 2013 και πήρε τα καλύτερα στοιχεία από Ember, React και Angular, βάζοντας όλα αυτά σε ένα εύχρηστο framework. Αποδεικνύεται ότι είναι ταχύτερη και πιο λεπτή, συγκριτικά με τη React και την AngularJS. Αναλυτικά, η Vue.js προσφέρει αμφίδρομη σύνδεση δεδομένων (server-side-rendering), απόδοση server όπως η Angular2 και React.

Ο ιδρυτής της δηλώνει ότι η Vue.js είναι ένα από τα ταχύτερα framework στο σύνολό της. Η Vue.js είναι η καλύτερη επιλογή για γρήγορη ανάπτυξη λύσεων cross-platform. Παρέχει πολύ πιο έξυπνες ενημερώσεις στο DOM, κάνοντας μόνο ό, τι πρέπει να αποδώσει.

Η Vue αναλαμβάνει τις πιο διαδεδομένες τεχνολογίες ιστού και χτίζει πάνω από αυτές για να είναι βολικό για τους χρήστες. Παρόλο που η React έχει μια απότομη καμπύλη εκμάθησης JSX (JavaScript + HTML αρχεία), η Vue χρησιμοποιεί απλά πρότυπα βασισμένα σε HTML και συστατικά ενός αρχείου. Επιτρέποντας στους προγραμματιστές να γράψουν το πραγματικό CSS (με υποστήριξη για τις λειτουργικές μονάδες CSS και τους προ-επεξεργαστές). Λόγω της μικρότερης πολυπλοκότητας και, πάλι, ενός μικρότερου μεγέθους κατασκευής, όπως γίνεται έξω από τις βιβλιοθήκες JavaScript.

Πλεονεκτήματα *του Vue.js*

Μέγεθος:

Ένας από τους λόγους, που η Vue.js έγινε πολύ δημοφιλές αμέσως είναι λόγω του μικρού μεγέθους αρχείο. Η βιβλιοθήκη είναι μόλις 18kb μετά το gzipping. Λόγω αυτού, οι χρήστες της Vue.js μπορούν να διαχωρίσουν τον μεταγλωττιστή template-to-virtual-DOM, μειώνοντας έτσι τον χρόνο εκτέλεσης. Ωστόσο, παρά το μέγεθός του, το Vue.js είναι σταθερά γνωστό ότι έχει ξεπεράσει τα ογκώδη πλαίσια όπως τα Angularκαι EmberJS.

Ευκολία υλοποίησης:

Η ανάπτυξη μεγάλων προτύπων είναι αρκετά απλή με τη Vue.js. Κάνοντας το σχετικά εύκολο για προγραμματιστές, ενώ παράλληλα εξοικονομούν χρόνο. Επίσης, η απλή δομή καθιστά πολύ εύκολη την αναζήτηση μπλοκ που περιέχουν σφάλματα, μειώνοντας έτσι τον απαιτούμενο χρόνο και προσπάθεια.

Απλή ενσωμάτωση:

Η Vue.js μπορεί να χρησιμοποιηθεί για την κατασκευή ολόκληρων εφαρμογών μιας σελίδας καθώς και για τη συνεισφορά στοιχείων σε υπάρχουσες εφαρμογές. Η βιβλιοθήκη μπορεί να χειριστεί τη δομή, τη λογική και το στυλ μιας συνιστώσας, όλα σε ένα αρχείο. Αυτό καθιστά πολύ απλό να δημιουργήσετε εύκαμπτα εξαρτήματα που μπορούν να επαναχρησιμοποιηθούν σε άλλα έργα. Η βιβλιοθήκη μπορεί επίσης να χρησιμοποιηθεί για την προσθήκη μικρών αντιδρώντων συστατικών σε υπάρχον πρότυπο.

Οδηγός Εκμάθησης:

Η Vue.js έχει απίστευτα καλά σχεδιασμένο οδηγό εκμάθησης που είναι πολύ εμπεριστατωμένος και καλά γραμμένος. Όλοι οι αρχάριοι απαιτούν να γράψουν την πρώτη τους εφαρμογή είναι μόνο μερικά βασικά JavaScript και HTML.

Μειονεκτήματα *του Vue.js*

Μικρή κοινότητα:

Όντας ένα σχετικά νέο πλαίσιο JavaScript, η Vue.js εξακολουθεί να μην έχει την ευρεία υποστήριξη όπως με η Angular και η React. Επιπλέον, υπάρχουν περισσότεροι κοινοτικοί πόροι για τη React από τη Vue.js. Ωστόσο, με την αυξανόμενη δημοτικότητα, αναμένουμε ότι η κατάσταση θα αλλάξει σύντομα.

Ευκαμψία:

Ενώ η μεγάλη ευελιξία γίνεται μερικές φορές απαραίτητη από τους προγραμματιστές, η πληθώρα επιλογών της Vue.js μπορεί να απαιτηθεί όταν αναπτυχθεί σε μεγαλύτερα έργα που περιλαμβάνουν πολλούς προγραμματιστές.

3 ΣΥΓΚΡΙΣΗ ΤΩΝ ΕΠΙΛΕΓΜΕΝΩΝ FRAMEWORK

3.1 Διαφορές Λειτουργικότητας

Σε αυτό το κεφάλαιο θα γίνει ανάλυση των βασικών διαφορών λειτουργίας του κάθε frameowork, σκοπός είναι η εύρεση των διαφορών και η καταγραφή τους.

3.1.1 Η χρησιμότητα των Components

Και τα τρία framework χρησιμοποιούν τη λειτουργικότητα των Components. Για τη React επεκτείνετε το React.Component, στην Angular ρυθμίζουμε ένα module για να καλύψει κάποια στοιχεία με τους @NgModule και @Component decorators, στη Vue χρησιμοποιείτε το Vue.component () για να καταχωριστούν τα στοιχεία.

Τα πάντα βασίζονται γύρω από τα components, συνδέονται μεταξύ τους, μεταδίδουν δεδομένα μεταξύ τους και ούτω καθεξής.

Ως μοντέλο σύνθεσης, τα components προορίζονται να είναι αυτοτελείς ενότητες ή "κομμάτια" της εφαρμογής, τα οποία μπορούν στη συνέχεια να επαναχρησιμοποιήθουν σε πιο συγκεκριμένα περιβάλλοντα. Το μεγάλο πράγμα που επιτρέπουν είναι ένας τρόπος να ενσωματωθεί η λογική, παρέχοντας εγγυήσεις API: περνάνε δηλαδή μεταβλητές στα components.

3.1.2 State και μεταλλαγές

Το πρόβλημα που αντιμετωπίζουν όλα αυτά τα framework είναι η δέσμευση δεδομένων στο DOM με κάποιο τρόπο. Αυτό είναι κάτι που ο προγραμματιστής θα πρέπει να κάνει χειροκίνητα στο jQuery για παράδειγμα. Αυτό σημαίνει ότι η πιο βασική εφαρμογή που χρησιμοποιεί ένα framework θα κρατήσει κάποια κατάσταση (state). Τα μοντέλα που εκθέτουν οι Vue, Angular και React είναι στην πραγματικότητα αρκετά διαφορετικές.

Η Angular έχει την πεποίθηση ότι το state θα πρέπει να είναι μεταβλητή. Έχει επίσης τις δυνατότητες για τη διέλευση των υπηρεσιών μέσω των components και των modules, διατηρώντας συνήθως το state αυτή ως μονόδρομο μέσω της έγχυσης εξάρτησης. Ένας προγραμματιστής μπορεί επομένως να γράψει εύκολα ένα container ανταλλαγής δεδομένων που θα ενημερώνει τα σχετικά components, συνήθως μέσω της υπηρεσίας που επιστρέφει τους παρατηρητές και τα components που αποθηκεύουν συνδρομές σε αυτά.

Η Vue χρησιμοποιεί ένα σύστημα αντιδραστικότητας για να ειδοποιήσει τα άλλα μέρη της εφαρμογής ότι έχει συμβεί κάποια αλλαγή στο state. Αυτό δίνει ένα πλεονέκτημα από τη χρήση αυτού. Η ιδιότητα είναι στην πραγματικότητα χρησιμοποιώντας ένα setter “υπόγεια”, σε αυτό το setter, η Vue μπορεί να στείλει ενημερώσεις όπου απαιτείται, και όχι μόνο να τους στείλει παντού. Ο προτιμώμενος μηχανισμός για τη σύνταξη της κατάστασης στο πρότυπο είναι υπολογισμένες ιδιότητες.

Η React έκανε την έννοια της μεταβλητής state ευρύτερα διαδεδομένη στο οικοσύστημα της JavaScript. Η κατάσταση δεν ενημερώνεται με μετάλλαξη (π.χ. χρησιμοποιώντας το state.myProperty), αντιθέτως υπάρχει η μέθοδος setState που ενημερώνει τα δεδομένα σε κάθε component.

Η ενθυλάκωση που παρέχουν τα components, ωστόσο, σημαίνει ότι η διαφορά μεταξύ των ιδιοτήτων της διαχείρισης του state δεν είναι τόσο προφανής όταν χρησιμοποιούμε όλα αυτά τα frameworks.

Το προτιμώμενο μοτίβο και στα 3 frameworks είναι να αποφεύγεται η άμεση μετάλλαξη δεδομένων που διαβιβάζονται από έναν γονέα με σκοπό την ενημέρωση του εν λόγω γονέα ότι πρέπει να συμβεί μια αλλαγή στο state.

3.1.3 Μετάβαση δεδομένων

Τα πρότυπα μετάδοσης δεδομένων απλοποιούνται με μια εφαρμογή βασισμένη σε στοιχεία: η επικοινωνία γίνεται μόνο από τον γονέα στο παιδί και το αντίστροφο.

Στη React, διαβιβάζονται props (pass data) για να περάσουν δεδομένα αλλά και λειτουργίες που σας επιτρέπουν να ενημερώσετε το μητρικό state από το παιδί.

Στην Angular συνδέσεις εισόδου και εξόδου ορίζονται στο στοιχείο και δεσμεύονται στο πρότυπο. Οι έξοδοι συμπεριφέρονται σαν γεγονότα στο ότι εκπέμπονται από το παιδί και ακούγονται από τον γονέα.

Στο Vue, τα props μεταφέρονται από τον γονέα στο παιδί και το παιδί μπορεί να εκπέμψει τα γεγονότα πίσω στον γονέα.

Ο τρόπος μεταβίβασης δεδομένων μεταξύ των "αδελφών" components επιλύεται με τον ίδιο τρόπο σε όλα αυτά τα frameworks με την εξεύρεση του πλησιέστερου κοινού γονέα στο δέντρο και την ενθυλάκωση του state.

3.1.4 Lifecycles, updates and re-render

Τα components σε React, Vue και Angular ενημερώνονται εάν αλλάξει η τοπική κατάσταση (state) ή τα props. Εάν δεν αποθηκεύσετε οποιαδήποτε κατάσταση σε τοπικό επίπεδο, θα μπορούσατε να αναγκάσετε τα εξαρτήματα να αλλάξουν μόνο όταν αλλάζουν τα props τους. Τα λειτουργικά components το κάνουν για τη React και τη Vue και η στρατηγική ανίχνευσης ChangeDetection.On change μπορεί να χρησιμοποιηθεί στην Angular.

Και τα τρία framework παρέχουν μια συναρμολόγηση / αποσυναρμολόγηση που αναφέρεται στο component που αρχικοποιείται στο DOM και δεν χρειάζεται πια.

3.1.5 Γενική Αποθήκη (Global Store)

Όταν μια εφαρμογή χρειάζεται κοινόχρηστα δεδομένα σε components που είναι αρκετά απομακρυσμένα στο δέντρο συνιστωσών, ήρθε η ώρα να χρησιμοποιήσετε μια αποθήκη (store). Αυτή η λειτουργικότητα έγινε αρχικά δημοφιλής από το οικοσύστημα της React με την αρχιτεκτονική ροής του Facebook. Η ροή συνίσταται στη μεταβίβαση ενεργειών στην αποθήκη που γνωρίζει πώς να ενημερώνει την κατάσταση (state) του component ανάλογα με τον τύπο ενέργειας.

Στη React οι επιλογές είναι redux ή MobX, η Vue έχει την επίσημα υποστηριζόμενη Vuex και η Angular έχει ngrx / store.

Αυτό το ενιαίο πρότυπο παγκόσμιας αποθήκευσης υποστηρίζεται από όλα αυτά τα framework. Η διαφορά είναι ότι στη React και στην Angular για να γίνει εφικτή αυτή η λειτουργικότητα πρέπει να εγκαταστήσεις εξωτερική βιβλιοθήκη. Ενώ η αποθήκη της Vue υποστηρίζεται από την κεντρική ομάδα.

Ο σχεδιασμός αυτών των λύσεων αποθήκευσης "goto" αντικατοπτρίζει μερικούς από τους ιδιωματικούς τρόπους για να γράψουμε κώδικα με αυτά τα framewors: στη Vue είναι αντικείμενο και getter / setter που βασίζονται, σε Angular it's Observables, καλά οργανωμένα αποτελέσματα και ούτω καθεξής. Η React έχει την μεγαλύτερη επιλογή, μόνο "JavaScript" + καθαρές λειτουργίες (redux), αντιδραστικοί παρατηρητές (MobX) ή ακόμα και απλά αντικείμενα JavaScript (ασταθής).

3.1.6 Τύποι και επικύρωση δεδομένων

Η επικύρωση των δεδομένων μέσα σε μια εφαρμογή είναι χρήσιμη για την ανάπτυξη και την αποσφαλμάτωση. Η JavaScript δεν πληκτρολογείται στατικά έτσι ώστε να παρακολουθεί τα πεδία, ενώ γίνεται μετάδοση δεδομένων μέσω πολλαπλών στοιχείων και λειτουργιών μπορεί να γίνει δύσκολη.

Η Vue και η React επιλύουν αυτό το θέμα με επικύρωση τύπου prop. Ένα component και τα props του ορίζονται με τύπους props, σε κατάσταση ανάπτυξης, η βιβλιοθήκη θα ελέγξει ότι τα περασμένα props ταιριάζουν με τα καθορισμένα props types. Μπορούν επίσης να χρησιμοποιηθούν για να επωφεληθούν από τα συστήματα τύπου όπως το TypeScript και Flow, τα οποία τους προσφέρουν παρόμοιες εγγυήσεις με την ανάπτυξη σε Angular και TypeScript.

Η Angular δεν έχει τέτοιο μηχανισμό επικύρωσης προτύπων, αλλά έχει το πλεονέκτημα ότι γενικά γράφεται σε TypeScript. Η εμπειρία ανάπτυξης που έχει στατικά πληκτρολογήσει

εισόδους και εξόδους είναι μεγάλη. Δεν είναι δυνατή η ανίχνευση αναντιστοιχιών τύπου κατά το χρόνο εκτέλεσης. Η πλειονότητα αυτών των αναντιστοιχιών τύπου καταλήγουν να συμβαίνουν στον IDE ή στον μεταγλωττιστή.

3.1.7 Πρότυπα, στυλ και εργαλεία

Οι βέλτιστες πρακτικές γύρω από τη δομή του αρχείου μιας εφαρμογής διαφέρουν μεταξύ των framework.

Η Angular στρέφεται προς ένα φάκελο ανά Module /Component όπου ζουν τα αρχεία TypeScript, το πρότυπο και το στυλ (css). Τα πρότυπα και τα στυλ μπορούν να γραφούν εν σειρά στη Angular αλλά η καλύτερη πρακτική είναι να έχετε ξεχωριστά αρχεία. Αυτή είναι μια καλή ιδέα για μεγάλες εφαρμογές μιας σελίδας.

Η Vue ονομάζεται "το προοδευτικό framework" επειδή προσφέρει διαφορετικά χαρακτηριστικά ανάλογα με το μέγεθος της αναπτυσσόμενης εφαρμογής. Στην απλούστερη περίπτωση (η Vue περιλαμβάνεται παγκοσμίως χρησιμοποιώντας μια ετικέτα CDN και μια ετικέτα (tag) script, ενθαρρύνεται η σύνταξη γραμματοσειρών σε έντυπη μορφή. Το Vue προσφέρει επίσης ένα CLI και πακέτα που ενσωματώνονται με εργαλεία κατασκευής όπως webpack. Ο προτιμώμενος τρόπος για την εγγραφή components σε αυτό το περιβάλλον είναι το συστατικό ενός αρχείου ένα αρχείο με ένα πρότυπο, ένα σενάριο και μια ετικέτα στυλ (style tag). Ο Vue-loader μεταγλωττίζει το πρότυπο σε JavaScript μαζί με την ενότητα δέσμης ενεργειών και εξάγει τα περιεχόμενα της ετικέτας στυλ σε ένα φύλλο στυλ κατά την κατασκευή.

Στη React, δεδομένου ότι τα λογικά και τα πρότυπα JSX δεν μπορούν να διαχωριστούν, υπάρχει μόνο το θέμα των στυλ. Υπάρχουν πολλές λύσεις: ξεχωριστά αρχεία styles, webpack για την εξαγωγή των αρχείων "style.css" ή τις βιβλιοθήκες CSS-in-JS.

Ανάλογα το μέγεθος του κάθε έργου, η Vue έχει την ωραιότερη εργονομία, για μικρά έργα. Η Angular έχει την μεγαλύτερη δομή οπότε είναι ικανή για μεγαλύτερα έργα, και η React κάθεται ενδιάμεσα, όπου η δομή components αφήνεται ως άσκηση στον προγραμματιστή του έργου.

3.1.8 Δοκιμή και απόδοση διακομιστή (Server-side Rendering)

Ο έλεγχος μονάδων στην Angular γίνεται κυρίως στις κλάσεις των components του TypeScript. Για να δοκιμαστεί η λογική του προτύπου θα απαιτούσε ένα πλήρες περιβάλλον DOM. Στη React και στη Vue, χάρη στη χρήση ενός Virtual DOM και λειτουργιών rendering, είναι δυνατή η δοκιμή της λογικής του προτύπου με χρήση enzyme και vue-test-utils αντίστοιχα. Η απόδοση των components σημαίνει ότι μόνο το πρώτο "layer" των παιδιών του component αποδίδεται, δηλαδή. όλα τα στοιχεία που βρίσκονται στα παιδιά δεν έχουν αξιολογηθεί πλήρως (αποδίδονται σε HTML), αλλά παραμένουν ως ComponentName στο δομικό στοιχείο. Αυτή η δυνατότητα αναπαραγωγής χωρίς ένα πλήρες περιβάλλον DOM είναι επίσης χρήσιμη για την απόδοση από

την πλευρά του διακομιστή της εφαρμογής JavaScript. Η Vue διαθέτει το πακέτο vue-server-renderer και το ReactDomServer. Αυτά επιτρέπουν σε μια εφαρμογή κόμβου να μετατρέψει μια εφαρμογή Vue ή React στη σήμανση (HTML) που συνήθως αποστέλλεται πίσω ως απόκριση HTML για το φορτίο πρώτης σελίδας.

Η Angular έχει παρόμοια ικανότητα σύνθεσης και απόδοση, αν και αυτά τα χαρακτηριστικά είναι λιγότερο "drop-in σε μια υπάρχουσα εφαρμογή" και περισσότερο "κατασκευάστε με αυτά τα εργαλεία στο μυαλό" δεδομένου ότι ορισμένα από τα χαρακτηριστικά του framework πρέπει να χρησιμοποιούνται με προσοχή κατά τη χρήση του Angular Universal, την απόδοση από την πλευρά του διακομιστή ή τον μεταγλωττιστή AoT του Angular, ο οποίος μεταγλωττίζει τα πρότυπα σε λειτουργίες JavaScript και rendering.

3.2 Σύγκριση Angular και React

Μια μεγάλη διαφορά μεταξύ Angular και React είναι ένας τρόπος έναντι αμφίδρομης δέσμευσης (one way VS two-way binding). Η αμφίδρομη δέσμευση της Angular αλλάζει την κατάσταση μοντέλου όταν ενημερώνεται το στοιχείο UI. Η React προτιμά με έναν τρόπο: ενημερώνει πρώτα το μοντέλο και στη συνέχεια καθιστά τα στοιχεία UI. Η μέθοδος της Angular είναι καθαρότερη στον κώδικα. Ο τρόπος της React έχει ως αποτέλεσμα καλύτερη επισκόπηση των δεδομένων, επειδή τα δεδομένα ρέουν μόνο προς μία κατεύθυνση, γεγονός που καθιστά ευκολότερη και ταχύτερη τη διαδικασία εντοπισμού σφαλμάτων.

Επίσης η προσαρμογή σε HTML , η Angular χρησιμοποιεί μια ειδική σύνταξη που σημαδεύει το HTML για να επιτύχει το απαραίτητο σχέδιο. Αυτή είναι η προβολή σε MVC.

Η React χρησιμοποιεί μια νεότερη ιδέα που ονομάζεται JSX ή JavaScript XML. Η JSX τοποθετεί την απαιτούμενη HTML μέσα στο JavaScript και επιτρέπει να χρησιμοποιηθεί απλό JavaScript για μεταβλητές προτύπων ή επαναλαμβανόμενα component. Αυτή δηλαδή που στην Angular το κάνει το View σε MVC.

Ένα ακόμα στοιχείο που διαφέρουν τα δύο αυτά framework είναι το πώς χρησιμοποιούν και με ποιον τρόπο το CSS, δηλαδή πως γίνεται το style των components.

Η Angular έχει δημιουργήσει μια εύκολη μέθοδο για την κάλυψη του CSS σε ένα συγκεκριμένο component. Αυτό είναι ιδιαίτερα χρήσιμο γιατί δημιουργείται ξεχωριστό αρχείο CSS για το συγκεκριμένο component. Αυτό είναι ένα χαρακτηριστικό που βρήκα ιδιαίτερα χρήσιμο στην Angular σε σχέση με τη React. Στην οποία ακόμα δεν έχει ξεκαθαρίσει ο καλύτερος τρόπος γραφής CSS για κάθε συγκεκριμένο component καθώς χρειάζονται εξωτερικές βιβλιοθήκες.

3.3 Σύγκριση Vue και React

Τόσο η React όσο και η Vue είναι εξαιρετικά ταχύτατα frameworks, επομένως η ταχύτητα είναι απίθανο να είναι ένας αποφασιστικός παράγοντας για την επιλογή μεταξύ τους.

Στη React, όταν αλλάζει η κατάσταση (state) ενός component, φορτώνουν όλα τα components από την αρχή. Ενώ στη Vue, οι εξαρτήσεις ενός component παρακολουθούνται αυτόματα κατά τη διάρκεια της απόδοσης του, έτσι το σύστημα ξέρει με ακρίβεια ποια components πρέπει πραγματικά να ξαναδοκιμάσουν όταν αλλάζει η κατάσταση (state).

Στη React, όλα είναι μόνο JavaScript. Όχι μόνο η HTML εκφράζεται μέσω JSX, οι πρόσφατες τάσεις τείνουν επίσης να θέτουν το CSS μέσα στη JavaScript. Αυτή η προσέγγιση έχει τα δικά της οφέλη, αλλά έρχεται και με διάφορους συμβιβασμούς που ίσως δεν αξίζει τον κόπο για κάθε προγραμματιστή.

Ομοιότητες Vue και React:

- Χρησιμοποιούν ένα εικονικό DOM
- παρέχουν αντιδραστικά και συνθετικά components
- να διατηρούν την κεντρική τους δομή, όπως η δρομολόγηση (routing) και η κεντρική κατάσταση (global state)

3.4 Σύγκριση Vue και AngularJS

Η σύνταξη της Vue μοιάζει πολύ με την AngularJS (π.χ. v-if vs ng-if). Αυτό συμβαίνει επειδή υπήρχαν πολλά πράγματα που η AngularJS πήρε σωστά και αυτά ήταν μια έμπνευση για την ανάπτυξη της Vue. Υπάρχουν επίσης πολλά αγκάθια που έρχονται με το AngularJS ωστόσο, όπου η Vue προσπάθησε να προσφέρει σημαντική βελτίωση.

Η Vue έχει σαφέστερο χωρισμό μεταξύ components και directives. Τα directives προορίζονται να ενσωματώνουν μόνο DOM χειρισμούς, ενώ τα components είναι αυτοτελείς μονάδες που έχουν τη δική τους οπτική γωνία και λογική δεδομένων. Στη AngularJS, τα directives κάνουν τα πάντα και τα components είναι απλώς ένα συγκεκριμένο είδος directives.

Η Vue έχει καλύτερη απόδοση και πιο εύκολο να βελτιστοποιηθεί. Η AngularJS γίνεται αργη όταν υπάρχουν πολλοί παρατηρητές, γιατί κάθε φορά που αλλάζει οτιδήποτε στο πεδίο εφαρμογής, όλοι αυτοί οι παρατηρητές πρέπει να επανεκτιμηθούν ξανά. Επίσης, ο κύκλος αφομοίωση μπορεί να πρέπει να τρέξει πολλές φορές για να "σταθεροποιηθεί" εάν κάποιος παρατηρητής ενεργοποιεί μια άλλη ενημερωμένη έκδοση. Οι χρήστες της AngularJS πρέπει συχνά να καταφεύγουν σε εσωτερικές τεχνικές για να ξεπεράσουν τον κύκλο αφομοίωσης και σε ορισμένες περιπτώσεις δεν υπάρχει τρόπος να βελτιστοποιηθεί ένα πεδίο με πολλούς παρατηρητές.

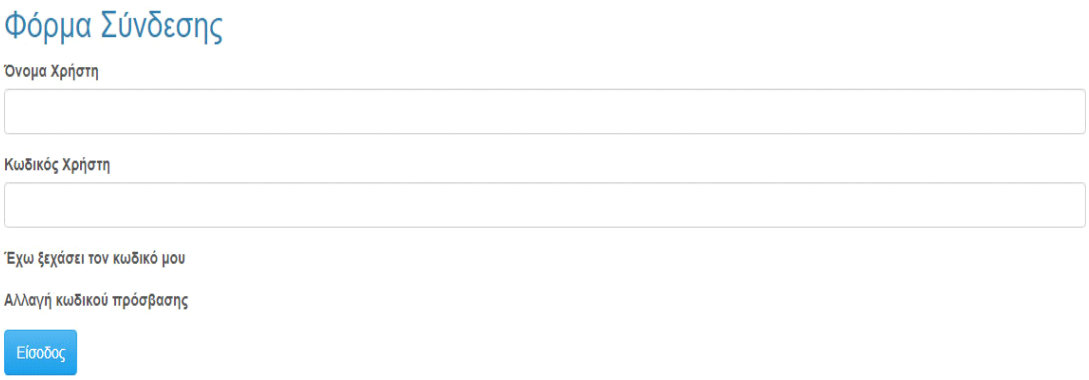
Η Vue δεν υποφέρει καθόλου από αυτό επειδή χρησιμοποιεί ένα διαφανές σύστημα παρατήρησης εντοπισμού εξάρτησης με ουρά αναμονής - όλες οι αλλαγές ενεργοποιούνται ανεξάρτητα, εκτός εάν έχουν ξεκάθαρες σχέσεις εξάρτησης.

Η Angular απαιτεί τη χρήση του TypeScript, δεδομένου ότι σχεδόν όλη η τεκμηρίωση και οι πόροι μάθησης βασίζονται στο TypeScript. Το TypeScript έχει τα πλεονεκτήματά του - ο έλεγχος στατικού τύπου μπορεί να είναι πολύ χρήσιμος για εφαρμογές μεγάλης κλίμακας και μπορεί να είναι μια μεγάλη ώθηση παραγωγικότητας για προγραμματιστές με περιβάλλοντα Java και C #.

4 ΚΩΔΙΚΑΣ ΦΟΡΜΩΝ ΜΕ ΧΡΗΣΗ JAVASCRIPT FRAMEWORKS

Σε αυτή την ενότητα θα γίνει παρουσίαση του κώδικα που χρησιμοποιήθηκε για τη μελέτη των φορμών με τη χρήση frameworks. Στο τέλος της ενότητας θα γίνει μια προσωπική εκτίμηση πάντα με κριτήρια το χρόνο υλοποίησης, την εύκολη κατανόηση του κώδικα καθώς και την υποστήριξη επαναχρησιμοποίησης κώδικα.

4.1 Φόρμα Σύνδεσης



Εικόνα 4.1 Φόρμα Σύνδεσης

4.1.1 Με χρήση Angular Framework

```
<h2>Φόρμα Σύνδεσης</h2>
<form [formGroup]="loginForm" (ngSubmit)="onSubmit()">
    <div class="form-group">
        <label for="username">Όνομα Χρήστη</label>
        <input type="text" formControlName="username" class="form-control"
[ngClass]="{ 'is-invalid': submitted && f.username.errors }" />
        <div *ngIf="submitted && f.username.errors" class="invalid-feedback">
            <div *ngIf="f.username.errors.required">Username is required</div>
        </div>
    </div>
    <div class="form-group">
        <label for="password">Κωδικός Χρήστη</label>
        <input type="password" formControlName="password" class="form-control"
[ngClass]="{ 'is-invalid': submitted && f.password.errors }" />
        <div *ngIf="submitted && f.password.errors" class="invalid-feedback">
            <div *ngIf="f.password.errors.required">Password is required</div>
        </div>
    </div>
    <div class="form-group">
        <span>Έχω ξεχάσει τον κωδικό μου</span>
        <span>Αλλαγή κωδικού πρόσβασης</span>
    </div>
    <div class="form-group">
        <button [disabled]="loading" class="btn btn-primary">Σύνδεση</button>
    </div>
</form>
```

```
import { Component, OnInit } from '@angular/core';
import { Router, ActivatedRoute } from '@angular/router';
import { FormBuilder, FormGroup, Validators } from '@angular/forms';
import { first } from 'rxjs/operators';

import { AlertService, AuthenticationService } from '../_services';

@Component({templateUrl: 'login.component.html'})
export class LoginComponent implements OnInit {
    loginForm: FormGroup;
    loading = false;
    submitted = false;
    returnUrl: string;

    constructor(
        private formBuilder: FormBuilder,
        private route: ActivatedRoute,
        private router: Router,
        private authenticationService: AuthenticationService,
        private alertService: AlertService) {}

    ngOnInit() {
        this.loginForm = this.formBuilder.group({
            username: ['', Validators.required],
            password: ['', Validators.required]
        });

        // reset login status
        this.authenticationService.logout();

        // get return url from route parameters or default to '/'
        this.returnUrl = this.route.snapshot.queryParams['returnUrl'] || '/';
    }

    // convenience getter for easy access to form fields
    get f() { return this.loginForm.controls; }

    onSubmit() {
        this.submitted = true;

        // stop here if form is invalid
        if (this.loginForm.invalid) {
            return;
        }
```

```
        this.loading = true;
        this.authenticationService.login(this.f.username.value,
this.f.password.value)
            .pipe(first())
            .subscribe(
                data => {
                    this.router.navigate([this.returnUrl]);
                },
                error => {
                    this.alertService.error(error);
                    this.loading = false;
                });
    }
}
```

4.1.2 Με χρήση React Framework

```
class FormApp extends React.Component {
  render() {
    return (
      <div className="container">
        <Header />
        <ContentBody />
      </div>
    );
  }
}
class Header extends React.Component {
  constructor(props) {
    super(props);
  }
  render() {
    return <h2>Φόρμα Σύνδεσης</h2>;
  }
}
class ContentBody extends React.Component {
  render() {
    return (
      <div className="row">
        <div className="col-md-12">
          <FormBody />
        </div>
      </div>
```

```
    );
  }
}

class FormBody extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      username: "",
      password: ""
    };
    this.saveFormData = this.saveFormData.bind(this);
  }
  saveFormData(e) {
    var myArr = this.state.data;
    myArr.push({
      username: ReactDOM.findDOMNode(this.refs.username).value,
      password: ReactDOM.findDOMNode(this.refs.password).value
    });
    this.setState({ data: myArr });
    ReactDOM.findDOMNode(this.refs.username).value = "";
    ReactDOM.findDOMNode(this.refs.userpassword).value = "";
  }
  render() {
    return (
      <form name="userform" id="userform">
        <div className="form-group">
          <label for="username">Όνομα Χρήστη</label>
          <input
            className="form-control"
            name="username"
            id="username"
            ref="username"
          />
        </div>
        <div className="form-group">
          <label for="useremail">Κωδικός Χρήστη</label>
          <input
            className="form-control"
            name="userpassword"
            ref="useremail"
            id="password"
          />
        </div>
        <div className="form-group">
```

```
          <div style={{ fontWeight: "700" }}>Έχω ξεχάσει τον κωδικό μου</div>
        </div>
        <div className="form-group">
          <div style={{ fontWeight: "700" }}>Αλλαγή κωδικού πρόσβασης</div>
        </div>
        <div className="form-group">
          <button
            className="btn btn-primary"
            onClick={this.saveFormData}
            type="button"
          >
            Είσοδος
          </button>
        </div>
      </form>
    );
  }
}
class DataUL extends React.Component {
  render() {
    return (
      <li>
        {this.props.singlerecord.username}, {this.props.singlerecord.email},{" "}
        {this.props.singlerecord.fullname}
      </li>
    );
  }
}
ReactDOM.render(<FormApp />, document.getElementById("react-form"));
```

4.1.3 Με χρήση Vue Framework

```
section class="section">
  <div class="container">
    <article class="card">
      <div class="card-content">
        <h1 class="title">Φόρμα Σύνδεσης</h1>
```

```
<form id="login-form" @submit.prevent="submitLogin">
  <div
    v-if="hasErrors"
    id="login-errors"
    role="alert"
    aria-live="assertive"
  >
    <div class="notification is-danger">
      Σφάλμα σύνδεσης
    </div>
  </div>
  <div class="field">
    <label for="email" class="label">Όνομα Χρήστη:</label>
    <input
      type="username"
      class="input"
      id="username"
      aria-describedby="login-errors"
      v-model="fields.username"
      required
    >
  </div>
  <div class="field">
    <div class="label space-between">
      <label for="password" id="password-label">Κωδικός Χρήστη:</label>
    </div>
    <input
      class="input"
      :class="{ 'is-danger': hasErrors }"
      id="password"
      aria-describedby="login-errors"
      v-model="fields.password"
      required
    >
  </div>
  <div class="field">
    <div class="control">
        <span class="checkbox-text">Έχω ξεχάσει τον κωδικό μου</span>
    </div>
    <div class="control">
         <span class="checkbox-text">Αλλαγή κωδικού πρόσβασης</span>
    </div>
  </div>
  <div class="field">
    <div class="control space-between">
```

```
              <div
                id="feedback"
                role="status"
                aria-live="polite"
              >
                <span class="msg-loading" v-if="isLoading">Loading...</span>
                <span class="msg-success" v-if="isSuccess">Επιτυχής
Σύνδεση</span>
              </div>
              <button
                type="submit"
                class="button is-medium is-link"
                :disabled="isLoading || isSuccess"
                aria-describedby="feedback"
              >Σύνδεση</button>
            </div>
          </div>
        </form>
      </div>
    </article>
  </div>
</section>
const defaultFields = {
  email: '',
  password: '',
};
new Vue({
  el: '#login-form',
  data: {
    hasErrors: false,
    passwordIsText: false,
    isLoading: false,
    isSuccess: false,
    fields: {
      ...defaultFields
    }
  },
  methods: {
    submitLogin() {
      this.isLoading = true;
      axios.post(
          'https://localhost/login?delay=1',
          {
            ...this.fields
          }
```

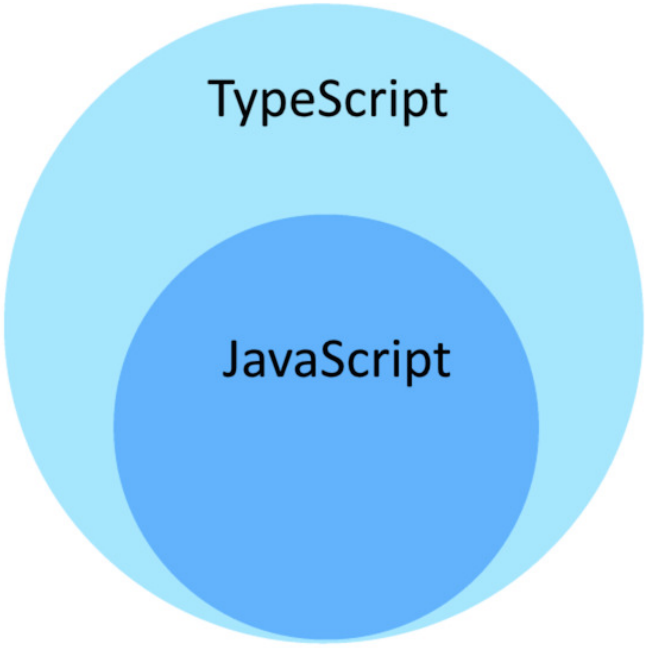
```
        )
        .then(() => {
          this.isLoading = false;
          this.isSuccess = true;
          this.fields = {
            ...defaultFields
          }
        })
        .catch(() => {
          this.isLoading = false;
          this.hasErrors = true;
        });
    }
  }
});
```

5 ΣΥΜΠΕΡΑΣΜΑΤΑ

5.1 Γενικά

Σε σύγκριση με τη Vue και τη React, η Angular έχει έναν τόνο αρχείων. Ένας μεγάλος λόγος γι 'αυτό είναι ότι ενώ η Vue βάζει τα πάντα για ένα component σε ένα αρχείο, και η React πετά το CSS στο δικό του αρχείο, η Angular βάζει το css σε ένα αρχείο, το html σε άλλο, και το component σε άλλο. Ενώ μπορούμε να βάλουμε όλα αυτά σε ένα φάκελο, θεωρείται η καλύτερη πρακτική να τα διατηρήσουμε χωριστά.

Ένας άλλος λόγος για τη δομή των αρχείων είναι ότι η Angular χρησιμοποιεί TypeScript. Το TypeScript είναι "ένα υπερσύνολο της JavaScript που παρέχει προαιρετική στατική πληκτρολόγηση, κλάσεις και διεπαφές"



Εικόνα 5.1 TypeScript

Η TypeScript πραγματικά προσθέτει ένα σωρό πράγματα πάνω από τη JavaScript. Αν γράψουμε μια γραμμή JavaScript στον κώδικα TypeScript, θα λειτουργήσει καλά. Αυτό που επιτρέπει η TypeScript είναι να δημιουργήσει εύκολα κλάσεις και να τις θέσουν σε ενέργεια.

Οι έννοιες γύρω από τα components και ο τρόπος δημιουργίας μιας εφαρμογής με αυτά είναι οι ίδιοι, με συγκεκριμένα-frameworks ονόματα. Αυτό που εξακολουθεί να χωρίζει τη React, τηο Vue και την Angular είναι οι υποκείμενες φιλοσοφίες και οι βάσεις στις οποίες είναι χτισμένα.

Η React προσανατολίζεται σε μεγάλο βαθμό στους προγραμματιστές να επιλέξουν τα προτιμώμενα εργαλεία τους από το οικοσύστημα (ή να τα κατασκευάσουν αν είναι απαραίτητο).

Η Vue είναι στη θέση να ξεκινήσει ως μια απλή ετικέτα script, αλλά παρέχει επίσης ένα συνεκτικό σύνολο εργαλείων για την κατασκευή μεγαλύτερων εφαρμογών (στοιχεία ενός αρχείου, Vuex, vue-router, τεκμηρίωση και ακόμη και οδηγός style).

Τα τρία frameworks βλέπουμε επίσης πως έχουν πολλές ομοιότητες στο επίπεδο γραφής. Όπως είπαμε και παραπάνω η Vue έχει πάρει τα καλά σημεία της Angular και της React, έτσι με μια γρήγορη ματιά βλέπουμε πως η Angular χρησιμοποιεί το tag **ng-** ενώ η Vue **v-** .

Τα JavaScript frameworks διαθέτουν ισχυρά εργαλεία για την γρήγορη δημιουργία εφαρμογών διαδικτύου με πλήρη λειτουργικότητα. Ταυτόχρονα, κανένα framework δεν ικανοποιεί κάθε συγκεκριμένη περίπτωση ή εφαρμογή ομάδας. Βέβαια υπάρχει και ένα μέρος προγραμματιστών

που αρνούνται να χρησιμοποιήσουν τα frameworks - θεωρούν ότι είναι πολύ ‘βαριά’ και πολύ αυστηρά στην υλοποίηση τους και αναγκάζουν τους υπεύθυνους ανάπτυξης να μάθουν κάθε καινούργια τάση του framework που συνήθως ανανεώνεται ανά τακτά χρονικά διαστήματα. Επίσης μερικοί προγραμματιστές δεν τα θεωρούν ως framework αλλά αντίθετα τα θεωρούν ότι είναι απλά βιβλιοθήκες κώδικα ή ένα γκρουπ χρήσιμων βοηθημάτων.

5.2 Το μέλλον των JavaScript frameworks

Είτε θεωρούνται frameworks ή όχι, τα AngularJS, ReactJS και Vue βρίσκονται στη μέση των θερμών αλλαγών που συμβαίνουν στον σημερινό ταχέως μεταβαλλόμενο και υψηλής ταχύτητας κόσμο της πληροφορικής. Η ποικιλία των συσκευών αναπτύσσεται με τρομερούς ρυθμούς τα τελευταία χρόνια, και αυτό θα ήταν πρακτικά αδύνατο χωρίς αυτά τα frameworks να μειώσουν το μέγεθος μιας εφαρμογής.

Μόνο μερικά χρόνια πριν, το μόνο πράγμα που μας ένοιαζε ήταν μια ιστοσελίδα να είναι μέσα στα πλαίσια του προγράμματος περιήγησης που εμφανίζεται σε μια μέση οθόνη, κάνοντας προσαρμογές για τις αρχαϊκές εκδόσεις του Internet Explorer. Τώρα, η πολυπλοκότητα είναι 100 φορές μεγαλύτερη πράγμα που τα JavaScript Frameworks βοηθούν τόσο στην υλοποίηση όσο και στα αμέτρητα χαρακτηριστικά που προσφέρουν.

6 ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] https://el.wikipedia.org/wiki/JavaScript

[2] https://en.wikipedia.org/wiki/Web_framework

[3] https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller

[4] https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93viewmodel

[5] https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93presenter

[6] https://en.wikipedia.org/wiki/Data_binding

[7]https://el.wikipedia.org/wiki/%CE%94%CE%B9%CE%B1%CE%B4%CE%B9%CE%BA%CF%84%CF%85%CE%B1%CE%BA%CE%AE_%CE%B5%CF%86%CE%B1%CF%81%CE%BC%CE%BF%CE%B3%CE%AE

[8] https://en.wikipedia.org/wiki/Node.js

[9] https://en.wikipedia.org/wiki/AngularJS

[10] https://en.wikipedia.org/wiki/React_(JavaScript_library)

[11] https://en.wikipedia.org/wiki/Vue.js

[12] https://trends.google.com/trends/explore?date=today%205-y&q=%2Fm%2F0j45p7w,React,Vue

[13] https://en.wikipedia.org/wiki/TypeScript

[14] https://en.wikipedia.org/wiki/Front-end_web_development