

ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ
ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΤΜΗΜΑΤΟΣ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ



ΣΥΓΚΡΙΤΙΚΗ ΑΝΑΛΥΣΗ ENTERPRISE ΕΦΑΡΜΟΓΗΣ ΜΕΤΑΞΥ ΤΟΥ
JAVASCRIPT FRAMEWORK VUE JS ΚΑΙ ΤΗΣ JAVASCRIPT ΒΙΒΛΙΟΘΗΚΗΣ
BACKBONE.JS

Διπλωματική Εργασία

του

Αχιλλέα Μαύρου

Θεσσαλονίκη, Αύγουστος 2020

ΣΥΓΚΡΙΤΙΚΗ ΑΝΑΛΥΣΗ ENTERPRISE ΕΦΑΡΜΟΓΗΣ ΜΕΤΑΞΥ ΤΟΥ
JAVASCRIPT FRAMEWORK VUE JS ΚΑΙ ΤΗΣ JAVASCRIPT ΒΙΒΛΙΟΘΗΚΗΣ
BACKBONE.JS

Αχιλλέας Μαύρος

Τα προηγούμενα πτυχία σας – π.χ. Πτυχίο Μηχανικών Πληροφορικής, ΤΕΙ Καβάλας,
2016

Διπλωματική Εργασία

υποβαλλόμενη για τη μερική εκπλήρωση των απαιτήσεων του

ΜΕΤΑΠΤΥΧΙΑΚΟΥ ΤΙΤΛΟΥ ΣΠΟΥΔΩΝ ΣΤΗΝ ΕΦΑΡΜΟΣΜΕΝΗ
ΠΛΗΡΟΦΟΡΙΚΗ

Επιβλέπων/ουσα Καθηγητής/τρια
Θεόδωρος Κασκάλης

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την ηη/μμ/εεεε

Όνοματεπώνυμο 1

Όνοματεπώνυμο 2

Όνοματεπώνυμο 3

.....

.....

.....

Αχιλλέας Μαύρος

Περίληψη

Στην παρούσα εργασία προχωράμε στην επιλογή δύο ανερχόμενων εργαλείων javascript του framework vue.js και και της βιβλιοθήκης backbone.js για να κάνουμε μια έρευνα μαθαίνοντας όσες περισσότερες πληροφορίες μπορούμε και να βρούμε τα πλεονεκτήματα αλλά και τα μειονεκτήματα έτσι ώστε να έχουμε μία εικόνα για τις δυνατότητες τις οποίες παρέχουν και να κατανοήσουμε την λειτουργία τους αλλά επιλέγουμε και μια εφαρμογή todolist η οποία είναι κατασκευασμένη και στα δύο framework για να μπορέσουμε να κάνουμε μια συγκριτική ανάλυση μεταξύ τους έτσι ώστε να βρούμε ποιό θα ήταν το καλύτερο javascript εργαλείο έτσι ώστε να το επιλέξουμε σε μια μελλοντική δημιουργία μιας εφαρμογής.

Λέξεις Κλειδιά: Javascript, Framework, Βιβλιοθήκη, Enterprise application, Vue.js, Backbone.js, Συγκριτική Ανάλυση

Abstract

In the present work we proceed to the selection of two ascending javascript tools of the framework vue.js and the backbone.js library to do a research by learning as much information as we can and to find the pros and cons so that we have an idea of the possibilities which we provide and understand their function but we also choose a todolist application which is built in both frameworks so that we can make a comparative analysis between them so that we can find which would be the best javascript tool so that we can choose it in a future creation of an application.

Keywords: Javascript, Framework, Enterprise application, library, Vue.js, Backbone, js, Comparative Analysis

Πρόλογος – Ευχαριστίες

Ευχαριστώ όσους ήταν δίπλα μου και με βοήθησαν σε ολή την χρονική διάρκεια της εργασίας.

Περιεχόμενα

1 Εισαγωγή.....	σελ.4
1.1 Σκοπός – Στόχοι.....	σελ.5
1.2 Συνεισφορά.....	σελ.5
1.3 Διάρθρωση της μελέτης.....	σελ.5
2 Βιβλιογραφική Επισκόπηση – Θεωρητικό Υπόβαθρο.....	σελ.6
2.1 Τι είναι ένα enterprise application.....	σελ.6
2.2 Τι είναι η javascript.....	σελ.9
2.3 Τι είναι το framework.....	σελ.12
2.4 Vue.js.....	σελ.14
2.5 Backbone.js.....	σελ.19
2.5.1 jquery.....	σελ.26
2.5.2 underscore.js.....	σελ.27
2.6 Μοντέλο αρχιτεκτονικής λογισμικού MVC (Model-View-Controller).....	σελ.27
2.7 Μοντέλο αρχιτεκτονικής λογισμικού MVP (Model-View-Presenter).....	σελ.28
2.8 Μοντέλο αρχιτεκτονικής λογισμικού MVVM (Model-View-ViewModel).....	σελ.30
3 Μεθοδολογία.....	σελ.32
3.1 Μελέτη των frameworks.....	σελ.32
3.2 Αξιολόγηση-Σύγκριση framework.....	σελ.32
4 Παρουσίαση-Έλεγχος μετρήσεων.....	σελ.33
4.1 Παρουσίαση To do list.....	σελ.33
4.2 Ποιότητα εφαρμογής.....	σελ.35
4.3 Ο χρόνος φόρτωσης της σελίδας.....	σελ.36
4.4 Κατανάλωση Μνήμης.....	σελ.37
4.5 Μέτρηση CPU time.....	σελ.39
5. Συμπεράσματα.....	σελ.41
5.1 Επεκτάσεις για το μέλλον.....	σελ.41
Βιβλιογραφία.....	σελ.42

Κατάλογος Εικόνων

Εικόνα 1	Δείγμα κώδικα της javascript.....	7
Εικόνα 2	Δημοτικότητα της javascript για το 2019	8
Εικόνα 3	Κώδικας Vue.js	11
Εικόνα 4	Δείγμα κώδικα backbone.js.....	12
Εικόνα 5	Μοντέλο αρχιτεκτονικής λογισμικού MVC	14
Εικόνα 6	Μοντέλο αρχιτεκτονικής λογισμικού MVP.....	15
Εικόνα 7	Μοντέλο αρχιτεκτονικής λογισμικού MVVM.....	17
Εικόνα 8	Η todolist κατασκευασμένη από τον Evan You	19
Εικόνα 9	Η index.html της todolist με Vue.js	20
Εικόνα 10	Η todolist κατασκευασμένη από τον Addy Osmani	20
Εικόνα 11	Η index.html της todolist με backbone.js	21
Εικόνα 12	Αποτελέσματα Lighthouse για την απόδοση της todolist με Vue.js.....	21
Εικόνα 13	Αποτελέσματα Lighthouse για την απόδοση της todolist με Backbone.js	21
Εικόνα 14	Παρουσίαση απόδοση της φόρτωσης της todolist με vue.js.....	22
Εικόνα 15	Παρουσίαση απόδοση της φόρτωσης της todolist με backbone.js.....	23
Εικόνα 16	Ο χώρος κατανάλωσης της εφαρμογής todolist σε vue.js	24
Εικόνα 17	Δομή της todolist με vue.js	24
Εικόνα 18	Δομή της todolist με backbone.js.....	25
Εικόνα 19	CPU time vue.js	26
Εικόνα 20	CPU time backbone.js.....	26

1 Εισαγωγή

Στην σημερινή εποχή έχει ανθίσει πολύ το web development και σε αυτό παίζει μεγάλο ρόλο η ανάγκη της κάθε επιχείρησης να υπάρχει στον διαδικτυακό ιστότοπο εκτός από την φυσική της παρουσία να προσελκύσει περισσότερο κόσμο μέσω του internet. Για αυτό τον λόγο η δημιουργία μιας ιστοσελίδας, ενός ηλεκτρονικού καταστήματος ή μιας εφαρμογής ολοένα και θεωρείται για τον καθένα αναγκαία. Για αυτό τον λόγο οι web developers χρησιμοποιώντας την κατάλληλη γλώσσα προγραμματισμού προσπαθούν να κάνουν το καλύτερο που μπορούν για τις ανάγκες του πελάτη. Μία γλώσσα προγραμματισμού για τις συγκεκριμένες υλοποιήσεις είναι η javascript η οποία παρόλο που στην αρχή δεν έδιναν πολύ προσοχή στην ύπαρξή και τη χρησιμοποίησή της με το πέρασμα των χρόνων έχει πια μια περίοπτη θέση στην εκτίμηση των web developer. Για την υλοποίηση όμως ενός έργου θα πρέπει να χρησιμοποιούνται και κατάλληλα εργαλεία για μια πιο ποιοτικότερη, γρήγορη και ομαλή υλοποίηση. Για την υλοποίηση αυτή πρόκειται να χρησιμοποιήσουμε ένα javascript frameworks το vue js και μία βιβλιοθήκη javascript το backbone.js. Η ιδιαιτερότητα των συγκεκριμένων επιλογών είναι ότι πρόκειται για δύο τεχνολογίες οι οποίες δεν είναι τόσο δημοφιλείς και δεν έχουν την ίδια απήχηση που έχουν άλλα εργαλεία του συγκεκριμένου είδους όπως η Angular , η React , η jQuery κ.λ.π

Ένα javascript framework είναι ένα εργαλείο το οποίο παρέχει έτοιμες συναρτήσεις και λύσεις για την δημιουργία μιας εφαρμογής .Είναι ιδιαίτερα εύχρηστο και πολύ εύκολο στην δημιουργία του κώδικα και στην επεκτασιμότητα από άλλους προγραμματιστές.

Τα framework με τις βιβλιοθήκες έχουν αρκετές διαφορές μεταξύ τους , η κυριότερη είναι ότι οι προγραμματιστές έχουν τον μεγαλύτερο έλεγχο στις βιβλιοθήκες και μπορούν να τις χρησιμοποιήσουν όπως κρίνουν απαραίτητο μπορεί π.χ. να καλέσει τις κατάλληλες συναρτήσεις που χρειάζεται ο

προγραμματιστής για να φέρει το αποτέλεσμα που θέλει. Ωστόσο αν καλέσουμε μια μέθοδο ενός framework μπορεί να φέρει αποτελέσματα τα οποία δεν έχουμε δηλώσει στην πρότυπο της εφαρμογής μας .

1.1 Σκοπός – Στόχοι

Όπως προείπαμε για την υλοποίηση ενός έργου θα πρέπει να χρησιμοποιούνται τα κατάλληλα εργαλεία, για αυτό σκοπός μας είναι να κάνουμε μια μελέτη και σύγκριση ανάμεσα σε 2 javascript frameworks το vue.js και το backbone.js για να εντοπίσουμε τις διαφορές τους αλλά και για να δούμε ποιο είναι καλύτερο για μελλοντική χρήση σε ένα έργο.

1.2 Συνεισφορά

Για την συγκεκριμένη έρευνα θα διεξάγουμε μια βιβλιογραφική μελετη για τα δύο framework και την γλώσσα javascript, θα γίνει σύγκριση ανάμεσα στο vue.js κι στο backbone.js και θα επιλέξουμε μια εφαρμογή που είναι κατασκευασμένη και στα δύο frameworks έτσι ώστε να γίνουν οι κατάλληλες μετρήσεις και συγκρίσεις και να βγάλουμε συμπεράσματα για το ποιο framework είναι πιο αποδοτικό.

1.3 Διάρθρωση της μελέτης

Στο δεύτερο κεφάλαιο παρουσιάζουμε ορισμένες αναφορές για το τι είναι η γλώσσα javascript αλλά κάνουμε και μία πρώτη γνωριμία με την framework vue.js και backbone.js μαθαίνοντας το τι είναι, ορισμένες πληροφορίες για το καθένα αλλά και τα πλεονεκτήματα και τα μειονεκτήματα και επίσης θα δούμε τα μοντέλα αρχιτεκτονικής λογισμικού που χρησιμοποιούν. Στο τρίτο κεφάλαιο αναφέρουμε την μεθοδολογία για την ανάλυση των εφαρμογών που θα χρησιμοποιήσουμε και τα εργαλεία που θα χρησιμοποιηθούν και στο τέταρτο κεφάλαιο παρουσιάζουμε τα αποτελέσματα της σύγκρισης.

2 Θεωρητικό Υπόβαθρο

2.1 Τι είναι ένα enterprise application

Ένα enterprise application ή αλλιώς μια επιχειρησιακή εφαρμογή είναι μια εφαρμογή η οποία δημιουργείται για ένα επιχειρησιακό περιβάλλον το οποίο διευκολύνει μια επιχείρηση στις εσωτερικές λειτουργίες και εργασίες της. Οι συγκεκριμένες εφαρμογές είναι σχεδιασμένες για να ενσωματώνονται ή για να γίνεται διασύνδεση με άλλες εφαρμογές που χρησιμοποιούνται για ανάπτυξη σε διάφορα δίκτυα όπως internet, εταιρικά δίκτυα κ.α.

Οι επιχειρησιακές εφαρμογές είναι συχνά ένας κεντρικός κόμβος ή αλλιώς ένα σημείο συνένωσης για μια μεγάλη ποικιλία δεδομένων που η προέλευση τους είναι από διάφορες πηγές. Για αυτό είναι πολύ χρήσιμες για την βελτίωση δεδομένων και εφόσον γίνει εισαγωγή δεδομένων τα δεδομένα επικυρώνονται ως υψηλής ποιότητας.

Επίσης χρησιμοποιείται για την δημιουργία εγγράφων, βάσεων δεδομένων, υπολογιστικών φύλλων, παρουσιάσεων γραφικών κ.α.

Ορίζονται ως “σχεδιασμένες να ενσωματώνουν συστήματα υπολογιστών που εκτελούν όλες τις φάσεις των σταδίων μιας επιχείρησης για να διευκολύνουν τη συνεργασία και το συντονισμό της εργασίας σε ολόκληρη την επιχείρηση”. Για παράδειγμα, μια εταιρική εφαρμογή μπορεί να συγκεντρώσει διαφορετικές πτυχές μιας επιχείρησης, όπως λογιστική, ανθρώπινο δυναμικό, οικονομικά και απογραφή. Οι επιχειρηματικές εφαρμογές προσφέρονται ως χρήσιμο εργαλείο για να βοηθήσουν τις επιχειρήσεις με τα πολλά εμπόδια που αντιμετωπίζουν σήμερα.

Η επιχειρησιακή εφαρμογή είναι ένας αρκετά ευρύς όρος. Συχνά προσαρμόζονται στις ανάγκες μιας επιχείρησης και οι έμπειροι οργανισμοί πρέπει να επιλέγουν τις εφαρμογές τους με σύνεση. Υπάρχουν ορισμένα εργαλεία που μπορεί να περιλαμβάνονται σε μια συγκεκριμένη εταιρική σουίτα λογισμικού, από οπουδήποτε από συστήματα διαχείρισης σχέσης πελατών (CRM), έως εργαλεία χρέωσης, ακόμη και λύσεις για τη μόδα, την επιβολή του νόμου ή την υπηρεσία τροφίμων.

Παραδείγματα επιχειρηματικών εφαρμογών

Η εταιρική εφαρμογή είναι ένας αρκετά ευρύς όρος. Συχνά προσαρμόζονται στις ανάγκες μιας επιχείρησης και οι έμπειροι οργανισμοί πρέπει να επιλέγουν τις εφαρμογές τους με σύνεση. Υπάρχουν ορισμένα εργαλεία που μπορεί να περιλαμβάνονται σε μια συγκεκριμένη εταιρική σουίτα λογισμικού, από οπουδήποτε από συστήματα διαχείρισης σχέσης πελατών (CRM), έως εργαλεία χρέωσης, ακόμη και λύσεις για τη μόδα, την επιβολή του νόμου ή την υπηρεσία τροφίμων.

Ακολουθούν περισσότερες λεπτομέρειες σχετικά με μερικές από τις πιο δημοφιλείς επιχειρηματικές εφαρμογές:

Συστήματα διαχείρισης σχέσης πελατών (CRM): Οι σημερινοί επαγγελματίες πωλήσεων μπορεί να είναι εξαιρετικά απασχολημένοι και μερικές φορές, ένας δυνητικός πελάτης μπορεί να περάσει από τα δάχτυλά του χωρίς προειδοποίηση.

Εάν οι οργανισμοί δεν προσλάβουν αντιπροσώπους πωλήσεων με δεξιότητες οργάνωσης και μνήμης υπεράνθρωπου επιπέδου, θα υπάρξουν φορές που οι εκπρόσωποι πωλήσεων πιθανώς να χάσουν ευκαιρίες να κλείσουν μια συμφωνία, να πουλήσουν, να πωλήσουν ή να προωθήσουν νέα προϊόντα σε υπάρχοντες πελάτες.

Ωστόσο, ένα σύστημα CRM μπορεί να δώσει στις ομάδες πωλήσεων μια πολύτιμη προοπτική για το πού βρίσκονται με κάθε πελάτη, ενώ ταυτόχρονα συγκεντρώνει τα στοιχεία επικοινωνίας των πελατών και τις λεπτομέρειες σχετικά με την προηγούμενη επικοινωνία.

Πολλά συστήματα CRM περιλαμβάνουν λειτουργίες όπως αυτοματοποίηση ροής εργασίας, πρόσβαση χρηστών βάσει ρόλων, ενοποίηση και παρακολούθηση email και αναφορά.

Συστήματα εταιρικού σχεδιασμού πόρων (ERP): Πολλοί άνθρωποι περιγράφουν ένα σύστημα ERP ως τον «εγκέφαλο» του οργανισμού. Ένα σύστημα ERP λαμβάνει όλα τα

διαφορετικά μέρη μιας επιχείρησης, τα ψηφιοποιεί και δημιουργεί δίκτυα μεταξύ τους. Συγκεκριμένα, ένα σύστημα ERP μπορεί να ενσωματώσει τις οικονομικές πληροφορίες ενός οργανισμού. Χωρίς ένα ολοκληρωμένο σύστημα, μεμονωμένα τμήματα, όπως χρηματοδότηση, πωλήσεις και ούτω καθεξής, πρέπει να βασίζονται σε ξεχωριστά συστήματα, καθένα από τα οποία θα έχει πιθανώς διαφορετικούς αριθμούς εσόδων και εξόδων. Η χρήση ενός ERP διασφαλίζει ότι όλα τα μέλη του οργανισμού εργάζονται από τους ίδιους βασικούς δίσκους και τις ενημερωμένες πληροφορίες.

Ένα από τα σημαντικότερα προνόμια ενός συστήματος ERP είναι ότι μπορεί να ενσωματώσει άλλες επιχειρηματικές εφαρμογές, όπως CRM, διαχείριση αλυσίδας εφοδιασμού (SCM) ή διαχείριση αποθεμάτων (IM). Η εφαρμογή ενός συστήματος ERP μπορεί να καθαρίσει δραστικά τον τρόπο λειτουργίας των μεγαλύτερων οργανισμών κάνοντας δραστηριότητες και ροές εργασίας πιο συνεκτικές και διαφανείς. Προσφέρουν επίσης σημαντικές πληροφορίες που μπορούν να βοηθήσουν στον εντοπισμό σημείων συμφόρησης και αναποτελεσματικότητας και να βελτιώσουν τον προγραμματισμό για το μέλλον.

Οι εφαρμογές διαχείρισης έργων δίνουν μια επισκόπηση όλων των ενεργών έργων. Επιτρέπουν στα μέλη της ομάδας να εμβαθύνουν στις λεπτομέρειες και να ελέγχουν κάθε ορόσημο, νέα λεπτομέρεια και επερχόμενη εργασία που μπορεί να χρειαστεί να ολοκληρωθούν για να διατηρηθεί η πρόοδος στο χρονοδιάγραμμα.

Όταν χρησιμοποιούνται σωστά, οι εφαρμογές διαχείρισης έργων είναι ένας πολύ καλός τρόπος για να παραμείνουν οι ομάδες όσο το δυνατόν πιο παραγωγικές, διασφαλίζοντας παράλληλα ότι οι διαδικασίες και οι διαδικασίες παραμένουν αποτελεσματικές και διαφανείς σε κάθε εργασία και πρωτοβουλία.

Συστήματα διαχείρισης ανθρώπινου κεφαλαίου (HCM): Ένα σύστημα HCM βοηθά τους οργανισμούς να διαχειρίζονται και να διατηρούν το εργατικό δυναμικό τους. Οι λειτουργίες ενός συστήματος HCM οργανώνονται συχνά σε κατηγορίες, όπως διαχείριση ταλέντων, διαχείριση εργατικού δυναμικού και παροχή υπηρεσιών.

Τα εργαλεία HCM βοηθούν τις ομάδες ανθρώπινου δυναμικού να προσαρμοστούν στις συνεχώς μεταβαλλόμενες απαιτήσεις διαχείρισης και απόκτησης ταλέντων, ενώ ταυτόχρονα ενισχύουν την παραγωγικότητα. Πολλές λύσεις HCM χρησιμοποιούν τώρα τεχνητή νοημοσύνη (AI) για να προτείνουν ρόλους, να προτείνουν μαθήματα και πρόσθετες ευκαιρίες μάθησης, και ακόμη και προβλέπουν ποιοι εργαζόμενοι σκέφτονται να εγκαταλείψουν την εταιρεία. Η χρήση του AI είναι επίσης επωφελής για την πρόσληψη HR. Για παράδειγμα, η έξυπνη αντιστοίχιση που τροφοδοτείται με τεχνητή νοημοσύνη βοηθά τους υπαλλήλους να εντοπίσουν τους υποψηφίους που ταιριάζουν καλύτερα και βοηθά τους υπαλλήλους να βρουν τις βέλτιστες διαδρομές σταδιοδρομίας τους.

2.2 Τι είναι η javascript

Η javascript είναι μια γλώσσα προγραμματισμού την οποία την συναντάμε και την χρησιμοποιούμε πάρα πολύ συχνά στην δημιουργία ιστοσελίδων και ηλεκτρονικών καταστημάτων(e-shop). Είναι υπεύθυνη για την διαδραστικότητα που παρέχετε στις σελίδες που εμπεριέχουν την html. Η javascript τρέχει στον browser και όχι στον server τις περισσότερες φορές. Βασίζεται στα πρωτότυπα, είναι δυναμική αντικειμενοστραφής καθώς και συναρτησιακή. Η σύνταξη της javascript έχει επηρεαστεί κατά κύριο λόγο από την σύνταξη της C. Πολλοί στην αρχή πιστεύουν ότι η javascript έχει πολλά κοινά με την γλώσσα προγραμματισμού Java στην πορεία όμως όσοι εμβαθύνουν στο προγραμματίσιμο βλέπουν πως στην πραγματικότητα είναι πολύ διαφορετικές μεταξύ τους. Είναι εύκολη στην σύνταξη αλλά και στο τρόπο εκτέλεσης της.

Αρχικά χρησιμοποιήθηκε για προγραμματισμό από την πλευρά του πελάτη (client), που ήταν ο φυλλομετρητής (browser) του χρήστη, και χαρακτηρίστηκε σαν *client-side* γλώσσα προγραμματισμού. Αυτό σημαίνει ότι η επεξεργασία του κώδικα Javascript και η παραγωγή του τελικού περιεχομένου HTML δεν πραγματοποιείται στο διακομιστή, αλλά στο πρόγραμμα περιήγησης των επισκεπτών, ενώ μπορεί να ενσωματωθεί σε στατικές σελίδες HTML. Αντίθετα, άλλες γλώσσες όπως PHP εκτελούνται στο διακομιστή (*server-side* γλώσσες προγραμματισμού).

Παρά την ευρεία χρήση της Javascript για συγγραφή προγραμμάτων σε περιβάλλον φυλλομετρητή, από την αρχή χρησιμοποιήθηκε και για τη συγγραφή κώδικα από την πλευρά του διακομιστή, από την ίδια τη Netscape στο προϊόν LiveWire, με μικρή επιτυχία. Η χρήση της Javascript στο διακομιστή εμφανίζεται πάλι σήμερα, με τη διάδοση του Node.js, ενός μοντέλου προγραμματισμού βασισμένο στα γεγονότα (events).

Όρισμένες ομοιότητες που υπάρχουν ανάμεσα στην Java και την Javascript είναι σε σχέση με την ονοματολογία και ότι ορισμένα αντικείμενα της javascript βασίζονται σε κλάσεις της java.

Είναι γνωστό ότι βρίσκετε στις πρώτες επιλογές των προγραμματιστών μερικοί από τους λόγους είναι οι εξής:

- Πολλοί browsers έχουν ενσωματωμένη javascript.
- Είναι πολύ γρήγορη για τον χρήστη.
- Είναι εύκολη στην εκμάθηση της.
- Δεν χρειάζεται ειδικό μεταγλωτιστή

Οι βασικές αρχές σχεδίασης βασίζονται στις γλώσσες προγραμματισμού Self και Scheme και υποστηρίζει το αντικειμενοστρεφές το προστατικό και το συναρτησιακό στυλ προγραμματισμού. Το πρότυπο της γλώσσας κατά τον οργανισμό της τυποποίησης ECMA ονομάζεται ECMAScript. Επίσης εκτός από ιστοσελίδες και ηλεκτρονικά καταστήματα χρησιμοποιείται και σε εφαρμογές εκτός ιστοσελίδων καθώς και σε εξειδικευμένους φυλλομετρητές.

Η πρώτη φορά που κυκλοφορήσε τον Νοέμβριο του 1995. Δημιουργήθηκε από τον Brendan Eich της εταιρείας Netscape με την ονομασία Mocha στη συνέχεια μετονομάστηκε σε Livescript και ύστερα σε Javascript επειδή η ανάπτυξη της επηρεάστηκε από την Java.

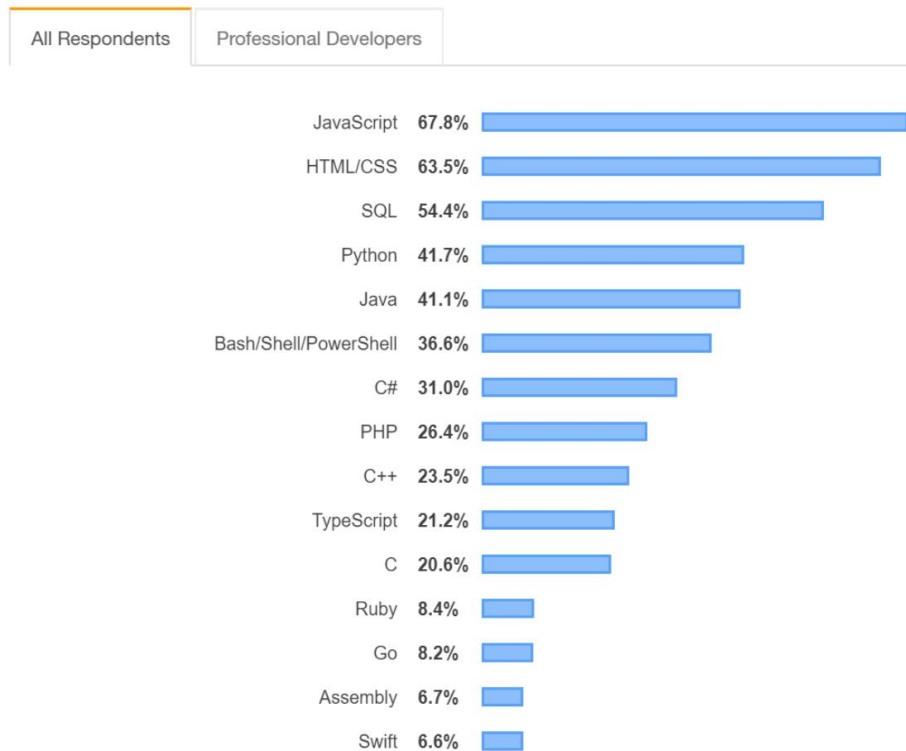
Η JavaScript απέκτησε μεγάλη επιτυχία ως γλώσσα στην πλευρά του πελάτη (client-side) για εκτέλεση κώδικα σε ιστοσελίδες, και περιλήφθηκε σε διάφορα προγράμματα περιήγησης στο Web. Κατά συνέπεια, η εταιρεία Microsoft ονόμασε την εφαρμογή της σε JScript για να αποφύγει δύσκολα θέματα εμπορικών σημάτων. 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. 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.

```
1  <?php
2  $i = 0;
3
4  function test ($arg1) {
5      exit($arg1);
6  }
7  ?>
8
9  <script type="text/javascript">
10 // test
11 function test () {
12     var i = 0;
13     var f = 0.05;
14 }
15
16 function test2 () {
17     //
18 }
19 </script>
```

Εικόνα 1. Δείγμα κώδικα της javascript.

Programming, Scripting, and Markup Languages



Εικόνα 2. Δημοτικότητα της javascript για το 2019.

2.3 Τι είναι το framework

Το framework είναι ένα λογισμικό ανοιχτού κώδικα το οποίο σχεδιάστηκε με σκοπό για την δημιουργία και την ανάπτυξη διάφορων εφαρμογών και δίνει την δυνατότητα να μην ξεκινήσουμε την υλοποίηση από το μηδέν αλλά να υπάρχει μια βάση η οποία θα διευκολύνει τον εκάστοτε προγραμματιστή στο θέμα των δυνατοτήτων αλλά και στην εξοικονόμηση του χρόνου. Το framework δίνει ένα τυπικό τρόπο δημιουργίας και ανάπτυξης εφαρμογών του ιστού στον παγκόσμιο ιστό και ακολουθεί το μοντέλο MVC(Model-View-Controller).

Διαφέρει από μια βιβλιοθήκη JavaScript. Μια βιβλιοθήκη προσφέρει συναρτήσεις που πρέπει να κληθούν από τον γονικό της κώδικα, ενώ ένα πλαίσιο ορίζει ολόκληρο το σχεδιασμό της εφαρμογής. Ένας προγραμματιστής δεν καλεί ένα framework. Αντ' αυτού, το framework καλεί και χρησιμοποιεί τον κώδικα με κάποιο συγκεκριμένο τρόπο. Ορισμένα πλαίσια JavaScript ακολουθούν το MVC (model-view controller) που

έχει σχεδιαστεί για να διαχωρίζει μια εφαρμογή Ιστού σε ορθογώνιες μονάδες για τη βελτίωση της ποιότητας του κώδικα και της συντηρησιμότητας.

Πλεονεκτήματα των framework

- Βοηθά στην καθιέρωση καλύτερων πρακτικών προγραμματισμού και στην εφαρμογή των σχεδιαστικών προτύπων.
- Ο κωδικός είναι πιο ασφαλής.
- Ο περιττός κωδικός μπορεί να αποφευχθεί.
- Βοηθά στην ανάπτυξη κώδικα με λιγότερα σφάλματα.
- Διευκολύνει την εργασία σε προηγμένες τεχνολογίες.
- Αρκετά τμήματα κώδικα και λειτουργίες είναι προ-κατασκευασμένα και προ-δοκιμασμένα. Αυτό καθιστά τις εφαρμογές πιο αξιόπιστες.
- Ο έλεγχος και ο εντοπισμός σφαλμάτων του κώδικα είναι πολύ ευκολότερος και μπορεί να γίνει ακόμη και από προγραμματιστές που είναι αδύναμοι στην σύνταξη του κώδικα.
- Ο χρόνος που απαιτείται για την ανάπτυξη μιας εφαρμογής μειώνεται σημαντικά.

2.4 VUE.JS

Το Vue.js είναι ένα javascript framework το οποίο δημιουργήθηκε από τον Evan You ο οποίος εργαζόταν στην Google και στην ανάθεση των έργων του χρησιμοποιούσε την Angular. Κάποια στιγμή σκέφτηκε πως θα μπορούσε να πάρει ένα κομμάτι που τον ενδιέφερε και ήταν χρήσιμο από την Angular για να έφτιαχνε κάτι το οποίο θα ήταν πολύ πιο ελαφρύ. Το Vue.js κυκλοφόρησε για πρώτη φορά τον φεβρουάριο του 2014 και η πιο πρόσφατη έκδοσή του είναι η 2.6 που κυκλοφόρησε στις 4 φεβρουαρίου του 2019. Η αρχιτεκτονική λογισμικού που χρησιμοποιεί το Vue.js είναι το MVVM(Model-View-ViewModel).

Το Vue js είναι open source framework το οποίο χρησιμοποιεί html, css και javascript που βοηθάνε για την δημιουργία και την ανάπτυξη πιο εύχρηστων και responsive interfaces. Είναι απόλυτα συμβατό με όλες τις βιβλιοθήκες της javascript που χρησιμοποιούνται από διάφορα javascript frameworks όπως η Angular. Είναι το κατάλληλο πιο πολύ για εφαρμογές μιας σελίδας(single page app) με την χρήση των κατάλληλων βιβλιοθηκών και εργαλείων που έχουν δημιουργηθεί για την συγκεκριμένη χρήση. Για την χρησιμοποίηση του vue js είναι η καλή γνώση και η σωστή χρήση της html, css και javascript όπως αναφέραμε παραπάνω. Ένα από τα θετικά του συγκεκριμένου framework είναι πως έχει πολύ ξεκάθαρη δομή για κάποιον που δεν το έχει ξαναχρησιμοποιήσει ξανά. Για να τρέξει η εφαρμογή θα πρέπει να υπάρχει το node.js server.

2.6	2019 Feb 4
2.5	2017 Oct 13
2.4	2017 Jul 13
2.3	2017 Apr 27

2.2	2017 Feb 26
2.1	2016 Nov 22
2.0	2016 Sep 30
1.0	2015 Oct 27
0.12	2015 Jun 12
0.11	2014 Nov 7
0.10	2014 Mar 23
0.9	2014 Feb 25

Πλεονεκτήματα

1. Το μέγεθος του Vue.js είναι αρκετά πιο μικρό γύρω στα 20KB το οποίο βοηθάει στην ταχύτητα του framework και το κάνει πιο ευέλικτο.
2. Λόγω της ομοιότητας με την Angular προσαρμόζεται πολύ εύκολα ειδικά σε προγραμματιστές
3. Παρέχει ένα πολύ καλά σχεδιασμένο οδηγό εκμάθησης έτσι ώστε και ένας αρχάριος με βασικές γνώσεις web προγραμματισμού μπορεί να κατασκευάσει μια πολύ απλή εφαρμογή.
4. Πολύ εύκολη ενσωμάτωση στην κατασκευή εφαρμογών σελίδας.

Μειονεκτήματα

1. Πάρα πολύ μικρή πηγή υποστήριξης. Μπορεί να έχει αρκετά πλεονεκτήματα αλλά ακόμα δεν έχει αποκτήσει την δυναμική άλλων framework όπως της Angular για αυτό δεν υπάρχει και επαρκής υποστήριξη σε διάφορα προβλήματα που μπορεί να προκύψουν.

Ορισμένα χαρακτηριστικά της Vue.js

Εικονικό DOM (Virtual DOM)

Το VueJS χρησιμοποιεί τη χρήση εικονικού DOM, το οποίο χρησιμοποιείται επίσης από άλλα πλαίσια όπως το React, το Ember, κ.λπ. Οι αλλαγές δεν γίνονται στο DOM, αλλά δημιουργείται ένα αντίγραφο του DOM που υπάρχει με τη μορφή δομών δεδομένων JavaScript . Όποτε πρέπει να γίνουν αλλαγές, γίνονται στις δομές δεδομένων της JavaScript και η τελευταία συγκρίνεται με την αρχική δομή δεδομένων. Οι τελικές αλλαγές στη συνέχεια ενημερώνονται στο πραγματικό DOM, το οποίο ο χρήστης θα δει να αλλάζει. Αυτό είναι καλό όσον αφορά τη βελτιστοποίηση και οι αλλαγές μπορούν να γίνουν με ταχύτερο ρυθμό.

Δεσμευτική δεδομένων (Data Binding)

Η δυνατότητα δέσμευσης δεδομένων βοηθά στον χειρισμό ή την αντιστοίχιση τιμών σε χαρακτηριστικά HTML, στην αλλαγή του στυλ, στην αντιστοίχιση τάξεων με τη βοήθεια της δεσμευτικής οδηγίας που ονομάζεται v-bind διαθέσιμο με το VueJS.

Συστατικά (Components)

Τα Components είναι ένα από τα σημαντικά χαρακτηριστικά του VueJS που βοηθά στη δημιουργία προσαρμοσμένων στοιχείων, τα οποία μπορούν να επαναχρησιμοποιηθούν σε HTML.

Χειρισμός εκδηλώσεων (Event Handling)

Το Vue js χρησιμοποιεί την εντολή v-on η οποία είναι το χαρακτηριστικό που προστίθεται στα στοιχεία DOM για να ακούσετε τα συμβάντα στο VueJS.

Κινούμενα σχέδια / Μετάβαση (Animation/ Transition)

Το VueJS παρέχει διάφορους τρόπους εφαρμογής της μετάβασης σε στοιχεία HTML όταν προστίθενται / ενημερώνονται ή καταργούνται από το DOM. Το VueJS διαθέτει ένα ενσωματωμένο στοιχείο μετάβασης που πρέπει να τυλίγεται γύρω από το στοιχείο για εφέ μετάβασης. Μπορούμε εύκολα να προσθέσουμε βιβλιοθήκες κινουμένων σχεδίων τρίτων και επίσης να προσθέσουμε περισσότερη διαδραστικότητα στη διεπαφή.

Υπολογισμένες ιδιότητες (Computed Properties)

Αυτό είναι ένα από τα σημαντικά χαρακτηριστικά του VueJS. Βοηθά να δούμε τις αλλαγές που έγιναν στα στοιχεία διεπαφής χρήστη και εκτελεί τους απαραίτητους υπολογισμούς. Δεν χρειάζεται επιπλέον κωδικοποίηση για αυτό.

Πρότυπα (Templates)

Το VueJS παρέχει πρότυπα που βασίζονται σε HTML που συνδέουν το DOM με τα δεδομένα παρουσίας Vue. Η Vue συγκεντρώνει τα πρότυπα σε εικονικές συναρτήσεις DOM Render. Μπορούμε να χρησιμοποιήσουμε το πρότυπο των λειτουργιών απόδοσης και για να το κάνουμε πρέπει να αντικαταστήσουμε το πρότυπο με τη λειτουργία απόδοσης.

Οδηγίες (Directives)

Το VueJS έχει ενσωματωμένες οδηγίες όπως v-if, v-else, v-show, v-on, v-bind και v-model, οι οποίες χρησιμοποιούνται για την εκτέλεση διαφόρων ενεργειών στο frontend.

Παρατηρητές (Watchers)

Οι παρατηρητές εφαρμόζονται σε δεδομένα που αλλάζουν. Για παράδειγμα, στα στοιχεία εισαγωγής φόρμας. Εδώ, δεν χρειάζεται να προσθέσουμε επιπλέον συμβάντα. Το Watcher φροντίζει να χειριστεί τυχόν αλλαγές δεδομένων κάνοντας τον κώδικα απλό και γρήγορο.

Δρομολόγηση (Routing)

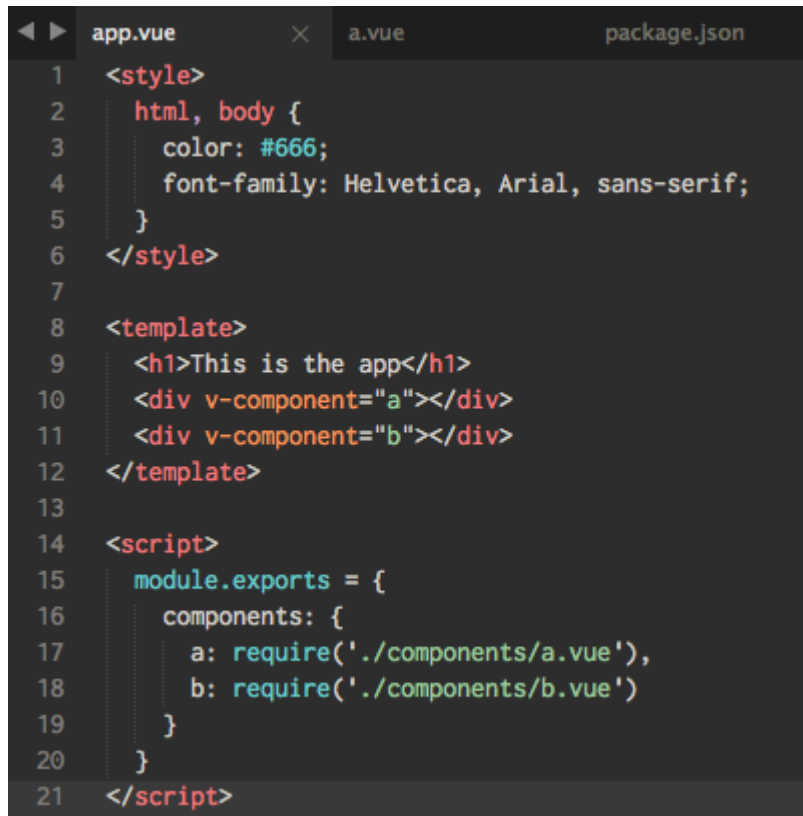
Η πλοήγηση μεταξύ σελίδων πραγματοποιείται με τη βοήθεια του vue-router.

LightWeight (LightWeight)

Το σενάριο VueJS είναι πολύ ελαφρύ και η απόδοση είναι επίσης πολύ γρήγορη.

Vue-CLI

Το VueJS μπορεί να εγκατασταθεί στη γραμμή εντολών χρησιμοποιώντας τη διεπαφή γραμμής εντολών vue-cli. Βοηθά στην κατασκευή και τη σύνταξη του έργου εύκολα χρησιμοποιώντας το vue-cli.



```
1 <style>
2   html, body {
3     color: #666;
4     font-family: Helvetica, Arial, sans-serif;
5   }
6 </style>
7
8 <template>
9   <h1>This is the app</h1>
10  <div v-component="a"></div>
11  <div v-component="b"></div>
12 </template>
13
14 <script>
15   module.exports = {
16     components: {
17       a: require('./components/a.vue'),
18       b: require('./components/b.vue')
19     }
20   }
21 </script>
```

Εικόνα 3. Κώδικας Vue.js

2.5 Backbone.js

Το backbone.js είναι μια βιβλιοθήκη JavaScript ανοιχτού κώδικα που έχει σχεδιαστεί με σκοπό την παροχή δομών σε πλούσιες εφαρμογές ιστού από πλευράς πελατών το οποίο δημιουργήθηκε στις 13 οκτωβρίου του 2010 από τον προγραμματιστή Jeremy Ashkenas ο οποίος ευθύνεται για την δημιουργία για την βιβλιοθήκη underscore.js καθώς και για την γλώσσα προγραμματισμού CoffeeScript. Για να καταφέρει να λειτουργήσει το συγκεκριμένο framework το μόνο που χρειάζεται είναι η παρουσία της βιβλιοθήκης underscore.js αλλά και της βιβλιοθήκης jquery. Παρακάτω διατυπώνουμε πως καλούμε τις βιβλιοθήκες που χρειαζόμαστε.

```
<script type = "text/javascript" src =  
"https://ajax.googleapis.com/ajax/libs/jquery/1.5.2/jquery.min  
.js"></script>
```

```
<script type = "text/javascript"  
    src =  
"https://ajax.cdnjs.com/ajax/libs/underscore.js/1.1.4/undersco  
re-min.js"></script>
```

```
<script type = "text/javascript"  
    src =  
"https://ajax.cdnjs.com/ajax/libs/backbone.js/0.3.3/backbone-  
min.js"></script>
```

Το backbone είναι βασισμένο στην αρχιτεκτονική λογισμικού MVP(Model-View-Presenter) το οποίο έχει βασιστεί από το MVC(Model-View-Controller) και χρησιμοποιείται για την δημιουργία ενός γραφικού περιβάλλοντος χρήστη. Διασφαλίζει ότι η επικοινωνία μεταξύ διακομιστή και πελάτη γίνεται εξ ολοκλήρου μέσω ενός RESTful JSON API. Επιπλέον, είναι γνωστό όχι μόνο για το ότι είναι απίστευτα ελαφρύ (κάτω των 8kb), αλλά και για τη διατήρηση διαφόρων στοιχείων μιας εφαρμογής ιστού σε συγχρονισμό.

Υπήρχε μια εποχή που οι διαδικτυακές εφαρμογές ήταν στατικές σελίδες HTML και απαιτούσαν από τους προγραμματιστές να αλλάξουν τον κώδικα HTML, CSS και JS προκειμένου να πραγματοποιήσουν αλλαγές στο περιεχόμενο. Με την εξέλιξη των γλωσσών προγραμματισμού από την πλευρά του διακομιστή όπως Java, PHP και Ruby

κ.λπ., οι στατικές σελίδες HTML έχουν δημιουργηθεί δυναμικά. Έτσι, αυτή είναι η προσέγγιση που χρησιμοποιούν οι περισσότερες διαδικτυακές εφαρμογές σήμερα.

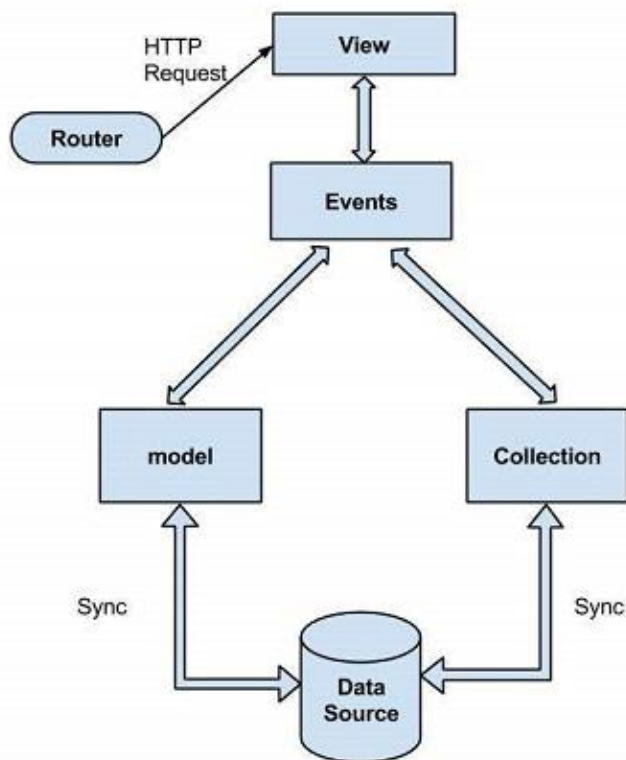
Το κύριο πρόβλημα με αυτήν την προσέγγιση είναι η μεγάλη χρήση του JavaScript, η οποία καθιστά τον κώδικα εφαρμογής εξαιρετικά δύσκολο να οργανωθεί και να διατηρηθεί. Εκεί μπαίνει το Backbone.js καθώς παρέχει στους προγραμματιστές μια πιο δομημένη προσέγγιση για τη δημιουργία εφαρμογών ιστού με βαριά JavaScript.

```
index.html
9 <div class="container">
10 <h1>User Manager</h1>
11 <hr />
12 <div class="page"></div>
13 </div>
14
15 <script src="//cdnjs.cloudflare.com/ajax/libs/jquery/1.8.2/jquery.min.js"></script>
16 <script src="//cdnjs.cloudflare.com/ajax/libs/underscore.js/1.4.2/underscore-min.js"></script>
17 <script src="//cdnjs.cloudflare.com/ajax/libs/backbone.js/0.9.2/backbone-min.js"></script>
18 </script>
19
20 $.ajaxPrefilter( function( options, originalOptions, jqXHR ) {
21   options.url = 'http://backbonejs-beginner.herokuapp.com' + options.url;
22 });
23
24 var Users = Backbone.Collection.extend({
25   url: '/users'
26 });
27
28 var UserList = Backbone.View.extend({
29   el: '.page',
30   render: function () {
31     var users = new Users();
32     users.fetch({
33       success: function () {
34         this.$el.html('!CONTENT SHOULD SHOW HERE!');
35       }
36     });
37   }
38 });
39
40
41 var Router = Backbone.Router.extend({
42   routes: {
43     '': 'home'
44   }
45 });
46
47 var userList = new UserList();
48
49 var router = new Router();
50 router.on('route:home', function () {
51   userList.render();
52 });
53
```

Εικόνα 4. Δείγμα κώδικα backbone.js

Η αρχιτεκτονική του backbone js περιέχει τις ακόλουθες ενότητες:

- HTTP Request
- Router
- View
- Events
- Model
- Collection
- Data Source



HTTP Request

Ο πελάτης στέλνει ένα αίτημα HTTP σε έναν server με τη μορφή μηνύματος όπου οι browsers, οι μηχανές αναζήτησης κ.λπ., ενεργούν όπως οι πελάτες. Ο χρήστης ζητά ένα αρχείο όπως έγγραφα, εικόνες κ.λπ., χρησιμοποιώντας το πρωτόκολλο αιτήσεων HTTP.

Router

Χρησιμοποιείται για τη δρομολόγηση εφαρμογών από την πλευρά του πελάτη και τις συνδέει με ενέργειες χρησιμοποιώντας διευθύνσεις URL. Αυτή η διεύθυνση URL αλλάζει χειροκίνητα από τον χρήστη. Η διεύθυνση URL χρησιμοποιείται από τη ραχοκοκαλιά, ώστε να κατανοεί ποια κατάσταση αίτησης θα αποσταλεί ή θα παρουσιαστεί στον χρήστη.

Ο δρομολογητής είναι ένας μηχανισμός που μπορεί να αντιγράψει τις διευθύνσεις URL για να φτάσει στην προβολή τους. Ο δρομολογητής απαιτείται όταν οι εφαρμογές ιστού παρέχουν συνδέσμους, σελιδοδείκτες και κοινόχρηστες διευθύνσεις URL για σημαντικές τοποθεσίες στην εφαρμογή.

View

Οι προβολές BackboneJS είναι υπεύθυνες για το πώς και τι πρέπει να εμφανίζονται από την εφαρμογή μας και δεν περιέχουν σήμανση HTML για την εφαρμογή. Καθορίζει μια ιδέα πίσω από την παρουσίαση των δεδομένων του μοντέλου στον χρήστη. Οι προβολές χρησιμοποιούνται για να αντικατοπτρίζουν πώς μοιάζει το μοντέλο δεδομένων.

Οι προβολές δεν γνωρίζουν τίποτα για το HTML και το CSS και σε κάθε προβολή μπορεί να ενημερωθεί ανεξάρτητα όταν αλλάζει το μοντέλο χωρίς να φορτωθεί ξανά ολόκληρη η σελίδα. Αντιπροσωπεύει το λογικό κομμάτι της διεπαφής χρήστη στο DOM.

Events

Οι εκδηλώσεις είναι τα κύρια μέρη κάθε εφαρμογής. Συνδέει τα προσαρμοσμένα συμβάντα του χρήστη σε μια εφαρμογή. Μπορούν να αναμιχθούν σε οποιοδήποτε αντικείμενο και μπορούν να δεσμεύσουν και να ενεργοποιήσουν προσαρμοσμένα συμβάντα. Μπορείτε να δεσμεύσετε τα προσαρμοσμένα συμβάντα χρησιμοποιώντας το επιθυμητό όνομα της επιλογής σας. Συνήθως, τα συμβάντα αντιμετωπίζονται συγχρόνως με τη ροή του προγράμματος.

Model

Είναι η καρδιά της εφαρμογής JavaScript που ανακτά και συμπληρώνει τα δεδομένα. Τα μοντέλα περιέχουν τα δεδομένα μιας εφαρμογής και αντιπροσωπεύουν το βασικό αντικείμενο δεδομένων στο πλαίσιο.

Τα μοντέλα αντιπροσωπεύουν επιχειρηματικές οντότητες με επιχειρηματική λογική και επικύρωση επιχείρησης. Χρησιμοποιούνται κυρίως για αποθήκευση δεδομένων και

επιχειρηματική λογική. Τα μοντέλα μπορούν να ανακτηθούν και να αποθηκευτούν στην αποθήκευση δεδομένων. Ένα μοντέλο λαμβάνει το αίτημα HTTP από τα συμβάντα που πέρασε από την προβολή χρησιμοποιώντας το δρομολογητή και συγχρονίζει τα δεδομένα από τη βάση δεδομένων και στέλνει την απάντηση πίσω στον πελάτη.

Collection

Η συλλογή είναι ένα σύνολο μοντέλων που δεσμεύει συμβάντα, όταν το μοντέλο έχει τροποποιηθεί στη συλλογή. Η συλλογή περιέχει μια λίστα μοντέλων που μπορούν να επεξεργαστούν στο βρόχο και υποστηρίζουν την ταξινόμηση και το φιλτράρισμα. Κατά τη δημιουργία μιας συλλογής, μπορούμε να καθορίσουμε τον τύπο του μοντέλου που θα έχει η συλλογή μαζί με την παρουσία ιδιοτήτων. Κάθε συμβάν που ενεργοποιείται σε ένα μοντέλο θα ενεργοποιηθεί επίσης στη συλλογή του μοντέλου.

Παίρνει επίσης το αίτημα από την προβολή, δεσμεύει συμβάντα και συγχρονίζει τα δεδομένα με τα ζητούμενα δεδομένα και στέλνει την απάντηση πίσω στον πελάτη HTTP.

Data Source

Είναι η σύνδεση που έχει ρυθμιστεί σε μια βάση δεδομένων από έναν διακομιστή και περιέχει τις πληροφορίες που ζητούνται από τον πελάτη. Η ροή της αρχιτεκτονικής BackboneJS μπορεί να περιγραφεί όπως φαίνεται στα ακόλουθα βήματα -

Ένας χρήστης ζητά τα δεδομένα χρησιμοποιώντας το δρομολογητή, ο οποίος δρομολογεί τις εφαρμογές στα συμβάντα χρησιμοποιώντας τις διευθύνσεις URL.

Η προβολή αντιπροσωπεύει τα δεδομένα του μοντέλου στον χρήστη.

Το μοντέλο και η συλλογή ανακτά και συμπληρώνει τα δεδομένα από τη βάση δεδομένων δεσμεύοντας προσαρμοσμένα συμβάντα.

Ορισμένα πλεονεκτήματα της backbone.js

- Είναι πολύ αρκετά σε μέγεθος και πολύ ελαφρύγυρω στα 72kb .

- Περιέχει περισσότερα από 100 διαθέσιμα extension.
- Όταν αλλάζει το μοντέλο του backbone.js ενημερώνεται αυτόματα η html της εφαρμογής που το χρησιμοποιεί.
- Το backbone.js έχει μία χαλαρή εξάρτηση με την jquery , δηλαδή θα μπορούσε να λειτουργήσει και χωρίς την υπαρξή της. Κάτι το οποίο δεν ισχύει για το underscore.js το οποίο χωρίς αυτό δεν υφίσταται λειτουργία του backbone.js
- Περιέχει διάφορους τύπους στοιχείων όπως τα models, τα views, τα routers, τα events και τα collections.
- Το BackboneJS επιτρέπει την ανάπτυξη εφαρμογών και το frontend με πολύ πιο εύκολο τρόπο χρησιμοποιώντας λειτουργίες JavaScript.
- Το BackboneJS παρέχει διάφορα δομικά στοιχεία όπως μοντέλα, προβολές, εκδηλώσεις, δρομολογητές και συλλογές για τη συναρμολόγηση εφαρμογών ιστού από τον πελάτη.
- Όταν ένα μοντέλο αλλάζει, ενημερώνει αυτόματα το HTML της εφαρμογής σας.
- Το BackboneJS είναι μια απλή βιβλιοθήκη που βοηθά στο διαχωρισμό της λογικής της διεπαφής χρήστη και χρήστη.
- Είναι δωρεάν και ανοιχτού κώδικα βιβλιοθήκη και περιέχει περισσότερες από 100 διαθέσιμες επεκτάσεις.
- Λειτουργεί σαν ραχοκοκαλιά για το έργο σας και βοηθά στην οργάνωση του κώδικα σας.
- Διαχειρίζεται το μοντέλο δεδομένων που περιλαμβάνει τα δεδομένα χρήστη και εμφανίζει αυτά τα δεδομένα στην πλευρά του διακομιστή με την ίδια μορφή γραμμένη στην πλευρά του πελάτη.

- Το BackboneJS έχει μια απαλή εξάρτηση με το jQuery και μια σκληρή εξάρτηση με το Underscore.js.
- Επιτρέπει τη δημιουργία εφαρμογών ιστού από τον πελάτη ή εφαρμογών για κινητά σε καλά δομημένη και οργανωμένη μορφή.

2.5.1 jQuery

Η jQuery είναι μια βιβλιοθήκη JavaScript σχεδιασμένη να απλοποιήσει την υλοποίηση σεναρίων (scripting) στην πλευρά του πελάτη (client-side) της HTML και υποστηρίζει πολλαπλούς φυλλομετρητές Ιστού. Κυκλοφόρησε τον Ιανουάριο του 2006 από τον Τζον Ρέριγκ (John Resig). Είναι ένα δωρεάν λογισμικό ανοιχτού κώδικα με άδεια του MIT. Απο τον Μάιο του 2019 χρησιμοποιείται από το 73% των 10 εκατομμυρίων πιο δημοφιλών ιστότοπων. Είναι η πιο διαδεδομένη βιβλιοθήκη javascript έχοντας 3 έως 4 φορές μεγαλύτερη χρήση από οποιαδήποτε άλλη βιβλιοθήκη. Η σύνταξη του jQuery έχει σχεδιαστεί για να διευκολύνει την πλοήγηση σε ένα έγγραφο, την επιλογή στοιχείων DOM, τη δημιουργία κινούμενων σχεδίων, τη διαχείριση συμβάντων και την ανάπτυξη εφαρμογών Ajax. Το jQuery παρέχει επίσης δυνατότητες στους προγραμματιστές να δημιουργούν προσθήκες πάνω από τη βιβλιοθήκη JavaScript. Αυτό επιτρέπει στους προγραμματιστές να δημιουργούν αφαιρέσεις για χαμηλού επιπέδου αλληλεπίδραση και κινούμενα σχέδια, προηγμένα εφέ και υψηλού επιπέδου.

2.5.2 underscore js

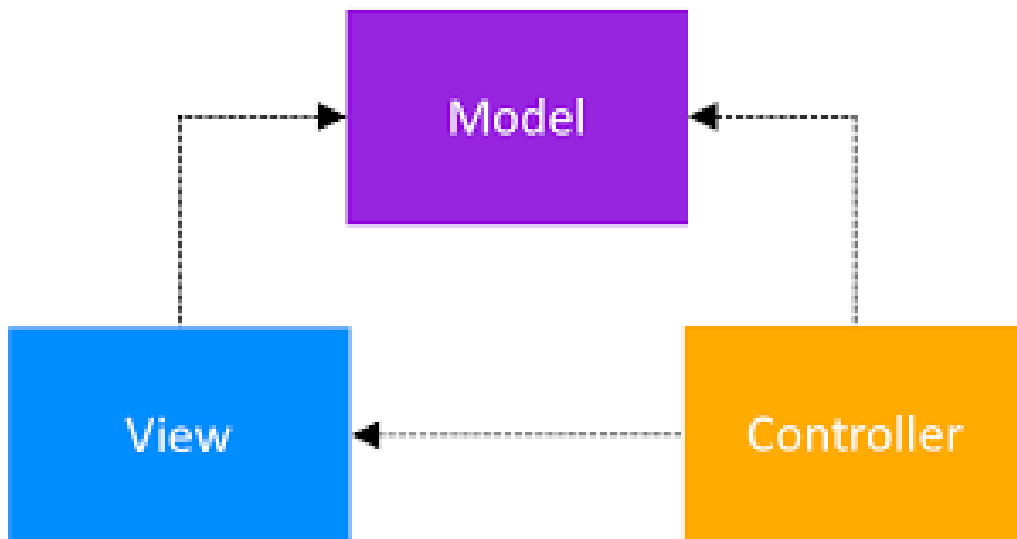
Η Underscore.js είναι μια βιβλιοθήκη της JavaScript, η οποία δίνει ένα μεγάλο μέρος της λειτουργικής υποστήριξης προγραμματισμού που παρέχεται σε γλώσσες όπως η Python, αλλά χωρίς την επέκταση οποιουδήποτε από τα ενσωματωμένα σε αυτήν αντικείμενα. Χρησιμοποιείται κυρίως για τη διαχείριση, τη μεταμόρφωση και τη χρησιμοποίηση των διαφόρων αντικειμένων. Μπορεί να αξιοποιηθεί με επάρκεια ως «μοντέλο» σε αρχιτεκτονικές τύπου Μοντέλο-Όψη-Ελεγκτής (Model-View-Controller/MVC). Το Underscore.js δημιουργήθηκε από τον Jeremy Ashkenas, ο οποίος είναι επίσης γνωστός για το Backbone.js και το CoffeeScript.

2.6 Μοντέλο αρχιτεκτονικής λογισμικού MVC (Model-View-Controller)

Το Model-View-Controller είναι ένα μοντέλο αρχιτεκτονικής λογισμικού που χρησιμοποιείται για την δημιουργία περιβάλλοντων αλληλεπίδρασης χρήστη. Το Model-View-Controller διαχωρίζεται σε τρία μέρη ώστε να διαχωριστεί η παρουσίαση της πληροφορίας στον χρήστη με την μορφή που έχει αποθηκευτεί στο σύστημα. Τα τρία μέρη είναι το αντικείμενο Model, το αντικείμενο View και ο Controller.

- Το Model ενημερώνει τις αντίστοιχες αναπαραστάσεις views και controller όταν υπάρχει αλλαγή στα δεδομένα. Αυτή η ενημέρωση επιτρέπει στα views να ενημερώνουν τη γραφική απεικόνιση.
- Το View αναπαριστά με γραφικό τρόπο την πληροφορία που περιέχει το model δημιουργώντας γραφική παρουσίαση.

- Ο Controller μπορεί να στέλνει εντολές στο μοντέλο και να ενημερώνει την κατάσταση του μοντέλου. Μπορεί επίσης να στέλνει εντολές ώστε να γίνει η αντίστοιχη αναπαράσταση των δεδομένων του μοντέλου μέσω του view.



Εικόνα 5. Μοντέλο αρχιτεκτονικής λογισμικού MVC

2.7 Μοντέλο αρχιτεκτονικής λογισμικού MVP (Model-View-Presenter)

Το Model-View-Presenter είναι μια παραλλαγή του μοντέλο αρχιτεκτονικής λογισμικού MVC και χρησιμοποιείται κυρίως για την κατασκευή διεπαφών χρήστη.

Το MVP δημιουργήθηκε στις αρχές της δεκαετίας του 1990 στο Taligent σε μια κοινοπραξία της Apple, IBM και της Hewlett-Packard. Χρησιμοποιείται για ανάπτυξη εφαρμογών σε περιβάλλον CommonPoint που βασίζεται στο C++ του Taligent. Στην συνέχεια μετεγκαταστάθηκε από την Taligent στην Java.

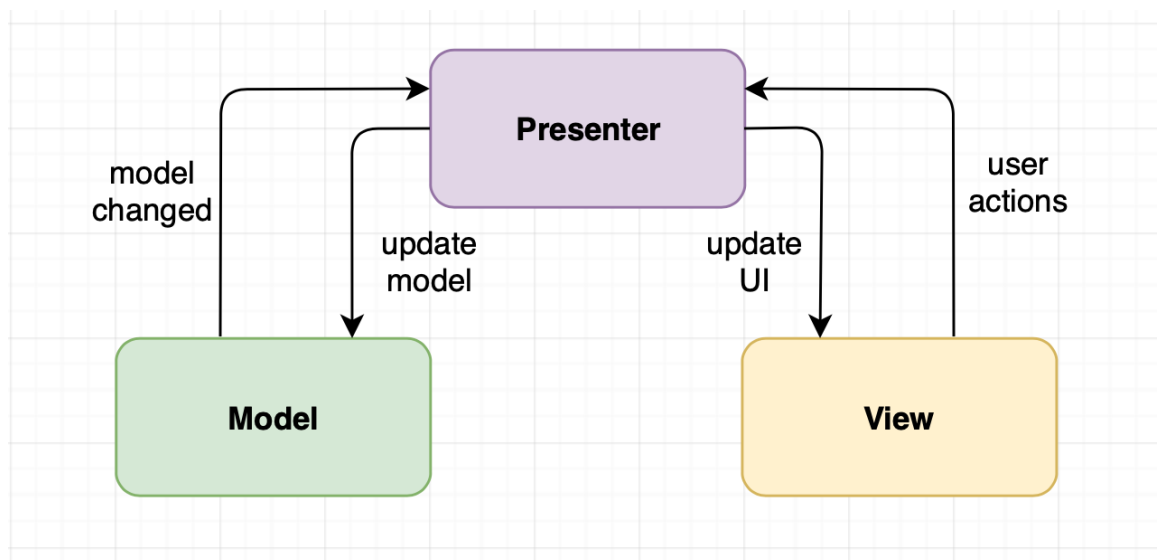
Μετά την διακοπή της Taligent το 1998 οι Andy Bower και Blair McGlashan προσάρμοσαν το MVP για να αποτελέσουν τη βάση για το πλαίσιο διεπαφής χρήστη smalltalk και η Microsoft το 2006 άρχισε να ενσωματώνει το MVP στο framework.NET.

Διαχωρίζεται σε τρία μέρη το Model, το View και ο Presenter.

1.Μοντέλο: Το μοντέλο είναι υπεύθυνο για τη διαχείριση των δεδομένων. Χαρτογραφεί τα δεδομένα που λαμβάνονται από την προέλευση δεδομένων (βάση δεδομένων ή κλήση δικτύου) μέσω του χώρου αποθήκευσης / υπηρεσίας σε αντικείμενα τομέα.

2.Παρουσιαστής: Ο παρουσιαστής περιέχει την επιχειρησιακή λογική. Είναι υπεύθυνος για την απόκτηση των δεδομένων από το χώρο αποθήκευσης / εξυπηρέτησης και την ενημέρωση της προβολής με βάση τα ληφθέντα δεδομένα. Διαχειρίζεται επίσης τις αλληλεπιδράσεις του χρήστη με την προβολή.

3.Προβολή: Είναι αποκλειστικά υπεύθυνη για την παρουσίαση των δεδομένων στον χρήστη.



Εικόνα 6. Μοντέλο αρχιτεκτονικής λογισμικού MVP

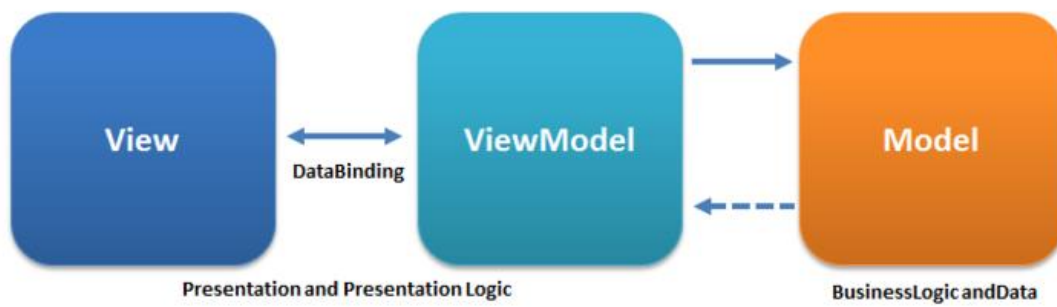
2.8 Μοντέλο αρχιτεκτονικής λογισμικού MVVM (Model-View-ViewModel)

Το MVVM είναι ένα μοντέλο αρχιτεκτονικής λογισμικού που διευκολύνει το διαχωρισμό της ανάπτυξης της γραφικής διεπαφής χρήστη είτε μέσω γλώσσας σήμανσης είτε μέσω κώδικα GUI έτσι ώστε να μην εξαρτάται η προβολή από καμία συγκεκριμένη πλατφόρμα μοντέλου. Το MVVM από τους εργαζόμενους της Microsoft Ken Cooper και Ted Peters για να απλοποιήσει τον προγραμματισμό των διεπαφών χρήστη βάσει συμβάντων. Ενσωματώθηκε στα Windows Presentation Foundation(WPF) και στο Silverlight. Το Model-View-ViewModel αναφέρεται επίσης και ως Model-View-Binder.

Model: Ο ρόλος του μοντέλου είναι ο ίδιος με τον ρόλο του στο μοντέλο MVP. Το μοντέλο χαρτογραφεί τα δεδομένα που λαμβάνονται από την προέλευση δεδομένων (βάση δεδομένων ή κλήση δικτύου) μέσω του χώρου αποθήκευσης / υπηρεσίας σε αντικείμενα τομέα.

View: Ο ρόλος της άποψης στην περίπτωση αυτή είναι ο ίδιος με εκείνον του ρόλου της στο πρότυπο MVP. Είναι υπεύθυνο για την παρουσίαση των δεδομένων στον χρήστη και είναι τόσο χαζός που μπορεί να είναι.

ViewModel: Το Viewmodel εκθέτει ροές δεδομένων σχετικών με την Προβολή. Ακόμη και στην περίπτωση αυτή είναι η γέφυρα μεταξύ του αποθετηρίου και της προβολής και περιέχει όλη την επιχειρησιακή λογική.



Εικόνα 7. Μοντέλο αρχιτεκτονικής λογισμικού MVVM

3 Μεθοδολογία

Στο πλαίσιο της σύγκρισης ανάμεσα στο backbone.js και στο vue.js θα χρησιμοποιήσουμε δύο εφαρμογές Todolist τα οποία είναι κατασκευασμένα το καθένα με τα αντίστοιχο εργαλεία Έτσι θα είναι πιο εύκολο να εξετάσουμε όσο χρειάζεται και να βρούμε τυχόν διαφορές που υπάρχουν ανάμεσα στα 2 Todolist. Τα συγκεκριμένα todolist μπορούμε να τα βρούμε στο παρακάτω link: <http://todomvc.com/> .

Τα δύο todolist τα οποία θα χρησιμοποιήσουμε είναι τα παρακάτω:

<http://todomvc.com/examples/vue/>

<http://todomvc.com/examples/backbone/>

Το κώδικα μπορούμε να τον βρούμε στο github στα αντίστοιχα link ανάλογα το freamework που μας ενδιαφέρει.

<https://github.com/tastejs/todomvc/tree/gh-pages/examples/backbone>

<https://github.com/tastejs/todomvc/tree/gh-pages/examples/vue>

3.1 Μελέτη των εργαλείων

Για την συγκεκριμένη μελέτη των δύο frameworks θα χρησιμοποιήσουμε τον φυλλομετρητή Google Chrome αλλά και το εργαλείο μέτρησης απόδοσης Dev tools. Όπως και ορισμένα extensions τα οποία έχουμε ενσωματώσει στον browser μας όπως το Performane-analyzer, το pageloadtime η το lighthouse.

3.2 Αξιολόγηση-Σύγκριση εργαλείων

Τα δεδομένα που θα μας βοηθήσουν να συγκρίνουμε και να αξιολογήσουμε τα εργαλεία είναι:

Η κατανάλωση μνήμης.

Ο χρόνος φόρτωσης της σελίδας.

Η CPU time

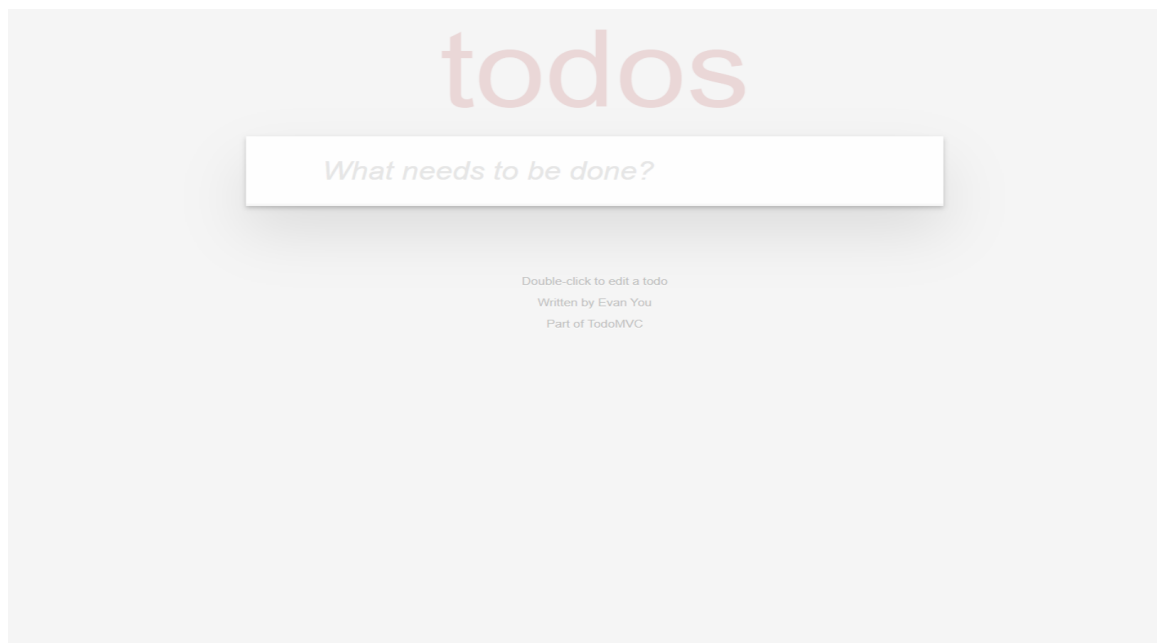
Ποιότητα εφαρμογής.

4. Παρουσίαση-Έλεγχος μετρήσεων

Στο κεφάλαιο 4 θα παρουσιάσουμε τα δύο todolist και επίσης θα αναφέρουμε ποια εργαλεία χρησιμοποιούμε για την κάθε μέτρηση που κάνουμε και θα δείξουμε και ορισμένα κομμάτια κώδικα των 2 εφαρμογών.

4.1 Παρουσίαση To do list

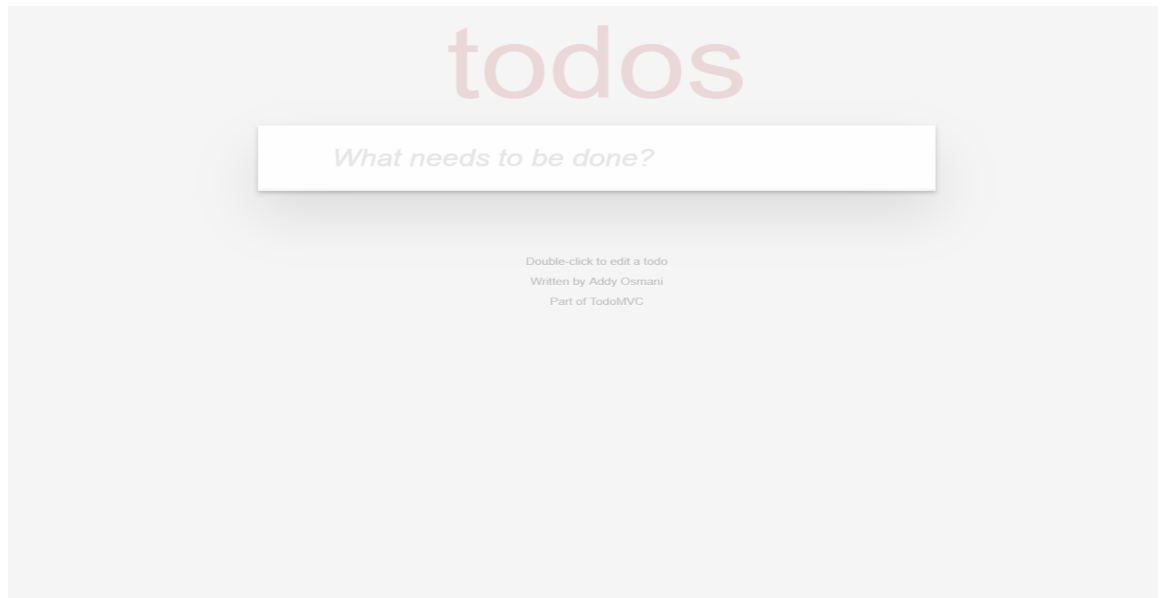
Μία εφαρμογή to do list είναι χρήσιμη για έναν χρήστη γιατί τον βοηθάει να θυμάται και να τακτοποιεί τις υποχρεώσεις του και τις δραστηριότητές του.



Εικόνα 8. Η todolist κατασκευασμένη από τον [Evan You](#)

```
15 </header>
16 <section class="main" v-show="todos.length">
17   <input id="toggle-all" class="toggle-all" type="checkbox" v-model="allDone">
18   <label for="toggle-all">Mark all as complete</label>
19   <ul class="todo-list">
20     <li class="todo" v-for="todo in filteredTodos" :key="todo.id" :class="{completed: todo.comple
21     <div class="view">
22       <input class="toggle" type="checkbox" v-model="todo.completed">
23       <label @dblclick="editTodo(todo)">{{todo.title}}</label>
24       <button class="destroy" @click="removeTodo(todo)"></button>
25     </div>
26     <input class="edit" type="text" v-model="todo.title" v-todo-focus="todo == editedTodo" @blu
27     </li>
28   </ul>
29 </section>
30 <footer class="footer" v-show="todos.length">
31   <span class="todo-count">
32     <strong v-text="remaining"></strong> {{pluralize('item', remaining)}} left
33   </span>
34   <ul class="filters">
35     <li><a href="#/all" :class="{selected: visibility == 'all'}">All</a></li>
36     <li><a href="#/active" :class="{selected: visibility == 'active'}">Active</a></li>
37     <li><a href="#/completed" :class="{selected: visibility == 'completed'}">Completed</a></li>
38   </ul>
39   <button class="clear-completed" @click="removeCompleted" v-show="todos.length > remaining">
40     Clear completed
41 </button>
```

Εικόνα 9. Η index.html της todolist με Vue.js



Εικόνα 10. Η todolist κατασκευασμένη από τον [Addy Osmani](#)


```

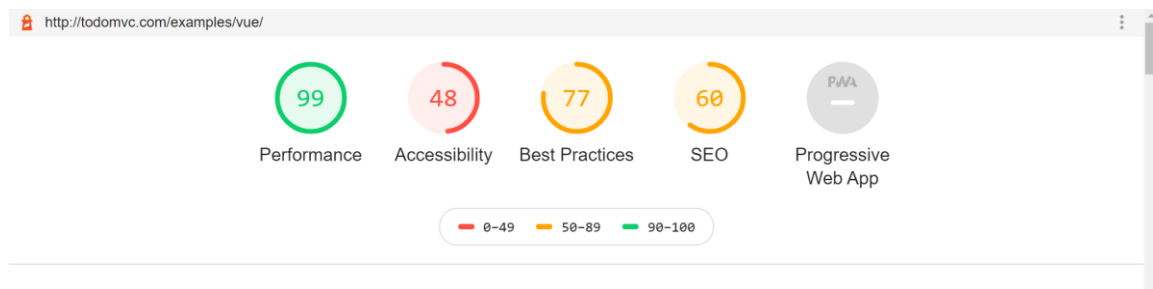
8 </head>
9 <body>
10 <section class="todoapp">
11 <header class="header">
12 <h1>todos</h1>
13 <input class="new-todo" placeholder="What needs to be done?" autofocus>
14 </header>
15 <section class="main">
16 <input class="toggle-all" id="toggle-all" type="checkbox">
17 <label for="toggle-all">Mark all as complete</label>
18 <ul class="todo-list"></ul>
19 </section>
20 <footer class="footer"></footer>
21 </section>
22 <footer class="info">
23 <p>Double-click to edit a todo</p>
24 <p>Written by <a href="https://github.com/addyosmani">Addy Osmani</a></p>
25 <p>Part of <a href="http://todomvc.com">TodoMVC</a></p>
26 </footer>
27 <script type="text/template" id="item-template">
28 <div class="view">
29 <input class="toggle" type="checkbox" <%= completed ? 'checked' : '' %>
30 <label><%= title %></label>
31 <button class="destroy"></button>
32 </div>
33 <input class="edit" value="<%= title %>">
34 </script>

```

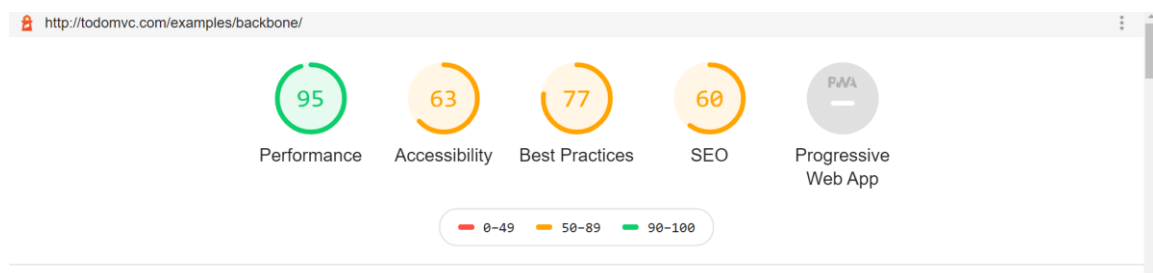
Εικόνα 11. Η index.html της todolist με backbone.js

4.2 Ποιότητα εφαρμογής

Για την ποιότητα εφαρμογής θα χρησιμοποιήσουμε το extension Lighthouse. Το Lighthouse είναι ένα open source εργαλείο το οποίο ελέγχει την ποιότητα της εφαρμογής και προτείνει λύσεις για την βελίωσή της. Κάνει ελέγχους σχετικά με την προσβασιμότητα της εφαρμογής, με το SEO κ.α.



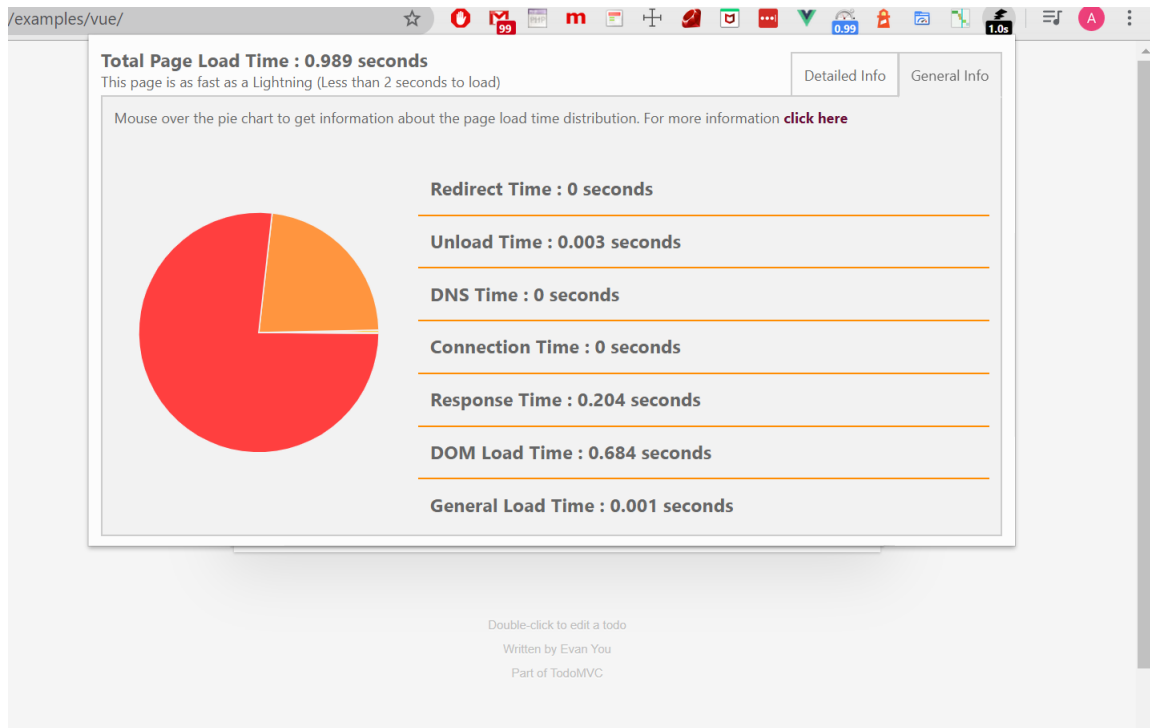
Εικόνα 12. Αποτελέσματα Lighthouse για την απόδοση της todolist με Vue.js



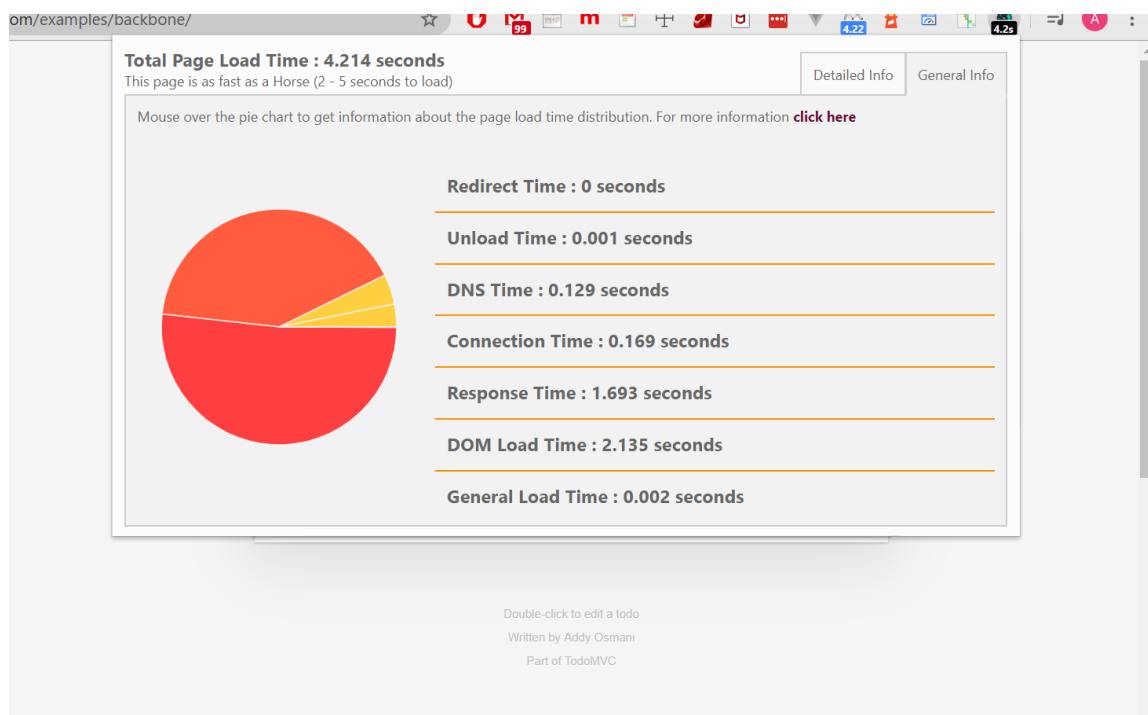
Εικόνα 13. Αποτελέσματα Lighthouse για την απόδοση της todolist με Backbone.js

4.3 Ο χρόνος φόρτωσης της σελίδας

Ο χρόνος φόρτωσης της σελίδας θα μετρηθεί με το εργαλείο Analyze page performance. Το Analyze page performance είναι ένα extension το οποίο ενσωματώνουμε στον chrome και ελέγχουμε την ταχύτητα της σελίδας. Παρέχει λεπτομερείς εξηγήσεις των συμβάντων όπως και γράφημα πίτας για την οπτική απεικόνιση των αποτελεσμάτων.



Εικόνα 14. Παρούσιαση απόδοση της φόρτωσης της todolist με vue.js



Εικόνα 15. Παρουσίαση απόδοση της φόρτωσης της todolist με backbone.js

4.4 Κατανάλωση μνήμης

Η μνήμη που καταναλώνει η εφαρμογή που είναι κατασκευασμένη σε vue.js φαίνεται στην παρακάτω εικόνα.

Μέγεθος:	260 KB (266.589 byte)
Μέγεθος στο δίσκο:	272 KB (278.528 byte)
Περιέχει:	12 αρχεία, 8 φακέλους

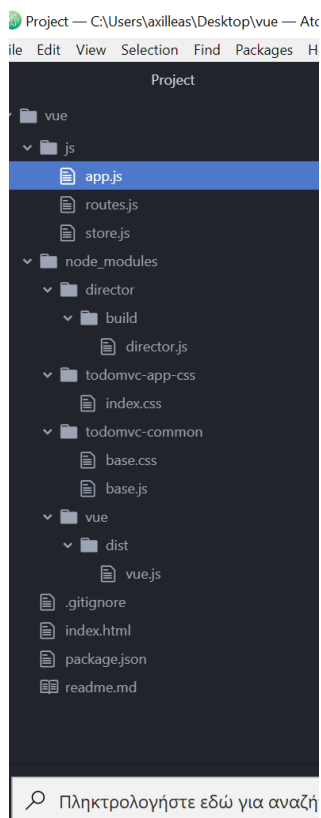
Εικόνα 16. Ο χώρος κατανάλωσης της εφαρμογής todolist σε vue.js

Ενώ η μνήμη που καταναλώνει η εφαρμογή που είναι κατασκευασμένη σε `backbone.js` φαίνεται στην παρακάτω εικόνα.

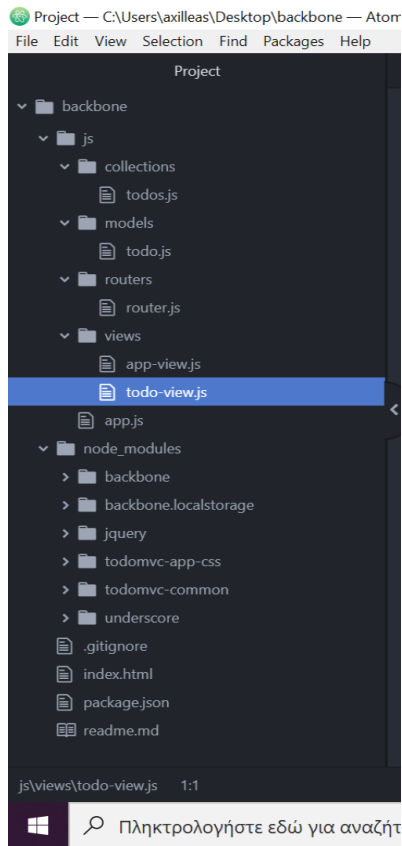
Μέγεθος:	399 KB (409.516 byte)
Μέγεθος στο δίσκο:	416 KB (425.984 byte)
Περιέχει:	17 αρχεία, 13 φακέλους

Εικόνα 16. Ο χώρος κατανάλωσης της εφαρμογής `todolist` σε `vue.js`

Παρατηρούμε συγκρίνοντας τις δύο εικόνες πως η `backbone.js` καταλαμβάνει πολύ περισσότερο χώρο στην μνήμη από την `vue.js` και σε αυτό παίζει και πολύ μεγάλο ρόλο η ποσότητα των αρχείων αλλά και το περιεχόμενό τους. Μπορούμε να θυμηθούμε την ύπαρξη 2 βιβλιοθηκών στην `backbone.js` το οποίο και αυτό παίζει μεγάλο ρόλο της `jquery` αλλά και της υποχρεωτικής `underscore.js` διότι χωρίς αυτή δεν μπορεί να λειτουργήσει η `backbone.js`.



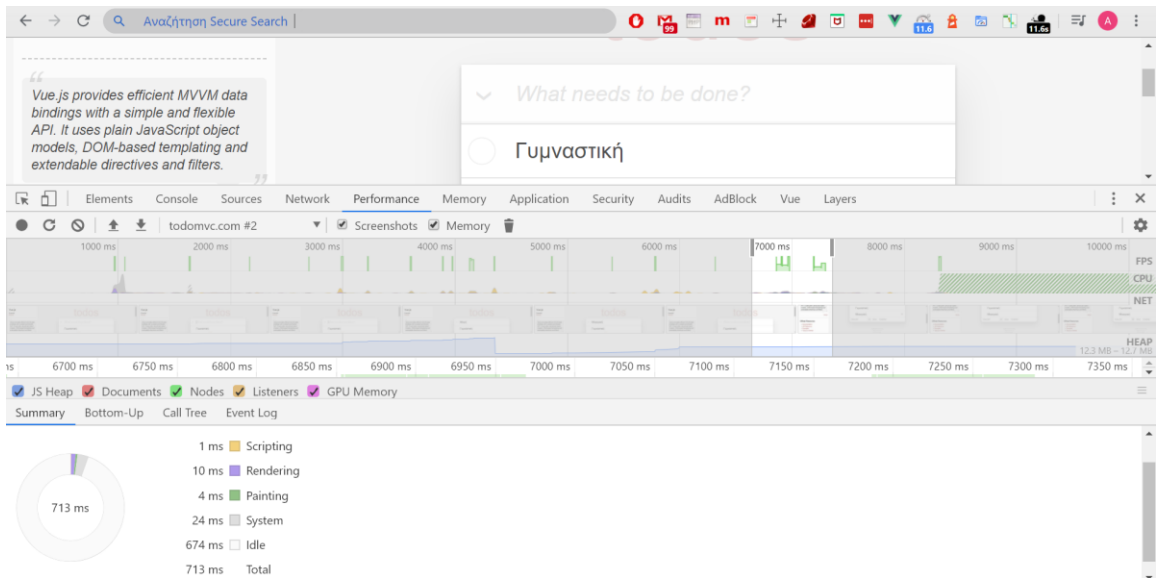
Εικόνα 17. Δομή της `todolist` με `vue.js`



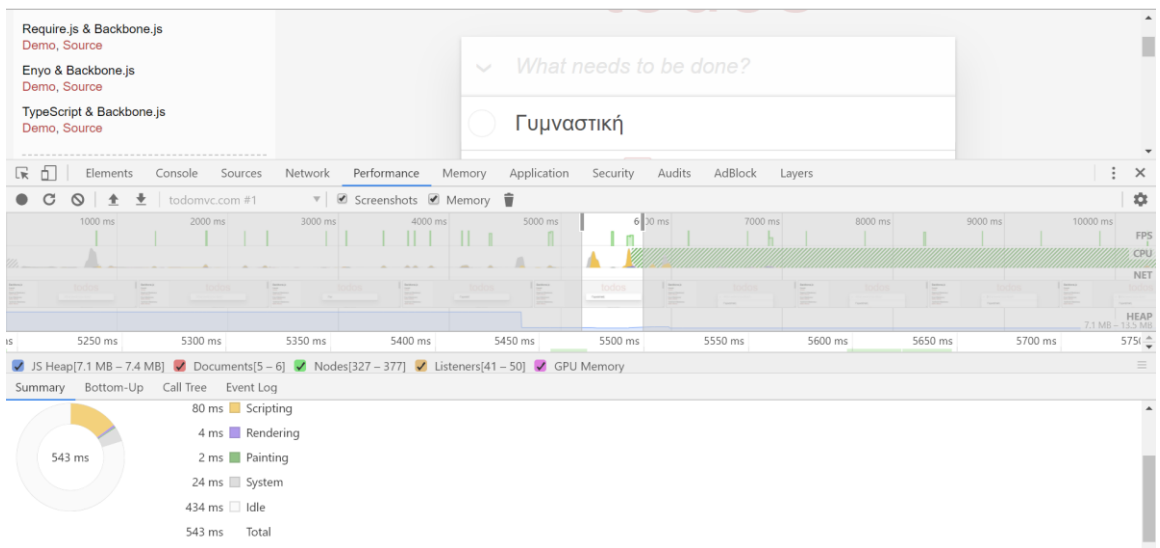
Εικόνα 18. Δομή της todolist με backbone.js

4.5 Μέτρηση CPU time

Για την μέτρηση της CPU χρησιμοποιήσαμε το dev tools του chrome. Διαλέγοντας την καρτέλα performance πατάμε Ctrl + E και ξεκινάει η εγγραφή της εφαρμογής βάζοντας στο πεδίο που έχουμε μια υποχρέωση μας και μόλις την εισάγουμε σταματάμε την εγγραφή. Στην συνέχεια ελέγχουμε τα διαγράμματα που δημιουργήθηκαν αλλά και τους χρόνους. Υπολογίζοντας την τιμή που υπάρχει κεντρικά στο διάγραμμα μείον την τιμή του πεδίου Idle υπολογίζεται η τιμή που χρειάστηκε για την υλοποίηση της ενέργειας που κάναμε.



Εικόνα 19. CPU time vue.js



Εικόνα 20. CPU time backbone.js

5. Συμπεράσματα

Σύμφωνα με την μελέτη που κάναμε αλλά και την σύγκριση ανάμεσα στα δύο frameworks βλέπουμε πως υπάρχει υπάρχουν ορισμένες διαφορές όπου μας δίνουν την εντύπωση πως η vue.js υπερτερεί από την backbone.js λόγω του ότι είναι πιο γρήγορη αλλά και πιο απλή δομή και δεν εξαρτάται από κάποια άλλη βιβλιοθήκη για την λειτουργία της . Παρόλα τα αποτελέσματα που είδαμε από τις συγκρίσεις που κάναμε πρέπει να καταλάβουμε ότι κάθε framework έχει τις δικές του δυνατότητες και τις δικές του εξαρτήσεις. Η επιλογή που θα κάνουμε για ένα framework βασίζεται πολύ και στις ανάγκες που έχει η εφαρμογή που πρόκειται να υλοποιηθεί. Αυτό η λογική φυσικά ισχύει και στα υπόλοιπα frameworks που υπάρχουν και όχι μόνο σε αυτά που χρησιμοποιήσαμε.

5.1 Επεκτάσεις για το μέλλον

Οι μετρήσεις και οι συγκρίσεις που κάναμε μας βοήθησαν να βγάλουμε ορισμένα συμπεράσματα αλλά θα πρέπει να γίνουν και άλλες επεκτάσεις στο μέλλον για να μπορέσουν να βγουν περισσότερα συμπεράσματα και να έχουμε μια πιο καθαρή εικόνα έτσι ώστε να μπορέσουμε να έχουμε περισσότερα κριτήρια για την επιλογή ενός framework όπως:

- Η εξέταση σε περισσότερους από έναν browser όπως τον mozilla firefox ή τον edge γιατί υπάρχει η πιθανότητα πολλές δυνατότητες να μην υποστηρίζονται από τους προαναφερθέντες browser ή και άλλους.
- Η σύγκριση των δύο συγκεκριμένων framework ή και περισσότερων και διαφορετικών όπως της Angular ,της Ember κ.α. με την χρήση μιας πιο πολύπλοκης εφαρμογής με περισσότερες δυνατότητες.

A.1 Αναφορές ή Παραπομπές

Η βιβλιογραφία είναι κατάλογος από βιβλία ή άλλο υλικό για ένα ορισμένο θέμα. Θα πρέπει να περιλαμβάνει τουλάχιστον όλες τις πηγές που έχουν αναφερθεί στην εργασία. Ένας κατάλογος παραπομπών περιλαμβάνει μόνο τις πηγές που αναφέρθηκαν στην εργασία. Η βιβλιογραφία και ο κατάλογος παραπομπών θα πρέπει να είναι σε αλφαβητική σειρά με πρώτο το επίθετο του συγγραφέα.

A.2 Βιβλιογραφία

A.2.1 Άρθρα σε βιβλία

O'Reilly Media.

Crockford, D. 2008. JavaScript: The Good Parts.

De, S., 2014. Backbone.js Patterns and Best Practices.

Packt Publishing.

By Osmani Addy

Developing Backbone.js Applications” <https://addyosmani.com/backbonefundamentals/>”

Alex Kyriakidis, Kostas Maniatis

The Majesty Of Vue.js

B.2 Ανέκδοτες Πηγές (Εργασίες / Διατριβές)

Χαρκιολάκη Δέσποινα (2019). ΣΥΓΚΡΙΤΙΚΗ ΜΕΛΕΤΗ ΜΕΤΑΞΥ ΤΗΣ JAVASCRIPT ΒΙΒΛΙΟΘΗΚΗΣ REACTJS ΚΑΙ ΤΟΥ ΠΛΑΙΣΙΟΥ ΑΝΑΠΤΥΞΗΣ ANGULAR

Μητρονάτσιος Αναστάσης (2018). ΑΝΑΠΤΞΗ ΕΦΑΡΜΟΓΗΣ ΔΙΑΧΕΙΡΙΣΗΣ ΧΡΟΝΟΥ: ΑΞΙΟΠΟΙΗΣΗ ΤΟΥ ANGULAR JAVASCRIPT FRAMEWORK ΚΑΙ ΣΥΓΚΡΙΣΗ ΤΟΥ ΜΕ ΤΗ ΒΙΒΛΙΟΘΗΚΗ JQUERY

Παπουτσάκης Χρήστος (2017). Backbone, AngularJS, Ember: Συγκριτική ανάλυση και σενάρια χρήσης JavaScript frameworks

Γεώργιος Καλλίνικος (2018) Μελέτη σύγχρονων JavaScript Frameworks και σύγκριση αυτών στον τομέα των φορμών

Αντώνης Κοτανίδης (2019) Σχεδίαση και ανάπτυξη διαδικτυακού καταστήματος με την χρήση VueJS , NuxtJS και Firebase

A.3 Ιστοσελίδες

<https://en.wikipedia.org/wiki/Backbone.js>

https://repository.kallipos.gr/bitstream/11419/3975/1/01_chapter_6.pdf

<https://en.wikipedia.org/wiki/Underscore.js>

<https://en.wikipedia.org/wiki/Vue.js>

<https://vuejs.org/>

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

<https://skillcrush.com/blog/what-is-a-javascript-framework/>

<https://devrlix.com/backbone-js-a-detailed-guide-for-beginners/>

<https://en.wikipedia.org/wiki/JQuery>

<https://en.wikipedia.org/wiki/Backbone.js>

<https://en.wikipedia.org/wiki/Vue.js>

<https://vuejs.org/>

<https://www.tutorialspoint.com/vuejs/index.htm>

<https://www.tutorialspoint.com/backbonejs/index.htm>

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

<https://skillcrush.com/blog/what-is-a-javascript-framework/>

A.4 Κυρώσεις για λογοκλοπή

Λαμβάνοντας υπόψη τον νόμο περί λογοκλοπής οι οποιαδήποτε πηγές έχουν χρησιμοποιηθεί για την μελέτη του συγκεκριμένου θέματος εργασίας αναφέρονται στις πηγές της βιβλιογραφίας.