

ΟΠΑ 2020

Συγκριτική μελέτη των εργαλείων Angular και React.

Συγκριτική μελέτη των αρχιτεκτονικών «Single-Page Application» και «Isomorphic Web Application».

[Δημήτρης-Σταύρος Κακούρης \(3160043\)](#)

[Γιώργος Βίδος \(3160014\)](#)

Εργασία Τεχνολογίες και προγραμματισμός στον Ιστό 2020

Περιεχόμενα

ΜΕΡΟΣ Α:	4
ANGULAR	4
Από που δημιουργήθηκε η ανάγκη για την Angular;.....	5
Γιατί χρησιμοποιούμε Angular:	5
Γιατί δεν χρησιμοποιούμε Angular:	7
REACT	10
Γιατί χρησιμοποιούμε REACT:	11
Γιατί δεν χρησιμοποιούμε REACT:.....	14
ΜΕΡΟΣ Β	21
Single-Page Application:	21
Γιατί χρησιμοποιούμε Single-Page Application:	21
Γιατί δεν χρησιμοποιούμε Single-Page Application:	23
Isomorphic Web Application.....	25
Γιατί χρησιμοποιούμε Isomorphic Web Application:	25
Γιατί δεν χρησιμοποιούμε Isomorphic Web Application:	27
Βιβλιογραφία	31
Μέρος Α	31
Μέρος Β	31

Εισαγωγή:

Αυτή η εργασία αποτελείται από δύο μέρη. Στο πρώτο μέρος ασχοληθήκαμε με την ANGULAR και την REACT οι οποίες είναι frameworks και libraries αντίστοιχα και βοηθάνε στην κατασκευή ιστοσελίδων και όχι μόνο. Η Angular είναι ένα Javascript framework, ανοιχτού κώδικα, της οποίας η χρήση είναι στην δημιουργία web applications με τη βοήθεια της HTML και της Typescript. Η REACT από την άλλη είναι μια ανοιχτή βιβλιοθήκη της JavaScript ή οποία δημιουργήθηκε από το facebook και χρησιμοποιείται για την δημιουργία διεπαφών χρήστη. Στο δεύτερο μέρος αναλύσαμε τις Single-Page Applications και τις Isomorphic Web Applications οι οποίες είναι τρόποι υλοποίησης web applications. Οι Single-Page Applications ανανεώνουν τα δεδομένα τους δύνάμικα χωρίς να χρειάζεται ο σέρβερ να αποστέλλει όλα τα δεδομένα από την αρχή παραμόνο τις καινούργιες πληροφορίες που ζήτησε ο χρήστης. Οι Isomorphic Web Applications είναι μια εφαρμογή ιστού που συνδυάζει μια server-rendered web app με μια single-page application.

ΜΕΡΟΣ Α:

ANGULAR

Η Angular δημιουργήθηκε από το 2008 από 2 υπαλλήλους της Google τον Misko Hevery και τον Adam Abrons. Στο πρωτοεκίνημα η Angular ονομαζόταν AngularJS και είχε αναπτυχθεί εξολοκλήρου σε JavaScript. Το 2014 η Angular ανακοινώθηκε και ο 2016 εκδόθηκε η νέα έκδοση της Angular ή επωνομαζόμενη Angular 2+.

Η Angular (Angular 2+) είναι μια πλατφόρμα και ένα JavaScript framework το οποίο χρησιμοποιείται για την κατασκευή εφαρμογών μια σελίδας χρησιμοποιώντας HTML και TypeScript και είναι ο απόγονος της AngularJS. Η Angular είναι γραμμένη σε TypeScript. Επίσης η Angular εφαρμόζει βασικές και προαιρετικές λειτουργίες ως σύνολο βιβλιοθηκών TypeScript που εισάγετε στις εφαρμογές σας. Επιπρόσθετα η Angular Αναπτύχθηκε και διατηρείται από την Google και πρωτοκυκλοφόρησε Σεπτέμβριο του 2016. Αυτή την στιγμή η πιο πρόσφατη έκδοση της Angular είναι η 10 (Ιούλιος 2020). Το Google AdWords, ένα από τα πιο σημαντικά έργα της Google, το οποίο χρησιμοποιεί την Angular, οπότε μπορούμε να πούμε ανεπιφύλακτα ότι η Angular ήρθε για να μείνει.

Η Angular θεωρείται ένα framework επειδή προσφέρει ισχυρές απόψεις σχετικά με το πώς θα πρέπει να δομηθεί η εφαρμογή μας. Έχει επίσης πολύ περισσότερη λειτουργικότητα «εκτός συσκευασίας». Δεν χρειάζεται να αποφασίσουμε ποιες βιβλιοθήκες δρομολόγησης θα χρησιμοποιήσουμε, μπορούμε απλώς να ξεκινήσουμε την κωδικοποίηση. Ωστόσο, ένα μειονέκτημα είναι ότι έχουμε λιγότερη ευελιξία - πρέπει δηλαδή να χρησιμοποιήσουμε αυτά που παρέχει η Angular.

Το Angular παρέχει τα εξής :

- Πρότυπα, βασισμένα σε εκτεταμένη έκδοση του HTML
- Προστασία XSS
- Ένωση εξάρτησης
- Αιτήματα Ajax από @ angular / HTTP
- Δρομολόγηση, παρέχεται από @ angular / router
- Ενσωμάτωση CSS συστατικών
- Βοηθητικά προγράμματα για στοιχεία δοκιμής μονάδας.
- @ γωνιακή / φόρμες για οικοδομικές φόρμες

Από που δημιουργήθηκε η ανάγκη για την Angular;

Πριν την δημιουργία της angular οι περισσότερες ιστοσελίδες ήταν βασισμένες σε εφαρμογές πολλών σελίδων . Δηλαδή όταν ένας χρήστης πατούσε πάνω σε ένα σύνδεσμο ο περιηγητής έπρεπε να φέρει τα απαιτούμενα αρχεία HTML που το είχαν ζητηθεί από τον διακομιστή. Ανάλογα λοιπόν με την ταχύτητα του ιντερνέτ και την ανταπόκριση του διακομιστή ,μπορεί να έπαιρνε πολύ ώρα μέχρι να μας εμφανιζόντουσαν τα αποτελέσματα που ζητήσαμε. Σταδιακά όμως οι συσκευές άρχισαν να έχουν πιο γρήγορη απόδοση ώστε οι εφαρμογές να εκτελούνται πιο γρηγορά στον περιηγητή. Έτσι λοιπόν δημιουργήθηκε η ανάγκη για την κατασκευή εφαρμογών μια σελίδας Single-Page Applications (SPAs)

Γιατί χρησιμοποιούμε Angular:

1)Λεπτομερής τεκμηρίωση

Η Angular διαθέτει λεπτομερή τεκμηρίωση όπου οι προγραμματιστές μπορούν να βρουν όλες τις απαραίτητες πληροφορίες χωρίς να ρωτήσουν τους συναδέλφους τους. Ως αποτέλεσμα, οι προγραμματιστές μπορούν γρήγορα να βρουν τεχνικές λύσεις και να επιλύσουν αναδυόμενα ζητήματα.

2)Υποστήριξη από την Google

Πολλοί προγραμματιστές θεωρούν ότι η Google υποστηρίζει την Angular οπότε είναι ένα άλλο πλεονέκτημα της Angular, καθιστώντας την πλατφόρμα αξιόπιστη. Στο ng-conf 2017, οι προγραμματιστές της Angular επιβεβαίωσαν ότι η Google θα υποστηρίξει την Angular σε μακροπρόθεσμη βάση.

3)Υπέροχο οικοσύστημα συστατικών τρίτων

Η δημοτικότητα της Angular είχε ως αποτέλεσμα την εμφάνιση χιλιάδων επιπλέον εργαλείων και στοιχείων που μπορούν να χρησιμοποιηθούν σε εφαρμογές Angular. Ως αποτέλεσμα, μπορείτε να λάβετε πρόσθετες λειτουργίες και βελτιώσεις παραγωγικότητας.

4)Αρχιτεκτονική με βάση στοιχεία

Στη δεύτερη έκδοση, η Angular μετατοπίστηκε από ένα MVC σε αρχιτεκτονική που βασίζεται σε στοιχεία. Σύμφωνα με αυτήν την αρχιτεκτονική, μια εφαρμογή χωρίζεται σε ανεξάρτητα λογικά και λειτουργικά στοιχεία. Αυτά τα στοιχεία μπορούν εύκολα να αντικατασταθούν και να αποσυνδεθούν, καθώς και να επαναχρησιμοποιηθούν σε άλλα μέρη μιας εφαρμογής. Επιπλέον, η ανεξαρτησία των στοιχείων διευκολύνει τη δοκιμή μιας εφαρμογής ιστού και διασφαλίζει ότι κάθε στοιχείο λειτουργεί άψογα.

5)Μεταγλωττιστής πριν από το χρόνο

Ο μεταγλωττιστής AOT της Angular μετατρέπει το TypeScript και το HTML σε JavaScript κατά τη διαδικασία κατασκευής. Αυτό σημαίνει ότι ο κώδικας καταρτίζεται πριν το πρόγραμμα περιήγησης φορτώσει την εφαρμογή ιστού σας, ώστε να αποδίδεται πολύ πιο γρήγορα. Ένας μεταγλωττιστής AOT είναι επίσης πολύ πιο ασφαλής από έναν μεταγλωττιστή just-in-time (JIT).

6)CLI

Είναι πιθανώς το πιο αγαπημένο χαρακτηριστικό για την πλειονότητα των προγραμματιστών Angular. Αυτοματοποιεί ολόκληρη τη διαδικασία ανάπτυξης καθιστώντας την αρχικοποίηση, τη διαμόρφωση και την ανάπτυξη της εφαρμογής όσο το δυνατόν πιο εύκολη. Η Angular CLI σας επιτρέπει να δημιουργήσετε ένα νέο έργο Angular, να προσθέσετε λειτουργίες σε αυτό και να εκτελέσετε δοκιμές μονάδας και end-to-end με μερικές απλές εντολές. Δεν αυξάνει μόνο την ποιότητα του κώδικα, αλλά επίσης διευκολύνει σημαντικά την ανάπτυξη.

7)Angular στοιχεία

Ξεκινώντας από την Angular 6, οι προγραμματιστές μπορούν εύκολα να προσθέσουν προσαρμοσμένα στοιχεία σε οποιαδήποτε εφαρμογή ιστού που έχει δημιουργηθεί με React, JQuery, Vue ή οποιοδήποτε άλλο περιβάλλον.

8) Ivy Renderer

Είναι ένα άλλο εξαιρετικό χαρακτηριστικό που έγινε διαθέσιμο με την Angular 6. Το Ivy Renderer μεταφράζει τα στοιχεία και τα πρότυπα μιας εφαρμογής σε κώδικα JavaScript που μπορεί να εμφανιστεί από ένα πρόγραμμα περιήγησης. Το κύριο χαρακτηριστικό αυτού του εργαλείου είναι η τεχνική ανακίνησης δέντρων. Κατά τη διάρκεια της απόδοσης, ο Ivy αφαιρεί τον αχρησιμοποίητο κώδικα για να τον κάνει σαφέστερο και μικρότερο. Ως αποτέλεσμα, οι εφαρμογές ιστού φορτώνουν πιο γρήγορα.

9) Angular υλικό

Αυτή η συλλογή στοιχείων σχεδιασμού υλικών βελτιστοποιημένη για την Angular επιτρέπει στους προγραμματιστές να ενσωματώνουν εύκολα στοιχεία διεπαφής χρήστη.

10) Έγχυση εξάρτησης

Είναι ένα αναμφισβήτητο πλεονέκτημα της Angular. Η έγχυση εξάρτησης αναφέρεται σε ένα αντικείμενο που παρέχει τις εξαρτήσεις ενός άλλου αντικειμένου. Αυτές οι εξαρτήσεις καθορίζουν τον τρόπο με τον οποίο συνδέονται διάφορα στοιχεία και δείχνουν πώς οι αλλαγές σε ένα μέρος του κώδικα επηρεάζουν άλλα μέρη. Από τη μία πλευρά, η χρήση έγχυσης εξάρτησης καθιστά τον κώδικα πιο ευανάγνωστο και διατηρήσιμο. Μπορεί να μειώσει σημαντικά το χρόνο που δοκιμάσατε και, συνεπώς, να μειώσετε το κόστος της ανάπτυξης ιστού. Ξεκινώντας από την έκδοση 2, η Angular παρέχει στους προγραμματιστές ένα ξεχωριστό δέντρο εγχυτήρων εξάρτησης που μπορεί να αλλάξει ή να αντικατασταθεί χωρίς να αναδιαμορφώσει όλα τα στοιχεία. Αλλά από την άλλη πλευρά, η έγχυση εξάρτησης μπορεί να είναι χρονοβόρα και μπορεί να είναι δύσκολο να δημιουργηθούν εξαρτήσεις για εξαρτήματα.

Γιατί δεν χρησιμοποιούμε Angular:

1) Δύσκολη να τη μάθεις

Οι περισσότεροι προγραμματιστές ισχυρίζονται ότι η Angular είναι αρκετά περίπλοκη, γι' αυτό μπορεί να είναι δύσκολη για τους νεοεισερχόμενους. Και παρόλο που η

αρχιτεκτονική που βασίζεται σε στοιχεία καθιστά τον κώδικα πιο ευανάγνωστο, είναι ακόμα δύσκολο να διαχειριστείτε και να ορίσετε εξαρτήσεις μεταξύ των στοιχείων. Επίσης η TypeScript είναι δύσκολη από μόνη της οπότε χρειάζεστε επιπλέον χρόνο για να την μάθετε.

2)Κακές επιλογές SEO

Παρόλο που η ομάδα Angular καταβάλλει κάθε δυνατή προσπάθεια για να κάνει την Angular SEO-friendly, πολλοί προγραμματιστές εξακολουθούν να παραπονούνται για την κακή προσβασιμότητα για τα προγράμματα ανίχνευσης αναζήτησης. Αυτό εξηγείται από το γεγονός ότι οι εφαρμογές μιας σελίδας αλλάζουν συχνά περιεχόμενο και μετα-ετικέτες χρησιμοποιώντας JavaScript. Ορισμένα bot αναζήτησης δεν μπορούν να δουν αυτές τις αλλαγές και επομένως βλέπουν μόνο το βασικό πρότυπο σελίδας. Εκτός αυτού, οι σύνδεσμοι σε βάθος σε μια εφαρμογή μιας σελίδας είναι δύσκολο να βρεθούν με ευρετήριο. Συνήθως, οι προγραμματιστές πρέπει να καταβάλουν επιπλέον προσπάθειες για τη βελτιστοποίηση SEO εφαρμογών ιστού που έχουν δημιουργηθεί με την Angular. Η Angular Universal και το Prerender.io είναι κοινές λύσεις.

3)Δυσκολίες με συμβατότητα προς τα πίσω

Οι προγραμματιστές δεν μπορούν να αλλάξουν απευθείας από AngularJS σε Angular. Υπάρχει μια ολόκληρη ενότητα στην γωνιακή τεκμηρίωση που απαριθμεί όλους τους πιθανούς τρόπους αντιμετώπισης της μετανάστευσης.

Οι παλαιότερες εκδόσεις του Angular είναι απόλυτα συμβατές προς τα πίσω. Για παράδειγμα, δεν θα υπάρχει καμία δυσκολία μετεγκατάστασης από την πέμπτη στην τέταρτη έκδοση.

4)Ελαφροί ιστότοποι με στατικό περιεχόμενο

Η Angular είναι ιδανική για ιστότοπους με δυναμικό περιεχόμενο, αλλά δεν επαρκεί κατά τη δημιουργία μικρών σελίδων με στατικό περιεχόμενο. Η εφαρμογή Angular σε αυτήν την περίπτωση αυξάνει μόνο το συνολικό μέγεθος του έργου και ως εκ τούτου αυξάνει σημαντικά το χρόνο φόρτωσης. Αυτός είναι ο κύριος λόγος για τον οποίο δεν σας

προτείνεται η χρήση της Angular για σελίδες προορισμού και ιστότοπους με στατικό περιεχόμενο.

5)Βελτιστοποιημένοι ιστότοποι SEO

Η Angular απέχει πολύ από το να είναι φιλική προς το SEO, οπότε θα πρέπει να αποφύγετε τη χρήση της Angular όταν ο στόχος είναι να φτάσετε στην κορυφή των αποτελεσμάτων αναζήτησης. Διαφορετικά, πρέπει να είσαστε έτοιμοι να κάνετε επιπλέον προσπάθειες για να βελτιστοποιήσετε την εφαρμογή ιστού για μηχανές αναζήτησης.

6)Βραχυπρόθεσμα έργα

Η Angular δεν είναι μια καλή λύση για βραχυπρόθεσμα έργα ή επιχειρήσεις με περιορισμένους πόρους. Επίσης, δεν είναι καλή ιδέα για έργα με λίγους προγραμματιστές, καθώς η πολυπλοκότητα του πλαισίου front-end μπορεί να είναι δύσκολο να ξεπεραστεί.

7)Εφαρμογές με προσέγγιση σχεδίασης μικροϋπηρεσιών

Η Angular δεν είναι κατάλληλη για εφαρμογές με αρχιτεκτονική microservice στο frontend. Εάν θέλετε να είστε πιο ευέλικτοι όσον αφορά την επιλογή πρόσθετων εργαλείων, μην επιλέξετε αυτό το εργαλείο. Η Angular είναι μια ολοκληρωμένη λύση - δεν σας δίνει πλήρη έλεγχο του μεγέθους της εφαρμογής, επιτρέποντάς σας να επιλέξετε μόνο τα πράγματα που πραγματικά χρειάζεστε.

REACT

Το React.js είναι μια βιβλιοθήκη JavaScript ανοιχτού κώδικα όπου αναπτύχθηκε και συντηρήθηκε από το Facebook και συγκεκριμένα από τον Jordan Walke, έναν μηχανικό λογισμικού που εργάζεται για το Facebook. Η React λοιπόν χρησιμοποιείται για τη δημιουργία διεπαφών χρήστη ειδικά για εφαρμογές μιας σελίδας. Επίσης χρησιμοποιείται για το χειρισμό του επιπέδου προβολής για εφαρμογές ιστού και για κινητά. Η React μας επιτρέπει ακόμα να δημιουργήσουμε επαναχρησιμοποιήσιμα στοιχεία διεπαφής χρήστη. Η React αναπτύχθηκε για πρώτη φορά στο newsfeed του Facebook το 2011 και στο Instagram.com το 2012. Οπώς καταλαβαίνουμε λοιπόν πολλές εφαρμογές που χρησιμοποιούμε καθημερινά βασίζονται στην React οπότε δεν πρέπει να την περνάμε αψήφιστα.

Η React επιτρέπει στους προγραμματιστές να δημιουργούν μεγάλες εφαρμογές ιστού που μπορούν να αλλάζουν δεδομένα, χωρίς να επαναφορτώνουν τη σελίδα. Ο κύριος σκοπός της React είναι να είναι γρήγορος, επεκτάσιμος και απλός και λειτουργεί μόνο σε διεπαφές χρήστη στην εφαρμογή. Η React παρέχει μόνο την προβολή στο MVC δηλαδή πρέπει να λύσετε μόνοι σας τα M και C. Λόγω αυτού, μπορείτε να επιλέξετε οποιαδήποτε από τις δικές σας βιβλιοθήκες ανάλογα με τις ανάγκες σας, θα καταλήξετε όμως να χρησιμοποιείτε πολλές ανεξάρτητες και γρήγορες βιβλιοθήκες. Εξαιτίας αυτού, θα πρέπει να φροντίσετε μόνοι σας για τις αντίστοιχες ενημερώσεις και μετεγκαταστάσεις τους. Επιπλέον, κάθε έργο React είναι διαφορετικό και απαιτεί απόφαση που απαιτεί την ιεραρχία και την αρχιτεκτονική του φακέλου και για αυτόν τον λόγο τα πράγματα μπορούν να πάνε στραβά πολύ πιο εύκολα λόγω αυτού.

Η React παρέχει τα εξής:

- Αντί για κλασικά πρότυπα έχει JSX μια γλώσσα που μοιάζει με XML η οποία είναι χτισμένη πάνω από το JavaScript
- Προστασία XSS
- Ανάκτηση αιτημάτων Ajax
- Βοηθητικά προγράμματα για unit testing

Ορισμένες δημοφιλείς βιβλιοθήκες για προσθήκη λειτουργικότητας είναι:

- React-router για δρομολόγηση
- Redux ή MobX για διαχείριση κατάστασης
- Ένζυμο για πρόσθετα βοηθητικά προγράμματα δοκιμών

Γιατί χρησιμοποιούμε REACT:

1)Καμπύλη γρήγορης μάθησης

Η React είναι μια απλή και ελαφριά βιβλιοθήκη που ασχολείται μόνο με το επίπεδο προβολής . Δεν είναι ένα θηρίο όπως άλλα MV * πλαίσια όπως η Angular ή η Ember. Οποιοσδήποτε προγραμματιστής Javascript μπορεί να καταλάβει τα βασικά και να αρχίσει να αναπτύσσει μια φοβερή εφαρμογή Ιστού μετά από λίγες μέρες ανάγνωσης κάποιου σεμιναρίου. Όπως λέει ο οδηγός React, το "Thinking in React" μπορεί να είναι λίγο διαφορετικό από το συνηθισμένο, καθώς φέρνει μια νέα προσέγγιση στο τραπέζι, αλλά θα γίνει πολύ πιο εύκολο και φυσικό καθώς αποκτάτε εμπειρία με αυτό.

2)Γρήγορη απόδοση με εικονικό DOM

Όταν πρόκειται να αναπτύξετε μια εφαρμογή ιστού που περιλαμβάνει υψηλή αλληλεπίδραση χρήστη και προβολή ενημερώσεων, όπως το νέο πρόγραμμα δημιουργίας φορμών στο JotForm 4.0, πρέπει να λάβετε υπόψη τα πιθανά ζητήματα απόδοσης. Αν και οι σημερινές μηχανές javascript είναι αρκετά γρήγορες για να χειριστούν τόσο περίπλοκες εφαρμογές, οι χειρισμοί DOM δεν είναι ακόμα τόσο γρήγοροι. **Η ενημέρωση του DOM είναι συνήθως η δυσχέρεια όσον αφορά την απόδοση του διαδικτύου** . Η React προσπαθεί να λύσει αυτό το πρόβλημα χρησιμοποιώντας κάτι που ονομάζεται **εικονικό DOM**; ένα DOM διατηρείται στη μνήμη. Οποιοσδήποτε αλλαγές προβολής αντικατοπτρίζονται πρώτα στο εικονικό DOM και, στη συνέχεια, ένας αποτελεσματικός αλγόριθμος διαφορών συγκρίνει τις προηγούμενες και τις τρέχουσες καταστάσεις του εικονικού DOM και υπολογίζει τον καλύτερο τρόπο (ελάχιστος απαιτούμενος αριθμός ενημερώσεων) για την εφαρμογή αυτών των αλλαγών. Τέλος, αυτές οι ενημερώσεις εφαρμόζονται στο DOM για να διασφαλιστεί ο ελάχιστος χρόνος ανάγνωσης / εγγραφής. Αυτός είναι ο κύριος λόγος πίσω από την υψηλή απόδοση της React.

3)Καθαρή αφαίρεση

Μία από τις ισχυρές πλευρές του React είναι ότι παρέχει μια καλή αφαίρεση που σημαίνει ότι δεν εκθέτει περίπλοκα εσωτερικά στον χρήστη. Η React δεν υπαγορεύει μοτίβα ή αρχιτεκτονική όπως το MVC / MVVM, μετά από όλα μόνο για το επίπεδο προβολής και είστε ελεύθεροι να σχεδιάσετε την αρχιτεκτονική της εφαρμογής σας με όποιον τρόπο θέλετε. Ωστόσο, υπάρχει μια καλή αρχιτεκτονική που ταιριάζει στη React που ονομάζεται Flux .

4)Αρχιτεκτονική Flux και Redux

Η αρχιτεκτονική Flux αναπτύχθηκε από το Facebook για τις διάφορες διαδικτυακές του εφαρμογές. Είναι παρόμοιο με τα στοιχεία React από τη μονόδρομη ροή του. Αυτή η αρχιτεκτονική έχει δημιουργούς δράσης που βοηθούν στη δημιουργία δράσης από παραμέτρους μεθόδου. Διατηρεί επίσης μια βιβλιοθήκη για αυτές τις μεθόδους. Όλες αυτές οι ενέργειες διατηρούνται μαζί από έναν κεντρικό διεκπεραιωτή για την ενημέρωση των αποθηκών. Όλες οι προβολές ενημερώνονται σύμφωνα με τις αποθήκες. Υπάρχει επίσης το Redux που είναι μια βελτιωμένη έκδοση της αρχιτεκτονικής ροής. Διαθέτει μια μοναδική αποθήκη που δεν υπάρχει σε ροή. Το Redux επιτρέπει επίσης μια δυνατότητα όπου το μεσαίο λογισμικό μπορεί να οριστεί ώστε να παρακολουθεί τις ενέργειες που αποστέλλονται.

5)Η δημιουργία δυναμικών εφαρμογών Ιστού είναι ευκολότερη

Νωρίτερα η δημιουργία μιας δυναμικής ιστοσελίδας απαιτούσε πολύ περίπλοκη κωδικοποίηση. Χρειαζόταν συγκεκριμένες συμβολοσειρές HTML για δυναμικές ιστοσελίδες. Η χρήση της ReactJS δεν αποτελεί πλέον πρόβλημα. Χρησιμοποιεί το JSX το οποίο έχει μια συγκεκριμένη σύνταξη και επιτρέπει σε HTML εισαγωγικά και ετικέτες να αποδίδουν συγκεκριμένα εξαρτήματα. Εκτός από αυτό, υποστηρίζει επίσης την κατασκευή κώδικα αναγνώσιμη από μηχανή και αναμινύει στοιχεία σε ένα μεμονωμένο μεταβλητό αρχείο. Η React το κανονικό JavaScript δεν χρησιμοποιείται για τη δημιουργία των προτύπων του.

6)Αυξάνει την παραγωγικότητα και βοηθά επίσης στη συντήρηση

Η επαναχρησιμοποίηση στοιχείων είναι το κύριο πλεονέκτημα της React JS. Ακόμη και το Facebook έχει εφαρμόσει τη React καθώς διευκολύνει την επαναχρησιμοποίηση των στοιχείων του συστήματος. Ένας προγραμματιστής μπορεί να ξεκινήσει με τα συνήθη στοιχεία όπως το πλαίσιο ελέγχου, το κουμπί κ.λπ. Μόλις γίνει αυτό, μπορείτε να μετακινηθείτε σε στοιχεία περιτυλίγματος που αποτελούνται από εσωτερική λογική που διευκολύνει τον χειρισμό και τον ορισμό των αντικειμένων που χρησιμοποιούνται. Αυτό διασφαλίζει τη συνοχή της εφαρμογής και διευκολύνει τη συντήρηση κώδικα. Πολλές εταιρείες όπως η Walmart τείνουν να χρησιμοποιούν το ReactJS. Καθώς όλες οι εταιρείες έχουν κοινά στοιχεία όπως στοιχεία φόρμας πληρωμής, καρουσέλ εικόνας, για παράδειγμα. Εδώ έρχεται στην εικόνα η επαναχρησιμοποίηση των στοιχείων. Επίσης, για την ενεργοποίηση της διαχείρισης εκδόσεων και την περαιτέρω εγκατάσταση στοιχείων, υπάρχουν πρότυπα κωδικοποίησης που ορίζονται για τη διατήρηση της ομοιομορφίας. Αυτό βελτιώνει επίσης την ταχύτητα της εφαρμογής, η οποία απολαμβάνει περισσότερη διατήρηση πελατών.

7)Εξασφαλίζει ταχύτερη απόδοση του εικονικού DOM

Η απόδοση οποιασδήποτε εφαρμογής εξαρτάται από τη δομή στην οποία βασίζεται. Για την κατασκευή μιας εφαρμογής υψηλού φορτίου, αυτός είναι ένας από τους σημαντικότερους παράγοντες που πρέπει να ληφθούν υπόψη. Αλλά οι κινητήρες που χρησιμοποιούνται αυτές τις μέρες δεν μπορούν να διασφαλίσουν σημεία συμφόρησης επειδή το DOM (μοντέλο αντικειμένου εγγράφου) δεν έχει δομή δέντρων και ακόμη και μικρά στρώματα μπορούν να κάνουν δραστικές αλλαγές. Για την επίλυση αυτού του εικονικού DOM εισήχθη. Το ReactJS μπορεί να χρησιμοποιηθεί για βαριές και δυναμικές λύσεις. Είναι μια εικονική αναπαράσταση στο μοντέλο αντικειμένου εγγράφου και ελέγχει εάν εφαρμόζονται όλες οι αλλαγές στο εικονικό DOM. Το πραγματικό δέντρο DOM ενημερώνεται αργότερα και διασφαλίζει επίσης ότι καταναλώνεται ελάχιστος χρόνος για το σκοπό αυτό. Η χρήση αυτής της μεθόδου εγγυάται καλύτερη εμπειρία και απόδοση του χρήστη.

8)Διασφαλίζει την ύπαρξη σταθερού κώδικα

Η ReactJS χρησιμοποιεί ροή δεδομένων προς τα κάτω. Αυτό συμβαίνει επειδή διασφαλίζει ότι οι μικρότερες αλλαγές που συμβαίνουν στη δομή του παιδιού δεν επηρεάζουν τους γονείς. Όταν ένα αντικείμενο αλλάζει από έναν προγραμματιστή, πρέπει να διασφαλιστεί ότι ενημερώνονται μόνο τα συγκεκριμένα αντικείμενα και ότι τροποποιείται μόνο η κατάστασή του. Τα δεδομένα συνδέονται μαζί με το αντικείμενο και αυτή η δομή φροντίζει για τη διατήρηση της σταθερότητας του κώδικα και της απόδοσης της εφαρμογής.

9)Έχει χαρακτηριστικά SEO Friendly

Όποτε υπάρχει μια κοινή αποτυχία της μηχανής αναζήτησης να διαβάσει το JavaScript εκείνη τη στιγμή, η ReactJS εκτελείται σε διακομιστή, αποδίδει και επιστρέφει εικονικό DOM στο πρόγραμμα περιήγησης μέσω μιας κανονικής ιστοσελίδας. Οι χρήσεις της ReactJs έχουν αυτή την ικανότητα να αντιμετωπίζουν αποτελεσματικά τις αποτυχίες. Με πολλές βελτιώσεις που πραγματοποιούνται καθημερινά στο Google, δεν μπορούμε να ξεχάσουμε άλλες μηχανές αναζήτησης όπως το Yahoo, το Bing κ.λπ.

10)Ένα χρήσιμο σύνολο εργαλείων προγραμματιστή

Μια άλλη σημαντική χρήση της React JS είναι μια φιλική προς το χρήστη πλατφόρμα ανάπτυξης. Τα δύο κύρια εργαλεία που παρέχονται είναι το React to Developer Tools και το Redux Developer Tools . Και τα δύο αυτά εργαλεία μπορούν να εγκατασταθούν ως επεκτάσεις Chrome. Το εργαλείο React developer χρησιμοποιείται για την επιθεώρηση στοιχείων αντίδρασης που βρίσκονται σε μια ιεραρχία και χρησιμοποιούνται για την

ενδοσκόπηση των ιδιοτήτων και των καταστάσεων. Βοηθά επίσης στην προβολή ιεραρχιών στοιχείων, στην ανακάλυψη εξαρτημάτων παιδιών και γονέων. Όταν χρησιμοποιείτε το Redux μπορείτε να παρατηρήσετε διάφορες ενέργειες που αποστέλλονται, τρέχουσες καταστάσεις καταστήματος και να ελέγξετε τις αλλαγές στα καταστήματα. Οι διάφορες ενέργειες αποστολής ή η τροποποίηση αποθηκών με διαφορετικές αλλαγές μπορούν επίσης να προβληθούν και να προβληθούν άμεσα. Ένας χρήστης μπορεί επίσης να εγγράψει και να επιστρέψει σε προηγούμενες καταστάσεις της εφαρμογής για σκοπούς εντοπισμού σφαλμάτων.

11) Το React Native είναι ιδανικό για ανάπτυξη εφαρμογών για κινητά

Η React παρέχει παρόμοια μοτίβα σχεδιασμού για να διευκολύνει τη διαδικασία μετάβασης μεταξύ ανάπτυξης εφαρμογών ιστού και κινητών. Χρησιμοποιώντας το JavaScript και τη React θα μπορείτε να δημιουργήσετε εφαρμογές UI προηγμένης τεχνολογίας που υποστηρίζονται από συστήματα Android και iOS.

12) Εύκολο στην εκμάθηση

Το ReactJS δεν είναι ένα πλαίσιο αλλά μια βιβλιοθήκη που συγκεντρώνει αντικείμενα και εστιάζει σε ένα συγκεκριμένο πράγμα για να το κάνει σωστά. Είναι ο ελεγκτής προβολής στην αρχιτεκτονική MVC. Κάθε προγραμματιστής που γνωρίζει JavaScript μπορεί εύκολα να κατανοήσει τη React, να μάθει τα βασικά του και να δημιουργήσει μια εφαρμογή.

Γιατί δεν χρησιμοποιούμε REACT:

1) Κακή τεκμηρίωση

Η ReactJS είναι μια σχετικά νέα τεχνολογία και συνεχίζει να αναπτύσσεται με τη βοήθεια προγραμματιστών που συμβάλλουν σε αυτό και διευρύνουν τις δυνατότητές του.

Ωστόσο, συχνά λείπει η τεκμηρίωση για τη ReactJS και τις σχετικές βιβλιοθήκες του. Οι προγραμματιστές τείνουν να κυκλοφορούν ενημερώσεις σε βιβλιοθήκες χωρίς να ενημερώνουν την τεκμηρίωση. Αυτό δυσκολεύει τους άλλους να εξοικειωθούν με νέες δυνατότητες και να ενσωματώσουν βιβλιοθήκες.

2) Σύγχυση JSX

Η ReactJS χρησιμοποιεί την επέκταση σύνταξης JSX για τη δημιουργία αντικειμένων JavaScript με σύνταξη HTML. Αυτό απλοποιεί τη διαδικασία τροποποίησης DOM και καθιστά τον κώδικα πιο ευανάγνωστο. Η σύνταξη HTML σε κώδικα JavaScript μπορεί να

φαίνεται συγκεχυμένη για τους νέους προγραμματιστές, οπότε θα χρειαστούν χρόνο για να το συνηθίσουν. Οι νέοι προγραμματιστές συχνά βρίσκουν ότι το JSX είναι μειονέκτημα του ReactJS.

3) Περίπλοκη βελτιστοποίηση μηχανών αναζήτησης

Η βελτιστοποίηση μηχανών αναζήτησης (SEO) είναι ζωτικής σημασίας για μια εφαρμογή Ιστού για την επισκεψιμότητα και την προσέλκυση νέων πελατών. Για να καταλάβουμε γιατί είναι δύσκολο να κάνουμε την εφαρμογή ReactJS φιλική για SEO, πρέπει να καταλάβουμε πώς η Google ευρετηριάζει ιστοσελίδες.

Η Google χρησιμοποιεί bots για ευρετηρίαση ιστοσελίδων. Αυτά τα bots σαρώνουν το περιεχόμενο μιας ιστοσελίδας και αποθηκεύουν πληροφορίες σχετικά με αυτό στο ευρετήριο Google. Όταν ένας χρήστης ζητά μια συγκεκριμένη πληροφορία, η Google ελέγχει τα δεδομένα που είναι αποθηκευμένα στο ευρετήριο Google για να παρέχει τις πιο σχετικές πηγές πληροφοριών.

Τα Google bots μπορούν εύκολα να ευρετηριάσουν σελίδες HTML. Ωστόσο, δεν λειτουργεί τόσο ομαλά με σελίδες JavaScript. Όλες οι δυναμικές εφαρμογές Ιστού JavaScript, συμπεριλαμβανομένων εκείνων που έχουν δημιουργηθεί με το ReactJS, πρέπει να περάσουν από μια πιο περίπλοκη διαδικασία ευρετηρίασης από τις στατικές ιστοσελίδες. Έτσι, μια σελίδα ReactJS ενδέχεται να μην ευρετηριαστεί σωστά ή η ευρετηρίαση μπορεί να πάρει πολύ χρόνο αναγκάζοντας τα bots Google να εγκαταλείψουν τη σελίδα.

Υπάρχουν διάφορες προσεγγίσεις που μπορείτε να χρησιμοποιήσετε για να κάνετε τον ιστότοπό σας ReactJS φιλικός προς το SEO. Για παράδειγμα, μπορείτε να σκεφτείτε την απόδοση πριν από την απόδοση ή από την πλευρά του διακομιστή.

4) Στενή εστίαση στο περιβάλλον χρήστη

Το ReactJS είναι μια βιβλιοθήκη JavaScript που περιέχει μια συλλογή εργαλείων για τη δημιουργία διεπαφών εφαρμογών Ιστού. Αλλά σε αντίθεση με τα πλαίσια, το ReactJS δεν είναι ένα εργαλείο all-in-one για τη δημιουργία μιας ολόκληρης εφαρμογής. Επομένως, εάν χρησιμοποιείτε ένα μοτίβο σχεδίασης μοντέλου-προβολέα (MVC) για την εφαρμογή Ιστού σας, το ReactJS θα είναι υπεύθυνο μόνο για την προβολή. Τα άλλα δύο μέρη - το μοντέλο και ο ελεγκτής - πρέπει να δημιουργηθούν με τη βοήθεια πρόσθετων εργαλείων. Εάν χρησιμοποιείτε το ReactJS, θα πρέπει να ενσωματώσετε πρόσθετα εργαλεία για δρομολόγηση, διεπαφές προγραμματισμού εφαρμογών (API) και άλλα μέρη της εφαρμογής σας.

Καθώς ορισμένα πλαίσια JavaScript είναι καθολικές λύσεις και επιτρέπουν τη δημιουργία όλων των στοιχείων του μοτίβου MVC, ορισμένοι προγραμματιστές θεωρούν ότι η στενή εστίαση της React στο UI είναι μειονέκτημα.

Πλεονεκτήματα:

<u>Πλεονεκτήματα Angular</u>	<u>Πλεονεκτήματα React</u>
<ul style="list-style-type: none">• Γρήγορη παράδοση και πρωτότυπος στυλ ταμπλό επιτρέποντας έτσι τη δημιουργία δυναμικών εφαρμογών μίας σελίδας.	<ul style="list-style-type: none">• Η διαδικασία ενημερώσεων βελτιστοποιείται και επιταχύνεται.
<ul style="list-style-type: none">• Ταχύτερη και απλούστερη ανάπτυξη σε σύγκριση με τη React, το Backbone και το Knockout.	<ul style="list-style-type: none">• Το JSX καθιστά αναγνώσιμα στοιχεία / μπλοκ κώδικα. Εμφανίζει πώς συνδέονται ή συνδυάζονται στοιχεία.
<ul style="list-style-type: none">• Απαιτεί ελάχιστη κωδικοποίηση	<ul style="list-style-type: none">• Η δέσμευση δεδομένων της React καθορίζει συνθήκες για τη δημιουργία δυναμικών εφαρμογών.
<ul style="list-style-type: none">• Ομαλή δοκιμή	<ul style="list-style-type: none">• Γρήγορη απόδοση. Η χρήση περιλαμβάνει μεθόδους για την ελαχιστοποίηση του αριθμού των λειτουργιών DOM βοηθά στη βελτιστοποίηση της διαδικασίας ενημέρωσης και την επιτάχυνσή της.
<ul style="list-style-type: none">• Ιδανικό για εφαρμογές με διαδραστική κωδικοποίηση από τον πελάτη	<ul style="list-style-type: none">• Τα εγγενή εργαλεία της React προσφέρονται για δοκιμή, εντοπισμό σφαλμάτων κώδικα.
<ul style="list-style-type: none">• Συστήματα έγχυσης εξάρτησης	<ul style="list-style-type: none">• Η ομάδα του Facebook υποστηρίζει τη βιβλιοθήκη. Συμβουλές ή δείγματα κώδικα μπορούν να δοθούν από την κοινότητα του Facebook.
<ul style="list-style-type: none">• Διπλές δυνατότητες δέσμευσης δεδομένων	<ul style="list-style-type: none">• Χρησιμοποιώντας το React + ES6 / 7, η εφαρμογή αποκτά υψηλή τεχνολογία και είναι κατάλληλη για συστήματα υψηλού φορτίου.
<ul style="list-style-type: none">• Επεκτείνει και εκτείνεται κωδικοποίηση HTML	<ul style="list-style-type: none">• Φιλικό προς το SEO. Το React παρουσιάζει την εμπειρία πρώτης φόρτωσης από την απόδοση του διακομιστή και συνδέοντας τους χειριστές συμβάντων στο πλάι του

	<p>χρήστη:</p> <ol style="list-style-type: none"> 1. διακομιστής <code>React.renderComponentToString</code> καλείται στο διακομιστή. 2. Το <code>React.renderComponent ()</code> καλείται από την πλευρά του πελάτη. 3. Το <code>React</code> διατηρεί σήμανση που παρέχεται από την πλευρά του διακομιστή, επισυνάπτει χειριστές συμβάντων.
<ul style="list-style-type: none"> • Πρωτότυπο γρήγορης και χωρίς προβλήματα 	
<ul style="list-style-type: none"> • Η Angular υποστηρίζεται από την Google, η οποία δημιουργεί ένα αξιόπιστο, αξιόπιστο πρόγραμμα που είναι πιθανό να συμβαδίζει με τις περιστασιακές ενημερώσεις και ανακοινώσεις της Google. Συνδυάστε το με μια πλούσια λεπτομερή τεκμηρίωση και έχετε ένα αξιόπιστο πλαίσιο που υποστηρίζεται από μια μεγάλη ποσότητα χρήσιμων πληροφοριών και απαντήσεων σε κοινές ερωτήσεις. 	
<ul style="list-style-type: none"> • Η Angular παρέχει μια μεγάλη ποικιλία ενσωματώσεων τρίτων που μπορούν να προστεθούν στο πλαίσιο με ευκολία. Αυτό δίνει στους προγραμματιστές ακόμη περισσότερα εργαλεία για τη βελτίωση της συνολικής μορφής και λειτουργίας του προϊόντος τους. 	
<ul style="list-style-type: none"> • Η Angular προσφέρει γρηγορότερο χρόνο φόρτωσης και αυξημένη ασφάλεια χρησιμοποιώντας μια λαμπρή ιδέα κοινώς γνωστή ως "μεταγλωττιστής πριν από το χρόνο". Η 	

<p>γωνιακή μεταγλώττιση HTML και TypeScript σε JavaScript κατά τη διάρκεια της ανάπτυξης, πράγμα που σημαίνει ότι ολόκληρος ο κώδικας καταρτίζεται πριν το πρόγραμμα περιήγησης φορτώσει ακόμη και εφαρμογή ιστού.</p>	
<ul style="list-style-type: none"> • Οι δυνατότητες έγχυσης εξάρτησης του γωνιακού πλαισίου μπορεί να είναι ένα τεράστιο πλεονέκτημα - αν ξέρετε πώς να το χρησιμοποιήσετε. Η δημιουργία των διαφόρων εξαρτήσεων μπορεί να είναι χρονοβόρα, αλλά αξίζει τον κόπο αν εργάζεστε σε ένα έργο μεγάλης κλίμακας που αναμένεται να εκτελέσει μια σειρά διαφορετικών λειτουργιών. 	
<ul style="list-style-type: none"> • Με γωνιακά στοιχεία και ενότητες, αυτό το πλαίσιο έχει σχεδιαστεί ώστε να είναι πλήρως προσαρμόσιμο, δίνοντας περισσότερη δύναμη στον προγραμματιστή και τον σχεδιαστή. Γωνιακά στοιχεία μπορούν επίσης εύκολα να προστεθούν σε έργα που δημιουργήθηκαν χρησιμοποιώντας ένα άλλο πλαίσιο, το οποίο απλώς προσθέτει την ελκυστικότητα αυτού του προγράμματος. 	

Μειονεκτήματα:

<u>Μειονεκτήματα της Angular</u>	<u>Μειονεκτήματα React:</u>
<ul style="list-style-type: none">Είναι πολύ εύκολο να χρησιμοποιείς τα πεδία αλλά είναι δύσκολο να τα εντοπίσουμε.	<ul style="list-style-type: none">Καμπύλη εκμάθησης. Δεδομένου ότι δεν είναι το πλήρες χαρακτηριστικό πλαίσιο, απαιτείται βαθιά γνώση για ενσωμάτωση δωρεάν βιβλιοθήκης διεπαφής χρήστη στο πλαίσιο MVC.
<ul style="list-style-type: none">Η τεκμηρίωση χρειάζεται βελτίωση.	<ul style="list-style-type: none">Ο προσανατολισμός προβολής είναι ένα από τα μειονεκτήματα της ReactJS. Θα πρέπει να βρεθεί «Μοντέλο» και «Ελεγκτής» για την επίλυση του προβλήματος «Προβολή».
<ul style="list-style-type: none">Παρά το ότι είναι δυναμικές και ισχυρές, είναι πολύ δύσκολο να χρησιμοποιηθούν οι οδηγίες.	<ul style="list-style-type: none">Η μη χρήση ισόμορφης προσέγγισης για την εκμετάλλευση της εφαρμογής οδηγεί σε προβλήματα ευρετηρίασης των μηχανών αναζήτησης.
<ul style="list-style-type: none">Μετά το Bootstrap, οι προγραμματιστές θα δυσκολευτούν να διαμορφώσουν το πλαίσιο.	<ul style="list-style-type: none">Πολλοί προγραμματιστές αντιπαθούν την τεκμηρίωση του JSX React, τα εγχειρίδια είναι δύσκολο για την κατανόηση των νεοεισερχόμενων.
<ul style="list-style-type: none">Περιορισμένες συνδέσεις δρομολογητή.	<ul style="list-style-type: none">Η βιβλιοθήκη μεγάλου μεγέθους του React.
<ul style="list-style-type: none">Υπάρχουν επίσης προβλήματα στην ευρετηρίαση των μηχανών αναζήτησης.	
<ul style="list-style-type: none">Ο Internet Explorer 8.0 δεν υποστηρίζει AngularJs.	
<ul style="list-style-type: none">Δεν υπάρχει εξουσιοδότηση και έλεγχος ταυτότητας διακομιστή σε	

<p>γωνιακό. Εξουσιοδότηση σημαίνει χορήγηση άδειας για πρόσβαση στα δεδομένα και αναγνώριση ενός χρήστη με επικύρωση διαπιστευτηρίων. Η AngularJs δεν μπορεί να παρέχει και τις δύο δυνατότητες, αυτό το λέμε λιγότερο ασφαλές.</p>	
<ul style="list-style-type: none"> • Τα AngularJs, εξαρτώνται πλήρως από τη JavaScript. Εάν αυτά τα σενάρια θα κρυφτούν, οι σελίδες που προκύπτουν μοιάζουν με μια βασική σελίδα. 	
<ul style="list-style-type: none"> • Είναι πολύ απέραντο και περίπλοκο. Σε γωνιακό, υπάρχουν πολλοί τρόποι εκτέλεσης της ίδιας εργασίας, επομένως είναι δύσκολο να προβλεφθεί ποιος είναι ο πιο βελτιστοποιημένος τρόπος εκτέλεσης μιας συγκεκριμένης εργασίας. 	

ΜΕΡΟΣ Β

Single-Page Application:

Μια εφαρμογή single page αποτελείται από ανεξάρτητα κομμάτια τα οποία μπορούν να αντικατασταθούν ή να ανανεωθούν ανεξάρτητα, χωρίς να ανανεωθεί ολόκληρη η σελίδα. Με αυτόν τον τρόπο υπάρχει εξοικονόμηση του bandwidth, αλλά δεν χρειάζεται επίσης να φορτώνονται συνέχεια εξωτερικά αρχεία κάθε φορά που ανανεώνεται η σελίδα, όπως πχ οι εικόνες τα αρχεία css κλπ. Ο σκοπός δηλαδή είναι να φτιάξουμε τις σελίδες να φορτώνουν πολύ γρήγορα σε σχέση με τις παραδοσιακές σελίδες οι οποίες κάθε φορά που γινόταν κάτι έπρεπε να φορτωθούν από τον σέρβερ. Οι single page applications είναι κατά κύριο λόγο γραμμένες με JavaScript, HTML5, AJAX. Οι περισσότεροι προγραμματιστές τους αρέσει να προγραμματίζουν με αυτόν τον τρόπο γιατί ορισμένα frameworks όπως η AngularJS η οποία είναι χτισμένη πάνω από την JavaScript τους βοηθάνε πολύ στο να στήσουν web application.

Στο παρελθόν στις σελίδες που στέλνóταν ένα αίτημα για να δούμε μια πληροφορία έπρεπε να φορτωθεί ολόκληρη. Για παράδειγμα αν είμασταν σε μια σελίδα που χρησιμοποιεί background εικόνα ,navigation bars ,logos κλπ θα πρέπει σε κάθε ανανέωση που γίνεται να στέλνουμε αίτημα στον σέρβερ και ο σέρβερ να μας στέλνει όλη τη σελίδα από την αρχή συν την καινούργια πληροφορία που ζητήθηκε. Αυτό αυξάνει κατακόρυφα το χρόνο φόρτωσης της σελίδας. Ενώ μια single page εφαρμογή στέλνει μόνο όσα αρχεία έχουν αλλάξει κάνοντας την πολύ γρήγορη το φόρτωμα της σελίδας. Στις περισσότερες σελίδες υπάρχει πολύ επαναλαμβανόμενο περιεχόμενο, όπως είναι το footer ,header ,background, navigation bar κλπ, οπότε αυτά δεν χρειάζονται να φορτωθούν εκ νέου παραμόνο οι καινούργιες πληροφορίες που ζητήθηκαν.

Γιατί χρησιμοποιούμε Single-Page Application:

1)Γρήγορη και γρήγορη ανταπόκριση

Δεδομένου ότι οι εφαρμογές μιας σελίδας δεν ενημερώνουν ολόκληρη τη σελίδα αλλά απαιτούν μόνο περιεχόμενο, βελτιώνουν σημαντικά την ταχύτητα ενός ιστότοπου. Οι περισσότεροι πόροι (HTML / CSS / Scripts) φορτώνονται μόνο μία φορά καθ 'όλη τη διάρκεια ζωής μιας εφαρμογής. Μόνο δεδομένα μεταδίδονται μπρος-πίσω. Αυτό είναι ένα μεγάλο πλεονέκτημα, και σύμφωνα με την έρευνα της Google , εάν η φόρτωση μιας σελίδας υπερβαίνει τα 200 χιλιοστά του δευτερολέπτου, μπορεί να έχει δυνητικά υψηλό αντίκτυπο στις επιχειρήσεις και τις πωλήσεις.

2)Δυνατότητες προσωρινής αποθήκευσης

Μια εφαρμογή μιας σελίδας μπορεί να αποθηκεύσει προσωρινά τα τοπικά δεδομένα. Ένα SPA στέλνει μόνο ένα αίτημα σε έναν διακομιστή και στη συνέχεια αποθηκεύει όλα τα δεδομένα που λαμβάνει. Στη συνέχεια, μπορεί να χρησιμοποιήσει αυτά τα δεδομένα και να λειτουργήσει ακόμη και εκτός σύνδεσης. Εάν ένας χρήστης έχει κακή συνδεσιμότητα, τα τοπικά δεδομένα μπορούν να συγχρονιστούν με το διακομιστή όταν το επιτρέπει η σύνδεση.

3)Γραμμική εμπειρία χρήστη

Τα SPA παρέχουν στους χρήστες μια απλή γραμμική εμπειρία. Αυτές οι εφαρμογές ιστού - όπως το Saucy , για παράδειγμα - περιέχουν μια σαφή αρχή, μέση και τέλος. Η εφαρμογή web Saucy παρέχει μια εξαιρετική διαδραστική εμπειρία χρησιμοποιώντας παράλληλη κύλιση και εκπληκτικές μεταβάσεις και εφέ για να παρουσιάσει το πλήρες ταξίδι των πελατών. Με τα SPA, η κύλιση είναι βολική και χωρίς διακοπές. Δεν χρειάζεται να κάνετε κλικ σε ατελείωτους συνδέσμους. Πράγματι, η κυλιόμενη φύση των SPA τα καθιστά ιδανικά για χρήστες κινητών που έχουν συνηθίσει την κύλιση.

4)Εντοπισμός σφαλμάτων με Chrome

Είναι εύκολο να κάνετε εντοπισμό σφαλμάτων SPA με το Chrome, καθώς τέτοιες εφαρμογές αναπτύσσονται σε πλαίσια όπως τα εργαλεία προγραμματιστών AngularJS Batarang και React . Αυτά τα πλαίσια έχουν τα δικά τους εργαλεία προγραμματιστή Chrome που κάνουν τον εντοπισμό σφαλμάτων πολύ πιο εύκολο από ό, τι με τα MPA. Επιπλέον, τα SPA σας επιτρέπουν να παρακολουθείτε τις λειτουργίες του δικτύου και να διερευνείτε στοιχεία σελίδας και δεδομένα που σχετίζονται με αυτά. Αλλά με όλα τα οφέλη τους, οι ΖΕΠ έχουν επίσης κάποια μειονεκτήματα που πρέπει να λάβετε υπόψη. Ας ρίξουμε μια ματιά σε αυτά.

5)Ευκολία στη δημιουργία εφαρμογών πλούσιων σε χαρακτηριστικά

Η εφαρμογή SPA διευκολύνει την προσθήκη προηγμένων λειτουργιών σε μια εφαρμογή ιστού. Για παράδειγμα, είναι πιο εύκολο να δημιουργήσετε μια εφαρμογή ιστού επεξεργασίας περιεχομένου με ανάλυση σε πραγματικό χρόνο χρησιμοποιώντας την ανάπτυξη SPA. Για να γίνει αυτό με μια παραδοσιακή εφαρμογή ιστού απαιτείται μια συνολική επαναφόρτωση σελίδας για την εκτέλεση ανάλυσης περιεχομένου.

6)Χρησιμοποιεί λιγότερο εύρος ζώνης

Δεν αποτελεί έκπληξη το γεγονός ότι οι SPA καταναλώνουν μικρότερο εύρος ζώνης, αφού φορτώνουν μόνο ιστοσελίδες μία φορά. Εκτός αυτού, μπορούν επίσης να κάνουν καλά σε περιοχές με αργή σύνδεση στο Διαδίκτυο. Ως εκ τούτου, είναι βολικό για όλους να χρησιμοποιούν, ανεξάρτητα από την ταχύτητα του Διαδικτύου.

Γιατί δεν χρησιμοποιούμε Single-Page Application:

1)Βελτιστοποίηση SEO

Μερικοί άνθρωποι πιστεύουν ότι τα SPA παρέχουν κακή βελτιστοποίηση SEO. Αυτό συμβαίνει επειδή οι εφαρμογές μιας σελίδας λειτουργούν σε JavaScript και κατεβάζουν δεδομένα κατόπιν αιτήματος από την πλευρά του πελάτη. Η διεύθυνση URL δεν αλλάζει πραγματικά και οι διαφορετικές σελίδες δεν έχουν τις μοναδικές διευθύνσεις URL τους. Είναι δύσκολο να βελτιστοποιήσετε αυτούς τους ιστότοπους για μηχανές αναζήτησης, καθώς οι περισσότερες σελίδες δεν μπορούν να σαρωθούν από bots αναζήτησης. Πρόσφατα, η Google ξεκίνησε ένα νέο σχήμα για την αύξηση της βελτιστοποίησης SEO μιας σελίδας. Το Google ευρετηριάζει τώρα δυναμικές σελίδες. Αλλά για αυτό, οι προγραμματιστές πρέπει να βεβαιωθούν ότι τα αρχεία JavaScript μπορούν να ευρετηριαστούν από την Google (επειδή η Google τα εκτελεί στο πρόγραμμα ανίχνευσης). Πρέπει επίσης να επαληθεύσουν ότι ένας ιστότοπος χρησιμοποιεί λειτουργία HTML5 στο σχήμα URL.

2)Ιστορικό προγράμματος περιήγησης

Ένα SPA δεν σώζει τα άλματα των επισκεπτών μεταξύ κρατών. Αυτό σημαίνει ότι όταν οι χρήστες κάνουν κλικ στο κουμπί επιστροφής, δεν θα επιστρέψουν. Ένα πρόγραμμα περιήγησης μεταφέρει τους χρήστες στην προηγούμενη σελίδα και όχι στην προηγούμενη κατάσταση σε μια εφαρμογή. Για την επίλυση αυτού του προβλήματος, υπάρχει ένα API Ιστορικού HTML5 με το οποίο οι προγραμματιστές μπορούν να εξοπλίσουν τα πλαίσια SPA τους. Το API ιστορικού προσφέρει στους προγραμματιστές πρόσβαση στο ιστορικό πλοήγησης του προγράμματος περιήγησης μέσω JavaScript.

3)Θέματα ασφάλειας

Οι εφαρμογές μιας σελίδας είναι λιγότερο απρόσβλητες από επιθέσεις σεναρίων μεταξύ ιστότοπων (XSS) από ότι οι εφαρμογές πολλαπλών σελίδων. Χρησιμοποιώντας το XSS, οι χάκερ μπορούν να εισάγουν σενάρια πελάτη σε εφαρμογές ιστού. Ένα ζήτημα ασφάλειας είναι η έκθεση ευαίσθητων δεδομένων. Εάν οι προγραμματιστές δεν είναι προσεκτικοί σχετικά με τα δεδομένα που περιέχονται στην αρχική φόρτωση σελίδας, μπορούν εύκολα να στείλουν δεδομένα που δεν πρέπει να εκτίθενται σε όλους τους χρήστες. Το σύνολο ενός SPA δεν είναι γενικά ορατό στο πρόγραμμα περιήγησης, το οποίο μπορεί να παρέχει μια ψευδή αίσθηση ασφάλειας. Ένας ακόμη λόγος για τον οποίο τα SPA μπορεί να είναι ανασφαλή λείπει ο έλεγχος πρόσβασης σε λειτουργικό επίπεδο. Δεδομένου ότι οι προγραμματιστές μετακινούν λειτουργίες και λογικές από τον διακομιστή και αποστέλλονται στον πελάτη, είναι πολύ εύκολο να παρέχεται στον πελάτη πρόσβαση σε λειτουργίες που δεν πρέπει να επιτρέπεται να χρησιμοποιούν.

4)Χρησιμοποιεί πολλούς πόρους προγράμματος περιήγησης

Τα SPA απαιτούν πολλούς πόρους από το πρόγραμμα περιήγησης, καθώς το πρόγραμμα περιήγησης εκτελεί τις περισσότερες από τις εργασίες για τα SPA. Η δημιουργία SPA συχνά απαιτεί από τους χρήστες να χρησιμοποιούν τα πιο πρόσφατα προγράμματα περιήγησης με υποστήριξη για ορισμένες σύγχρονες λειτουργίες.

Isomorphic Web Application

Μια Isomorphic Web Application κατηγοριοποιείται ως Single Page Application, αλλά διαφέρει στον τρόπο με τον οποίο τα δεδομένα και οι επιχειρηματικές λογικές χειρίζονται. Σε μια Isomorphic Web Application ο κώδικας στον περιηγητή και στον σέρβερ είναι ίδιος, δηλαδή μοιράζονται τον ίδιο κώδικα εφαρμογής. Επειδή ο κώδικας είναι κοινόχρηστος αυτό έχει ως αποτέλεσμα να υπάρχουν πλεονεκτήματα.

The main criteria for an isomorphic web application is that it needs to be written in the JavaScript language; since it needs to be supported in the web browser. There are several server software available for parsing and executing JavaScript, one of them is NodeJS[17], which efficiently runs the application code and supports multi-core technology.

- Οι προγραμματιστές πρέπει να διατηρούν μόνο μία βάση κώδικα
- Οι clients μπορούν να επισκεφθούν τον ιστότοπο χωρίς να έχουν ενεργοποιημένη τη JavaScript
- Χρησιμοποίηση των ίδιων βιβλιοθηκών τόσο στον διακομιστή όσο και στον client

Τα βασικά κριτήρια για μια Isomorphic Web Application είναι ότι χρειάζεται να είναι γραμμένη σε γλώσσα JavaScript επειδή πρέπει να υποστηρίζεται στον περιηγητή. Υπάρχουν πολλά διαθέσιμα λογισμικά διακομιστή για ανάλυση και εκτέλεση JavaScript, ένα από αυτά είναι το NodeJS, το οποίο εκτελεί αποτελεσματικά τον κώδικα των εφαρμογών και υποστηρίζει ακόμη τεχνολογία πολλαπλών πυρήνων. Η κοινή χρήση του κωδικού εφαρμογής μεταξύ client και διακομιστή και σέρβερ δεν επηρεάζει την, αντιθέτως την βελτιστοποιεί. Κατά τη διάρκεια κατασκευής του έργου, ο πηγαίος κώδικας κάνει compile και ελαχιστοποιείται σε πακέτα και μοιράζεται αντίστοιχα ένα πακέτο για τον client και ένα πακέτο για το σέρβερ. Κάθε πακέτο έχει βελτιστοποιηθεί για το συγκεκριμένο περιβάλλον, το οποίο οδηγεί σε μη περιττό κώδικα στο σέρβερ ή / και τον client.

Γιατί χρησιμοποιούμε Isomorphic Web Application:

1)Εξάλειψη του FOUC

Με την αρχιτεκτονική AngularJS, Ember (για τώρα) και τύπου SPA, όταν ένας χρήστης χτυπήσει για πρώτη φορά τη σελίδα, πρέπει να γίνει λήψη όλων των στοιχείων. Με τις εφαρμογές SPA, αυτό μπορεί να διαρκέσει ένα δευτερόλεπτο και οι περισσότεροι χρήστες

αυτές τις μέρες αναμένουν χρόνο φόρτωσης μικρότερο από δύο δευτερόλεπτα. Κατά τη φόρτωση του περιεχομένου, η σελίδα δεν αποδίδεται. Αυτό ονομάζεται "flash του ασταθούς περιεχομένου" (FOUC). Ένα πλεονέκτημα μιας ισόμορφης προσέγγισης στη δημιουργία εφαρμογών είναι ότι λαμβάνετε τα οφέλη ταχύτητας απόδοσης στον διακομιστή και μπορείτε ακόμα να αποδίδετε στοιχεία μετά τη φόρτωση της σελίδας στον πελάτη.

Η δουλειά μιας ισόμορφης εφαρμογής δεν είναι να αντικαταστήσει το παραδοσιακό API διακομιστή, αλλά απλώς να βοηθήσει στην εξάλειψη του FOUC και να δώσει στους χρήστες την καλύτερη, ταχύτερη εμπειρία που συνηθίζουν.

2)Κοινόχρηστος κώδικας

Ένα μεγάλο πλεονέκτημα είναι η χρήση του ίδιου κώδικα στον πελάτη και στον διακομιστή. Απλώς δημιουργήστε τα στοιχεία σας και θα λειτουργήσουν και στα δύο μέρη. Στα περισσότερα συστήματα, όπως Rails , ASP.NET MVC , συνήθως θα έχετε `erb` `cshtml` προβολές για απόδοση στο διακομιστή. Στη συνέχεια, πρέπει να έχετε πρότυπα από την πλευρά του πελάτη, όπως Handlebars ή Hogan.js, τα οποία συχνά αντιγράφουν λογική. Με το React, τα ίδια στοιχεία λειτουργούν και στα δύο μέρη.

3)Προοδευτική βελτίωση

Η απόδοση διακομιστή σας επιτρέπει να στείλετε το HTML barebones που χρειάζεται ένας πελάτης για να εμφανίσει έναν ιστότοπο. Στη συνέχεια, μπορείτε να βελτιώσετε την εμπειρία ή να αποδώσετε περισσότερα στοιχεία στον πελάτη.

Η παροχή μιας ωραίας εμπειρίας σε έναν χρήστη με ένα ανατρεπόμενο τηλέφωνο στην Αφρική, καθώς και μια βελτιωμένη εμπειρία σε έναν χρήστη σε MacBook Pro 15 ιντσών με οθόνη Retina, συνδεδεμένη με τη νέα οθόνη 4K, είναι συνήθως μια αρκετά κουραστική εργασία.

Το React πηγαίνει πάνω και πέρα από την κοινή χρήση στοιχείων. Όταν αποδίδετε στοιχεία React στον διακομιστή και αποστέλλετε τον HTML στον πελάτη, το React από την πλευρά του πελάτη παρατηρεί ότι ο HTML υπάρχει ήδη. Συνδέει απλά χειριστές συμβάντων στα υπάρχοντα στοιχεία και είστε έτοιμοι να ξεκινήσετε.

Αυτό σημαίνει ότι μπορείτε να αποστείλετε μόνο τον HTML που απαιτείται για την απόδοση της σελίδας. Στη συνέχεια, τυχόν πρόσθετα πράγματα μπορούν να τραβηχτούν και να αποδοθούν στον πελάτη, όπως απαιτείται. Μπορείτε να επωφεληθείτε από τη γρήγορη φόρτωση σελίδων με την απόδοση διακομιστή και μπορείτε να επαναχρησιμοποιήσετε τα στοιχεία.

Γιατί δεν χρησιμοποιούμε Isomorphic Web Application:

1)Μεγάλος χρόνος φόρτωσης

Επειδή ο ιστότοπος είναι ιμάντας εκκίνησης, χρειάζεται αρκετός χρόνος για να εμφανιστεί το περιεχόμενο της σελίδας μετά την αρχική φόρτωση. Η αρχική φόρτωση πραγματοποιείται όταν πληκτρολογείτε το `example.com` στο πρόγραμμα περιήγησής σας και πατάτε `enter`. Οτιδήποτε επιστρέφει το πρόγραμμα περιήγησης από το αρχικό φορτίο είναι ό, τι στέλνει ο διακομιστής. Εάν ο διακομιστής αποστέλλει ένα κενό πρότυπο HTML και `javascript` για να αποδώσει περιεχόμενο σε αυτό το πρότυπο, τότε ο χρήστης θα δει μια κενή σελίδα και μια σελίδα φόρτωσης ίσως. Πόσο καιρό ο χρήστης πρέπει να περιμένει έως ότου εμφανιστεί κάτι να κλιμακώνεται με την πολυπλοκότητα της ιστοσελίδας και πόσο γρήγορα είναι η υπηρεσία του διαδικτύου σε μια κινητή συσκευή, οι σελίδες τείνουν να φορτώνουν πολύ πιο αργά.

2)Κακό SEO

Οι μηχανές αναζήτησης και η κοινή χρήση κοινωνικών δικτύων είναι δύο από τα πιο σημαντικά μέσα απόκτησης νέων χρηστών. Για να κατατάξει η Google το περιεχόμενο της ιστοσελίδας σας, πρέπει να γνωρίζει ποιο περιεχόμενο υπάρχει στην ιστοσελίδα σας. Η Google αναπτύσσει μια σειρά από προγράμματα ανίχνευσης, τα οποία είναι απλά προγράμματα που υποβάλλουν αιτήματα σε ιστοσελίδες, εξετάζουν την απόκριση, απορρίπτουν το περιεχόμενο από το HTML και εξετάζουν πώς να κατατάξουν αυτήν την ιστοσελίδα μεταξύ άλλων ιστοσελίδων στο Διαδίκτυο με βάση τη συνάφεια. Αυτά τα προγράμματα ανίχνευσης δεν εκτελούν γενικά JavaScript ή περιμένουν για μεγάλο χρονικό διάστημα για να εμφανιστεί η σελίδα. Εάν η ιστοσελίδα σας δίνει κενές σελίδες στο πρόγραμμα ανίχνευσης κατά την αρχική φόρτωση, τότε η Google δεν θα γνωρίζει ποια είναι η σελίδα σας για να τοποθετήσει με ακρίβεια την ιστοσελίδα σας στη λίστα επισκέψεων όταν καταχωρείται σχετικό ερώτημα αναζήτησης στο `google.com`.

Πλεονεκτήματα:

<u>Πλεονεκτήματα Single-Page Application:</u>	<u>Πλεονεκτήματα Isomorphic Web Application:</u>
<ul style="list-style-type: none">• Γρήγορος χρόνος φόρτωσης	<ul style="list-style-type: none">• Απλοποιημένο και βελτιωμένο SEO — τα bot και τα προγράμματα ανίχνευσης μπορούν να διαβάσουν όλα τα δεδομένα κατά τη φόρτωση της σελίδας.
<ul style="list-style-type: none">• Απρόσκοπτη εμπειρία χρήστη	<ul style="list-style-type: none">• Η απόδοση αυξάνεται στην απόδοση που γίνεται αντιληπτή από τον χρήστη.
<ul style="list-style-type: none">• Επιτρέπει ευκολία στην κατασκευή εφαρμογών πλούσιων σε χαρακτηριστικά	<ul style="list-style-type: none">• Κέρδη συντήρησης.
<ul style="list-style-type: none">• Χρησιμοποιεί λιγότερο εύρος ζώνης	<ul style="list-style-type: none">• Βελτιωμένη προσβασιμότητα, επειδή ο χρήστης μπορεί να δει την εφαρμογή χωρίς JavaScript
<ul style="list-style-type: none">• Δεν υπάρχουν επιπλέον ερωτήματα στον διακομιστή για λήψη σελίδων.	<ul style="list-style-type: none">• Οι παλιές συσκευές μπορούν να περιηγηθούν στον ιστότοπό μας επειδή η εφαρμογή επιστρέφει HTML, η οποία διαφέρει από τις κοινές Εφαρμογές Μίας Σελίδας (SPA), όπου η ετικέτα περιέχει JavaScript.
<ul style="list-style-type: none">• Φιλικό προς τον χρήστη.	<ul style="list-style-type: none">• Το αίτημα της πρώτης σελίδας είναι γρήγορο και τα επόμενα είναι ακόμη πιο γρήγορα. σε αντίθεση με τα κοινά ΖΕΠ, όπου το πρώτο αίτημα χρησιμοποιείται για τη φόρτωση της εφαρμογής και στη συνέχεια,πραγματοποιείται μετ 'επιστροφής για να ληφθεί αυτό που ζητήθηκε.
<ul style="list-style-type: none">• Βελτίωση απόδοσης, Εφαρμογή μίας σελίδας μπορεί να βελτιώσει την απόδοση με πολλούς τρόπους, Φόρτωση αρχείου μίας φορά σε HTML, CSS, JS.	<ul style="list-style-type: none">• Υπάρχει λιγότερος κώδικας, καθώς κοινοποιείται τόσο από τον πελάτη όσο και από τον διακομιστή.
<ul style="list-style-type: none">• Εφαρμογές μίας σελίδας είναι	

<p>εξαιρετικά εύκολο να εφαρμοστούν στην παραγωγή, ακόμη και στην έκδοση με την πάροδο του χρόνου.</p>	
<ul style="list-style-type: none"> Μια εφαρμογή μίας σελίδας είναι εξαιρετικά απλή στην εφαρμογή, σε σύγκριση με τις πιο παραδοσιακές εφαρμογές που παρέχονται από διακομιστή: είναι πραγματικά μόνο ένα index.html αρχείο, με δέσμη CSS και δέσμη Javascript 	
<ul style="list-style-type: none"> Μπορείτε να διαμορφώσετε τον διακομιστή που εξυπηρετεί το SPA σας με μια παράμετρο που καθορίζει ποια έκδοση της εφαρμογής frontend θα κατασκευαστεί. 	
<ul style="list-style-type: none"> Σε ένα SPA, μετά την αρχική φόρτωση της σελίδας, δεν αποστέλλεται πλέον HTML μέσω του δικτύου. Αντ 'αυτού, ζητούνται μόνο δεδομένα από τον διακομιστή (ή αποστέλλονται στον διακομιστή). 	
<ul style="list-style-type: none"> Η εφαρμογή θα αισθανθεί γρηγορότερα επειδή χρησιμοποιείται μικρότερο εύρος ζώνης και δεν πραγματοποιούνται ανανέωση πλήρους σελίδας καθώς ο χρήστης πλοηγείται στην εφαρμογή. 	
<ul style="list-style-type: none"> Μπορείτε επίσης να χωρίσετε τα πακέτα σε πολλά μέρη, αν χρειαστεί, χρησιμοποιώντας διαχωρισμό κώδικα. 	
<ul style="list-style-type: none"> Το τμήμα frontend της εφαρμογής είναι πολύ απλό στην έκδοση στην παραγωγή, επιτρέποντας απλοποιημένη ανάπτυξη και επαναφορά στην προηγούμενη έκδοση του frontend, εάν χρειάζεται 	
<ul style="list-style-type: none"> Μετά την εκκίνηση, μόνο τα δεδομένα αποστέλλονται μέσω του καλωδίου ως ωφέλιμο φορτίο JSON ή κάποια άλλη μορφή. Αλλά κανένα HTML ή CSS δεν αποστέλλεται πλέον μέσω του καλωδίου μετά την εκτέλεση της εφαρμογής. 	

Μειονεκτήματα:

<u>Μειονεκτήματα Single-Page Application:</u>	<u>Μειονεκτήματα <i>Isomorphic Web Application</i>:</u>
<ul style="list-style-type: none">• Δεν αποδίδει καλά με το SEO	<ul style="list-style-type: none">• Ο διακομιστής Ιστού μπορεί πλέον να γραφτεί μόνο σε javascript
<ul style="list-style-type: none">• Χρησιμοποιεί πολλούς πόρους του προγράμματος περιήγησης	<ul style="list-style-type: none">• Οι Isomorphic Web Apps δεν μπορούν ποτέ να είναι Προοδευτική Βελτίωση για μη ασήμαντες εφαρμογές
<ul style="list-style-type: none">• Ασφάλεια : Η σύγκριση με την παραδοσιακή σελίδα Η εφαρμογή μίας σελίδας είναι λιγότερο ασφαλής λόγω της δέσμης ενεργειών μεταξύ ιστότοπων (XSS).	<ul style="list-style-type: none">• Αποκλεισμός έναντι μη αποκλεισμού ροής δεδομένων
<ul style="list-style-type: none">• Διαρροή μνήμης : Η διαρροή μνήμης στο JavaScript μπορεί ακόμη και να προκαλέσει επιβράδυνση του ισχυρού συστήματος.	<ul style="list-style-type: none">• Χρόνος-προς-αλληλεπίδραση και η Uncanny Valley
<ul style="list-style-type: none">• Ο πελάτης πρέπει να ενεργοποιήσει το JavaScript, Δημιουργία μίας σελίδας με JavaScript, οπότε το JavaScript πρέπει να είναι ενεργοποιημένο στο πρόγραμμα περιήγησης πελάτη. Η JavaScript είναι ενεργοποιημένη σε όλα τα σύγχρονα προγράμματα περιήγησης από προεπιλογή.	<ul style="list-style-type: none">• Οι κινητές συσκευές παγώνουν κατά την ανάλυση του javascript

Βιβλιογραφία

Μέρος Α

<https://stories.jotform.com/7-reasons-why-you-should-use-react-ad420c634247>

<https://railsware.com/blog/why-use-react/>

<https://www.simform.com/why-use-react/>

<https://www.educba.com/uses-of-react-js/>

<https://ddi-dev.com/blog/programming/pros-and-cons-reactjs-web-app-development/>

<https://rubygarage.org/blog/pros-and-cons-of-reactjs>

<https://programmingwithmosh.com/react/react-vs-angular/>

<https://angular.io/guide/architecture>

https://edoc.sub.uni-hamburg.de/haw/volltexte/2018/4350/pdf/BA_Wohlgethan_2176410.pdf

<https://www.goodworklabs.com/angular-js-development-pros-cons/>

<https://www.pluralsight.com/blog/software-development/angular-101>

<https://www.edureka.co/blog/advantages-and-disadvantages-of-angular/>

<https://yalantis.com/blog/when-to-use-angular/>

Μέρος Β

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.736.4771&rep=rep1&type=pdf>

<https://www.bloomreach.com/en/blog/2018/07/what-is-a-single-page-application.html>

<https://www.diva-portal.org/smash/get/diva2:1037481/FULLTEXT01.pdf>

<https://rubygarage.org/blog/single-page-app-vs-multi-page-app>

<https://blog.angular-university.io/why-a-single-page-application-what-are-the-benefits-what-is-a-spa/>

<https://www.netsolutions.com/insights/single-page-application/>

<https://www.zymphonies.com/blog/single-page-application-advantages-and-disadvantages>

<https://livebook.manning.com/book/isomorphic-web-applications/chapter-1/>

<https://www.lullabot.com/articles/what-is-an-isomorphic-application>

<https://www.smashingmagazine.com/2015/04/react-to-the-future-with-isomorphic-apps/>

<https://www.diva-portal.org/smash/get/diva2:825784/FULLTEXT01.pdf>