



ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Σχεδιασμός και ανάπτυξη συστήματος για ηλεκτρονικό φάκελο
ασθενούς

Ανδρέας Γλαράκης

A.M. 1894

Επιβλέπων: Σωτήριος Χριστοδούλου, Επίκουρος Καθηγητής



Πάτρα, 2022

Πανεπιστήμιο Πελοποννήσου, Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών,
Γλαράκης Ανδρέας

© 2022 – Με την επιφύλαξη παντός δικαιώματος



SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

DIPLOMA THESIS

Design and development of a program for electronic patient record

Andreas Glarakis

A.M. 1894

Supervisor: Sotirios Christodoulou, Assistant professor



Patras, 2022

University of Peloponnese, Department of Electrical and Computer Engineering,

Glarakis Andreas

© 2022– All rights reserved

*Η παρούσα πτυχιακή εργασία αφιερώνεται σε αυτούς που με την αγάπη τους αγκάλιασαν την
προσπάθειά μου.*

ΠΡΟΛΟΓΟΣ

Εκφράζω τις βαθύτατες ευχαριστίες μου στον επιβλέποντα Καθηγητή του τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Πελοποννήσου, κ. Σωτήριο Χριστοδούλου για την συνεχή επιστημονική καθοδήγηση που μου προσέφερε, την εποικοδομητική κριτική του και το αμέριστο ενδιαφέρον του καθ'όλη την διάρκεια της εκπόνησης αυτής της εργασίας.

Επίσης θα ήθελα να ευχαριστήσω εκείνους που με την υποστήριξή τους βοήθησαν την προσπάθειά μου και ομόρφυναν τη διαδρομή των φοιτητικών μου χρόνων.

ΠΕΡΙΛΗΨΗ

Τις τελευταίες δεκαετίες, ιδιαίτερα όμως στην σύγχρονη εποχή, το Διαδίκτυο είναι απαραίτητο και αναπόσπαστο κομμάτι της καθημερινότητας μας. Η χρήση του, αδιαμφισβήτητα, προσφέρει ποικίλα οφέλη σε οποιονδήποτε είναι πρόθυμος να το χρησιμοποιήσει με κατάλληλο τρόπο. Είναι κοινώς αποδεκτό, το γεγονός ότι το Διαδίκτυο αποτελεί την ισχυρότερη πηγή πληροφορίας και ένα από τα χρησιμότερα και πολυτιμότερα εργαλεία της καθημερινότητας. Ένα από τα βασικά προτερήματά του είναι η δυνατότητα για γρήγορες λύσεις που προσφέρει σε προβλήματα της καθημερινότητας. Αντικείμενο της παρούσας διπλωματικής εργασίας είναι ο σχεδιασμός και η υλοποίηση ενός διαχειριστικού περιβάλλοντος για τη χρήση και αποθήκευση φακέλων ασθενών στοχεύοντας στην διευκόλυνση και καλύτερη οργάνωση της νοσοκομειακής ρουτίνας. Για την εφαρμογή αυτή χρησιμοποιήθηκαν τεχνολογίες αιχμής όπως το framework React Admin και το διαχειριστικό περιβάλλον (CMS) Strapi.

Λέξεις κλειδιά: React JS, React-admin, Material UI, Strapi, MySQL, GraphQL

ABSTRACT

Over the last few decades, especially in the recent contemporary age, the internet has become a necessity and an integral part of our everyday lives. Anyone willing to make an effort to use it properly, will without a doubt, enjoy its various benefits. It's commonly acceptable, the fact that the internet constitutes the most powerful source of information, and one of its most useful, valuable and everyday tools. One of the basic advantages it offers is the possibility for quick solutions to everyday problems. The object of this thesis is the design and implementation of an administrative environment for the use and storage of patient files, aiming to facilitate and better organize the hospital routine. Cutting-edge technologies such as the React Admin framework and the content management system (CMS) Strapi were used for this application.

Keywords: React JS, React-admin, Material UI, Strapi, MySQL, GraphQL

ΠΕΡΙΕΧΟΜΕΝΑ

1. ΕΙΣΑΓΩΓΗ.....	1
2. ΕΡΓΑΛΕΙΑ ΓΙΑ ΤΗΝ ΑΝΑΠΤΥΞΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.....	2
2.1 Frontend Technologies	2
2.1.1 Hypertext Markup Language (HTML).....	2
2.1.2 Cascading Style Sheets (CSS).....	3
2.1.3 JavaScript (JS).....	4
2.1.4 ReactJS	6
2.1.5 React admin	8
2.2 Backend Technologies	9
2.2.1 Database	9
2.2.1.1 MySQL	10
2.2.2 Node.js	11
2.2.3 Content Management System (CMS)	13
2.2.4 Strapi	14
2.3 Tools and compilers	15
2.3.1 Node Package Manager (NPM).....	15
2.3.2 Webpack	15
2.3.3 Babel	16
2.4 API	17
2.4.1 Rest API	17
2.4.2. GraphQL	18
3. ΛΕΙΤΟΥΡΓΙΑ ΕΦΑΡΜΟΓΗΣ	19
3.1 Ενότητα Ιατροί.....	19
3.2 Ενότητα Νοσοκόμοι	21
3.3 Ενότητα Ασθενείς.....	22
3.4 Ενότητα Ανοσοθεραπεία	23
4. ΕΓΚΑΤΑΣΤΑΣΗ ΕΦΑΡΜΟΓΗΣ.....	26

1.ΕΙΣΑΓΩΓΗ

Είναι γνωστό ότι οι διάφορες υπηρεσίες έχουν σαν τροχοπέδη την επιβράδυνση της ομαλής λειτουργίας τους εξαιτίας της γραφειοκρατίας. Ιδιαίτερα όταν πρόκειται για λειτουργίες αρχειοθέτησης και λιστών υπάρχει σημαντικό πρόβλημα διαχείρισης και διεκπεραίωσης των αναγκών που προκύπτουν. Συνεπώς είναι απαραίτητη η δημιουργία ενός διαχειριστικού περιβάλλοντος που θα καθιστά εύκολη και γρήγορη την πρόσβαση σε αρχεία που αλλιώς θα ήταν ιδιαίτερα χρονοβόρο να βρεθούν.

Η παρούσα εργασία εκπονήθηκε για την διευκόλυνση ενός νοσοκομειακού περιβάλλοντος ώστε να βοηθήσει στην ψηφιοποίηση φακέλων ασθενών και διαφόρων άλλων ενεργειών που θα βελτιώσουν την λειτουργία του. Προσφέρει μία ολοκληρωμένη λύση και δύναται να εφαρμοστεί άμεσα.

2. ΕΡΓΑΛΕΙΑ ΓΙΑ ΤΗΝ ΑΝΑΠΤΥΞΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

2.1 Frontend Technologies

2.1.1 Hypertext Markup Language (HTML)

Η HTML (Hyper Text Markup Language) είναι η κύρια γλώσσα σήμανσης για τις ιστοσελίδες, και τα στοιχεία της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων. Το Hyper Text είναι η μέθοδος με την οποία επιτρέπεται η μετακίνηση στον ιστό κάνοντας κλικ σε ένα ειδικό κείμενο που ονομάζεται υπερσύνδεσμος. Ο όρος Hyper (υπερβολικό) σημαίνει απλώς ότι δεν είναι γραμμικό, δηλαδή η μετάβαση σε οποιοδήποτε σημείο του διαδικτύου γίνεται οποτεδήποτε, δεν υπάρχει καμία σειρά για να γίνουν τα πράγματα. Markup (σήμανση) είναι αυτό που κάνουν οι ετικέτες (tags) HTML στο κείμενο μέσα σε αυτές, δηλαδή το επισημαίνουν ως ένα συγκεκριμένο τύπο κειμένου (συνηθίζεται η πλάγια γραφή). Η HTML είναι μια γλώσσα (Language) καθώς έχει λέξεις-κλειδιά και σύνταξη όπως κάθε άλλη γλώσσα. Η γραφή της είναι υπό μορφή στοιχείων τα οποία αποτελούνται από ετικέτες (tags), οι οποίες περικλείονται μέσα σε σύμβολα “< >”, (μεγαλύτερο από) και (μικρότερο από) μέσα στο περιεχόμενο της ιστοσελίδας. Η λειτουργία των ετικετών είναι συνήθως ανά ζεύγη, για παράδειγμα ως ετικέτα έναρξης και ως ετικέτα λήξης και ανάμεσα από αυτές τοποθετείται το κείμενο. Τα αρχεία HTML χρησιμοποιούνται από το πρόγραμμα περιήγησης, χωρίς όμως να τα εμφανίζει, ώστε να προβάλλει το περιεχόμενο της ιστοσελίδας για να είναι κατανοητό από τον άνθρωπο. Μέσω αυτής της γλώσσας ενσωματώνονται ποικίλα αντικείμενα, διαδραστικές φόρμες και εικόνες μέσα στη σελίδα, καθώς επίσης καθορίζονται δομικά στοιχεία για το κείμενο και ενσωματώνονται scripts σε γλώσσες όπως η JavaScript που επηρεάζουν την συμπεριφορά των ιστοσελίδων HTML και έτσι από στατικές γίνονται διαδραστικές. Η ύπαρξη της προήλθε από τον φυσικό Tim Berners-Lee το 1989 τον πρώτο web developer και εφευρέτη του WWW (World Wide Web) και αποτελεί τον σκελετό όλων των ιστοτόπων. Με την πάροδο του χρόνου, η HTML εξελίσσεται και εκδίδονται βελτιωμένες εκδόσεις ή εκδόσεις με πρόσθετο περιεχόμενο. Η HTML5 είναι η πέμπτη και τελευταία ως σήμερα έκδοση της. Στον Πίνακα 1 παρουσιάζονται οι βασικές ετικέτες (tags) της HTML, όπως και η περιγραφή τους.

Πίνακας 1: Παρουσιάζει τα βασικά στοιχεία της HTML.

Στοιχείο HTML	Περιγραφή στοιχείου
<html> ... </html>	Δηλώνει την ιστοσελίδα που θα γραφτεί σε HTML
<head> ... </head>	Οριοθετεί την κεφαλίδα της σελίδας
<title> ... </title>	Καθορίζει τον τίτλο (δεν εμφανίζεται στην σελίδα)
<body> ... </body>	Οριοθετεί το περιεχόμενο της σελίδας
<h1> ... </h1>	Οριοθετεί την επικεφαλίδα (n : αριθμός)
 ... 	Ορίζει το κείμενο ανάμεσα ως bold
<i> ... </i>	Ορίζει το κείμενο ανάμεσα ως italics
 ... 	Εμφανίζει μία ομαδοποιημένη λίστα αντικειμένων
<div> ... </div>	Ξεχωρίζει ή ομαδοποιεί στοιχεία σε ένα μέρος της σελίδας
<p> ... </p>	Ορίζει μια παράγραφο
	Εμφανίζει μια εικόνα
 ... 	Ορίζει έναν υπερσύνδεσμο

2.1.2 Cascading Style Sheets (CSS)

Η CSS (Cascading Style Sheets) είναι μια γλώσσα εμφάνισης για το στυλ που χρησιμοποιείται για την περιγραφή της παρουσίας ενός εγγράφου γραμμένου σε μια γλώσσα σήμανσης, όπως η HTML. Αποτελεί τεχνολογία ακρογωνιαίου λίθου του World Wide Web, παράλληλα με την HTML και την JavaScript. Έχει σχεδιαστεί για να επιτρέπει τον διαχωρισμό της παρουσίασης και του περιεχομένου, συμπεριλαμβανομένης της διάταξης, των χρωμάτων και των γραμματοσειρών. Μέσω αυτού του διαχωρισμού βελτιώνεται η προσβασιμότητα του περιεχομένου, παρέχεται μεγαλύτερη ευελιξία και έλεγχος στις προδιαγραφές των χαρακτηριστικών παρουσίασης, επιτρέπεται σε πολλαπλές σελίδες να μοιράζονται τη μορφοποίηση καθορίζοντας το σχετικό style sheet σε ένα ξεχωριστό αρχείο CSS μειώνοντας έτσι την πολυπλοκότητα και την επανάληψη σε δομικό περιεχόμενο. Η παρουσίαση ενός εγγράφου σε έναν χρήστη σημαίνει μετατροπή του σε μια μορφή που μπορεί να χρησιμοποιήσει το κοινό. Τα προγράμματα περιήγησης έχουν σχεδιαστεί για να παρουσιάζουν οπτικά έγγραφα, για παράδειγμα σε μια οθόνη υπολογιστή, έναν προβολέα ή έναν εκτυπωτή. Η CSS διαθέτει επίσης κανόνες για εναλλακτική μορφοποίηση εάν το περιεχόμενο είναι προσβάσιμο σε μια κινητή συσκευή. Με τον καιρό η CSS εξελίσσεται όλο και περισσότερο επεκτείνοντας τις δυνατότητες της. Η CSS3 είναι η τελευταία έκδοση έως σήμερα και στοχεύει στην βέλτιστη παρουσίαση των αντικειμένων σε μια ιστοσελίδα. Ακόμα, η σύνταξη της CSS χρησιμοποιείται και από άλλες γλώσσες προγραμματισμού, όπως η SASS, η οποία με μερικές μικρές εξαιρέσεις είναι ένα υπερσύνολο της CSS, δηλαδή όλα τα έγκυρα CSS είναι και έγκυρα SCSS και λόγω της ομοιότητας της με την CSS γίνεται ευκολότερη στη σύνταξη. Ουσιαστικά, η CSS χρησιμοποιώντας τα στοιχεία ή τις ετικέτες της HTML προσδίδει μια φιλική προς τον άνθρωπο παρουσίαση στον web browser. Στο Σχήμα 1 παρουσιάζεται ένα παράδειγμα γραφής της.

```

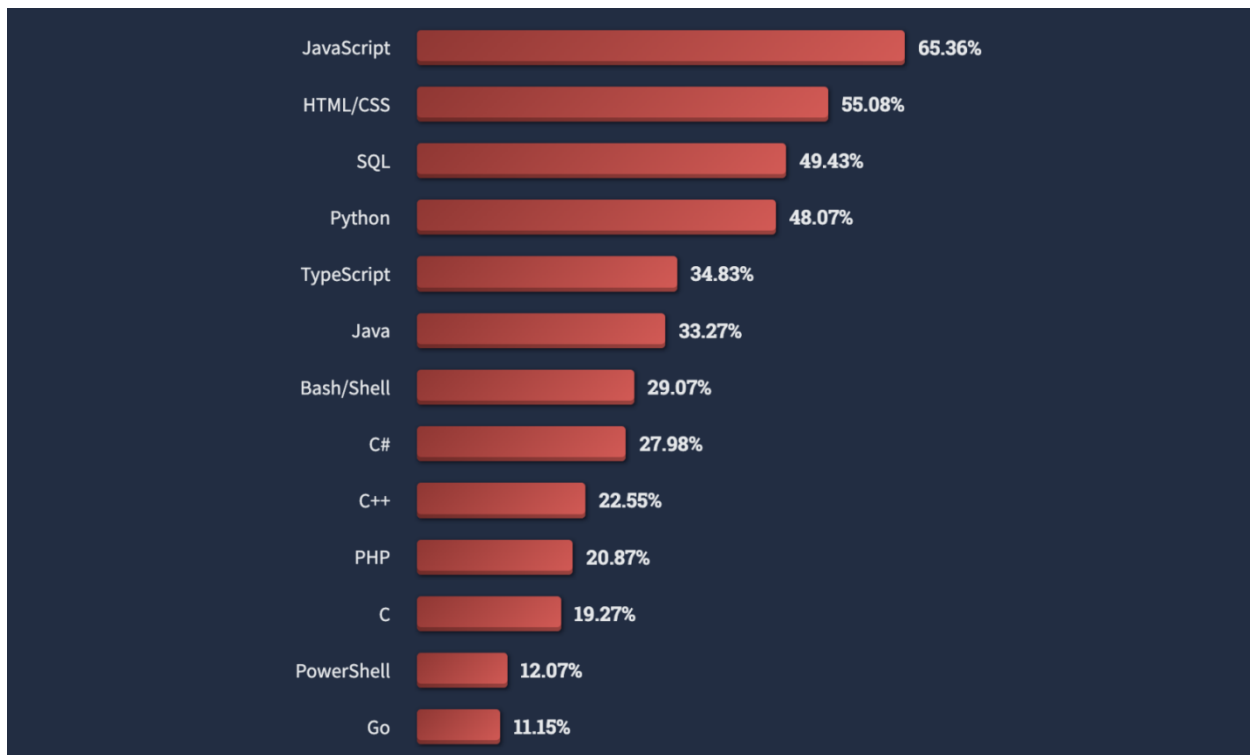
1  /* GENERAL STYLES
2  *-----*/
3  html, body, form, fieldset, img, img a {
4      margin: 0;
5      padding: 0;
6      border: 0;
7  }
8  body {
9      color: #414141;
10     background: url(../images/bg.jpg) repeat-x #ebe8df;
11     font-family: Arial, Helvetica, sans-serif;
12     line-height: 120%;
13     font-size: 12px;
14 }
15
16 a:link, a:visited {
17     color: #685966;
18     text-decoration: underline;
19 }
20 a:hover {
21     color: #2b212c;
22 }
23 .article_separator {
24     line-height: 5px;
25     height: 5px;
26     font-size: 5px;
27 }
28 /* SITE WIDTH
29 *-----*/
30 .rht_container {
31     width: 1020px;
32     margin: 0 auto;
33     margin-top: 25px;

```

Σχήμα1: Παράδειγμα σύνταξης CSS

2.1.3 JavaScript (JS)

Η JavaScript, Vanilla JavaScript, ECMAScript ή συντομογράφως JS, είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού υψηλού επιπέδου με σύντομη σύνταξη, δυναμική πληκτρολόγηση, πρωτότυπα με βάση τον προσανατολισμό των αντικειμένων (prototype-based object-orientation) και πρώτης κατηγορίας λειτουργίες/συναρτήσεις (first-class functions). Παράλληλα με την HTML και τη CSS, η JavaScript είναι μία από τις βασικές τεχνολογίες του World Wide Web. Στο Σχήμα 2 εμφανίζεται η κατάταξη με τις πιο δημοφιλείς γλώσσες προγραμματισμού σύμφωνα με το Stack Overflow (το Stack Overflow είναι μία ιστοσελίδα που διαθέτει ερωτήσεις και απαντήσεις σε ένα ευρύ φάσμα θεμάτων στον προγραμματισμό των υπολογιστών).



Σχήμα 2: Δημοφιλείς Τεχνολογίες

Ενεργοποιεί διαδραστικές ιστοσελίδες και αποτελεί ουσιαστικό μέρος των εφαρμογών ιστού. Χρησιμοποιείται από την πλειοψηφία των ιστοτόπων και τα πιο διαδεδομένα προγράμματα περιήγησης στο Διαδίκτυο διαθέτουν μια αποκλειστική μηχανή JavaScript, όπως η V8 JavaScript Engine του Google Chrome, για να την εκτελέσουν. Ως γλώσσα πολλαπλών παραδειγμάτων, η JavaScript υποστηρίζει ένα στυλ προγραμματισμού που βασίζεται σε γεγονότα, λειτουργικά (functional) και επιτακτικά (imperative). Διαθέτει APIs (Application Programming Interface) ώστε να εργάζεται με κείμενα, συστοιχίες, ημερομηνίες, κανονικές εκφράσεις και το DOM (Document Object Model), όμως δεν περιλαμβάνει I / O (Input / Output), όπως εγκαταστάσεις δικτύωσης, αποθήκευσης ή γραφικών. Είναι βασισμένη στο περιβάλλον υποδοχής στο οποίο είναι ενσωματωμένο για να παρέχει αυτά τα χαρακτηριστικά. Στην αρχή, εφαρμοζόταν μόνο από πλευράς πελάτη (client-side) στους web browsers, όμως πλέον οι μηχανές JavaScript ενσωματώνονται σε πολλούς άλλους τύπους λογισμικού, συμπεριλαμβανομένου του server-side στους διακομιστές διαδικτύου (web servers), στις βάσεις δεδομένων (databases), καθώς και σε προγράμματα εκτός δικτύου και σε περιβάλλοντα εκτέλεσης καθιστώντας την JavaScript διαθέσιμη για τη σύνταξη εφαρμογών για κινητά τηλέφωνα και εφαρμογές υπολογιστών. Θα πρέπει να επισημανθεί πως αν και υπάρχουν αρκετές ομοιότητες μεταξύ της JavaScript και της Java, με κύριες ομοιότητες το όνομα, την σύνταξη και τις αντίστοιχες τυποποιημένες βιβλιοθήκες, οι δυο γλώσσες είναι διακριτές και διαφέρουν σημαντικά στον σχεδιασμό. Η JavaScript επηρεάστηκε από γλώσσες προγραμματισμού όπως η Self και η Scheme. Εξίσου σημαντικό να αναφερθεί, είναι πως η μορφή σειριοποίησης JSON,

που χρησιμοποιείται για την αποθήκευση δομών δεδομένων σε αρχεία ή για την μετάδοση τους σε δίκτυα, βασίζεται στη JavaScript και πως δημιουργήθηκε το 1995 από το πρόγραμμα περιήγησης ιστού της Netscape. Σε αυτό το έργο, έγινε χρήση πολλών εκδόσεων JavaScript, με κύριες τις ES5-ES6 (ECMAScript v5-6). Στο Σχήμα 3 φαίνεται η σύνταξη της.

```
1 // Sum Numbers over an array
2 ▾ const sum = array => array.reduce((acum, number) => {
3     return acum + number;
4 }, 0);
5
6 console.log('01', { sum: sum([ 1, 5, 9 ]) });
7
8 // Sum Numbers multiplied by a factor
9 ▾ const sumProduct = array => array.reduce((acum, pair) => {
10     const [ number, factor ] = pair;
11     return acum + number * factor;
12 }, 0);
13
14 console.log('02', { sumProduct: sumProduct([ [ 1, 1 ] , [ 5, 3 ] , [ 9, 2 ] ]) });
15
16 // Get the max value
17 ▾ const maxNumber = array => array.reduce((acum, number) => {
18     if (!acum) return number;
19     if (acum < number) return number;
20     return acum;
21 }, null);
22
23 console.log('03', { maxNumber: maxNumber([ 1, 5, 9 ]) });
```

Σχήμα 3: Παράδειγμα σύνταξης JS

2.1.4 ReactJS

Η React, γνωστή και ως React.js ή ReactJS, είναι μια βιβλιοθήκη JavaScript για τη δημιουργία διεπαφών χρήστη. Δημιουργήθηκε και διατηρείται από την ομάδα του Facebook και από μια κοινότητα μεμονωμένων προγραμματιστών και εταιρειών. Χρησιμοποιείται ως βάση για την ανάπτυξη εφαρμογών μιας ιστοσελίδας ή κινητών τηλεφώνων, καθώς είναι βέλτιστη ως προς την εξαγωγή ταχέων μεταβαλλόμενων δεδομένων τα οποία πρέπει να καταγράφονται. Ωστόσο, η λήψη δεδομένων είναι μόνο η αρχή όσων συμβαίνουν σε μια ιστοσελίδα, για αυτό και οι σύνθετες εφαρμογές React συνήθως απαιτούν τη χρήση πρόσθετων βιβλιοθηκών, όπως React Router, για το state management, τη δρομολόγηση και την αλληλεπίδραση με ένα API. Ουσιαστικά, είναι το view layer (στρώμα προβολής) για εφαρμογές ιστού. Η React λειτουργεί με τη χρήση components (στοιχείων), δηλαδή μιας αυτοτελούς ενότητας που επιστρέφει κάποιο output. Γράφονται στοιχεία διεπαφής (interface elements), όπως ένα κουμπί ή ένα πεδίο εισαγωγής, ως ένα στοιχείο (component). Τα components μπορεί να είναι σύνθετα, και πιο συγκεκριμένα ένα component μπορεί να περιλαμβάνει ένα ή περισσότερα άλλα components στο output του. Σε γενικές γραμμές, προκειμένου να υλοποιηθεί μια εφαρμογή γράφονται στοιχεία

React (React components) που αντιστοιχούν σε διάφορα στοιχεία διεπαφής, και στη συνέχεια οργανώνονται αυτά τα στοιχεία μέσα σε άλλα στοιχεία υψηλότερου επιπέδου που καθορίζουν τη δομή της εφαρμογής. Είναι σημαντικό να αναφερθεί, ότι κάθε στοιχείο σε μια εφαρμογή React τηρεί αυστηρές αρχές διαχείρισης δεδομένων. Οι σύνθετες, διαδραστικές διεπαφές χρήστη συχνά περιλαμβάνουν σύνθετα δεδομένα και εφαρμογή state. Η επιφάνεια της React είναι περιορισμένη και έχει στόχο να δώσει τα εργαλεία για να μπορέσει να προβλεφθεί ο τρόπος εμφάνισης μια εφαρμογής με ένα δεδομένο σύνολο περιπτώσεων. Ειδικότερα, η React είναι ένα JavaScript framework. Η χρήση αυτού του framework είναι τόσο απλή όσο περιλαμβάνει ένα αρχείο JavaScript στην HTML και χρησιμοποιεί τις εξαγωγές της React σε μια εφαρμογή JavaScript. Σε αντίθεση με πολλούς προκατόχους, η React δε λειτουργεί άμεσα στο DOM (Document Object Model) του προγράμματος περιήγησης, αλλά σε ένα εικονικό DOM (virtual DOM). Αναλυτικότερα, αντί να χειρίζεται το έγγραφο σε ένα πρόγραμμα περιήγησης μετά από αλλαγές στα δεδομένα, που μπορεί να είναι αρκετά αργό, επιλύει τις αλλαγές σε ένα DOM που έχει δημιουργηθεί εξ ολοκλήρου στη μνήμη. Στη συνέχεια, αφού ενημερωθεί το εικονικό DOM, η React καθορίζει με έξυπνο τρόπο τις αλλαγές που πρέπει να κάνει στο DOM του πραγματικού προγράμματος περιήγησης. Εξίσου σημαντικό να αναφερθεί, είναι η συνεχόμενη εξέλιξη και ανανέωση της React από τους εμπειρότερους προγραμματιστές του κόσμου. Η έκδοση που ακολούθησε το συγκεκριμένο έργο είναι η 17.0.2. Στο Σχήμα 6 παρουσιάζεται η σύνταξη JSX (JavaScript XML) ενός component σε React.

```

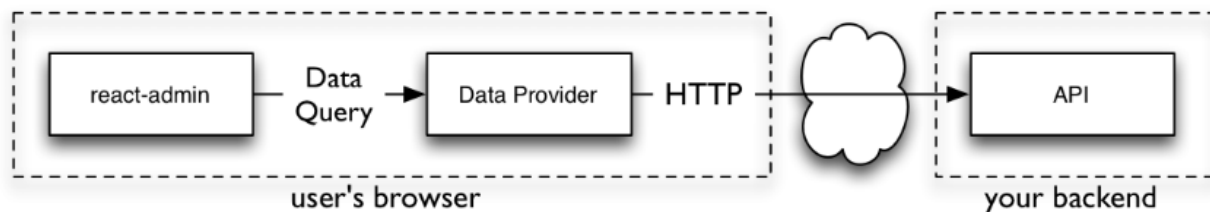
1  import React from "react";
2  import addButtonStyles from "./addButton.styles";
3  import { Button, makeStyles } from "@material-ui/core";
4  import AddIcon from "../../assets/icons/addIcon";
5
6  const AddButton = (props) => {
7    const { label, icon } = props;
8    const useStyles = makeStyles(addButtonStyles);
9    const classes = useStyles();
10
11   return (
12     <Button
13       {...props}
14       startIcon={icon ? icon : <AddIcon/>}
15     >
16       <span className={classes.buttonLabel}>{label}</span>
17     </Button>
18   );
19 }
20
21 export default AddButton;

```

Σχήμα 4: Παράδειγμα ενός React Component.

2.1.5 React admin

Το React-admin είναι ένα framework που χρησιμοποιεί React, Material UI, React Router, Redux και React-final-form για να δώσει ένα διαχειριστικό περιβάλλον που να μπορεί να προσαρμοστεί ανάλογα με τις ανάγκες που μπορεί να προκύψουν. Αναπτύχθηκε και συντηρείται από την εταιρεία προγραμματιστών Marmelab. Πρόκειται για ένα frontend framework το οποίο είναι φτιαγμένο για να συνυπάρχει με οποιοδήποτε REST/GraphQL API. Ο τρόπος επικοινωνίας που χρησιμοποιεί για να επικοινωνήσει με το backend ονομάζεται data provider και η μεθοδολογία του μας δίνεται από το ίδιο το framework. Στο Σχήμα 9 βλέπουμε τον τρόπο λειτουργίας ενός data provider.

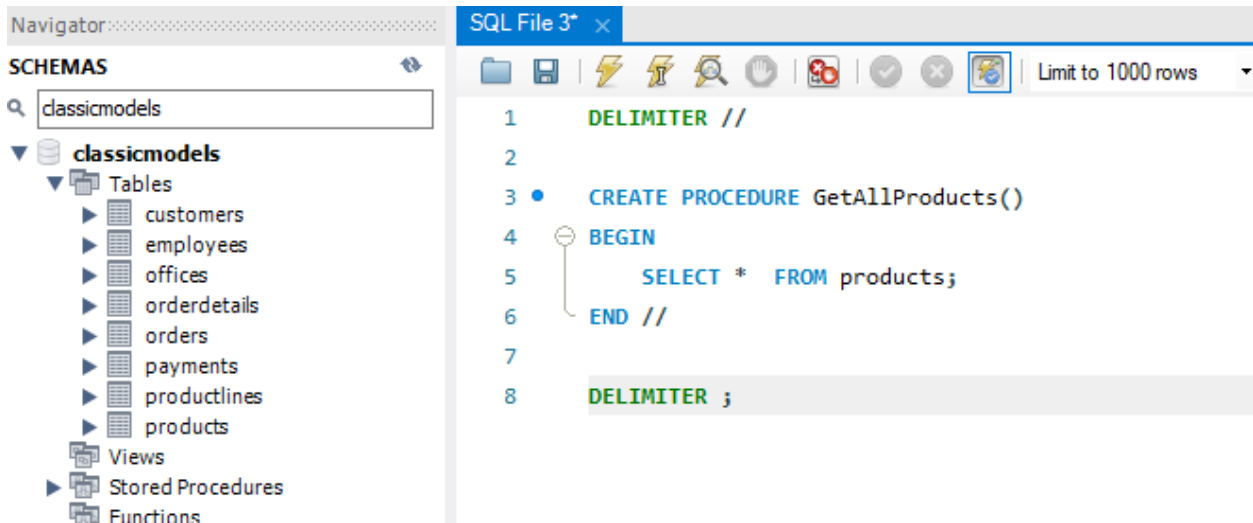


Σχήμα 5: Τρόπος λειτουργίας data provider.

2.2 Backend Technologies

2.2.1 Database

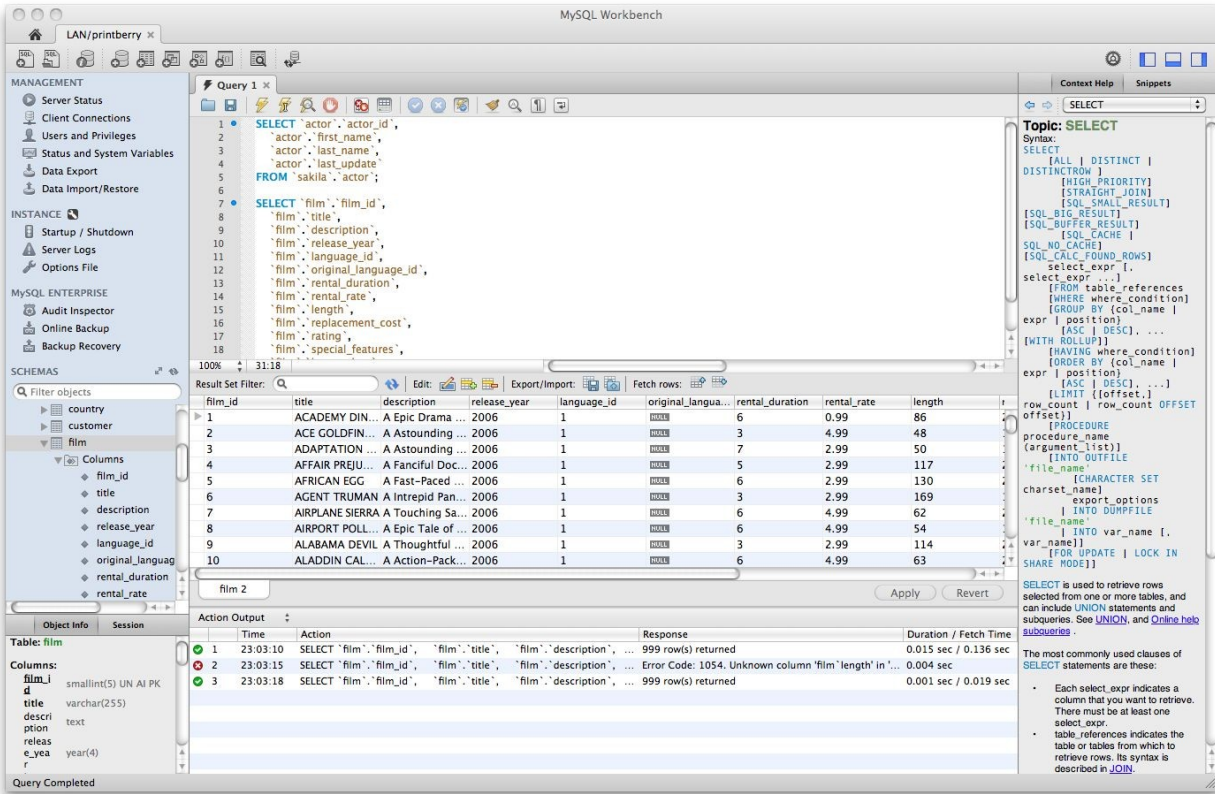
Μια βάση δεδομένων είναι μια οργανωμένη συλλογή δεδομένων, που αποθηκεύεται γενικά και προσπελάζεται ηλεκτρονικά από ένα σύστημα υπολογιστή. Σε 27 μεγάλες και πολύπλοκες βάσεις δεδομένων χρησιμοποιούνται τεχνικές σχεδιασμού και μοντελοποίησης. Το σύστημα διαχείρισης βάσεων δεδομένων DBMS (Database Management System) είναι το λογισμικό που αλληλοεπιδρά με τους τελικούς χρήστες, τις εφαρμογές και την ίδια τη βάση δεδομένων για την καταγραφή και ανάλυση των δεδομένων. Το λογισμικό DBMS περιλαμβάνει επιπλέον τις βασικές διευκολύνσεις που παρέχονται για τη διαχείριση της βάσης δεδομένων. Γενικότερα, ο όρος “βάση δεδομένων” χρησιμοποιείται επίσης για να αναφέρεται σε οποιοδήποτε από τα DBMS, πιο συγκεκριμένα το σύστημα βάσης δεδομένων (database system) ή μια εφαρμογή (application) που σχετίζεται με τη βάση δεδομένων. Τα σύστημα διαχείρισης βάσεων δεδομένων ταξινομούνται σύμφωνα με τα μοντέλα βάσης δεδομένων που υποστηρίζουν. Οι σχεσιακές βάσεις δεδομένων (Relational Databases) κυριαρχούσαν ως προς τη χρήση τους από τους επιστήμονες υπολογιστών τη δεκαετία του '80. Αυτά τα μοντέλα κατηγοριοποιούν τα δεδομένα ως σειρές και στήλες πινάκων και η πλειονότητα των προγραμματιστών χρησιμοποιεί τη γλώσσα υπολογιστών SQL (Structured Query Language) για τη γραφή και την αναζήτηση δεδομένων. Με την εξέλιξη του χρόνου, στο προσκήνιο εμφανίστηκαν και οι μη σχεσιακές βάσεις δεδομένων (Non-Relational Databases), οι οποίες έγιναν δημοφιλείς, γνωστές ως “NoSQL” επειδή χρησιμοποιούν διαφορετικές γλώσσες ερωτημάτων. Ειδικότερα, μια βάση δεδομένων πέρα από την ικανότητα της να αποθηκεύει δεδομένα, παρέχει μέσω του σχεδιασμού και του τρόπου ιεράρχησης των δεδομένων, τα αποκαλούμενα συστήματα διαχείρισης περιεχομένου, δηλαδή τη δυνατότητα γρήγορης άντλησης και ανανέωσης των δεδομένων. Ένα σύστημα διαχείρισης δεδομένων ανοιχτού κώδικα (RDBMS) είναι η MySQL και χρησιμοποιείται από πολλούς δημοφιλείς ιστούς, όπως το Facebook και το YouTube. Παρέχει πρόσβαση πολλών χρηστών σε ένα σύνολο βάσεων δεδομένων “τρέχοντας” έναν εξυπηρετητή (server). Το Σχήμα 10 απεικονίζει τη σύνταξη της SQL.



Σχήμα 6: Παράδειγμα σύνταξης SQL

2.2.1.1 MySQL

Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων. Ο κώδικας είναι διαθέσιμος με άδεια χρήσης δύο ειδών. Μπορεί να χρησιμοποιηθεί η δωρεάν άδεια ανοικτού κώδικα, εφόσον ο χρήστης συμφωνεί με τους όρους αυτής ή η εμπορική άδεια όπου ο χρήστης μπορεί να διανείμει μία εφαρμογή που δεν είναι ανοικτού κώδικα και να περιλαμβάνει την MySQL. Η ονομασία MySQL αποτελείται από δύο στοιχεία. Το My είναι το όνομα της κόρης του συνιδρυτού συστήματος Monty Widenius και το SQL αναφέρεται στην γλώσσα SQL (Structured Query Language). Ένας κώδικας SQL θα πρέπει να εκτελείται με τον ίδιο τρόπο σε όλα τα συστήματα (MySQL, Oracle Database, SQLite κ.α). Κάθε σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων μπορεί να διαφέρει με κάποιο άλλο σε γραφικό επίπεδο αλλά και να παρέχει επιπλέον δυνατότητες. Η μεταφορά ενός κώδικα SQL από μια βάση σε μια διαφορετική υλοποίηση μπορεί να παρουσιάσει σφάλματα και μπορεί να χρειάζεται να γίνουν τροποποιήσεις στον κώδικα για την εκτέλεση του. Για την διαχείριση των βάσεων και την εκτέλεση SQL ερωτημάτων ανακτήθηκαν από τον ιστότοπο: <http://dev.mysql.com/downloads/> δύο βασικά πακέτα λογισμικού: το MySQL Community Server και MySQL Workbench (γραφικού περιβάλλοντος περιεχόμενο). Το πρώτο για την εγκατάσταση του εξυπηρετητή και το δεύτερο για την εκτέλεση λειτουργιών σε αυτόν αντίστοιχα. Ακολουθεί εικόνα από το γραφικό περιβάλλον του MySQL Workbench.



Σχήμα 7: Γραφικό περιβάλλον του MySQL Workbench.

2.2.2 Node.js

Το Node.js είναι μια πλατφόρμα ανάπτυξης λογισμικού ανοιχτού κώδικα που υποστηρίζεται σε πολλές πλατφόρμες, κυρίως διακομιστών, που εκτελεί κώδικα JavaScript έξω από ένα πρόγραμμα περιήγησης. Επιτρέπει στους προγραμματιστές να χρησιμοποιούν JavaScript για να γράψουν εργαλεία γραμμής εντολών (command line tools) και για scripts που εκτελούνται από την πλευρά του διακομιστή (server side scripting) για την παραγωγή δυναμικού περιεχομένου ιστοσελίδας πριν από την αποστολή της σελίδας στο πρόγραμμα περιήγησης του χρήστη. Γενικότερα, το Node.js αντιπροσωπεύει το γνωστό “JavaScript everywhere”, δηλαδή χρήση της γλώσσας JavaScript σε όλο το περιβάλλον λειτουργίας, ενοποιώντας την ανάπτυξη εφαρμογών ιστού γύρω από μία μόνο γλώσσα προγραμματισμού, αντί για διαφορετικές γλώσσες για δέσμες ενεργειών διακομιστή και πελάτη (server-and client-side scripts). Η αρχιτεκτονική του Node.js βασίζεται σε γεγονότα (event-driven) και είναι ικανή για ασύγχρονα I / O. Λόγω των συγκεκριμένων επιλογών σχεδιασμού της, στόχος της είναι η βελτιστοποίηση της απόδοσης και κλιμάκωσης στις εφαρμογές ιστού με πολλές λειτουργίες εισόδου / εξόδου (input / output operations), καθώς και για εφαρμογές ιστού σε πραγματικό χρόνο (real-time Web applications). Το χαρακτηριστικό που ξεχωρίζει το Node είναι, όπως προαναφέρθηκε, η ασύγχρονη επικοινωνία μεταξύ των υπολογιστικών πόρων και αυτό γίνεται εφικτό με την χρήση γεγονότων (events) που προσφέρει η JavaScript, τα ονομαζόμενα callbacks. Ειδικότερα, όταν ένας web browser φορτώσει πλήρως ένα αρχείο, όπου ένας χρήστης πατάει κάποιο κουμπί,

ολοκληρώνεται ένα αίτημα AJAX (Asynchronous JavaScript and XML), και τα γεγονότα αυτά καλούν ένα συγκεκριμένο callback, το οποίο με τη σειρά του επιτρέπει την ροή του κώδικα χωρίς να αφήνει ανενεργό τον επεξεργαστή προκειμένου να εκτελεστεί μια λειτουργία. Το Node.js αρχικά, γράφτηκε από τον Ryan Dahl το 2009 και η ανάπτυξη και συντήρηση του καθοδηγούνταν από τον Dahl και χρηματοδοτήθηκε από τον Joyent. Στο Σχήμα 4 παρουσιάζεται ένα χαρακτηριστικό παράδειγμα Node για έναν HTTP εξυπηρετητή.

```
var sys = require("sys"),
    http = require("http"),
    url = require("url"),
    path = require("path"),
    fs = require("fs");

http.createServer(function(request, response) {
  var uri = url.parse(request.url).pathname;
  var filename = path.join(process.cwd(), uri);
  path.exists(filename, function(exists) {
    if(!exists) {
      response.writeHead(404, {"Content-Type": "text/plain"});
      response.end("404 Not Foundn");
      return;
    }

    fs.readFile(filename, "binary", function(err, file) {
      if(err) {
        response.writeHead(500, {"Content-Type": "text/plain"});
        response.end(err + "n");
        return;
      }

      response.writeHead(200);
      response.end(file, "binary");
    });
  });
}).listen(8080);

console.log("Server running at http://localhost:8080/");
```

Σχήμα 8: Σύνταξη Node.js

2.2.3 Content Management System (CMS)

Μιλώντας για Συστήματα Διαχείρισης Περιεχομένου (Content Management Systems ή CMS), αναφερόμαστε σε διάφορες εφαρμογές παρόμοιες με αυτές ενός κειμενογράφου που επιτρέπουν τη δημιουργία, τον εντοπισμό πληροφοριών, την διαχείριση τους τη δημοσίευση και επίσης στη δυνατότητα που μπορεί κανείς να διαχειρίζεται μια ιστοσελίδα, εύκολα χωρίς ειδικές γνώσεις προγραμματισμού ή δημιουργίας γραφικών. Συνήθως τα κείμενα γράφονται μέσω κάποιων κειμενογράφων παρόμοιων με το Microsoft Word on line, δηλαδή μέσα από το ίδιο το site, με τη χρήση κάποιου εργαλείου WYSIWYG ("What You See Is What You Get"). Αυτού του είδους τα εργαλεία επιτρέπουν στον χρήστη οποιουδήποτε επιπέδου να «γράφει» ιστοσελίδες χωρίς να έχει την παραμικρή γνώση προγραμματισμού ενώ αν σε λίγες περιπτώσεις χρειάζεται η βασική γνώση της γλώσσας HTML. Το περιβάλλον του χρήστη είναι παρόμοιο με κάποια εφαρμογή γραφείου και ο χρήστης μπορεί να γράφει κείμενο, να προσθέτει εικόνες, γραφικά κλπ και αυτό που βλέπει είναι και αυτό που θα φανεί στην ιστοσελίδα του. Τα εργαλεία αυτά υπάρχουν έτοιμα στις πλατφόρμες CMS ως ένθετα, πρόσθετα ή modules όπως αναφέρονται πιο κάτω Τα περισσότερα CMS έχουν την δυνατότητα να διαχειριστούν περιεχόμενο στις εξής μορφές: κείμενα, εικόνες, βίντεο, java animation, πρότυπα σχεδίασης, βάσεις δεδομένων κ.α. Το Σύστημα Διαχείρισης Περιεχομένου είναι λογισμικό το οποίο επιτρέπει στον οποιονδήποτε, ακόμα και αν δεν έχει ιδιαίτερες γνώσεις προγραμματισμού και γλώσσας HTML, να δημιουργήσει και να διαχειριστεί με τρόπο εύκολο και γρήγορο την ιστοσελίδα του. Διαχειρίζεται ολόκληρο τον κύκλο ζωής μιας σελίδας από την δημιουργία της μέχρι και την αρχειοθέτησή της. Τα περισσότερα Συστήματα Διαχείρισης Περιεχομένου είναι λογισμικό ανοικτού κώδικα, διατίθεται δωρεάν και μπορεί κανείς να τα εγκαταστήσει σε έναν εξυπηρετητή που υποστηρίζει βάση δεδομένων MySQL και τη γλώσσα προγραμματισμού PHP και να το χρησιμοποιήσει για τη δημιουργία μιας ιστοσελίδας.

2.2.3.1 CMS Ανοικτού Κώδικα

Τα συστήματα ανοικτού κώδικα είναι αυτά που επιτρέπουν αλλαγές στον πηγαίο κώδικα του συστήματος, είναι δωρεάν χωρίς κανένα απολύτως κόστος και υποστηρίζονται από κοινότητες χρηστών και προγραμματιστών, οι οποίες δημιουργούν και προσφέρουν τα διάφορα plug-ins και τα templates. Τα λογισμικά ανοικτού κώδικα θεωρούνται βασικά κομμάτι της αγοράς και έχουν πλέον καθιερωθεί. Υπάρχει υποστήριξη από μεγάλες εταιρείες οι οποίες θεωρούν τα λογισμικά ανοικτού κώδικα την πλέον βιώσιμη επιλογή. Τα λογισμικά αυτά έχουν βελτιωθεί πάρα πολύ που μπορούν πλέον να συγκριθούν με τα αντίστοιχα τους εμπορικά πακέτα. Εάν διαπιστωθεί ένα πρόβλημα στο σύστημα μπορούμε να το λύσουμε μόνοι μας αφού έχουμε πλήρη πρόσβαση στον κώδικα του συστήματος ή να ζητήσουμε βοήθεια από τις κοινότητες υποστήριξης, όπου συνήθως μέσα σε λίγες μέρες το πρόβλημα θα έχει λυθεί. Ενώ αντίστοιχα σε ένα εμπορικό σύστημα πρέπει να ενημερώσουμε τον κατασκευαστή και να περιμένουμε να το φτιάξει ή ακόμα να περιμένουμε την επόμενη έκδοση του συστήματος η οποία θα είναι ανανεωμένη και χωρίς προβλήματα. Μερικά από τα πλέον δημοφιλή Συστήματα Διαχείρισης Περιεχομένου είναι τα Strapi, Drupal, Joomla, Mambo και Wordpress.

2.2.4 Strapi

Το Strapi είναι ένα «headless» σύστημα διαχείρισης περιεχομένου (ΣΔΠ, Content Management System, CMS) ανοιχτού κώδικα βασισμένο στο Node.js - ένα περιβάλλον που επιτρέπει την εκτέλεση κώδικα JavaScript εκτός των περιηγητών ιστού (browsers) για τους οποίους προοριζόταν αρχικά. Ο όρος «headless» βασίζεται στην ιδέα ύπαρξης μιας διεπαφής για την προσθήκη περιεχομένου και ενός RESTful API που είναι υπεύθυνο για την παροχή του περιεχομένου αυτού. Σε αντίθεση με τα παραδοσιακά ΣΔΠ (CMS), όπως είναι το WordPress, τα headless συστήματα δε διαθέτουν κάποιο μέσο, όπως μια ιστοσελίδα, για άμεση προβολή του περιεχομένου στον χρήστη. Ευθύνη των headless συστημάτων αποτελούν μόνο η αποθήκευση και η παροχή δομημένου περιεχομένου. Η ιδέα γύρω από τη δημιουργία του Strapi είναι να παρέχει τη δυνατότητα κατασκευής ενός πρακτικού, έτοιμου για χρήση API, σε λίγες ώρες αντί για ολόκληρες εβδομάδες, χάρη στο εύχρηστο περιβάλλον και το πλήρως διαμορφώσιμο API που διαθέτει. Μερικά από τα κύρια και καινοτόμα χαρακτηριστικά του Strapi είναι τα εξής:

- Πανεύκολη διαχείριση περιεχομένου: Μετά τον καθορισμό της δομής του περιεχομένου, η διαχείρισή του μπορεί να γίνει από οποιονδήποτε χρήστη, με ή χωρίς εμπειρία, χάρη στο εύχρηστο γραφικό περιβάλλον διαχείρισής του.
- Ευέλικτη δομή περιεχομένου: Το Strapi προσφέρει εκτός από τους βασικούς τύπους πεδίων, και άλλους πιο σύνθετους, όπως τύπο πολυμέσων, email και κωδικών πρόσβασης. Πιο έμπειροι χρήστες μπορούν ακόμη να κατασκευάσουν δικούς τους τύπους περιεχομένου ανάλογα με τις ανάγκες τους.
- Επαλήθευση και άδειες: Χάρη στην υποστήριξη ρόλων, η πρόσβαση μιας ομάδας σε κάποιο API endpoint μπορεί να χορηγηθεί ή να ανακληθεί πανεύκολα και γρήγορα.
- Διαμορφώσιμο API: Με πρόσβαση στα αρχεία κώδικα του project, το API που κατασκευάζεται μπορεί να προσφέρει στους χρήστες του περισσότερα και ιδιαίτερα χαρακτηριστικά.
- Μεταφόρτωση αρχείων: Υπάρχει υποστήριξη μεταφόρτωσης όλων των τύπων αρχείων, τόσο στον ίδιο διακομιστή που φιλοξενείται το Strapi όσο και σε εξωτερικούς παρόχους, όπως AWS S3, Cloudflare, κλπ.
- Βάσεις δεδομένων: Είτε πρέπει να χρησιμοποιηθεί σχεσιακή είτε μη σχεσιακή βάση δεδομένων, το Strapi έχει κατασκευαστεί να είναι συνεργάσιμο και με τους δύο τύπους. Προσφέρεται υποστήριξη για τις βάσεις δεδομένων MongoDB, PostgreSQL, MySQL, MariaDB και για την SQLite η οποία χρησιμοποιείται από προεπιλογή.

Ενσωματωμένο σύστημα email: Με κάθε καινούριο project, εγκαθίσταται από προεπιλογή και ένας διακομιστής SMTP (Simple Mail Transfer Protocol) για διαχείριση των email (εγγραφής, ανάκτησης κωδικού πρόσβασης, κλπ.), χωρίς την ανάγκη επιπλέον κώδικα.

2.3 Tools and compilers

2.3.1 Node Package Manager (NPM)

Το NPM (Node Package Manager) είναι ένας διαχειριστής πακέτων για τη γλώσσα προγραμματισμού JavaScript. Είναι ο προεπιλεγμένος διαχειριστής πακέτων για το περιβάλλον εκτέλεσης JavaScript Node.js (JavaScript runtime-environment). Αποτελείται από έναν client γραμμής εντολών, που ονομάζεται επίσης NPM, και μια ηλεκτρονική βάση δεδομένων τόσο δημόσιων όσο και ιδιωτικών (private paid for) πακέτων. Τα αρχεία είναι προσβάσιμα από τον client και τα διαθέσιμα πακέτα μπορούν να αναζητηθούν μέσω της ιστοσελίδας του NPM, και υπεύθυνοι για αυτά είναι η NPM, Inc. Το NPM είναι γραμμένο εξ ολοκλήρου σε JavaScript και αναπτύχθηκε από τον Isaac Z. Schlueter με την έμπνευση από άλλα παρόμοια έργα όπως το PEAR (PHP). Ειδικότερα, είναι ένα online αποθετήριο για τη δημοσίευση έργων Node.js και ένα βοηθητικό πρόγραμμα γραμμής εντολών για αλληλεπίδραση με το εν λόγω αποθετήριο που βοηθά στην εγκατάσταση πακέτων, στη διαχείριση των εκδόσεων τους και στη διαχείριση εξάρτησης. Ένα πακέτο ουσιαστικά, είναι ένας φάκελος που περιέχει ένα πρόγραμμα το οποίο περιγράφεται από ένα αρχείο που ονομάζεται “package.json”. Το Σχήμα 5 δείχνει μια σύνοψη των εντολών χρήσης npm τόσο για την εγκατάσταση πακέτων όσο και την ενημέρωσή τους.

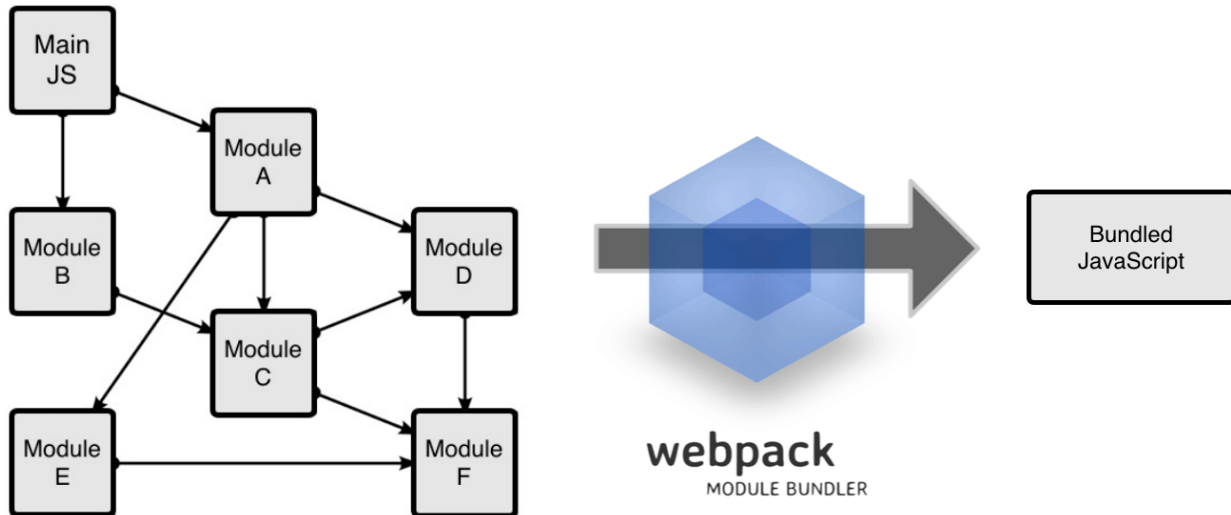
```
npm install (with no args, in package dir)
npm install [<@scope>/]<name>
npm install [<@scope>/]<name>@<tag>
npm install [<@scope>/]<name>@<version>
npm install [<@scope>/]<name>@<version range>
npm install <git-host>:<git-user>/<repo-name>
npm install <git repo url>
npm install <tarball file>
npm install <tarball url>
npm install <folder>
```

Σχήμα 9: Σύνταξη για εγκατάσταση npm/package.

2.3.2 Webpack

Το Webpack είναι ένα πακέτο στοιχείων JavaScript ανοιχτού κώδικα (opensource JavaScript module bundler). Κυρίως είναι ένας συνδυασμός στοιχείων για JavaScript, αλλά μπορεί να μετασχηματίσει και στοιχεία front-end όπως HTML, CSS και εικόνες αν συμπεριληφθούν οι αντίστοιχοι φορτωτές. Το Webpack παίρνει τα στοιχεία με εξαρτήσεις (module with dependencies) και παράγει στατικά στοιχεία που αντιπροσωπεύουν αυτά τα στοιχεία, δηλαδή λαμβάνει τις εξαρτήσεις και δημιουργεί ένα γράφημα εξάρτησης που επιτρέπει τους προγραμματιστές να χρησιμοποιούν με αρθρωτή προσέγγιση για τους σκοπούς ανάπτυξης εφαρμογών ιστού. Μπορεί να χρησιμοποιηθεί από το command line ή να ρυθμιστεί

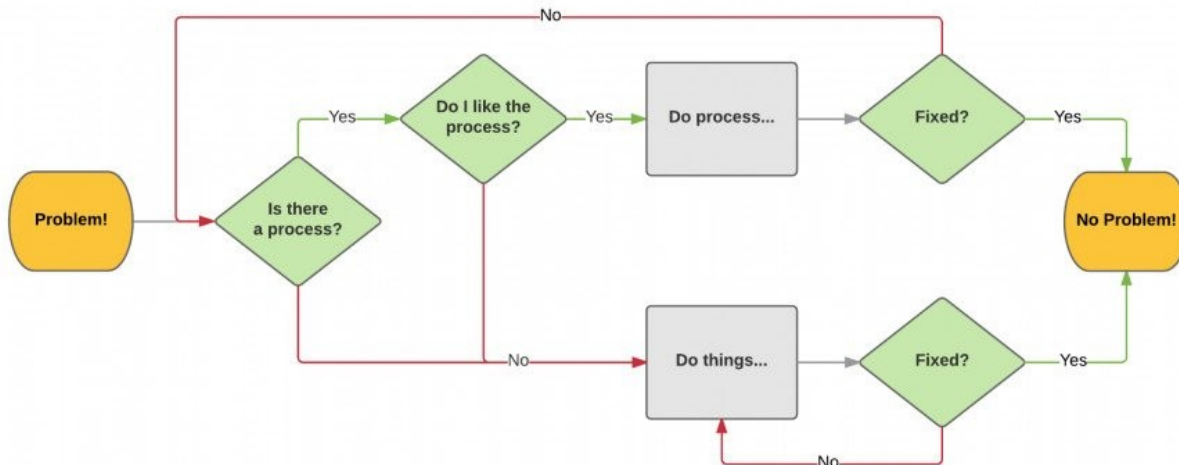
χρησιμοποιώντας ένα αρχείο ρυθμίσεων που ονομάζεται “webpack.config.js”. Είναι εξαιρετικά επεκτάσιμο μέσω κανόνων που επιτρέπουν στους προγραμματιστές να γράφουν custom tasks που θέλουν να εκτελέσουν όταν συνδυάζουν τα αρχεία μαζί. Η χρήση του Webpack απαιτεί το Node.js. Η React όπως και η HMR (Hot Module Replacement) βοήθησαν στη διάδοση του Webpack και σε άλλα περιβάλλοντα, όπως η Ruby στο Rails. Έτσι, πλέον το Webpack υπάρχει προ εγκατεστημένο (pre-build) στα περισσότερα frameworks λόγω του μεγάλου μεγέθους των βιβλιοθηκών που χρησιμοποιούνται για την τελική εφαρμογή. Στο Σχήμα 7 παρουσιάζονται οι λειτουργίες που εκτελούνται.



Σχήμα 10: Λειτουργίες του Webpack (workflow)

2.3.3 Babel

Η Babel είναι ένας μεταγλωττιστής (compiler) ανοιχτού κώδικα JavaScript που χρησιμοποιείται κυρίως για την μετατροπή του κώδικα των εκδόσεων ECMAScript 2015 και άνω, σε μια συμβατή με το παρελθόν έκδοση της JavaScript, η οποία μπορεί να χρησιμοποιηθεί από παλαιότερους μηχανισμούς της. Είναι ένα δημοφιλές εργαλείο για τη χρήση των νέων δυνατοτήτων της γλώσσας προγραμματισμού JavaScript. Μέσω του Babel, οι προγραμματιστές μπορούν να χρησιμοποιήσουν νέες λειτουργίες γλώσσας JavaScript για να μετατρέψουν τον πηγαίο κώδικα σε εκδόσεις JavaScript που επεξεργάζονται οι εξελιγμένες μηχανές αναζήτησης. Παραδείγματα αυτών των νέων λειτουργιών είναι οι λειτουργίες βέλους, οι οποίες καθορίζονται στο ES6 και μετατρέπονται σε κανονικές δηλώσεις λειτουργίας, καθώς και η μη τυποποιημένη σύνταξη της JavaScript (JSX) μπορεί επίσης να μετασχηματιστεί. Το Σχήμα 8 προβάλλει τον τρόπο με τον οποίο εκτελεί τις λειτουργίες.



Σχήμα 11: Λειτουργίες του Babel (workflow)

2.4 API

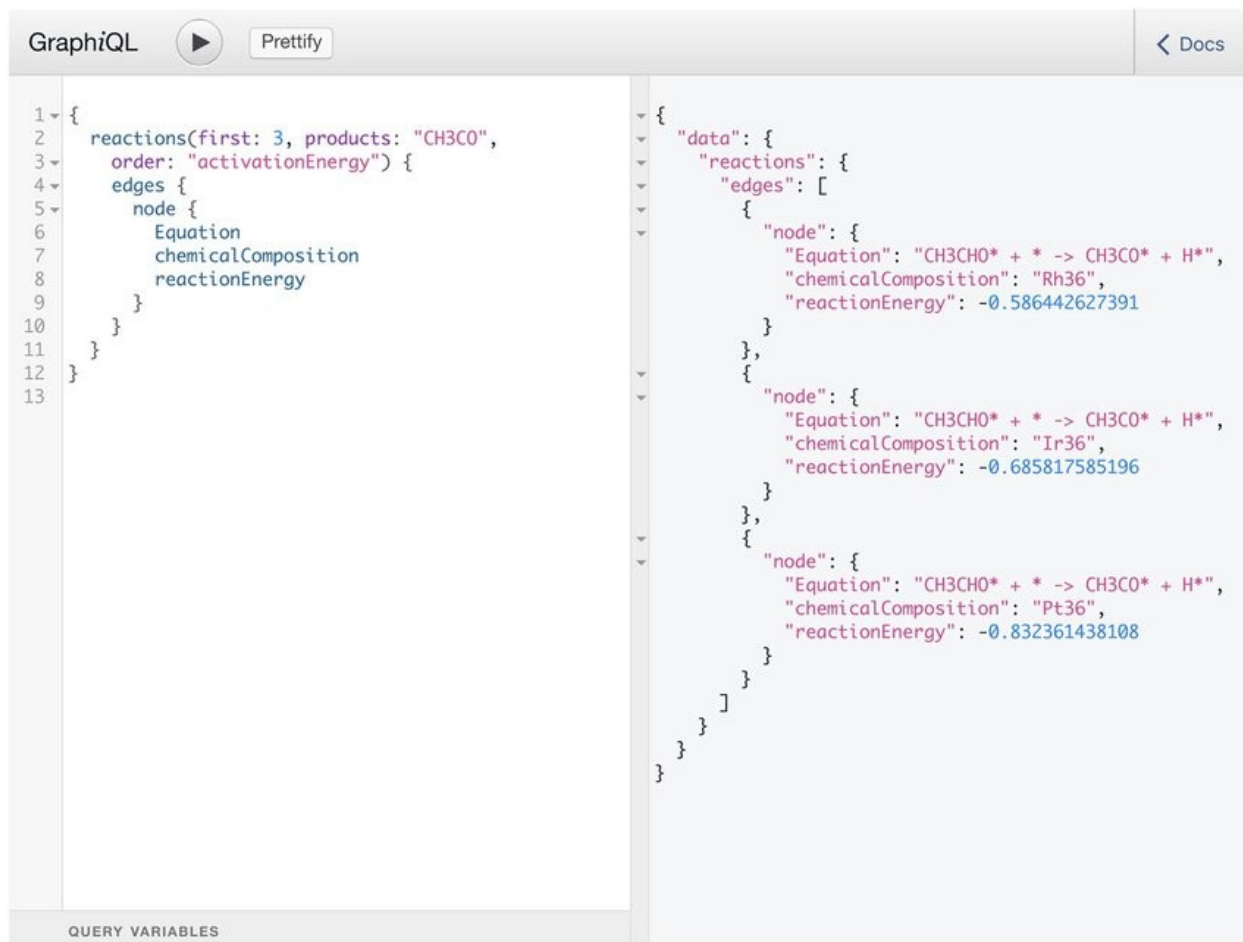
2.4.1 Rest API

Το REST (Representational State Transfer) είναι ένα αρχιτεκτονικό στυλ λογισμικού το οποίο ορίζει ένα σύνολο από αρχές σχεδίασης διαδικτυακών υπηρεσιών. Όπως είναι φανερό και από το όνομά του, σε μια REST διαδικτυακή υπηρεσία υπάρχει ένας διακομιστής ο οποίος επιστρέφει στον πελάτη μια αναπαράσταση της κατάστασης της ζητούμενης πληροφορίας. Οι διαδικτυακές υπηρεσίες που συμμορφώνονται σύμφωνα με τη τεχνολογία REST ονομάζονται «RESTful» και παρέχουν δια λειτουργικότητα μεταξύ συστημάτων υπολογιστών στο διαδίκτυο. Συνεπώς, οι RESTful διεπαφές προγραμματισμού εφαρμογών (API) βασίζονται στη τεχνολογία REST και χρησιμοποιούνται για την επικοινωνία μεταξύ εφαρμογών. Η επικοινωνία βασίζεται στο πρωτόκολλο HTTP (Hypertext Transfer Protocol). Συγκεκριμένα, η μέθοδος «GET» που ορίζει το πρωτόκολλο χρησιμοποιείται για την ανάκτηση δεδομένων, η μέθοδος «POST» για τη δημιουργία δεδομένων, η μέθοδος «PUT» για ενημέρωση και η μέθοδος «DELETE» για διαγραφή αυτών. Η αναπαράσταση των δεδομένων συνήθως γίνεται σε μορφή JSON (JavaScript Object Notation), αλλά υπάρχουν και περιπτώσεις που η αναπαράσταση γίνεται σε μορφή XML (Extensible Markup Language) ή HTML (Hypertext Markup Language). Και οι τρεις προηγούμενες μορφές δεδομένων αφορούν την κωδικοποίηση εγγράφων σε μορφή που είναι ταυτόχρονα κατανοητή από τον άνθρωπο και τους ηλεκτρονικούς υπολογιστές. Η επικοινωνία με τα API γίνεται μέσω των διάφορων «endpoint» που διαθέτουν. Πρακτικά, τα API endpoint είναι πολλά διαφορετικά URL (Uniform Resource Locator), χάρη στα οποία γίνεται διαχωρισμός μεταξύ των πληροφοριών που μπορεί να προσφέρει το API. Ένα API μπορεί να διαθέτει μόνο ένα ή εκατοντάδες endpoint. Ανάλογα με το ποια είναι η ζητούμενη πληροφορία, το αίτημα πρέπει να προορίζεται προς το ένα συγκεκριμένο που έχει καθοριστεί να επιστρέφει την πληροφορία αυτή.

2.4.2. GraphQL

Η GraphQL είναι μία query γλώσσα για API. Μια εναλλακτική του RESTful API. Αναπτύχθηκε από το Facebook για εσωτερική χρήση το 2012, ενώ το 2015 έγινε public release. Το 2018 το project “τρέχει” από το GraphQL Foundation, κάτω από το Linux Foundation.

Η GraphQL λύνει κάποια από τα προβλήματα που έχει το RESTful API, όπως τα πολλά μπρος πίσω, τα πολλαπλά calls για να πάρει κανείς πολύπλοκα sets δεδομένων. Είναι επίσης πολύ πιο ευέλικτη στις όποιες μελλοντικές αλλαγές μπορεί να γίνουν στο API.



The image shows a screenshot of the GraphQL Playground interface. On the left, a query is entered in a text area with line numbers 1 through 13. The query is:

```
1 {
2   reactions(first: 3, products: "CH3CO",
3     order: "activationEnergy") {
4     edges {
5       node {
6         Equation
7         chemicalComposition
8         reactionEnergy
9       }
10    }
11  }
12 }
13
```

 Above the query area are a play button and a 'Prettify' button. On the right, the JSON response is displayed, showing a list of three reaction nodes with their respective equations, chemical compositions, and reaction energies. The response is:

```
{
  "data": {
    "reactions": {
      "edges": [
        {
          "node": {
            "Equation": "CH3CHO* + * -> CH3CO* + H*",
            "chemicalComposition": "Rh36",
            "reactionEnergy": -0.586442627391
          }
        },
        {
          "node": {
            "Equation": "CH3CHO* + * -> CH3CO* + H*",
            "chemicalComposition": "Ir36",
            "reactionEnergy": -0.685817585196
          }
        },
        {
          "node": {
            "Equation": "CH3CHO* + * -> CH3CO* + H*",
            "chemicalComposition": "Pt36",
            "reactionEnergy": -0.832361438108
          }
        }
      ]
    }
  }
}
```

 At the bottom of the interface, there is a section for 'QUERY VARIABLES'.

Σχήμα 12: Παράδειγμα GraphQL query.

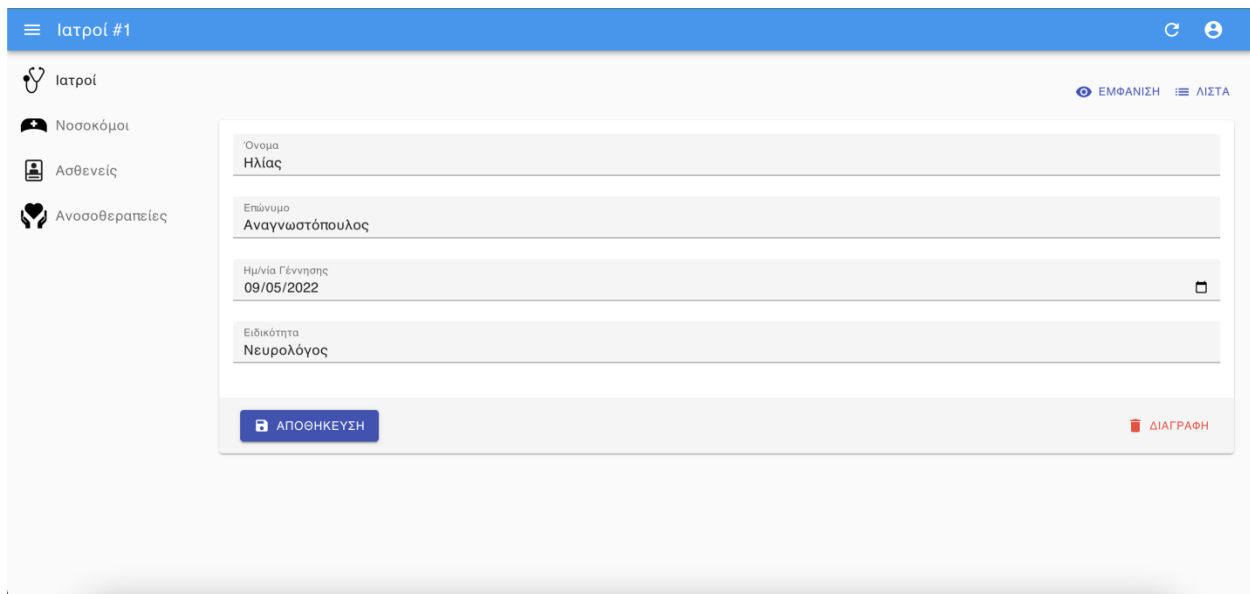
3. ΛΕΙΤΟΥΡΓΙΑ ΕΦΑΡΜΟΓΗΣ

Η εφαρμογή απευθύνεται σε χρήση εντός νοσοκομειακού χώρου. Συγκεκριμένα η χρήση της προορίζεται για την καταγραφή των ασθενών που υπόκεινται σε ανοσοθεραπείες, τον αριθμό των δόσεων που θα λάβουν, τους ιατρούς που θα τους αναλάβουν καθώς και τους νοσοκόμους.

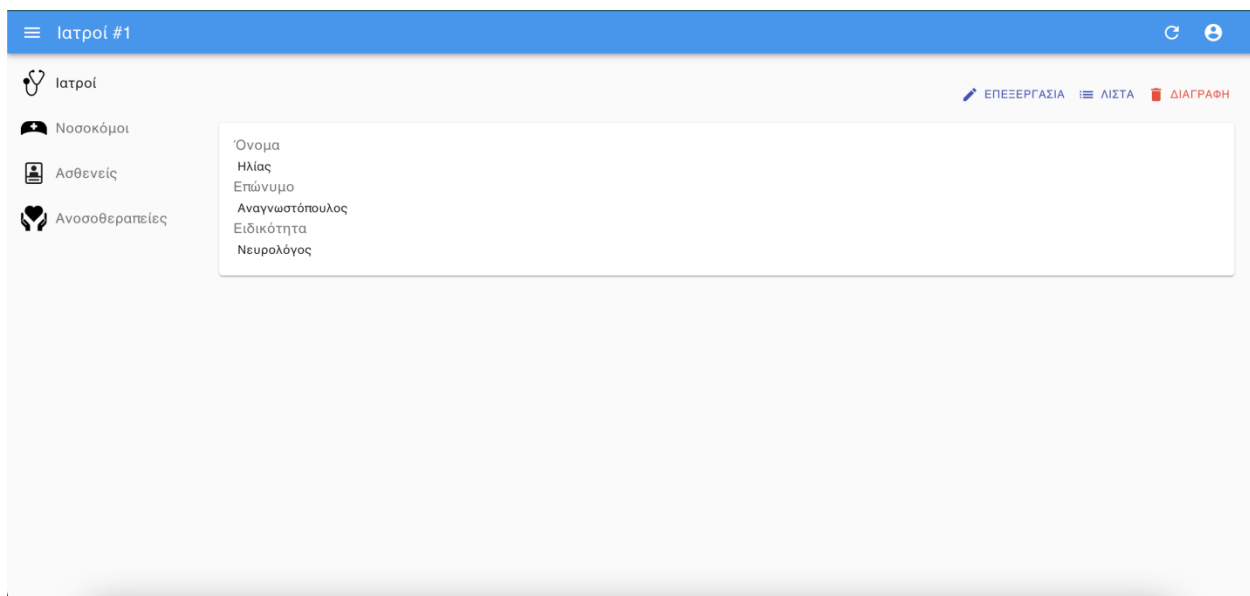
3.1 Ενότητα Ιατροί

<input type="checkbox"/>	Όνομα	Επώνυμο	Ειδικότητα	
<input type="checkbox"/>	Ηλίας	Αναγνωστόπουλος	Νευρολόγος	
<input type="checkbox"/>	Ανδρέας	Γλαράκης	Καρδιολόγος	
<input type="checkbox"/>	Ελευθερία	Σταματοπούλου	Γαστρεντερολόγος	
<input type="checkbox"/>	Παναγιώτης	Γιατρόπουλος	Γαστρεντερολόγος	
<input type="checkbox"/>	Μαργαρίτα	Γούτου	Παιδίατρος	
<input type="checkbox"/>	Γιάννης	Πετρόπουλος	Παθολόγος	
<input type="checkbox"/>	Φώτης	Σταματοπούλος	Παθολόγος	

Εικόνα 1: Λίστα Ιατρών

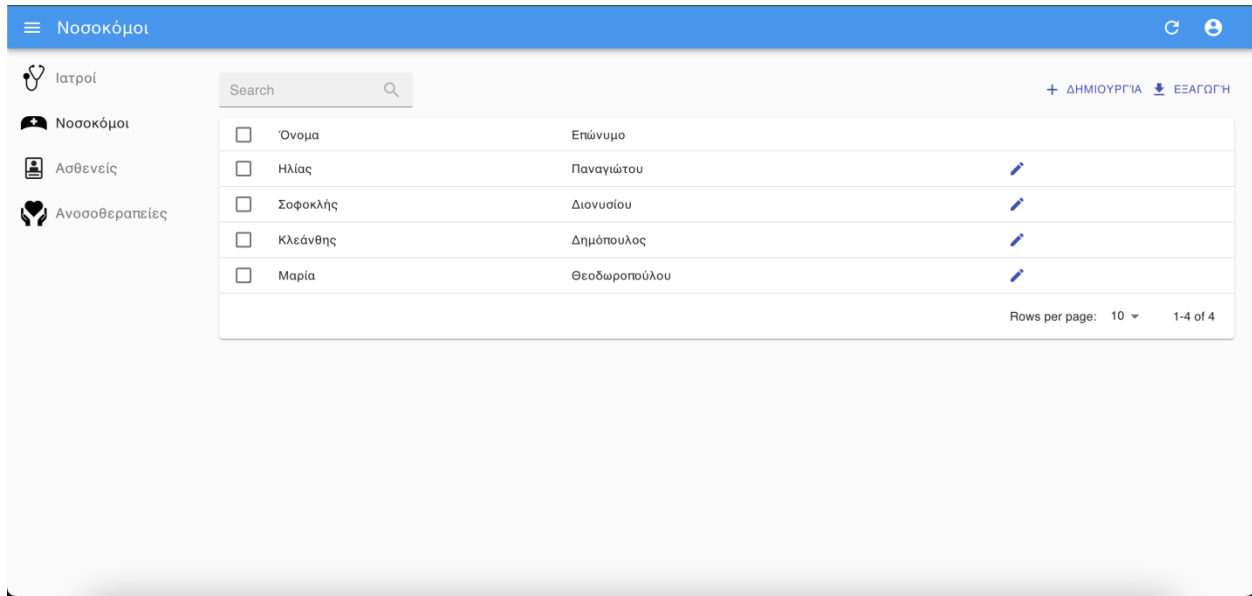


Εικόνα 2: Δημιουργία-Επεξεργασία

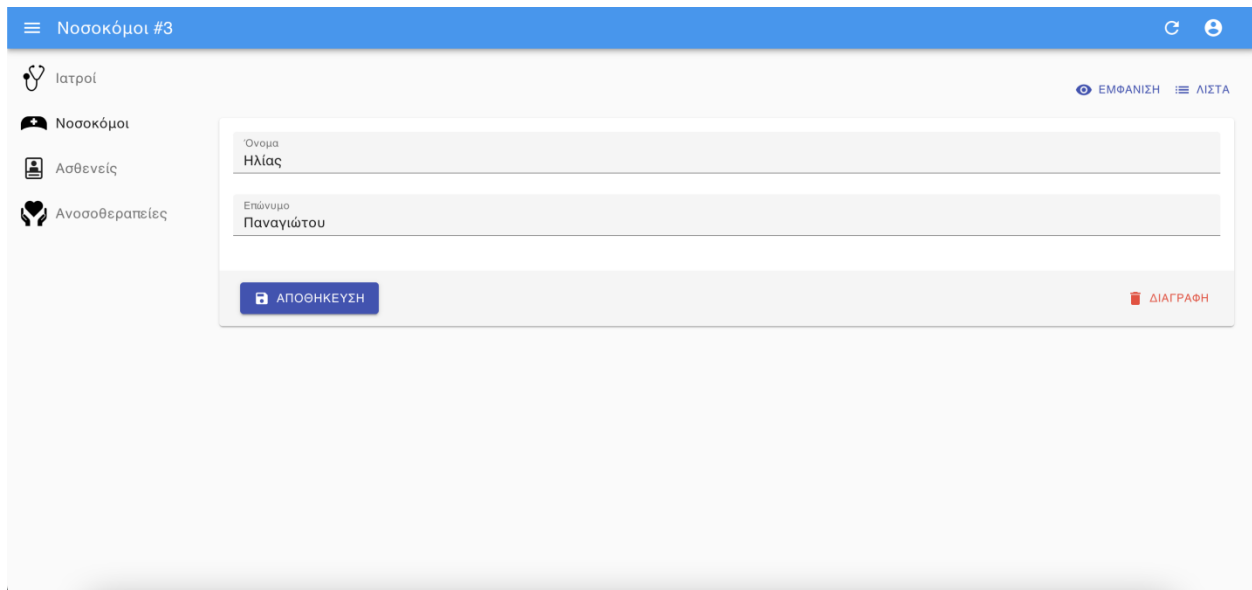


Εικόνα 3: Εμφάνιση

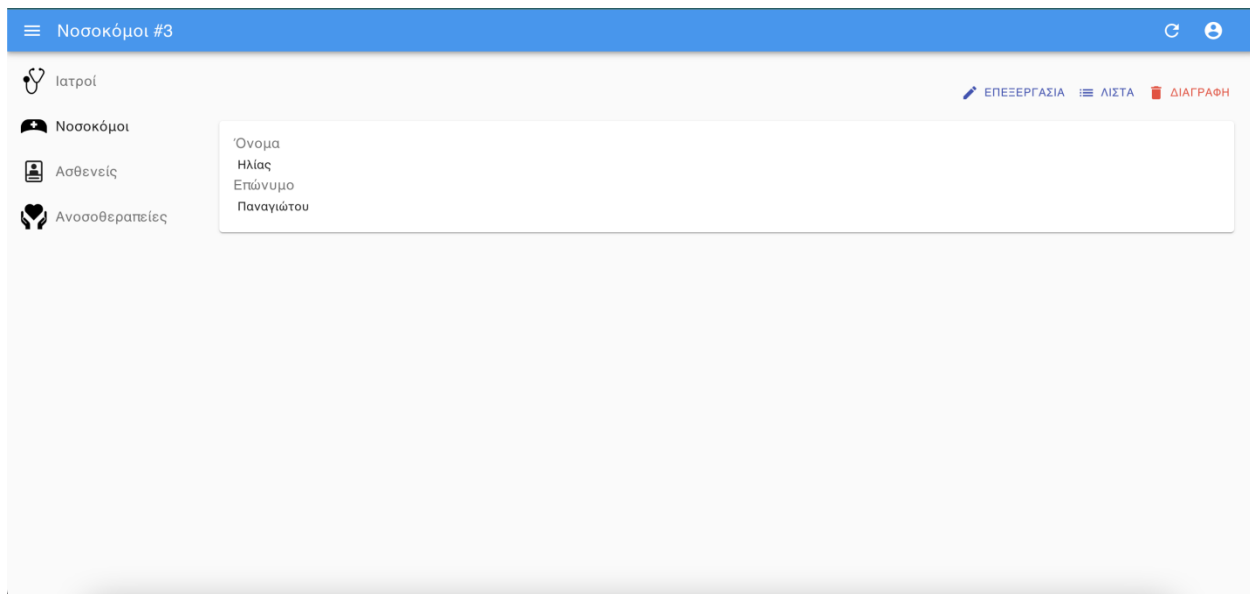
3.2 Ενότητα Νοσοκόμοι



Εικόνα 4: Λίστα

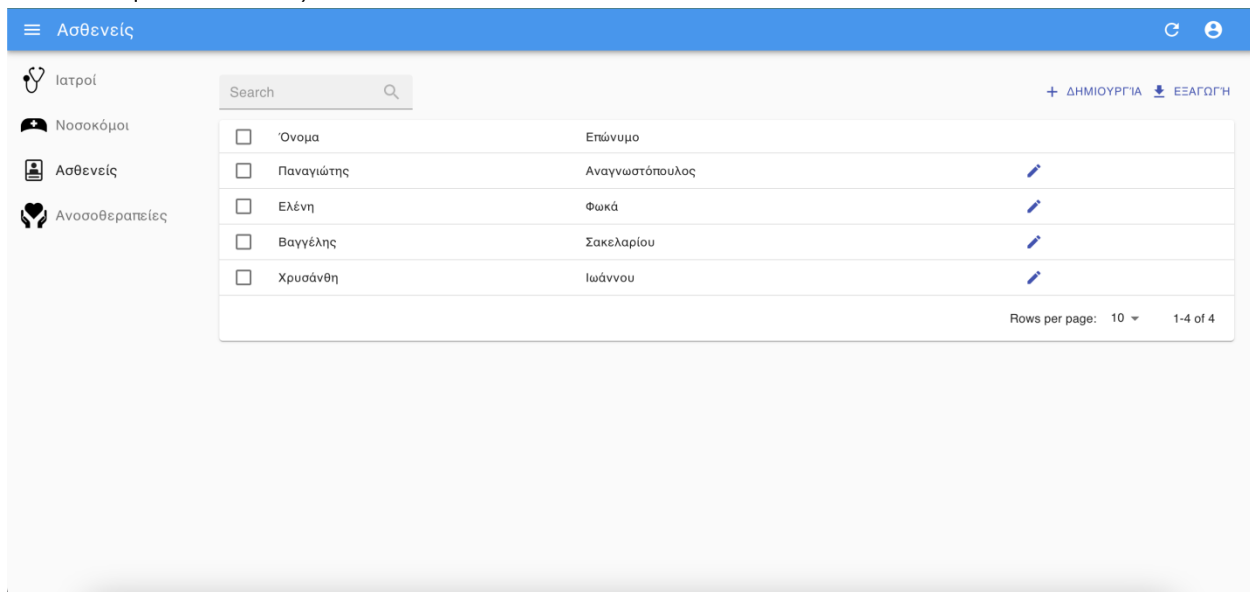


Εικόνα 5: Δημιουργία-Επεξεργασία

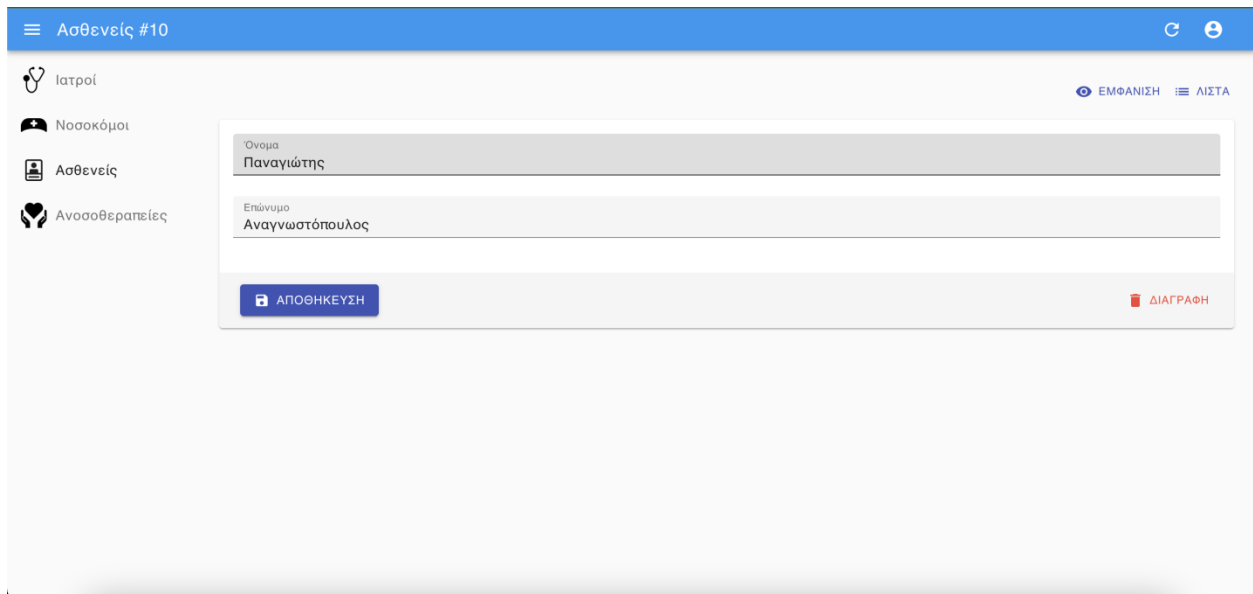


Εικόνα 6: Εμφάνιση

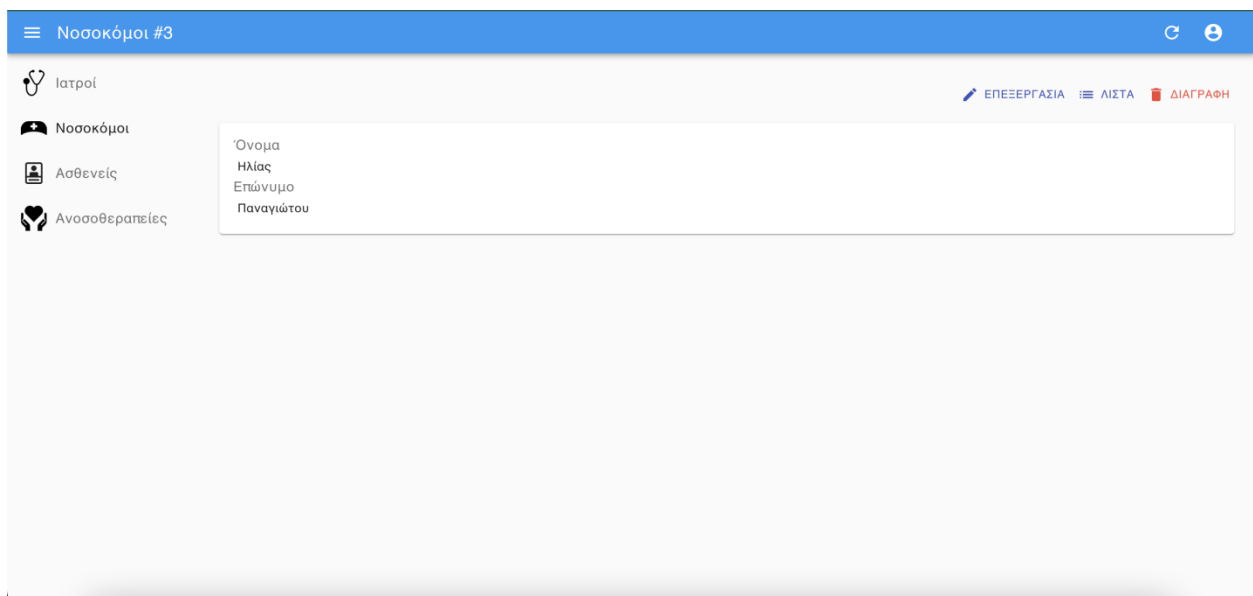
3.3 Ενότητα Ασθενείς



Εικόνα 7: Λίστα



Εικόνα 8: Δημιουργία-Επεξεργασία



Εικόνα 9: Εμφάνιση

3.4 Ενότητα Ανοσοθεραπεία

Η ενότητα αυτή περιέχει τόσο τις δόσεις όσο και τις παρενέργειες που προκαλεί η κάθε δόση.

Ανοσοθεραπείες

Ιατροί
 Νοσοκόμεοι
 Ασθενείς
 Ανοσοθεραπείες

Search

+ ΔΗΜΙΟΥΡΓΙΑ

<input type="checkbox"/>	Ημ/νία Έναρξης	Υπευθυνος Ιατρός	Ασθενής	Δόσεις	
<input type="checkbox"/>	22/09/2022	Ελευθερία Σταματοπούλου	Ελένη Φωκά	2	<input type="button" value="ΕΔΙΤΑΞΕΙΣ"/>
<input type="checkbox"/>	07/09/2022	Ηλίας Αναγνωστόπουλος	Χρυσάνθη Ιωάννου	2	<input type="button" value="ΕΔΙΤΑΞΕΙΣ"/>

Rows per page: 10 1-2 of 2

Εικόνα 10: Λίστα

Ανοσοθεραπείες #24

Ιατροί
 Νοσοκόμεοι
 Ασθενείς
 Ανοσοθεραπείες

ΕΜΦΑΝΙΣΗ ΛΙΣΤΑ

Ημ/νία Έναρξης
 22/09/2022

Ασθενής
 Ελένη Φωκά

Ιατρός
 Ελευθερία Σταματοπούλου

Αγωγή
 Κάποια αγωγή

Εικόνα 11: Δημιουργία-επεξεργασία ανοσοθεραπείας

Ανοσοθεραπείες #24

Δημιουργία δόσης

1

Αριθμός Έγκρισης 21321321321	Ημερομηνία 21/09/2022	Πορεία Νόσου Καλή
Παρατηρήσεις Καμία Παρατήρηση	Νοσοκόμος Σοφοκλής Διονυσίου	Αριθμός δόσης mg 1 x 500

Παρενέργειες

Ημερομηνία 24/09/2022	Όργανο Στομάχι/Έντερο	Σύμπτωμα Πόνος στο στομάχι	Άλλο Σύμπτωμα Φαγούρα
Ημερομηνία 27/09/2022	Όργανο Δέρμα	Σύμπτωμα Φαγούρα	Άλλο Σύμπτωμα Κανένα άλλο σύμπτωμα

ΠΡΟΣΘΗΚΗ ΠΑΡΕΝΕΡΓΕΙΑΣ

2

Αριθμός Έγκρισης 90321151321	Ημερομηνία 29/09/2022	Πορεία Νόσου Καλή
---------------------------------	--------------------------	----------------------

ΑΠΟΘΗΚΕΥΣΗ ΑΚΥΡΩΣΗ

Εικόνα 12: Δημιουργία-επεξεργασία δόσεων και παρενεργειών

Ανοσοθεραπείες #25

Ιατροί
Νοσοκόμοι
Ασθενείς
Ανοσοθεραπείες

Ημ/νία Έναρξης
2022-09-07
Φαρμακευτική Αγωγή
Κάποια αγωγή
Ασθενής
Χρυσάνθη Ιωάννου
Ιατροί
Ηλίας Αναγνωστόπουλος

Δόσεις

Αριθμός έγκρισης: 21321321321 Πορεία νόσου: Καλή	Ημερομηνία έναρξης: 2022-09-21 Παρατηρήσεις: Καμία		
Καμία παρενέργεια			
Αριθμός έγκρισης: 21321321322 Πορεία νόσου: Καλή	Ημερομηνία έναρξης: 2022-10-01 Παρατηρήσεις: Καμία Παρατήρηση		
Ημερομηνία	Όργανο	Σύμπτωμα	Άλλο
2022-09-11	Μάτια	Θολερότητα στην όραση	Κανένα άλλο σύμπτωμα

Εικόνα 12: Εμφάνιση

4. ΕΓΚΑΤΑΣΤΑΣΗ ΕΦΑΡΜΟΓΗΣ

Αρχικά θα πρέπει να υπάρχει εγκατεστημένη κάποια έκδοση node (κατά προτίμηση η 14.17.0). Επιπροσθέτως θα πρέπει να τρέχει ένας MySQL server τοπικά στον υπολογιστή μας.

1^ο βήμα: Κατεβάζουμε το frontend κώδικα και τρέχουμε τις εντολές “yarn install” και “yarn start”.

2^ο βήμα: Κατεβάζουμε το backend κώδικα και στο αρχείο database.js (project root/config/database.js) θα χρειαστεί να αλλάξουμε τα configurations ούτως ώστε να συνδέσουμε το δικό μας MySQL server.

3^ο βήμα: Τέλος τρέχουμε τις εντολές “yarn build” και “yarn develop”.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Babel. <https://babeljs.io/docs/en/>. [Accessed February-2020].
2. NPM Node Package Manager. <https://docs.npmjs.com/about-npm/>. [Accessed February-2020].
3. Webpack. <https://www.freecodecamp.org/news/an-intro-to-webpack-what-it-is-and-how-to-use-it-8304ecdc3c60/>. [Accessed February-2020].
4. G. Blokdyk. Full Stack JavaScript A Complete Guide - 2020 Edition. 5STARCOOKS, 2019.
5. G. Blokdyk. Web Development Team A Complete Guide - 2020 Edition. Emereo Pty Limited, 2019.
6. V. Bojinov, D. Herron, and D. Resende. Node.js Complete Reference Guide: Discover a more sustainable way of writing software with high levels of reusability and collaboration using Node.js. Packt Publishing, 2018
7. D. Cederholm and J. Zeldman. CSS3 for Web Designers. Book Apart. A Book Apart, 2014
8. B. Sanders. Smashing HTML5. Smashing magazine book series. Wiley, 2010.
9. Douglas Crockford. JavaScript: The Good Parts. OReilly, 2008