



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

**Ανάπτυξη διαχειριστικού Web Application, για εφαρμογή αγοράς
ομαδικών δώρων**

Διπλωματική

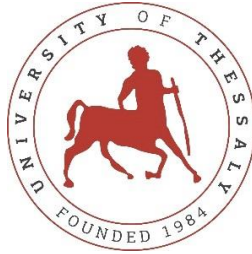
Εργασία

Τάσσης Αλέξανδρος

Επιβλέπων:

Σταμούλης Γεώργιος

Μάρτιος 2023



UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

**Development of a vendor - side web application for group purchased
gifts**

Diploma Thesis

Tassis Alexandros

Supervisor: George Stamoulis

March 2023

Εγκρίνεται από την Επιτροπή Εξέτασης:

Επιβλέπων

Σταμούλης Γεώργιος

Καθηγητής, Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών
Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

Μέλος

Ευμορφόπουλος Νέστωρ

Αναπληρωτής Καθηγητής, Τμήμα Ηλεκτρολόγων Μηχανικών και
Μηχανικών Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

Μέλος

Σταμπουλής Γεώργιος

Αναπληρωτής Καθηγητής, Τμήμα Οικονομικών Επιστημών,
Πανεπιστήμιο Θεσσαλίας

ΥΠΕΥΘΥΝΗ ΔΗΛΩΣΗ ΠΕΡΙ ΑΚΑΔΗΜΑΪΚΗΣ ΔΕΟΝΤΟΛΟΓΙΑΣ ΚΑΙ ΠΝΕΥΜΑΤΙΚΩΝ ΔΙΚΑΙΩΜΑΤΩΝ

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ρητά ότι η παρούσα διπλωματική εργασία, καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας, αποτελούν αποκλειστικά προϊόν προσωπικής μου εργασίας, δεν προσβάλλουν οποιασδήποτε μορφής δικαιώματα διανοητικής ιδιοκτησίας, προσωπικότητας και προσωπικών δεδομένων τρίτων, δεν περιέχουν έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον και πληρούν τους κανόνες της επιστημονικής παράθεσης. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή/και πηγές άλλων συγγραφέων αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Δηλώνω επίσης ότι τα αποτελέσματα της εργασίας δεν έχουν χρησιμοποιηθεί για την απόκτηση άλλου πτυχίου. Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής.

Ο Δηλών,

Τάσσης Αλέξανδρος

08/03/2023

DISCLAIMER ON ACADEMIC ETHICS AND INTELLECTUAL PROPERTY RIGHTS

Being fully aware of the implications of copyright laws, I expressly state that this diploma thesis, as well as the electronic files and source codes developed or modified in the course of this thesis, are solely the product of my personal work and do not infringe any rights of intellectual property, personality and personal data of third parties, do not contain work / contributions of third parties for which the permission of the authors / beneficiaries is required and are not a product of partial or complete plagiarism, while the sources used are limited to the bibliographic references only and meet the rules of scientific citing. The points where I have used ideas, text, files and / or sources of other authors are clearly mentioned in the text with the appropriate citation and the relevant complete reference is included in the bibliographic references section. I also declare that the results of the work have not been used to obtain another degree. I fully, individually and personally undertake all legal and administrative consequences that may arise in the event that it is proven, in the course of time, that this thesis or part of it does not belong to me because it is a product of plagiarism.

The Declarant,

Tassis Alexandros
08/03/2023

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου, για όλη τη στήριξη και την καθοδήγηση κατά τη διάρκεια συγγραφής της παρούσας διπλωματικής εργασίας. Επίσης, εκτιμώ αμέριστα τη βοήθεια του Αναπληρωτή Καθηγητή κύριου Σταμπουλή Γεώργιου, για την εμπιστοσύνη που μου έδειξε στην υλοποίηση της ιδέας, αλλά και για την καθοδήγησή του στο πλάνο και στον σχεδιασμό της εφαρμογής.

Επιπλέον, θερμές ευχαριστίες στους συναδέλφους μου Μέμου Ελβίρα και Μπανταβάνη Χρήστο, όπου η συνεργασία μας ήταν ένα συνεχές ταξίδι γνώσης.

Τέλος, δε θα μπορούσα να παραλείψω να ευχαριστήσω την οικογένειά μου που στάθηκε αρωγός σε όλη τη διάρκεια όχι μόνο της διπλωματικής εργασίας, αλλά ολόκληρης της φοιτητικής μου ζωής.

Διπλωματική Εργασία

Ανάπτυξη Web Application, για την ομαδική αγορά δώρων

Τάσσης Αλέξανδρος

Περίληψη

Στην σημερινή εποχή, η τεχνολογία αποτελεί αναπόσπαστο κομμάτι της καθημερινής μας ζωής, φέρνοντας την επανάσταση στα μέσα με τα οποία επικοινωνούμε, εργαζόμαστε και δεχόμαστε πληροφορίες. Από τα έξυπνα τηλέφωνα έως τους ηλεκτρονικούς υπολογιστές, η τεχνολογία καθιστά πλέον άμεση και εύκολη την επαφή με όλο τον υπόλοιπο κόσμο. Τα ηλεκτρονικά καταστήματα ή οι πλατφόρμες ηλεκτρονικών αγορών έχουν επίσης επιφέρει αλλαγές στον τρόπο που πραγματοποιούμε τις αγορές μας επιτρέποντας την απόκτηση προϊόντων και υπηρεσιών από την άνεση του σπιτιού μας.

Μέσω αυτού το εμπόριο αποκτά ακόμα μεγαλύτερη προσβασιμότητα αλλά επίσης αυξάνεται ραγδαία και η ποικιλία για τους αγοραστές.

Αφού λοιπόν η ενσωμάτωση της τεχνολογίας στο εμπόριο σαφώς βελτιώνει το βιοτικό μας επίπεδο, ο σκοπός της παρούσας διπλωματικής είναι η ανάπτυξη του διαχειριστικού μέλους μιας εφαρμογής όπου τα συμβεβλημένα σε αυτή καταστήματα, πωλούν τα προϊόντα τους με τη δυνατότητα συνδρομής περισσότερων από ενός ατόμου για την ολοκλήρωση του ποσού του προϊόντος. Δημιουργήθηκε μέσω της τεχνολογίας MERN STACK που αναλύεται παρακάτω, μαζί με άλλα εργαλεία που χρησιμοποιήθηκαν, ενώ αναφέρονται και πιθανές μελλοντικές επεκτάσεις για την εφαρμογή μας.

Λέξεις-κλειδιά:

Εφαρμογή Ιστού, Βάση δεδομένων, MERN STACK, MongoDB, Express, React, Node, JavaScript, Axios.

Diploma Thesis

Development of a web application for buying presents

Tassis Alexandros

Abstract

In today's age, technology is an integral part of our daily lives, revolutionizing the means by which we communicate, work and receive information. From smart phones to computers, technology now makes it instant and easy to connect with the rest of the world. Online stores or online shopping platforms have also brought about changes in the way we shop by allowing us to acquire products and services from the comfort of our homes.

Through this trade becomes even more accessible, but also the variety for buyers increases rapidly.

So since the integration of technology in trade clearly improves our standard of living, the purpose of this thesis is to develop the management member of an application where the stores contracted to it, sell their products with the possibility of the assistance of more than one person for the completion of the product amount. It was created using MERN STACK technology which is discussed below, along with other tools used, and possible future extensions for our application are also mentioned.

Keywords:

Data Base, Axios, Express, JavaScript, MERN STACK, MongoDB, Web Application, React.

Περιεχόμενα

Ευχαριστίες	4
Περίληψη	5
Abstract	6
Συνομογραφίες	9
Κεφάλαιο 1 Εισαγωγή	10
1.1. Αντικείμενο της διπλωματικής	10
1.2. Οργάνωση του τόμου	10
Κεφάλαιο 2 – Web application	12
2.1. Διαφορές Back - end και front - end development	12
2.2 Front-End	13
2.2.1 Οι γλώσσες HTML + CSS (HyperText Markup Language + Cascading Style Sheets)	13
2.2.2. React	15
2.2.3. AXIOS.....	17
2.3. Back-End.....	18
2.3.1 JavaScript	18
2.3.2. Node JS.....	19
2.3.3. Express	21
2.3.4. JSON	22
Κεφάλαιο 3 – Βάσεις δεδομένων	23
3.1. Διαφορές SQL με NoSQL (Σχεσιακές και μη Σχεσιακές βάσεις δεδομένων)	23
3.1.1. SQL βάσεις	23
3.1.2. NO SQL βάσεις	24
3.2. MongoDB	25
3.3. Επικοινωνία με τις βάσεις δεδομένων	26
Κεφάλαιο 4 – Η εφαρμογή	28
4.1. Μέσα ανάπτυξης εφαρμογής	29
4.2. Set Up.....	31
4.3. Μοντέλα Βάσεις δεδομένων (Models)	32
4.4. Routes	36
4.5. Upload JSON αρχείου.....	38
4.6. User interface (Διεπαφή Χρήστη).....	39
4.7. Προσθήκη νέας κατηγορίας.....	49
Κεφάλαιο 5 – Συμπεράσματα και μελλοντικές επεκτάσεις	49
5.1 Συμπεράσματα.....	49

5.2 Μελλοντικές Επεκτάσεις.....	50
Βιβλιογραφία	51

Συντομογραφίες

JSON	JavaScript Object Notation
MERN	MongoDB Express React Nodejs
API	Application Programming Interface
I/O	Input/Output
JS	JavaScript
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
PHP	PHP: Hypertext Preprocessor
HTTP	HyperText Transfer Protocol
npm	Node Package Manager
URL	Uniform Resource Locator
SQL	Structured Query Language
NoSQL	Not only SQL
DOM	Document Object Model

Κεφάλαιο 1 Εισαγωγή

Αποτελεί κοινή παραδοχή πως, ιδιαίτερα μετά την πανδημία, ο τομέας του εμπορίου έχει αναπτύξει ραγδαία την ηλεκτρονική του μορφή. Σχεδόν κάθε κατάσταση έχει το δικό του ηλεκτρονικό κατάστημα, γνωστό ως e-shop, ενώ δεν είναι λίγα εκείνα που κατευθύνονται προς μεγάλους vendors, ώστε να προσελκύσουν περισσότερους πελάτες. Πλατφόρμες σαν αυτή παρέχουν τη δυνατότητα αναζήτησης προϊόντος, δημιουργίας προσωπικής λίστας δώρων (wishlist), δίχως όμως να καλύπτουν την ανάγκη αγορών ομαδικών δώρων.

1.1. Αντικείμενο της διπλωματικής

Η εφαρμογή Web που δημιουργήθηκε είναι ένα ολοκληρωμένο σύστημα διαχείρισης καταστημάτων που έχει σχεδιαστεί για να βοηθά τους ιδιοκτήτες να διαχειρίζονται εύκολα τα ηλεκτρονικά τους καταστήματα. Η ανάπτυξη αυτής της εφαρμογής Ιστού ακολουθεί τις ίδιες τεχνολογίες και δομή με μία διαφορετική εφαρμογή όπου οι χρήστες αγοράζουν ομαδικά δώρα στους φίλους τους. Ωστόσο, η δική μας εφαρμογή έχει διαφορετική εστίαση, η οποία είναι στη διαχείριση καταστήματος. Οι ιδιοκτήτες καταστημάτων μπορούν να χειρίζονται παραγγελίες, προϊόντα, να βλέπουν στατιστικά στοιχεία και να διαχειρίζονται τις συνολικές και τις τρέχουσες συναλλαγές τους σε μια κεντρική πλατφόρμα. Η διεπαφή χρήστη έχει σχεδιαστεί για να είναι φιλική, με καθαρή και οργανωμένη διάταξη που βοηθά τους ιδιοκτήτες καταστημάτων να διαχειρίζονται αποτελεσματικά τα καταστήματά τους. Με αυτήν την εφαρμογή, οι ιδιοκτήτες μπορούν να έχουν μεγαλύτερο έλεγχο στα ηλεκτρονικά τους καταστήματα και να διασφαλίζουν ότι οι πελάτες τους έχουν μια απρόσκοπτη εμπειρία αγορών.

1.2. Οργάνωση του τόμου

- Στο Κεφάλαιο 2 αναφέρεται ο ορισμός του web application καθώς και λεπτομέρειες για τις τεχνολογίες του front-end και του back-end.

- Στο Κεφάλαιο 3 αναλύονται οι τρεις τύποι βάσεων δεδομένων και η τεκμηρίωση για την επιλογή της βάσης που τελικά επιλέξαμε.
- Στο Κεφάλαιο 4 δίνονται αναλύεται η δομή του προγράμματος και παρουσιάζονται στιγμιότυπα από την εφαρμογή.
- Στο κεφάλαιο 5 συνοψίζονται τα συμπεράσματα και προτείνονται μελλοντικές επεκτάσεις της εφαρμογής μας.

Κεφάλαιο 2 – Web application

Ένα web application είναι ουσιαστικά μία εφαρμογή, πρόσβαση στην οποία αποκτούμε μέσω περιηγητή, δίχως να είναι η απαραίτητη η λήψη κάποιου αρχείου. Βρίσκονται σε κάποιο απομακρυσμένο εξυπηρετητή, με χαρακτηριστικά του την ευκολία στην πρόσβαση αλλά και στην χρήση. Συνήθεις μέθοδοι υλοποίησης τους είναι η HTML, CSS και JavaScript, γλώσσες δηλαδή που χρησιμοποιούνται συχνότερα στην κατασκευή ιστοσελίδων.

2.1. Διαφορές Back - end και front - end development

Το web development βασίζεται τόσο στο back - end development, δηλαδή στον κώδικα ο οποίος σχηματίζει ένα λειτουργικό περιβάλλον ιστοσελίδας, όσο και στο front - end development, δηλαδή το πως φαίνεται το περιεχόμενο που βλέπει ο τελικός χρήστης.

Βασική διαφορά των δύο, είναι ο εκάστοτε στόχος. Το back - end development στοχεύει μέσω του κώδικα στη διαχείριση των δεδομένων τα οποία **δεν είναι ορατά** στον χρήστη, αλλά του επιτρέπουν να διαχειριστεί το περιεχόμενο της ιστοσελίδας του.

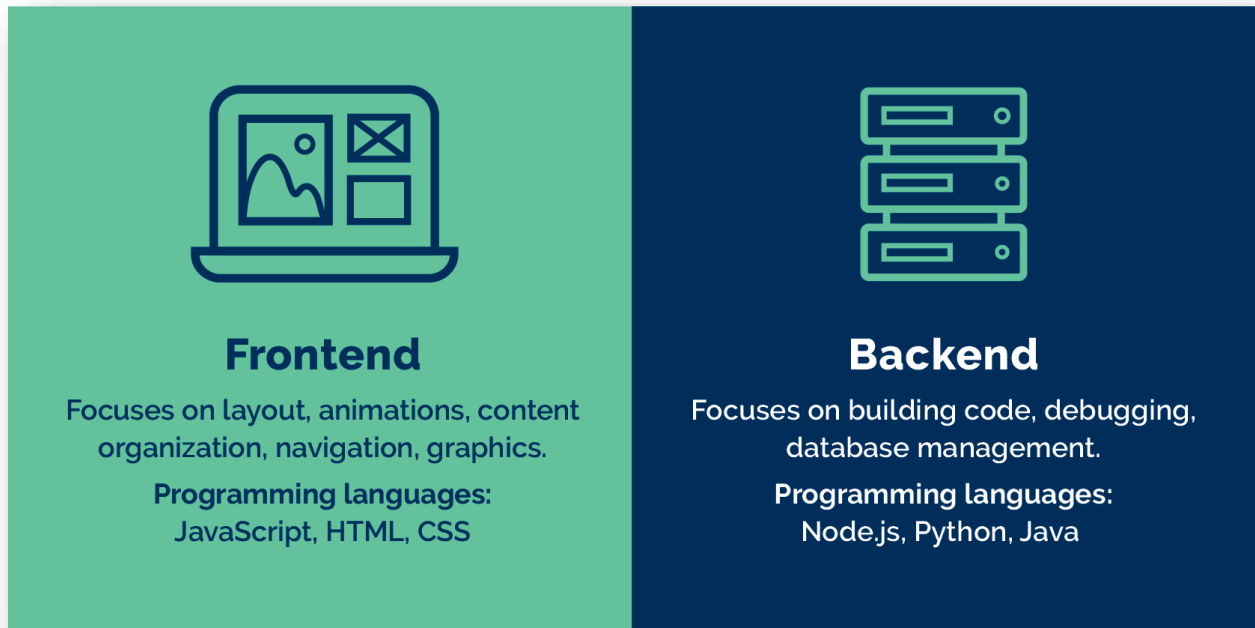
Αντίθετα, το front - end development στοχεύει στην αρχιτεκτονική της ηλεκτρονικής πλατφόρμας, δηλαδή καθορίζοντας τη δομή της (περιεχόμενα, εικόνες κλπ.) δημιουργεί ένα ελκυστικό και χρηστικό site.

Εξίσου σημαντική διαφορά του back – end και του front – end development είναι οι γλώσσες προγραμματισμού που χρησιμοποιείται για το κάθε ένα. Ένας προγραμματιστής του τμήματος front – end ουσιαστικά ασχολείται με τον γραφικό – εικαστικό περιβάλλον της ιστοσελίδας αναπτύσσοντας το σε κώδικα. Οι 3 γλώσσες προγραμματισμού που είναι απαραίτητες για να επιτευχθεί αυτό είναι:

- HTML
- CSS
- JavaScript

Αντίθετα, η γλώσσα προγραμματισμού που θα χρησιμοποιήσει κατά βάση ένας προγραμματιστής του τμήματος back – end είναι η JavaScript, ενώ μπορούν να

χρησιμοποιηθούν και άλλες γλώσσες προγραμματισμού όπως η Python, η PHP και η RUBY.



Εικόνα 2.1. – Βασικές διαφορές Front - end και Back – end development

2.2 Front-End

2.2.1 Οι γλώσσες HTML + CSS (HyperText Markup Language + Cascading Style Sheets)

Η HTML (HyperText Markup Language, Γλώσσα Σήμανσης του Υπερκειμένου) αποτελεί μια από τις χαρακτηριστικότερες γλώσσες του web development, και πρόκειται για μια γλώσσα σήμανσης που αφορά τα δομικά στοιχεία της ιστοσελίδας [1]. Το κυριότερο χαρακτηριστικό της είναι τα στοιχεία tags (ετικέτες), τα οποία είναι υπεύθυνα για την μορφολογία και τη δόμηση της ιστοσελίδας, ενώ ανάλογα με την σύνταξή τους, μπορεί να γίνει εισαγωγή κουμπιού, εικόνας, βίντεο. Ο εκάστωτε browser που χρησιμοποιούμε, καλείτε να δημιουργήσει και να παρουσιάσει το περιεχόμενο κάθε ιστοσελίδας, λαμβάνοντας αρχεία html, και διαβάζοντας τα tags, γίνεται η προβολή του περιεχομένου τους.

Tag	Description
<html> ... </html>	Declares the Web page to be written in HTML
<head> ... </head>	Delimits the page's head
<title> ... </title>	Defines the title (not displayed on the page)
<body> ... </body>	Delimits the page's body
<h <i>n</i> > ... </h <i>n</i> >	Delimits a level <i>n</i> heading
 ... 	Set ... in boldface
<i> ... </i>	Set ... in italics
<center> ... </center>	Center ... on the page horizontally
 ... 	Brackets an unordered (bulleted) list
 ... 	Brackets a numbered list
 ... 	Brackets an item in an ordered or numbered list
 	Forces a line break here
<p>	Starts a paragraph
<hr>	Inserts a horizontal rule
	Displays an image here
 ... 	Defines a hyperlink

Εικόνα 2.2. – HTML Tags

Η CSS (Cascading Style Sheets), είναι μια γλώσσα υπεύθυνη για την εύμορφη παρουσίαση αρχείων που είναι γραμμένα σε markup γλώσσες όπως η HTML. Μας παρέχει πολλές περισσότερες επιλογές για την δομή, τα χαρακτηριστικά, τα χρώματα και την στοίχιση στην ιστοσελίδα μας. Διαχωρίζοντας την οπτική παρουσίαση ενός ιστότοπου από το περιεχόμενό του, το CSS διευκολύνει τη συντήρηση και την ενημέρωση του σχεδιασμού ενός ιστότοπου χωρίς να χρειάζεται να τροποποιήσετε τον κώδικα HTML. Το CSS δίνει επίσης τη δυνατότητα στους σχεδιαστές να δημιουργούν αποκριτικά σχέδια που μπορούν να προσαρμοστούν σε διαφορετικά μεγέθη οθόνης και συσκευές [2].

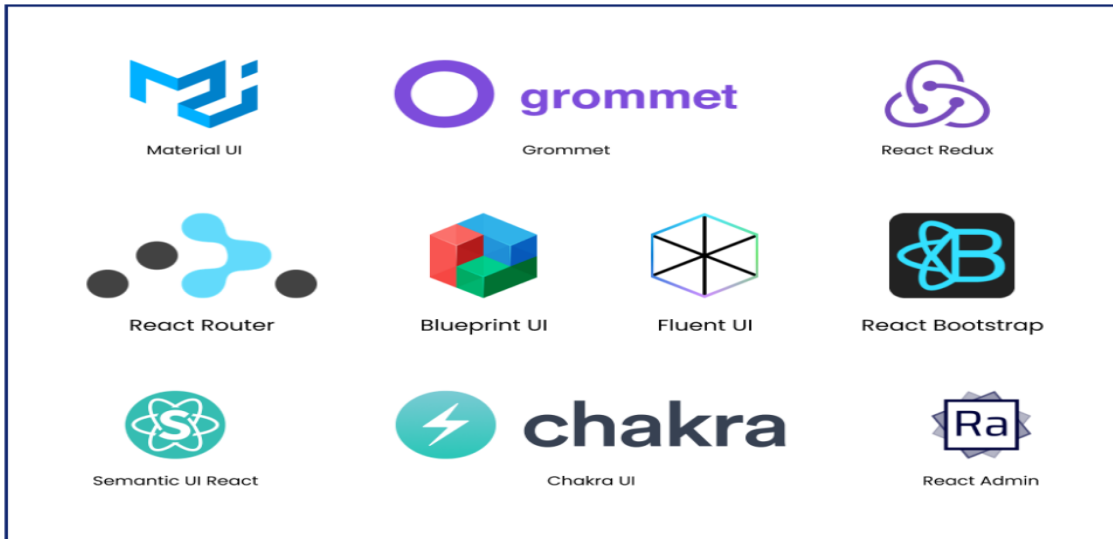
2.2.2. React

Το React είναι μια βιβλιοθήκη JavaScript ανοιχτού κώδικα χρησιμοποιείται ευρέως για τη δημιουργία διεπαφών χρήστη σε εφαρμογές ιστού. Επιτρέπει στους προγραμματιστές να δημιουργήσουν επαναχρησιμοποιήσιμα στοιχεία διεπαφής χρήστη που μπορούν να συνδυαστούν για να δημιουργήσουν πολύπλοκες διεπαφές. Το React χρησιμοποιείται συνήθως σε συνδυασμό με άλλα εργαλεία και βιβλιοθήκες για τη δημιουργία διαδικτυακών εφαρμογών πλήρους στοίβας, όπως η στοίβα MERN [3].

Η στοίβα MERN είναι μια δημοφιλής στοίβα ανάπτυξης ιστού που αποτελείται από τέσσερις τεχνολογίες: MongoDB, Express.js, React και Node.js. Το React χρησιμοποιείται για το front-end της στοίβας, ενώ τα MongoDB, Express.js και Node.js χρησιμοποιούνται για το back-end.

Το React είναι μια δημοφιλής επιλογή για τη δημιουργία διεπαφών front-end λόγω της ευελιξίας και της ευκολίας χρήσης του. Ένα από τα βασικά πλεονεκτήματα του React είναι ότι χρησιμοποιεί ένα εικονικό DOM, το οποίο είναι μια ελαφριά αναπαράσταση του πραγματικού DOM που επιτρέπει στο React να ενημερώνει γρήγορα τη διεπαφή χρήστη όταν γίνονται αλλαγές. Αυτό το καθιστά ιδανικό για τη δημιουργία πολύπλοκων, δυναμικών εφαρμογών web που απαιτούν συχνές ενημερώσεις [4].

Ένα άλλο πλεονέκτημα του React είναι η εκτεταμένη βιβλιοθήκη στοιχείων του, η οποία μπορεί εύκολα να προσαρμοστεί και να επαναχρησιμοποιηθεί. Αυτό βοηθά στον εξορθολογισμό της ανάπτυξης και στη μείωση του όγκου του κώδικα που πρέπει να γραφτεί. Το React διαθέτει επίσης μια μεγάλη και ενεργή κοινότητα προγραμματιστών, πράγμα που σημαίνει ότι οι προγραμματιστές μπορούν εύκολα να βρουν πόρους και υποστήριξη για να τους βοηθήσουν με τα έργα τους.



Εικόνα 2.3 – Διασημότερα React libraries

Σε σύγκριση με άλλα πλαίσια διεπαφής όπως το Angular και το Vue.js, το React έχει απλούστερη σύνταξη και είναι πιο εύκολο να το μάθεις για προγραμματιστές που είναι νέοι στην ανάπτυξη ιστού. Έχει επίσης καλύτερη απόδοση, χάρη στο εικονικό DOM του και τον τρόπο που χειρίζεται τις ενημερώσεις στο UI.

Συνοπτικά, το React είναι ένα ισχυρό και ευέλικτο front-end framework που είναι κατάλληλο για τη δημιουργία πολύπλοκων, δυναμικών web apps. Η ευκολία χρήσης, η εκτεταμένη βιβλιοθήκη στοιχείων και η ενεργή κοινότητα προγραμματιστών το καθιστούν δημοφιλή επιλογή μεταξύ των προγραμματιστών ιστού. Όταν χρησιμοποιείται σε συνδυασμό με τις άλλες τεχνολογίες της στοίβας MERN, το React βοηθά στον εξορθολογισμό της ανάπτυξης και στη δημιουργία υψηλής ποιότητας, κλιμακούμενων εφαρμογών ιστού.



Εικόνα 2.4 – Διασημότερα React libraries

2.2.3. AXIOS

Το εργαλείο που χρησιμοποιήθηκε για την αποστολή HTTP requests είναι το Axios, μια βιβλιοθήκη HTTP η οποία δίνει την δυνατότητα εύκολης επικοινωνίας με εξωτερικά API's, με το ίδιο το back-end (server), ενώ μπορεί και να διαχειριστεί τα δεδομένα που λαμβάνει μέσω των requests, για την React εφαρμογή μας.

```
useEffect(() => {
  let isApiSubscribed = true;
  axios.get('/orders/latestOrders')
    .then(res =>{
      if (isApiSubscribed) {
        setOrders(res.data)
      }
    }).catch(err=>{
      console.log(err);
    })
})
```

Εικόνα 2.5 – Axios Request στην εφαρμογή μας

Ένα από τα βασικά χαρακτηριστικά του Axios είναι η υποστήριξή του για "promises" [5], που επιτρέπει στους προγραμματιστές να γράφουν ασύγχρονο κώδικα που είναι πιο εύκολο να διαβαστεί και να διατηρηθεί. Με το Axios, οι προγραμματιστές μπορούν να γράφουν καθαρότερο κώδικα που είναι πιο οργανωμένος, διευκολύνοντας τον εντοπισμό σφαλμάτων. Το Axios παρέχει επίσης έναν αριθμό επιλογών διαμόρφωσης που επιτρέπουν στους προγραμματιστές να προσαρμόζουν τη συμπεριφορά των αιτημάτων τους HTTP, συμπεριλαμβανομένων των κεφαλίδων, των παραμέτρων ερωτήματος και των τύπων του respond. Επίσης, στην κλήση κλασικών HTTP requests, χρησιμοποιεί παρόμοια ονοματολογία στις συναρτήσεις του (πχ ένα POST request γίνεται με μέθοδο που ονομάζεται ".post()"). ενώ φάνηκε αρκετά χρήσιμο διότι διαχειρίζεται έξυπνα, JSON αρχεία, που χρησιμοποιούνται αρκετά στην εργασία αυτή [6].

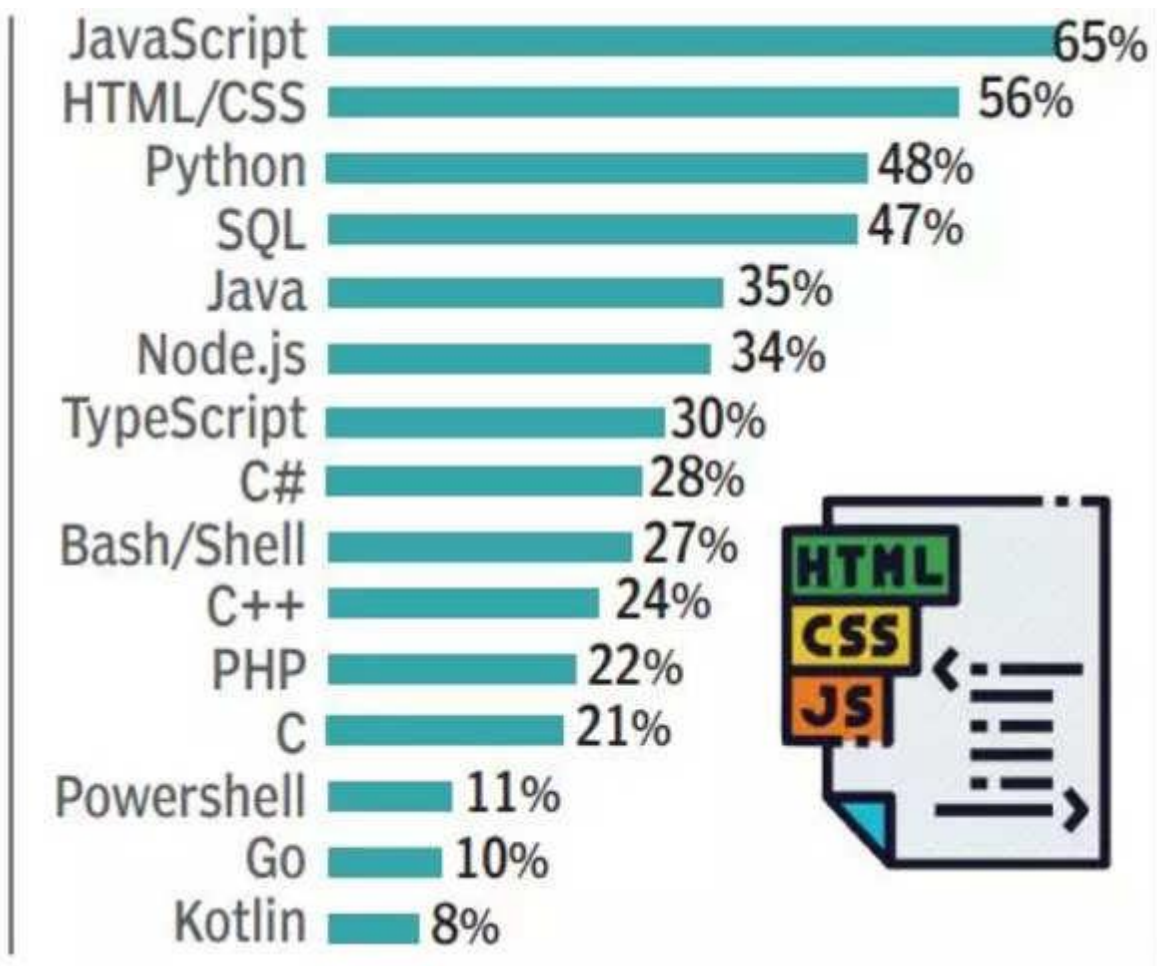
2.3. Back-End

2.3.1 JavaScript

Η κύρια γλώσσα προγραμματισμού της εφαρμογής είναι η JavaScript. Αποτελεί μια από τις πιο διευρυμένες programming languages, για ακρίβεια το 2022 κατέκτησε ξανά την πρώτη θέση ως η πιο διαδεδομένη γλώσσα μεταξύ των προγραμματιστών.

Ως αναπόσπαστο κομμάτι της τριάδας HTML, CSS, JavaScript, όσο αφορά τη δημιουργία ιστοσελίδας, καθιστά εφικτή την asynchronous ανταλλαγή δεδομένων της σελίδας με τον server, ενώ δίνει τη δυνατότητα δυναμικής ενημέρωσης της ιστοσελίδας μας. Αφορά, δηλαδή, την αλληλεπίδραση με τον χρήστη [7].

Χρησιμοποιείται όλο και συχνότερα από προγραμματιστές για full stack προγραμματισμό, δηλαδή και στο front και στο back-end, μιας και η ύπαρξη του event-driven Node.JS, ενός εργαλείου που περιγράφεται παρακάτω, επέκτεινε την λειτουργικότητα της γλώσσας server-side.



Εικόνα 2.6 – Διασημότερες γλώσσες μεταξύ των προγραμματιστών

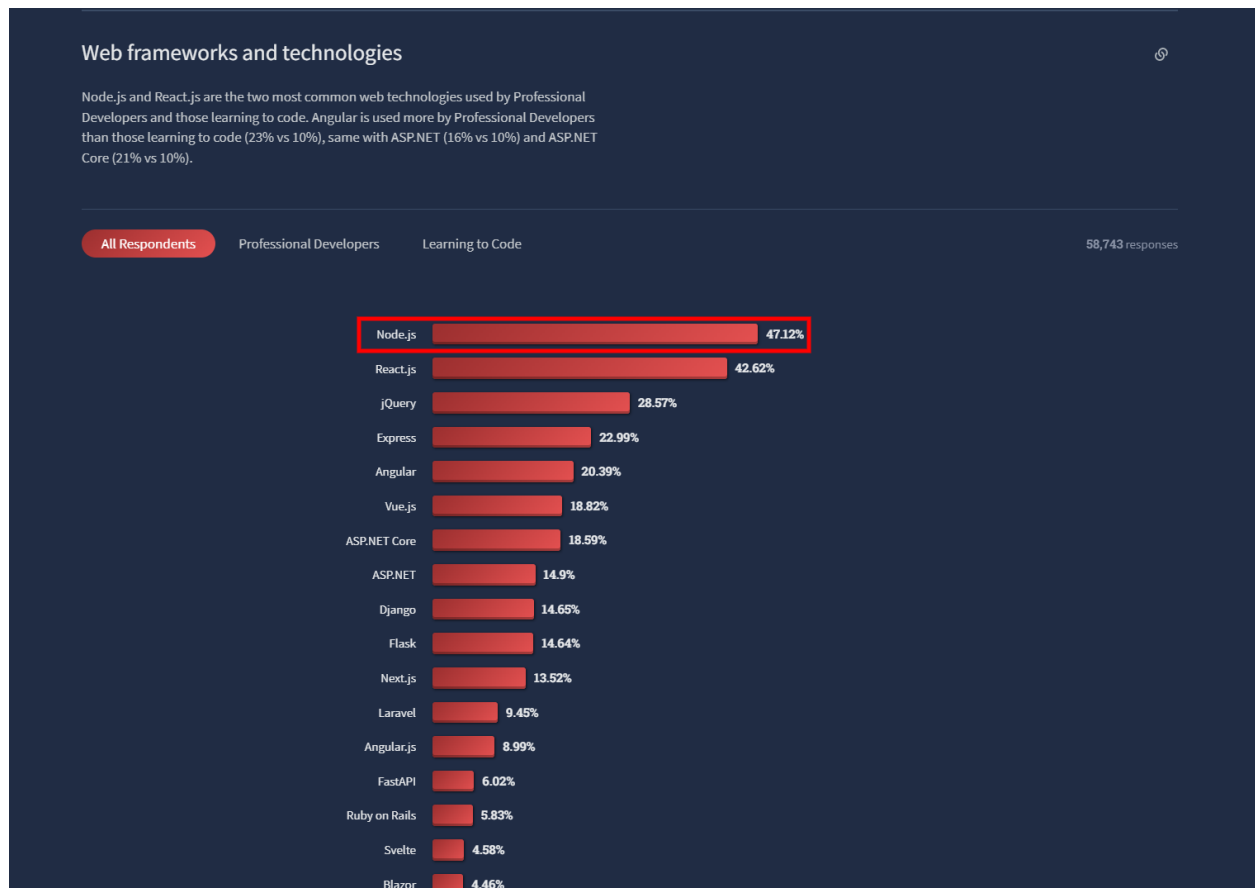
2.3.2. Node JS

Όπως προαναφέρθηκε, χρήσιμο εργαλείο για την επικοινωνία της JavaScript με τον Server είναι το Node JS, ένα open source περιβάλλον και βιβλιοθήκη της JavaScript, το οποίο τρέχει σε ποικίλες πλατφόρμες και λειτουργικά.

Καθιστά εφικτό το τρέξιμο της εφαρμογής μας, δηλαδή, εκτός του περιηγητή του client, δηλαδή του server. Συμβάλλει στο να παρέχεται δυναμικό περιεχόμενο στη σελίδα και επιτρέπει τις κλασικές λειτουργίες: create, open, read, write, delete και close σε αρχεία στον εξυπηρετητή και add, delete, modify στην database [8]. Τα Node JS αρχεία λήγουν σε “.js”,

ενώ πρέπει πάντα να είναι ορισμένα στον server.

Αυτό που καθιστά το Node JS ξεχωριστό είναι η τεχνολογία που χρησιμοποιεί για να βελτιώσει την αποδοτικότητα του. Ακολουθώντας, δηλαδή, ένα non – blocking I/O (Input / Output) μοντέλο. Μπορεί, λοιπόν, να δοθεί ένα request προς τον server ζητώντας δεδομένα, έπειτα από ένα άλλο request, ανεξαιρέτως εάν έχει δοθεί απάντηση από τον εξυπηρετητή για αυτό που προηγείται. Σε αντίθετη περίπτωση, δηλαδή σε blocking i/o δημιουργείται πολλαπλό νήμα για τα ξεχωριστά requests, χωρίς να μπορεί ο server να χειριστεί πολλά requests ταυτοχρόνως, γεγονός που σημαίνει περισσότεροι κύκλοι CPU [9]. Με αυτόν λοιπόν τον τρόπο, το Node JS πετυχαίνει μεγαλύτερη αποδοτικότητα στη διαχείριση των tasks και της μνήμης.



2.3.3. Express

Τα frameworks, δηλαδή μια δομή υποστήριξης για τη δημιουργία εφαρμογών, παίζουν καθοριστικό ρόλο στη διευκόλυνση του προγραμματιστή. Μας δίνεται πρόσβαση σε αυτά μέσω του NPM και μεταξύ των γνωστότερων Node JS frameworks, επιλέχθηκε το διασημότερο, το Express JS.

Με κύρια έμφαση στο υψηλό performance, παρέχει τη δυνατότητα δημιουργίας επιπροσθέτου software, των λεγόμενων middlewares. Μεσολαμβάνει για την επεξεργασία των requests για τα οποία δημιουργεί handlers σε ποικίλα routes, διάφορες παραμέτρους ανεξαρτέτως HTTP τύπου, ενώ επίσης αφορούν τα cookies και τη σύνδεση του χρήστη [10].

```
const app = express();
connectToDB();

app.use(cors());
app.use(express.json());
app.use(express.urlencoded({extended: true}));

app.use("/", authAdminRoutes);
app.use("/shop/", shopsRoutes);
app.use("/orders/", ordersRoutes);
app.use("/products/", productsRoutes);
app.use("/imageShop/", imageShopRoutes);
app.use("/charts", chartsRoutes);
```

Στη δική μας εφαρμογή, γίνεται εισαγωγή του express αρχικά ορίζοντας το ως συνάρτηση σε μια μεταβλητή app, ενώ με τη συνάρτηση app use, ορίζουμε path για διάφορα middlewares

που έχουν οριστεί προηγουμένως.

	Github Stars	Contributors	Monthly Downloads	Age
Express.js	54.61K	269	74.22M	12 years
Koa.js	31.78K	225	4.36M	8 years
NestJS	41.14k	263	3.5M	5 years
Socket.io	54.52K	195	17.9M	12 years
Fastify	20.7K	424	1.12M	5 years

Εικόνα 2.9 – Express.js μεταξύ άλλων βιβλιοθηκών

2.3.4. JSON

JSON, ή αλλιώς JavaScript Object Notation, είναι ένας συγκεκριμένος τρόπος δόμησης κειμένου, το οποίο χρησιμοποιείται κυρίως για αποθήκευση και μεταφορά δεδομένων μεταξύ client και Server. Με κύριο ανταγωνιστή την XML, την οποία ξεπέρασε λόγω λιγότερης πολυπλοκότητας στην μετατροπή του αρχείου σε JS, αλλά και στην έκταση [11]. Υποστηρίζεται από πληθώρα γλωσσών προγραμματισμού και εφόσον αναφέραμε δεδομένα, βάσεων δεδομένων, τόσο SQL όσο και NO SQL.

```
package.json x
1 {
2   "name": "myapplication",
3   "description": "some description here",
4   "version": "0.0.1",
5   "private": true,
6   "scripts": {
7     "start": "node ./bin/www"
8   },
9   "dependencies": {
10    "express": "~4.12.2",
11    "jade": "~1.9.2"
12  }
13 }
```

Εικόνα 2.10 – Παράδειγμα ενός package.json αρχείου

Κεφάλαιο 3 – Βάσεις δεδομένων

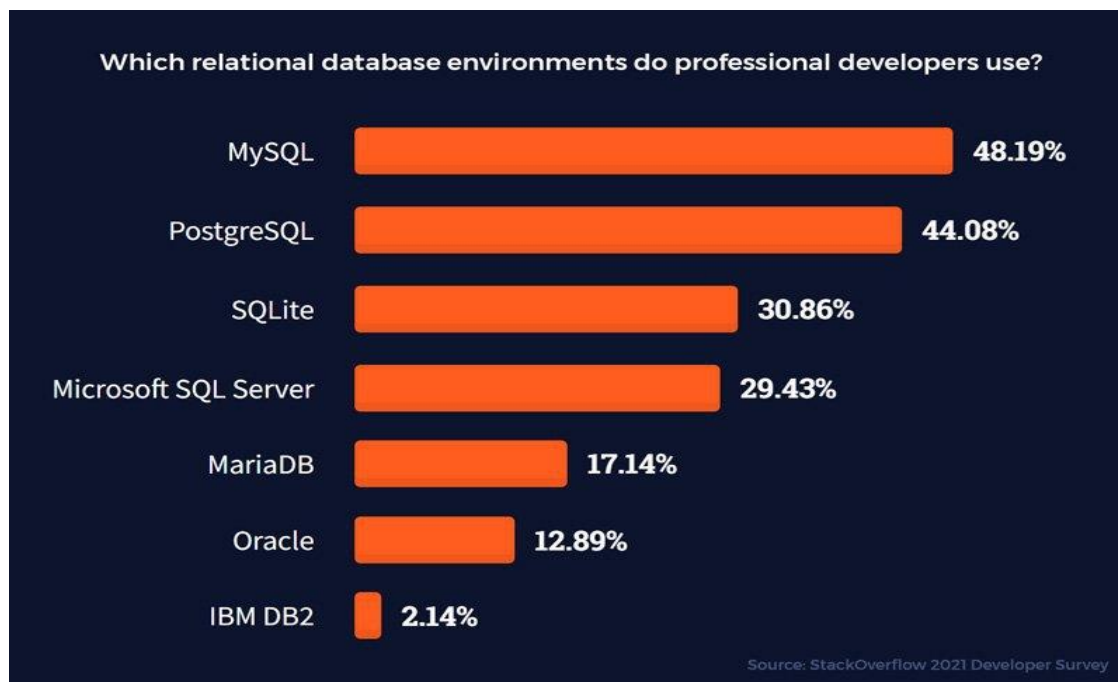
Η διεπαφή με μία βάση δεδομένων συναντάται σε ποικίλες δραστηριότητες μέσα στη ζωή μας όταν αυτές έχουν να κάνουν με την αποθήκευση αλλά και γενικότερα τη διαχείριση δεδομένων. Μία ηλεκτρονική συναλλαγή, ένας λογαριασμός σε κάποια εφαρμογή, μία συνδρομή σε κάποια υπηρεσία είναι κάποια από τα παραδείγματα, μιας και σε αυτά αποθηκεύονται διάφορα στοιχεία και πληροφορίες του χρήστη. Στην πραγματικότητα μια βάση δεδομένων είναι ένας αποθηκευτικός χώρος δεδομένων, αρχείων, τα οποία βέβαια είναι προσβάσιμα για προβολή, διαχείριση και ανάλυση [12].

Τα συνηθέστερα είδη δεδομένων είναι οι Σχεσιακές βάσεις (SQL), οι Μη Σχεσιακές βάσεις (NoSQL) και οι βάσεις αποθήκευσης στο Cloud Database.

3.1. Διαφορές SQL με NoSQL (Σχεσιακές και μη Σχεσιακές βάσεις δεδομένων)

3.1.1. SQL βάσεις

Το κύριο χαρακτηριστικό των βάσεων SQL είναι ότι είναι σχεσιακές. Αυτό σημαίνει ότι ακολουθούν πάντα ένα σχεσιακό μοντέλο με την οργάνωση των δεδομένων να γίνεται σε πίνακες με κλειδιά και ορισμένες στήλες. Τα queries δηλαδή οι εντολές αναζήτησης – επεξεργασίας δεδομένων σε μία SQL βάση μπορεί να είναι περίπλοκα, αλλά λόγω αυτής της προκαθορισμένης δομής που ακολουθούν μια αλλαγή στο σχήμα καθίσταται αρκετά πολύπλοκη ενώ όλα τα δεδομένα που καταχωρούν είναι ανάγκη να είναι ομοίως δομημένα [13].



Εικόνα 3.1 – Διασημότερες Σχεσιακές Βάσεις

Η επέκταση μιας SQL βάσης για την απόκτηση περισσότερων χώρων γίνεται κάθετα, δηλαδή, ενισχύουμε με τον εξυπηρετητή τις βάσεις μας προσθέτοντας επιπλέον hardware, δηλαδή επεξεργαστική δύναμη, μνήμη RAM ή επέκταση χωρητικότητας δίσκου.

3.1.2. NO SQL βάσεις

Οι NoSQL βάσεις ακολουθούν το μη σχεσιακό μοντέλο, δηλαδή δεν αποθηκεύουν τα δεδομένα σε κάποια δομημένη μορφή όπως πίνακα, αλλά με ποικίλους τρόπους όπως η αποθήκευση σε μορφή κλειδιού – τιμής (key value store), είναι ένας από τους απλούστερους τρόπους αποθήκευσης, σε JSON, XML, YAML documents, με τα δεδομένα να ακολουθούν τα αντίστοιχα format, ενώ δίνεται και η δυνατότητα αποθήκευσης σε γράφιμα. Φαίνεται λοιπόν πως αυτή η ευελιξία στη δομή και των τρόπων αποθήκευσης των δεδομένων πέρα από το ότι καθιστά την γενικότερη χρήση τους ευκολότερη, επιτρέπει να γίνονται εύκολα αλλαγές στα σχήματα και τα format των δεδομένων [14].

Επίσης, τα queries είναι αξιολογούμενα γρηγορότερα, αφού τα δεδομένα είναι αποθηκευμένα

και προσβάσιμα συγκεντρωτικά, δίχως να χρειάζονται joins (συνδέσεις) μεταξύ πολλών πινάκων για την αναζήτηση των δεδομένων που επιθυμούμε. Καταλαβαίνουμε όμως πως τα queries που χρησιμοποιούμε στις no SQL βάσεις είναι περιορισμένα στην πολυπλοκότητά τους.

Η επέκταση σε αντίθεση με τις SQL βάσεις γίνεται οριζόντια, δηλαδή απλά μεταφέροντας τη βάση μας σε ένα νέο μεγαλύτερο server, βέβαια συνήθως με μεγαλύτερο κόστος [15].

Types of Non-Relational Databases

upwork

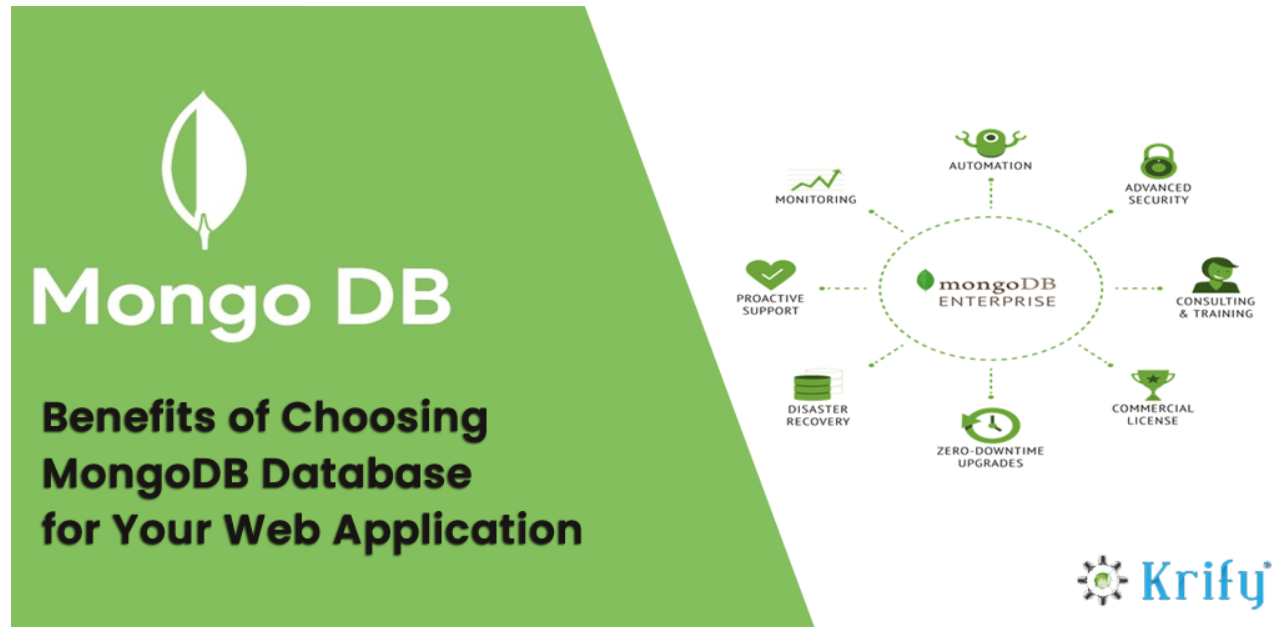
Types	Performance	Scalability	Flexibility	Complexity
Key-Value Store	High	High	High	None
Column Store	High	High	Moderate	Low
Document	High	Variable (High)	High	Low
Graph Database	Variable	Variable	Hight	High

Εικόνα 3.2 – Τύποι Μη Σχεσιακών Βάσεων Δεδομένων

3.2. MongoDB

Η MongoDB είναι ένα πρόγραμμα διαχείρισης, μιας NO SQL βάσης δεδομένων, ανοιχτού κώδικα και είναι η πλέον δημοφιλέστερη μεταξύ των υπόλοιπων NO SQL βάσεων. Σε αυτή, οι καταχωρήσεις γίνονται σε μορφή αρχείων με ζευγάρια κλειδιών τιμής, χωρισμένα σε διάφορα collections, αναλόγως πως θέλουμε να τα διαχωρίσουμε. Τα ζεύγη αυτά μπορούν να είναι διαφορετικά σε κάθε document, άρα κάποιο collection μπορεί να περιέχει documents με data

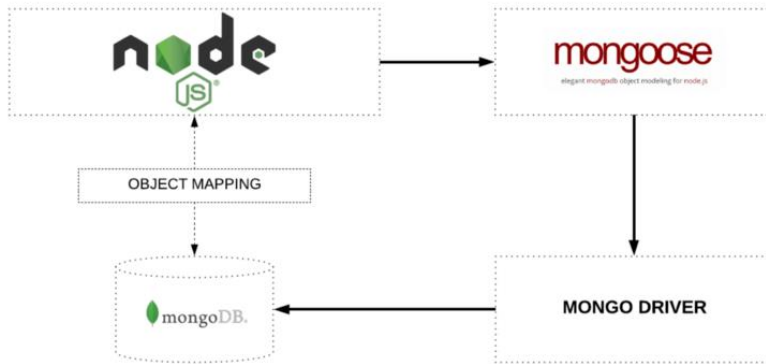
διαφορετικής δομής και αφού είναι JSON αρχεία τα δεδομένα αυτά μπορούν να είναι όλων των τύπων που υποστηρίζει το JSON format [16]. Υποστηρίζεται απο όλα τα γνωστά λειτουργικά και γλώσσες προγραμματισμού.



Εικόνα 3.3 – Προτερήματα της MongoDB

3.3. Επικοινωνία με τις βάσεις δεδομένων

Για να επιτευχθεί η επικοινωνία μεταξύ του Server και της βάσης δεδομένων, αλλά και η δημιουργία ενός model για κάποιο object που θέλουμε να χρησιμοποιήσουμε στη βάση μας, χρησιμοποιούμε μία ODM (Object Data Modeling) βιβλιοθήκη της Mongo, την Mongoose. Εκτελεί την διεπαφή server- βάσεις με δυνατότητες add, delete, modify, search, με τα κατάλληλα objects που είναι απαραίτητα στον κώδικά μας για να συνδέονται με τη βάση. Εγκαθίσταται και αυτή μέσω npm αφού είναι βιβλιοθήκη ODM που χρησιμοποιεί και το NodeJS [17].

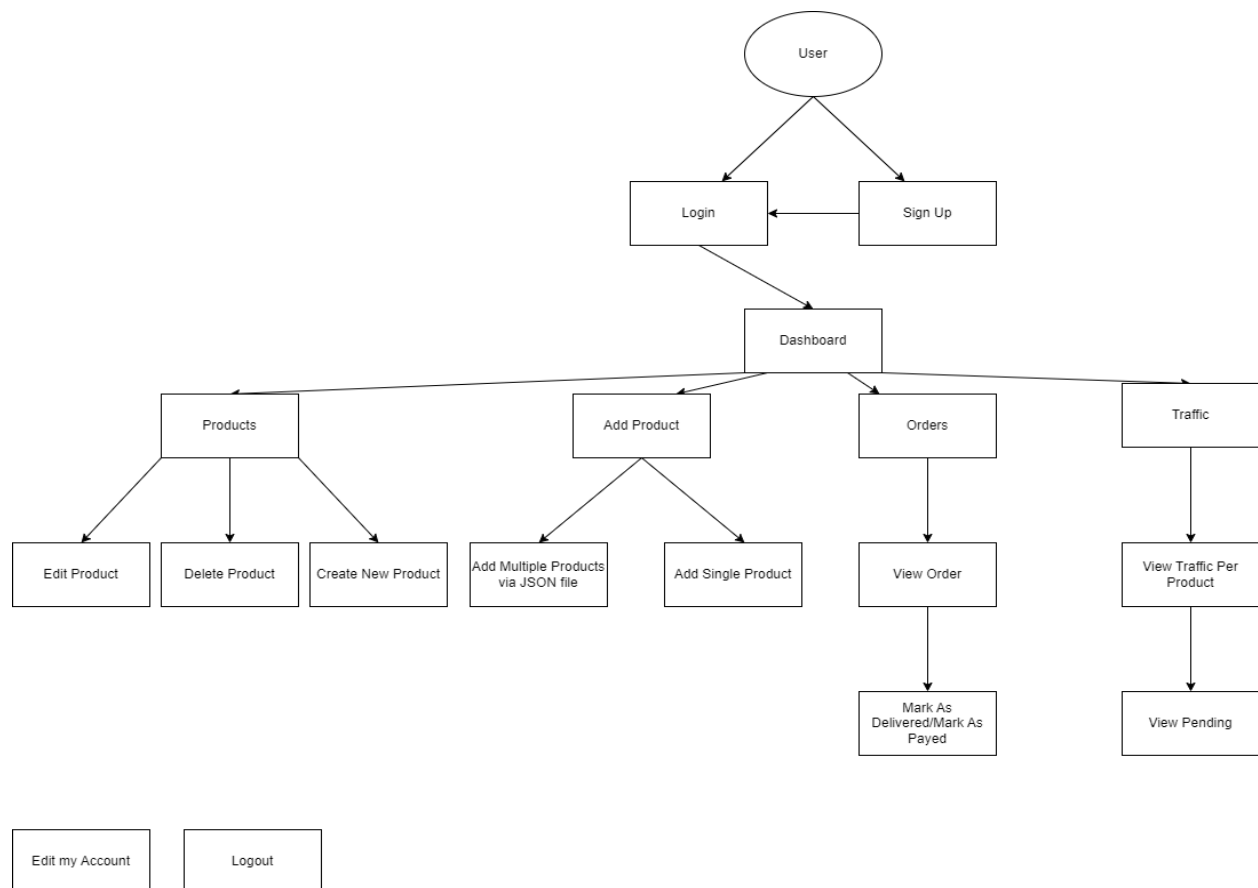


*Εικόνα 3.4 –
Σύνδεση μεταξύ
Node.JS και βάσης
με το Mongoose*

Κεφάλαιο 4 – Η εφαρμογή

Το web application αυτό αποτελεί μια προσθήκη σε μια προϋπάρχουσα εφαρμογή αγορών δώρων, δίνοντας στα συμβεβλημένα καταστήματά της τη δυνατότητα να διαχειριστούν σε ένα user friendly περιβάλλον τα προϊόντα, τις παραγγελίες τους και να ενημερώνονται με διάφορα στατιστικά για την πορεία των εσόδων από την εγγραφή τους στην εφαρμογή.

Αναλυτικότερα, το κάθε συμβεβλημένο κατάστημα επισκέπτεται μέσω browser την εφαρμογή και εγγράφεται σε αυτή κάνοντας sign up καταθέτοντας τα στοιχεία που απαιτούνται. Έπειτα, κατά τη σύνδεση, βρίσκεται στην αρχική σελίδα τύπου dashboard, στην οποία βρίσκονται δύο διαγράμματα αναφορικά με τις πωλήσεις ανά χρονικά διαστήματα, τα συνολικά έσοδα, τις συνολικές παραγγελίες και προϊόντα. Στο κάτω μέρος της σελίδας υπάρχει ο πίνακας latest orders, όπου υπάρχει μια προεπισκόπηση των τελευταίων παραγγελιών με τη δυνατότητα επέκτασης κάθε παραγγελίας για περισσότερες πληροφορίες.



Εικόνα 4.1 – Αρχιτεκτονική Εφαρμογής

Από την αριστερή στήλη, πατώντας το πλήκτρο products βρίσκεται στη σελίδα των προϊόντων του. Εκεί παρουσιάζονται όλα τα προϊόντα με δυνατότητα, μέσω των αντίστοιχων πλήκτρων, αναλυτικής προβολής και επεξεργασίας είτε διαγραφής του προϊόντος. Παράλληλα, τα προϊόντα μπορούν να ταξινομηθούν ανάλογα με την κατηγορία τους ή την ημερομηνία που προστέθηκαν. Δίνεται επίσης η επιλογή, είτε από την σελίδα products είτε από την επόμενη επιλογή στην αριστερή στήλη, το πλήκτρο 'Add Products', η προσθήκη προϊόντων. Όπως προαναφέρθηκε, αυτό πραγματοποιείται μέσω συμπλήρωσης των κατάλληλων πεδίων στη σελίδα addproduct, είτε με το ανέβασμα JSON αρχείου, επιλογή η οποία διαγράφει όσα προϊόντα διαθέτει μέχρι αυτή τη στιγμή το κατάστημα από την εφαρμογή μας, και την με ενημερώνει με νέα, τα οποία βρίσκονται σε αυτό.

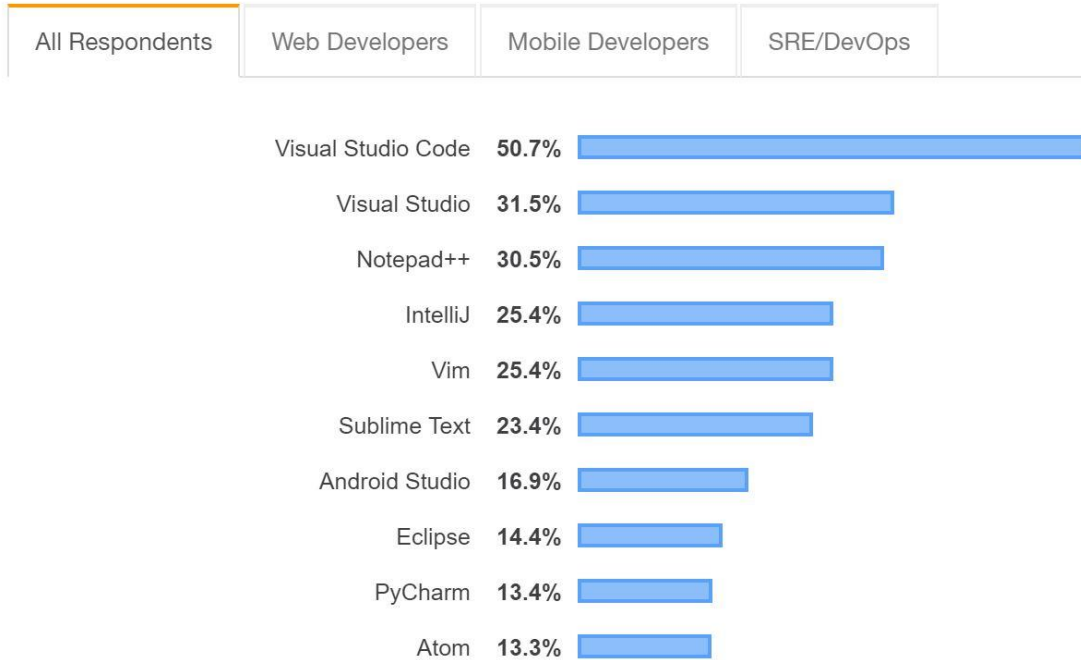
Επιπλέον, είναι δυνατή η παρακολούθηση των παραγγελιών μέσω της σελίδας "Orders". Σε αυτή, αποτυπώνεται μια προεπισκόπηση των παραγγελιών του καταστήματος, με επιλογή φιλτραρίσματος σε παραδομένες και μη παραγγελίες. Κλικάροντας σε μια παραγγελία για περισσότερες πληροφορίες, διακρίνουμε το προϊόν, τα στοιχεία του παραλήπτη, και μπορούμε να ορίσουμε μια παραγγελία ως παραδομένη.

Τέλος, έχουμε εποπτεία όλων των συναλλαγών που πραγματοποιούνται για κάποιο δώρο μέσω της σελίδας transactions. Σε αυτή, εντοπίζουμε, για κάθε προϊόν που βρίσκεται σε κάποια ενεργή wishlist, πληροφορίες για κάθε χρήστη και τα χρήματα που συμπληρώνει για την ολοκλήρωση του ποσού.

4.1. Μέσα ανάπτυξης εφαρμογής

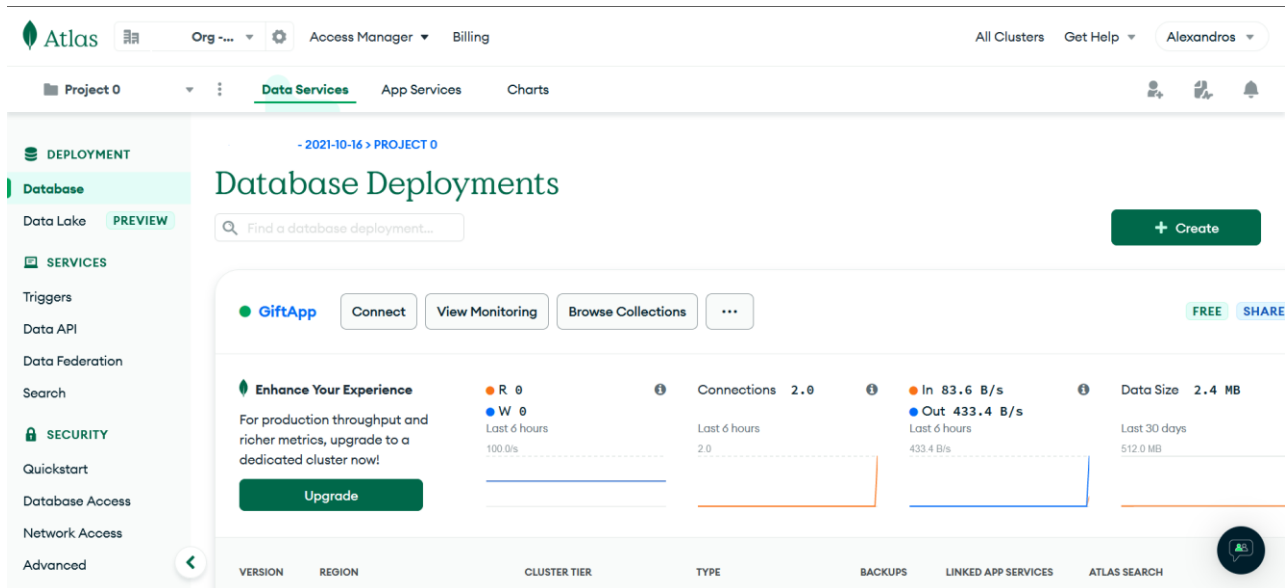
Ο συνδυασμός των εργαλείων MongoDB ως βάση δεδομένων, React, Express και NodeJS, ονομάζεται MERN STACK και δίνει τη δυνατότητα εύκολης ανάπτυξης σε full stack επίπεδο με κάθε εργαλείο από αυτά όπως προαναφέρθηκε, να χρησιμοποιείται είτε για λειτουργίες front ή back end. Με τη στοίβα MERN, οι προγραμματιστές μπορούν να δημιουργήσουν εφαρμογές Ιστού από άκρο σε άκρο με μία μόνο γλώσσα προγραμματισμού, τη JavaScript, και να επωφεληθούν από το εκτεταμένο οικοσύστημα εργαλείων και βιβλιοθηκών του. Η MERN είναι μια εξαιρετική επιλογή για προγραμματιστές που θέλουν να δημιουργήσουν σύγχρονες, επεκτάσιμες και υψηλής απόδοσης εφαρμογές web [18]. Ο κώδικας συντάχθηκε μέσω του Microsoft Visual Studio Code και ως browser χρησιμοποιήθηκε το Google Chrome.

Most Popular Development Environments



Εικόνα 4.3 – Διασημότερα Development environments 2023

Για user friendly επαφή και διαχείριση της βάσης δεδομένων έγινε χρήση του MongoDB Atlas που και αυτό βρίσκουμε μέσω browser παρέχοντας τη δυνατότητα έπειτα από τη δημιουργία λογαριασμού και σύνδεσης ενός νέου project με τους servers μας να έχουμε πρόσβαση στη βάση μας από όπου και αν βρισκόμαστε. Το MongoDB Atlas είναι μια υπηρεσία βάσης δεδομένων cloud που προσφέρεται από την MongoDB, Inc. που παρέχει όλες τις δυνατότητες της δημοφιλούς βάσης δεδομένων MongoDB, όπως αυτόματη κλιμάκωση, ασφάλεια και απόρρητο δεδομένων. Με το MongoDB Atlas, οι χρήστες μπορούν να αναπτύξουν, να λειτουργήσουν και να κλιμακώσουν μια βάση δεδομένων MongoDB σε παρόχους cloud όπως το AWS, το Google Cloud Platform και το Microsoft Azure. Η υπηρεσία είναι διαθέσιμη σε περισσότερες από 80 περιοχές σε όλο τον κόσμο, επιτρέποντας στους προγραμματιστές να αναπτύξουν τις βάσεις δεδομένων τους πιο κοντά στους χρήστες τους με λιγότερο χρόνο και καλύτερη απόδοση. Το MongoDB Atlas προσφέρει επίσης δυνατότητες δημιουργίας αντιγράφων ασφαλείας και ανάκτησης, παρακολούθησης και ειδοποιήσεων για τη διασφάλιση της υγείας και της διαθεσιμότητας της βάσης δεδομένων. Συνολικά, το MongoDB Atlas παρέχει έναν αξιόπιστο και αποτελεσματικό τρόπο εκτέλεσης του MongoDB στο cloud χωρίς την ανάγκη σύνθετης διαχείρισης υποδομής.



Εικόνα 4.4 – Screenshot από το UI του δικού μας ATLAS

4.2. Set Up

Αφού μέσω του Code editor ανοίξουμε το φάκελο της εφαρμογής μας ως ένα νέο project, το αρχικοποιούμε ως ένα NodeJS project μέσω του npm και συγκεκριμένα μέσω του terminal και αρχικά της εντολής “npm init”. Με αυτό τον τρόπο στο αρχείο package.json, βλέπουμε πληροφορίες για το project μας και για τις βιβλιοθήκες σε αυτό και θα αποτυπώνονται όσες προστεθούν μελλοντικά. Έπειτα απαιτείται επικοινωνία του server μας και της βάσης δεδομένων μας MongoDB, το οποίο επιτυγχάνεται, όπως προαναφέρθηκε, μέσω του Mongoose.

```
const mongoose = require("mongoose");

mongoose.set('useNewUrlParser', true);
mongoose.set('useFindAndModify', false);
mongoose.set('useCreateIndex', true);
mongoose.set('useUnifiedTopology', true);

const connectToDB = async () =>{
  try {
    await mongoose.connect(process.env.MONGO_URL)
    console.log("Connected to DB");
  } catch (error) {
    console.log("Error: connectToDB: " + error);
    process.exit(1);
  }
}

module.exports = connectToDB;
```

Εικόνα 4.5 – Σύνδεση mongoose με βάση

4.3. Μοντέλα Βάσεις δεδομένων (Models)

Παρακάτω παρατίθενται screenshots από τα μοντέλα που χρησιμοποιήθηκαν στον κώδικα μας για τη βάση δεδομένων μας.

```
const transactionSchema = new mongoose.Schema(
  {
    gift: {
      type: Schema.Types.ObjectId,
      required: true,
      ref: 'Gift'
    },
    donator: {
      type: Schema.Types.ObjectId,
      required: true,
      ref: 'User'
    },
    amount: {
      type: Number
    }
  },
  { timestamps: true }
);
```

Εικόνα 4.6. – Μοντέλο των transactions

```

const productSchema = new mongoose.Schema(
  {
    shop: {
      type: Schema.Types.ObjectId,
      ref: 'Shop'
    },
    productId: {
      type: String,
      default: ""
    },
    productName: {
      type: String,
      default: ""
    },
    productLink: {
      type: String,
      default: ""
    },
    imageLink: {
      type: String,
      default: ""
    },
    price: {
      type: Number,
      default: 0
    },
    countInStock: {
      type: Number,
      default: 0
    },
    categoryPath: {
      type: String,
      default: ""
    },
    manufacturer: {

```

```

const shopSchema = new mongoose.Schema(
  {
    brand: {
      type: String,
      required: [true, "Brand is required."],
      unique: true
    },
    email: {
      type: String,
      required: [true, "Email is required."],
      unique: true
    },
    phone: {
      type: String,
      required: [true, "Phone is required."]
    },
    fax: {
      type: String,
      required: [true, "Fax is required."]
    },
    edra: {
      type: String,
      required: [true, "Εδρα is required."]
    },
    afm: {
      type: String,
      required: [true, "ΑΦΜ is required."]
    },
    doy: {
      type: String,
      required: [true, "ΔΟΥ is required."]
    },
    gemh: {
      type: String,
      required: [true, "ΓΕΜΗ is required."]
    },
    password: {
      type: String,
      required: [true, "Password is required."]
    },
    shopPic: {
      type: String,
      default: "6203c7e55e961b5f220ffb6"
    }
  }, { timestamps: true }
);

```

Εικόνα 4.7 – Μοντέλα προϊόντος (αριστερά), καταστήματος (δεξιά)

```
product:{
  type: Schema.Types.ObjectId,
  ref: 'Product'
},
productName:{
  type: String,
  default: ""
},
imageLink:{
  type: String,
  default: ""
},
price:{
  type: Number,
  default: 0
},
shop: {
  type: Schema.Types.ObjectId,
  ref: 'Shop'
},
shippingCost:{
  type: Number,
  default: 0
},
delivered:{
  type: String,
  default: "false"
},
members: [{userId: Schema.Types.ObjectId, username: String, amount: Number, donatedAt: String}]
},
{ timestamps: true }
);
```

Εικόνα 4.8 – Συνέχεια μοντέλου προϊόντος

```

const orderSchema = new mongoose.Schema(
  {
    owner: {
      ownerId: Schema.Types.ObjectId,
      ownerUsername: String,
      ownerEmail: String,
      fName: {
        type: String,
        default: ""
      },
      lName: {
        type: String,
        default: ""
      },
      city: {
        type: String,
        default: ""
      },
      country: {
        type: String,
        default: ""
      },
      postalCode: {
        type: String,
        default: ""
      },
      address: {
        type: String,
        default: ""
      },
      ringbell: {
        type: String,
        default: ""
      },
      floor: {
        type: String,
        default: ""
      },
      phone: {
        type: String,
        default: ""
      }
    },
  },
);

```

Εικόνα 4.9 – Μοντέλο παραγγελίας

4.4. Routes

Τα routes είναι υπεύθυνα για την ανακατεύθυνση μέσα στον server. Χωρίζονται σε δύο κατηγορίες private και public, δηλαδή δημόσια και ιδιωτικά, ανάλογα με το αν ο χρήστης έχει αποκτήσει token μέσω authentication, δηλαδή επιτυχημένης σύνδεσης με τα περισσότερα, πέρα από αυτά που αφορούν τη σύνδεση και την εγγραφή του χρήστη, να είναι private.

Αναλυτικότερα:

Routes	Purposes
/registeradmin	POST: Εισαγωγή των στοιχείων που συμπληρώνει το κατάστημα στη βάση, authentication token.
/loginAdmin	POST: Έλεγχος email και password και αποστολή token.
/charts/orderPerDate	GET: Εμφάνιση του διαγράμματος παραγγελιών ανά μήνα στο τρέχων έτος.
/charts/orderPerYear	GET: Εμφάνιση του διαγράμματος ποσοστού παραγγελιών ανά έτος.
/orders/totalOrders	GET: Επιστρέφει τον αριθμό συνολικών παραγγελιών στο συγκεκριμένο κατάστημα από την εγγραφή στην εφαρμογή.
/orders/totalAmount	GET: Επιστρέφει τον αριθμό συνολικών εσόδων από την εγγραφή στην εφαρμογή.
/orders/numofProducts	GET: Επιστρέφει τον αριθμό συνολικών προϊόντων που πωλούνται από το συγκεκριμένο κατάστημα μέσω της εφαρμογής.
/orders/allOrders	GET: Επιστρέφει όλες τις παραγγελίες στο συγκεκριμένο κατάστημα από την εγγραφή στην εφαρμογή.
/orders/latestOrders	GET: Επιστρέφει τις πιο πρόσφατες παραγγελίες.
/orders/markAsDelivered	POST: Ορίζει μια παραγγελία ως απεσταλμένη, στην σελίδα με όλες τις παραγγελίες.
/products/allProducts	POST: Αποστολή friend request στον χρήστη με id που στέλνεται στο body.
/products/allProductsWithTranstactions	POST: Ακύρωση friend request στον χρήστη με id που στέλνεται στο body.
/products/editProduct	POST: Αποθηκεύει τις αλλαγές που έγιναν στα πεδία ενός προϊόντος.
/products/deleteProduct	POST: Διαγράφει ένα προϊόν από το κατάστημα και τη βάση δεδομένων μας.

/products/addSingleProduct	POST: Δημιουργεί στη βάση ένα νέο προϊόν για το κατάστημα με βάση τα στοιχεία που συμπληρώθηκαν στα αντίστοιχα πεδία στη σελίδα των προϊόντων.
/shops/auth	GET: Επιστροφή στοιχείων συνδεδεμένου καταστήματος.
/shops/edit	POST: Αποθήκευση των πληροφοριών που το συμπληρώνει το κατάστημα στα πεδία της σελίδας αλλαγής πληροφοριών καταστήματος.
/shops/uploadFile	POST: Αποθήκευση προϊόντων μέσω αρχείου JSON.
/shops/delete	DELETE: Διαγραφή του καταστήματος και των προϊόντων του, ενημέρωση κατάλληλων πεδίων.
/imageShop/:id	GET: Επιστροφή εικόνας με το συγκεκριμένο id.
/imageShop/delete	DELETE: Διαγραφή της φωτογραφίας προφίλ του καταστήματος.
/imageShop/upload	POST: Ενημέρωση της φωτογραφίας προφίλ του καταστήματος.

Εικόνα 4.10 – Πίνακας με τα Routes της εφαρμογής

4.5. Upload JSON αρχείου

Όπως προαναφέρθηκε, παρέχεται στα καταστήματα η δυνατότητα να ενημερώνουν τα προϊόντα τους, κάνοντας upload ενός JSON αρχείου στο αντίστοιχο πεδίο. Κατά την ενημέρωση του JSON αρχείου από τον χρήστη, προϊόντα που λείπουν από το καινούριο αρχείο αφαιρούνται αυτόματα από τη βάση μας και από το κατάστημα, λόγω μη εύρεσης του Id του προϊόντος στο αρχείο.

Τα πεδία και η μορφή του αρχείου αυτού είναι προκαθορισμένα. Συγκεκριμένα είναι:


```
{
  "productId": "13014602",
  "productName": "HUAWEI WATCH GT 2 - SPORT EDITION BLACK (55024474)" ,
  "productLink": "https://www.mobilepro-shop.gr/el/smartwatches-accessories/25118-huawei-watch-gt-2-sport-edition-b",
  "imageLink": "https://www.mobilepro-shop.gr/123613-thickbox_default/huawei-watch-gt-2-sport-edition-black-5502447",
  "price": 249.05 ,
  "countInStock": 5,
  "categoryPath": "Technology/Smartwatches",
  "manufacturer": " HUAWEI",
  "weight": "0.215",
  "size": "41mm",
  "color": "Black",
  "shippingCost": 2.00,
  "description": ""
},
```

Εικόνα 4.11 – Παράδειγμα JSON αρχείου

4.6. User interface (Διεπαφή Χρήστη)

```
{
  "productId": Id Προϊόντος στην βάση του καταστήματος,
  "productName": Όνομα προϊόντος ,
  "productLink": "URL απο σελίδα καταστήματος",
  "imageLink": "URL φωτογραφίας προϊόντος",
  "price": Τιμή προϊόντος σε δολάρια,
  "countInStock": Αριθμός διαθέσιμων προϊόντων,
  "categoryPath": "Κατηγορία Προϊόντος",
  "manufacturer": "Κατασκευαστής",
  "weight": "Βάρος",
  "size": "Μέγεθος",
  "color": "Χρώμα",
  "shippingCost": Κόστος μεταφορικών εξόδων,
  "description": "Περιγραφή"
},
```

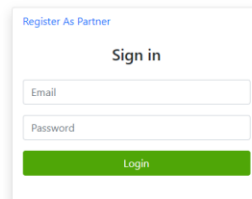
Εικόνα 4.12 – Πεδία JSON αρχείου

Η διεπαφή χρήστη (UI) μιας εφαρμογής ιστού είναι η γραφική αναπαράσταση της εφαρμογής με την οποία αλληλοεπιδρούν οι χρήστες. Περιλαμβάνει όλα τα στοιχεία που βλέπουν και αλληλοεπιδρούν οι χρήστες, όπως κουμπιά, μενού, φόρμες και άλλα γραφικά στοιχεία. Το UI έχει σχεδιαστεί για να είναι φιλικό προς το χρήστη και διαισθητικό, με στόχο να διευκολύνει τους χρήστες να πλοηγούνται στην εφαρμογή και να εκτελούν τις εργασίες τους εύκολα και αποτελεσματικά. Ένα καλά σχεδιασμένο UI θα πρέπει να είναι αισθητικά ευχάριστο, με συνεπή διάταξη και σχεδιασμό και θα πρέπει να παρέχει σαφή και συνοπτική ανατροφοδότηση στους χρήστες όταν αλληλοεπιδρούν μαζί του. Συνολικά, η διεπαφή χρήστη διαδραματίζει κρίσιμο ρόλο στην επιτυχία μιας διαδικτυακής εφαρμογής, καθώς επηρεάζει άμεσα την εμπειρία του χρήστη και μπορεί να επηρεάσει τον τρόπο με τον οποίο οι χρήστες αντιλαμβάνονται την εφαρμογή.

Ακολουθούν στιγμιότυπα (screenshots) από την εφαρμογή μας.

Η πρώτη επαφή έρχεται μέσω της σελίδας σύνδεσης/εγγραφής.

Στην σελίδα σύνδεσης ζητούνται το email και ο κωδικός πρόσβασης του χρήστη, ενώ στην σελίδα εγγραφής χρειάζεται να συμπληρωθούν και τα στοιχεία που απεικονίζονται.



The image shows a login form with the following elements:

- A link labeled "Register As Partner" in blue text at the top left.
- A heading "Sign in" centered above the form.
- An input field labeled "Email".
- An input field labeled "Password".
- A green button labeled "Login" at the bottom.

Εικόνα 4.13 – Login Page

Login

Sign in

Brand

Phone

Fax

ΑΦΜ

Εδρα

ΔΟΥ

ΓΕΜΗ

Email

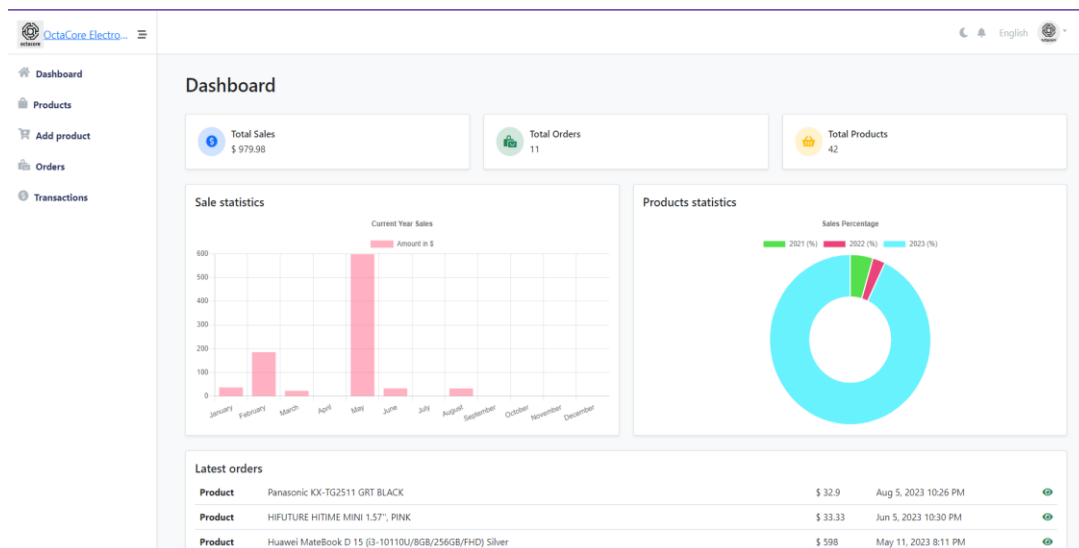
Password

Confirm Password

Sign Up

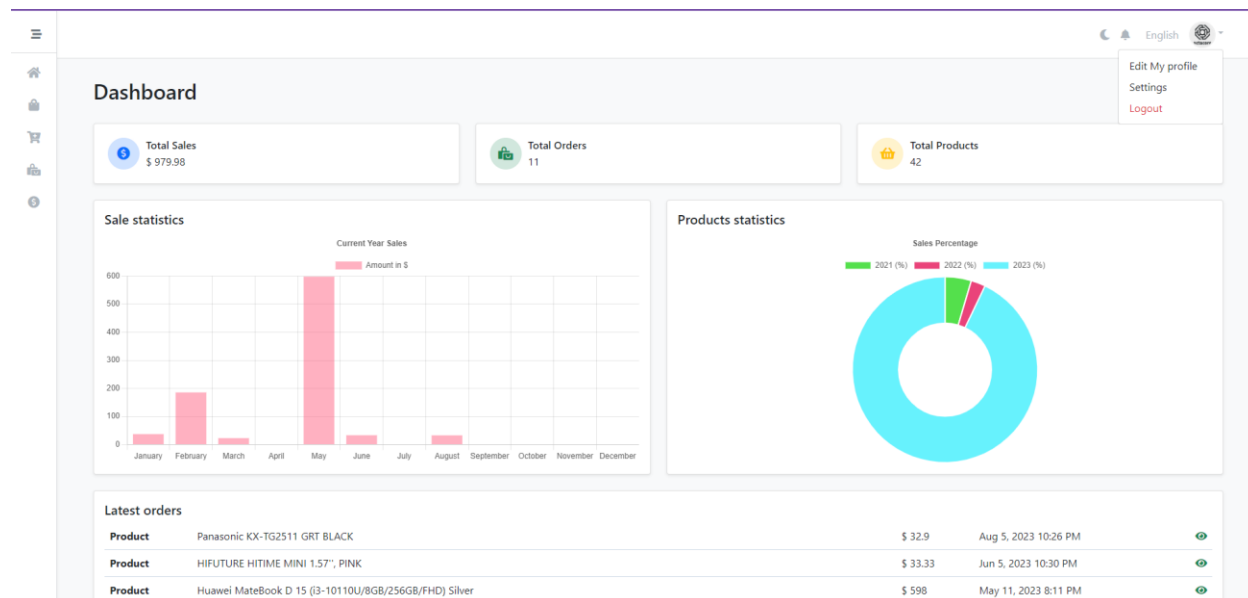
Εικόνα 4.14 – Sign in page

Έπειτα από την σύνδεση, ο χρήστης ανακατευθύνεται στην κεντρική σελίδα τύπου dashboard. Σε αυτή, αποτυπώνονται διαγράμματα σχετικά με τις πωλήσεις του καταστήματος έσοδα/μήνα για αυτή τη χρονιά και ποσοστό πωλήσεων/χρόνια εγγραφής στην εφαρμογή. Επίσης φαίνονται τα συνολικά έσοδα, οι συνολικές παραγγελίες και τα προϊόντα που διαθέτει αυτή τη στιγμή το κατάστημα στην εφαρμογή.



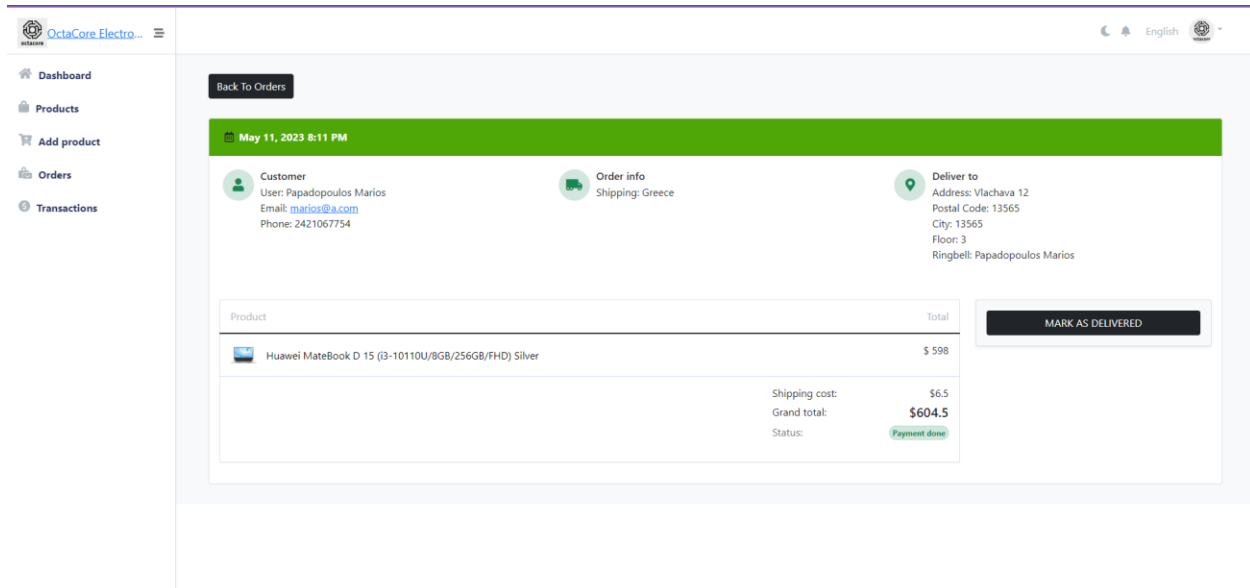
Εικόνα 4.15 – Dashboard Page

Ο χρήστης έχει την δυνατότητα να περιορίσει την μπάρα πλοήγησης στην αριστερή πλευρά, έχοντας ακόμα δυνατότητα χρήσης των επιλογών που διαθέτει, καθώς και να ανοίξει το μενού περισσότερων επιλογών από το logo δεξιά στην επάνω μπάρα, όπως φαίνεται και στο screenshot. Από εκεί μπορεί να αποσυνδεθεί, να ενημερώσει τα στοιχεία του, ή να αλλάξει το logo του καταστήματος.



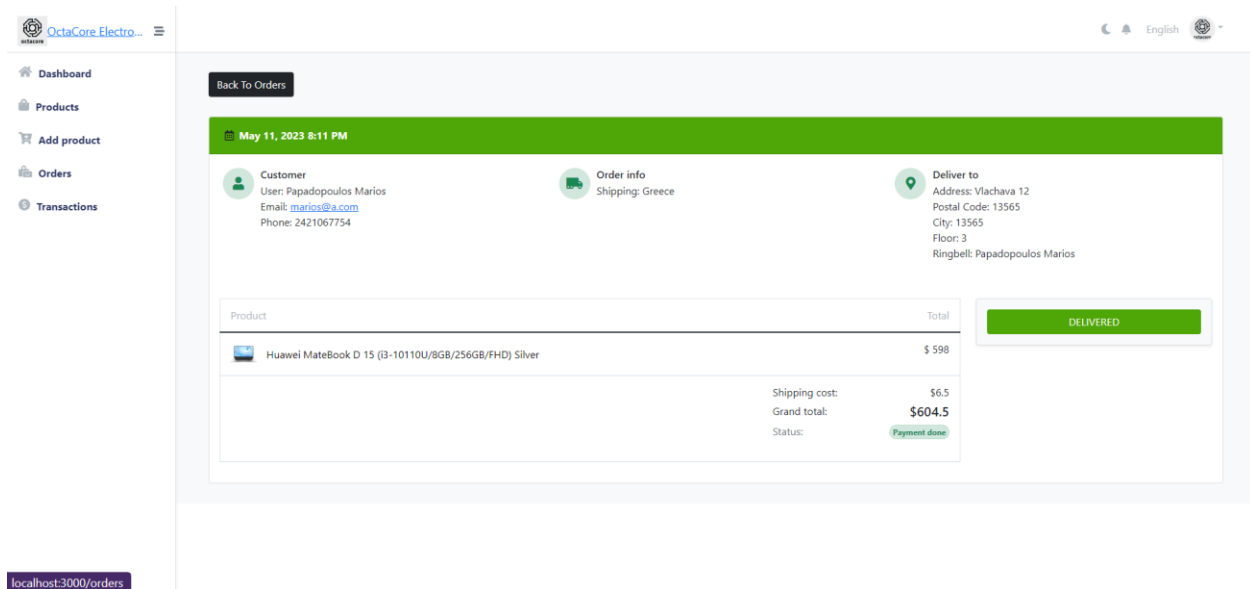
Εικόνα 4.16 – Top Left Menu

Στο κάτω μέρος διακρίνεται μια λίστα από τις τελευταίες παραγγελίες, όπου αναγράφεται το προϊόν, το ποσό, η ημερομηνία της παραγγελίας αλλά και ένα πλήκτρο που δίνει την δυνατότητα αναλυτικότερης προβολής της παραγγελίας.



Εικόνα 4.17 – Order Details

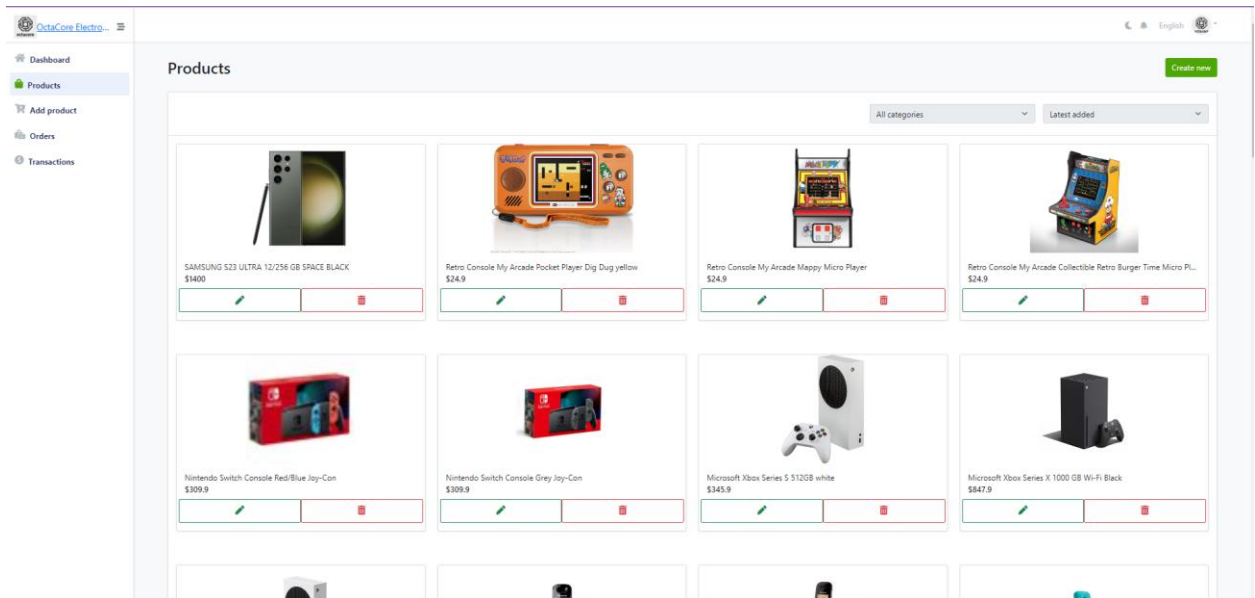
Διακρίνουμε τα στοιχεία του παραλήπτη, πληροφορίες για την διεύθυνση παράδοσης, ενώ αναγράφεται αναλυτικά το όνομα του προϊόντος, το κόστος του και το κόστος μεταφοράς. Όταν η παραγγελία παραδοθεί, το κατάστημα την ορίζει ως “delivered” μέσω του πλήκτρου στη φωτογραφία, και η παραγγελία μετατρέπεται παραδομένη.



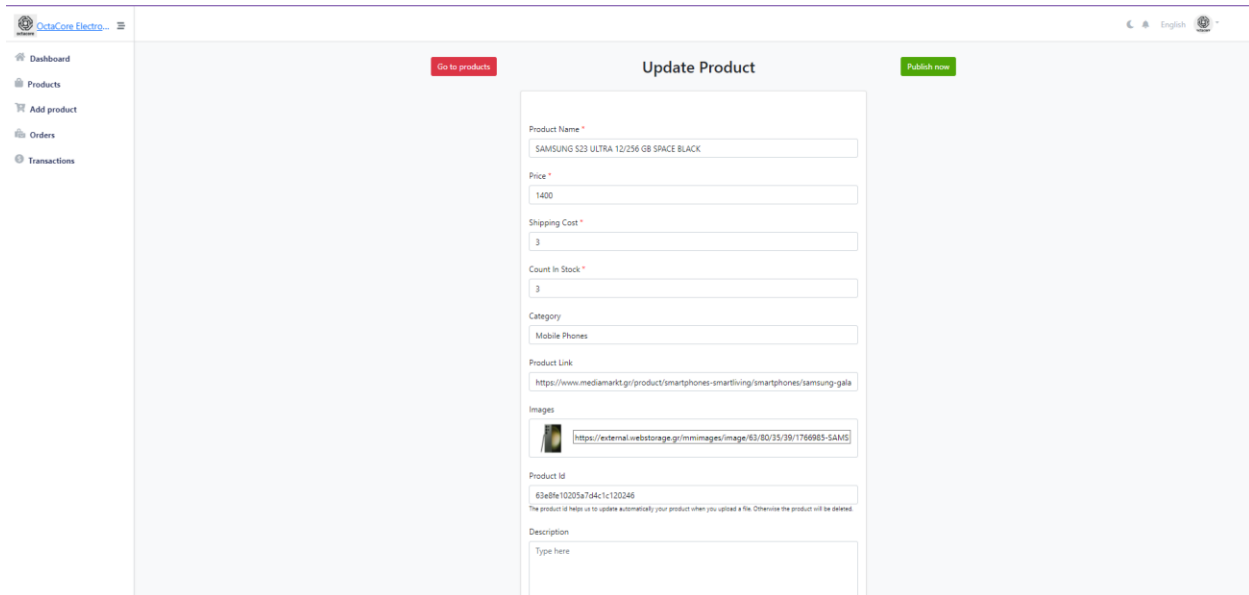
Εικόνα 4.18 – Order Details, Marked as Delivered

Όπως προαναφέρθηκε, η πλοήγηση του χρήστη γίνεται από την αριστερή στήλη, που πέρα από την επιστροφή στην αρχική σελίδα με το πλήκτρο “Dashboard” ο χρήστης έχει τις εξής επιλογές:

- Products: Επισκόπηση των προϊόντων με επιλογή φιλτραρίσματος ανάλογα την ημερομηνία δημιουργίας ή την κατηγορία. Δυνατότητα επεξεργασίας, διαγραφής και δημιουργίας νέου προϊόντος.

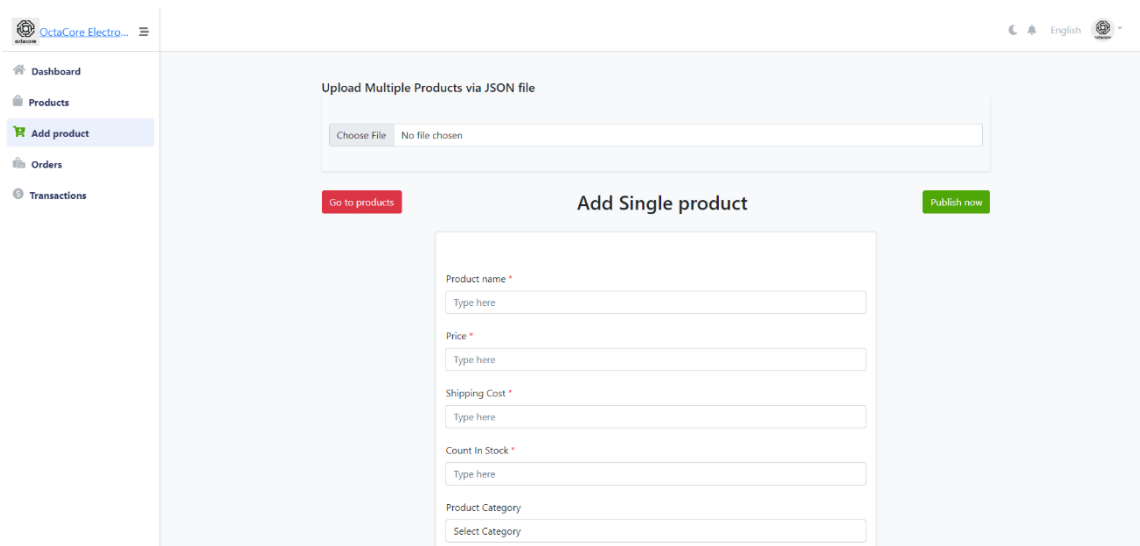


Εικόνα 4.19 – Products Page



Εικόνα 4.20 – Update/Edit Product Page

- Add product: Δημιουργία νέου προϊόντος ή μαζική ανανέωση της λίστας προϊόντων του καταστήματος μέσω ανάρτησης κατάλληλου JSON αρχείου.



Εικόνα 4.21 – Σελίδα προσθήκης νέου προϊόντος

Shipping Costs
Type here

Count In Stock *
Type here

Product Category
Select Category

Product Link
Inter Product URL from your website

Images
Inter Image URL

Product ID
Type here
The product id helps us to update automatically your product when you upload a file. Otherwise the product will be deleted.

Description
Type here

Εικόνα 4.22 – Σελίδα προσθήκης νέου προϊόντος (συνέχεια)

- Orders: Προβολή των παραγγελιών με δυνατότητα φιλτραρίσματος σχετικά με την κατάσταση της οαράδοσης της παραγγελίας. Αναλυτική προβολή κάθε παραγγελίας κατά το κλικ στο αντίστοιχο κουμπί.

Name	Email	Total	Date	Status	Action
Panasonic KX-TG2511 GRT BLACK	sakis@a.com	\$ 35.40	Aug 5, 2023 10:26 PM	Not Delivered	🔍
HIFUTURE HITIME MINI 1.57", PINK	elena@a.com	\$ 33.83	Jun 5, 2023 10:30 PM	Delivered	🔍
Huawei MateBook D 15 (3-10110U/8GB/256GB/FHD) Silver	marios@a.com	\$ 604.50	May 11, 2023 8:11 PM	Delivered	🔍
Panasonic KX-TG1611GRF PINK	marios@a.com	\$ 25.50	Mar 5, 2023 9:35 PM	Not Delivered	🔍
HAMMER HMW01 1.3", BLACK	marios@a.com	\$ 78.15	Feb 11, 2023 7:16 PM	Not Delivered	🔍
AULA ΜΗΧΑΝΙΚΟ ΠΛΗΚΤΡΟΛΟΓΙΟ F2067, RGB, BLUE SWITCH, 87 ΠΛΗΚΤΡΑ, SILVER	elena@a.com	\$ 25.15	Feb 11, 2023 7:15 PM	Not Delivered	🔍
NOKIA 5310 (2020) (DUAL SIM), RED BLACK	patrickpen@a.com	\$ 58.50	Feb 11, 2023 7:12 PM	Not Delivered	🔍
HIFUTURE HITIME MINI 1.57", PINK	patrickpen@a.com	\$ 33.83	Feb 5, 2023 9:35 PM	Delivered	🔍
XIAOMI MIBRO AIR XPW001 42MM BLACK	elena@a.com	\$ 39.82	Jan 5, 2023 9:33 PM	Not Delivered	🔍
Retro Console My Arcade Collectible Retro Burger Time Micro Player	sakis@a.com	\$ 26.40	Feb 5, 2022 9:25 PM	Not Delivered	🔍
POWERTECH ΚΙΝΗΤΟ ΘΑΞΕΘΩΝΟ SENTRY DUAL II, BLACK	marios@a.com	\$ 45.40	May 5, 2021 10:21 PM	Not Delivered	🔍

Εικόνα 4.23 – Σελίδα παραγγελιών

The screenshot shows the 'Orders' page in the OctaCore Electro... system. The left sidebar contains navigation options: Dashboard, Products, Add product, Orders (selected), and Transactions. The main content area is titled 'Orders' and features a dropdown menu set to 'Delivered'. Below the menu is a table with the following data:

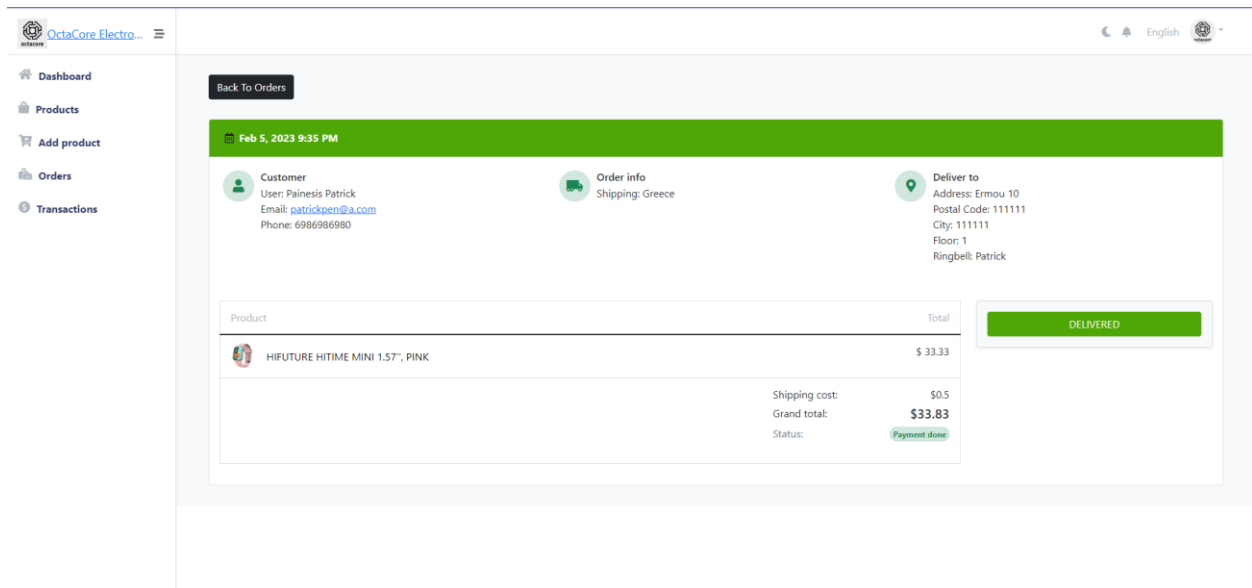
Name	Email	Total	Date	Status	Action
HIFUTURE HITIME MINI 1.57", PINK	elena@a.com	\$ 33.83	Jun 5, 2023 10:30 PM	Delivered	👁️
Huawei MateBook D 15 (i3-10110U/8GB/256GB/FHD) Silver	marios@a.com	\$ 604.50	May 11, 2023 8:11 PM	Delivered	👁️
HIFUTURE HITIME MINI 1.57", PINK	patrickpen@a.com	\$ 33.83	Feb 5, 2023 9:35 PM	Delivered	👁️

Εικόνα 4.24 – Φιλτραρισμένες παραδομένες παραγγελίες

The screenshot shows the 'Orders' page in the OctaCore Electro... system. The left sidebar contains navigation options: Dashboard, Products, Add product, Orders (selected), and Transactions. The main content area is titled 'Orders' and features a dropdown menu set to 'Not Delivered'. Below the menu is a table with the following data:

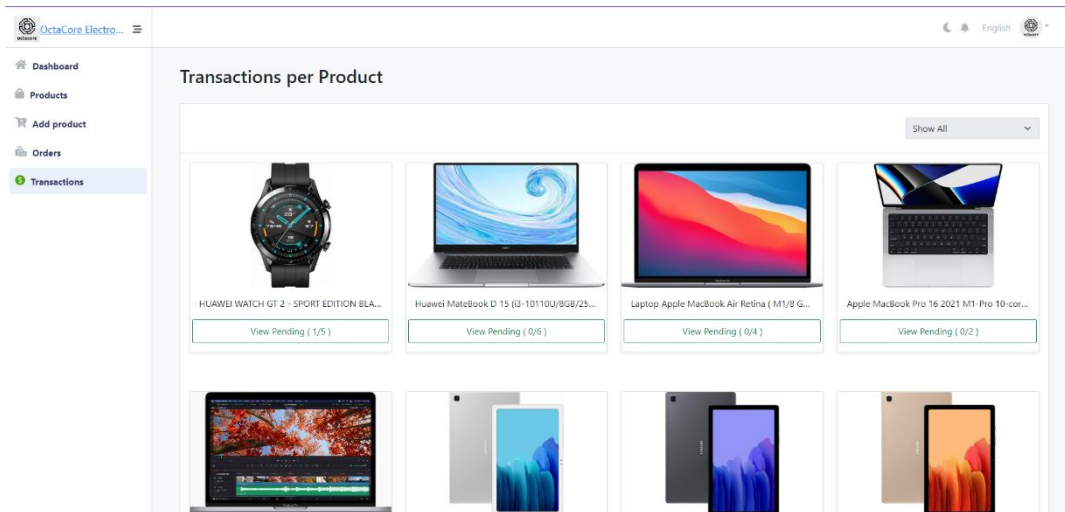
Name	Email	Total	Date	Status	Action
Panasonic KX-TG2511 GRT BLACK	sakis@a.com	\$ 35.40	Aug 5, 2023 10:26 PM	Not Delivered	👁️
Panasonic KX-TG1611GRF PINK	marios@a.com	\$ 25.50	Mar 5, 2023 9:35 PM	Not Delivered	👁️
HAMMER HMW01 1.3", BLACK	marios@a.com	\$ 78.15	Feb 11, 2023 7:16 PM	Not Delivered	👁️
AULA ΜΗΧΑΝΙΚΟ ΠΛΗΚΤΡΟΛΟΓΙΟ F2067, RGB, BLUE SWITCH, 87 ΠΛΗΚΤΡΑ, SILVER	elena@a.com	\$ 25.15	Feb 11, 2023 7:15 PM	Not Delivered	👁️
NOKIA 5310 (2020) (DUAL SIM), RED BLACK	patrickpen@a.com	\$ 58.50	Feb 11, 2023 7:12 PM	Not Delivered	👁️
XIAOMI MIBRO AIR ΧΡΑΥ001 42MM BLACK	elena@a.com	\$ 39.82	Jan 5, 2023 9:33 PM	Not Delivered	👁️
Retro Console My Arcade Collectible Retro Burger Time Micro Player	sakis@a.com	\$ 26.40	Feb 5, 2022 9:25 PM	Not Delivered	👁️
POWERTECH ΚΙΝΗΤΟ ΤΗΛΕΦΩΝΟ SENTRY DUAL II, BLACK	marios@a.com	\$ 45.40	May 5, 2021 10:21 PM	Not Delivered	👁️

Εικόνα 4.25 – Φιλτραρισμένες Μη Παραδομένες παραγγελίες



Εικόνα 4.26 – Προβολή λεπτομερειών παραγγελίας

- **Transactions:** Εμφάνιση των προϊόντων και των συναλλαγών που έχουν γίνει για αυτά, όταν βρίσκονται στη λίστα κάποιου απλού χρήστη, χωρίς να έχει ολοκληρωθεί το ποσό, και επομένως δεν έχει δημιουργηθεί παραγγελία για αυτά. Κατά το κλικ σε κάποιο transaction, βλέπουμε τους χρήστες που έχουν κάποιο προϊόν μας στη λίστα τους, και το ποσό που απομένει.

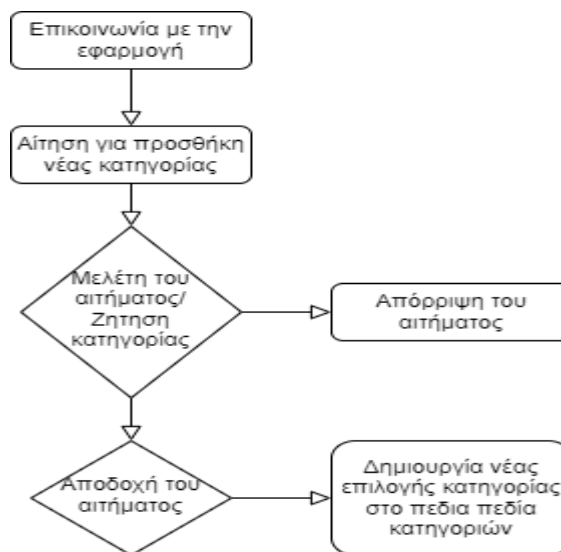


Εικόνα 4.27 – Σελίδα συναλλαγών

4.7. Προσθήκη νέας κατηγορίας

Είναι προφανές πως λόγω της multi-vendor φύση της εφαρμογής, δηλαδή την δυνατότητα πώλησης προϊόντων από πολλαπλά καταστήματα σε μία πλατφόρμα, η προσθήκη νέας κατηγορίας δεν μπορεί να γίνεται από το κάθε κατάστημα. Θα δημιουργηθεί πληθώρα κατηγοριών που πολλές φορές μπορεί να εξυπηρετεί μόνο αυτούς που της δημιούργησαν.

Για αυτό τον λόγο, οποιοδήποτε κατάστημα επιθυμεί να γίνει προσθήκη μίας νέας κατηγορίας προϊόντων, χρειάζεται να έρθει σε επικοινωνία με την εφαρμογή. Έπειτα από ανάλυση του συγκεκριμένου αιτήματος, σε ζητήματα όπως αντίστοιχα αιτήματα παρόμοιας ή ίδιας κατηγορίας, γενικότερη ζήτηση αυτής της κατηγορίας κλπ., το αίτημα είτε απορρίπτεται, είτε εγκρίνεται, με τις αντίστοιχες αλλαγές να ακολουθούν στον κώδικα, για την προσθήκη της νέας κατηγορίας.



Εικόνα 4.28 – Screenshot από το UI του δικού μας ATLAS

Κεφάλαιο 5 – Συμπεράσματα και μελλοντικές επεκτάσεις

5.1 Συμπεράσματα

Μετά τη δημιουργία ενός web app ηλεκτρονικού εμπορίου με το MERN STACK, μπορεί να

συναχθεί το συμπέρασμα ότι αυτή η στοίβα τεχνολογίας είναι ένα ισχυρό εργαλείο για τη δημιουργία εφαρμογών ιστού που να ανταποκρίνονται, να κλιμακώνονται και να είναι ασφαλείς. Το MERN STACK προσφέρει μια ολοκληρωμένη λύση για τη δημιουργία πολύπλοκων ιστοσελίδων ηλεκτρονικού εμπορίου, με τα διάφορα εξαρτήματά του να συνεργάζονται άψογα. Το front-end του έργου, που αναπτύχθηκε με το React, παρέχει μια διαισθητική και διαδραστική διεπαφή χρήστη, ενώ το back-end, που αναπτύχθηκε με το Node.js και το MongoDB, παρέχει ισχυρές δυνατότητες διαχείρισης και επεξεργασίας δεδομένων. Επιπλέον, η χρήση του πλαισίου Express διευκολύνει τον χειρισμό λειτουργιών από την πλευρά του διακομιστή. Η χρήση αυτής της στοίβας τεχνολογίας μπορεί να οδηγήσει σε μια εφαρμογή ιστού που είναι βελτιστοποιημένη για απόδοση, δυνατότητα συντήρησης και εμπειρία χρήστη. Επίσης, με την ολοκλήρωση ενός Web-App με τη στοίβα MERN, οι προγραμματιστές μπορούν να αναμένουν ότι θα έχουν αποκτήσει βαθιά κατανόηση των διαφορετικών τεχνολογιών και του τρόπου με τον οποίο συνεργάζονται. Θα έχουν μάθει πώς να αναπτύσσουν επεκτάσιμες και υψηλής απόδοσης εφαρμογές ιστού, πώς να χειρίζονται δεδομένα και βάσεις δεδομένων και πώς να σχεδιάζουν διεπαφές χρήστη που να ανταποκρίνονται και οπτικά ελκυστικές. Επιπλέον, οι προγραμματιστές θα έχουν αναπτύξει δεξιότητες στην αντιμετώπιση προβλημάτων και στην επίλυση προβλημάτων, καθώς η κατασκευή οποιουδήποτε έργου αυτού του μεγέθους είναι πιθανό να συνοδεύεται από τις δικές του προκλήσεις.

5.2 Μελλοντικές Επεκτάσεις

Από τη στιγμή που ο σκοπός του project είναι να υλοποιηθεί και να γίνει προϊόν, είναι κατανοητό πως οι πιθανές αλλαγές και μελλοντικές επεκτάσεις είναι ποικίλες, τόσο από άποψη front-end όσο και λειτουργικές. Απαραίτητη είναι η βελτίωση της ασφάλειας της εφαρμογής, και η προσθήκη κάποιου API τρίτου μέρους για πραγματοποίηση πληρωμών μέσω κάρτας. Αλλαγές επίσης θα μπορούσαν να πραγματοποιηθούν στο front-end, δηλαδή στην εμφάνιση της εφαρμογής ώστε να βελτιωθεί το UI, ενώ θα μπορούσαν να προστεθούν περισσότερες επιλογές φιλτραρίσματος, ή ειδοποιήσεις για νέες παραγγελίες, για την βελτιστοποίηση της εμπειρίας χρήσης του χρήστη. Κάποια πιο φιλόδοξη επέκταση θα ήταν η δημιουργία εφαρμογής για smartphones κινητά, για ακόμα πιο εύκολη πρόσβαση στην εφαρμογή μας. Τέλος, το API ή το module μιας τράπεζας είναι απαραίτητη προσθήκη στην εφαρμογή. Αυτή η ενότητα θα είναι υπεύθυνη για το χειρισμό των οικονομικών συναλλαγών μεταξύ των χρηστών και της εφαρμογής. Θα διαχειριζόταν την κατάθεση κεφαλαίων από τους συνεισφέροντες μέχρι να ολοκληρωθεί η πλήρης τιμή του προϊόντος και στη συνέχεια θα πραγματοποιούσε την πληρωμή στον πωλητή. Επιπλέον, εάν το προϊόν αφαιρεθεί από τη λίστα επιθυμιών ή διαγραφεί από την εφαρμογή, η ενότητα θα είναι επίσης υπεύθυνη για την επιστροφή των χρημάτων των συντελεστών. Αυτή η δυνατότητα όχι μόνο θα διασφάλιζε την ασφαλή διαχείριση των οικονομικών συναλλαγών, αλλά θα παρέχει επίσης διαφάνεια και εμπιστοσύνη στους χρήστες της εφαρμογής.

Βιβλιογραφία

- [1] Mozilla Developer Network, "HTML: HyperText Markup Language.", χ.η.
- [2] Mozilla Developer Network, "Learn CSS", 2021.
- [3] Jordana A., "What is React & How Does it Actually Work?", 2022.
- [4] Durga Prasad Acharya, "15 Best React Tutorials and Resources for Developers" , 2023.
- [5] Joel Olawanle , "Axios React – How to Make Get, Post, and Delete API Requests" , 2022.
- [6] Jarrett Retz , "How to Display API Data Using Axios with React", 2022.
- [7] Dickson Boateng, "Asynchronous Programming in JavaScript – Guide for Beginners", 2023. Node.js, "Introduction Node.js", 2021.
- [8] Priyesh Patel, "What exactly is Node.js", 2018.
- [9] Martin Heller, "What is Node.js? The JavaScript runtime explained", 2022. Npm Docs, "About npm", 2022.
- [10] Anubhav Sharma, "What Is Express JS In Node JS?", 2023.
- [11] W3Schools, "What is JSON?", 2023.
- [12] Elmasri, Ramez, and Shamkant B. Navathe. , "Fundamentals of Database Systems.", 2016.
- [13] Laura Shiff, Walker Rowe, "SQL vs NoSQL Databases: What's The Difference?", 2018.
- [14] IBM, "NoSQL databases explained", χ.η.
- [15] GeeksForGeeks, "Difference between SQL and NoSQL", 2022.
- [16] MongoDB, "MongoDB Overview" MongoDB", 2023.
- [17] Mongoose, "Mongoose Documentation", 2023.
- [18] GeeksForGeeks, "Mern Stack", 2021.
- [19] Fred Churchville, "User interface (UI)", 2021