



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

**Σχεδιασμός και ανάπτυξη ψηφιακής υπηρεσίας σύστασης
βιβλίων**

Διπλωματική Εργασία

Αντρέας Τσαγγάρης

Επιβλέπων/πουσα: Τσαλαπάτα Χαρίκλεια

Μήνας 2022



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

**Σχεδιασμός και ανάπτυξη ψηφιακής υπηρεσίας σύστασης
βιβλίων**

Διπλωματική Εργασία

Αντρέας Τσαγγάρης

Επιβλέπων/πουσα: Τσαλαπάτα Χαρίκλεια

Μήνας 2022



UNIVERSITY OF THESSALY
SCHOOL OF ENGINEERING
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

**The design and implementation of an e-commerce service
for online book sales**

Diploma Thesis

Andreas Tsangaris

Supervisor: Tsalapata Hariklia

Month 2022

Εγκρίνεται από την Επιτροπή Εξέτασης:

Επιβλέπων/πουσα **Τσαλαπάτα Χαρίκλεια**

Ε.ΔΙ.Π, Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

Μέλος **Φεύγας Αθανάσιος**

Ε.ΔΙ.Π, Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

Μέλος **Πλέσσας Φώτιος**

Αναπληρωτής Καθηγητής, Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, 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, αποτελεί αποκλειστικά προϊόν προσωπικής μου εργασίας, δεν προσβάλλει κάθε μορφής δικαιώματα διανοητικής ιδιοκτησίας, προσωπικότητας και προσωπικών δεδομένων τρίτων, δεν περιέχει 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/Η Δηλών/ούσα

Αντρέας Τσαγγάρης

Περίληψη

Σε πολλές χώρες, η πρόσβαση σε βιβλία και πληροφορίες είναι περιορισμένη λόγω έλλειψης φυσικών βιβλιοθηκών και έλλειψης ηλεκτρονικών πηγών. Αυτή η έλλειψη πρόσβασης σε πληροφορίες μπορεί να έχει σημαντικές αρνητικές συνέπειες στην εκπαίδευση, στην οικο-νομική και προσωπική ανάπτυξη του κάθε ατόμου. Για την αντιμετώπιση αυτού του προβλήματος, σκοπός της παρούσας διπλωματικής εργασίας είναι να σχεδιάσει και να υλοποιήσει μια πλατφόρμα μηχανής αναζήτησης για βιβλία που επιτρέπει στους χρήστες να αναζητούν και να έχουν πρόσβαση σε ένα ευρύ φάσμα βιβλίων ηλεκτρονικά. Η πλατφόρμα έχει δημιουργηθεί μέσω της `node.js` για το backend, ενώ χρησιμοποιεί ένα frontend βασισμένο σε JavaScript, σχεδιασμένο με το πλαίσιο Materialize και ενσωματωμένο με HTML & CSS. Παράλληλα, μελετήθηκαν και χρησιμοποιήθηκαν web-scraping techniques όπως `puppeteer` και `cheerio`. Οι τεχνολογίες που χρησιμοποιούνται για την υλοποίησή του συζητούνται και εξετάζονται στην παρούσα διατριβή, μαζί με την ανάλυση κάθε σταδίου εφαρμογής. Τέλος, παραθέτουμε ορισμένες προτάσεις για τη μελλοντική ανάπτυξη της εφαρμογής.

Η πλατφόρμα μας έχει σχεδιαστεί ώστε να είναι φιλική προς το χρήστη και εύκολη στη χρήση, με προηγμένες δυνατότητες αναζήτησης που βοηθούν τους χρήστες να βρουν τα βιβλία που χρειάζονται. Εφαρμόστηκαν επίσης μέτρα για τη διασφάλιση της ασφάλειας και της ιδιωτικής ζωής των χρηστών μας. Μέσω εκτεταμένων δοκιμών και αξιολογήσεων από τους χρήστες, αποδεικνύουμε την αποτελεσματικότητα και τη χρησιμότητα της πλατφόρμας μας στην παροχή πρόσβασης σε βιβλία και πληροφορίες σε μια αναπτυσσόμενη χώρα. Το έργο μας όχι μόνο αντιμετωπίζει μια επιτακτική ανάγκη στις χώρες αυτές, αλλά έχει επίσης τη δυνατότητα να χρησιμεύσει ως πρότυπο για άλλες παρόμοιες πρωτοβουλίες στο μέλλον.

Λέξεις-κλειδιά:

Nodejs , MySQL , Express, Materialize CSS, JavaScript.

Abstract

In many developing countries, access to books and information is limited due to a lack of physical libraries and a lack of online resources. This lack of access to information can have significant negative consequences on education, economic development, and personal growth. To address this problem, the purpose of this thesis is to design and implement a search engine platform for books that allows users to search and access a wide range of books electronically. The platform's backend has been developed via node.js and utilizes a JavaScript-based frontend, designed with Materialize framework and integrated with HTML & CSS. At the same time, web-scraping techniques such as puppeteer and cheerio were studied and used. The technologies employed for its implementation are discussed and examined in this article, along with a breakdown of each application stage. Finally, we offer some suggestions for the application's future development.

The technologies employed for its execution are discussed and examined in this article, along with a breakdown of each application step. Finally, we provide some suggestions for the application's future development.

Our platform is designed to be user-friendly and easy to use, with advanced search capabilities and a variety of filters to help users find the books they need. We also implemented measures to ensure the security and privacy of our users. Through extensive user testing and evaluations, we demonstrate the effectiveness and usefulness of our platform in providing access to books and information in a developing country. Our work not only addresses a pressing need in these countries, but also has the potential to serve as a model for other similar initiatives in the future

Keywords:

Nodejs , MySQL , Express, Materialize CSS, JavaScript.

Πίνακας περιεχομένων

Περίληψη	x
Abstract	xi
Πίνακας περιεχομένων	xiii
Κατάλογος σχημάτων	xv
Συντομογραφίες	xvii
1 Εισαγωγή	1
1.1 Αντικείμενο της διπλωματικής	1
1.2 Οργάνωση του τόμου	2
2 Backend	5
2.1 Back-end developers	6
2.2 JavaScript	7
2.2.1 Node.js	9
2.2.2 Express	10
2.3 API	12
2.3.1 JSON	13
3 Databases	15
3.0.1 Χαρακτηριστικά σχεσιακών βάσεων δεδομένων, βάσεις NoSQL, και βάσεων δεδομένων στη μνήμη	16
3.0.2 MySQL σε σχέση με άλλες databases	17

4 Frontend	19
4.1 Σύζευξη HTML,CSS και JS	20
4.1.1 HTML	21
4.1.2 HTML vs HTML5	21
4.1.3 CSS	21
4.1.4 JS	22
4.1.5 JQuery	23
4.2 Materieliaze CSS	24
5 Ανάπτυξη του BookshelfCy	27
5.1 Συνεργασία συνεργατών	28
5.2 Συλλογή δεδομένων	28
5.2.1 Web scraping methods and modules using node.js	28
5.3 Backend	35
5.4 Routes	38
5.4.1 Results	39
5.5 Frontend	40
5.5.1 Optimization	41
5.5.2 Περιήγηση user	43
5.5.3 Product page	45
6 Συμπεράσματα	47
6.1 Σύνοψη και συμπεράσματα	47
6.2 Μελλοντικές επεκτάσεις	47
Βιβλιογραφία	49
ΠΑΡΑΡΤΗΜΑΤΑ	53

Κατάλογος σχημάτων

2.1	Μέρη του backend.	
	<i>source: https://ddi-dev.com</i>	6
2.2	Αναγκαίες δεξιότητες backend developer.	
	<i>source: https://www.interviewbit.com</i>	7
2.3	Δεξιότητες προγραμματισμού που επιθυμούν οι διευθυντές προσλήψεων παγκοσμίως.	
	<i>source: https://codingnomads.co</i>	9
2.4	10 Κορυφαία web frameworks και τεχνολογίες που ψηφίστηκαν από επαγγελματίες προγραμματιστές στην έρευνα για τους προγραμματιστές του stackoverflow 2022.	
	<i>source: survey.stackoverflow.co</i>	10
2.5	Αυτή η εφαρμογή εκκινεί έναν διακομιστή και ακούει στη θύρα 3000 για συνδέσεις. Η εφαρμογή απαντά με "Hello World!" για αιτήσεις στη ριζική διεύθυνση URL (/) ή στη διαδρομή. Για κάθε άλλη διαδρομή, θα απαντήσει με 404 Not Found.	
	<i>source: https://expressjs.com</i>	11
2.6	Πως δουλεύουν οι APIs.	
	<i>source: https://www.g2.com</i>	12
5.1	Πίνακας database για συλλογή ονόματος, εικόνας, περίληψης και isbn.	31
5.2	Παράδειγμα χρήσης ssr παράλληλα με το cheerio για την συλλογή δεδομένων.	33
5.3	Επιστροφή σελίδας σε html για την απόξεση της.	34
5.4	Πίνακας database για συλλογή κόστους και συνδέσμου.	35
5.5	Εδραίωση σύνδεσης μεταξύ MySQL database και Node.js.	
	<i>source: https://w3school.com</i>	35

5.6 Στην περίπτωση της ιστοσελίδας που δημιουργήσαμε, η σύνδεση μεταξύ MySQL και Node.js φαίνεται κάπως έτσι	36
5.7 Προεπιλογή πορτ σελίδας.	36
5.8 Ορισμός φακέλου views ως μηχανή προβολής εις αρχείων.	37
5.9 Χρήση app.get() και res.render()	37
5.10 Συλλογή δεδομένων μέσω Sqlquery.	38
5.11 Προσπέλαση του φακέλου public, καθώς και τους φακέλους js,css,images που βρίσκονται σε αυτό.	38
5.12 Routing to Search.ejs	39
5.13 Αλλαγή url	40
5.14 Εύρεση όλων των βιβλίων από την database	40
5.15 Αρχική Σελίδα	41
5.16 Αρχική Σελίδα για tablets και smartphones	42
5.17 Αρχική Σελίδα για tablets και smartphones	43
5.18 Εύρεση βιβλίων.	44
5.19 Δημιουργία autocomplete function μέσω javascript, όπου data περιέχει πληροφορίες του κάθε βιβλίου απο το database και limit το όριο επιλογών που είναι διαθέσιμες	44
5.20 Προβολή όλων των σχετικών βιβλίων.	45
5.21 Επιλογή συγκεκριμένου βιβλίου.	45
5.22 Product page ex.1.	46
5.23 Product page ex.2.	46

Συντομογραφίες

κ.λπ.	και λοιπά
JS	Javascript
CSS	Cascading Style Sheets
SQL	Structured Query Language
NoSQL	Not only SQL
HTML	HyperText Markup Language
API	Application Programming Interface

Κεφάλαιο 1

Εισαγωγή

Στις μέρες μας, οι ηλεκτρονικές αγορές έχουν γίνει κανόνας για πολλούς καταναλωτές. Ωστόσο, σε πολλές χώρες, οι επιλογές για ηλεκτρονικές αγορές συχνά περιορίζονται σε ξένους ιστότοπους όπως το Amazon, το eBay και το Skrutz, οι οποίοι μπορεί να μην προσφέρουν την ίδια επιλογή τοπικών προϊόντων σε σύγκριση με τους εγχώριους ιστότοπους. Αυτό ισχύει ιδιαίτερα για τον κλάδο του βιβλίου, όπου τα τοπικά βιβλιοπωλεία μπορεί να μην έχουν διαδικτυακή παρουσία ή τους πόρους για να ανταγωνιστούν τις μεγαλύτερες, ξένες εταιρείες.

1.1 Αντικείμενο της διπλωματικής

Για την αντιμετώπιση αυτού του προβλήματος, προτείνουμε μια λύση με τη μορφή μιας πλατφόρμας μηχανής αναζήτησης για τοπικά βιβλία. Η πλατφόρμα μας, επιτρέπει στους χρήστες να αναζητούν και να έχουν πρόσβαση σε ένα ευρύ φάσμα βιβλίων ηλεκτρονικά, παρέχοντας μια βολική και αξιόπιστη πηγή πληροφοριών σε χώρες όπου οι εν λόγω πόροι είναι σπάνιοι. Ο ιστότοπος αυτός έχει ως στόχο να παρέχει μια πλατφόρμα στα τοπικά βιβλιοπωλεία της Κύπρου για να προβάλλουν τα προϊόντα τους και για να μπορούν οι καταναλωτές να βρίσκουν και να αγοράζουν εύκολα βιβλία από τοπικά βιβλιοπωλεία. Με τη δημιουργία ενός κεντρικού σημείου για τα τοπικά βιβλιοπωλεία να διαθέτουν τα προϊόντα τους, ο ιστότοπος αυτός θα διευκολύνει τους καταναλωτές να ανακαλύψουν και να υποστηρίξουν τις τοπικές επιχειρήσεις, προωθώντας την ανάπτυξη της τοπικής οικονομίας. Επιπλέον, ο ιστότοπος θα διευκολύνει τους καταναλωτές να εντοπίζουν και να αγοράζουν τα φθηνότερα βιβλία από τοπικά βιβλιοπωλεία χωρίς να χρειάζονται να περιηγηθούν σε δεκάδες σελίδες.

Στην παρούσα διατριβή περιγράφεται ο σχεδιασμός και η υλοποίηση της πλατφόρμας και αποδεικνύεται η αποτελεσματικότητα και τη χρησιμότητά της μέσω εκτεταμένων δοκιμών και αξιολογήσεων από τους χρήστες. Το έργο αυτό όχι μόνο καλύπτει μια πειστική ανάγκη στην χώρα μας, αλλά έχει επίσης τη δυνατότητα να χρησιμεύσει ως πρότυπο για άλλες παρόμοιες πρωτοβουλίες στο μέλλον.

1.2 Οργάνωση του τόμου

Κεφάλαιο 2: Ανάλυση backend

- Επισκόπηση της αρχιτεκτονικής backend και των τεχνολογιών που χρησιμοποιήθηκαν (Node.js, Express).
- Συζήτηση των μεθόδων και τεχνικών για τη διαχείριση του λογισμικού και τη διαχείριση δεδομένων στην πλευρά του server.
- Εξηγήσεις για τον προσανατολισμό του server και τη σχεδίαση των API.

Κεφάλαιο 3: Ανάλυση βάσης δεδομένων

- Επισκόπηση των διαφόρων τύπων βάσεων δεδομένων και των περιπτώσεων τους.
- Επίκεντρο στην συγκεκριμένη τεχνολογία βάσης δεδομένων που χρησιμοποιήθηκε στο project (MySQL).
- Συζήτηση του σχεδιασμού της βάσης δεδομένων και των ερωτημάτων που χρησιμοποιούνται για την αλληλεπίδραση με τα δεδομένα.

Κεφάλαιο 4: Ανάλυση frontend

- Επισκόπηση της αρχιτεκτονικής του frontend και των τεχνολογιών που χρησιμοποιούνται (JavaScript, HTML, CSS, Materialize).
- Ανάλυση της HTML: Επισκόπηση της δομής και του σχεδιασμού του δικτυακού τόπου, χρήση ετικετών και στοιχείων HTML για τη δημιουργία σημασιολογικής και προσβάσιμης διάταξης.

- Ανάλυση της JavaScript: Συζήτηση των βιβλιοθηκών και των πλαισίων που χρησιμοποιούνται, όπως η jQuery, για την υλοποίηση διαδραστικών και δυναμικών χαρακτηριστικών στον ιστότοπο.
- Ανάλυση της CSS: Επεξήγηση του τρόπου με τον οποίο χρησιμοποιείται η CSS για τον έλεγχο της διάταξης και της παρουσίασης του ιστότοπου, καθώς και της χρήσης του Materialize CSS ως πλαίσιο για τη δημιουργία ενός ευέλικτου και οπτικά ελκυστικού σχεδιασμού.

Κεφάλαιο 5: Ανάλυση ιστοτόπου

- Επισκόπηση του συνολικού σχεδιασμού και της λειτουργικότητας του δικτυακού τόπου.
- Συζήτηση της υλοποίησης του κώδικα τόσο για το backend όσο και για το frontend και πώς προέκυψαν τα αποτελέσματα.
- Επεξήγηση του τρόπου επικοινωνίας του διακομιστή και του πελάτη και του τρόπου μεταφοράς και αποθήκευσης των δεδομένων.

Κεφάλαιο 6: Συμπεράσματα

- Σύνοψη των κύριων ευρημάτων και συμπερασμάτων της διατριβής.
- Προτάσεις για μελλοντική έρευνα και βελτιώσεις.
- Γενικό συμπέρασμα της διατριβής και η συνολική της σημασία.

Κεφάλαιο 2

Backend

Το back-end αναφέρεται στην λειτουργία που εμφανίζεται στο παρασκήνιο ενός ιστοτόπου ή μιας εφαρμογής. Αναφέρεται σε όλα τα πράγματα που γίνονται στον εξυπηρετητή (server-side), ενώ το front-end (client-side) αναφέρεται σε ό,τι ο χρήστης εμφανίζει και συναντά στον περιηγητή ιστού. [1].

Το back-end είναι υπεύθυνο για εργασίες όπως την αποθήκευση και την οργάνωση δεδομένων, την επεξεργασία αιτημάτων που καταγράφονται από το front-end και την υποβολή των κατάλληλων δεδομένων ή σελίδων στο χρήστη. Αυτό συνήθως συμπεριλαμβάνει την εργασία με εξυπηρετητές, βάσεις δεδομένων και γλώσσες προγραμματισμού, όπως τα PHP, Ruby, Python και τα τελευταία χρόνια την Javascript [1].

Η ανάπτυξη back-end αναφέρεται στην ανάπτυξη της πλευράς του διακομιστή μιας διαδικτυακής εφαρμογής. Περιλαμβάνει την εργασία με τεχνολογίες που τροφοδοτούν την εφαρμογή ιστού αλλά δεν είναι ορατές στον τελικό χρήστη [2].



Σχήμα 2.1: Μέρη του backend.

source: <https://ddi-dev.com>

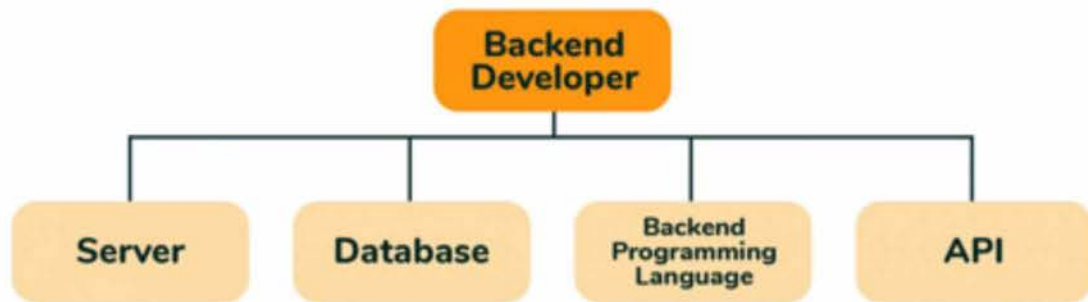
2.1 Back-end developers

Οι προγραμματιστές back-end εργάζονται στο server-side των εφαρμογών web και είναι υπεύθυνοι για την ενσωμάτωση του front-end με το back-end. Αυτό περιλαμβάνει εργασίες όπως τη σύνδεση της εφαρμογής με μία βάση δεδομένων, τη ρύθμιση της server-side λογικής και την ενσωμάτωση APIs (Application Programming Interfaces) [3].

Ορισμένες κοινές εργασίες για έναν back-end προγραμματιστή περιλαμβάνουν [4]:

- Την υλοποίηση της λογικής της εφαρμογής και των στοιχείων της πλευράς του διακομιστή.
- Ενσωμάτωση της εφαρμογής με εξωτερικά API και υπηρεσίες.
- Διαχείριση και χειρισμός βάσεων δεδομένων.
- Εφαρμογή μέτρων ασφάλειας και ελέγχου ταυτότητας.

- Ισχυρές δεξιότητες επίλυσης προβλημάτων και καλή γνώση διάφορων γλωσσών προγραμματισμού. Ορισμένες από τις βασικές δεξιότητες και τεχνολογίες που χρησιμοποιούνται συνήθως στην ανάπτυξη back-end περιλαμβάνουν [4]:
 1. Γλώσσες από την πλευρά του διακομιστή, όπως η Java, η Python, η PHP και η Javascript.
 2. Πλαίσια ιστού όπως το Django, το Flask , το Spring και την Express.
 3. Βάσεις δεδομένων και συστήματα διαχείρισης βάσεων δεδομένων όπως η MySQL, η PostgreSQL, η MongoDB και η MariaDB.
 4. Συστήματα ελέγχου εκδόσεων όπως το Git.



Σχήμα 2.2: Αναγκαίες δεξιότητες backend developer.

source: <https://www.interviewbit.com>

2.2 JavaScript

Η JavaScript είναι μία γλώσσα προγραμματισμού που χρησιμοποιείται συχνά για την κατασκευή εφαρμογών front-end και back-end.

Στο front-end, η JavaScript χρησιμοποιείται για να προσθέτει δυνατότητες και παραγωγικότητα σε ιστοσελίδες. Συνήθως τρέχει στον περιηγητή ιστού του χρήστη και επιτρέπει στους κατασκευαστές να δημιουργούν δυναμικά και παραγωγικά user-interface.

Στο back-end, η JavaScript μπορεί να χρησιμοποιηθεί για τη κατασκευή εφαρμογών server-side χρησιμοποιώντας περιβάλλοντα εκτέλεσης όπως η Node.js. Αυτό επιτρέπει στους κατασκευαστές να χρησιμοποιούν την JavaScript για το front-end και το back-end μίας εφαρμογής web, κάνοντάς ευκολότερη την κοινή χρήση κώδικα [5].

Σύμφωνα με έρευνα που διεξήγαγε ο ιστότοπος Indeed, η JavaScript ήταν η πιο περιζήτητη γλώσσα για τους διευθυντές προσλήψεων το 2020. Μερικοί λόγοι για τους οποίους η JavaScript μπορεί να είναι ιδιαίτερα δημοφιλής μεταξύ των διευθυντών σε σύγκριση με άλλες γλώσσες [6, 7]:

- Συχνή χρήση και καλή υποστήριξη: Η JavaScript είναι μια από τις πιο δημοφιλείς προγραμματιστικές γλώσσες και υποστηρίζεται από όλους τους σύγχρονους περιηγητές ιστού. Αυτό την κάνει μια καλή επιλογή για την κατασκευή εφαρμογών web, καθώς επιτρέπει στους κατασκευαστές να επιτύχουν μία μεγάλη κοινότητα χρηστών χωρίς να δημιουργούνται ανησυχίες για ζητήματα συμβατότητας.
- Ευελιξία: Η JavaScript μπορεί να χρησιμοποιηθεί σε full-stack application, που την καθιστά μια καλή επιλογή για τη δημιουργία εφαρμογών πλήρους επιπέδου. Μπορεί επίσης να χρησιμοποιηθεί για την κατασκευή εφαρμογών για κινητά τηλέφωνα, εφαρμογών γραφείου και παιχνιδιών καθώς και για μια μεγάλη ποικιλία έργων.
- Οικοσύστημα εργαλείων και βιβλιοθηκών: Η JavaScript διαθέτει μια μεγάλη και ενεργή ομάδα προγραμματιστών, η οποία έχει συνεισφέρει μια μεγάλη ποικιλία εργαλείων και βιβλιοθηκών που μπορούν να χρησιμοποιηθούν για την επιτάχυνση της ανάπτυξης. Αυτό περιλαμβάνει βιβλιοθήκες για εργασίες όπως ο χειρισμός αιτημάτων HTTP, η εργασία με ημερομηνίες και ώρες, η ανάλυση και ο χειρισμός δεδομένων.
- Καλή για αρχάριους: Η JavaScript είναι μια γλώσσα υψηλού επιπέδου με σχετικά απλό συντακτικό, κάτι που την αναδεικνύει ως μια καλή επιλογή για αρχάριους που μόλις μαθαίνουν να προγραμματίζουν.



Σχήμα 2.3: Δεξιότητες προγραμματισμού που επιθυμούν οι διευθυντές προσλήψεων παγκοσμίως.

source: <https://codingnomads.co>

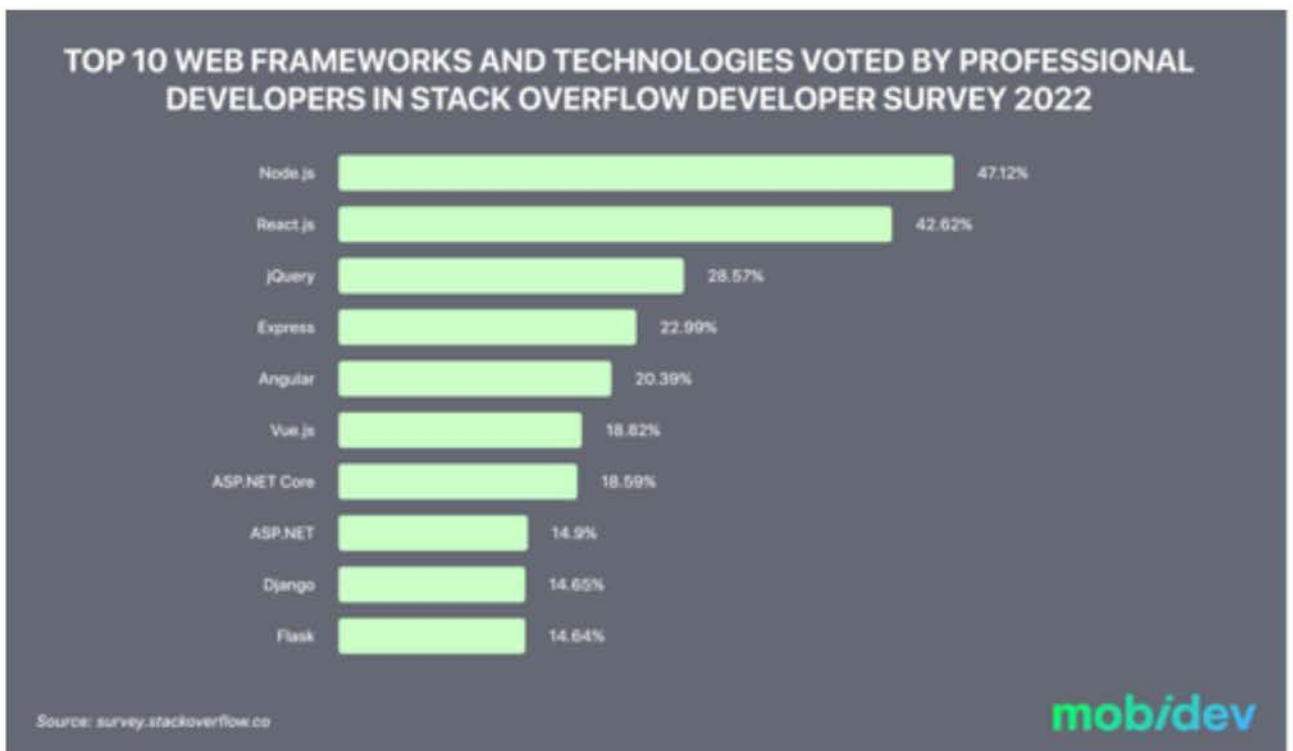
2.2.1 Node.js

Η Node.js είναι ένα περιβάλλον εκτέλεσης ανοικτού κώδικα που επιτρέπει στους προγραμματιστές να εκτελούν κώδικα JavaScript στην πλευρά του διακομιστή. Χτίστηκε πάνω στη μηχανή JavaScript του Chrome και επιτρέπει στους προγραμματιστές να δημιουργούν εφαρμογές από την πλευρά του διακομιστή χρησιμοποιώντας JavaScript [8].

Η Node.js έχει πολλά πλεονεκτήματα που αναδεικνύουν την χρησιμότητα της ως μέρος του οικοσυστήματος JavaScript [9]:

- Επιτρέπει στους προγραμματιστές να χρησιμοποιούν μια ενιαία γλώσσα (JavaScript), γεγονός που καθιστά ευκολότερη την κοινή χρήση κώδικα και τη δημιουργία εφαρμογών πλήρους επιπέδου.

- Η Node.js χρησιμοποιεί ένα μοντέλο I/O που βασίζεται σε μη αποκλειστική επεξεργασία (non-dedicated processing). Είναι δηλαδή κατάλληλη για εφαρμογές υψηλής απόδοσης καθώς και για κατασκευή εφαρμογών web σε πραγματικό χρόνο, όπως εφαρμογές συνομιλίας και παιχνίδια online.
- Διαθέτει μια μεγάλη και ενεργή κοινότητα προγραμματιστών, η οποία έχει συνεισφέρει με μια μεγάλη ποικιλία εργαλείων και βιβλιοθηκών που μπορούν να χρησιμοποιηθούν για την περαιτέρω επεξεργασία ενός συστήματος.
- Η Node.js επιτρέπει την εύκολη δημιουργία εφαρμογών δικτύου, κάνοντας την ιδανική επιλογή για την κατασκευή των microservices και μεγάλων συστημάτων.



Σχήμα 2.4: 10 Κορυφαία web frameworks και τεχνολογίες που ψηφίστηκαν από επαγγελματίες προγραμματιστές στην έρευνα για τους προγραμματιστές του stack overflow 2022.

source: survey.stackoverflow.co

2.2.2 Express

Ένα δημοφιλές πλαίσιο που χρησιμοποιείται συχνά σε συνδυασμό με την Node.js είναι η Express. Η Express είναι ένα framework για τη Node.js που παρέχει ένα σύνολο χαρακτηριστικών και λειτουργιών για τη δημιουργία εφαρμογών ιστού. [10, 11].

Η χρήση της Express σε συνδυασμό με την Node.js μπορεί να διευκολύνει τη δημιουργία εφαρμογών από την πλευρά του διακομιστή με JavaScript. Μαζί, αυτά τα εργαλεία παρέχουν μια ισχυρή και ευέλικτη πλατφόρμα για τη δημιουργία εφαρμογών ιστού που μπορούν να χρησιμοποιηθούν για τα πάντα, από απλές ιστοσελίδες μέχρι σύνθετα API [10].

Ένα από τα βασικά χαρακτηριστικά της Express είναι το σύστημα δρομολόγησης (routing), το οποίο επιτρέπει στους προγραμματιστές να ορίζουν διαφορετικά τελικά σημεία και μεθόδους HTTP μέσα στην εφαρμογή τους. Αυτό διευκολύνει τη δημιουργία API που μπορούν να χειριστούν διαφορετικούς τύπους αιτημάτων, όπως GET, POST, PUT και DELETE [11].

```
const express = require('express' 4.18.2 )
const app = express()
const port = 3000

app.get('/', (req, res) => {
  res.send('Hello World!')
})

app.listen(port, () => {
  console.log(`Example app listening on port ${port}`)
})
```

Σχήμα 2.5: Αυτή η εφαρμογή εκκινεί έναν διακομιστή και ακούει στη θύρα 3000 για συνδέσεις. Η εφαρμογή απαντά με "Hello World!" για αιτήσεις στη ριζική διεύθυνση URL (/) ή στη διαδρομή. Για κάθε άλλη διαδρομή, θα απαντήσει με 404 Not Found.

source: <https://expressjs.com>

Η Express παρέχει επίσης έναν αριθμό συναρτήσεων middleware που μπορούν να χρησιμοποιηθούν για την τροποποίηση των εισερχόμενων αιτήσεων και των εξερχόμενων απαντήσεων. Αυτές οι συναρτήσεις μπορούν να χρησιμοποιηθούν για την προσθήκη προσαρμοσμένης λογικής σε μια εφαρμογή, όπως έλεγχος ταυτότητας, επικύρωση ή χειρισμός σφαλμάτων [11].

Η Express έχει σχεδιαστεί ως ένα πλαίσιο το οποίο είναι ελαφρύ και εύκολο στη χρήση, που την καθιστά μια καλή επιλογή για τη δημιουργία απλών, γρήγορων εφαρμογών. Διαθέτει μια μεγάλη και ενεργή κοινότητα, η οποία προσφέρει μεγάλη ποικιλία από add-ons που μπορούν να χρησιμοποιηθούν για την βελτίωση της λειτουργικότητας σε μια εφαρμογή [11].

Ορισμένες κοινές περιπτώσεις χρήσης της Express περιλαμβάνουν την κατασκευή REST APIs, την απόδοση από την πλευρά του διακομιστή για εφαρμογές μίας σελίδας και την εξυπηρέτηση στατικών αρχείων. Μπορεί να χρησιμοποιηθεί σε συνδυασμό με άλλα εργαλεία

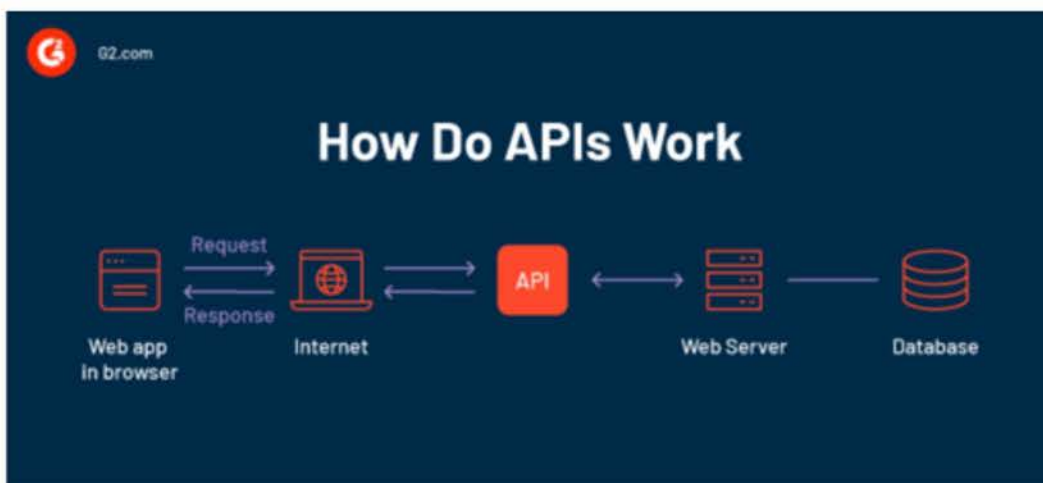
και τεχνολογίες, όπως μια βάση δεδομένων, μια μηχανή προτύπων ή ένα πλαίσιο front-end, για τη δημιουργία εφαρμογών ιστού πλήρους επιπέδου [11].

2.3 API

Ένα API, ή αλλιώς διεπαφή προγραμματισμού εφαρμογών, είναι ένα σύνολο κανόνων και πρωτοκόλλων που καθορίζει τον τρόπο με τον οποίο δύο συστήματα λογισμικού μπορούν να επικοινωνούν μεταξύ τους. Ένα API ορίζει τον τρόπο με τον οποίο ένας προγραμματιστής μπορεί να ζητήσει δεδομένα ή υπηρεσίες από ένα άλλο σύστημα, και πώς αυτό το σύστημα μπορεί να ανταποκριθεί σε αυτά τα αιτήματα [12].

Τα API μπορούν να χρησιμοποιηθούν για την ενσωμάτωση διαφορετικών συστημάτων, τη δημιουργία προσαρμοσμένων εφαρμογών ή για να επιτρέψουν σε τρίτους προγραμματιστές να δημιουργήσουν εφαρμογές που κάνουν χρήση υφιστάμενων δεδομένων ή λειτουργιών [12].

Επιπρόσθετα, χρησιμοποιούν ένα μοντέλο αίτησης-απόκρισης, στο οποίο ένας προγραμματιστής στέλνει ένα αίτημα σε ένα API, και το API στέλνει πίσω μια απάντηση με τα ζητούμενα δεδομένα ή το αποτέλεσμα της ζητούμενης λειτουργίας. Τα API μπορούν να παρέχονται μέσω του διαδικτύου με χρήση HTTP ή να χρησιμοποιούν άλλα πρωτόκολλα, όπως WebSockets ή gRPC [13].



Σχήμα 2.6: Πως δουλεύουν οι APIs.

source: <https://www.g2.com>

Τα API χρησιμοποιούν συχνά JSON για την αποστολή δεδομένων μεταξύ του διακομιστή και του πελάτη. Όταν ένας πελάτης στέλνει ένα αίτημα σε ένα API, μπορεί να περιλαμβάνει

ένα ωφέλιμο φορτίο JSON με δεδομένα που πρέπει να επεξεργαστεί ο διακομιστής. Ο διακομιστής μπορεί στη συνέχεια να απαντήσει με ένα ωφέλιμο φορτίο JSON που περιέχει τα δεδομένα ή το αποτέλεσμα της ζητούμενης λειτουργίας.

2.3.1 JSON

Το JSON (JavaScript Object Notation) είναι ένα ελαφρύ μορφότυπο ανταλλαγής δεδομένων που χρησιμοποιείται συχνά σε συνδυασμό με τα APIs. Είναι ένα μορφότυπο βασισμένο σε κείμενο που έχει σχεδιαστεί να είναι εύκολο για τους ανθρώπους να διαβάσουν και να γράψουν, και εύκολο για τις μηχανές να αναλύσουν και να δημιουργήσουν [14].

Το JSON βασίζεται σε ένα υποσύνολο της γλώσσας προγραμματισμού JavaScript και χρησιμοποιεί μια παρόμοια σύνταξη με τα αντικείμενα JavaScript. Περιέχει ζεύγη κλειδιού-τιμής, όπου τα κλειδιά είναι συμβολοσειρές και οι τιμές μπορεί να είναι οποιεσδήποτε από ένα σύνολο τύπων δεδομένων, όπως συμβολοσειρές, αριθμοί, Booleans, πίνακες και objects [14].

Το JSON υποστηρίζεται από τις περισσότερες γλώσσες προγραμματισμού και πλατφόρμες ανάπτυξης. Ένα από τα πλεονεκτήματα της χρήσης του JSON είναι ότι πρόκειται για μια απλή και ευέλικτη μορφή που μπορεί εύκολα να αναλυθεί και να παραχθεί από τις περισσότερες γλώσσες προγραμματισμού. Είναι επίσης αυτοπεριγραφικό, που σημαίνει ότι η δομή των δεδομένων περιλαμβάνεται στο ωφέλιμο φορτίο, γεγονός που οδηγεί στην εύκολη την ερμηνεία και την κατανόησή τους [15].

Κεφάλαιο 3

Databases

Μια βάση δεδομένων είναι μια συλλογή δομημένων στοιχείων, που είναι ταξινομημένα και αποθηκεύονται με έναν τρόπο που επιτρέπει την επεξεργασία αυτών των στοιχείων. Οι βάσεις δεδομένων χρησιμοποιούνται για την αποθήκευση και διαχείριση στοιχείων για μια ευρεία ποικιλία εφαρμογών, όπως εφαρμογές ιστού, χρηματοοικονομικά συστήματα και ανάλυση επιστημονικών δεδομένων.

Υπάρχουν πολλοί διαφορετικοί τύποι βάσεων δεδομένων, η καθεμία με τα δικά της χαρακτηριστικά και δυνατότητες. Μερικοί από τους πιο συνηθισμένους τύπους βάσεων δεδομένων περιλαμβάνουν [16]:

- **Σχεσιακές βάσεις δεδομένων:** Οι σχεσιακές βάσεις δεδομένων είναι ο πιο ευρέως χρησιμοποιούμενος τύπος βάσης δεδομένων και χαρακτηρίζονται από τη χρήση δομημένης γλώσσας ερωτημάτων (SQL) για τον χειρισμό και την αναζήτηση δεδομένων. Οι σχεσιακές βάσεις δεδομένων αποθηκεύουν δεδομένα σε πίνακες, με κάθε πίνακα να αποτελείται από γραμμές και στήλες. Οι γραμμές αντιπροσωπεύουν μεμονωμένες εγγραφές και οι στήλες αντιπροσωπεύουν τα διάφορα πεδία ή χαρακτηριστικά αυτών των εγγραφών. Παραδείγματα σχεσιακών βάσεων δεδομένων είναι η MySQL, η Oracle και η MariaDB.
- **Βάσεις δεδομένων NoSQL:** Οι βάσεις δεδομένων NoSQL (συντομογραφία του "Not Only SQL") έχουν σχεδιαστεί για να χειρίζονται μεγάλους όγκους μη δομημένων στοιχείων και υποστηρίζουν ένα ευρύ φάσμα μοντέλων, συμπεριλαμβανομένων ζευγών κλειδιών-τιμών, δεδομένων προσανατολισμένων σε έγγραφα καθώς και γραφημάτων. Οι βάσεις δεδομένων NoSQL χρησιμοποιούνται συχνά για εφαρμογές που απαιτούν

γρήγορη πρόσβαση ανάγνωσης και εγγραφής σε μεγάλες ποσότητες δεδομένων όπως οι αναλύσεις σε πραγματικό χρόνο, τα μέσα κοινωνικής δικτύωσης και διάφορα τυχερά παιχνίδια. Παραδείγματα βάσεων δεδομένων NoSQL περιλαμβάνουν τις MongoDB, Cassandra και Redis.

- Βάσεις δεδομένων στη μνήμη: Οι βάσεις δεδομένων στη μνήμη έχουν σχεδιαστεί για να αποθηκεύουν και να επεξεργάζονται τα δεδομένα εξ' ολοκλήρου στη μνήμη, γεγονός που επιτρέπει εξαιρετικά γρήγορους χρόνους πρόσβασης και επεξεργασίας. Επίσης χρησιμοποιούνται συχνά για εφαρμογές πραγματικού χρόνου και αναλύσεις δεδομένων (π.χ. MemSQL και SAP HANA).

3.0.1 Χαρακτηριστικά σχεσιακών βάσεων δεδομένων, βάσεις NoSQL, και βάσεων δεδομένων στη μνήμη

Συνοπτικά οι βάσεις δεδομένων έχουν τα εξής χαρακτηριστικά:

Σχεσιακές βάσεις δεδομένων [17]:

- Αποθηκεύουν τα δεδομένα σε πίνακες με γραμμές και στήλες.
- Χρησιμοποιούν τη δομημένη γλώσσα (SQL) για την επεξεργασία δεδομένων.
- Παρέχουν σταθερή συμφωνία και συμμόρφωση ACID.
- Ιδανικά για εφαρμογές με πολύπλοκες σχέσεις μεταξύ δεδομένων, όπως χρηματοοικονομικά συστήματα.

Βάσεις δεδομένων NoSQL [18]:

- Υποστηρίζουν ένα ευρύ φάσμα μοντέλων δεδομένων.
- Προσφέρουν γρήγορη πρόσβαση.
- Κατάλληλες για εφαρμογές όπως τα Μ.Κ.Δ και τυχερά παιχνίδια.

Βάσεις δεδομένων στη μνήμη [19]:

- Αποθηκεύουν και επεξεργάζονται τα δεδομένα εξ' ολοκλήρου στη μνήμη, επιτρέποντας εξαιρετικά γρήγορους χρόνους πρόσβασης και επεξεργασίας.
- Ενδείκνυνται για εφαρμογές πραγματικού χρόνου και αναλύσεις δεδομένων.

3.0.2 MySQL σε σχέση με άλλες databases

Η MySQL είναι ένα δημοφιλές λογισμικό διαχείρισης βάσεων δεδομένων ανοιχτού κώδικα που χρησιμοποιείται σε ευρύ φάσμα σε εφαρμογές ιστού και άλλων λογισμικών. Προσφέρει διάφορα πλεονεκτήματα σε σχέση με άλλες βάσεις δεδομένων όπως [20, 21]:

- Ευκολία χρήσης: Η MySQL είναι σχετικά εύκολη στη χρήση και έχει ένα απλό, ευχερές interface. Αυτό την καθιστά μια καλή επιλογή για τους προγραμματιστές που είναι νέοι στη διαχείριση βάσεων δεδομένων.
- Μεγάλη διαδεδομένη χρήση: Η MySQL χρησιμοποιείται και υποστηρίζεται σε ευρύ φάσμα χρηστών και προγραμματιστών. Υπάρχει ένα πλούσιο υλικό εγχειριδίων, πόρων υποστήριξης και εργαλείων τρίτων που είναι διαθέσιμα για τη MySQL.
- Υψηλές επιδόσεις: Η MySQL είναι γνωστή για τις υψηλές επιδόσεις και την αξιοπιστία της. Είναι μια πολύ καλή επιλογή για εφαρμογές με μεγάλο αριθμό ταυτόχρονων χρηστών ή υψηλά επίπεδα κίνησης.
- Επεκτασιμότητα: Η MySQL έχει σχεδιαστεί για να κλιμακώνεται καλά και μπορεί να διαχειριστεί μεγάλες ποσότητες δεδομένων και υψηλά επίπεδα ταυτόχρονης χρήσης.
- Κόστος: Η MySQL είναι δωρεάν και ανοικτού κώδικα, κάτι που την κάνει οικονομικά αποδοτική επιλογή για πολλές εφαρμογές.

Κατά την επιλογή μιας βάσης δεδομένων για μια συγκεκριμένη εφαρμογή, είναι σημαντικό να λαμβάνονται υπόψη οι συγκεκριμένες απαιτήσεις και ανάγκες της συγκεκριμένης εφαρμογής. Οι παράγοντες που πρέπει να ληφθούν υπόψη περιλαμβάνουν τον τύπο και τη δομή των δεδομένων που αποθηκεύονται, τον αναμενόμενο φόρτο εργασίας, τις απαιτήσεις επιδόσεων, καθώς και το επίπεδο κλιμάκωσης και αξιοπιστίας που απαιτείται [21].

Για παράδειγμα, για την δημιουργία μιας εφαρμογής ιστού που απαιτεί γρήγορη πρόσβαση ανάγνωσης και εγγραφής σε μεγάλες ποσότητες δεδομένων, όπως μια πλατφόρμα μέσω κοινωνικής δικτύωσης ή μια εφαρμογή τυχερών παιχνιδιών, μια βάση δεδομένων NoSQL όπως η MongoDB ή η Cassandra είναι ιδανικές επιλογές. Από την άλλη πλευρά, για την δημιουργία μιας οικονομικής εφαρμογής που απαιτεί αυστηρή συνέπεια και συμμόρφωση με το ACID, μια σχεσιακή βάση δεδομένων όπως η MySQL ή η Oracle είναι προτιμότερες.

Κεφάλαιο 4

Frontend

Το front-end είναι ένας όρος που χρησιμοποιείται για να περιγράψει το τμήμα μιας εφαρμογής λογισμικού που είναι ορατό στον χρήστη και αλληλεπιδρά με αυτόν. Στην ανάπτυξη ιστοσελίδων, το front-end είναι το τμήμα της εφαρμογής που εκτελείται στο πρόγραμμα περιήγησης ιστού του χρήστη και είναι υπεύθυνο για την απόδοση και την εμφάνιση της διεπαφής χρήστη (user interface), καθώς και για τον χειρισμό των εισόδων και εξόδων του χρήστη [22].

Η ανάπτυξη front-end μπορεί να περιλαμβάνει τη χρήση HTML, CSS και JavaScript για τη δημιουργία της διεπαφής χρήστη και για να καταστεί η εφαρμογή διαδραστική και ανταποκρινόμενη. Οι προγραμματιστές front-end εργάζονται για να διασφαλίσουν ότι η εφαρμογή είναι οπτικά ελκυστική, εύχρηστη και αποδοτικά βελτιστοποιημένη.

Οι τεχνολογίες front-end χρησιμοποιούνται σε ένα ευρύ φάσμα εφαρμογών, συμπεριλαμβανομένων εφαρμογών ιστού και κινητών τηλεφώνων, πλατφορμών ηλεκτρονικού εμπορίου και προϊόντων λογισμικού ως υπηρεσία (SaaS). Η ανάπτυξη front-end αποτελεί σημαντικό μέρος της διαδικασίας ανάπτυξης λογισμικού, καθώς διαδραματίζει καθοριστικό ρόλο στην εμπειρία του χρήστη και στη συνολική επιτυχία της εφαρμογής [22].

Η ανάπτυξη front-end περιλαμβάνει την υλοποίηση διαδραστικών και ευέλικτων χαρακτηριστικών. Οι προγραμματιστές χρησιμοποιούν διάφορες γλώσσες για να δημιουργήσουν και να διαμορφώσουν τη διεπαφή χρήστη και για να καταστήσουν την εφαρμογή ανταποκρινόμενη [23].

Στην ανάπτυξη ιστοσελίδων, οι προγραμματιστές εκτελούν εργασίες όπως η διάταξη, ο σχεδιασμός με δυνατότητα απόκρισης και η υλοποίηση διαδραστικών στοιχείων όπως φόρμες, κουμπιά και μενού.

Για τον έλεγχο απόδοσης και ταχύτητας της εφαρμογής, οι προγραμματιστές εκτελούν

τεχνικές όπως η ελαχιστοποίηση του μεγέθους των αρχείων και η χρήση εργαλείων βελτιστοποίησης για την απόδοση της εφαρμογής [23].

Γενικότερα το frontend αποτελεί έναν ταχέως εξελισσόμενο τομέα, με νέες τεχνολογίες και τεχνικές να εισάγονται και να αναπτύσσονται σε τακτική βάση.

4.1 Σύζευξη HTML, CSS και JS

Η HTML, η CSS και η JavaScript είναι οι τρεις κύριες τεχνολογίες που χρησιμοποιούνται στην ανάπτυξη ιστοσελίδων front-end. Μια σύντομη επισκόπηση κάθε τεχνολογίας έχει ως εξής [24]:

- Η HTML (HyperText Markup Language) είναι μια γλώσσα σήμανσης που χρησιμοποιείται για τη διάρθρωση και τη μορφοποίηση του περιεχομένου στον ιστό. Η HTML αποτελείται από ένα σύνολο στοιχείων και χαρακτηριστικών που καθορίζουν τη δομή και το περιεχόμενο μιας ιστοσελίδας, όπως επικεφαλίδες, παραγράφους, λίστες και συνδέσμους.
- CSS (Cascading Style Sheets) είναι μια γλώσσα φύλλων στυλ (style sheets) που χρησιμοποιείται για τον έλεγχο της εμφάνισης και της διάταξης μιας ιστοσελίδας. Η CSS επιτρέπει στους προγραμματιστές να καθορίζουν τον τρόπο εμφάνισης των στοιχείων HTML, συμπεριλαμβανομένων του χρώματος, της γραμματοσειράς και της διάταξης.
- Η JavaScript είναι μια γλώσσα προγραμματισμού που χρησιμοποιείται συνήθως στην front-end ανάπτυξη ιστοσελίδων για τη δημιουργία διαδραστικών και δυναμικών εφαρμογών ιστού. Η JavaScript μπορεί να χρησιμοποιηθεί για την προσθήκη λειτουργιών σε ιστοσελίδες, όπως επικύρωση φορμών, ρυθμιστικά εικόνων και κινουμένων σχεδίων.

Η HTML, η CSS και η JavaScript συνεργάζονται για να δημιουργήσουν την εικόνα μιας διαδικτυακής εφαρμογής. Οι προγραμματιστές μπορούν να χρησιμοποιούν αυτές τις τεχνολογίες για τη δημιουργία της διάταξης και του σχεδιασμού της εφαρμογής, καθώς και για την προσθήκη ευέλικτων λειτουργιών.

4.1.1 HTML

Η HTML είναι μια βασική τεχνολογία στην ανάπτυξη ιστοσελίδων front-end, καθώς αποτελεί τη βάση για τη δομή και το περιεχόμενο μιας ιστοσελίδας. [25].

Η HTML έχει υποστεί σημαντικές αλλαγές από την ίδρυσή της στις αρχές της δεκαετίας του 1990, με την εισαγωγή νέων εκδόσεων όπως η HTML 4.01 και η HTML5. Η τελευταία έκδοση της HTML, η HTML5, εισήγαγε νέα στοιχεία και χαρακτηριστικά που επιτρέπουν στους προγραμματιστές να δημιουργούν πιο διαδραστικές και δυναμικές εφαρμογές ιστού, όπως η δυνατότητα αναπαραγωγής ήχου και βίντεο εγγενώς στο πρόγραμμα περιήγησης και η δημιουργία γραφικών με τη χρήση του στοιχείου canvas [25].

4.1.2 HTML vs HTML5

Μερικές από τις κύριες διαφορές μεταξύ της HTML και της HTML5 περιλαμβάνουν [26]:

- Υποστήριξη πολυμέσων: Η HTML5 εισήγαγε νέα στοιχεία και API, εξαλείφοντας την ανάγκη για πρόσθετα όπως το Flash.
- Τοπική αποθήκευση: Η HTML5 εισήγαγε την αποθήκευση στο διαδίκτυο, η οποία επιτρέπει στους προγραμματιστές να αποθηκεύουν δεδομένα τοπικά στο πρόγραμμα περιήγησης και να έχουν πρόσβαση σε αυτά εκτός σύνδεσης.
- Σημασιολογικά στοιχεία: Η HTML5 εισήγαγε νέα σημασιολογικά στοιχεία, όπως κεφαλίδα, υποσέλιδο και παν, τα οποία παρέχουν πρόσθετη δομή και νόημα στο περιεχόμενο του ιστού.
- Συμβατότητα: Η HTML5 έχει σχεδιαστεί ώστε να είναι συμβατή προς τα πίσω με παλαιότερες εκδόσεις της HTML, πράγμα που σημαίνει ότι οι ιστοσελίδες και οι εφαρμογές που έχουν γραφτεί σε HTML θα λειτουργούν σε παλαιότερα προγράμματα περιήγησης που υποστηρίζουν την HTML.

4.1.3 CSS

Η CSS είναι μια γλώσσα φύλλων που χρησιμοποιείται για τον έλεγχο της εμφάνισης περιεχομένου. Η CSS χρησιμοποιείται για τον ορισμό στυλ για στοιχεία HTML και για την

εφαρμογή αυτών των στυλ σε ιστοσελίδες και εφαρμογές. Η CSS επιτρέπει στους προγραμματιστές να διαχωρίσουν το περιεχόμενο μιας ιστοσελίδας ή εφαρμογής από την παρουσίασή του, διευκολύνοντας έτσι τη διατήρηση και την ενημέρωση της εμφάνισης και της ατμόσφαιρας μιας ιστοσελίδας ή εφαρμογής με την πάροδο του χρόνου [27].

Η CSS διαθέτει μια σειρά χαρακτηριστικών και δυνατοτήτων που την καθιστούν ένα ισχυρό εργαλείο για την ανάπτυξη ιστοσελίδων, όπως [27]:

- Διαχωρισμός περιεχομένου και παρουσίασης: Η CSS επιτρέπει στους προγραμματιστές να διαχωρίζουν το περιεχόμενο μιας ιστοσελίδας ή εφαρμογής από την παρουσίασή της, κάνοντας ευκολότερη τη διατήρηση και την ενημέρωση της εμφάνισης και της ατμόσφαιρας ενός ιστότοπου ή μιας εφαρμογής με την πάροδο του χρόνου.
- Επαναχρησιμοποίηση: Τα στυλ CSS μπορούν να επαναχρησιμοποιηθούν σε πολλές σελίδες ή εφαρμογές, διευκολύνοντας τη διατήρηση μιας συνεπούς εμφάνισης και αίσθησης σε έναν ιστότοπο ή μια εφαρμογή.
- Αρθρωτότητα: Τα στυλ CSS μπορούν να οργανωθούν και να σπονδυλωθούν χρησιμοποιώντας τεχνικές όπως το cascade, η κληρονομικότητα και το μοντέλο box, γεγονός που επιτρέπει στους προγραμματιστές να δημιουργούν ευέλικτα και κλιμακούμενα φύλλα στυλ.

4.1.4 JS

Ένα από τα κύρια πλεονεκτήματα της χρήσης της JavaScript ως γλώσσας front-end είναι η ικανότητά της να χειρίζεται το Μοντέλο Αντικειμένου Εγγράφου (DOM) μιας ιστοσελίδας. Το DOM είναι μια δενδροειδής δομή που αναπαριστά το περιεχόμενο και τη διάταξη μιας ιστοσελίδας και η JavaScript μπορεί να χρησιμοποιηθεί για την τροποποίηση του DOM σε πραγματικό χρόνο, επιτρέποντας στους προγραμματιστές να δημιουργούν διαδραστικές και δυναμικές ιστοσελίδες που μπορούν να ανταποκρίνονται στις εισροές του χρήστη και στις αλλαγές των δεδομένων [28].

Επίσης, ακόμα ένα από τα βασικά χαρακτηριστικά της JavaScript που την καθιστά μια ισχυρή γλώσσα front-end είναι η ικανότητά της να χειρίζεται ασύγχρονα συμβάντα. Αυτό σημαίνει ότι η JavaScript μπορεί να εκτελεί κώδικα βάσει της εισόδου που λαμβάνει από τον χρήστη, από συμβάντα δικτύου ή άλλες ασύγχρονες ενέργειες, χωρίς να μπλοκάρει την εκτέλεση άλλου κώδικα. Αυτό επιτρέπει στους προγραμματιστές να δημιουργούν ευέλικτες

και δυναμικές εφαρμογές ιστού που μπορούν να χειρίζονται πολλαπλά αιτήματα και ενημερώσεις ταυτόχρονα [29].

Η JavaScript είναι επίσης μια εξαιρετικά ευέλικτη γλώσσα, με μια μεγάλη και ενεργή ομάδα προγραμματιστών που δημιουργεί συνεχώς νέες βιβλιοθήκες, πλαίσια και εργαλεία για τη βελτίωση της εμπειρίας ανάπτυξης. Αυτό σημαίνει ότι υπάρχουν πολλοί διαθέσιμοι πόροι για τους προγραμματιστές, από τους οποίους μπορούν να μάθουν και να χρησιμοποιήσουν στα έργα τους, ενώ η γλώσσα εξελίσσεται και βελτιώνεται συνεχώς [29].

Όσον αφορά τις επιδόσεις, η JavaScript έχει διανύσει πολύ δρόμο τα τελευταία χρόνια, με την ανάπτυξη ταχύτερων μηχανών εκτέλεσης και βελτιστοποιήσεων στους σύγχρονους φυλλομετρητές. Αυτό σημαίνει ότι η JavaScript είναι πλέον ικανή να εκτελεί σύνθετες και απαιτητικές εφαρμογές και χρησιμοποιείται συχνά για την κατασκευή εφαρμογών ιστού μεγάλης κλίμακας και υψηλής επισκεψιμότητας [29].

Η JavaScript χρησιμοποιείται επίσης συνήθως για τη δημιουργία κινούμενων εικόνων και μεταβάσεων σε ιστοσελίδες, καθώς και για την υλοποίηση σύνθετων στοιχείων UI, όπως ρυθμιστικά, καρουσέλ και modal παράθυρα. Αυτά τα χαρακτηριστικά μπορούν να βελτιώσουν σημαντικά την εμπειρία χρήσης ενός ιστότοπου και να τον κάνουν πιο ελκυστικό και διαδραστικό [28].

4.1.5 JQuery

Η jQuery είναι μια δημοφιλής βιβλιοθήκη JavaScript που χρησιμοποιείται συχνά για την ανάπτυξη ιστοσελίδων front-end, ιδίως για τη δημιουργία κινούμενων εικόνων, ρυθμιστικών και καρουσέλ [30].

Ένα από τα κύρια πλεονεκτήματα της χρήσης της jQuery για τέτοιου είδους εργασίες είναι ότι παρέχει μια απλή και διαισθητική διεπαφή για το χειρισμό του Μοντέλου Αντικειμένου Εγγράφου (DOM) μιας ιστοσελίδας. Αυτό επιτρέπει στους προγραμματιστές να δημιουργούν και να τροποποιούν εύκολα στοιχεία σε μια σελίδα, όπως η απόκρυψη ή η εμφάνιση στοιχείων, η αλλαγή του στυλ τους, η προσθήκη ή η αφαίρεση κλάσεων.

Το jQuery διαθέτει επίσης μια μεγάλη και ενεργή κοινότητα, που συμβάλλει στη δημιουργία πολλών πόρων για τους χρήστες και για την αξιοποίηση των έργων τους. Αυτό μπορεί να καταστήσει ευκολότερη και ταχύτερη την υλοποίηση σύνθετων και ισχυρών εφέ σε μια ιστοσελίδα, χωρίς να χρειάζεται να ξαναγραφτεί όλος ο κώδικας από την αρχή.

Η jQuery μπορεί επίσης να χρησιμοποιηθεί για τη δημιουργία γραμμών αναζήτησης, λει-

τουργιών αυτόματης συμπλήρωσης και δυναμικών ενημερώσεων μιας ιστοσελίδας με βάση της τρέχουσας διεύθυνσης URL [30].

Για παράδειγμα, μια μπάρα αναζήτησης μπορεί να δημιουργηθεί με τη χρήση της jQuery επισυνάπτοντας έναν χειριστή συμβάντος στο πεδίο εισαγωγής που ενεργοποιεί μια συνάρτηση για την εκτέλεση της αναζήτησης όταν ο χρήστης υποβάλλει τη φόρμα. Η συνάρτηση αναζήτησης θα μπορούσε στη συνέχεια να κάνει μια αίτηση AJAX σε μια δέσμη ενεργειών από την πλευρά του διακομιστή, η οποία θα εκτελούσε την αναζήτηση και θα επέστρεφε τα αποτελέσματα στον πελάτη [30].

Η ικανότητα της jQuery να χειρίζεται το DOM και να ανταποκρίνεται σε αλλαγές στη διεύθυνση URL μπορεί να χρησιμοποιηθεί για τη δημιουργία εφαρμογών μίας σελίδας (SPA) χωρίς να απαιτείται πλήρης ανανέωση της σελίδας. Αυτό μπορεί να προσφέρει μια πιο απρόσκοπτη και ευέλικτη εμπειρία για τον χρήστη [30].

Συνολικά, η jQuery είναι ένα πολύτιμο εργαλείο για το front-end που προσφέρει αρκετά προνόμια και δυναμικά στοιχεία στις εφαρμογές.

4.2 Materieliaze CSS

Το Materialize CSS είναι ένα σύγχρονο front-end πλαίσιο που βασίζεται στις κατευθυντήριες γραμμές Material Design της Google. Παρέχει ένα μεγάλο εύρος απο style-sheets που μπορούν να χρησιμοποιηθούν για τη δημιουργία γραφικών και ευέλικτων εφαρμογών ιστού [31].

Ένα από τα βασικά χαρακτηριστικά του Materialize CSS είναι το σύστημα πλέγματος, το οποίο επιτρέπει στους προγραμματιστές να δημιουργούν σύνθετες και ευέλικτες διατάξεις χρησιμοποιώντας έναν συνδυασμό γραμμών, στηλών και σημείων διακοπής πλέγματος. Το σύστημα πλέγματος προσαρμόζεται αυτόματα στο μέγεθος της οθόνης, καθιστώντας εύκολη τη δημιουργία εφαρμογών που ταιριάζουν απόλυτα τόσο σε κινητές όσο και σε επιτραπέζιες συσκευές.

Εκτός από το σύστημα πλέγματος, το Materialize CSS παρέχει επίσης ένα ευρύ φάσμα προσχεδιασμένων στοιχείων που μπορούν να χρησιμοποιηθούν για τη δημιουργία κοινών στοιχείων διεπαφής χρήστη, όπως κουμπιά, κάρτες, φόρμες, γραμμές πλοήγησης και άλλα. Αυτά τα στοιχεία είναι σχεδιασμένα ώστε να ανταποκρίνονται και να είναι ευέλικτα και μπορούν εύκολα να προσαρμοστούν χρησιμοποιώντας μια σειρά από διαθέσιμες επιλογές.

Για παράδειγμα, τα κουμπιά μπορούν να προσαρμοστούν με διαφορετικά χρώματα, μεγέθη και σχήματα και οι κάρτες μπορούν να διαμορφωθούν με διαφορετικά στυλ και διατάξεις [31].

Το Materialize CSS περιλαμβάνει επίσης μια σειρά από βοηθητικές κλάσεις που μπορούν να χρησιμοποιηθούν για την εφαρμογή style-sheets, όπως ευθυγραμμίσεις, χρώματα και τυπογραφία, καθώς και μια σειρά από πρόσθετα JavaScript που μπορούν να χρησιμοποιηθούν για την προσθήκη διαδραστικών εφέ, όπως modals, sliders και καρτέλες. Αυτά τα πρόσθετα μπορούν εύκολα να διαμορφωθούν και να προσαρμοστούν χρησιμοποιώντας μια σειρά επιλογών και μεθόδων, επιτρέποντας στους προγραμματιστές να δημιουργήσουν ένα ευρύ φάσμα δυναμικών και διαδραστικών στοιχείων [31].

Ένα από τα πλεονεκτήματα της χρήσης του Materialize CSS είναι το σύνολο αρχών και γραμμών σχεδιασμού για τη δημιουργία οπτικά ελκυστικών και διαισθητικών διεπαφών χρήστη. Αυτές οι γραμμές βασίζονται σε έρευνες σχετικά με τον τρόπο με τον οποίο οι χρήστες αντιλαμβάνονται και αλληλεπιδρούν με τις ψηφιακές διεπαφές και έχουν σχεδιαστεί για να βοηθήσουν τους προγραμματιστές να δημιουργήσουν εφαρμογές που είναι εύχρηστες και οπτικά ελκυστικές. Τηρώντας αυτές τις κατευθυντήριες γραμμές, οι προγραμματιστές μπορούν να δημιουργήσουν εφαρμογές που είναι πιο πιθανό να είναι επιτυχείς και να τύχουν καλής υποδοχής από τους χρήστες.

Το Materialize CSS έχει επίσης σχεδιαστεί για να είναι εύκολο στη προσαρμογή, με μια ολοκληρωμένη τεκμηρίωση και ένα ευρύ φάσμα παραδειγμάτων και προτύπων που μπορούν να βοηθήσουν τους προγραμματιστές να ξεκινήσουν γρήγορα. Επίσης, συντηρείται και υποστηρίζεται ενεργά από μια μεγάλη και αυξανόμενη κοινότητα, που οδηγεί σε πληθώρα πόρων υποστήριξης για τους χρήστες.

Συνολικά, το Materialize CSS είναι ένα ισχυρό και εύχρηστο πλαίσιο front-end που μπορεί να βοηθήσει τους προγραμματιστές να δημιουργήσουν γρήγορα και αποτελεσματικά οπτικά ελκυστικές εφαρμογές ιστού. Είναι ένα πολύτιμο εργαλείο για κάθε προγραμματιστή που επιθυμεί να δημιουργήσει σύγχρονες ιστοσελίδες [31].

Κεφάλαιο 5

Ανάπτυξη του BookshelfCy

Ο ιστότοπος υλοποιεί μια υπηρεσία ηλεκτρονικού εμπορίου για online πωλήσεις βιβλίων και παρέχει μια πλατφόρμα όπου οι πελάτες μπορούν να περιηγηθούν, να αγοράσουν και να τους παραδοθούν τα βιβλία ηλεκτρονικά ή με παραδοσιακές μεθόδους αποστολής.

Σαν code editor της σελίδας χρησιμοποιήθηκε Visual Studio Code. Για την αποθήκευση δεδομένων στην σελίδα χρησιμοποιήθηκε η MySQL. Ως back-end χρησιμοποιήθηκε η ευρέως γνωστή node.js , ενώ για frontend η Html.js,CSS και το framework της Materialize css.

Ορισμένα βασικά χαρακτηριστικά που υλοποιεί η ιστοσελίδα για ηλεκτρονικές πωλήσεις βιβλίων περιλαμβάνουν:

- Ένας κατάλογος ή μια βάση δεδομένων των βιβλίων που είναι διαθέσιμα προς αγορά, συμπεριλαμβανομένων περιγραφών, περίληψης, εικόνων και πληροφοριών τιμολόγησης. Μια λειτουργία αναζήτησης που επιτρέπει στους πελάτες να βρίσκουν συγκεκριμένα βιβλία με βάση τα ενδιαφέροντα τους.
- Εύκολη περιήγηση στην σελίδα όπου το προϊόν είναι διαθέσιμο για αγορά, όπου θα βρει το καλάθι αγορών της αντίστοιχης σελίδας. Επίσης θα έχει την ευελιξία να προσθέσει και να εξετάσει βιβλία, καθώς και μια ασφαλή πύλη πληρωμών για την ολοκλήρωση της αγοράς του.
- Άμεση εύρεση διαθεσιμότητας η όχι του ανάλογου βιβλίου, σε όλα τα μαγαζιά της Κύπρου.
- Συστήματα όπως η παρακολούθηση των παραγγελιών και την παροχή ενημερώσεων

στους πελάτες σχετικά με την κατάσταση των αγορών τους, θα βρίσκονται εύκολα μέσω της αντίστοιχης σελίδας αγοράς (π.χ Parga.com.cy).

- Βελτιστοποιημένη σελίδα για κινητά και tablets.

Συνολικά, ο ιστότοπος παρέχει μια βολική και φιλική προς τον χρήστη πλατφόρμα για την αγορά βιβλίων από τους πελάτες και βοηθά τις επιχειρήσεις να διαχειρίζονται αποτελεσματικά τη διαδικασία πώλησης βιβλίων στο διαδίκτυο.

5.1 Συνεργασία συνεργατών

Οι συνεργασίες συνεργατών (affiliate partnerships) είναι πολύ σημαντικές για τα έσοδα μιας τέτοιας ιστοσελίδας, επειδή κυρίως προσφέρουν μία ευκαιρία για να κερδίζει έσοδα μέσω των προμηθειών που λαμβάνει από τις εμπορικές ιστοσελίδες που συμμετέχουν στην προβολή κάθε βιβλίου. Αυτό είναι ιδιαίτερα σημαντικό για τις ιστοσελίδες που δεν παρέχουν καμία άλλη υπηρεσία ή προϊόντα για πώληση, καθώς δεν έχουν άλλο τρόπο για να κερδίζουν έσοδα. Το μηνιαίο εισόδημα μπορεί να προκύψει από τον αριθμό των κλικ ή των ανακατευθύνσεων στις ιστοσελίδες τους.

5.2 Συλλογή δεδομένων

Η συλλογή δεδομένων είναι σημαντική για τις μηχανές αναζήτησης γιατί επιτρέπει πολλαπλή αναζήτηση στους χρήστες. Όταν οι χρήστες εισάγουν μια λέξη-κλειδί στη μηχανή αναζήτησης, η μηχανή αναζήτησης ψάχνει στο database για ιστοσελίδες που περιέχουν τη λέξη-κλειδί και παρέχει τα αποτελέσματα στη σειρά που πιστεύει ότι είναι πιο ικανοποιητική για τον χρήστη. Για αποτελεσματικότητα η μηχανή αναζήτησης χρειάζεται έγκυρα και επαρκή δεδομένα που να συλλέγονται από τις ιστοσελίδες που είναι διαθέσιμες στο Διαδίκτυο [32].

5.2.1 Web scraping methods and modules using node.js

Web-scraping tools όπως crawlers βοηθάνε σε μεγάλο βαθμό να επιτευχθούν τα πιο πάνω. Τα crawlers είναι προγράμματα που σκοπό έχουν να συλλέγουν δεδομένα από το Διαδίκτυο. Οι ανιχνευτές ιστού (crawlers) λειτουργούν ξεκινώντας με μια αρχική διεύθυνση URL ή ένα

σύνολο αρχικών διευθύνσεων URL και στη συνέχεια ακολουθώντας τους συνδέσμους σε αυτές τις σελίδες ανακαλύπτουν νέες σελίδες. Καθώς επισκέπτονται κάθε σελίδα, εξάγουν τις διευθύνσεις URL των συνδέσμων σε αυτή τη σελίδα και τις προσθέτουν σε μια ουρά διευθύνσεων URL που πρόκειται να επισκεφθούν. Αυτή η διαδικασία επαναλαμβάνεται μέχρι ο ανιχνευτής να επισκεφθεί όλες τις σελίδες που θέλει να ανιχνεύσει ή μέχρι να φτάσει σε ένα προκαθορισμένο όριο [32].

Οι ανιχνευτές ιστού μπορούν να ταξινομηθούν σε δύο τύπους, ανιχνευτές γενικού σκοπού και εξειδικευμένοι ανιχνευτές. Ένας ανιχνευτής γενικού σκοπού έχει σχεδιαστεί για να ανακαλύπτει ένα ευρύ φάσμα σελίδων στον ιστό, ενώ ένας εξειδικευμένος ανιχνευτής έχει σχεδιαστεί για να συλλέγει δεδομένα για μια συγκεκριμένη εφαρμογή, όπως σύγκριση τιμών, ανάλυση συναισθήματος ή συγκέντρωση ειδήσεων.

Οι ανιχνευτές γενικού σκοπού μπορούν επίσης να ταξινομηθούν σε δύο κατηγορίες, ευγενικοί ανιχνευτές και αγενείς ανιχνευτές. Οι ευγενικοί ανιχνευτές έχουν σχεδιαστεί για να σέβονται τους κανόνες και τους κανονισμούς που θέτουν οι διαχειριστές ιστοσελίδων, όπως το αρχείο robots.txt και η οδηγία crawl-delay. Οι αγενείς ανιχνευτές, από την άλλη πλευρά, αγνοούν αυτούς τους κανόνες και μπορεί να προκαλέσουν ζημιά σε έναν ιστότοπο υπερφορτώνοντάς τον με αιτήσεις ή αποσπώντας ευαίσθητα δεδομένα [32].

Υπάρχουν διάφορες βιβλιοθήκες και πλαίσια που μπορούν να χρησιμοποιηθούν για την υλοποίηση ανιχνευτών ιστού, όπως το Scrapy, το BeautifulSoup και το Cheerio.

Το Cheerio είναι μια ελαφριά και γρήγορη βιβλιοθήκη για την ανάλυση ιστού σε JavaScript. Επιτρέπει στους προγραμματιστές να χρησιμοποιούν μια σύνταξη παρόμοια με την jQuery για την επιλογή και τον χειρισμό στοιχείων σε μια ιστοσελίδα. Παρέχει επίσης πολλές μεθόδους για το φιλτράρισμα των στοιχείων HTML, καθιστώντας εύκολη την εξαγωγή των δεδομένων. Το Cheerio είναι χτισμένο πάνω στον βασικό αναλυτή της βιβλιοθήκης JSDom, η οποία είναι μια υλοποίηση του DOM (Document Object Model) [33].

Το Puppeteer είναι μια βιβλιοθήκη Node.js που αναπτύχθηκε από την Google και επιτρέπει στους προγραμματιστές να ελέγχουν ένα πρόγραμμα περιήγησης Chrome με ή χωρίς κεφάλι. Παρέχει ένα υψηλού επιπέδου API για την αλληλεπίδραση με ιστοσελίδες, όπως το κλικ σε συνδέσμους, τη συμπλήρωση φορμών και τη λήψη στιγμιότυπων οθόνης. Το Puppeteer μπορεί να χρησιμοποιηθεί για την αυτοματοποίηση εργασιών του προγράμματος περιήγησης, την εκτέλεση δοκιμών του προγράμματος περιήγησης και την επεξεργασία δυναμικών ιστοσελίδων που βασίζονται σε JavaScript. Επιτρέπει την αλληλεπίδραση με ιστοσε-

λίδες ακριβώς όπως θα έκανε ένας χρήστης και μπορεί να χρησιμοποιηθεί για την απόσπαση δεδομένων από ιστοσελίδες που δεν είναι εύκολα προσβάσιμες μέσω άλλων μεθόδων [34].

Με την χρήση του Cheerio, η διαδικασία της επεξεργασίας δεδομένων ιστού περιλαμβάνει συνήθως την υποβολή ενός αιτήματος HTTP σε έναν ιστότοπο και, στη συνέχεια, την ανάλυση της σελίδας σε HTML ή XML. Μόλις γίνει η ανάλυση των δεδομένων, χρησιμοποιείται το API του Cheerio για την διαχείριση του έγγραφου και την εξαγωγή των πληροφοριών. Για παράδειγμα, με την χρήση της μέθοδου `(selector).text()` για να επιτευχθεί η εξαγωγή του περιεχομένου ενός συγκεκριμένου κειμένου, ή τη μέθοδο `(selector).attr(attribute)` για την εξαγωγή ενός χαρακτηριστικού [35].

Κατά τη χρήση του Puppeteer, η διαδικασία της επεξεργασίας δεδομένων ιστού συνήθως περιλαμβάνει την εκκίνηση ενός προγράμματος περιήγησης χωρίς κεφαλή, τη δημιουργία μιας νέας σελίδας, την πλοήγηση σε έναν ιστότοπο και στη συνέχεια την εξαγωγή των πληροφοριών αναζήτησης χρησιμοποιώντας το API του Puppeteer. Για παράδειγμα, μπορεί να χρησιμοποιηθεί η μέθοδος `page.goto()` για πλοήγηση σε έναν ιστότοπο, η μέθοδος `page.evaluate()` για εκτέλεση JavaScript στη σελίδα και η μέθοδος `page.eval()` για να εξαχθεί το περιεχόμενο κειμένου ή ένα χαρακτηριστικό ενός συγκεκριμένου στοιχείου [36].

Τόσο το Cheerio όσο και το Puppeteer είναι εργαλεία τα οποία χρησιμοποιούνται ευρέως από προγραμματιστές για την εξαγωγή και συλλογή δεδομένων από ιστότοπους, έχοντας τα δικά τους πλεονεκτήματα και μειονεκτήματα.

```

const puppeteer = require('puppeteer');

const RENDER_CACHE = new Map();

async function ssr(url, selector, scroll, pages, loadMoreExists) {
  if (RENDER_CACHE.has(url)) {
    return RENDER_CACHE.get(url);
  }

  let data = '';

  const browser = await puppeteer.launch({
    args: puppeteer.args,
    defaultViewport: puppeteer.defaultViewport,
    headless: false,
    ignoreHTTPSErrors: true,
  });

  const page = await browser.newPage();
  let currentPage = 1, pagestoscrape = pages;

  try {
    // networkidle0 waits for the network to be idle (no requests for 500ms).
    // The page's JS has likely produced markup by this point, but wait longer
    // if site lazy loads, etc.
    await page.goto(url, {waitUntil: 'networkidle0'});

    // if scroll is needed to load lazy data
    if (scroll) {
      await page.evaluate(() => {
        window.scrollTo(0, document.body.scrollHeight);
      });
    }

    // iterate pages to get data
    while (currentPage <= pagestoscrape) {
      if (currentPage < pagestoscrape) {
        // Wait for the load more button to be visible and clickable
        await page.waitForSelector(selector, {visible: true});
        page.waitForNavigation({waitUntil: 'networkidle0'})
        await page.click(selector);
      }
    }
  }

```

```

// Append
data = data + await page.content();
currentPage++;
} catch (err) {
  console.log(err);
}

await console.log("Rendered url: "+url);
return {browser, page };
}

async function autoScroll(page) {
  await page.evaluate(async () => {
    await new Promise((resolve, reject) => {
      var totalHeight = 0;
      var distance = 100;
      var timer = setInterval(() => {
        var scrollHeight = document.body.scrollHeight;
        window.scrollBy(0, distance);
        totalHeight += distance;

        if (totalHeight >= scrollHeight) {
          clearInterval(timer);
          resolve();
        }
      }, 100);
    });
  });

  // Stop scrolling after 30 seconds
  setTimeout(() => {
    clearInterval(timer);
  }, 30 * 1000);
}

module.exports = { ssr, autoScroll };

```

Σχήμα 5.1: Πίνακας database για συλλογή ονόματος,εικόνας,περίληψης και isbn.

Όπως φαίνεται παραπάνω, ο κώδικας εξάγει μια συνάρτηση με το όνομα "ssr" που χρη-

σιμοποιείται για την απόδοση από την πλευρά του διακομιστή (SSR) ενός ιστότοπου χρησιμοποιώντας τη βιβλιοθήκη puppeteer. Η συνάρτηση λαμβάνει 4 παραμέτρους:

- **Url:** Η διεύθυνση URL της ιστοσελίδας που πρόκειται να αποδοθεί.
- **Selector:** Ο επιλογέας CSS για το κουμπί "load more" στον ιστότοπο, που χρησιμοποιείται για τη φόρτωση περισσότερων δεδομένων καθώς ο χρήστης μετακινείται προς τα κάτω.
- **Scroll:** Μια boolean τιμή που υποδεικνύει αν θα γίνει κύλιση προς τα κάτω για να φορτωθούν περισσότερα δεδομένα.
- **Pages:** Ο αριθμός των σελίδων που θα ανιχνευθούν. Η συνάρτηση χρησιμοποιεί επίσης ένα αντικείμενο Map με όνομα "RENDERCACHE" για την αποθήκευση της rendered HTML ενός ιστότοπου, έτσι ώστε εάν ζητηθεί να αποδοθεί ξανά ο ίδιος ιστότοπος, να επιστραφεί η cached έκδοση αντί της εκ νέου απόδοσης του ιστότοπου.

Η συνάρτηση χρησιμοποιεί το puppeteer για την εκκίνηση ενός headless browser και τη δημιουργία μιας νέας σελίδας. Στη συνέχεια, πλοηγείται στην καθορισμένη διεύθυνση URL και περιμένει το δίκτυο να μείνει σε αδράνεια (κανένα αίτημα για 500ms) προτού αποδώσει τον ιστότοπο.

Εάν η παράμετρος 'scroll' έχει οριστεί σε true, η συνάρτηση χρησιμοποιεί τη μέθοδο evaluate() για να προσομοιώσει την κύλιση της σελίδας προς τα κάτω. Στη συνέχεια, χρησιμοποιεί έναν βρόχο while για την επανάληψη του αριθμού των σελίδων που καθορίζονται στην παράμετρο 'pages'. Εάν παρέχεται η παράμετρος 'selector', η συνάρτηση χρησιμοποιεί τη μέθοδο waitForSelector() για να περιμένει να είναι ορατό το κουμπί "load more" και στη συνέχεια κάνει κλικ σε αυτό για να φορτώσει περισσότερα δεδομένα. Στη συνέχεια επισυνάπτει την HTML της τρέχουσας σελίδας στη μεταβλητή 'data'.

Η συνάρτηση επιστρέφει ένα αντικείμενο που περιέχει το πρόγραμμα περιήγησης και τη σελίδα και εξάγει επίσης μια συνάρτηση με όνομα autoScroll που μετακινεί τη σελίδα συνεχώς προς τα κάτω για 30 δευτερόλεπτα.

Είναι σημαντικό να σημειωθεί ότι αυτός ο κώδικας είναι μόνο ένα μέσο ως προς την διαδικασία επεξεργασίας δεδομένων, δεν συλλέγει πληροφορίες από μόνος του, και ο στόχος του είναι να αποδώσει την ιστοσελίδα και να επιστρέψει την HTML ώστε να μπορεί να αναλυθεί με το cheerio ή άλλες μεθόδους.

```

ssr(mainUrl, selector, scroll, pages, loadMoreExists)
  .then(async ({ browser, page }) => {
    // Get all the links for each product
    const links = await page.evaluate(() => {
      const elements = Array.from(document.querySelectorAll('.books-details-lft a'));
      return elements.map(e => e.getAttribute('href'));
    });
    linksfinal = links.map(link => baseUrl + link);
    console.log(linksfinal);

    let pattern = /\\/kategories-vivliwn\/(\d+)/;

    linksfinal.forEach(function(link) {
      let match = link.match(pattern);
      if (match) {
        let number = match[1];
        console.log(number);
        ISBN.push(number);
        console.log(number);
      }
    });

    console.log("LINKS TOTAL:", linksfinal);

    for (const link of links) {
      try {
        const productPage = await browser.newPage();
        await productPage.goto(baseUrl + link, { waitUntil: 'networkidle0' });
        autoScroll(page);
        const $ = cheerio.load(await productPage.content());
        const name = $('fullBox h1.product').text().trim();
        const imageUrl = $('authorGalBox img').attr('src');
        const price = $('fullBox aligncenterMob').find('span.productPrice').last().text().trim();

        const summary = $('productDetailsBox').text().trim();

        imageUrls.push(baseUrl + imageUrl);
        names.push(name);
        summaries.push(summary);
        prices.push(price);

        await productPage.close();
      }
    }
  });

```

Σχήμα 5.2: Παράδειγμα χρήσης ssr παράλληλα με το cheerio για την συλλογή δεδομένων.

Αυτός ο κώδικας χρησιμοποιεί τις ενότητες `render.js` και `database.js` για να σαρώσει έναν ιστότοπο για πληροφορίες βιβλίων και να τις αποθηκεύσει σε μια βάση δεδομένων MySQL.

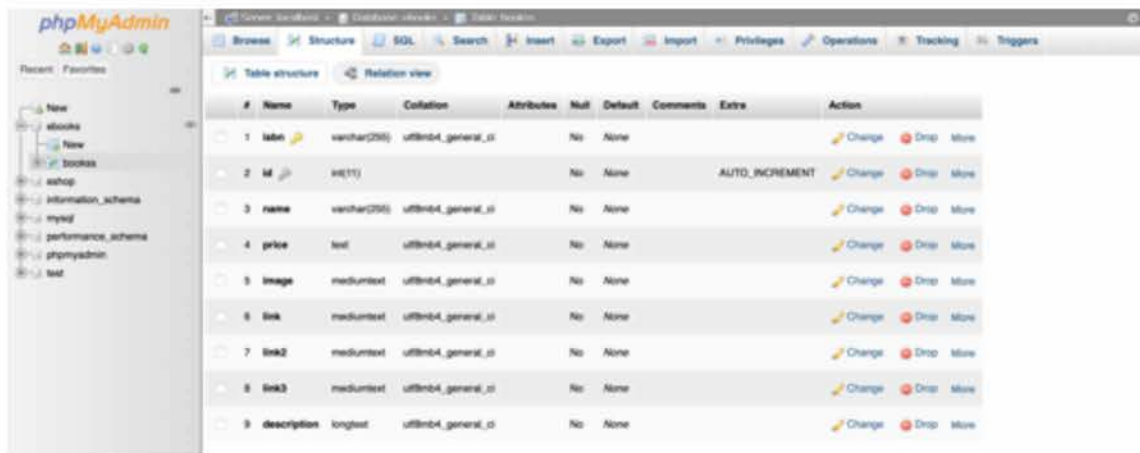
Αρχικά το js file εξάγει δύο συναρτήσεις, `ssr` και `autoScroll`. Η συνάρτηση `ssr` χρησιμοποιεί τη βιβλιοθήκη `Puppeteer` για να πλοηγηθεί σε μια καθορισμένη διεύθυνση URL και στη συνέχεια περιμένει να φορτωθεί πλήρως η σελίδα πριν επιστρέψει το περιεχόμενο της σελίδας. Η συνάρτηση `autoScroll` χρησιμοποιείται για την κύλιση της σελίδας προς τα κάτω για να φορτώσει το περιεχόμενο που έχει φορτωθεί με τεμπέλικό τρόπο (`lazy load`).

Το σενάριο ξεκινά με την εισαγωγή των εννοιών `render.js` και `database.js`. Στη συνέχεια ορίζονται ορισμένες σταθερές μεταβλητές, όπως η κύρια διεύθυνση URL για την επεξεργασία, ο επιλογέας CSS για το κουμπί "load more" και ο αριθμός των σελίδων αναζήτησης.

Στη συνέχεια, το σενάριο καλεί τη συνάρτηση `ssr` για να πλοηγηθεί στην κύρια διεύθυνση URL και να πάρει το περιεχόμενο της σελίδας. Παράλληλα, χρησιμοποιείται η συνάρτηση `evaluate` για να επιλέξει όλα τα στοιχεία με την κλάση CSS `.books-details-lft a` και να εξα-

χθούν τα χαρακτηριστικά href τους. Επίσης, το σενάριο χρησιμοποιεί τη συνάρτηση map για να προσθέσει τη βασική διεύθυνση URL σε κάθε σύνδεσμο, δημιουργώντας έναν πίνακα πλήρων διευθύνσεων URL.

Το σενάριο χρησιμοποιεί μια κανονική έκφραση για να εξάγει το ISBN από κάθε σύνδεσμο και να το αποθηκεύσει στον πίνακα ISBN. Ακολούθως, χρησιμοποιεί ένα loop για να επισκεφτεί όλες τις διευθύνσεις URL της σελίδας. Σε κάθε σελίδα, χρησιμοποιεί τη συνάρτηση goto για να πλοηγηθεί στη διεύθυνση URL και στη συνέχεια χρησιμοποιεί τη συνάρτηση content για να πάρει την HTML της σελίδας. Με την βοήθεια της βιβλιοθήκης cheerio, αναλύεται η HTML και εξάγεται το όνομα του βιβλίου, η διεύθυνση URL της εικόνας, η τιμή του βιβλίου και η περίληψη. Αυτά αποθηκεύονται στη συνέχεια σε διάφορες μεταβλητές που έχουν οριστεί, και τέλος προσθέτονται στο database για να προβληθούν στην σελίδα.



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)	utf8mb4_general_ci		No	None		AUTO_INCREMENT	Change Drop More
2	isbn	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
3	title	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
4	price	text	utf8mb4_general_ci		No	None			Change Drop More
5	image	mediumtext	utf8mb4_general_ci		No	None			Change Drop More
6	link	mediumtext	utf8mb4_general_ci		No	None			Change Drop More
7	link2	mediumtext	utf8mb4_general_ci		No	None			Change Drop More
8	link3	mediumtext	utf8mb4_general_ci		No	None			Change Drop More
9	description	longtext	utf8mb4_general_ci		No	None			Change Drop More

Σχήμα 5.3: Επιστροφή σελίδας σε html για την απόξεση της.

Όπως διαφαίνεται και από τον πίνακα books, το ISBN αποτελεί πρωτεύον κλειδί του κάθε βιβλίου και είναι ένας εύκολος τρόπος εύρεσης του κάθε ανάλογου βιβλίου στο database (Κάθε βιβλίο έχει ένα δικό του ξεχωριστό ISBN). Ομοίως το id βοηθά στην εύρεση του κάθε βιβλίου όπως για παράδειγμα με ένα SQL query (Select * where id='x'). Επίσης, υπάρχουν μεταβλητές όπως name, description, image κ.λ.π. Το description για παράδειγμα αποθηκεύει την περίληψη του κάθε βιβλίου.

Εκτός από τον πίνακα books, στον πίνακα bookprices, αποθηκεύονται όλα οι σύνδεσμοι και η τιμή του κάθε βιβλίου. Αυτό είναι ιδιαίτερα χρήσιμο, καθώς το κάθε βιβλίο μπορεί να βρίσκεται σε περισσότερες από μια σελίδες με διαφορετική τιμή. Για την δημιουργία του πίνακα αυτού, είναι αναγκαία η σύνδεση του πίνακα μας με τον αρχικό πίνακα με μια unique τιμή, όπως είναι το id, ή το isbn.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	book_id	int(11)			Yes	NULL			Change Drop More
3	link	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
4	price	text	utf8mb4_general_ci		Yes	NULL			Change Drop More

Check all With selected: Browse Change Drop Primary Unique Index Spatial Fulltext Add to or
 Remove from central columns

Σχήμα 5.4: Πίνακας database για συλλογή κόστους και συνδέσμου.

5.3 Backend

Για την σύνδεση της βάσης δεδομένων (π.χ. MySQL) με τον διακομιστή χρησιμοποιώντας τη γλώσσα προγραμματισμού Node.js, θα πρέπει να εγκατασταθεί το επίσημο εργαλείο για τη σύνδεση με τη βάση δεδομένων mysql, το οποίο ονομάζεται "mysql", με την εντολή "npm install mysql". Στη συνέχεια, φορτώνοντας το mysql στο αρχείο JavaScript υπάρχει η δυνατότητα σύνδεσης της βάσης δεδομένων μέσω ενός συνδέσμου (connection string). Χρησιμοποιώντας τη μέθοδο `createConnection()` του mysql, δημιουργείται ένας νέος σύνδεσμος με τη βάση δεδομένων, και στη συνέχεια με τη μέθοδο `query()` εκτελούνται ερωτήματα SQL στη βάση δεδομένων. Μπορούν παράλληλα να οριστούν προαιρετικά παράμετροι όπως το όνομα χρήστη και το συνθηματικό για τη σύνδεση.

```

var mysql = require('mysql');

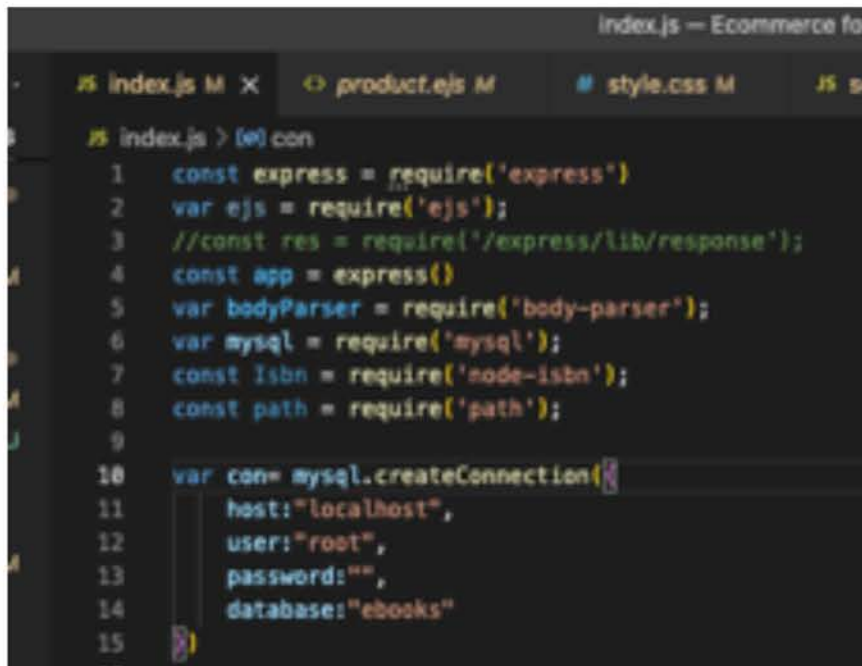
var con = mysql.createConnection({
  host: "localhost",
  user: "yourusername",
  password: "yourpassword"
});

con.connect(function(err) {
  if (err) throw err;
  console.log("Connected!");
});

```

Σχήμα 5.5: Εδραίωση σύνδεσης μεταξύ MySQL database και Node.js.

source: <https://w3school.com>



```

Index.js — Ecommerce for
# index.js M X  product.ejs M  style.css M  # se

# index.js > [0] con
1  const express = require('express')
2  var ejs = require('ejs');
3  //const res = require('/express/lib/response');
4  const app = express()
5  var bodyParser = require('body-parser');
6  var mysql = require('mysql');
7  const isbn = require('node-isbn');
8  const path = require('path');
9
10 var con= mysql.createConnection({
11   host:"localhost",
12   user:"root",
13   password:"",
14   database:"ebooks"
15 }

```

Σχήμα 5.6: Στην περίπτωση της ιστοσελίδας που δημιουργήσαμε, η σύνδεση μεταξύ MySQL και Node.js φαίνεται κάπως έτσι.

Στη συνέχεια, με την βοήθεια της express, αφού γίνει η εγκατάσταση στην node "npm install express –save", υπάρχει η δυνατότητα κατασκευής του server.

Η συνάρτηση `app.listen()` χρησιμοποιείται για τη δέσμευση και την ακρόαση των συνδέσεων στον καθορισμένο κεντρικό υπολογιστή και τη θύρα. Εάν ο αριθμός θύρας παραλειφθεί ή είναι 0, το λειτουργικό σύστημα θα αναθέσει μια αυθαίρετη αχρησιμοποίητη θύρα, κάτι που είναι χρήσιμο για περιπτώσεις όπως αυτοματοποιημένες εργασίες (δοκιμές κ.λ.π.).



```

app.listen(3000, function () {
  console.log('Example app listening on port 3000!');
});

```

Σχήμα 5.7: Προεπιλογή πορτ σελίδας.

Χρησιμοποιώντας τη μέθοδο `app.get()` της express, πραγματοποιείτε διαχείρισή σε διάφορες αιτήσεις HTTP GET της εφαρμογή μας. Το path είναι μια συμβολοσειρά που αντιπροσωπεύει τη διαδρομή URL που θα χειριστεί η μέθοδος `app.get()`. Για παράδειγμα, το `/'` αντιπροσωπεύει τη διαδρομή root της εφαρμογής, το `'/users'` αντιπροσωπεύει τη διαδρομή `'/users'` κ.ο.κ.

Η μέθοδος `res.render()` χρησιμοποιείται για την απόδοση ενός προτύπου προβολής σε μια εφαρμογή Express.js. Συνήθως χρησιμοποιείται στη συνάρτηση επανάκλησης μιας διαδρομής `app.get()`, η οποία καλείται όταν γίνεται μια αίτηση GET στην καθορισμένη διαδρομή.

Όπως φαίνεται παρακάτω, χρησιμοποιώντας την `app.get()` και την `res.render()`, η διαδρομή `app.get()` ακούει για αιτήσεις GET στη ριζική διαδρομή ("/") και εμφανίζει την προβολή `index` όταν γίνεται μια αίτηση. Η προβολή `index.ejs` όπως και οι υπόλοιπες, βρίσκονται στον φάκελο `views`. Η μηχανή προβολής έχει οριστεί σε `ejs`, πράγμα που σημαίνει ότι η εφαρμογή θα αναζητήσει ένα αρχείο `index.ejs` στον κατάλογο `views` για να το εμφανίσει ως απάντηση.

```
app.set('view engine', 'ejs')
```

Σχήμα 5.8: Ορισμός φακέλου `views` ως μηχανή προβολής `ejs` αρχείων.

```
app.get('/', function (req, res) {  
  res.render('index', {  
    names: name,  
    id: id,  
    img: img,  
    title: "books:"  
  });  
});  
});
```

Σχήμα 5.9: Χρήση `app.get()` και `res.render()`

Ως δεύτερο όρισμα της `res.render()`, καθορίζονται διάφορα δεδομένα και μεταβλητές. Τα δεδομένα αυτά συλλέγονται με ένα SQLqueryy απο το database, αποθηκεύονται σε ένα πίνακα και στην συνέχεια γίνονται map σε ξεχωριστές μεταβλητές, π.χ μια μεταβλητή για όλες τις εικόνες, μια για όλα τα ονόματα κ.τ.λ.

```

con.connect(function(err) {
  if (err) throw err;
  con.query("SELECT name,id,image FROM bookss", function (err, result, fields) {
    if (err) throw err;
    Orders = result;

    const name = Orders.map(function(ele){
      return ele.name;
    })

    const id = Orders.map(function(ele){
      return ele.id;
    })
    const img = Orders.map(function(ele){
      return ele.image;
    })
  })
})

```

Σχήμα 5.10: Συλλογή δεδομένων μέσω Sqlqueryry.

Η μέθοδος `app.use()` χρησιμοποιείται για την σύνδεση middleware συναρτήσεων σε ένα συγκεκριμένο path σε μια εφαρμογή Express.

Η συνάρτηση middleware `express.static` είναι μια ενσωματωμένη συνάρτηση middleware στο Express που παρέχει στατικά στοιχεία, όπως αρχεία HTML, εικόνες κλπ. Απαιτεί τη διαδρομή προς τον κατάλογο που περιέχει τα στατικά στοιχεία ως όρισμά της.

Η συνάρτηση `”app.use(express.static(’public’))”` συνδέει τη συνάρτηση middleware `express.static` στη ρίζα και παρέχει τα στατικά στοιχεία που βρίσκονται στο κατάλογο `public`. Αυτό μας επιτρέπει την προσπέλαση όλων των αρχείων html με μια αίτηση όπως `”http://localhost:3000/file.html”`.

```

app.use(express.static('public'))
app.use('/css', express.static(__dirname + 'public/css'))
app.use('/js', express.static(__dirname + 'public/js'))
app.use('/img', express.static(__dirname + 'public/img'))

```

Σχήμα 5.11: Προσπέλαση του φακέλου `public`, καθώς και τους φακέλους `js`, `css`, `images` που βρίσκονται σε αυτό.

5.4 Routes

Το routing αναφέρεται στον τρόπο με τον οποίο μια εφαρμογή απαντά σε μια αίτηση πελάτη σε έναν συγκεκριμένο τερματικό σημείο, το οποίο είναι ένα URI (ή διαδρομή) και μια συγκεκριμένη μέθοδο αιτήματος HTTP (GET, POST κλπ).

Στο Express, τα routes καθορίζονται χρησιμοποιώντας μεθόδους όπως το `app.get()` ή το

`app.post()`, οι οποίες αντιστοιχούν στις μεθόδους HTTP GET και POST αντίστοιχα. Κάθε route μπορεί να έχει μία ή περισσότερες συναρτήσεις handler, οι οποίες εκτελούνται όταν το route ταιριάζει. Με την χρήση του κώδικα παραπάνω, δημιουργείται ένα routing στην αρχική μας σελίδα. Παράλληλα, δημιουργείται ένα routing μέσα σε κάθε επιπλέον προσβάσιμη σελίδα.

```

app.get('/search=', async function (req, res) {
  |
  con.query("SELECT name,id,image FROM bookss", function (err, results, fields) {
    if (err) throw err;
    Orders = results;

    const searchname = Orders.map(function(ele){
      | return ele.name;
    })

    const searching = Orders.map(function(ele){
      | return ele.image;
    })

    var toSearch = JSON.stringify(req.query).replace(/"/g, '');
    toSearch = toSearch.replace("'", '');
    toSearch = toSearch.replace(':', '');
    //FIND THE BOOKS THAT MATCHES THE TITLE YOU SEARCHED
    con.query('SELECT name,image,id FROM bookss WHERE name Like ?', ['%${toSearch}%'], function (err, result) {
      if(err) throw err;
      var Searchedbooks = result;
      const names = Searchedbooks.map(function(ele){
        | return ele.name;
      })
      const images = Searchedbooks.map(function(ele){
        | return ele.image;
      })
      const ids = Searchedbooks.map(function(ele){
        | return ele.id;
      })

      res.render('search', {
        | name: names,
        | id: ids,
        | image: images,
        | search: toSearch,

```

Σχήμα 5.12: Routing to Search.ejs

5.4.1 Results

Η `JSON.stringify()` είναι μια μέθοδος που μετατρέπει ένα αντικείμενο ή μια τιμή JavaScript σε συμβολοσειρά JSON. Λαμβάνει ως πρώτο όρισμα το αντικείμενο που θα μετατραπεί και ένα προαιρετικό δεύτερο όρισμα που ονομάζεται συνάρτηση αντικατάστασης, η οποία μπορεί να χρησιμοποιηθεί για τον μετασχηματισμό της εξόδου.

Στην περίπτωση της `JSON.stringify(req.query)`, το `req.query` είναι ένα αντικείμενο που περιέχει την αναλυμένη συμβολοσειρά ερωτήματος της αίτησης. Η ιδιότητα `query` του αντικειμένου `request` είναι ένα αντικείμενο που περιέχει μια ιδιότητα για κάθε παράμετρο της

συμβολοσειράς ερωτήματος στη διαδρομή. Για παράδειγμα, εάν η διαδρομή είναι /search?q=tacos, τότε το req.query θα είναι q: "tacos".

Σε αυτό το σημείο, η χρήση της JQuery μας φαίνεται ιδιαίτερα χρήσιμη. Εκτελώντας τον παρακάτω κώδικα, το url μετατρέπεται σε αυτό που ζητείται από τον χρήστη.

```
$(document).ready(function(){
  $("#auto").attr("placeholder", "Search...");
});

// When user hits enter
$("#auto").keypress(function(event) {
  if (event.keyCode == 13) {
    var str = $("#auto").val();
    if(str != ""){
      window.location.replace('/search?'+str);
    }
  }
});
```

Σχήμα 5.13: Αλλαγή url

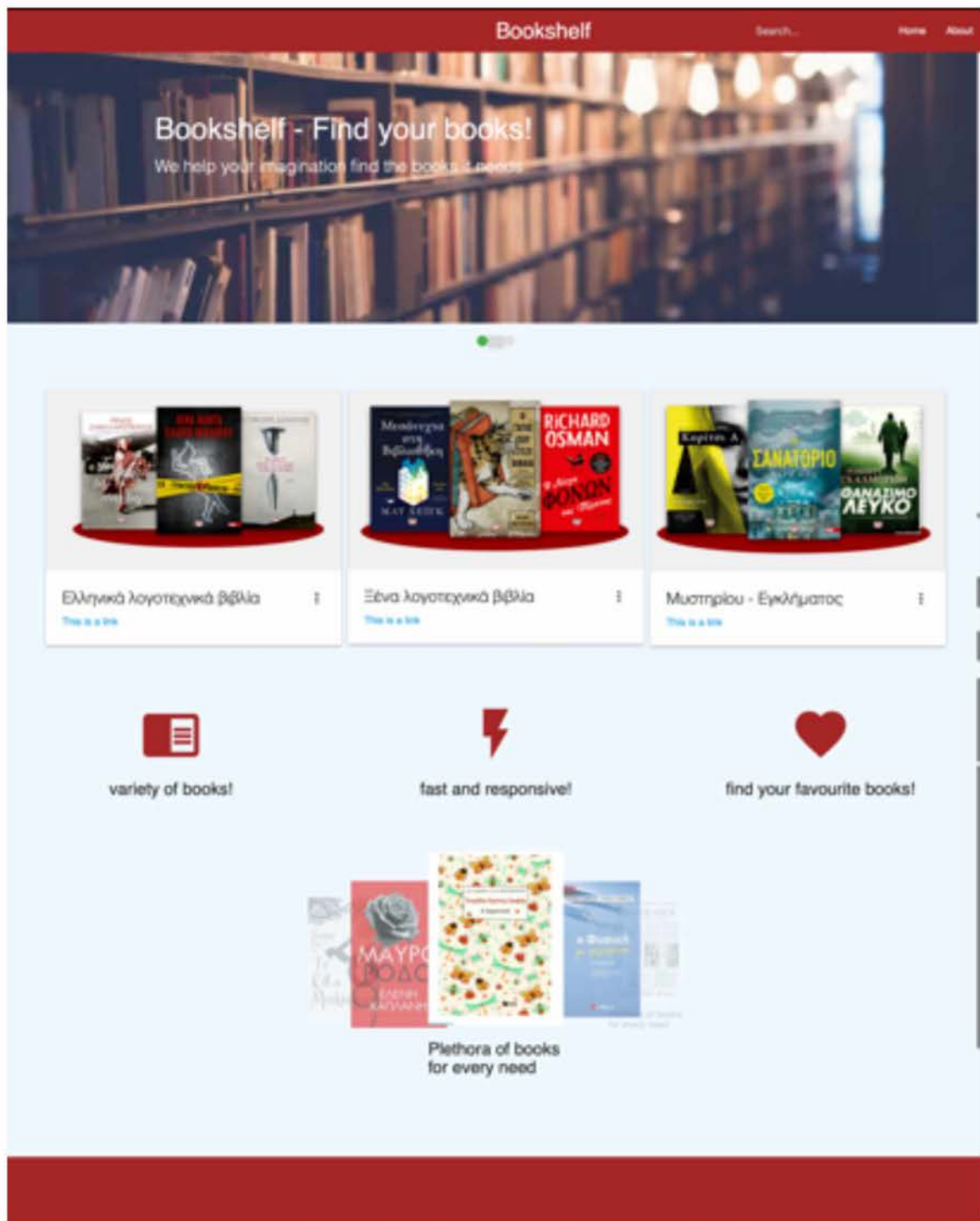
Στην συνέχεια, με την κλήση του JSON.stringify(req.query), γίνεται συλλογή του βιβλίου που ψάχνει ο χρήστης, και στη συνέχεια σύγκριση με τα βιβλία της βάσης δεδομένων μας.

```
//FIND THE BOOKS THAT MATCHES THE TITLE YOU SEARCHED
con.query('SELECT name,image,id FROM bookss WHERE name like ?', ['%${toSearch}%'], function (err, result) {
  if(err) throw err;
```

Σχήμα 5.14: Εύρεση όλων των βιβλίων από την database

5.5 Frontend

Με την είσοδο του χρήστη στη σελίδα, μπορεί με σχετική ευκολία να περιηγηθεί και να βρει διάφορα βιβλία τα οποία τον ενδιαφέρουν.



Σχήμα 5.15: Αρχική Σελίδα

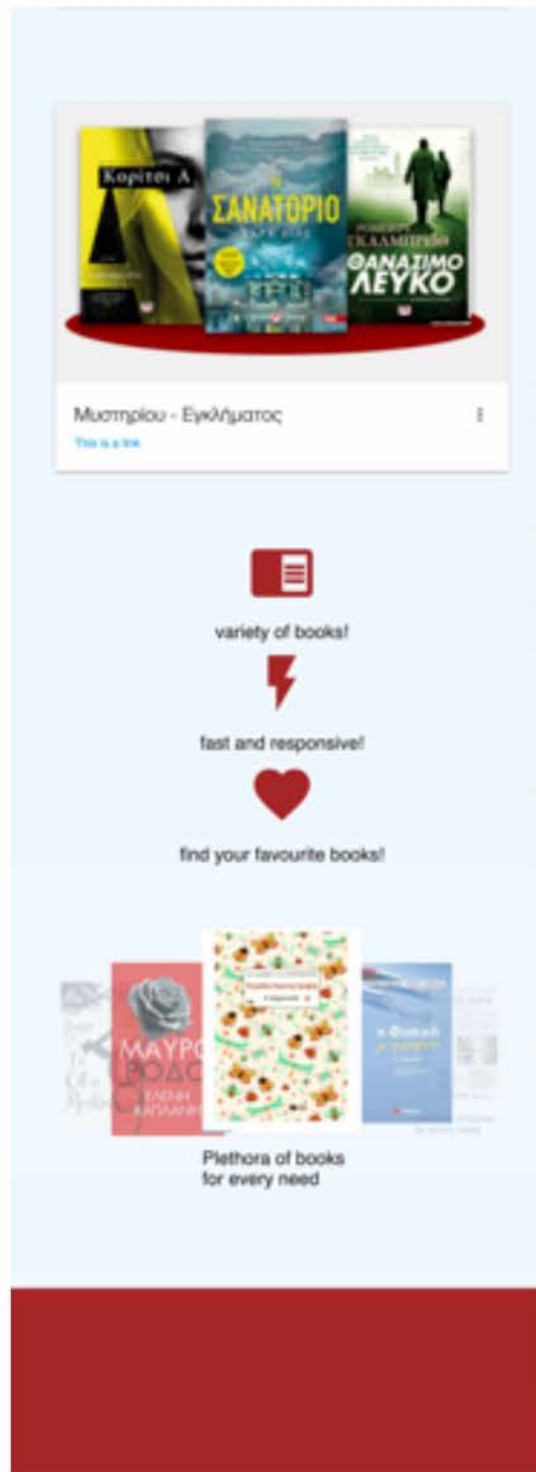
5.5.1 Optimization

Με την βοήθεια του front-end framework της Materializecss, η σελίδα εκτός από την ελκυστική εμφάνιση, έχει σχεδιαστεί ώστε να ανταποκρίνεται και να είναι βελτιστοποιημένη για διάφορα μεγέθη οθόνης. Περιλαμβάνει ένα ανταποκρινόμενο σύστημα πλέγματος

που επιτρέπει να δημιουργούνται διατάξεις που προσαρμόζουν αυτόματα τη διάταξή τους με βάση το μέγεθος της οθόνης. Αυτό έχει ως αποτέλεσμα ο ιστότοπος να έχει καλή εμφάνιση και να είναι εύχρηστος σε ένα ευρύ φάσμα συσκευών, συμπεριλαμβανομένων των smartphones.



Σχήμα 5.16: Αρχική Σελίδα για tablets και smartphones

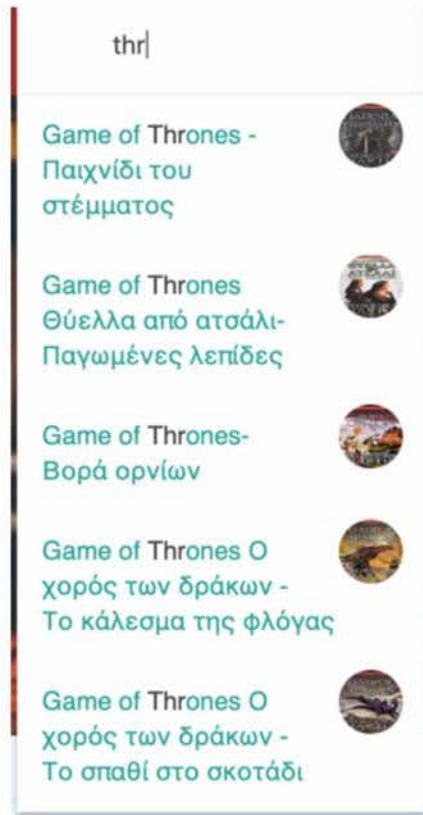


Σχήμα 5.17: Αρχική Σελίδα για tablets και smartphones

5.5.2 Περιήγηση user

Ο χρήστης μέσω του Search-bar μπορεί να περιηγηθεί σε όλα τα βιβλία που υπάρχουν διαθέσιμα στη σελίδα ανά πάσα στιγμή. Με την βοήθεια της javascript, δημιουργήθηκε ένα παράθυρο autocomplete, το οποίο βοηθά τον χρήστη να εντοπίσει ευκολότερα αυτό που ψά-

χνει ή να δεί περισσότερες επιλογές.

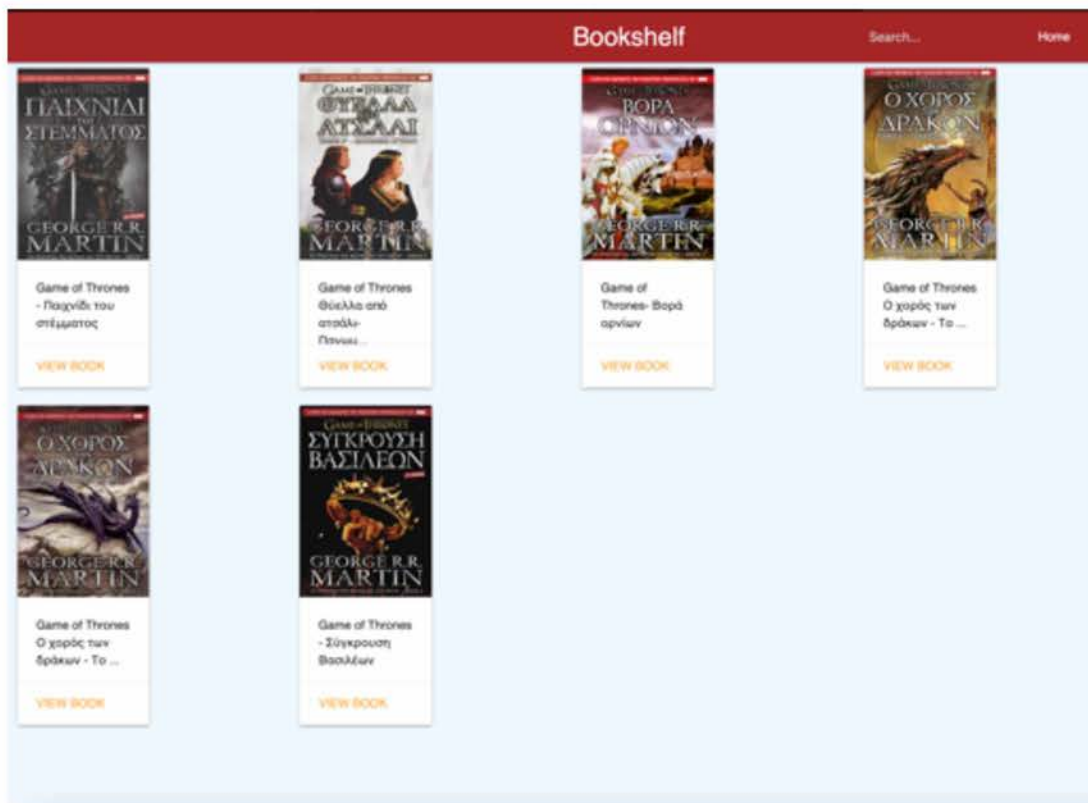


Σχήμα 5.18: Εύρεση βιβλίων.

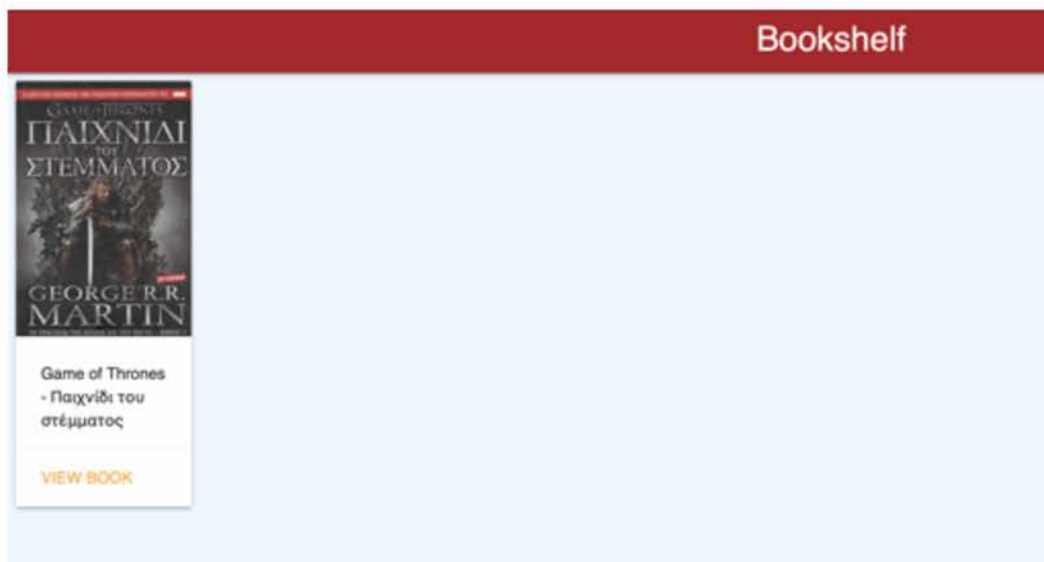
```
document.addEventListener('DOMContentLoaded', function() {  
  var auto = document.querySelectorAll('.autocomplete');  
  var instances = M.Autocomplete.init(auto, {  
  
    data: data,  
    limit: 5  
  });  
});
```

Σχήμα 5.19: Δημιουργία autocomplete function μέσω javascript, όπου data περιέχει πληροφορίες του κάθε βιβλίου απο το database και limit το όριο επιλογών που είναι διαθέσιμες

Στην συνέχεια, ο χρήστης μπορεί είτε να επιλέξει ένα από τα βιβλία που το εμφανίστηκαν στην λίστα, ή να πληκτρολογήσει enter για να δει όλες τις σχετικές διαθέσιμες επιλογές.



Σχήμα 5.20: Προβολή όλων των σχετικών βιβλίων.



Σχήμα 5.21: Επιλογή συγκεκριμένου βιβλίου.

5.5.3 Product page

Με την επιλογή κάποιου βιβλίου, ο χρήστης μεταφέρεται στην σελίδα Product page, όπου θα έχει την δυνατότητα να δει πιο λεπτομερείς πληροφορίες για το επιλεγμένο βιβλίο, όπως

είναι η περίληψη, η τιμή, καθώς και τα καταστήματα διαθεσιμότητας όπου μπορεί να το προμηθευτεί.



Γ'Η Αλεξάνδρα Πανοπούλου είναι όμορφη, πλούσια και σύντομα θα γίνει πολύ διάσημη κατακτώντας τον τίτλο της βασίλισσας της ελληνικής τηλεόρασης. Παντρεύτηκε τον μεγάλο της έρωτα, τον Λεωνίδα, ο οποίος την λατρεύει, κι απέκτησε μαζί του έναν γιο, τον Όμηρο. Έχει πάντα κοντά της την παιδική της φίλη Ελίνα, που τρέχει σε κάθε της ανάγκη, και τους γονείς της, που δεν της χαλούν κανένα χατίρι. Θα μπορούσε να πει κάποιος πως είναι μια πολύ τυχερή κι απόλυτα ευτυχημένη γυναίκα. Όμως δεν είναι. Στην ιστορία του βιβλίου η ηρωίδα δεν είναι η Αλεξάνδρα, η πρωταγωνίστρια του. Πραγματική ηρωίδα είναι η Ελίνα, και τραγικοί ήρωες ο Λεωνίδας και ο μικρός Όμηρος. Σύντομα θα καταλάβετε το γιατί... Ο Ανήφορος Σιλλάβος γεννηθήκαμε και πελευούμε όλη μας τη ζωή να γίνουμε ελεύθεροι. Στο μέσο της δεκαετίας του '40, αμέσως μετά τον Ζορμπά, ο Καζαντζάκης γράφει τον Ανήφορο, ένα κείμενο εσωτερικό, που το διακρίνει μελαγχολία βαθιά και λυτρωτική. Η δράση του εκτυλίσσεται αμέσως μετά τον πόλεμο, σε Κρήτη και Αγγλία. Ο Κοσμάς...Οι κότες Ο Έντουαρντ Φόσκα είναι δολοφόρος. Η Μαριάννα είναι βέβαιη γι' αυτό. Όμως, δεν μπορεί να τον αγγίξει κανείς. Όμορφος και χαρισματικός, ο Φόσκα διδάσκει αρχαία τραγωδία στο Πανεπιστήμιο του Κέμριτζ κι λατρεύεται τόσο από το προσωπικό όσο και από τους φοιτητές του - ειδικά από τα μέλη μιας κλειστής ομάδας... Σίρα Σάκραστο Λυκόφως του 19ου αιώνα η Ερμούσση είναι μια από τις πλουσιότερες πάλαι της Ελλάδας. Το χρωστά στους πρόσφυγες από τα καμένα νησιά της Επανάστασης, κυρίως Χιώτες αλλά και Κασσιώτες και Ψαριανούς, που βρήκαν στη Σύρο καταφύγιο. Οι ίδιοι οι Ψαριανοί, κατά βάση άνθρωποι φτωχοί, καθολικοί στο θρήσκευμα, βρίσκονται μαζεμένοι...]

[View on kyknos.cy €](#)

Cost on kyknos.cy: €16.40

Σχήμα 5.22: Product page ex.1.



Γ'Καμιά φορά, το να μην εγκαταλείψεις κάτι μπορεί να ισοδυναμεί με θάνατο. Καμιά φορά, ζωή σημαίνει να παρατάς αυτό που κάποτε σου έδωσε τη ζωή. *Υποθέτω πως ακόμα και σήμερα κάποιος μπορεί να νιώσουν απογοήτευση βλέποντάς με να περιπλανιέμαι σε παλιούς και διάφορους δρόμους ψάχνοντας τη συνταγή της ευτυχίας, για να καταλήξω ότι, δυστυχώς, δεν υπάρχει συνταγή κι ότι δεν έχει νόημα να χθίνουμε τον καιρό μας αναζητώντας συνταγές. *Θα μου άρεσε πολύ να μ' αγαπήνε όσοι αγαπά, κι αυτοί που αγαπά με μου το ανταποδίδουν. Αν όμως κάποιος δε μ' αγαπά, ως μου το πει κι ως φύγει. Κι αν δε βρούκε το κουράγιο να το πει, ως φύγει χωρίς να το πει. Ούτε μνησκακία ούτε πεκρία είν' αυτό. Απλά, δε θέλω να ζω δίπλα σε κάποιον που δε θέλει να ζει δίπλα σ' εμένα. *Κανείς δεν αλλάζει επειδή "αυτό απαιτείται". Κανείς δε μεταβάλλεται σ' αλήθεια μέσω του φόβου. Κανείς δεν αναπτύσσεται με την καταπίεση. * ΠΟΙΟΣ ΕΙΜΑΙ; ΠΟΥ ΠΗΓΑΙΝΩ; ΜΕ ΠΟΙΟΝ; *Αυτά είναι τα τρία υπαρκτά ερωτήματα που συνοδεύουν τον άνθρωπο από τη στιγμή που άρχισε να διαμορφώνει τη σκέψη του. Ερωτήματα που δεν μπορούμε ν' αποφύγουμε, καθώς αποτελούν μέρος κάθε πορείας που σχεδιάζουμε. Ερωτήματα που θα πρέπει ν' απαντηθούν ένα προς ένα --αν σκοπεύουμε ν' αντιμετωπίσουμε την πρόκληση που ο Καρλ Ρόιζερς αποκάλεσε «διαδικασία μετατροπής σε πρόσωπο»--, γιατί μόνο αναζητώντας μιν απάντηση σ' αυτό μπορεί κανείς να μάθει τα αναγκαία για να πάει μπροστά στη ζωή του. * Ραγισμένο ειδικολογικά τη δεκαετία του 1960 στην Ελλάδα, μεταξύ άλλων, αναδείχθηκαν τα μεγαλύτερα εγχώρια κινηματογραφικά αστέρια. Ένα από αυτά υπήρξα κι εγώ, ο Βίκτορας Ντοταράς. Ελάχιστα έμαθον πως τα πρώτα μου εφηβικά βήματα τα έκανα στα διαβήτητα στενωσόκκια της πιο αμαρτωλής γειτονιάς, της Τρούμπας, με τα μπαντέλα και τις γυναίκες που...Ο έρωτας είναι παιχνίδι, μαρό μουΗ ηρωίδα είναι μια από τις συνηθασμένες περιπτώσεις που έλαβε την πάγια εντολή να σπουδάσει, να γίνει ένα αξιοπρόσεχτο μέλος της αγέλης και, αν είναι δυνατόν, ο αρχηγός της. Όλα τα άλλα μοιάζαν μόνο με χόσιμο χρόνο. Κι έτσι βρέθηκε φοιτήτρια στα Εζάρχεια, στα μπαρ, στις καταλήψεις και στη γεμάτη... Γυνακκία υπόθεση-Είνα υψηλό το τίμημα που καλούνται να πληρώσουν τα φαντάσματα. Κι αυτό ακριβώς είμαι πλέον εγώ. Ένα φάντασμα, ένα στοιχειό. Με τη δική μου θέληση, με τον δικό μου τρόπο. Ένα φάντασμα που κεντάει αδύρμηλα ανάμεσα στον κόσμο. Κι εκείνος Εκείνος με καταξύνει, αλλά δε με βλέπουν. Με ακούνε αλλά...*Καμιά φορά, το να μην εγκαταλείψεις κάτι μπορεί να ισοδυναμεί με

[View on kyknos.cy](#)

Cost on kyknos.cy: €24.15

[View on Parga.cy](#)

Cost on Parga.cy: €14.83

Σχήμα 5.23: Product page ex.2.

Κεφάλαιο 6

Συμπεράσματα

6.1 Σύνοψη και συμπεράσματα

Προς το παρόν δεν υπάρχουν παρόμοιοι ιστότοποι στην Κύπρο που να επικεντρώνονται αποκλειστικά στην προώθηση και πώληση τοπικών βιβλίων, γεγονός που καθιστά τον ιστότοπο αυτό, μια μοναδική και καινοτόμο ιδέα. Παρέχοντας μια πλατφόρμα για τα τοπικά βιβλιοπωλεία να διαθέτουν τα προϊόντα τους και για τους καταναλωτές να εντοπίζουν και να αγοράζουν εύκολα βιβλία από τοπικά βιβλιοπωλεία, ο ιστότοπος θα καλύψει ένα κενό στην αγορά και θα προσφέρει μια πολύτιμη υπηρεσία τόσο για τα τοπικά βιβλιοπωλεία όσο και για τους καταναλωτές. Επιπλέον, παρέχοντας μια μηχανή αναζήτησης για βιβλία, οι καταναλωτές θα διευκολυνθούν στην εξεύρεση του φθηνότερου βιβλίου, κάτι που δεν παρέχεται από ξένους ιστότοπους όπως το Amazon και το Skrutz.

Ο στόχος της εργασίας ήταν να αναπτυχθεί μια σύνθετη εφαρμογή-μηχανή αναζήτησης χρησιμοποιώντας την Node.js, html.js,css και Materialize Css. Ο λόγος για τη χρήση της nodejs ως back-end αναφέρεται σαφώς στο θεωρητικό κεφάλαιο με λεπτομερή περιγραφή. Εξηγείται παράλληλα η λειτουργία της εφαρμογής στον τελικό χρήστη μέσω του front-end, καθώς και συγκεκριμένα ο τρόπο λειτουργίας της βάσης δεδομένων και του back-end.

6.2 Μελλοντικές επεκτάσεις

Υπάρχουν πολλοί τρόποι με τους οποίους η παραπάνω σελίδα θα μπορούσε να βελτιωθεί στο μέλλον. Η σελίδα χτίστηκε έτσι ώστε να μπορεί να ικανοποιήσει τις προκλήσεις και τις ευκαιρίες που σχετίζονται με τη δημιουργία μιας μηχανής αναζήτησης για βιβλία σε

μια αναπτυσσόμενη χώρα. Αναλύοντας την τρέχουσα κατάσταση της πρόσβασης σε βιβλία στη χώρα και διεξάγοντας έρευνα χρηστών, εντοπίσαμε μια σημαντική ανάγκη για μια μηχανή αναζήτησης που θα επέτρεπε την αποτελεσματικότερη ανακάλυψη σχετικών βιβλίων. Ωστόσο, κάποιοι τρόποι με τους οποίους θα μπορούσε να βελτιωθεί η σελίδα είναι:

- **Log in/Sign up.** Η Δημιουργία λογαριασμού στον χρήστη θα του επιτρέψει να μπορεί να αποθηκεύει προσωρινά τα βιβλία τα οποία του τράβηξαν το ενδιαφέρον, και να αποφασίσει σε μεταγενέστερο στάδιο αν θέλει να τα αγοράσει ή όχι.
- **Personalization.** Η προσθήκη εξατομικευμένων συστάσεων, όπως προϊόντα που είναι όμοια με αυτά που ο πελάτης έχει δει ή αγοράσει στο παρελθόν στην αρχική σελίδα, θα μπορούσε να βοηθήσει στην αύξηση της δέσμευσης των πελατών και των πωλήσεων.
- **Expand:** Καθώς ο ιστότοπος αναπτύσσεται, θα μπορούσε να επεκταθεί για να συμπεριλάβει περισσότερα τοπικά βιβλιοπωλεία. Αυτό θα έδινε στους καταναλωτές ακόμη περισσότερες επιλογές κατά την αναζήτηση των καλύτερων τιμών των βιβλίων.
- **Εξυπηρέτηση πελατών:** Η βελτίωση της εξυπηρέτησης πελατών με την προσθήκη λειτουργιών όπως η ζωντανή συνομιλία ή μια ενότητα Συχνών Ερωτήσεων μπορεί να βοηθήσει στην έγκαιρη αντιμετώπιση των ανησυχιών και των ερωτήσεων των πελατών.

Βιβλιογραφία

- [1] Martin Kleppmann. *Designing data-intensive applications: The big ideas behind reliable, scalable, and maintainable systems.* ” O’Reilly Media, Inc.”, 2017.
- [2] Introduction to web development. <https://www.digitalocean.com/community/learning-paths/web-development>. Ημερομηνία πρόσβασης: 30-12-2022.
- [3] Backend development in 2020: key languages, technologies, features. <https://ddi-dev.com/blog/programming/backend-development-key-languages-technologies-features-in-2020/>. Ημερομηνία πρόσβασης: 30-12-2022.
- [4] Backend developer skills you must have. <https://www.interviewbit.com/blog/backend-developer-skills/>. Ημερομηνία πρόσβασης: 30-12-2022.
- [5] Javascript. <https://developer.mozilla.org/en-US/docs/Web/JavaScript>. Ημερομηνία πρόσβασης: 30-12-2022.
- [6] Advantages of javascript. <https://codeinstitute.net/global/blog/advantages-of-javascript/>. Ημερομηνία πρόσβασης: 30-12-2022.
- [7] The state of javascript 2018. <https://www.freecodecamp.org/news/the-state-of-javascript-2018-8322bcc51bd8/>. Ημερομηνία πρόσβασης: 30-12-2022.
- [8] Node.js. <https://developer.mozilla.org/en-US/docs/Glossary/Node.js>. Ημερομηνία πρόσβασης: 30-12-2022.
- [9] Mike Cantelon, Marc Harter, TJ Holowaychuk, and Nathan Rajlich. *Node.js in Action.* Manning Greenwich, 2014.

- [10] Ethan Brown. *Web development with node and express: leveraging the JavaScript stack*. O'Reilly Media, 2019.
- [11] Express. fast, unopinionated, minimalist web framework for node.js. <https://expressjs.com>. Ημερομηνία πρόσβασης: 30-12-2022.
- [12] Daniel Jacobson, Greg Brail, and Dan Woods. *APIs: A strategy guide*. ” O'Reilly Media, Inc.”, 2012.
- [13] What is an api? <https://www.programmableweb.com>. Ημερομηνία πρόσβασης: 02-01-2023.
- [14] Introducing json. <https://www.json.org/json-en.html>. Ημερομηνία πρόσβασης: 02-01-2023.
- [15] Jason Levitt. *Json and the dynamic script tag: easy, xml-less web services for javascript*, 2005.
- [16] Databases. <https://online.stanford.edu/courses/soe-ydatabases-databases>. Ημερομηνία πρόσβασης: 02-01-2023.
- [17] Edgar F Codd. A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387, 1970.
- [18] Felix Gessert, Wolfram Wingerath, Steffen Friedrich, and Norbert Ritter. Nosql database systems: a survey and decision guidance. *Computer Science-Research and Development*, 32(3):353–365, 2017.
- [19] In-memory database. <https://www.heavy.ai/technical-glossary/in-memory-database>. Ημερομηνία πρόσβασης: 02-01-2023.
- [20] Michael Kofler. What is mysql? In *The Definitive Guide to MySQL*, pages 3–19. Springer, 2004.
- [21] Mariella Di Giacomo. Mysql: lessons learned on a digital library. *IEEE software*, 22(3):10–13, 2005.
- [22] Jennifer Niederst Robbins. *Learning web design: A beginner's guide to HTML, CSS, JavaScript, and web graphics*. ” O'Reilly Media, Inc.”, 2012.

- [23] Chris Aquino and Todd Gandee. *Front-End Web Development: The Big Nerd Ranch Guide*. Pearson Technology Group, 2016.
- [24] William Harrel. *HTML, CSS, and JavaScript Mobile Development For Dummies*. John Wiley & Sons, 2011.
- [25] Jon Duckett. *HTML & CSS: design and build websites*, volume 15. Wiley Indianapolis, IN, 2011.
- [26] World Wide Web Consortium et al. *Html5, a vocabulary and associated apis for html and xhtml*. <http://www.w3.org/TR/html5/>, 2014.
- [27] Eric A Meyer. *CSS: The Definitive Guide: The Definitive Guide*. ” O’Reilly Media, Inc.”, 2006.
- [28] David Flanagan and Gregor M Novak. *Java-script: The definitive guide*, 1998.
- [29] Douglas Crockford. *JavaScript: The Good Parts: The Good Parts*. ” O’Reilly Media, Inc.”, 2008.
- [30] Earle Castledine and Craig Sharkie. *JQuery: Novice to Ninja*. SitePoint, 2012.
- [31] Alvin Wang, Alan Chang, Alex Mark, and Kevin Louie. *Materialize-a modern responsive front-end framework based on material design*.
- [32] Andrei Broder. A taxonomy of web search. In *ACM Sigir forum*, volume 36, pages 3–10. ACM New York, NY, USA, 2002.
- [33] Web scraping with cheerio. <https://www.digitalocean.com/community/tutorials/web-scraping-with-cheerio>. Ημερομηνία πρόσβασης: 20-01-2022.
- [34] Getting started with puppeteer. <https://developers.google.com/web/tools/puppeteer/get-started>. Ημερομηνία πρόσβασης: 20-01-2022.
- [35] Web scraping and parsing html with node.js and cheerio. <https://www.twilio.com/blog/web-scraping-and-parsing-html-with-node-js-and-cheerio>. Ημερομηνία πρόσβασης: 20-01-2022.

- [36] Web scraping using puppeteer and nodejs. <https://www.mitrahsoft.com/blog/web-scraping-using-puppeteer-nodejs>. Ημερομηνία πρόσβασης: 20-01-2022.