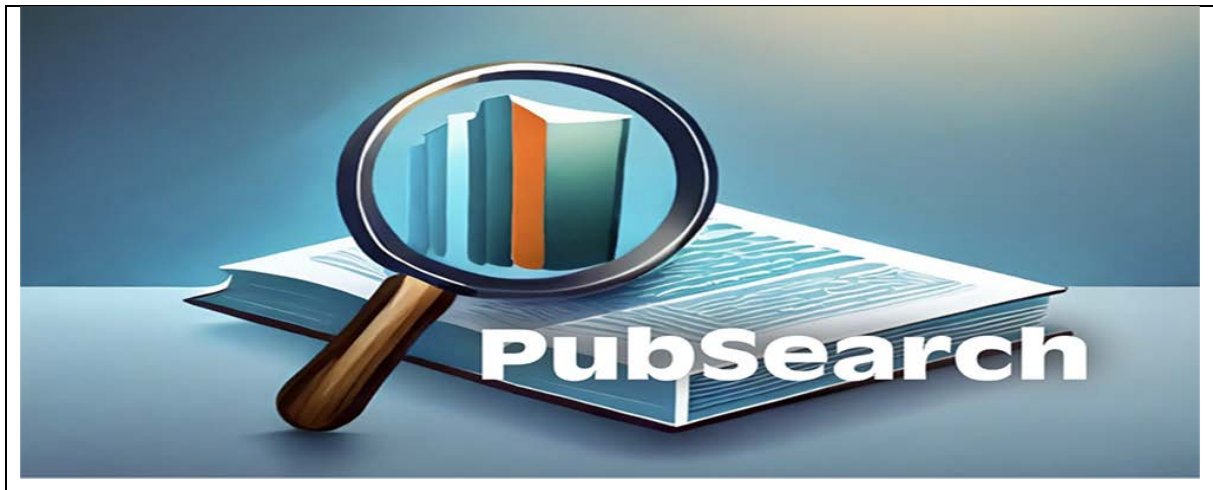




ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
«ΔΙΑΔΙΚΤΥΑΚΗ ΕΦΑΡΜΟΓΗ ΑΝΑΖΗΤΗΣΗΣ
ΒΙΒΛΙΟΓΡΑΦΙΚΩΝ ΔΗΜΟΣΙΕΥΣΕΩΝ»



Του φοιτητή
Διαμαντή Σταμάτη
Αρ. Μητρώου: 164784

Επιβλέπων
ΣΙΔΗΡΟΠΟΥΛΟΣ ΑΝΤΩΝΙΟΣ
Επίκουρος Καθηγητής

Ημερομηνία 31-01-2024

Τίτλος Δ.Ε. ΔΙΑΔΙΚΤΥΑΚΗ ΕΦΑΡΜΟΓΗ ΑΝΑΖΗΤΗΣΗΣ ΒΙΒΛΙΟΓΡΑΦΙΚΩΝ
ΔΗΜΟΣΙΕΥΣΕΩΝ

Κωδικός Δ.Ε. 22231

Όνοματεπώνυμο φοιτητή ΣΤΑΜΑΤΗΣ ΔΙΑΜΑΝΤΗΣ
Όνοματεπώνυμο εισηγητή ΑΝΤΩΝΙΟΣ ΣΙΔΗΡΟΠΟΥΛΟΣ

Ημερομηνία ανάληψης Δ.Ε. 29-05-2022

Ημερομηνία περάτωσης Δ.Ε. 31-01-2024

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Διαμαντή Σταμάτη που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιοδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

Πρόλογος

Στις μέρες μας η ανάλυση παραπομπών έχει αποκτήσει πολύ μεγάλη σημασία και χρησιμοποιείται ευρέως για την αξιολόγηση αναφορών που γίνονται σε επιστημονικά έργα. Με τον αριθμό των επιστημονικών δημοσιεύσεων να αυξάνεται καθημερινά, έχει δημιουργηθεί η ανάγκη για εργαλεία τα οποία θα βοηθήσουν να αντιμετωπίσουμε την πληθώρα των δημοσιεύσεων και να αξιοποιήσουμε την πληροφορία που μας παρέχουν. Με την ανάλυση παραπομπών, μπορούμε κατανοήσουμε τον τρόπο με τον οποίο οι επιστημονικές εργασίες συνδέονται και επηρεάζουν η μία την άλλη. Μπορούμε επίσης να δούμε ποια έργα αναφέρονται σε άλλα, ποιοι επιστήμονες επηρεάζουν άλλους, και να ανακαλύψουμε νέες τάσεις και πεδία έρευνας. Ακόμη, η ανάλυση παραπομπών μας επιτρέπει να αξιολογήσουμε την επιρροή και την αναγνωρισιμότητα των επιστημονικών περιοδικών, των εκδοτών και των ιδρυμάτων. Μπορούμε να μετρήσουμε τον αριθμό και την ποιότητα των αναφορών που λαμβάνουν, προσφέροντας έτσι μια αξιόπιστη εικόνα για τη σημασία και την επιρροή τους στην επιστημονική κοινότητα.

Περίληψη

Αντικείμενο της παρούσας διπλωματικής εργασίας είναι η ανάπτυξη μιας διαδικτυακής εφαρμογής, η οποία κατασκευάστηκε χρησιμοποιώντας την δημοφιλή βιβλιοθήκη της JavaScript React, αναπτυγμένη από το Facebook. Η εφαρμογή λειτουργεί στην πλευρά του χρήστη (client-side) και χρησιμοποιεί ασύγχρονα αιτήματα AJAX με πολλαπλές παραμέτρους ερωτημάτων, όπως το όνομα του συγγραφέα και το όνομα της δημοσίευσης για να ανακτήσει σχετικές βιβλιογραφικές πληροφορίες. Η εφαρμογή χρησιμοποιεί τα δεδομένα από γνωστές βιβλιογραφικές βάσεις όπως οι Scopus, Crossref, Open Alex και Semantic Scholar. Ο κύριος στόχος είναι να παρέχει στον χρήστη ένα εύχρηστο περιβάλλον χρήσης για ταξινόμηση, φιλτράρισμα και εξαγωγή δεδομένων. Επιπλέον, οι χρήστες μπορούν να περιηγηθούν άνετα στις λειτουργίες της εφαρμογής, χάρη στο ανταποκρινόμενο σχεδιασμό που εξυπηρετεί τόσο τους υπολογιστές όσο και τα προγράμματα περιήγησης κινητών. Η εφαρμογή εκμεταλλεύεται την αρχιτεκτονική των συστατικών της React, επιτρέποντας να διαχωρίζονται λειτουργίες του κώδικα σε επαναχρησιμοποιήσιμα στοιχεία τα οποία βοηθούν στην καλύτερη συντήρηση και επεκτασιμότητα της εφαρμογής. Ακόμη, η εφαρμογή περιλαμβάνει την δυνατότητα προβολής ιστορικού που επιτρέπει στους χρήστες να επανεξετάζουν και να αποκτούν πρόσβαση σε προηγούμενα αποτελέσματα αναζήτησης με άνεση. Το ιστορικό αναζήτησης αποθηκεύεται τοπικά στον περιηγητή του χρήστη χρησιμοποιώντας τον τοπικό αποθηκευτικό χώρο (Local Storage) και εξασφαλίζοντας έτσι την διατήρηση των δεδομένων ανάμεσα στις συνεδρίες. Παρά το γεγονός ότι δεν αναφέρεται ρητά, ο σχεδιασμός και η έμπνευση της εφαρμογής αντλούνται από τις λειτουργίες και την εμπειρία χρήστη της δημοφιλούς εφαρμογής «Perish or Publish». Αυτή η ενσωμάτωση προσθέτει ένα στοιχείο καινοτομίας και εξοικείωσης για τον χρήστη, ενισχύοντας τη συνολική εμπειρία του. Στη παρούσα εργασία, αρχικά περιγράφονται οι απαιτήσεις και οι περιορισμοί του εφαρμογής. Στη συνέχεια, αναλύονται οι τεχνολογίες και τα εργαλεία που χρησιμοποιήθηκαν. Έπειτα παρουσιάζεται η αρχιτεκτονική και οι βάσεις δεδομένων που χρησιμοποιήθηκαν. Τέλος, παρουσιάζεται η εφαρμογή με παραδείγματα και επεξηγήσεις για τη λειτουργικότητά της.

WEB SEARCH APPLICATION FOR BIBLIOGRAPHIC PUBLICATIONS

STAMATIS DIAMANTIS

Abstract

The subject of this thesis is the development of a web application, which was built using the popular JavaScript React library, developed by Facebook. The application works on the client-side and uses asynchronous AJAX requests with query parameters such as author name and publication name to retrieve relevant bibliographic information. The application uses data from well-known bibliographic bases such as Scopus, Crossref, Open Alex and Semantic Scholar. The main objective is to provide the user with an easy-to-use interface for sorting, filtering and extracting data from the aforementioned databases. Additionally, users can comfortably navigate the app's features thanks to a responsive design that caters to both desktop and mobile browsers. The app takes advantage of React's component architecture, allowing different pieces of code to be split into reusable components that help make the app more maintainable and scalable. Also, the app includes a history view feature that allows users to review and access past search results conveniently. Search history is stored locally in the user's browser using Local Storage, ensuring that data is preserved between sessions. Although not explicitly stated, the app's design and inspiration are drawn from the features and user experience of the popular app 'Perish or Publish'. This integration adds an element of novelty and familiarity for the user, enhancing the overall user experience. In this paper, the requirements and limitations of the application are first described. Then, the technologies and tools used are analyzed. Then the architecture and databases used are presented. Finally, the application is presented with examples and explanations of its functionality.

Ευχαριστίες

Με την ολοκλήρωση της παρούσας Διπλωματικής Εργασίας θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου, κ. Αντώνη Σιδηρόπουλο για την ανάθεση αυτής της διπλωματικής εργασίας όπως επίσης και για την πολύτιμη βοήθεια και καθοδήγηση του. Τέλος, θα ήθελα να ευχαριστήσω την οικογένεια μου και την φίλη μου Νατάσα για την στήριξη που μου παρείχαν κατά τη συγγραφή της διπλωματικής μου εργασίας, καθώς και για την εμπιστοσύνη που μου έδειξαν κατά τη διάρκεια των σπουδών μου.

Περιεχόμενα

Πρόλογος.....	3
Περίληψη.....	4
Abstract	5
Ευχαριστίες	6
Περιεχόμενα	7
Κατάλογος Σχημάτων	10
Κατάλογος Πινάκων.....	10
Κατάλογος Κώδικα	10
Κατάλογος Εικόνων	10
Συνομογραφίες.....	12
Κεφάλαιο 1ο: Εισαγωγή.....	13
1.1 Εισαγωγή.....	13
1.2 Διαδικτυακή εφαρμογή	13
1.3 Σκοπός της εφαρμογής.....	14
1.4 Απαιτήσεις και Περιορισμοί	14
1.5 Παρόμοια Συστήματα.....	14
1.6 Περιπτώσεις χρήσης.....	15
1.7 Επίλογος.....	19
Κεφάλαιο 2ο: Τεχνολογίες και Εργαλεία	20
2.1 Εισαγωγή.....	20
2.2 HTML/CSS	20
2.3 Γλώσσα Προγραμματισμού JavaScript.....	21
2.4 Ajax	22
2.5 Javascript Frameworks	22
2.6 Λόγοι επιλογής React.....	22
2.7 React.js	23
2.7.1 Ιστορία.....	23
2.7.2 Virtual DOM	23
2.7.3 JSX	24
2.7.4 Βασικά Components.....	24
2.7.5 Hooks.....	26

2.7.6	Routes	27
2.7.7	Material UI	27
2.7.8	Axios	28
2.8	Αποθήκευση δεδομένων στην πλευρά του χρήστη	29
2.9	Node Package Manager	29
2.10	React Developer Tools	30
2.11	Visual Code	30
2.12	Postman	30
2.13	Επίλογος	31
Κεφάλαιο 3ο: Βιβλιογραφικές Βάσεις		32
3.1	Εισαγωγή	32
3.2	API Βιβλιογραφιών	32
3.2.1	Crossref	33
3.2.2	Scopus	38
3.2.3	Open Alex	41
3.2.4	Semantic Scholar	44
3.3	Ροή Δεδομένων	45
3.4	Επίλογος	46
Κεφάλαιο 4ο: Παρουσίαση Εφαρμογής		47
4.1	Εισαγωγή	47
4.2	Αρχική οθόνη	47
4.3	Μπάρα επιλογής βάσης αναζήτησης	49
4.4	Πίνακας αποτελεσμάτων αναζήτησης	49
4.5	Πίνακας ιστορικού αναζητήσεων	51
4.6	Πεδία αναζήτησης Crossref	52
4.7	Πεδία αναζήτησης Scopus	53
4.8	Πεδία αναζήτησης Semantic Scholar	53
4.9	Πεδία αναζήτησης Open Alex	54
4.10	Σελίδα πλήρης ιστορικού	55
4.11	Σελίδα Ρυθμίσεων	56
4.12	Επίλογος	57
Κεφάλαιο 5ο: Σύνοψη		58
5.1	Σύνοψη	58
5.2	Προτάσεις Βελτίωσης	58
ΒΙΒΛΙΟΓΡΑΦΙΑ		59

ΠΑΡΑΡΤΗΜΑ Α : Οδηγός Εγκατάστασης React js.....	61
ΠΑΡΑΡΤΗΜΑ Β : Οδηγός Εγκατάστασης Visual Studio Code	62

Κατάλογος Σχημάτων

Σχήμα 2.4.1 Λογότυπο React.js.....	23
Σχήμα 3.2.1 Λογότυπο Crossref.....	33
Σχήμα 3.2.2 Λογότυπο Scopus.....	38
Σχήμα 3.2.3 Λογότυπο Open Alex.....	41
Σχήμα 3.2.4 Λογότυπο Semantic Scholar	44
Σχήμα 3.3.1 Διάγραμμα ροής δεδομένων εφαρμογής.....	46

Κατάλογος Πινάκων

Πίνακας 1.6.1 Περίπτωση χρήσης: Αναζήτηση βιβλιογραφικών αναφορών.....	15
Πίνακας 1.6.2 Περίπτωση χρήσης: Προβολή ιστορικού αναζητήσεων	16
Πίνακας 1.6.3 Περίπτωση χρήσης: Ρύθμιση Scopus API Key	17
Πίνακας 1.6.4 Περίπτωση χρήσης: Ταξινόμηση Δεδομένων Αναζήτησης.....	18
Πίνακας 3.2.1 Τα βασικά API του Crossref.....	34
Πίνακας 3.2.2 Endpoints του Crossref Work API.....	35
Πίνακας 3.2.3 Το Scopus API που χρησιμοποιείται στην εφαρμογή.....	39
Πίνακας 3.2.4 Όρια δεδομένων για τα Scopus Search API	40
Πίνακας 3.2.5 Τα βασικά API που έχει η βάση OpenAlex.....	43

Κατάλογος Κώδικα

Κώδικας 2.1 Παράδειγμα μια απλής ιστοσελίδας HTML με μια επικεφαλίδα και μια παράγραφο.....	20
Κώδικας 2.2 Παράδειγμα ορισμού μια παραγράφου με κόκκινο χρώμα με την χρήση CSS	21
Κώδικας 2.3 Παράδειγμα μια απλής Function σε JavaScript.....	21
Κώδικας 2.4 Παράδειγμα μια σταθερά σε JSX.....	24
Κώδικας 2.5 Παράδειγμα κώδικα που ορίζει ένα λειτουργικό component που ονομάζεται HelloWorld	25
Κώδικας 2.6 Παράδειγμα κώδικα που ορίζει ένα δομικό component που ονομάζεται Counter και μετράει τον αριθμό κλικ του χρήστη	25
Κώδικας 2.7 Παράδειγμα κώδικα που ορίζει το Hook useState για να δηλώσουμε ένα state με το όνομα text, το οποίο ξεκινά με μια κενή τιμή	27
Κώδικας 2.8 Παράδειγμα κώδικα για την δημιουργία ενός κουμπιού με την χρήση της Material UI .	28
Κώδικας 2.9 Παράδειγμα κώδικα για ένα αίτημα GET σε έναν backend server	29
Κώδικας 3.1 Απάντηση αιτήματος στο Crossref Work API.....	36
Κώδικας 3.2 Σύνθετο παράδειγμα API Endpoint του Scopus.....	41

Κατάλογος Εικόνων

Εικόνα 4.1 Αρχική Οθόνη Εφαρμογής από υπολογιστή.....	48
---	----

Εικόνα 4.2 Αρχική Οθόνη Εφαρμογής σε κινητού κάτω μέρος.....	48
Εικόνα 4.3 Αρχική Οθόνη Εφαρμογής σε κινητού πάνω μέρος	48
Εικόνα 4.4 Αρχική Οθόνη Εφαρμογής από υπολογιστή με Light Mode	49
Εικόνα 4.5 Μπάρα επιλογής βάσης αναζήτησης.....	49
Εικόνα 4.6 Πίνακας αποτελεσμάτων αναζήτησης	50
Εικόνα 4.7 Πίνακας ιστορικού αναζητήσεων	51
Εικόνα 4.8 Προβολή αποθηκευμένων δεδομένων στο Local Storage του browser	52
Εικόνα 4.9 Πεδία αναζήτησης του Crossref.....	53
Εικόνα 4.10 Πεδία αναζήτησης του Scopus.....	53
Εικόνα 4.11 Πεδία αναζήτησης του Semantic Scholar	54
Εικόνα 4.12 Πεδία αναζήτησης του OpenAlex.....	55
Εικόνα 4.13 Πίνακας πλήρως ιστορικού αναζητήσεων	56
Εικόνα 4.14 Σελίδα ρυθμίσεων εφαρμογής.....	57
Εικόνα 4.15 Προβολή αποθηκευμένων κλειδιού Scopus στο Local Storage του browser	57

Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
ΔΙΠΙΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε.	Πτυχιακή Εργασία
Π.Χ.	Περίπτωση Χρήσης
PoP	Publish or Perish
HTML	HyperText Markup Language
CSS	Cascading Style Sheet
DOM	Document Object Model
MVC	Model View Controller
JSX	JavaScript XML
UI	User Interface
API	Application Programming Interface
npm	Node Package Manager

Κεφάλαιο 1ο: Εισαγωγή

1.1 Εισαγωγή

Η βιβλιομετρία ή βιβλιομετρική ανάλυση αναφέρεται στη χρήση ποσοτικών τεχνικών έρευνας και στατιστικών μεθόδων για την εξέταση επιστημονικών δημοσιεύσεων. Στόχος της είναι η ανάλυση και η εκτίμηση των διαδικασιών παραγωγής, επικοινωνίας και χρήσης των επιστημονικών πληροφοριών. Η εφαρμογή της εκτείνεται σε όλους τους επιστημονικούς κλάδους με στόχο την αξιολόγηση του αντικτύπου της επιστημονικής έρευνας μέσω ποσοτικής ανάλυσης[1].

Η πρακτική εφαρμογή της βιβλιομετρίας απαιτεί συγκεκριμένα βήματα, όπως [1]:

- 1) Ποσοτικοποίηση: Αυτή αφορά τη συλλογή, περιγραφή, και αποσαφήνιση των δεδομένων με χρήση ποσοτικών τεχνικών.
- 2) Μέτρηση: Απαιτεί τον καθορισμό του τι μπορούμε και πρέπει να μετρήσουμε, καθώς και των πιο αντιπροσωπευτικών μετρήσεων που θα πραγματοποιηθούν.
- 3) Συγκριτική Αξιολόγηση: Τα δεδομένα τίθενται σε πλαίσια προκειμένου να καθοριστούν οι περιορισμοί για τη διενέργεια σύγκρισης.
- 4) Παρουσίαση και Ερμηνεία: Περιλαμβάνει την παρουσίαση και την ερμηνεία των αποτελεσμάτων με την χρήση διαγραμμάτων και χαρτών.

Τα οφέλη που προκύπτουν από μια βιβλιομετρική ανάλυση σχετίζονται κυρίως με [1]:

- 1) Επιστημονικές Πληροφορίες: Σχετικές με το αντικείμενο της επιστημονικής κοινότητας και την πρόσβαση σε αυτές.
- 2) Δομή και Δυναμική της Επιστήμης: Παρακολούθηση των νέων τάσεων, πεδίων έρευνας, και επίκαιρων θεμάτων.
- 3) Έρευνες Αξιολόγησης και Χρηματοδότησης: Προς την κατεύθυνση της ανάπτυξης κατάλληλων μεθόδων μέτρησης, αξιολόγησης και τύπων χρηματοδότησης.

Επιπλέον, είναι σημαντικό να αναφερθεί και η ποσοτική μέθοδος της βιβλιομετρίας που είναι η ανάλυση παραπομπών (Citation Analysis). Αντικείμενο αυτής της μεθόδου είναι η σχέση μεταξύ μιας εργασίας και των άρθρων που έχει παραθέσει, με βάση την υπόθεση ότι τα άρθρα που αναφέρονται μεταξύ τους σχετίζονται με κάποιο τρόπο. Οι σχέσεις που μελετώνται με την ανάλυση παραπομπών περιλαμβάνουν συγγραφείς, χώρες και ιδρύματα [2]. Βασισόμενα πάνω σε αυτή την τεχνική στηρίζεται και η ανάπτυξη της εφαρμογής της παρούσας Δ.Ε. .

1.2 Διαδικτυακή εφαρμογή

Μια διαδικτυακή εφαρμογή είναι ένα λογισμικό που τρέχει σε έναν διακομιστή και είναι προσβάσιμο από οποιαδήποτε συσκευή που έχει πρόσβαση στο Διαδίκτυο, όπως υπολογιστές, κινητά τηλέφωνα ή τάμπλετ. Οι διαδικτυακές εφαρμογές παρέχουν συνήθως λειτουργικότητες ή υπηρεσίες στους χρήστες τους μέσω ενός προγραμματιστικού διεπαφής (API), το οποίο επιτρέπει την αλληλεπίδραση με το λογισμικό από μια απομακρυσμένη τοποθεσία. Οι διαδικτυακές εφαρμογές μπορούν να περιλαμβάνουν ιστοσελίδες, παιχνίδια, πλατφόρμες κοινωνικής δικτύωσης, ηλεκτρονικά καταστήματα, εφαρμογές γραφείου και πολλά άλλα, προσφέροντας μια ευρεία γκάμα υπηρεσιών και λειτουργιών στους χρήστες τους.

1.3 Σκοπός της εφαρμογής

Βασικός σκοπός της εφαρμογής είναι να διασφαλίζει την πρόσβαση του χρήστη στην εφαρμογή από οποιαδήποτε έξυπνη συσκευή, εξασφαλίζοντας την ευελιξία και την ανεξαρτησία από συγκεκριμένα συστήματα λειτουργίας. Αυτό επιτυγχάνεται μέσω της διαθεσιμότητας της εφαρμογής μέσω προγραμμάτων περιήγησης στο διαδίκτυο, όπως περιηγητές ιστού σε φορητές συσκευές, υπολογιστές, κινητά τηλέφωνα κ.λπ.

Επιπλέον, η αξιοπιστία των δεδομένων που παρέχονται είναι κρίσιμη για την εμπιστοσύνη του χρήστη στην εφαρμογή. Προσφέροντας ακριβή, συνεπή και ασφαλή δεδομένα κατά τη μετάδοση και επεξεργασία, εξασφαλίζουμε ότι ο χρήστης μπορεί να βασίζεται πλήρως στην ακεραιότητα των πληροφοριών που λαμβάνει.

Τέλος, η βέλτιστη αξιοποίηση των διαθέσιμων πόρων του συστήματος αποτελεί καίριο παράγοντα για την ομαλή λειτουργία και την ικανοποίηση του χρήστη. Η εφαρμογή οφείλει να λειτουργεί με τρόπο αποδοτικό, αξιοποιώντας στο έπακρο τους διαθέσιμους πόρους, όπως η ισχύς επεξεργασίας, η μνήμη RAM και η χωρητικότητα αποθήκευσης. Η υλοποίηση των προαναφερθέντων οδηγεί σε μία εφαρμογή η οποία λειτουργεί ομαλά, χωρίς καθυστερήσεις ή σφάλματα, προσφέροντας στον χρήστη μία άψογη εμπειρία.

1.4 Απαιτήσεις και Περιορισμοί

Σε μια εφαρμογή, οι απαιτήσεις του λογισμικού καθορίζουν τις δυνατότητες και τους περιορισμούς της. Συνολικά, έχουν ως στόχο τον καθορισμό του σκοπού του προϊόντος και την περιγραφή της προσφοράς του, προσαρμόζοντας πάντα τις λειτουργίες του σε συγκεκριμένες ανάγκες [3]. Λαμβάνοντας υπόψη τα παραπάνω, προκύπτουν οι τρεις βασικές ανάγκες που πρέπει να καλύπτει:

- 1) Προβολή και ευχρηστία: Η εφαρμογή χρειάζεται να είναι Responsive, δηλαδή να μπορεί να προσαρμόζεται αυτόματα τόσο σε μεγάλες οθόνες, όσο και σε μικρές, όπως για παράδειγμα ένα κινητό τηλέφωνο. Με αυτόν τον τρόπο ευνοείτε σημαντικά η εμπειρία του χρήστη.
- 2) Επεκτασιμότητα: Είναι σημαντικό να μπορεί να συντηρείτε εύκολα ώστε να είναι εφικτό να εξελίσσεται και να καλύπτει ανάγκες που μπορεί να προκύψουν στο μέλλον. Έτσι, με αυτό σαν κριτήριο επιλέχθηκε η βιβλιοθήκη της React που θα αναλυθεί στο 2^ο κεφάλαιο.
- 3) Αξιοπιστία: Είναι ζωτικής σημασίας οι πληροφορίες που χρησιμοποιεί η εφαρμογή να είναι αξιόπιστες και να έχουν επαληθευτεί, προσφέροντας έτσι στον χρήστη βεβαιότητα. Η συλλογή δεδομένων από αξιόπιστες πηγές και η επαλήθευσή τους διασφαλίζουν την ακρίβεια και την αξιοπιστία των αναλύσεων. Επιπλέον, η συνεχής ενημέρωση των δεδομένων διατηρεί την αξιοπιστία - εγκυρότητα της εφαρμογής για σωστή ερμηνεία και χρήση των αποτελεσμάτων.

Ως περιορισμούς της εφαρμογής, πρέπει να αναφέρουμε ότι η ύπαρξη σύνδεσης στο διαδίκτυο αποτελεί απαραίτητη προϋπόθεση για τη σωστή λειτουργία της. Τέλος, σε περίπτωση που κάποια βάση δεδομένων βρίσκεται προσωρινά εκτός λειτουργίας, η πρόσβαση στα δεδομένα της εφαρμογής θα είναι περιορισμένη.

1.5 Παρόμοια Συστήματα

Στη σημερινή εποχή, η αξιολόγηση της επιστημονικής επίδοσης είναι κρίσιμη για τους ερευνητές, τα εκπαιδευτικά ιδρύματα και τους χρηματοδότες. Για αυτό το σκοπό υπάρχουν διάφορα εργαλεία που

μπορούν να βοηθήσουν. Η παρούσα εφαρμογή επηρεάζεται σημαντικά από το "Publish or Perish" (PoP), ένα εργαλείο που έχει αναδειχθεί ως ένα από τα πιο σημαντικά σε αυτό το πλαίσιο. Το «PoP» παρέχει βιβλιομετρικούς δείκτες όπως το h-index και ο αριθμός παραπομπών. Αυτό επιτρέπει στους ερευνητές να αξιολογήσουν την επίδοσή τους, συγκρίνοντάς την με άλλους και αναδεικνύοντας τη σημασία της έρευνάς τους. Μέσω αυτού του εργαλείου, επιτυγχάνεται η παρακολούθηση των τάσεων και η κατανόηση της επίδρασης στον επιστημονικό χώρο, αναδεικνύοντας τον ρόλο της δημοσιευτικής δραστηριότητας στην ακαδημαϊκή κοινότητα [4]. Τέλος, αξίζει να σημειωθεί ότι το «PoP» είναι σχεδιασμένο για χρήση σε υπολογιστή και απαιτείται να γίνει εγκατάσταση πριν την χρήση του, αλλά και να υπάρχει πρόσβαση στο διαδίκτυο για τη συλλογή δεδομένων.

1.6 Περιπτώσεις χρήσης

Οι περιπτώσεις χρήσης μιας εφαρμογής αποτελούν ουσιαστικό στοιχείο στη διαδικασία ανάπτυξης και λειτουργίας της. Καθορίζουν τη σχέση μεταξύ του προγράμματος και του τελικού χρήστη, καθώς και τον τρόπο με τον οποίο οι χρήστες θα επωφεληθούν από αυτό. Με την ανάλυση των περιπτώσεων χρήσης, μπορούμε να κατανοήσουμε καλύτερα τις ανάγκες, τις προτιμήσεις και τις απαιτήσεις των χρηστών και να σχεδιάσουμε την εφαρμογή, έτσι ώστε να παρέχει την καλύτερη δυνατή εμπειρία. Επιπλέον, στις περιπτώσεις χρήσης φαίνεται η κύρια ροή γεγονότων, αλλά και οι εναλλακτικές ροές, που επιτρέπουν την ανταπόκριση σε διαφορετικά σενάρια και καταστάσεις, που μπορεί να προκύψουν κατά τη χρήση της εφαρμογής.

Πίνακας 1.6.1 Περίπτωση χρήσης: Αναζήτηση βιβλιογραφικών αναφορών

Περιπτώσεις χρήσης	Αναζήτηση βιβλιογραφικών αναφορών
Χρήστης	Ανώνυμος χρήστης
Προϋποθέσεις	1. Σύνδεση στο διαδίκτυο από πρόγραμμα περιήγησης
Κύρια ροή γεγονότων	<ol style="list-style-type: none"> 1. Ο ανώνυμος χρήστης ανοίγει την κύρια σελίδα 2. Ο χρήστης διαλέγει την βιβλιογραφική που θέλει να κάνει αναζήτηση 3. Ο χρήστης συμπληρώνει λέξεις-κλειδιά στα πεδία αναζήτησης 4. Ο χρήστης πατάει το κουμπί «Search» 5. Το σύστημα στέλνει αίτημα στην επιλεγμένη βάση και λαμβάνει τα δεδομένα 6. Το σύστημα αποθηκεύει τα δεδομένα στο Local Storage του περιηγητή του χρήστη 7. Το σύστημα εμφανίζει τα δεδομένα στον πίνακα αποτελεσμάτων 8. Το σύστημα εμφανίζει εγγραφή στον πίνακα ιστορικού με τις λέξεις-κλειδιά και την βάση δεδομένων αναζήτησης

Εναλλακτικές ροές γεγονότων	<ol style="list-style-type: none"> 1. Στο βήμα 3, το σύστημα διαπιστώνει ότι ο χρήστης δεν έχει συμπληρώσει κανένα πεδίο για αναζήτηση. <ol style="list-style-type: none"> a) Το σύστημα ενημερώνει τον χρήστη με μήνυμα «You must fill in at least one of the search fields» b) Ο χρήστης συμπληρώνει τα πεδία c) Επαναφορά στο βήμα 3 2. Στο βήμα 4, το σύστημα διαπιστώνει ότι η βάση δεδομένων δεν είναι διαθέσιμη <ol style="list-style-type: none"> a) Το σύστημα ενημερώνει τον χρήστη με μήνυμα «There was an error communicating with the database, please try another database or try again later» b) Επαναφορά στο βήμα 3 3. Στο βήμα 4, δεν βρέθηκαν δεδομένα που να αντιστοιχούν στις συγκεκριμένες λέξεις-κλειδιά <ol style="list-style-type: none"> a) Το σύστημα ενημερώνει τον χρήστη με μήνυμα στον πίνακα αποτελεσμάτων «No rows» b) Επαναφορά στο βήμα 3 4. Στο βήμα 2, ο χρήστης διάλεξε την βάση δεδομένων του Scopus χωρίς να έχει καταχωρήσει κλειδί στην σελίδα των ρυθμίσεων <ol style="list-style-type: none"> a) Το σύστημα ενημερώνει τον χρήστη με μήνυμα «Please enter your Scopus API key in the settings page to use this search” b) Επαναφορά στο βήμα 3 5. Στο βήμα 7, ο χρήστης πατάει το βελάκι του Pager για την επόμενη σελίδα <ol style="list-style-type: none"> a) Το σύστημα ενημερώνει τον πίνακα με τα δεδομένα της δεύτερης σελίδας b) Επαναφορά στο βήμα 7
-----------------------------	---

Πίνακας 1.6.2 Περίπτωση χρήσης: Προβολή ιστορικού αναζητήσεων

Περιπτώσεις χρήσης	Προβολή ιστορικού αναζητήσεων
Χρήστης	Ανώνυμος χρήστης
Προϋποθέσεις	<ol style="list-style-type: none"> 1. Σύνδεση στο διαδίκτυο από πρόγραμμα περιήγησης 2. Ο χρήστης πρέπει να έχει πραγματοποιήσει προηγούμενη αναζήτηση
Κύρια ροή γεγονότων	<ol style="list-style-type: none"> 1. Ο ανώνυμος χρήστης πατάει την καρτέλα «History» από το κεντρικό μενού 2. Το σύστημα ανακατευθύνει τον χρήστη στην σελίδα του ιστορικού 3. Το σύστημα εμφανίζει ένα πίνακα που περιέχει τις προηγούμενες αναζητήσεις του χρήστη με τα στοιχεία (Όρος αναζήτησης, Βάση αναζήτησης, Αριθμός αποτελεσμάτων, Ημερομηνία) και δυο επιλογές για εξαγωγή του πίνακα και καθαρισμό ιστορικού

Εναλλακτικές ροές γεγονότων	<ol style="list-style-type: none"> 1. Στο βήμα 3, το σύστημα διαπιστώνει ότι ο χρήστης δεν έχει συμπληρώσει κανένα πεδίο για αναζήτηση. <ol style="list-style-type: none"> a) Το σύστημα ενημερώνει τον χρήστη με μήνυμα «You must fill in at least one of the search fields» b) Ο χρήστης συμπληρώνει τα πεδία c) Επαναφορά στο βήμα 3 2. Στο βήμα 3, ο χρήστης πατάει ο κουμπί «Clear» <ol style="list-style-type: none"> a) Το σύστημα καθαρίζει το ιστορικό του χρήστη και αδειάζει τον πίνακα ιστορικού b) Επαναφορά στο βήμα 3 3. Στο βήμα 3, ο χρήστης θέλει να εξαγάγει τα δεδομένα του ιστορικού του <ol style="list-style-type: none"> a) Ο χρήστης πατάει το κουμπί «EXPORT» πάνω από τον πίνακα ιστορικού b) Στο υπό μενού που εμφανίζει ο χρήστης πατάει την επιλογή «Download as CSV» c) Το σύστημα προετοιμάζει τα δεδομένα για εξαγωγή d) Το σύστημα φορτώνει το αρχείο δεδομένων και το κατεβάζει στο περιηγητή του χρήστη e) Επαναφορά στο βήμα 3
-----------------------------	---

Πίνακας 1.6.3 Περίπτωση χρήσης: Ρύθμιση Scopus API Key

Περιπτώσεις χρήσης	Ρύθμιση Scopus API Key
Χρήστης	Ανώνυμος χρήστης
Προϋποθέσεις	<ol style="list-style-type: none"> 1. Σύνδεση στο διαδίκτυο από πρόγραμμα περιήγησης 2. Δημιουργία λογαριασμού και λήψη κλειδιού από την ιστοσελίδα του elsevier.com
Κύρια ροή γεγονότων	<ol style="list-style-type: none"> 1. Ο ανώνυμος χρήστης πατάει την καρτέλα «Settings» από το κεντρικό μενού 2. Το σύστημα ανακατευθύνει τον χρήστη στην σελίδα των ρυθμίσεων 3. Το σύστημα εμφανίζει ένα πεδίο για την εισαγωγή του Scopus API Key και ένα κουμπί αποθήκευσης 4. Ο χρήστης εισάγει το κλειδί του στο πεδίο 5. Ο χρήστης πατάει το κουμπί «Save» 6. Το σύστημα αποθηκεύει το κλειδί στο Local Storage

Εναλλακτικές ροές γεγονότων	<ol style="list-style-type: none"> 1. Στο βήμα 4, ο χρήστης έχει ήδη εισάγει κάποιο κλειδί <ol style="list-style-type: none"> a) Ο χρήστης αδειάζει το πεδίο του Scopus Key b) Ο χρήστης πατάει το κουμπί «Save» c) Το σύστημα διαγράφει το κλειδί του χρήστη από το Local Storage d) Επιστροφή στο βήμα 3
Μετά-συνθήκες	Ο χρήστης μπορεί να πραγματοποιήσει αναζητήσεις στην βάση δεδομένων του Scopus

Περιπτώσεις χρήσης	Εναλλαγή Εφαρμογής σε Light Mode
Χρήστης	Ανώνυμος χρήστης
Προϋποθέσεις	<ol style="list-style-type: none"> 1. Σύνδεση στο διαδίκτυο από πρόγραμμα περιήγησης
Κύρια ροή γεγονότων	<ol style="list-style-type: none"> 1. Ο χρήστης ανοίγει την κύρια σελίδα 2. Ο χρήστης πατάει στο κουμπί εναλλαγής του θέματος της σελίδας 3. Το σύστημα αλλάζει το θέμα της σελίδας
Εναλλακτικές ροές γεγονότων	<ol style="list-style-type: none"> 1. Στο βήμα 2, ο χρήστης δεν πατάει το κουμπί. <ol style="list-style-type: none"> a) Επιστροφή στο βήμα 1

Πίνακας 1.6.4 Περίπτωση χρήσης: Ταξινόμηση Δεδομένων Αναζήτησης

Περιπτώσεις χρήσης	Ταξινόμηση Δεδομένων Αναζήτησης
Χρήστης	Ανώνυμος χρήστης
Προϋποθέσεις	<ol style="list-style-type: none"> 1. Π.Χ. 01 Αναζήτηση βιβλιογραφικών αναφορών
Κύρια ροή γεγονότων	<ol style="list-style-type: none"> 1. Ο χρήστης πατάει το κουμπί επιλογών σε μια από τις στήλες του πίνακα αναζητήσεων που θέλει να ταξινομήσει 2. Ο χρήστης στο υπό μενού που εμφανίζεται διαλέγει την ταξινόμηση που θέλει να κάνει «Sort by ASC» ή «Sort by DES» 3. Το σύστημα ταξινομεί τον πίνακα με βάση την επιλογή του χρήστη
Εναλλακτικές ροές γεγονότων	<ol style="list-style-type: none"> 1. Στο βήμα 2, ο χρήστης πατάει την επιλογή «Unsort» <ol style="list-style-type: none"> a) Το σύστημα επαναφέρει την αρχική ταξινόμηση που είχε ο πίνακας b) Επιστροφή στο βήμα 1 2. Στο βήμα 2, ο χρήστης πατάει την επιλογή «Filter» <ol style="list-style-type: none"> a) Το σύστημα εμφανίζει ένα μενού με τρία πεδία. b) Ο χρήστης στο πρώτο πεδίο διαλέγει την στήλη που θέλει να φιλτράρει

	<ul style="list-style-type: none"> c) Ο χρήστης στο δεύτερο πεδίο διαλέγει την πράξη που θέλει να γίνει d) Ο χρήστης στο τρίτο πεδίο διαλέγει την τιμή που θέλει να φιλτράρει e) Το σύστημα φιλτράρει τον πίνακα δεδομένων με τα στοιχεία που έδωσε ο χρήστης f) Επιστροφή στο βήμα 1 <p>3. Στο βήμα 2, ο χρήστης πατάει την επιλογή «Hide»</p> <ul style="list-style-type: none"> a) Το σύστημα κρύβει την συγκεκριμένη στήλη b) Επιστροφή στο βήμα 1 <p>4. Στο βήμα 2, ο χρήστης πατάει την επιλογή «Show columns»</p> <ul style="list-style-type: none"> a) Το σύστημα εμφανίζει ένα μενού με όλες τις στήλες b) Ο χρήστης επιλέγει ποιες στήλες θέλει να φαίνονται c) Το σύστημα κρύβει τις στήλες που επέλεξε ο χρήστης d) Επιστροφή στο βήμα 1 <p>5. Στο βήμα 1, ο χρήστης πατάει το βελάκι που βρίσκεται στην στήλη του πίνακα</p> <ul style="list-style-type: none"> a) Το σύστημα αλλάζει την τρέχουσα ταξινόμηση b) Επιστροφή στο βήμα 1
--	--

1.7 Επίλογος

Στο κεφάλαιο αυτό αναλύθηκε ο ορισμός την βιβλιομετρίας μαζί με τις εφαρμογές και τα οφέλη της. Ακόμα, έγινε αναφορά στην μέθοδο της ανάλυσης παραπομπών όπου βασίζεται και η εφαρμογή της παρούσας Δ.Ε. . Στην συνέχεια αναφέραμε ένα από τα βασικότερα εργαλεία για την αξιολόγηση της επιστημονικής επίδοσης των ερευνητών, το «Publish or Perish» μαζί με τις χρήσεις του. Τέλος, παρουσιάστηκαν κάποιες από τις απαιτήσεις και τους περιορισμούς της εφαρμογής αλλά και οι περιπτώσεις χρήσης της. Στο επόμενο κεφάλαιο παρουσιάζονται και αναλύονται οι τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής.

Κεφάλαιο 2ο: Τεχνολογίες και Εργαλεία

2.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα γίνει παρουσίαση και αναφορά σε όλες τις τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής. Η εφαρμογή έχει αναπτυχθεί στην γλώσσα προγραμματισμού JavaScript. Το κύριο εργαλείο όμως είναι η βιβλιοθήκη React της JavaScript που δίνει πολλές δυνατότητες για την δημιουργία επαναχρησιμοποιήσιμου κώδικα [5]. Η εφαρμογή βρίσκεται εξ ολοκλήρου στην μεριά του client (Frontend), οπότε δεν υπάρχει κάποιο είδος βάσης ή Backend. Για την δημιουργία και την επεξεργασία του κώδικα, χρησιμοποιήθηκε το πρόγραμμα Visual Studio Code.

2.2 HTML/CSS

Η HTML (HyperText Markup Language) δημιουργήθηκε από τον Tim Berners-Lee το 1991 και αποτελεί τη βάση για τον σχεδιασμό ιστοσελίδων στο διαδίκτυο, χρησιμοποιώντας μια γλώσσα σημείωσης. Η HTML χρησιμοποιεί "ετικέτες" ή "tags" για να καθορίσει τη δομή του κειμένου και των στοιχείων σε μια ιστοσελίδα, όπως κείμενο, εικόνες, συνδέσμους, πίνακες και πολλά άλλα. Οι ιστοσελίδες που βασίζονται σε HTML είναι διαδικτυακές σελίδες που μπορούν να προβληθούν με τη χρήση προγραμμάτων περιήγησης (όπως ο Chrome, ο Firefox, ο Safari κ.λπ.). Η HTML συνήθως συνδυάζεται με άλλες τεχνολογίες όπως η CSS (Cascading Style Sheets) για τον σχεδιασμό και η JavaScript για τη δυναμική αλληλεπίδραση [6].

Κώδικας 2.1 Παράδειγμα μια απλής ιστοσελίδας HTML με μια επικεφαλίδα και μια παράγραφο

```
<!DOCTYPE html>
<html>
  <head>
    <title>Τίτλος Ιστοσελίδας</title>
  </head>
  <body>
    <h1>Η Επικεφαλίδα μου </h1>
    <p>Η παράγραφος μου<p>
  </body>
</html>
```

Παράλληλα με την HTML χρησιμοποιείται και η CSS για την εικαστική μορφοποίηση των ιστοσελίδων. Η CSS ή Cascading Style Sheet, είναι μια γλώσσα σχεδιασμού με κύριο ρόλο τον καθορισμό της εμφάνισης και της μορφοποίησης των στοιχείων που υπάρχουν σε μια HTML σελίδα. Με άλλα λόγια, ενώ η HTML περιγράφει τη δομή και το περιεχόμενο της σελίδας, η CSS καθορίζει το πως αυτή η σελίδα θα παρουσιαστεί στον χρήστη. Χρησιμοποιώντας τη CSS, μπορείτε να καθορίσετε

τα χρώματα, τις γραμματοσειρές, τα περιθώρια, τις διαστάσεις, τις διακεκομμένες γραμμές και άλλες στυλιστικές πτυχές των στοιχείων στην ιστοσελίδα [6]. Επίσης, η CSS υποστηρίζει την έννοια του "κληρονομικού στυλ" (inheritance), καθιστώντας ευκολότερη την εφαρμογή ενιαίων στυλ σε ολόκληρη τη σελίδα [7]. Τέλος, τα CSS αρχεία μπορούν να είναι είτε ενσωματωμένα απευθείας σε μια HTML σελίδα, είτε να είναι εξωτερικά, με τη χρήση ενός ξεχωριστού αρχείου CSS. Η χρήση της CSS βοηθάει στη διαχείριση και στη συντήρηση του σχεδιασμού μιας ιστοσελίδας, καθώς ορίζει τα στυλ κεντρικά και τα εφαρμόζει σε όλα τα σχετικά στοιχεία [6].

Κώδικας 2.2 Παράδειγμα ορισμού μια παραγράφου με κόκκινο χρώμα με την χρήση CSS

```
p {
    color: red;
}
```

2.3 Γλώσσα Προγραμματισμού JavaScript

Η JavaScript είναι μια γλώσσα προγραμματισμού που δημιουργήθηκε για την ενίσχυση της διαδραστικότητας του περιηγητή στον κόσμο του διαδικτύου. Από τη δημιουργία της στα μέσα της δεκαετίας του '90, έχει υποστεί σημαντικές αλλαγές και σήμερα αποτελεί ένα από τα πλέον κυρίαρχα εργαλεία στον τομέα της διαδικτυακής ανάπτυξης [6].

Η JavaScript δημιουργήθηκε από τον Brendan Eich, όταν εργαζόταν στην Netscape Communications Corporation. Το 1995, η Netscape κυκλοφόρησε την πρώτη έκδοση του προγράμματος περιήγησης Netscape Navigator 2.0, η οποία ενσωμάτωσε τη JavaScript. Η γλώσσα δημιουργήθηκε αρχικά με το όνομα "Mocha", αλλά αργότερα μετονομάστηκε σε "Livescript" και τελικά, σε "JavaScript" για λόγους μάρκετινγκ, λόγω της δημοτικότητας που είχε η Java εκείνη την εποχή [8].

Στα πρώτα χρόνια, η JavaScript χρησιμοποιούνταν κυρίως για την προσθήκη διαδραστικότητας σε ιστοσελίδες, όπως για παράδειγμα τα γραφικά rollover φωτογραφιών, καθώς ο χρήστης άφηνε το ποντίκι να αιωρηθεί πάνω από μια φωτογραφία και γινόταν αντικατάσταση με μια άλλη. Πρόκειται για μια γλώσσα προγραμματισμού που εκτελείται στον πελάτη (client-side) και επιτρέπει τη δημιουργία δυναμικών ιστοσελίδων με αλληλεπίδραση από τον χρήστη [6].

Σήμερα, η JavaScript έχει εξελιχθεί σε μια πλήρη γλώσσα προγραμματισμού που χρησιμοποιείται σε διάφορα πεδία. Εκτός από τον κόσμο του διαδικτύου, χρησιμοποιείται εκτενώς και για την ανάπτυξη server-side εφαρμογών με τη χρήση πλατφορμών όπως το Node.js. Επίσης, η JavaScript χρησιμοποιείται στην ανάπτυξη mobile εφαρμογών με τη χρήση των frameworks όπως το React Native.

Κώδικας 2.3 Παράδειγμα μια απλής Function σε JavaScript

```
function showAlert() {
    alert("Hello, this is a simple JavaScript function!");
}
```

2.4 Ajax

Το Ajax (Asynchronous JavaScript and XML) είναι μια τεχνική που χρησιμοποιείται στην ανάπτυξη ιστοσελίδων για να επιτρέψει την ασύγχρονη ανταλλαγή δεδομένων μεταξύ του περιηγητή του χρήστη και του διακομιστή. Χρησιμοποιώντας Ajax, μια ιστοσελίδα μπορεί να ανανεώνει δυναμικά περιεχόμενο χωρίς την ανάγκη για ανανέωση της πλήρους σελίδας. Αυτό δίνει τη δυνατότητα στους προγραμματιστές να δημιουργήσουν εφαρμογές που ανταποκρίνονται γρήγορα και δυναμικά στις ενέργειες του χρήστη, χωρίς να τους αναγκάζουν να περιμένουν για τη φόρτωση νέας σελίδας. Με τη χρήση Ajax, οι εφαρμογές μπορούν επίσης να ανταλλάσσουν δεδομένα με το διακομιστή σε φόρμα JSON ή XML, επιτρέποντας την ανάπτυξη πλούσιων και διαδραστικών διαδικτυακών εφαρμογών [6].

2.5 Javascript Frameworks

Τα JavaScript frameworks αποτελούν ισχυρά εργαλεία για την ανάπτυξη δυναμικών και ευέλικτων web εφαρμογών. Κάθε ένα από αυτά τα frameworks έχει τα δικά του χαρακτηριστικά και πλεονεκτήματα, τα οποία το καθιστούν κατάλληλο για συγκεκριμένες ανάγκες και προτιμήσεις ανάπτυξης. Μερικά από τα πιο δημοφιλή JavaScript frameworks περιλαμβάνουν το Angular, το Vue.js και η React.

Το Angular είναι ένα πλήρες MVC (Model-View-Controller) framework της Google, το οποίο προσφέρει ένα ολοκληρωμένο σετ λειτουργιών για την ανάπτυξη μοντέρνων ενισχυμένων εφαρμογών. Ένα από τα κύρια πλεονεκτήματα του Angular είναι η δομή του, η οποία είναι πολύ συμπαγής και περιλαμβάνει προεπιλεγμένες λύσεις για πολλά προβλήματα ανάπτυξης. Αυτό καθιστά το Angular ιδανική επιλογή για προγραμματιστές που αναζητούν μια ολοκληρωμένη και έτοιμη λύση για την ανάπτυξη εφαρμογών μεγάλης κλίμακας[9], [10].

Το Vue.js είναι ένα ελαφρύ, ευέλικτο και εύκολο στην εκμάθηση framework, το οποίο προσφέρει μια απλή σύνταξη και μια ευέλικτη αρχιτεκτονική για την ανάπτυξη επικείμενων UIs. Η ελαφρότητα του Vue.js το καθιστά κατάλληλο για μικρές ή μεσαίες εφαρμογές, ενώ η εύκολη εκμάθηση του το καθιστά επιλογή προτίμησης για αρχάριους προγραμματιστές. Επιπλέον, οι δυνατότητες του Vue.js για επαναχρησιμοποίηση και συντήρηση κώδικα είναι αξιοσημείωτες, καθιστώντας το ιδανική επιλογή για έργα που απαιτούν ταχύτητα ανάπτυξης και συντήρησης[9], [11].

Τέλος, η React είναι επίσης μια από τις πιο δημοφιλείς JavaScript βιβλιοθήκες, γνωστή για την απλότητα της χρήσης του JSX συντακτικού, της αποδοτικής ανανέωσης του DOM με τη χρήση του Virtual DOM και της ενσωμάτωσης της σε μια ευέλικτη αρχιτεκτονική, που επιτρέπει την εύκολη δημιουργία επαναχρησιμοποιήσιμων και συντηρήσιμων επικεφαλίδων. Επιπλέον, η έντονη κοινότητα της React προσφέρει μεγάλο αριθμό πόρων, βιβλιοθηκών και εργαλείων που επιτρέπουν την ανάπτυξη πολύπλοκων εφαρμογών με υψηλή απόδοση και ποιότητα [12]. Καθώς κάθε ένα από αυτά τα frameworks έχει τα δικά του πλεονεκτήματα και χαρακτηριστικά, η επιλογή του κατάλληλου εξαρτάται από τις απαιτήσεις και τις ανάγκες του συγκεκριμένου έργου.

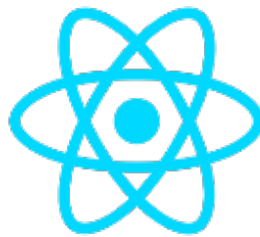
2.6 Λόγοι επιλογής React

Για την ανάπτυξη της παρούσας εφαρμογής έχει επιλεγεί η βιβλιοθήκη της React. Η React αποτελεί μια δημοφιλή επιλογή για την κατασκευή διαδικτυακών εφαρμογών, προσφέροντας πληθώρα πλεονεκτημάτων. Είναι μια βιβλιοθήκη ανοιχτού κώδικα που μπορεί να χρησιμοποιηθεί για τη δημιουργία εφαρμογών σε διάφορες πλατφόρμες, όπως οι ιστοσελίδες, το Android και το iOS. Η React

είναι βασισμένη στη JavaScript, μια ευέλικτη γλώσσα προγραμματισμού που είναι εύκολη στην ανάγνωση.

Η React υιοθετεί μια εξειδικευμένη αρχιτεκτονική που ενισχύει την απόδοση των εφαρμογών. Είναι ιδανική για τη δημιουργία δυναμικών ιστοσελίδων και εφαρμογών, παρέχοντας εργαλεία και λειτουργίες που διευκολύνουν τη δημιουργία δυναμικού περιεχομένου. Επιπλέον, η React ανταποκρίνεται αποτελεσματικά σε σύνθετα δεδομένα, ενώ παραμένει ευέλικτη και προσαρμόσιμη για την ανάπτυξη εφαρμογών με διάφορα στυλ και απαιτήσεις. Συνολικά, η React αποτελεί μια ισχυρή βιβλιοθήκη που προσφέρει πλούσια χαρακτηριστικά για την ανάπτυξη διαδικτυακών εφαρμογών. Στο παράρτημα Α παρουσιάζεται ο οδηγός της εγκατάστασης της React.

2.7 React.js



Σχήμα 2.4.1 Λογότυπο React.js

2.7.1 Ιστορία

Η React είναι μια πολύ γνωστή βιβλιοθήκη της JavaScript ανοιχτού κώδικα που δημιουργήθηκε από την Meta (πρώην Facebook) [13]. Η React επιτρέπει στους προγραμματιστές να δημιουργούν μεγάλες, διαδικτυακές εφαρμογές που μεταβάλλουν δεδομένα χωρίς να απαιτούνται ανανεώσεις της σελίδας και χρησιμοποιείται ως η "όψη" (View) στο πρότυπο αρχιτεκτονικής Model-View-Controller (MVC). Η React αφαιρεί το Document Object Model (DOM), δίνοντας στους προγραμματιστές μια πιο απλή και εύκολη εμπειρία ανάπτυξης κώδικα. Η React κατά κύριο λόγο γίνεται render στην πλευρά του server χρησιμοποιώντας το Node.js. Επιπλέον, παρέχει υποστήριξη για native εφαρμογές για κινητά χρησιμοποιώντας την React Native [12].

Η βασική αρχή πίσω από την React είναι η έννοια του virtual DOM. Η React αξιοποιεί αποτελεσματικά το Virtual DOM, το οποίο μπορεί να εμφανιστεί είτε από την πλευρά του χρήστη (client) είτε από την πλευρά του διακομιστή (server) και να αλληλεπιδρά μεταξύ τους. Το virtual DOM δημιουργεί ένα υποσύνολο κόμβων με βάση τις αλλαγές κατάστασης και πραγματοποιεί όσο το δυνατόν λιγότερες αλλαγές στο DOM, ώστε να διατηρήσει το component ενημερωμένο χωρίς περιττές ανανεώσεις [14].

2.7.2 Virtual DOM

Το Virtual DOM είναι μια δομή δεδομένων που χρησιμοποιείται από η React για να βελτιστοποιήσει την απόδοση των δυναμικών εφαρμογών. Το πραγματικό DOM (Document Object Model) είναι μια λεπτομερής προβολή HTML της ιστοσελίδας που έχει φορτώσει το πρόγραμμα περιήγησης και

παρουσιάζει την σελίδα ως κόμβους και αντικείμενα. Με αυτό τον τρόπο οι γλώσσες προγραμματισμού μπορούν να αλληλοεπιδρούν με την σελίδα [15]. Αντίθετα, το Virtual DOM είναι μια ελαφριά αντιγραφή του πραγματικού DOM. Όταν αλλάζει κάτι στην εφαρμογή, η React συγκρίνει τον Virtual DOM με τον πραγματικό DOM και εντοπίζει μόνο τις διαφορές. Στη συνέχεια, ενημερώνει μόνο τα μέρη του πραγματικού DOM που έχουν αλλάξει, αλλάζοντας την εμφάνιση της εφαρμογής χωρίς να χρειάζεται να δημιουργηθεί ξανά ολόκληρο το πραγματικό DOM. Αυτό καθιστά την React πιο αποδοτική από άλλα frameworks της JavaScript που δεν χρησιμοποιούν Virtual DOM [12].

2.7.3 JSX

Η JSX (JavaScript XML) είναι μια σύνταξη της JavaScript και επιτρέπει την ανάπτυξη κώδικα παρόμοια με την HTML μέσα στην JavaScript. Η κύρια χρήση της είναι η δημιουργία στοιχείων διεπαφής χρήστη (UI). Η JSX επεκτείνει τη JavaScript για την ανάπτυξη στοιχείων React με την ίδια ευκολία που χρησιμοποιείται για τη δημιουργία σελίδων HTML. Παρόλο που η JSX δεν είναι απαραίτητη για τη δημιουργία React στοιχείων, αποτελεί ένα σημαντικό κομμάτι της, καθιστώντας τη γλώσσα πιο ευανάγνωστη και ευέλικτη [13].

Κώδικας 2.4 Παράδειγμα μια σταθερά σε JSX

```
const element = <h1>Γεια σου, JSX!</h1>;
```

2.7.4 Βασικά Components

Στην React, τα components αποτελούν τα βασικά δομικά στοιχεία των εφαρμογών React. Είναι ανακυκλώσιμα κομμάτια κώδικα που καθορίζουν τη δομή και τη συμπεριφορά των στοιχείων UI. Τα components ενθυλακώνουν λογική, στίλ και δεδομένα, καθιστώντας τα εύκολα στη συντήρηση και την επαναχρησιμοποίηση. Τα React components είναι δηλωτικά, πράγμα που σημαίνει ότι περιγράφουν πώς πρέπει να φαίνεται το UI, αντί για το πώς να το επιτύχουμε. Αυτό τα καθιστά ευκολότερη την χρήση τους, επιτρέποντας ταυτόχρονα στη React να βελτιστοποιήσει το UI για την απόδοση [16].

Υπάρχουν δύο κύριοι τύποι React components: λειτουργικά και δομικά components. Τα λειτουργικά (functional) components είναι μη καταστατικά (stateless) και καθαρά, που σημαίνει ότι δεν έχουν καμία εσωτερική κατάσταση και επιστρέφουν πάντα το ίδιο αποτέλεσμα για την ίδια είσοδο. Τα δομικά (class) components, από την άλλη, μπορούν να έχουν και κατάσταση και μεθόδους lifecycle, επιτρέποντάς τους να ανταποκρίνονται σε αλλαγές στην κατάσταση της εφαρμογής.

2.7.4.1 Λειτουργικά Components

Τα λειτουργικά components είναι ο πιο απλός και κοινός τύπος React component. Ορίζονται χρησιμοποιώντας τη λέξη-κλειδί function και επιστρέφουν κώδικα JSX [16]. Για παράδειγμα, ο παρακάτω κώδικας ορίζει ένα λειτουργικό component που ονομάζεται HelloWorld.

Κώδικας 2.5 Παράδειγμα κώδικα που ορίζει ένα λειτουργικό component που ονομάζεται HelloWorld

```
function HelloWorld() {  
  return <h1>Hello World!</h1>;  
}
```

2.7.4.2 Δομικά Components

Τα δομικά components είναι πιο περίπλοκα από τα λειτουργικά components, αλλά προσφέρουν μεγαλύτερη ευελιξία. Ορίζονται χρησιμοποιώντας τη λέξη-κλειδί class και μπορούν να έχουν και κατάσταση (state) και μεθόδους ζωής (lifecycle). Η κατάσταση είναι τα δεδομένα που διατηρεί ένα component και οι μέθοδοι ζωής είναι συναρτήσεις που καλούνται σε ορισμένα σημεία στη διάρκεια ζωής του component, όπως όταν το component δημιουργείται για πρώτη φορά ή όταν τα δεδομένα του αλλάζουν [16]. Για παράδειγμα, ο ακόλουθος κώδικας ορίζει ένα δομικό component που ονομάζεται Counter και μετράει τον αριθμό των φορών που έχει κάνει κλικ ο χρήστης.

Κώδικας 2.6 Παράδειγμα κώδικα που ορίζει ένα δομικό component που ονομάζεται Counter και μετράει τον αριθμό κλικ του χρήστη

```
class Counter extends React.Component {  
  constructor(props) {  
    super(props);  
    this.state = { count: 0 };  
  }  
  
  handleClick() {  
    this.setState((prevState) => ({ count: prevState.count + 1 }));  
  }  
  
  render() {  
    return (  
      <div>  
        <h1>Count: {this.state.count}</h1>  
        <button onClick={this.handleClick}>Κάντε κλικ μου</button>  
      </div>  
    );  
  }  
}
```

2.7.4.3 Άλλοι τύποι Components

Εκτός από τα λειτουργικά και δομικά components, υπάρχουν και άλλοι τύποι React components που αξίζει να αναφερθούν [16]:

- Καταστασιακά λειτουργικά components: Αυτά είναι λειτουργικά components που μπορούν να έχουν μία μεταβλητή κατάσταση. Χρησιμοποιούνται όταν χρειάζεται να υπάρχει πρόσβαση μόνο στην κατάσταση της μεθόδου render.
- Components υψηλότερης τάξης (HOCs): Αυτές είναι συναρτήσεις που λαμβάνουν ως παράμετρο άλλα components και επιστρέφουν νέα components με ενισχυμένη λειτουργικότητα. Τα HOCs είναι ένα ισχυρό εργαλείο για την επαναχρησιμοποίηση του κώδικα και τη δημιουργία επαναχρησιμοποιήσιμων components.
- Custom hooks: Αυτές είναι συναρτήσεις που επιτρέπουν την προσθήκη λειτουργικοτήτων σε React components χωρίς να χρειάζεται να γραφτούν class components.

2.7.5 Hooks

Τα React Hooks είναι μια πρόσφατη προσθήκη στη React που επιτρέπει τη χρήση καταστάσεων και άλλων React λειτουργιών χωρίς τη χρήση των class components. Με τα Hooks, οι προγραμματιστές μπορούν να δημιουργούν συναρτήσεις (ή "hooks") που να διαχειρίζονται την κατάσταση και την πλεονασματική λογική των συνιστωσών τους, κάνοντας τον κώδικα τους πιο απλό και ευανάγνωστο. Τα Hooks παρέχουν πρόσβαση στις βασικές λειτουργίες της React, όπως το state και το lifecycle των συνιστωσών, χωρίς να απαιτούν τη χρήση class components. Με αυτόν τον τρόπο, τα Hooks επιτρέπουν στους προγραμματιστές να γράφουν πιο συνοπτικό και ευανάγνωστο κώδικα, κάνοντας την ανάπτυξη εφαρμογών με τη React πιο ευέλικτη και αποδοτική.

Κώδικας 2.7 Παράδειγμα κώδικα που ορίζει το Hook `useState` για να δηλώσουμε ένα state με το όνομα `text`, το οποίο ξεκινά με μια κενή τιμή

```
import React, { useState } from 'react';

function ExampleComponent() {
  // Δήλωση ενός state με τη χρήση του Hook useState για το
  κείμενο που εισάγεται
  const [text, setText] = useState('');

  return (
    <div>
      <p>Το εισαγόμενο κείμενο είναι: {text}</p>
      /* Ενημέρωση της τιμής του state με κάθε αλλαγή στο πεδίο
      εισόδου */
      <input
        type="text"
        value={text}
        onChange={(e) => setText(e.target.value)}
      />
    </div>
  );
}

export default ExampleComponent;
```

2.7.6 Routes

Τα React Routes είναι ένα δημοφιλές εργαλείο που χρησιμοποιείται για τη δρομολόγηση (routing) σε μια εφαρμογή React. Τα Routes επιτρέπουν την αντιστοίχιση των διαφόρων μονοπατιών URL με συγκεκριμένες συνιστώσες της εφαρμογής, επιτρέποντας έτσι την αλλαγή του περιεχομένου της σελίδας χωρίς την ανάγκη για φόρτωση νέας σελίδας. Αυτό δημιουργεί μια ομαλή και απρόσκοπτη εμπειρία πλοήγησης για τους χρήστες της εφαρμογής. Τα React Routes χρησιμοποιούνται ευρέως σε μεγάλες εφαρμογές React για τη δημιουργία πολυσέλιδων ιστοσελίδων ή εφαρμογών με πολλές διαφορετικές σελίδες ή περιοχές. Επιπλέον, τα Routes επιτρέπουν την παράλληλη φόρτωση συνιστωσών, βελτιώνοντας την απόδοση της εφαρμογής και τον χρόνο απόκρισης.

2.7.7 Material UI

Για την δημιουργία διάφορων πινάκων προβολής δεδομένων, αλλά και μενού περιήγησης χρησιμοποιήθηκε η βιβλιοθήκη Material UI. Το Material UI είναι μια δημοφιλής βιβλιοθήκη UI για την

React που παρέχει μια συλλογή από έτοιμα components και στυλ που βασίζονται στο Material Design της Google. Η βιβλιοθήκη προσφέρει μια σειρά από πλεονεκτήματα, όπως [17]:

- Εύκολη εγκατάσταση και χρήση: Η Material UI είναι εύκολη στην εγκατάσταση και τη χρήση, ακόμη και για τους αρχάριους προγραμματιστές React.
- Πλούσιο σύνολο από components: Η βιβλιοθήκη περιλαμβάνει ένα πλούσιο σύνολο από components, από απλά κουμπιά και μορφές, έως σύνθετα components όπως πίνακες και καρτέλες.

Κώδικας 2.8 Παράδειγμα κώδικα για την δημιουργία ενός κουμπιού με την χρήση της Material UI

```
import React from 'react';

import Button from '@mui/material/Button';

function MaterialUIButton() {
  return (
    <div>
      <h1>Welcome to Material-UI Example</h1>
      <Button variant="contained" color="primary">
        Click me
      </Button>
    </div>
  );
}

export default MaterialUIButton;
```

2.7.8 Axios

Η εφαρμογή για να λάβει δεδομένα από τα API των βιβλιογραφικών βάσεων δεδομένων που θα αναλυθούν στο 3^ο κεφάλαιο, χρησιμοποιεί HTTP αιτήματα. Βασικό εργαλείο για αυτή την δουλειά είναι η βιβλιοθήκη της JavaScript Axios. Η βιβλιοθήκη Axios είναι μια δημοφιλής βιβλιοθήκη HTTP για την JavaScript που χρησιμοποιείται για την επικοινωνία με backend servers. Η βιβλιοθήκη παρέχει μια σειρά από λειτουργίες που βοηθάει τους προγραμματιστές να δημιουργούν HTTP αιτήματα, να χειρίζονται απαντήσεις και σφάλματα. Για να χρησιμοποιηθεί η βιβλιοθήκη Axios σε μια εφαρμογή React, μπορεί να γίνει εγκατάσταση χρησιμοποιώντας το npm ή το yarn. Μόλις εγκατασταθεί η βιβλιοθήκη, μπορεί να χρησιμοποιηθεί εισάγοντας τη στο component που θέλουμε.

Κώδικας 2.9 Παράδειγμα κώδικα για ένα αίτημα GET σε έναν backend server

```
const fetchData = async () => {  
    const response = await Axios.get("/api/data");  
  
    setData(response.data);  
};
```

2.8 Αποθήκευση δεδομένων στην πλευρά του χρήστη

Όπως αναφέραμε προηγουμένως, η εφαρμογή χρησιμοποιεί HTTP αιτήματα για να λάβει δεδομένα από τις βιβλιογραφικές βάσεις. Αυτά τα δεδομένα όμως χρειάζεται κάπως να αποθηκευτούν για να είναι διαθέσιμα στον χρήστη. Πιο συγκεκριμένα, η εφαρμογή δεν χρησιμοποιεί κάποια backend βάση δεδομένων για να αποθηκεύει στοιχεία για τους χρήστες της, αλλά αντιθέτως αξιοποιεί την αποθήκευση από την πλευρά του πελάτη. Με την τοπική αυτή αποθήκευση, τα δεδομένα αποθηκεύονται στο ίδιο το πρόγραμμα περιήγησης, δίνοντας στον χρήστη την δυνατότητα γρήγορης πρόσβασης σε όλες τις προηγούμενες αναζητήσεις του, χωρίς να περιμένει να ξανά φορτωθούν από τις εξωτερικές βάσεις.

Υπάρχουν διάφοροι τρόποι για την αποθήκευση δεδομένων στην πλευρά του χρήστη, όπως τα Cookie, το LocalStorage και το IndexedDB. Στο πλαίσιο της παρούσας εργασίας, επιλέχτηκε η χρήση του LocalStorage. Το LocalStorage είναι μια απλή αποθηκευτική λύση που επιτρέπει τη διατήρηση δεδομένων ως ζεύγη κλειδιού-τιμής στον τοπικό περιηγητή [18]. Αυτή η διαδικασία εξασφαλίζει την πρόσβαση των δεδομένων ακόμη και μετά το κλείσιμο του προγράμματος περιήγησης. Πριν από την αποθήκευσή τους, τα δεδομένα υφίστανται μετατροπή σε μορφή JSON (JavaScript Object Notation), η οποία αντιπροσωπεύει μια απλή και κατανοητή δομή δεδομένων.

2.9 Node Package Manager

Το NPM (Node Package Manager) είναι το μεγαλύτερο μητρώο λογισμικού στον κόσμο. Προγραμματιστές που ασχολούνται με την ανάπτυξη ανοιχτού κώδικα από όλο τον κόσμο, το χρησιμοποιούν για να μοιράζονται αλλά και να δανείζονται πακέτα και βιβλιοθήκες. Επιπλέον, πολλές μεγάλες εταιρίες το χρησιμοποιούν για την διαχείριση ιδιωτικής ανάπτυξης [19].

Το npm αποτελείται από τρία ξεχωριστά στοιχεία [19]:

- Τον ιστότοπο: όπου χρησιμοποιείται για την ανακάλυψη νέων πακέτων αλλά και για την ρύθμιση του προφίλ των χρηστών
- Τη Γραμμή Εντολών (CLI) : είναι ένα τερματικό που χρησιμοποιείται από τους προγραμματιστές για την αλληλεπίδραση με το NPM.
- Το μητρώο: είναι μια μεγάλη δημόσια βάση δεδομένων λογισμικού JavaScript και διάφορων σχετικών μετα-πληροφοριών.

2.10 React Developer Tools

Τα React Developer Tools είναι μια επέκταση του Chrome DevTools που παρέχει δυνατότητες για την επιθεώρηση και τη διαχείριση React εφαρμογών. Η επέκταση προσθέτει δύο νέες καρτέλες στα Chrome DevTools [20]:

- **Components:** Η καρτέλα αυτή εμφανίζει μια ιεραρχία των React components που εμφανίζονται στην τρέχουσα σελίδα. Ο χρήστης μπορεί να επιλέξει ένα component για να δει τα props του και το state του, καθώς και τις συναρτήσεις και τα hooks που χρησιμοποιεί.
- **Profiler:** Η καρτέλα αυτή επιτρέπει στον χρήστη να παρακολουθήσει την απόδοση μιας React εφαρμογής. Ο χρήστης μπορεί να βλέπει συγκεκριμένες ενέργειες που πραγματοποιούνται ή δεδομένα για να εντοπίσει σημεία σφάλματος ακόμη και βελτιστοποίησης.

Τα React Developer Tools είναι ένα σημαντικό εργαλείο για προγραμματιστές React και μπορεί να χρησιμοποιηθεί για την επίλυση προβλημάτων και τη βελτίωση της απόδοσης της εφαρμογής τους. Το εργαλείο αυτό είναι διαθέσιμο για όλα τα δημοφιλή προγράμματα περιήγησης και παρέχεται δωρεάν.

2.11 Visual Code

Το Visual Studio Code (VSCode) είναι ένα δωρεάν, ανοιχτού κώδικα (open-source) περιβάλλον ανάπτυξης (IDE) που παρέχει πολλές λειτουργίες για την ανάπτυξη λογισμικού. Δημιουργήθηκε από τη Microsoft και είναι διαθέσιμο για χρήση σε Windows, macOS, και Linux. Το Visual Studio Code είναι ένα ισχυρό και ευέλικτο εργαλείο που μπορεί να χρησιμοποιηθεί για την ανάπτυξη λογισμικού σε μια ποικιλία γλωσσών προγραμματισμού, συμπεριλαμβανομένων των JavaScript, HTML, CSS, Python, C/C++, Java, και πολλά άλλα [21]. Επίσης, υποστηρίζει μια μεγάλη ποικιλία από επεκτάσεις δίνοντας στον προγραμματιστή την δυνατότητα να προσαρμόσει το πρόγραμμα στις δικές του ανάγκες και βελτιστοποιώντας την παραγωγικότητα του.

2.12 Postman

Το Postman είναι ένα εργαλείο για την ανάπτυξη και δοκιμή API. Προορίζεται, ώστε να διευκολύνει τη διαδικασία δημιουργίας και δοκιμής των HTTP αιτημάτων. Αρχικά, το Postman ξεκίνησε ως μια επέκταση του Google Chrome, αλλά αργότερα εξελίχθηκε σε μια ανεξάρτητη εφαρμογή που υποστηρίζεται από διάφορα λειτουργικά συστήματα, συμπεριλαμβανομένων των Windows, macOS και Linux [22].

Οι βασικές λειτουργίες του Postman περιλαμβάνουν [22]:

- **Δημιουργία Αιτημάτων (Requests):** Μπορείτε να δημιουργήσετε διάφορα είδη HTTP αιτημάτων, όπως GET, POST, PUT, DELETE και PATCH, και να προσαρμόσετε τα δεδομένα και τις ρυθμίσεις των αιτημάτων σας.
- **Οργάνωση Αιτημάτων σε Συλλογές:** Μπορείτε να οργανώσετε τα αιτήματά σας σε συλλογές για ευκολότερη διαχείριση και κατανόηση.
- **Αυτοματοποίηση Δοκιμών (Testing):** Το Postman επιτρέπει τη δημιουργία αυτοματοποιημένων δοκιμών για τα αιτήματά σας, βοηθώντας στη διασφάλιση της ποιότητας των API.
- **Κοινοποίηση (Collaboration):** Παρέχει δυνατότητες κοινοποίησης και συνεργασίας μεταξύ των μελών μιας ομάδας ανάπτυξης.

- Παρακολούθηση (Monitoring): Το Postman προσφέρει λειτουργίες παρακολούθησης των API για εντοπισμό προβλημάτων και βελτιστοποίηση της απόδοσης.

2.13 Επίλογος

Στο κεφάλαιο αυτό παρουσιάστηκαν και αναλύθηκαν όλες οι τεχνολογίες και τα εργαλεία που χρησιμοποιήθηκαν για την δημιουργία της τρέχουσας εφαρμογής. Συνεχίζοντας στο επόμενο κεφάλαιο θα γίνει περιγραφή των βιβλιογραφικών βάσεων που χρησιμοποιήθηκαν για την αναζήτηση δεδομένων και τα αντίστοιχα API endpoints του καθενός.

Κεφάλαιο 3ο: Βιβλιογραφικές Βάσεις

3.1 Εισαγωγή

Σε αυτό το κεφάλαιο περιγράφονται οι βιβλιογραφικές βάσεις και η χρήση των αντίστοιχων API. Η ενότητα αυτή ξεκινά με μια εισαγωγή που προσφέρει γενική επισκόπηση του τι είναι οι βάσεις βιβλιογραφικών δεδομένων, περιγράφοντας τον τρόπο διαχείρισης και συντήρησής τους. Στη συνέχεια, εξετάζονται αναλυτικά οι τέσσερις κύριες βάσεις που είναι: το Crossref, το Scopus, το Open Alex και το Semantic Scholar. Για κάθε μία από αυτές, παρέχονται βασικές πληροφορίες, ενώ εστιάζεται η ανάλυση των Endpoints που χρησιμοποιούνται για την ανάκτηση δεδομένων, καθώς και των ερωτημάτων (query data) που αποστέλλονται προς αυτά. Τέλος, παρουσιάζεται η δημιουργία ενός διαγράμματος που εικονογραφεί τη ροή των δεδομένων στο πλαίσιο της εργασίας, καταγράφοντας τα διάφορα στάδια από τη στιγμή της εισαγωγής των δεδομένων έως την παρουσίαση των αποτελεσμάτων στον χρήστη.

3.2 API Βιβλιογραφιών

Τα τελευταία χρόνια, έχουν εμφανιστεί πολλές νέες πηγές βιβλιογραφικών δεδομένων, γεγονός που έχει διευρύνει το πλήθος των εργαλείων που είναι διαθέσιμα για την αναζήτηση επιστημονικής βιβλιογραφίας. Σε αντίθεση με τις παραδοσιακές βιβλιογραφικές βάσεις δεδομένων που παρέχονται από εμπορικές εταιρείες, όπως το Scopus και το Web of Science, ορισμένοι από αυτούς τους παρόχους βιβλιογραφικών πληροφοριών προσφέρουν τα μεταδεδομένα τους ελεύθερα στο κοινό. Αυτά τα μεταδεδομένα είναι ελεύθερα να χρησιμοποιηθούν καθώς θεωρούνται γεγονότα, έτσι αυτά δεν μπορούν να ιδιωτικοποιηθούν και επομένως δεν απαιτούν άδεια χρήσης [23].

Οι βιβλιογραφικές βάσεις δεδομένων είναι συλλογές δεδομένων που περιέχουν πληροφορίες σχετικά με δημοσιεύσεις, όπως άρθρα, βιβλία, μελέτες και δημοσιεύσεις. Οι πληροφορίες αυτές περιλαμβάνουν συνήθως τον τίτλο της δημοσίευσης, τους συγγραφείς, το έτος δημοσίευσης, το περιοδικό ή τον εκδοτικό οίκο [24]. Επιπλέον, οι πληροφορίες αυτές μπορούν να χρησιμοποιηθούν για διάφορους σκοπούς, όπως η έρευνα, η ανασκόπηση της βιβλιογραφίας και η παρακολούθηση της επιστήμης και της τεχνολογίας.

Τα δεδομένα των βάσεων συλλέγονται από διάφορες πηγές, όπως επιστημονικά περιοδικά, εκδοτικούς οίκους και κυβερνητικές υπηρεσίες. Τα δεδομένα μπορούν να συλλεχθούν αυτόματα ή χειροκίνητα. Η αυτόματη συλλογή δεδομένων είναι πιο αποτελεσματική, αλλά μπορεί να οδηγήσει σε σφάλματα. Αντίθετα, η χειροκίνητη συλλογή δεδομένων είναι λιγότερο αποτελεσματική, αλλά πιο ακριβής.

Οι περισσότερες βιβλιογραφικές βάσεις δεδομένων προσφέρουν πολλαπλές διεπαφές προγραμματισμού εφαρμογών (API) που επιτρέπουν την αυτοματοποιημένη ανάκτηση και χρήση των δεδομένων τους από ερευνητές και εφαρμογές. Αυτά τα API παρέχουν διάφορα Endpoints και δυνατότητες ερωτημάτων που επιτρέπουν την εξαγωγή συγκεκριμένων πληροφοριών ανάλογα με τις ανάγκες του χρήστη. Παρακάτω θα αναλύσουμε τις τέσσερις βιβλιογραφικές βάσεις που χρησιμοποιούνται στην εφαρμογή.

3.2.1 Crossref



Σχήμα 3.2.1 Λογότυπο Crossref

Το Crossref είναι μια μη κερδοσκοπική ένωση μελών για εκδότες επιστημονικών εργασιών, που ιδρύθηκε το 2000. Είναι ο μεγαλύτερος οργανισμός καταχώρισης ψηφιακών αναγνωριστικών αντικειμένων (DOI) και παρέχει στα μέλη του τη δυνατότητα να καταθέτουν DOI μαζί με τα μεταδεδομένα τους, για να υποστηρίξουν τη μόνιμη σύνδεση μεταξύ διαφορετικών τύπων ακαδημαϊκού περιεχομένου [25]. Τα DOI χρησιμοποιούνται κυρίως για εντοπισμό και την σύνδεση μεταξύ ερευνητικών έργων και πηγών.

Μια άλλη βασική υπηρεσία του Crossref είναι η υποβολή μεταδεδομένων. Το Crossref παρέχει μια πλατφόρμα για την υποβολή μεταδεδομένων για ερευνητικά αντικείμενα. Τα μεταδεδομένα είναι πληροφορίες σχετικά με ένα ερευνητικό αντικείμενο, όπως ο τίτλος, οι συγγραφείς, το έτος δημοσίευσης και οι λέξεις-κλειδιά. Επιπλέον, το Crossref παρέχει μια ποικιλία από API για την εξαγωγή δεδομένων από το σύστημα του. Τα API μπορούν να χρησιμοποιηθούν για την δημιουργία εφαρμογών που επιτρέπουν στους χρήστες να αναζητούν και να εξάγουν πληροφορίες σχετικά με ερευνητικά αντικείμενα. Επίσης, νέα δεδομένα και πληροφορίες που καταθέτονται στο Crossref, ξεκινάνε να εμφανίζονται 20 λεπτά αργότερα στο API [26].

Παρακάτω παρουσιάζονται κάποια από τα βασικά API που προσφέρει το Crossref [26]:

Όνομα API	Περιγραφή	Δεδομένα που επιστρέφει
Works	Επιστρέφει πληροφορίες σχετικά με ερευνητικά αντικείμενα, συμπεριλαμβανομένων των DOI, των μεταδεδομένων και των συνδέσμων προς τα ερευνητικά αντικείμενα.	<ul style="list-style-type: none"> • DOI • Τίτλος • Συγγραφείς • Έτος δημοσίευσης • Διεύθυνση URL • Μεταδεδομένα
Journals	Επιστρέφει πληροφορίες σχετικά με επιστημονικά περιοδικά, συμπεριλαμβανομένων των DOI, των	<ul style="list-style-type: none"> • DOI • Τίτλος • Εκδότης • Έτος έναρξης

	μεταδεδομένων και των συνδέσμων προς τα περιοδικά.	<ul style="list-style-type: none"> • Διεύθυνση URL • Μεταδεδομένα
Funders	Επιστρέφει πληροφορίες σχετικά με οργανισμούς χρηματοδότησης, συμπεριλαμβανομένων των DOI, των μεταδεδομένων και των συνδέσμων προς τους οργανισμούς.	<ul style="list-style-type: none"> • DOI • Όνομα • Γραφείο • Διεύθυνση URL • Μεταδεδομένα
Prefixes	Επιστρέφει πληροφορίες σχετικά με τα πρόθεμα DOI, συμπεριλαμβανομένων των ονομάτων, των ονομάτων τομέα και των περιγραφών.	<ul style="list-style-type: none"> • Όνομα • Όνομα τομέα • Περιγραφή
Members	Επιστρέφει πληροφορίες σχετικά με τα μέλη του Crossref, συμπεριλαμβανομένων των ονομάτων, των ιστότοπων και των τύπων μελών.	<ul style="list-style-type: none"> • Όνομα • Ιστότοπος • Τύπος μέλους
Types	Επιστρέφει πληροφορίες σχετικά με τους τύπους ερευνητικών αντικειμένων, συμπεριλαμβανομένων των ονομάτων, των περιγραφών και των κωδικών.	<ul style="list-style-type: none"> • Όνομα • Περιγραφή • Κωδικός
Licenses	Επιστρέφει πληροφορίες σχετικά με τις άδειες χρήσης ερευνητικών αντικειμένων, συμπεριλαμβανομένων των ονομάτων, των περιγραφών και των συνδέσμων.	<ul style="list-style-type: none"> • Όνομα • Περιγραφή • Σύνδεσμος
Deposits	Επιστρέφει πληροφορίες σχετικά με τις καταθέσεις ερευνητικών αντικειμένων στο Crossref, συμπεριλαμβανομένων των DOI, των ημερομηνιών και των συντελεστών.	<ul style="list-style-type: none"> • DOI • Ημερομηνία • Συντελεστής

Πίνακας 3.2.1 Τα βασικά API του Crossref

Για να διασφαλίσει το Crossref στους χρήστες του δωρεάν χρήση, χρειάζεται να επιβάλει περιορισμούς στην ταχύτητα αλλά και στον αριθμό των αιτημάτων που μπορούν να γίνουν. Οι περιορισμοί αυτοί αναφέρονται στις απαντήσεις των HTTP headers που δίνει το API. Τα “X-Rate-Limit-Limit” και “X-Rate-Limit-Interval”, είναι οι headers που καθορίζουν ότι επιτρέπονται να γίνουν X αιτήματα ανά ορισμένη χρονική περίοδο. Οι αριθμοί αυτοί μπορεί να διαφέρουν από αίτημα σε αίτημα, επομένως δεν μπορεί να παρέχεται ένας περιορισμός ταχύτητας που θα ισχύει πάντα.

Η εφαρμογή για την ανάκτηση των δεδομένων χρησιμοποιεί το Works API. Στον παρακάτω πίνακα παρουσιάζονται τα βασικά Endpoint που χρησιμοποιούνται από το Work API. Στη συνέχεια γίνεται αναλυτικότερη περιγραφή του καθενός.

Endpoint	Περιγραφή	Τύπος Request
/works?query={keyword}	Επιστρέφει ένα σύνολο ερευνητικών αντικειμένων με βάση διάφορα κριτήρια αναζήτησης.	GET
/works?query.author={keyword}	Επιστρέφει ένα σύνολο ερευνητικών αντικειμένων με συγγραφείς που ταιριάζουν σε ένα συγκεκριμένο μοτίβο.	GET
/works?query.affiliation={keyword}	Επιστρέφει ένα σύνολο ερευνητικών αντικειμένων με συγγραφείς που ανήκουν σε ένα συγκεκριμένο θεσμικό προσάρτημα.	GET
/works?query.container-title={keyword}	Επιστρέφει ένα σύνολο ερευνητικών αντικειμένων που έχουν δημοσιευτεί σε ένα συγκεκριμένο περιοδικό ή βιβλίο.	GET
/works?query.publisher-name={keyword}	Επιστρέφει ένα σύνολο ερευνητικών αντικειμένων που έχουν εκδοθεί από έναν συγκεκριμένο εκδότη.	GET

Πίνακας 3.2.2 Endpoints του Crossref Work API

3.2.1.1 Crossref Endpoints Εφαρμογής

/works?query={keyword}

Αυτό το endpoint επιστρέφει ένα σύνολο ερευνητικών αντικειμένων που ταιριάζουν σε ένα συγκεκριμένο κριτήριο αναζήτησης. Το κριτήριο αναζήτησης μπορεί να είναι ένα ή περισσότερα από τα ακόλουθα:

- Authors: Τα ονόματα των συγγραφέων ενός ερευνητικού αντικειμένου.
- Affiliation: Οι θεσμικές προσαρτήσεις των συγγραφέων ενός ερευνητικού αντικειμένου.
- Publication Name: Το όνομα του περιοδικού ή του βιβλίου στο οποίο έχει δημοσιευτεί ένα ερευνητικό αντικείμενο.
- Title Words: Οι λέξεις-κλειδιά του τίτλου ενός ερευνητικού αντικειμένου.
- Keywords: Οι λέξεις-κλειδιά ενός ερευνητικού αντικειμένου.

Κεφάλαιο 3

Για παράδειγμα, το ακόλουθο αίτημα θα επιστρέψει μια γραμμή ερευνητικών αντικειμένων που ταιριάζουν σε ένα κριτήριο αναζήτησης που είναι η συνάρτηση OR των 'Antonis' και 'Sidiropoulos' με τις πληροφορίες DOI, τίτλος, συγγραφέα, ημερομηνία έκδοσης, εκδότης:

<https://api.crossref.org/works?query=Antonis+Sidiropoulos&select=DOI,title,author,issued,publisher&rows=1>

Κώδικας 3.1 Απάντηση αιτήματος στο Crossref Work API

```
{
  "status": "ok",
  "message-type": "work-list",
  "message-version": "1.0.0",
  "message": {
    "facets": {},
    "total-results": 11897,
    "items": [
      {
        "DOI": "10.21203/rs.3.rs-2980768/v1",
        "title": [
          "Improved online DMP spatial generalization and
incorporation of dynamic via-points"
        ],
        "author": [
          {
            "given": "Antonis",
            "family": "Sidiropoulos",
            "sequence": "first",
            "affiliation": [
              {
                "name": "Aristotle University of
Thessaloniki"
              }
            ]
          },
          {
            "given": "Zoe",
            "family": "Doulgeri",
            "sequence": "additional",
            "affiliation": [
              {
```

```

    "name": "Aristotle University of
Thessaloniki"
    }
  ]
}
],
"issued": {
  "date-parts": [
    [
      2023,
      5,
      29
    ]
  ]
},
"publisher": "Research Square Platform LLC"
}
],
"items-per-page": 1,
"query": {
  "start-index": 0,
  "search-terms": "Antonis Sidiropoulos"
}
}
}

```

/works?query.author={keyword}

Επιστρέφει ένα σύνολο ερευνητικών αντικειμένων με συγγραφείς που ταιριάζουν σε ένα συγκεκριμένο μοτίβο. Το μοτίβο μπορεί να περιλαμβάνει οποιοδήποτε συνδυασμό λέξεων, αριθμών και συμβόλων.

Με το ακόλουθο αίτημα επιστρέφεται ένα σύνολο ερευνητικών αντικειμένων με συγγραφείς με το όνομα 'Giwrgos ή 'Papadopoulos':

<https://api.crossref.org/works?query.author=Giwrgos+Papadopoulos>

/works?query.affiliation={keyword}

Επιστρέφει ένα σύνολο ερευνητικών αντικειμένων με συγγραφείς που ανήκουν σε ένα συγκεκριμένο θεσμικό προσάρτημα. Το μοτίβο μπορεί να περιλαμβάνει οποιοδήποτε συνδυασμό λέξεων, αριθμών και συμβόλων.

Με το ακόλουθο αίτημα θα επιστρέψει ένα σύνολο ερευνητικών αντικειμένων με συγγραφείς που ανήκουν στο θεσμικό προσάρτημα "International Hellenic University":

<https://api.crossref.org/works?query.affiliation=International+Hellenic+University>

/works?query.container-title={keyword}

Επιστρέφει ένα σύνολο ερευνητικών αντικειμένων που έχουν δημοσιευτεί σε ένα συγκεκριμένο περιοδικό ή βιβλίο. Το μοτίβο μπορεί να περιλαμβάνει οποιοδήποτε συνδυασμό λέξεων, αριθμών και συμβόλων.

Με το ακόλουθο αίτημα θα επιστρέψει ένα σύνολο ερευνητικών αντικειμένων που έχουν δημοσιευτεί στο περιοδικό "Science":

<https://api.crossref.org/works?query.container-title=Science>

/works?query.publisher-name={keyword}

Επιστρέφει ένα σύνολο ερευνητικών αντικειμένων που έχουν εκδοθεί από έναν συγκεκριμένο εκδότη. Το μοτίβο μπορεί να περιλαμβάνει οποιοδήποτε συνδυασμό λέξεων, αριθμών και συμβόλων.

Με το ακόλουθο αίτημα θα επιστρέψει ένα σύνολο ερευνητικών αντικειμένων που έχουν εκδοθεί από τον εκδότη "Elsevier":

<https://api.crossref.org/works?query.publisher-name=Elsevier>

3.2.2 Scopus



Σχήμα 3.2.2 Λογότυπο Scopus

Το Scopus είναι μια διεθνής βιβλιογραφική βάση δεδομένων με περιλήψεις και παραπομπές, καθώς και συνδέσμους σε πλήρη κείμενα. Ιδρύθηκε το 2004 από την εταιρεία Elsevier και είναι μια από τις μεγαλύτερες βάσεις δεδομένων στον κόσμο, περιέχοντας μεγάλο αριθμό δεδομένων περιλήψεων και παραπομπών επιστημονικής βιβλιογραφίας. Καλύπτει περίπου 36.000 τίτλους επιστημονικών περιοδικών, βιβλίων και συνεδριακών εργασιών [27]. Το όνομα Scopus προέρχεται από το ελληνικό πτηνό κουκουβάγια, το οποίο είναι γνωστό για τις εξαιρετικές του ικανότητες πλοήγησης. Αυτό το

όνομα αντανακλά το στόχο του Scopus να βοηθήσει τους χρήστες να προηγηθούν στην τεράστια ποσότητα επιστημονικής βιβλιογραφίας [28].

Το Scopus χρησιμοποιείται από ερευνητές, φοιτητές, βιβλιοθηκονόμους και άλλους επαγγελματίες του χώρου της έρευνας για να εντοπίσουν και να αξιολογήσουν επιστημονικές δημοσιεύσεις. Για παράδειγμα, οι ερευνητές μπορούν να χρησιμοποιήσουν το Scopus για να βρουν πληροφορίες σχετικά με τις πιο πρόσφατες εξελίξεις στον τομέα τους ή για να αξιολογήσουν την ποιότητα μιας συγκεκριμένης δημοσίευσης. Οι φοιτητές μπορούν να χρησιμοποιήσουν το Scopus για να βρουν πληροφορίες για θέματα που μελετούν ή για να εντοπίσουν ερευνητές που εργάζονται σε σχετικά θέματα. Οι βιβλιοθηκονόμοι μπορούν να χρησιμοποιήσουν το Scopus για να ενημερώσουν τις συλλογές τους ή για να παρέχουν πληροφορίες στους χρήστες τους [28], [29].

Εκτός από τη χρήση του Scopus μέσω της διαδικτυακής ιστοσελίδας, οι χρήστες μπορούν να έχουν πρόσβαση στα δεδομένα του μέσω μιας μεγάλης ποικιλίας API που διαθέτει. Αυτά τα API επιτρέπουν στους χρήστες να αναπτύξουν τις δικές τους εφαρμογές που θα χρησιμοποιούν τα δεδομένα του Scopus [30]. Για παράδειγμα, ένας ερευνητής μπορεί να αναπτύξει μια εφαρμογή που θα του επιτρέψει να παρακολουθεί τις δημοσιεύσεις άλλων ερευνητών στο τομέα του. Ένας βιβλιοθηκονόμος μπορεί να αναπτύξει μια εφαρμογή που θα επιτρέψει στους χρήστες της βιβλιοθήκης να αναζητούν επιστημονικές δημοσιεύσεις.

Τα API του Scopus προσφέρουν μια ποικιλία επιλογών για τους χρήστες. Οι χρήστες μπορούν να επιλέξουν να χρησιμοποιήσουν τα API για να έχουν πρόσβαση σε συγκεκριμένα δεδομένα, όπως περιλήψεις, παραπομπές ή συγγραφείς. Οι χρήστες μπορούν επίσης να επιλέξουν να χρησιμοποιήσουν τα API για να έχουν πρόσβαση σε σύνθετα δεδομένα, όπως πίνακες δεδομένων ή γραφήματα [30].

Για να μπορέσει κάποιος να χρησιμοποιήσει τα API του Scopus, πρέπει να δημιουργήσει έναν λογαριασμό και να λάβει ένα κλειδί API. Το κλειδί API είναι ένα μοναδικό αναγνωριστικό που χρησιμοποιείται για την ταυτοποίηση όταν πραγματοποιεί αιτήσεις προς τα API του Scopus. Μόλις κάποιος λάβει το κλειδί API, μπορεί να ξεκινήσει να χρησιμοποιεί τα API του Scopus [30].

3.2.2.1 Scopus API

Η εφαρμογή για την ανάκτηση δεδομένων από την βάση του Scopus χρησιμοποιεί το Scopus Search API που παρουσιάζεται στον παρακάτω πίνακα.

Όνομα API	Περιγραφή
Scopus Search API	Αναζήτηση επιστημονικών δημοσιεύσεων με βάση το θέμα, τον συγγραφέα ή τον τίτλο τους

Πίνακας 3.2.3 Το Scopus API που χρησιμοποιείται στην εφαρμογή

Προηγουμένως, αναφέραμε ότι για την εκτέλεση αιτημάτων στα API του Scopus χρειάζεται να υπάρχει ένα κλειδί API. Το κάθε κλειδί παρέχει τη δυνατότητα να πραγματοποιούνται ορισμένες αιτήσεις για δεδομένα από τα API. Τα όρια αιτήσεων, δηλαδή ο αριθμός των αιτήσεων που πραγματοποιούνται,

επαναρυθμίζονται κάθε επτά ημέρες. Τα μέγιστα όρια αιτημάτων είναι μοναδικά για κάθε API, δεν υπάρχει μια ενιαία παγκόσμια ρύθμιση για ένα δεδομένο κλειδί API [30].

Σε κάθε απάντηση API, υπάρχουν τα ακόλουθα πεδία που αναφέρονται στα όρια:

- X-RateLimit-Limit: Δείχνει το μέγιστο αριθμό αιτημάτων (quota) του API.
- X-RateLimit-Remaining: Δείχνει τα υπόλοιπα διαθέσιμα αιτήματα (quota) του API.
- X-RateLimit-Reset: Η ημερομηνία/ώρα σε δευτερόλεπτα Epoch κατά την οποία επαναρυθμίζεται το όριο αιτημάτων του API.

Για παράδειγμα, εάν η απάντηση API έχει τα ακόλουθα πεδία:

X-RateLimit-Limit: 100

X-RateLimit-Remaining: 50

X-RateLimit-Reset: 1690966800

Αυτό σημαίνει ότι το μέγιστο όριο αιτημάτων για το συγκεκριμένο API είναι 100 αιτήσεις ανά λεπτό. Ο αριθμός των υπολειπόμενων αιτημάτων είναι 50, επομένως ο χρήστης μπορεί να πραγματοποιήσει ακόμα 50 αιτήσεις πριν ξεπεράσει το όριό του. Ο αριθμός αυτός θα επαναρυθμιστεί στις 2023-08-02 12:00:00 UTC.

Όνομα API	Προβολή Δεδομένων	Εβδομαδιαίο όριο δεδομένων	Αιτήματα το δευτερόλεπτο
Scopus Search	Standard View/ Max 200 αποτελέσματα Complete View / Max 25 αποτελέσματα Component View / Max 25 αποτελέσματα	20000	9

Πίνακας 3.2.4 Όρια δεδομένων για τα Scopus Search API

3.2.2.2 Scopus Endpoint εφαρμογής

/content/search/scopus

Το endpoint αυτό χρησιμοποιείται για την αναζήτηση δημοσιεύσεων. Πιο συγκεκριμένα, επιτρέπει την αναζήτηση στο πλήρες σώμα αποσπασμάτων στο Scopus. Κάθε αναζήτηση περιλαμβάνει έναν σύνδεσμο σε ένα απόσπασμα Scopus όπως επίσης και συνδέσμους σε ένα πλήρες άρθρο. Το endpoint πρέπει πάντα να παίρνει σαν παράμετρο το κλειδί API του χρήστη 'apiKey' και ακόμη χρειάζεται να παρέχεται και η παράμετρος 'query'. Η παράμετρος αυτή μπορεί να περιλαμβάνει μια σειρά από τελεστές και λειτουργίες που επιτρέπουν στους χρήστες να πραγματοποιούν ακριβείς και αποτελεσματικές αναζητήσεις.

Οι βασικές τιμές αναζήτησης της παραμέτρου 'query' είναι:

- Τελεστές: Οι τελεστές χρησιμοποιούνται για να καθορίσουν τον τρόπο με τον οποίο θα πραγματοποιηθεί η αναζήτηση. Για παράδειγμα, ο τελεστής AND χρησιμοποιείται για να

αναζητήσει έγγραφα που περιέχουν και τα δύο όρια αναζήτησης, ενώ ο τελεστής OR χρησιμοποιείται για να αναζητήσει έγγραφα που περιέχουν τουλάχιστον ένα από τα δύο όρια αναζήτησης.

- Λειτουργίες: Οι λειτουργίες χρησιμοποιούνται για να επεξεργαστούν τα όρια αναζήτησης. Για παράδειγμα, η λειτουργία AUTHOR-NAME() χρησιμοποιείται για να αναζητήσει έγγραφα που έχουν συγγραφέα με συγκεκριμένο όνομα.

Μερικά παραδείγματα αναζητήσεων στη γλώσσα αναζήτησης του Scopus είναι τα εξής:

- "covid-19" AND "vaccine": Αυτή η αναζήτηση θα επιστρέψει έγγραφα που περιέχουν και τις λέξεις "covid-19" και "vaccine".
- "artificial intelligence" OR "machine learning": Αυτή η αναζήτηση θα επιστρέψει έγγραφα που περιέχουν τουλάχιστον μία από τις λέξεις "artificial intelligence" ή "machine learning".
- AUTHOR-NAME("Giwrgos Papadopoulos"): Αυτή η αναζήτηση θα επιστρέψει έγγραφα που έχουν συγγραφέα με το όνομα "Giwrgos Papadopoulos".

Εδώ είναι ένα πιο σύνθετο παράδειγμα:

Κώδικας 3.2 Σύνθετο παράδειγμα API Endpoint του Scopus

```
https://api.elsevier.com/content/search/scopus?apiKey=XXX&query=TITLE(Machine learning for natural language processing) AND (AUTHOR-NAME(Giwrgos Papadopoulos) OR AUTHOR-NAME(Nikos Nikas)) AND (PUBYEAR > 2022 AND PUBYEAR < 2024)
```

Αυτή η αναζήτηση θα επιστρέψει έγγραφα που έχουν τον τίτλο "Machine learning for natural language processing" και έχουν συγγραφέα με το όνομα "Giwrgos Papadopoulos" ή "Nikos Nikas" και έχουν δημοσιευθεί μεταξύ του 2022 και του 2023).

3.2.3 Open Alex



Σχήμα 3.2.3 Λογότυπο Open Alex

Το OpenAlex είναι μια ανοιχτή βάση δεδομένων επιστημονικής βιβλιογραφίας που δημιουργήθηκε από μια ομάδα ερευνητών και ακαδημαϊκών με στόχο την παροχή ελεύθερης και ανοιχτής πρόσβασης σε επιστημονική γνώση. Το όνομα της είναι εμπνευσμένο από την αρχαία βιβλιοθήκη της Αλεξάνδρειας. Η βάση δεδομένων περιέχει πληροφορίες για πάνω από 200 εκατομμύρια επιστημονικά έγγραφα, συμπεριλαμβανομένων άρθρων, βιβλίων, συνεδριακών ανακοινώσεων και άλλων μορφών δημοσιεύσεων [31].

Η ιδέα για τη δημιουργία της βάσης δεδομένων προέκυψε από την ανάγκη για μια ανοιχτή και ελεύθερη πρόσβαση στην επιστημονική γνώση. Οι δημιουργοί του OpenAlex πίστευαν ότι η ελεύθερη πρόσβαση στην επιστημονική γνώση είναι απαραίτητη για την προώθηση της επιστημονικής έρευνας και της καινοτομίας [31]. Το OpenAlex χρησιμοποιείται από ερευνητές, φοιτητές και άλλους ενδιαφερόμενους με σκοπό την εύρεση πληροφοριών για επιστημονικές έρευνες. Η βάση δεδομένων είναι ιδιαίτερα χρήσιμη για την εύρεση πληροφοριών για έρευνες που δεν είναι διαθέσιμες σε εμπορικές βάσεις δεδομένων, όπως έρευνες από αναπτυσσόμενες χώρες ή έρευνες που δημοσιεύονται σε λιγότερο γνωστά περιοδικά.

Το OpenAlex προσφέρει τα δεδομένα του με τρεις τρόπους [31]:

- Μέσω της ιστοσελίδας του, οι χρήστες μπορούν να αναζητήσουν επιστημονικά έγγραφα με βάση διάφορα κριτήρια, όπως τίτλος, συγγραφείς, λέξεις-κλειδιά, κλπ. Η ιστοσελίδα παρέχει επίσης διάφορα εργαλεία και δυνατότητες για την ανάλυση των δεδομένων, όπως η δημιουργία γραφημάτων και πινάκων.
- Μέσω του API του, οι προγραμματιστές μπορούν να έχουν πρόσβαση στα δεδομένα του OpenAlex προγραμματιστικά. Το API παρέχει μια ποικιλία λειτουργιών, όπως η αναζήτηση, η ανάκτηση δεδομένων και η δημιουργία γραφημάτων.
- Μέσω του στιγμιότυπου δεδομένων, οι χρήστες μπορούν να κατεβάσουν ένα πλήρες αντίγραφο των δεδομένων του OpenAlex σε ένα αρχείο. Το στιγμιότυπο δεδομένων είναι διαθέσιμο δωρεάν και ενημερώνεται τακτικά.

Ο κάθε τρόπος προσφέρει τα δεδομένα του OpenAlex με διαφορετικό τρόπο και για διαφορετικές ανάγκες. Η ιστοσελίδα του OpenAlex είναι η πιο εύκολη επιλογή για τους χρήστες που θέλουν να αναζητήσουν πληροφορίες για επιστημονικές έρευνες. Το API του OpenAlex είναι η καλύτερη επιλογή για τους προγραμματιστές που θέλουν να δημιουργήσουν εφαρμογές και υπηρεσίες που βασίζονται στα δεδομένα του OpenAlex. Το στιγμιότυπο δεδομένων είναι η καλύτερη επιλογή για τους χρήστες που θέλουν να έχουν πρόσβαση σε ένα πλήρες αντίγραφο των δεδομένων του OpenAlex [31].

3.2.3.1 OpenAlex API

Το OpenAlex προσφέρει μια ποικιλία API που παρέχουν πρόσβαση σε πληροφορίες για επιστημονικά έγγραφα, συγγραφείς, ιδρύματα, πηγές και άλλες οντότητες. Τα API είναι διαθέσιμα με τη μορφή REST API και είναι γρήγορα και εύκολα στη χρήση.

Τα βασικά API του OpenAlex είναι τα εξής:

Όνομα API	Περιγραφή
Works	Επιστρέφει πληροφορίες για επιστημονικά έγγραφα, όπως τίτλος, συγγραφείς και λέξεις-κλειδιά

Authors	Επιστρέφει πληροφορίες για συγγραφείς, όπως όνομα, επώνυμο και ιδιότητα
Institutions	Επιστρέφει πληροφορίες για ιδρύματα, όπως όνομα και χώρα
Sources	Επιστρέφει πληροφορίες για πηγές, όπως τίτλος και εκδότης

Πίνακας 3.2.5 Τα βασικά API που έχει η βάση OpenAlex

Τα API του OpenAlex διαθέτουν ορισμένα όρια ταχύτητας για την προστασία του συστήματος από υπερφόρτωση. Τα όρια αυτά είναι τα εξής [31]:

- Μέγιστος αριθμός κλήσεων ανά δευτερόλεπτο: 10 κλήσεις
- Μέγιστος αριθμός κλήσεων ανά ημέρα: 100.000 κλήσεις

Για χρήστες που έχουν ανάγκη να κάνουν περισσότερες κλήσεις από το επιτρεπόμενο όριο, μπορούν να εγγραφούν στο Premium πρόγραμμα που παρέχει το OpenAlex και δίνει την δυνατότητα για 1.000.000 κλήσεις ανά ημέρα.

3.2.3.2 OpenAlex Endpoint εφαρμογής

Η εφαρμογή για τη λήψη δεδομένων από τη βάση δεδομένων του OpenAlex χρησιμοποιεί το Works API, το οποίο παρέχει διάφορα Endpoints για την ανάκτηση πληροφοριών. Παρακάτω αναλύονται τα συγκεκριμένα Endpoints:

1. `/works?filter=title.search:{keywords}`

Αυτό το Endpoint επιστρέφει μια λίστα επιστημονικών έργων που περιέχουν τις καθορισμένες λέξεις-κλειδιά στους τίτλους τους. Για παράδειγμα, μια αίτηση όπως `GET /works?filter=title.search:τεχνητή+νοημοσύνη` θα επιστρέψει μια λίστα έργων που περιέχουν τη φράση "τεχνητή νοημοσύνη" στους τίτλους τους.

2. `/works?filter=authorships.author.id:{keywords}`

Αυτό το Endpoint επιστρέφει μια λίστα επιστημονικών έργων στα οποία οι καθορισμένες λέξεις-κλειδιά αναφέρονται στον συγγραφέα, είτε ως πρώτος είτε ως τελευταίος. Για παράδειγμα, μια αίτηση όπως `GET /works?filter=authorships.author.id:Γιώργος+Παπαδόπουλος` θα επιστρέψει μια λίστα έργων στα οποία ο συγγραφέας "Γιώργος Παπαδόπουλος" αναφέρεται είτε ως πρώτος είτε ως τελευταίος.

3. `/works?filter=authorships.institutions:{keywords}`

Αυτό το Endpoint επιστρέφει μια λίστα επιστημονικών έργων που έχουν δημοσιευθεί από ιδρύματα που ταιριάζουν στις καθορισμένες λέξεις-κλειδιά ιδρύματος. Για παράδειγμα, μια αίτηση όπως `GET`

`/works?filter=authorships.institutions:International+Hellenic+University` θα επιστρέψει μια λίστα έργων που έχουν δημοσιευθεί από το International Hellenic University (IHU).

4. `/works?filter=keywords.keyword:{keywords}`

Αυτό το Endpoint επιστρέφει μια λίστα επιστημονικών έργων που περιέχουν οποιαδήποτε από τις καθορισμένες λέξεις-κλειδιά στους τίτλους, τα abstract ή το πλήρες κείμενο τους. Για παράδειγμα, μια αίτηση όπως `GET /works?filter=keywords.keyword:τεχνητή+νοημοσύνη+μαθησιακή+μηχανή` θα επιστρέψει μια λίστα έργων που περιέχουν είτε τη φράση "τεχνητή νοημοσύνη" είτε τη φράση "μαθησιακή μηχανή" στους τίτλους, τα Abstract ή το πλήρες κείμενο τους.

3.2.4 Semantic Scholar



Σχήμα 3.2.4 Λογότυπο Semantic Scholar

Το Semantic Scholar, δημιουργήθηκε το 2015 από το Allen Institute for Artificial Intelligence και σχεδιάστηκε για να βοηθήσει τους επιστήμονες να αντιμετωπίσουν την υπερπληροφόρηση και να κατανοήσουν πιο αποτελεσματικά την πιο σχετική ερευνητική βιβλιογραφία. Μέσω μιας αυξανόμενης σειράς συνεργασιών με επιστημονικούς εκδότες και υπηρεσίες προεκτύπωσης, το Semantic Scholar έχει δημιουργήσει μια πλήρης και ανοιχτή συλλογή επιστημονικών δημοσιεύσεων ως δημόσια υπηρεσία. Ο ιστότοπος του Semantic Scholar παρέχει πολλές δυνατότητες, όπως η αυτόματη δημιουργία ιστοσελίδων για συγγραφείς, οι εξατομικευμένες βιβλιοθήκες και οι συστάσεις άρθρων. Επιπλέον, η πλειοψηφία των δεδομένων και λειτουργιών του, παρέχονται μέσω του ιστότοπου, των open-source βιβλιοθηκών και των υπηρεσιών API [32].

Το Semantic Scholar είναι μια ολοκληρωμένη βάση δεδομένων επιστημονικών έργων που μπορεί να χρησιμοποιηθεί από ερευνητές, φοιτητές, δημοσιογράφους και ενδιαφερόμενους πολίτες. Η βάση δεδομένων περιλαμβάνει πληροφορίες για περισσότερα από 200 εκατομμύρια έργα και παρέχει μια σειρά από εργαλεία που μπορούν να βοηθήσουν τους χρήστες να βρουν, να αξιολογήσουν και να χρησιμοποιήσουν τις πληροφορίες αυτές [33].

3.2.4.1 Semantic Scholar API

Το Semantic Scholar διαθέτει δύο διαφορετικά API: το δημόσιο API και το API για συνεργάτες. Το δημόσιο API παρέχει βασικές πληροφορίες σχετικά με επιστημονικά έργα, όπως τίτλους, συγγραφείς, χρονολογίες δημοσίευσης, λήμματα, παραπομπές και άλλες σχετικές πληροφορίες. Από την άλλη πλευρά, το API για συνεργάτες παρέχει πρόσβαση σε πρόσθετες πληροφορίες και λειτουργίες που επεκτείνουν τη λειτουργικότητα του, όπως υψηλότερα όρια ταχύτητας, πλήρη κείμενα επιστημονικών έργων, αναλύσεις παραπομπών, συγκριτικές πληροφορίες για επιστημονικά έργα και δυνατότητα ανάπτυξης νέων λειτουργιών [33].

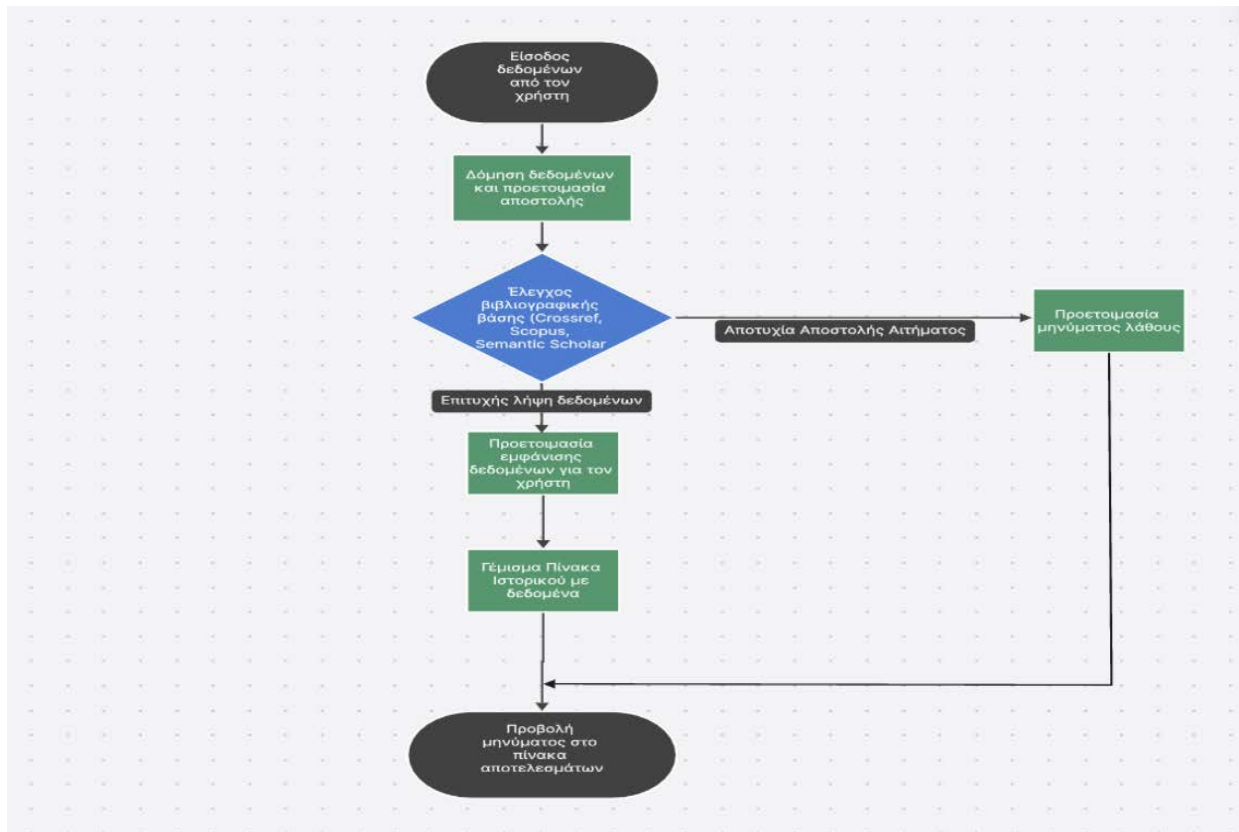
3.2.4.2 Semantic Endpoint Εφαρμογής

Η εφαρμογή για τη λήψη δεδομένων από τη βάση του Semantic Scholar χρησιμοποιεί το Δημόσιο API του. Αυτό το βασικό API παρέχει τη λειτουργία του Paper Relevance Search (Προσδιορισμός Σχετικότητας Εργασιών). Μέσω αυτής της λειτουργίας, επιστημονικά έργα αξιολογούνται και ταξινομούνται βάσει της συνάφειάς τους με ένα συγκεκριμένο ερώτημα ή θέμα. Ο στόχος είναι να βοηθήσει τους ερευνητές να εντοπίζουν γρήγορα και εύκολα τα πιο σχετικά και χρήσιμα έργα για την έρευνά τους.

Το endpoint αυτής της λειτουργίας είναι το `‘/graph/v1/paper/search’` και έχει σαν υποχρεωτική παράμετρο το πεδίο "query". Το "query" μπορεί να είναι μια λέξη-κλειδί, ο τίτλος ή το όνομα του συγγραφέα για αναζήτηση. Ως απάντηση, το API επιστρέφει μόνο τα 1000 πιο σχετικά αποτελέσματα με το ερώτημα και με μέγιστο μέγεθος τα 10MB.

3.3 Ροή Δεδομένων

Η ροή των δεδομένων της εφαρμογής είναι η εξής. Αρχικά ο χρήστης εισάγει δεδομένα στα πεδία που υπάρχουν και στην συνέχεια τα δεδομένα ετοιμάζονται για να σταλθούν με ένα ασύγχρονο αίτημα Ajax στην αντίστοιχη βιβλιογραφική βάση που επέλεξε ο χρήστης. Ύστερα, παίρνοντας απάντηση από το API της βάσης η εφαρμογή θα εκτελέσει μια από τις δυο ακόλουθες λειτουργίες. Σε περίπτωση που δεν υπάρχει κάποιο πρόβλημα στο αίτημα που έγινε, τα δεδομένα θα προετοιμαστούν για να εισαχθούν στους πίνακες και ταυτόχρονα θα αποθηκευτούν στην μνήμη του ιστορικού αναζήτησης του χρήστη. Αντίθετα, αν υπάρχει κάποιο πρόβλημα με την επικοινωνία ή κάποιο λάθος, θα ετοιμαστεί το αντίστοιχο μήνυμα για να παρουσιαστεί στο χρήστη. Τέλος, αναλόγως την επιτυχία/αποτυχία του αιτήματος θα εμφανιστεί στο χρήστη το αντίστοιχο μήνυμα.



Σχήμα 3.3.1 Διάγραμμα ροής δεδομένων εφαρμογής

3.4 Επίλογος

Στο κεφάλαιο αυτό παρουσιάστηκαν όλες οι βιβλιογραφικές βάσεις δεδομένων που χρησιμοποιήθηκαν για την λήψη δεδομένων στην εφαρμογή. Επίσης, αναλύθηκαν όλα τα API και τα endpoints που χρησιμοποιήθηκαν. Τέλος, παρουσιάστηκε και εξηγήθηκε το διάγραμμα που δείχνει την εξέλιξη της ροής από την εισαγωγή πληροφορίας από τον χρήστη (User Input) μέχρι και την επικοινωνία με τα API endpoints και την προβολή τους στην εφαρμογή.

Κεφάλαιο 4ο: Παρουσίαση Εφαρμογής

4.1 Εισαγωγή

Η παρούσα εφαρμογή είναι ένα εργαλείο που επιτρέπει στους χρήστες να αναζητήσουν και να εξάγουν πληροφορίες σχετικά με δημοσιεύσεις από διαφορετικές βιβλιογραφικές βάσεις δεδομένων. Η εφαρμογή παρέχει μια απλή και φιλική προς το χρήστη διεπαφή που καθιστά εύκολη την εύρεση των πληροφοριών που χρειάζεται.

Η εφαρμογή υποστηρίζει μια ευρεία γκάμα βιβλιογραφικών βάσεων δεδομένων, συμπεριλαμβανομένων:

- Crossref
- Scopus
- Semantic Scholar
- Open Alex

Επίσης, παρέχει διάφορες λειτουργίες που καθιστούν εύκολη την εύρεση των πληροφοριών που χρειάζεται ο χρήστης, συμπεριλαμβανομένων:

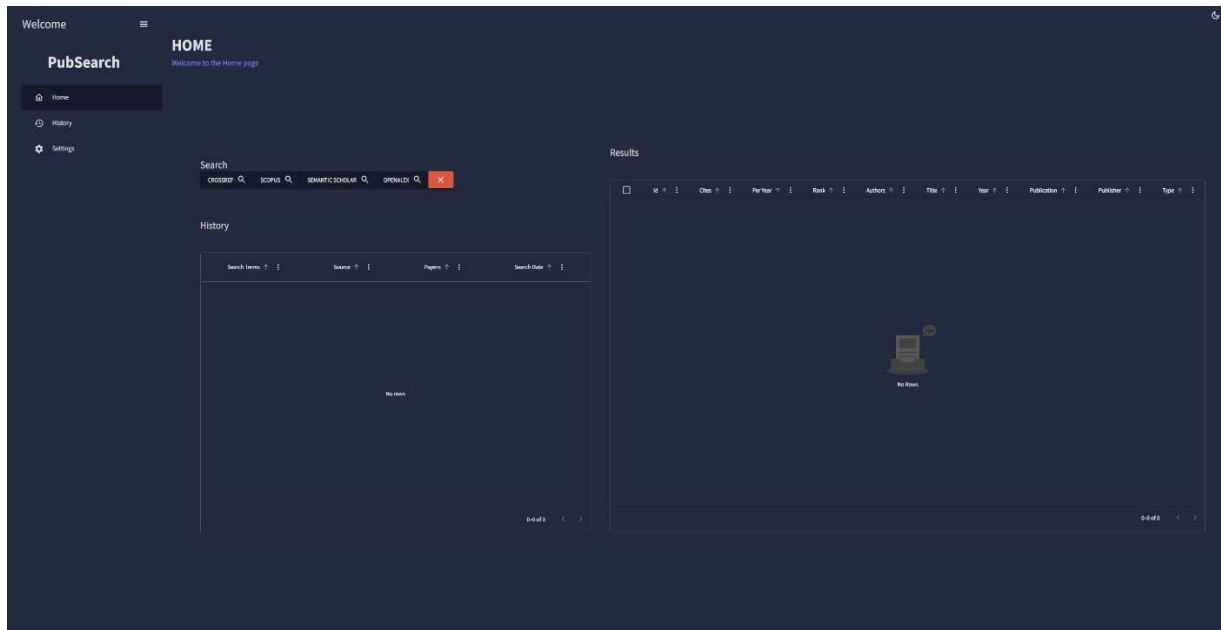
- Αναζήτηση με βάση τον τίτλο, τους συγγραφείς, το θέμα ή άλλες πληροφορίες
- Ταξινόμηση των αποτελεσμάτων αναζήτησης με βάση τον τίτλο, τους συγγραφείς, την ημερομηνία δημοσίευσης ή άλλες πληροφορίες
- Φιλτράρισμα των αποτελεσμάτων αναζήτησης με βάση διάφορα κριτήρια
- Προβολή πληροφοριών για μια συγκεκριμένη δημοσίευση, όπως τον τίτλο, τους συγγραφείς, το θέμα, την ημερομηνία δημοσίευσης, το περιοδικό ή τον εκδοτικό οίκο, και το πλήρες κείμενο της δημοσίευσης

Η εφαρμογή είναι ένα χρήσιμο εργαλείο για φοιτητές, ερευνητές και επαγγελματίες που χρειάζονται πρόσβαση σε πληροφορίες σχετικά με δημοσιεύσεις. Σε αυτό το κεφάλαιο παρουσιάζονται όλες οι οθόνες της εφαρμογής. Με τις οθόνες θα γίνει παρουσίαση των λειτουργιών κάθε οθόνης και με τα κατάλληλα στιγμιότυπα στις περιπτώσεις χρήσης.

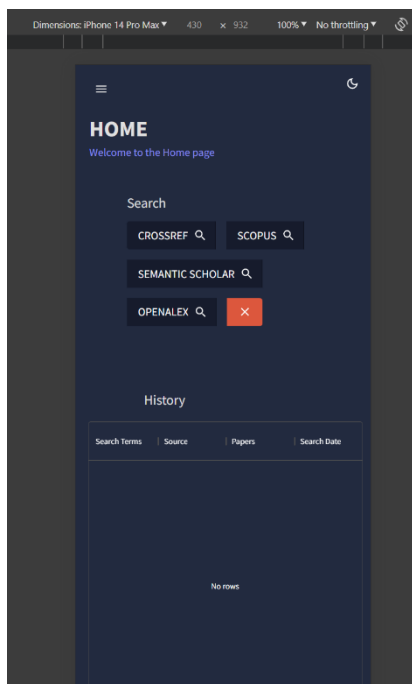
4.2 Αρχική οθόνη

Ο χρήστης όταν ανοίγει την εφαρμογή από ένα browser, το πρώτο πράγμα που βλέπει είναι η αρχική οθόνη, η οποία αποτελείται από τέσσερα βασικά στοιχεία. Όλες οι βασικές δυνατότητες βρίσκονται σε αυτό το σημείο. Το πρώτο και κύριο στοιχείο είναι ο πίνακας των αποτελεσμάτων που βρίσκεται στην αριστερή μεριά και περιλαμβάνει τα δεδομένα από τις αναζητήσεις που γίνονται από το χρήστη. Το δεύτερο είναι η μπάρα αναζήτησης που περιλαμβάνει τέσσερις διαφορετικές καρτέλες με πεδία που χρησιμοποιούνται για να κάνει ο χρήστης αιτήματα προς τις βιβλιογραφικές βάσεις δεδομένων και ένα κουμπί για το κλείσιμο του πτυσσόμενου μενού των πεδίων εφόσον έχει ανοιχτεί. Οι δυο από τις καρτέλες περιλαμβάνουν ίδια πεδία καθώς τα endpoints για την λήψη των δεδομένων που παρέχονται είναι πανομοιότυπα. Επιπλέον, στην αριστερή μεριά βρίσκεται ένα μενού περιήγησης που εμφανίζει τις τρεις σελίδες στις οποίες μπορεί να περιηγηθεί ο χρήστης στην εφαρμογή όπου είναι η Αρχική, το Ιστορικό και οι Ρυθμίσεις. Τέλος, ακριβώς κάτω από την μπάρα αναζήτησης βρίσκεται ένα πίνακας ιστορικού για γρήγορη πρόσβαση σε δεδομένα παλιότερων αναζητήσεων.

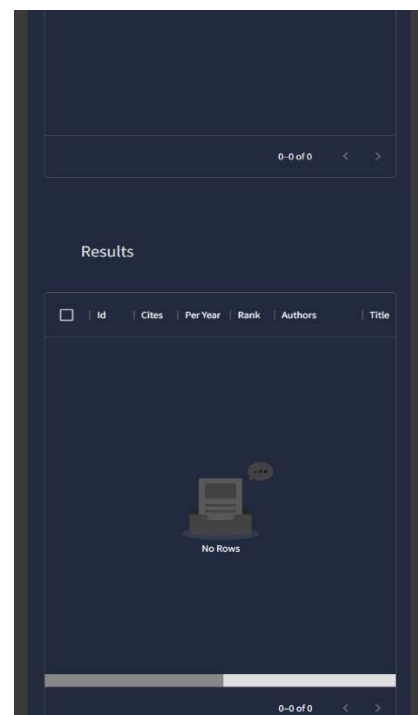
Κεφάλαιο 4



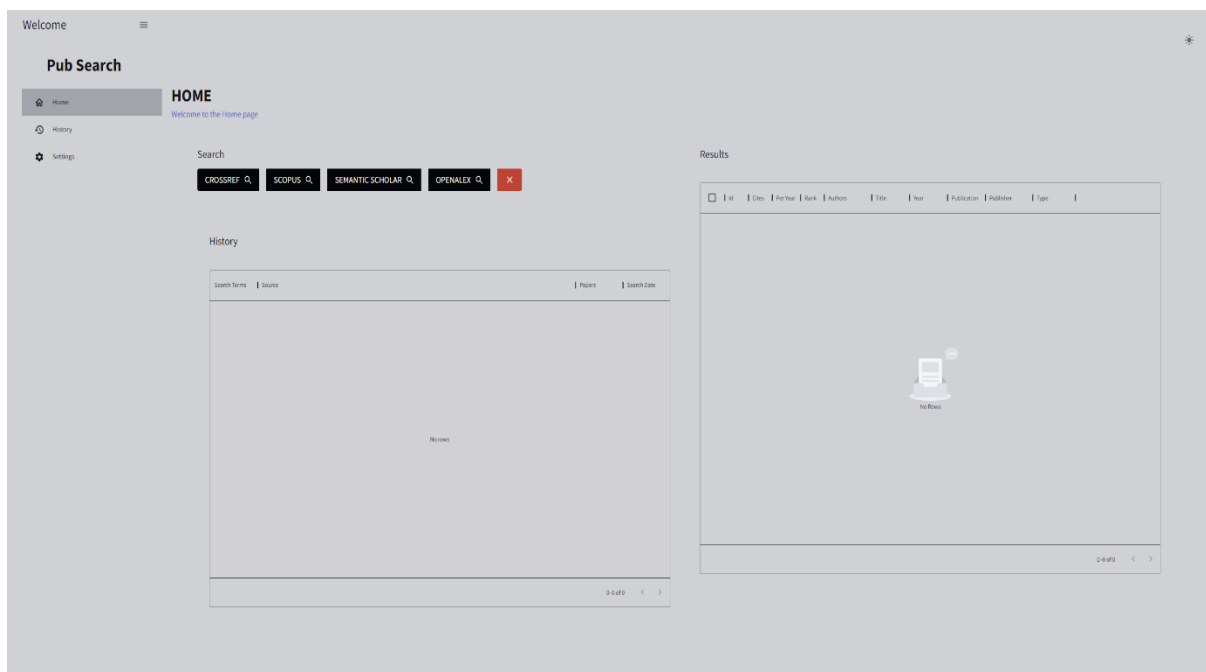
Εικόνα 4.1 Αρχική Οθόνη Εφαρμογής από υπολογιστή



Εικόνα 4.3 Αρχική Οθόνη Εφαρμογής σε κινητού πάνω μέρος



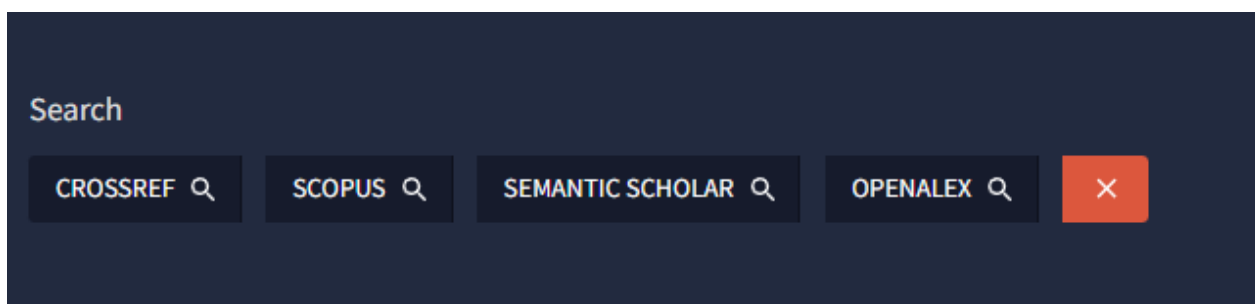
Εικόνα 4.2 Αρχική Οθόνη Εφαρμογής σε κινητού κάτω μέρος



Εικόνα 4.4 Αρχική Οθόνη Εφαρμογής από υπολογιστή με Light Mode

4.3 Μπάρα επιλογής βάσης αναζήτησης

Για να πραγματοποιήσει ο χρήστης οποιαδήποτε ενέργεια στην εφαρμογή, πρέπει να χρησιμοποιήσει την μπάρα αναζήτησης. Στην μπάρα αναζήτησης παρουσιάζονται οι τέσσερις βάσεις δεδομένων που αναλύθηκαν προηγουμένως, ξεκινώντας από την αριστερή πλευρά με το Crossref, ακολουθούμενο από το Scopus, το Semantic Scholar και το Open Alex. Κάθε μία από αυτές τις βάσεις έχει τους δικούς της περιορισμούς και πλεονεκτήματα για τη χρήση της. Τέλος, στο δεξί μέρος της μπάρας αναζήτησης βρίσκεται το κουμπί για το κλείσιμο του πτυσσόμενου μενού που εμφανίζεται, όταν ο χρήστης επιλέξει μια από τις τέσσερις βάσεις δεδομένων.



Εικόνα 4.5 Μπάρα επιλογής βάσης αναζήτησης

4.4 Πίνακας αποτελεσμάτων αναζήτησης

Ο πίνακας αποτελεσμάτων είναι το πιο βασικό στοιχείο στην εφαρμογή, καθώς δείχνει βασικές πληροφορίες για την αναζήτηση που θα πραγματοποιήσει ο χρήστης. Αρχικά, τα δεδομένα για να

Κεφάλαιο 4

μπορούν να ταξινομηθούν εύκολα στον πίνακα παίρνουν ένα μοναδικό ID, το οποίο χρησιμοποιείται για την ταξινόμηση και αποθήκευση τους. Οι τιμές που παίρνει είναι ακέραιοι αριθμοί που προσαρξάνονται καθώς έρχονται τα δεδομένα από ένα αίτημα σε μια βάση δεδομένων. Στην συνέχεια έχουμε την δεύτερη στήλη που περιέχει τα Cites. Τα Cites δείχνουν τον αριθμό των παραπομπών που έχουν γίνει για μια δημοσίευση που έχει έρθει από την βάση. Για παράδειγμα, όταν ένας επιστήμονας δημοσιεύσει μια έρευνα και κάποιος άλλος την χρησιμοποιήσει για να βασίσει μια ιδέα ή ένα επιχειρήμα, αυτό δείχνει ότι έγινε μια παραπομπή (Citation) στην συγκεκριμένη εργασία. Το συγκεκριμένο πεδίο αποτελεί το βασικότερο στο συγκεκριμένο πίνακα, καθώς επιβεβαιώνει ότι η έρευνα ενός επιστήμονα έχει μεγάλη επιρροή αλλά και αναγνωρισιμότητα στο πεδίο του. Η επόμενη στήλη (Per Year), δείχνει τις παραπομπές ανά έτος που λαμβάνει ένα άρθρο κάθε χρόνο από την ημερομηνία δημοσίευσής του. Για να την υπολογίσουμε, διαιρούμε το συνολικό αριθμό των παραπομπών που έχει λάβει το άρθρο, με την ηλικία του άρθρου σε έτη (δηλαδή, τον αριθμό των ετών που έχουν περάσει από τη δημοσίευσή του). Εάν η χρονολογία δημοσίευσης δεν είναι διαθέσιμη, η στήλη αυτή εμφανίζει την τιμή 0.00. Αυτός ο δείκτης μας βοηθάει να κατανοήσουμε την επίδραση ενός άρθρου στην επιστημονική κοινότητα μέσα στο χρόνο, με βάση τις παραπομπές που λαμβάνει κάθε χρόνο. Οι επόμενες τέσσερις στήλες είναι οι βασικές πληροφορίες ενός έργου, αυτά είναι το όνομα του ή των συγγραφέων του άρθρου (Authors), τον τίτλο που έχει (Title), την ημερομηνία έκδοσης (Year) και το όνομα του περιοδικού που δημοσιεύει το άρθρο (Publish). Φτάνοντας στο τέλος του πίνακα έχουμε τις τελευταίες δυο στήλες, όπου η πρώτη δείχνει τον εκδότη ή τον οργανισμό που εκδίδει ένα έγγραφο, όπως ένα βιβλίο, ένα άρθρο ή μια εργασία. Αναφέρει το όνομα του εκδότη ή του οργανισμού που έχει χρηματοδοτήσει, παράξει ή αναλάβει τη δημοσίευση του έργου (Publisher) και η στήλη (Type) αναφέρεται στον τύπο ή την κατηγορία του έργου που δημοσιεύεται. Για παράδειγμα, μπορεί να αναφέρεται σε ένα άρθρο, ένα βιβλίο, μια διδακτορική διατριβή, μια επιστημονική ανακοίνωση, ένα πρακτικό άρθρο κ.λπ. Αυτό το πεδίο παρέχει πληροφορίες σχετικά με τη μορφή ή το είδος του έργου και βοηθά στον εντοπισμό και την ταξινόμηση των διαφόρων ειδών δημοσιεύσεων. Τέλος, ο πίνακας δίνει την δυνατότητα στο χρήστη να ταξινομήσει και να φιλτράρει τα δεδομένα, όπως επίσης και να κρύψει κάποια στήλη που δεν το ενδιαφέρει.

The screenshot shows a search interface with a dark theme. At the top left, it says 'HOME' and 'Welcome to the Home page'. Below this is a search bar with several database options: CROSSREF, SCOPUS (highlighted in green), SEMANTIC SCHOLAR, and OPENALEX. There are also input fields for Authors (Sidiroupolos Antonis), Affiliation, Publication Name, Title Words, and Keywords. A 'SEARCH' button and a 'CLEAR' button are visible. Below the search bar is a 'History' section with a table of search terms, sources, papers, and search dates.


On the right side, there is a 'Results' section displaying a table of search results. The table has columns for Id, Cites, Per Year, Rank, Authors, Title, Year, Publication, Publisher, and Type. The results are sorted by Year in descending order.

Id	Cites	Per Year	Rank	Authors	Title	Year	Publication	Publisher	Type
21	0			Sidropo...	Dynamic Via-points and...	2024	Journal o...	Ar	
13	4			Sidropo...	A novel framework for g...	2023	Autonom...	Ar	
22	0			Psychop...	Monitoring capabilities ...	2023	Journal o...	Ar	
23	0			Bikas I.N.	Model-free control of bl...	2023	IFAC-Pap...	Co	
24	0			Baltoun...	NucEnvOB: A Database ...	2023	Membranes	Ar	
8	7			Nikolopo...	Demyelinating Syndro...	2022	Frontiers ...	Ar	
14	4			Ngo A.T.P.	Building a better NET: ...	2022	Research ...	Re	
15	3			Wassenh...	Comromise Planning: ...	2022	Compro...	Re	

Εικόνα 4.6 Πίνακας αποτελεσμάτων αναζήτησης

4.5 Πίνακας ιστορικού αναζητήσεων

Εφόσον ο χρήστης πραγματοποιήσει μια αναζήτηση με ένα συνδυασμό από λέξεις κλειδιά από τα πεδία της μπάρας αναζήτησης, οι εισερχόμενες πληροφορίες του αιτήματος θα αποθηκευτούν στο LocalStorage του browser του χρήστη. Ο χρήστης μπορεί να δει το ιστορικό του ανά πάσα στιγμή και να διαλέξει μια προηγούμενη αναζήτηση που έχει κάνει. Τα δεδομένα θα φορτωθούν στον πίνακα των αποτελεσμάτων όπως ακριβώς ήταν, όταν πρώτο έκανε την αναζήτηση. Ο πίνακας αποτελείται από τέσσερις στήλες. Η πρώτη στήλη περιέχει τον όρο της αναζήτησης που έκανε ο χρήστης και οι όροι χωρίζονται με το πρόθεμα '+' που δείχνει τον συνδυασμό των όρων. Για παράδειγμα αν ο χρήστης έγραψε στο πεδίο Author και στο πεδίο Affiliation τότε θα εμφανιστεί με την μορφή 'Author+Affiliation'. Στην δεύτερη στήλη φαίνεται η βάση που έγινε η αναζήτηση, που μπορεί να είναι το Crossref, το Scopus, το Semantic Scholar ή το OpenAlex. Στην τρίτη στήλη φαίνεται ο αριθμός των αποτελεσμάτων που βρέθηκαν από την αναζήτηση και τέλος, στην τελευταία στήλη εμφανίζεται η ημερομηνία που έγινε η αναζήτηση.

Search Term	Source	Papers	Search Date
Sidiropoulos+	 Scopus	48578	18-01-2024

Εικόνα 4.7 Πίνακας ιστορικού αναζητήσεων

Key	Value
search_history	[{"searchTerm": "Antonis Sidiropoulos+", "source": "Crossref", "papers": 11582, "sea...}
	<pre> [,...] 0: {searchTerm: "Antonis Sidiropoulos+", source: "Crossref", papers: 11582, searchDate: "18/01/2024",...} input_data: {author: "Antonis Sidiropoulos", affiliation: "", publication: "", title: "", keyword: ""} affiliation: "" author: "Antonis Sidiropoulos" keyword: "" publication: "" title: "" papers: 11582 recordData: [{id: "10.1007/s10514-018-9705-x", cites: 8, perYear: 39, rank: 23.501633,...},...] searchDate: "18/01/2024" searchTerm: "Antonis Sidiropoulos+" source: "Crossref" 1: {searchTerm: "Antonis Sidiropoulos+", source: "Scopus", papers: "37", searchDate: "18/01/2024",...} 2: {searchTerm: "Test+", source: "SemanticScholar", papers: 12319723, searchDate: "18/01/2024",...} 3: {searchTerm: "Internation Helenic University+", source: "OpenAlex", searchDate: "18/01/2024",...} </pre>

Εικόνα 4.8 Προβολή αποθηκευμένων δεδομένων στο Local Storage του browser

4.6 Πεδία αναζήτησης Crossref

Στο πτυσσόμενο μενού των πεδίων αναζήτησης για το Crossref, παρέχονται πέντε διαφορετικά πεδία. Αρχικά, το πρώτο πεδίο είναι το πεδίο των 'Συγγραφέων' (Authors), όπου ο χρήστης μπορεί να πραγματοποιήσει αναζήτηση βάσει ονομάτων ή ονομάτων συγγραφέων. Στη συνέχεια, υπάρχει το πεδίο 'Συνεργασία' (Affiliation), το οποίο αναφέρεται στο θεσμικό ή οργανωτικό περιβάλλον στο οποίο ανήκει ένα άτομο, συνήθως ένας συγγραφέας ή ερευνητής. Συνήθως, περιλαμβάνει το όνομα του πανεπιστημίου, του ερευνητικού ιδρύματος ή του οργανισμού στον οποίο ανήκει. Έπειτα, υπάρχει το πεδίο 'Όνομα Δημοσίευσης' (Publication Name), που εμφανίζει το όνομα του περιοδικού, της εφημερίδας, του βιβλίου ή άλλου εκδοτικού οίκου στον οποίο δημοσιεύτηκε το έργο. Στη συνέχεια, υπάρχει το πεδίο 'Λέξεις Τίτλου' (Title Words), που περιέχει τις λέξεις που εμφανίζονται στον τίτλο ενός έργου ή ενός άρθρου, και τέλος το πεδίο 'Λέξεις-Κλειδιά' (Keywords), που μπορεί να περιλαμβάνει λέξεις ή φράσεις που περιγράφουν το περιεχόμενο ή το θέμα του έργου.

Κάτω από αυτά τα πεδία αναζήτησης, υπάρχουν δύο κουμπιά ενεργειών: το κουμπί αναζήτησης, το οποίο ο χρήστης μπορεί να χρησιμοποιήσει για να πραγματοποιήσει την αναζήτηση, αφού έχει συμπληρώσει όλα τα πεδία που επιθυμεί και το κουμπί 'Καθαρισμός', το οποίο καθαρίζει όλα τα πεδία αναζήτησης αλλά και τον πίνακα αποτελεσμάτων, για να επιτρέψει στον χρήστη να ξεκινήσει μια νέα αναζήτηση από την αρχή.

Εικόνα 4.9 Πεδία αναζήτησης του Crossref

4.7 Πεδία αναζήτησης Scopus

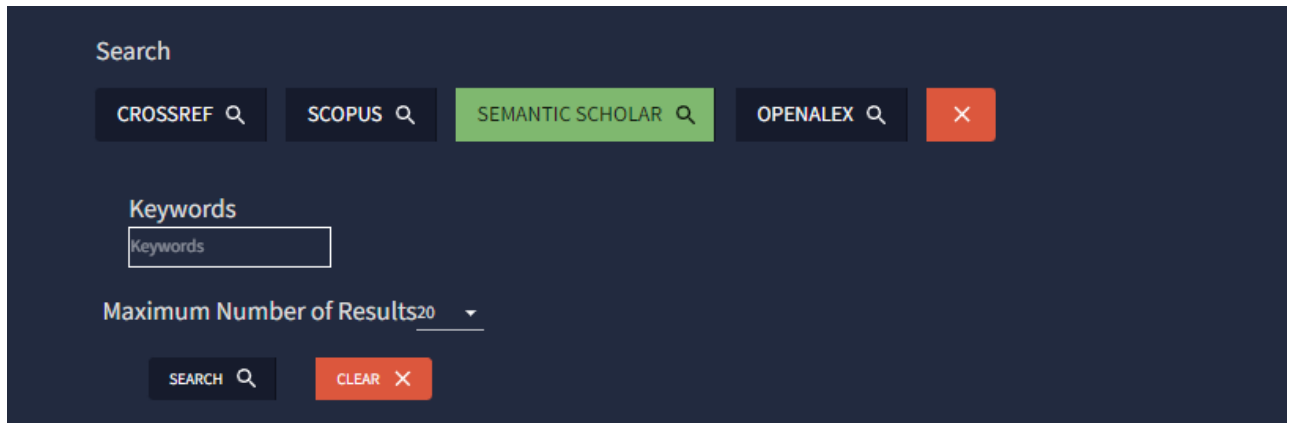
Παρόμοια με το Crossref, το Scopus παρέχει και αυτό πεδία αναζήτησης για τους χρήστες του, συμπεριλαμβανομένων των πεδίων 'Authors', 'Affiliation', 'Publication Name', 'Title Words' και 'Keywords'. Για να επιτραπεί στον χρήστη η αναζήτηση στη βάση δεδομένων του Scopus, απαιτείται η δημιουργία ενός λογαριασμού στην ιστοσελίδα του elsevier.com και η απόκτηση ενός προσωπικού κλειδιού API. Αυτό το κλειδί μπορεί να προστεθεί στην εφαρμογή μέσω της καρτέλας των ρυθμίσεων, επιτρέποντας έτσι στους χρήστες να απολαμβάνουν πρόσβαση στις λειτουργίες αναζήτησης του Scopus.

Εικόνα 4.10 Πεδία αναζήτησης του Scopus

4.8 Πεδία αναζήτησης Semantic Scholar

Στην καρτέλα των πεδίων του Semantic Scholar υπάρχει μόνο ένα πεδίο αναζήτησης. Το πεδίο αυτό είναι το πεδίο 'Keywords' και σε αυτό ο χρήστης μπορεί να πληκτρολογήσει τις λέξεις-κλειδιά ή το θέμα που τον ενδιαφέρει να αναζητήσει. Αυτό συνοδεύεται από ένα μενού με τον αριθμό των συνολικών

αποτελεσμάτων που επιθυμεί ο χρήστης να του επιστρέψει. Το μενού αυτό έχει τις τιμές 10, 20, 50, 100, 200, 500 και 1000.



Εικόνα 4.11 Πεδία αναζήτησης του Semantic Scholar

4.9 Πεδία αναζήτησης Open Alex

Στην καρτέλα των πεδίων του Open Alex έχουμε τέσσερα πεδία, το πρώτο πεδίο είναι το πεδίο ‘Author ID / Orcid’ που αναφέρεται στο μοναδικό αναγνωριστικό του συγγραφέα ή του ερευνητή στο σύστημα ORCID (Open Researcher and Contributor ID). Το ORCID είναι μια διεθνής κοινότητα που προσφέρει ένα μοναδικό αναγνωριστικό για κάθε ερευνητή, το οποίο μπορεί να χρησιμοποιηθεί για την ταυτοποίηση και την αναφορά της έρευνας τους. Στην συνέχεια υπάρχει το πεδίο ‘Affiliation ID’ που αναφέρεται στο μοναδικό αναγνωριστικό της συνδρομής ή του θεσμικού φορέα στον οποίο είναι συνδεδεμένος ένας συγγραφέας ή ένας ερευνητής. Το τρίτο πεδίο είναι πεδίο ‘Title Words’ που αναφέρεται στις λέξεις που εμφανίζονται στον τίτλο ενός άρθρου ή μιας δημοσίευσης και τέλος υπάρχει το πεδίο ‘Keywords’ που είναι λέξεις ή φράσεις που χαρακτηρίζουν το περιεχόμενο ή το θέμα μιας δημοσίευσης και χρησιμοποιούνται για αναζήτηση ή κατάταξη.

Εικόνα 4.12 Πεδία αναζήτησης του OpenAlex

4.10 Σελίδα πλήρης ιστορικού

Στο μενού στην αριστερή πλευρά της εφαρμογής, υπάρχει η επιλογή του ιστορικού (History). Εκεί, ο χρήστης μπορεί να δει τον πίνακα με τις προηγούμενες αναζητήσεις και του δίνεται η δυνατότητα να εξάγει αυτόν τον πίνακα σε μορφή CSV μαζί με τα αποτελέσματά τους ή να τον εκτυπώσει. Ο πίνακας είναι παρόμοιος με αυτόν που βρίσκεται στην αρχική οθόνη της εφαρμογής. Κάτω από τον πίνακα υπάρχει ένα κουμπί εκκαθάρισης που διαγράφει όλο το αποθηκευμένο ιστορικό από τον τοπικό αποθηκευτικό χώρο (Local Storage) του προγράμματος περιήγησης (browser) του χρήστη. Κάθε στήλη του πίνακα μπορεί να ταξινομηθεί, να φιλτραρισθεί ή να κρυφτεί αν ο χρήστης δεν επιθυμεί να φαίνεται.

Pub Search

Home
History
Settings

HISTORY
History Results

History

EXPORT

<input type="checkbox"/>	Search Term	Source	Papers	Search Date
<input type="checkbox"/>	Antonis Sidiropoulos+	Crossref	11582	18/01/2024
<input type="checkbox"/>	Antonis Sidiropoulos+	Scopus	37	18/01/2024
<input type="checkbox"/>	Test+	SemanticScholar	12319723	18/01/2024
<input type="checkbox"/>	Internation Helenic Uni...	OpenAlex		18/01/2024

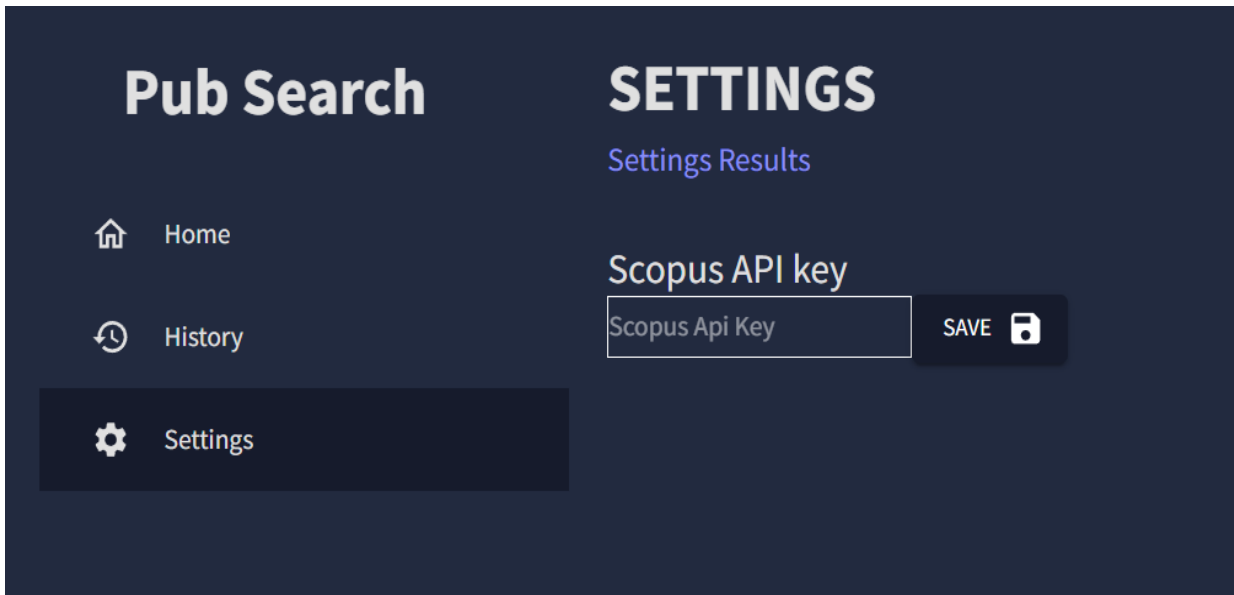
1-4 of 4

CLEAR ✕

Εικόνα 4.13 Πίνακας πλήρως ιστορικού αναζητήσεων

4.11 Σελίδα Ρυθμίσεων

Στο μενού των ρυθμίσεων, υπάρχει η επιλογή για την προσθήκη του Scopus API key από τη σελίδα του elsevier.com. Αφού ο χρήστης αποκτήσει ένα API key από την πλατφόρμα της Elsevier, μπορεί να το προσθέσει στο πεδίο που παρέχεται. Αυτό το κλειδί επιτρέπει στην εφαρμογή να επικοινωνεί με τη βάση δεδομένων του Scopus και να πραγματοποιεί αναζητήσεις σχετικά με επιστημονικές δημοσιεύσεις, έρευνες και άλλες πληροφορίες.



Εικόνα 4.14 Σελίδα ρυθμίσεων εφαρμογής

Key	Value
scopusApiKey	"1234567890"
1	"1234567890"

Εικόνα 4.15 Προβολή αποθηκευμένων κλειδιού Scopus στο Local Storage του browser

4.12 Επίλογος

Στο κεφάλαιο αυτό παρουσιάστηκαν όλες οι οθόνες και οι λειτουργίες της εφαρμογής και ο τρόπος με τον οποίο μπορεί να γίνει η χρήση τους από τον χρήστη.

Κεφάλαιο 5ο: Σύνοψη

5.1 Σύνοψη

Η παρούσα Δ.Ε. είχε σκοπό την ανάπτυξη μιας διαδικτυακής εφαρμογής για όλες τις έξυπνες συσκευές που μπορούν να χρησιμοποιήσουν ένα πρόγραμμα περιήγησης με σκοπό την χρήση της οποιαδήποτε στιγμή. Η εφαρμογή δίνει στους χρήστες την δυνατότητα να αναζητούν και να βλέπουν τις αναφορές που έχουν σχετικές επιστημονικές εργασίες που μπορεί να τους ενδιαφέρει, αλλά ακόμα και για συγγραφείς που μπορεί να έχουν δημοσιοποιήσει μια εργασία. Έτσι, ελέγχουν τις συνεισφορές που έχουν σε διάφορα έργα. Επίσης, ο κάθε χρήστης έχει το δικό του προσωπικό ιστορικό με την χρήση του Local Storage API, με το μόνο μειονέκτημα να μπορεί να το βλέπει στην συσκευή που είναι καθώς δεν υπάρχει προς το παρόν κάποιο σύστημα χρηστών για να αποθηκεύονται σε βάση δεδομένων. Ακόμη, η εφαρμογή δίνει την δυνατότητα στο χρήστη να μπορεί να χρησιμοποιήσει μια από τις τέσσερις μεγάλες βάσεις βιβλιογραφικών δεδομένων (Scopus, Crossref, Semantic Scholar και Open Alex) για να κάνει τις αναζητήσεις του.

5.2 Προτάσεις Βελτίωσης

Για τη δημιουργία της εφαρμογής χρησιμοποιήθηκε η βιβλιοθήκη React της JavaScript, η οποία λειτουργεί αποκλειστικά στην πλευρά του πελάτη (Client) χωρίς τη χρήση κάποιας τεχνολογίας Backend. Ως αποτέλεσμα, δεν υπάρχει κάποια βάση δεδομένων που να υποστηρίζει την εφαρμογή για την αποθήκευση δεδομένων χρήστη και την πρόσβασή τους από διαφορετικές συσκευές. Επομένως, μια πιθανή βελτίωση για το μέλλον θα μπορούσε να περιλαμβάνει την προσθήκη ενός συστήματος χρηστών με αυθεντικοποίηση, επιτρέποντας στους χρήστες να αποθηκεύουν τα δεδομένα τους και να τα διαμοιράζονται σε διάφορες συσκευές. Επιπλέον, θα μπορούσαν να προστεθούν επιπλέον βάσεις δεδομένων, όπως για παράδειγμα το PubMed, προκειμένου να επεκταθεί η λειτουργικότητα και η ποικιλία των διαθέσιμων πληροφοριών. Τέλος, μια άλλη πρόταση βελτίωσης θα μπορούσε να είναι η σύνδεση της εφαρμογής με το Scopus ID ή το ORCID του χρήστη, επιτρέποντας την εμφάνιση δεδομένων για τις δικές του δημοσιεύσεις και δίνοντας τη δυνατότητα να διακρίνεται η συμμετοχή του σε αναζητήσεις και αποτελέσματα.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Σώζων Παπαβλασόπουλος, *Βιβλιομετρία*. 2015.
- [2] M. K. McBurney and P. L. Novak, “What is bibliometrics and why should you care?,” in *Proceedings. IEEE International Professional Communication Conference*, IEEE, pp. 108–114. doi: 10.1109/IPCC.2002.1049094.
- [3] Sveta Cherednichenko, “Sveta Cherednichenko.” Accessed: Dec. 15, 2023. [Online]. Available: <https://www.mobindustry.net/blog/what-are-constraints-in-a-software-development-project/>
- [4] Anne-Wil Harzing, *The Publish or Perish Book*. 2010.
- [5] “Importing and Exporting Components.” Accessed: Dec. 18, 2023. [Online]. Available: <https://react.dev/learn/importing-and-exporting-components>
- [6] Randy Connolly and Ricardo Hoar, *Προγραμματισμός για το Web*. Μ. Γκιούρδας, 2015.
- [7] “Inheritance.” Accessed: Dec. 26, 2023. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/CSS/Inheritance>
- [8] Larry Ullman, *Modern JavaScript: Develop and Design*, 1st ed. Peachpit Press, 2012.
- [9] Abhishek Kothari, “12 Best JavaScript (JS) Frameworks You Should Know.” Accessed: Mar. 01, 2024. [Online]. Available: <https://geekflare.com/best-javascript-frameworks/#geekflare-toc-angularjs>
- [10] “What is Angular?” Accessed: Mar. 01, 2024. [Online]. Available: <https://angular.io/guide/what-is-angular>
- [11] “Introduction.” Accessed: Mar. 01, 2024. [Online]. Available: <https://vuejs.org/guide/introduction.html>
- [12] Sanchit Aggarwal, “Modern Web-Development using ReactJS,” *International Journal of Recent Research Aspects*, vol. 5, no. 1, pp. 133–137, Mar. 2018.
- [13] Prateek Rawat and Archana N. Mahajan, “ReactJS: A Modern Web Development Framework,” *Int J Innov Sci Res Technol*, vol. 5, no. 11, pp. 698–702, Nov. 2020.
- [14] Anurag Kumar and Ravi Kumar Singh, “Comparative analysis of angularjs and reactjs,” *International Journal of Latest Trends in Engineering and Technology*, vol. 7, no. 4, 2016, doi: 10.21172/1.74.030.
- [15] “Introduction to the DOM.” Accessed: Jan. 10, 2024. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction
- [16] “React Components.” Accessed: Jan. 10, 2024. [Online]. Available: https://www.w3schools.com/react/react_components.asp
- [17] “Material UI - Overview.” Accessed: Oct. 18, 2024. [Online]. Available: <https://mui.com/material-ui/getting-started/>
- [18] “How to Use localStorage with React Hooks to Set and Get Items.” Accessed: Mar. 19, 2023. [Online]. Available: <https://www.freecodecamp.org/news/how-to-use-localstorage-with-react-hooks-to-set-and-get-items/>

- [19] “About npm.” Accessed: Mar. 16, 2023. [Online]. Available: <https://docs.npmjs.com/about-npm>
- [20] “React Developer Tools.” Accessed: Jan. 10, 2024. [Online]. Available: <https://react.dev/learn/react-developer-tools>
- [21] “Documentation for Visual Studio Code.” Accessed: Feb. 16, 2023. [Online]. Available: <https://code.visualstudio.com/docs>
- [22] “Introduction to Postman for API Development.” Accessed: Apr. 23, 2023. [Online]. Available: <https://www.geeksforgeeks.org/introduction-postman-api-development/>
- [23] Á. Borrego, J. Ardanuy, and L. Arguimbau, “Crossref as a bibliographic discovery tool in the arts and humanities,” *Quantitative Science Studies*, vol. 4, no. 1, pp. 91–104, Mar. 2023, doi: 10.1162/qss_a_00240.
- [24] N. K. Patra, “From Electronic Resources to Electronic Resource Management,” in *Digital Disruption and Electronic Resource Management in Libraries*, Elsevier, 2017, pp. 13–25. doi: 10.1016/B978-0-08-102045-6.00002-9.
- [25] R. Lammey, “Using the Crossref Metadata API to explore publisher content,” *Science Editing*, vol. 3, no. 2, pp. 109–111, Aug. 2016, doi: 10.6087/kcse.75.
- [26] “Crossref Unified Resource API.” Accessed: Mar. 22, 2023. [Online]. Available: <https://api.staging.crossref.org/swagger-ui/index.html>
- [27] V. K. Singh, P. Singh, M. Karmakar, J. Leta, and P. Mayr, “The journal coverage of Web of Science, Scopus and Dimensions: A comparative analysis,” *Scientometrics*, vol. 126, no. 6, pp. 5113–5142, Jun. 2021, doi: 10.1007/s11192-021-03948-5.
- [28] J. F. Burnham, “Scopus database: a review,” *Biomed Digit Libr*, vol. 3, no. 1, p. 1, Dec. 2006, doi: 10.1186/1742-5581-3-1.
- [29] “Scopus for Government Research.” Accessed: Dec. 09, 2023. [Online]. Available: <https://www.elsevier.com/products/scopus/government-research>
- [30] “How to Guides.” Accessed: Jan. 12, 2023. [Online]. Available: https://dev.elsevier.com/technical_documentation.html
- [31] “Overview - OpenAlex technical documentation.” Accessed: Dec. 25, 2023. [Online]. Available: <https://docs.openalex.org/>
- [32] R. Kinney *et al.*, “The Semantic Scholar Open Data Platform,” Jan. 2023.
- [33] “Semantic Scholar.” Accessed: Jan. 16, 2023. [Online]. Available: <https://www.semanticscholar.org>

ΠΑΡΑΡΤΗΜΑ Α : Οδηγός Εγκατάστασης React js

Παρακάτω ακολουθεί ένας σύντομος οδηγός για την εγκατάσταση της React:

1. Εγκατάσταση του Node.js: Η React χρησιμοποιεί το Node.js για τη διαχείριση πακέτων και την εκτέλεση του κώδικα. Μπορείτε να κατεβάσετε και να εγκαταστήσετε το Node.js από τον επίσημο ιστότοπο του: <https://nodejs.org>.
2. Δημιουργία νέου έργου (Project): Ανοίξτε ένα τερματικό (command prompt) και δημιουργήστε ένα νέο φάκελο για το έργο σας. Μεταβείτε σε αυτό το φάκελο και εκτελέστε την ακόλουθη εντολή για να αρχικοποιήσετε ένα νέο έργο React:

```
npx create-react-app my-react-app
```

Αυτή η εντολή θα δημιουργήσει έναν νέο κατάλογο με το όνομα "my-react-app" και θα εγκαταστήσει όλα τα απαραίτητα πακέτα για να ξεκινήσετε ένα έργο React.

3. Εκτέλεση του React App: Μεταβείτε στον κατάλογο του νέου έργου σας (π.χ., `cd my-react-app`) και εκτελέστε την ακόλουθη εντολή για να ξεκινήσετε το React app:

```
npm start
```

Αυτή η εντολή θα ξεκινήσει έναν τοπικό server και θα ανοίξει το React app σας στον προεπιλεγμένο φυλλομετρητή σας.

4. Επεξεργασία του React App: Το React app σας τώρα είναι έτοιμο να το επεξεργαστείτε. Στον φάκελο `src` περιέχει τον πηγαίο κώδικα της εφαρμογής. Ανοίξτε αυτόν τον φάκελο στον πρόγραμμα επεξεργασίας κώδικα της επιλογής σας και αρχίστε να τροποποιείτε τα αρχεία όπως επιθυμείτε.

Αυτός είναι ένας βασικός οδηγός για την εγκατάσταση της React.

ΠΑΡΑΡΤΗΜΑ Β : Οδηγός Εγκατάστασης Visual Studio Code

Το Visual Studio Code είναι ένα δημοφιλές περιβάλλον ανάπτυξης (IDE) που παρέχει πλούσια λειτουργικότητα για πολλές γλώσσες προγραμματισμού. Ακολουθούν οι βασικές οδηγίες για την εγκατάστασή του:

Βήμα 1: Λήψη του Visual Studio Code

1. Πηγαίνετε στην επίσημη ιστοσελίδα του Visual Studio Code: <https://code.visualstudio.com/>
2. Κάντε κλικ στο κουμπί "Λήψη" για να λάβετε το πρόγραμμα εγκατάστασης.

Βήμα 2: Εγκατάσταση του Visual Studio Code

1. Εκτελέστε το πρόγραμμα εγκατάστασης που κατεβάσατε.
2. Ακολουθήστε τις οδηγίες στην οθόνη για να ολοκληρώσετε την εγκατάσταση.
3. Αφού η εγκατάσταση ολοκληρωθεί, εκκινήστε το Visual Studio Code.

Βήμα 3: Ρύθμιση του Visual Studio Code (προαιρετικό)

1. Μπορείτε να προσαρμόσετε τις ρυθμίσεις σύμφωνα με τις προτιμήσεις σας. Για παράδειγμα, μπορείτε να αλλάξετε το θέμα, τις γραμματοσειρές, τα πρόσθετα κ.λπ.

Βήμα 4: Εγκατάσταση πρόσθετων (προαιρετικό)

1. Το Visual Studio Code υποστηρίζει πολλά πρόσθετα που μπορούν να βελτιώσουν την εμπειρία ανάπτυξής σας. Μπορείτε να εγκαταστήσετε πρόσθετα από το Visual Studio Code Marketplace (<https://marketplace.visualstudio.com/>).
2. Ανοίξτε το Visual Studio Code και επιλέξτε το μενού "Πρόσθετα" από τη γραμμή μενού.
3. Αναζητήστε τα πρόσθετα που θέλετε να εγκαταστήσετε και κάντε κλικ στο κουμπί "Εγκατάσταση".