



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Πληροφορική»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Ιστοσελίδα καταγραφής δεδομένων πανδημίας, υλοποιημένη με mvc αρχιτεκτονική και spring boot framework Website for pandemic data recording, implemented with the MVC architecture and the Spring Boot framework
Όνοματεπώνυμο Φοιτητή	Ιωάννης Καραγεώργος
Πατρώνυμο	Δημήτριος
Αριθμός Μητρώου	ΜΠΠΛ21023
Επιβλέπων	Ευθύμιος Αλέπης, Καθηγητής

Ημερομηνία Παράδοσης **Οκτώβριος 2024**

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

ΕΥΘΥΜΙΟΣ ΑΛΕΠΗΣ
ΚΑΘΗΓΗΤΗΣ

ΜΑΡΙΑ ΒΙΡΒΟΥ
ΚΑΘΗΓΗΤΡΙΑ

ΕΥΑΓΓΕΛΟΣ
ΣΑΚΚΟΠΟΥΛΟΣ
ΑΝΑΠΛΗΡΩΤΗΣ
ΚΑΘΗΓΗΤΗΣ

Περίληψη

Στην παρούσα εργασία θα γίνει συλλογή δεδομένων διαφόρων περιοχών του πλανήτη καθώς και συλλογή δεδομένων για την ασθένεια Covid-19 για αυτές τις περιοχές. Τα δεδομένα αυτά θα συλλεχθούν από την ιστοσελίδα ourworldindata.org και θα υπάρχει η δυνατότητα ανανέωσης τους μέσω του προγράμματος (application) που θα δημιουργηθεί. Τα δεδομένα αυτά θα αποθηκευτούν σε βάση δεδομένων (database). Επιπλέον θα δημιουργηθεί διακομιστής ιστού (web server), ο οποίος θα διακομίζει ιστοσελίδες (web pages) για την προβολή των δεδομένων που συλλέχθηκαν, την επεξεργασία τους καθώς και την διαγραφή τους. Ο διακομιστής εισόδου θα προστατεύει την βάση δεδομένων με σύστημα εισόδου (login system).

Για τη υλοποίηση της παρούσας εντοπίστηκε στο διαδίκτυο η ιστοσελίδα ourworldindata.org, η οποία δίνει την δυνατότητα στους χρήστες της να κατεβάσουν (download) δεδομένα για την ασθένεια Covid-19 για διάφορες περιοχές του κόσμου σε μορφή αρχείου. Η ιστοσελίδα ανανεώνει τα δεδομένα αυτά καθημερινά. Με χρήση γλώσσας προγραμματισμού (Java) τα δεδομένα αυτά θα κατέβουν, θα καθαριστούν και θα εισαχθούν σε βάση δεδομένων (SQL). Με χρήση γλώσσας προγραμματισμού (Java) θα δημιουργηθεί διακομιστής ιστού ο οποίος θα επικοινωνεί με την βάση δεδομένων ώστε να έχει πρόσβαση σε αυτά. Με χρήση γλώσσας προγραμματισμού (Javascript) θα δημιουργηθούν ιστοσελίδες, οι οποίες θα παρουσιάζουν τα δεδομένα της βάσης δεδομένων, θα τα επεξεργάζονται και θα τα διαγράφουν. Οι ιστοσελίδες αυτές θα διακομίζονται από τον διακομιστή ιστού, ώστε οι χρήστες του προγράμματος να έχουν πρόσβαση σε αυτές μέσω φυλλομετρητή ιστού (web browser). Για την προστασία των δεδομένων της βάσης δεδομένων θα δημιουργηθεί σύστημα εισόδου πρόσβασης στις ιστοσελίδες του διακομιστή ιστού. Για την δημιουργία του διακομιστή ιστού θα χρησιμοποιηθεί το αρχιτεκτονικό πρότυπο ανάπτυξης λογισμικού Μοντέλο – Προβολή – Ελεγκτής (Model-View-Controller), ενώ για την δημιουργία του συστήματος εισόδου, θα χρησιμοποιηθεί το τυποποιημένο πρότυπο ανταλλαγής πληροφοριών μεταξύ δυο μερών με μορφή αντικειμένου λεξικού JWT (JSON Web Token).

Η παρούσα εργασία έχει ως αποτέλεσμα την δημιουργία μιας εφαρμογής ιστού (web application) έτοιμης για την προσθήκη της σε οποιοδήποτε πάροχο υπολογιστικού νέφους (cloud), η οποία δίνει την δυνατότητα σε χρήστες από οποιοδήποτε μέρος του κόσμου να δημιουργήσουν λογαριασμό σε αυτήν και να έχουν πρόσβαση σε δεδομένα της ασθένειας Covid-19. Επιπλέον δίνει την δυνατότητα στους διαχειριστές (administrators) αυτής να μπορούν να επεξεργάζονται τα δεδομένα αυτά, να τα ανανεώνουν ή να τα διαγράφουν.

Οι ιστοσελίδες (.org) που ανήκουν σε μη κερδοσκοπικούς οργανισμούς, φιλανθρωπικές οργανώσεις και άλλα μη κερδοσκοπικά σχήματα παρέχουν πρόσβαση σε δεδομένα σε όλους τους χρήστες του διαδικτύου για διάφορα ζητήματα που αφορούν τον παγκόσμιο πληθυσμό, όπως η ασθένεια Covid-19, με αποτέλεσμα την δυνατότητα συλλογής τους και ανάλυσής τους από πανεπιστήμια, εταιρίες και ιδιώτες.

Παρά την παλαιότητα της γλώσσας προγραμματισμού Java και της αντικατάστασης της σε πολλές περιπτώσεις από μοντέρνες γλώσσες προγραμματισμού, όπως η Python ή η Javascript, η γλώσσα αυτή μπορεί να δημιουργήσει αξιόπιστες εφαρμογές ιστού, οι οποίες μπορούν να εγκατασταθούν σε περιβάλλοντα υπολογιστικού νέφους και να αλληλοεπιδράσουν με τεχνολογίες αιχμής όπως η βιβλιοθήκη React, της γλώσσας προγραμματισμού Javascript, η οποία χρησιμοποιείται για την κατασκευή διαδραστικών και δυναμικών διεπαφών χρήστη (UI) σε εφαρμογές ιστού.

Λέξεις - Κλειδιά: Συλλογή δεδομένων, διακομιστής ιστού, εφαρμογή ιστού, MVC, JWT, React, Java, Spring Boot.

Abstract

This thesis aims to collect data from various regions around the world and gather data on the Covid-19 disease for these regions. The data will be sourced from the website ourworldindata.org and will be updated through an application that will be developed. The data will be stored in a database. Additionally, a web server will be created to host web pages for displaying, processing, and deleting the collected data. The server will protect the database with a login system.

To implement this, the website ourworldindata.org was identified as a source that allows users to download Covid-19 data for various regions in file format. This data is updated daily. Using Java, the data will be downloaded, cleaned, and inserted into an SQL database. A web server will be created using Java to communicate with the database for data access. JavaScript will be used to develop web pages that display, process, and delete the database data. These web pages will be served by the web server, allowing users to access them through a web browser. To protect the database, a login system will be implemented. The web server will be built using the Model-View-Controller (MVC) architecture pattern, while the login system will use the JSON Web Token (JWT) standard for secure information exchange.

The result of this work is the creation of a web application ready to be deployed on any cloud service provider. This application allows users from anywhere in the world to create accounts and access Covid-19 data. Additionally, it enables administrators to manage this data by updating or deleting it.

Websites (.org) belonging to non-profit organizations and charities provide access to global data on issues like Covid-19, enabling collection and analysis by universities, companies, and individuals. Despite the age of Java and its replacement by more modern programming languages like Python or JavaScript in many cases, Java remains capable of creating reliable web applications. These applications can be deployed in cloud environments and interact with cutting-edge technologies like the React library used for building interactive and dynamic user interfaces in web applications.

Keywords: Data Collection, Web Server, Web Application, MVC, JWT, React, Java, Spring Boot.

Contents

10	Περίληψη	3
11	Abstract	3
1.	Εισαγωγή	7
1.1	Το πλαίσιο της Διπλωματικής Εργασίας	7
1.2	Σκοπός της Διπλωματικής Εργασίας	7
1.3	Αντικείμενο της Διπλωματικής Εργασίας	7
2.	Ανάλυση Τρέχοντος Επιπέδου	7
2.1	Βασικές Έννοιες και Ορισμοί	8
2.1.1	Ιστοσελίδα (Web page)	8
2.1.2	Web Browser (φυλλομετρητής ιστού – πλοηγός)	9
2.1.3	Web Server (διακομιστής ιστού)	11
2.1.4	Διεπαφή Προγραμματισμού Εφαρμογών (API)	12
2.1.5	JSON Web Token	14
2.1.6	Βάσεις Δεδομένων (Databases)	15
2.1.7	Structured Query Language (SQL)	16
2.1.8	Java	17
2.1.9	Spring Boot	18
2.1.10	HTML (Hypertext Markup Language)	19
2.1.11	CSS (Cascading Style Sheets)	20
2.1.12	Javascript	22
2.1.13	React.js	24
2.2	Απαιτήσεις Συστήματος	25
2.2.1	Model-View-Controller (MVC)	25
2.3	Διάγραμμα Δραστηριότητας	25
3.	Το Σύστημα	26
3.1	Υποφάκελος covidApi (Spring Boot)	27
3.1.1	Υποφάκελος model (Μοντέλο)	28
3.1.2	Υποφάκελος service	29

3.1.3	Υποφάκελος scheduled.....	29
3.1.4	Υποφάκελος controller (Ελεγκτής)	29
3.1.5	Υποφάκελος config - CORS.....	30
3.1.6	Υποφάκελος Dto	30
3.1.7	Υποφάκελος exceptions.....	31
3.1.8	Υποφάκελος filter	31
3.1.9	Υποφάκελος map.....	31
11.1.1	Υποφάκελος reader	32
3.1.10	Υποφάκελος repository	32
3.2	Υποφάκελος covidapifrontend (React.js)	32
3.2.1	Υποφάκελος public	33
3.2.2	Υποφάκελος node_modules	33
3.2.3	Αρχείο package.json	34
3.2.4	Υποφάκελος components	34
3.2.5	Υποφάκελος hooks	42
3.2.6	Αρχείο App.js και δρομολόγηση (Routing)	42
4.	Συμπεράσματα και Προοπτικές.....	43
4.1	Συμπεράσματα επί των αποτελεσμάτων	43
4.2	Συμπερασματα επι των εργαλειων	43
4.3	Προοπτικές / Επομενα βηματα	43
5.	Βιβλιογραφία	44

1. Εισαγωγή

1.1 Το πλαίσιο της Διπλωματικής Εργασίας

Η ευρεία χρήση του διαδικτύου έχει οδηγήσει μη κερδοσκοπικούς οργανισμούς, κοινωφελείς οργανισμούς, ιδρύματα, εκπαιδευτικά ιδρύματα, ή άλλους τύπους οργανισμών που δραστηριοποιούνται για το κοινό καλό ή παρέχουν πληροφορίες και υπηρεσίες να δημιουργήσουν ιστοσελίδες που δίνουν δωρεάν δεδομένα.

Μια από αυτές τις ιστοσελίδες είναι η ourworldindata.org, η οποία είναι μια μη κερδοσκοπική ιστοσελίδα που παρέχει ανοιχτά δεδομένα και αναλύσεις για παγκόσμια ζητήματα. Στόχος της είναι να προσφέρει πληροφορίες που μπορούν να χρησιμοποιηθούν για την κατανόηση και τη βελτίωση της κατάστασης του κόσμου, με βάση δεδομένα και τεκμηρίωση (ourworldindata_wiki, 2013).

Η ιστοσελίδα καλύπτει ένα ευρύ φάσμα θεμάτων, ένα εκ των οποίων είναι η παροχή δεδομένων για την ασθένεια Covid-19 σε μορφή αρχείου.

1.2 Σκοπός της Διπλωματικής Εργασίας

Η παρούσα εργασία έχει ως στόχο την συλλογή δεδομένων από την ιστοσελίδα ourworldindata.org για την ασθένεια Covid-19, και την παρουσίαση αυτών μέσω εφαρμογής ιστού. Η εφαρμογή ιστού πρέπει να προστατεύει τα δεδομένα με χρήση συστήματος εισόδου πρόσβασης, το οποίο έχει δυο ρόλους χρηστών, τους απλούς χρήστες και τους διαχειριστές του συστήματος. Οι διαχειριστές του συστήματος μπορούν να δουν, να επεξεργαστούν και να διαγράψουν τα δεδομένα, ενώ οι απλοί χρήστες μπορούν μόνο να τα δουν.

1.3 Αντικείμενο της Διπλωματικής Εργασίας

Για τη ολοκλήρωση της παρούσας εργασίας πρέπει να υλοποιηθούν τα παρακάτω βήματα:

- 1) Κατέβασμα δεδομένων για την ασθένεια Covid-19 από την ιστοσελίδα ourworldindata.org και αποθήκευση αυτών σε αρχείο (.csv)
- 2) Εισαγωγή αυτών των δεδομένων σε βάση δεδομένων (SQL).
- 3) Δημιουργία προγράμματος για την ανανέωση των δεδομένων της ασθένειας Covid-19 στην βάση δεδομένων.
- 4) Δημιουργία Διεπαφής Προγραμματισμού Εφαρμογών (API) για την αλληλεπίδραση με την βάση δεδομένων μέσω δημιουργίας διακομιστή ιστού.
- 5) Δημιουργία συστήματος εισόδου πρόσβασης της Διεπαφής Προγραμματισμού Εφαρμογών (API) για την προστασία των δεδομένων της βάσης με χρήση προτύπου JWT (JSON Web Token). Το σύστημα παρέχει δυο ρόλους χρηστών.
- 6) Δημιουργία ιστοσελίδων για την προβολή, επεξεργασία και διαγραφή δεδομένων ανάλογα με τον ρόλο του χρήστη που συνδέθηκε στον διακομιστή ιστού.
- 7) Προσθήκη των ιστοσελίδων στο διακομιστή ιστού.

2. Ανάλυση Τρέχοντος Επιπέδου

2.1 Βασικές Έννοιες και Ορισμοί

Για την δημιουργία, καθώς και για την κατανόηση της παρούσας εργασίας απαιτείται γνώση της έννοιας της ιστοσελίδας (web page), της έννοιας του πλοηγού (web browser), της έννοιας του διακομιστή ιστού (web server), της έννοιας της Διεπαφής Προγραμματισμού Εφαρμογών (API), του προτύπου JWT (JSON Web Token), της έννοιας των βάσεων δεδομένων, την γνώση της γλώσσας SQL για την διαχείριση των βάσεων δεδομένων, της γλώσσας προγραμματισμού Java, του πλαισίου ανάπτυξης εφαρμογών (Framework) αυτής Spring Boot, του Html, του CSS, της γλώσσας προγραμματισμού JavaScript, της βιβλιοθήκης δημιουργίας ιστοσελίδων αυτής που ονομάζεται React.

2.1.1 Ιστοσελίδα (Web page)

Η ιστοσελίδα είναι μια συλλογή σχετικών πληροφοριών που δημοσιεύονται στο διαδίκτυο και είναι προσβάσιμες μέσω ενός ιστότοπου. Στην ουσία, είναι ένα σύνολο συνδεδεμένων ηλεκτρονικών εγγράφων που μπορούν να περιλαμβάνουν κείμενο, εικόνες, βίντεο και άλλα πολυμέσα. Αυτά τα έγγραφα είναι συνήθως γραμμένα σε γλώσσα HTML (Hypertext Markup Language) και μπορούν να συνδεθούν μεταξύ τους με συνδέσμους (links).

Μια ιστοσελίδα συνήθως ανήκει σε έναν ιστότοπο, ο οποίος μπορεί να περιλαμβάνει πολλές διαφορετικές σελίδες, κατηγορίες, ή τμήματα. Οι ιστοσελίδες μπορούν να χρησιμοποιούνται για ποικίλους σκοπούς, όπως ενημέρωση, ψυχαγωγία, εκπαίδευση, εμπορική προώθηση και κοινωνική δικτύωση.

Ιστορικά δεδομένα για τις ιστοσελίδες:

1. Δημιουργία του World Wide Web (WWW):

- Το World Wide Web δημιουργήθηκε από τον Tim Berners-Lee, έναν επιστήμονα της πληροφορικής στο CERN, το 1989. Το WWW επιτρέπει την πρόσβαση σε έγγραφα συνδεδεμένα μέσω υπερσυνδέσεων.
- Η πρώτη ιστοσελίδα δημιουργήθηκε από τον Tim Berners-Lee και τέθηκε σε λειτουργία στις 6 Αυγούστου 1991. Η διεύθυνση της πρώτης ιστοσελίδας ήταν <http://info.cern.ch>, και περιείχε πληροφορίες για το τι είναι το World Wide Web και πώς να δημιουργήσει κάποιος τη δική του ιστοσελίδα.

2. Πρώτο Web Browser και Web Server:

- Ο πρώτος web browser, ο οποίος λειτουργούσε και ως επεξεργαστής ιστοσελίδων, δημιουργήθηκε από τον Tim Berners-Lee και ονομαζόταν WorldWideWeb (μετέπειτα μετονομάστηκε σε Nexus).
- Ο πρώτος web server ονομάστηκε CERN HTTPd και χρησιμοποιήθηκε για να φιλοξενεί τις πρώτες ιστοσελίδες στο CERN.

3. Πρώτες Μηχανές Αναζήτησης και Κατάλογοι Ιστοσελίδων:

- Η πρώτη μηχανή αναζήτησης για το WWW ήταν το Archie, το οποίο αναπτύχθηκε το 1990 και ήταν ουσιαστικά μια βάση δεδομένων για αρχεία FTP.
- Η πρώτη πλήρης μηχανή αναζήτησης ιστού, το World Wide Web Wanderer, δημιουργήθηκε το 1993 από τον Matthew Gray.

4. Δημοφιλείς Πρώτες Ιστοσελίδες και Πύλες:

- **Yahoo!**: Ξεκίνησε το 1994 ως ένας κατάλογος ιστότοπων. Ήταν μία από τις πρώτες πύλες του Διαδικτύου.

- **Amazon:** Ιδρύθηκε το 1994 και έγινε μία από τις πρώτες μεγάλες ιστοσελίδες ηλεκτρονικού εμπορίου.
- **Google:** Ιδρύθηκε το 1998 και γρήγορα έγινε η πιο δημοφιλής μηχανή αναζήτησης στο διαδίκτυο.

5. Ανάπτυξη του Web 2.0:

- Το Web 2.0, που εμφανίστηκε γύρω στο 2004, αναφέρεται στην εξέλιξη των ιστοσελίδων προς πιο διαδραστικές και συνεργατικές πλατφόρμες, όπως τα κοινωνικά δίκτυα (Facebook, Twitter) και οι πλατφόρμες συνεργασίας (Wikipedia, YouTube).

6. Εξέλιξη της Τεχνολογίας:

- Οι τεχνολογίες που χρησιμοποιούνται για την ανάπτυξη ιστοσελίδων έχουν εξελιχθεί σημαντικά από την αρχή. Εκτός από το HTML, σήμερα χρησιμοποιούνται εκτενώς το CSS για τη διαμόρφωση της εμφάνισης και η JavaScript για την προσθήκη διαδραστικών στοιχείων.
- Οι μοντέρνες τεχνολογίες και frameworks, όπως το React, Angular και Vue.js, επιτρέπουν τη δημιουργία δυναμικών και ευέλικτων ιστοσελίδων και εφαρμογών ιστού.

Οι ιστοσελίδες έχουν εξελιχθεί από απλά κείμενα με υπερσυνδέσεις σε πολύπλοκες, διαδραστικές πλατφόρμες που επιτρέπουν τη συνεργασία, την κοινωνική δικτύωση, το ηλεκτρονικό εμπόριο και πολλά άλλα. Η συνεχής ανάπτυξη της τεχνολογίας συνεχίζει να διαμορφώνει το μέλλον των ιστοσελίδων και του διαδικτύου γενικότερα.

2.1.2 Web Browser (φυλλομετρητής ιστού – πλοηγός).

Ο Web browser (ή απλώς "πλοηγός ιστού") είναι ένα λογισμικό πρόγραμμα που επιτρέπει στους χρήστες να περιηγούνται στο διαδίκτυο. Ένας Web browser αναλαμβάνει την ανάκτηση, την ερμηνεία και την προβολή των ιστοσελίδων και των περιεχομένων τους, που είναι αποθηκευμένα σε διάφορους διακομιστές ιστού.

Οι κύριες λειτουργίες ενός Web browser περιλαμβάνουν:

1. Ανάκτηση Ιστοσελίδων: Στέλνει αιτήματα σε διακομιστές ιστού μέσω του πρωτοκόλλου HTTP ή HTTPS για να κατεβάσει δεδομένα ιστοσελίδων.
2. Ερμηνεία Κώδικα: Διερμηνεύει τον κώδικα HTML, CSS και JavaScript που περιέχει η ιστοσελίδα για να την εμφανίσει σωστά στον χρήστη.
3. Προβολή Περιεχομένου: Εμφανίζει τα κείμενα, τις εικόνες, τα βίντεο και άλλες πολυμεσικές μορφές που συνθέτουν την ιστοσελίδα.
4. Διαχείριση Συνδέσεων: Διαχειρίζεται τα συνδέσμους και τις διασυνδέσεις μεταξύ των σελίδων.
5. Προστασία και Ασφάλεια: Παρέχει διάφορες δυνατότητες ασφαλείας, όπως προστασία από κακόβουλο λογισμικό, διαχείριση απορρήτου και διαγραφή ιστορικού.

Κάποιοι από τους πιο δημοφιλείς Web browsers περιλαμβάνουν τον Google Chrome, τον Mozilla Firefox, τον Microsoft Edge και τον Safari.

Ιστορικά στοιχεία για τους Web Browsers:

1. WorldWideWeb (αργότερα Nexus) - 1990:

- Ο πρώτος web browser δημιουργήθηκε από τον Tim Berners-Lee το 1990 και ονομάστηκε WorldWideWeb. Αργότερα μετονομάστηκε σε Nexus για να αποφευχθεί σύγχυση με το ίδιο το World Wide Web.
- Ο WorldWideWeb λειτουργούσε και ως επεξεργαστής HTML, επιτρέποντας στους χρήστες να δημιουργούν και να επεξεργάζονται ιστοσελίδες.

2. Mosaic - 1993:

- Ο Mosaic, αναπτύχθηκε από το Εθνικό Κέντρο Εφαρμογών Υπερυπολογιστών (NCSA) το 1993 και ήταν ο πρώτος web browser που υποστήριζε γραφικά. Εισήγαγε τη δυνατότητα εμφάνισης εικόνων μέσα στο κείμενο των ιστοσελίδων, γεγονός που έκανε το διαδίκτυο πιο ελκυστικό και προσβάσιμο.
- Ο Mosaic συνέβαλε σημαντικά στη δημοτικότητα του παγκόσμιου ιστού και πολλοί από τους προγραμματιστές του συνέχισαν να δημιουργούν τον Netscape Navigator.

3. Netscape Navigator - 1994:

- Κυκλοφόρησε το 1994 και έγινε ο πιο δημοφιλής web browser της εποχής του. Ο Netscape έφερε καινοτομίες όπως τα cookies και τη δυνατότητα για plug-ins, που επέκτειναν τις δυνατότητες του browser.
- Η επιτυχία του Netscape οδήγησε στην ίδρυση της Netscape Communications Corporation, η οποία αργότερα κυκλοφόρησε τον Netscape Communicator, μια σουίτα διαδικτυακών εφαρμογών.

4. Internet Explorer - 1995:

- Αναπτύχθηκε από τη Microsoft και κυκλοφόρησε το 1995. Ήταν μέρος του λειτουργικού συστήματος Windows και σύντομα έγινε ο πιο χρησιμοποιούμενος browser, κυρίως λόγω της στρατηγικής ενσωμάτωσής του στα Windows.
- Η κυριαρχία του Internet Explorer οδήγησε στον πόλεμο των browsers της δεκαετίας του 1990 και στις αρχές της δεκαετίας του 2000, όταν οι χρήστες και οι εταιρείες αναζητούσαν εναλλακτικές λύσεις.

5. Mozilla Firefox - 2004:

- Αναπτύχθηκε από το Mozilla Foundation και κυκλοφόρησε το 2004 ως απάντηση στην αυξανόμενη δυσαρέσκεια με τον Internet Explorer. Ο Firefox προσέφερε βελτιωμένη ασφάλεια, ταχύτητα και επεκτασιμότητα μέσω των add-ons και των extensions.
- Ο Firefox ήταν ιδιαίτερα δημοφιλής μεταξύ των προγραμματιστών και των χρηστών που αναζητούσαν έναν εναλλακτικό και πιο ασφαλή browser.

6. Google Chrome - 2008:

- Κυκλοφόρησε από την Google το 2008 και γρήγορα έγινε ο πιο δημοφιλής web browser λόγω της ταχύτητας, της απλότητας και της ισχυρής υποστήριξης για σύγχρονες τεχνολογίες ιστού.
- Ο Chrome έφερε την έννοια των "tabs" στο προσκήνιο και υποστήριξε την ανάπτυξη του ανοικτού κώδικα έργου Chromium, που αποτελεί τη βάση για άλλους browsers όπως ο Microsoft Edge.

7. Microsoft Edge - 2015:

- Κυκλοφόρησε από τη Microsoft ως αντικαταστάτης του Internet Explorer. Ο αρχικός Edge χρησιμοποιούσε μια νέα μηχανή απόδοσης, αλλά το 2019 η

Microsoft αποφάσισε να υιοθετήσει το ανοιχτού κώδικα Chromium project ως βάση για τον Edge.

- Ο νέος Edge είναι συμβατός με τις επεκτάσεις του Chrome και προσφέρει βελτιωμένη απόδοση και ασφάλεια.

Οι web browsers έχουν εξελιχθεί σημαντικά από την πρώτη τους εμφάνιση και συνεχίζουν να αποτελούν ένα κρίσιμο εργαλείο για την πρόσβαση και την αλληλεπίδραση με το διαδίκτυο.

2.1.3 Web Server (διακομιστής ιστού)

Ο Web server (διακομιστής ιστού) είναι ένα λογισμικό ή ένα υλικό σύστημα που αποθηκεύει, επεξεργάζεται και παραδίδει περιεχόμενο μέσω του διαδικτύου. Η κύρια λειτουργία του είναι να απαντά σε αιτήματα που υποβάλλονται από Web browsers, παρέχοντας τα δεδομένα των ιστοσελίδων που ζητούνται.

Τα κύρια χαρακτηριστικά και λειτουργίες ενός Web server περιλαμβάνουν:

1. Αποθήκευση Περιεχομένου: Ο Web server φιλοξενεί αρχεία και δεδομένα ιστοσελίδων, όπως HTML, CSS, JavaScript, εικόνες και βίντεο.
2. Επεξεργασία Αιτημάτων: Όταν ένας χρήστης ζητήσει μια ιστοσελίδα μέσω του browser του, ο Web server επεξεργάζεται το αίτημα και ανακτά τα σχετικά αρχεία από τη βάση δεδομένων ή το σύστημα αρχείων του.
3. Αποστολή Περιεχομένου: Ο Web server στέλνει τα ζητούμενα αρχεία πίσω στον browser του χρήστη μέσω του πρωτοκόλλου HTTP ή HTTPS.
4. Διαχείριση Δεδομένων: Ο Web server μπορεί επίσης να διαχειρίζεται βάσεις δεδομένων και να εκτελεί server-side scripts για δυναμικό περιεχόμενο, όπως είναι οι εφαρμογές που απαιτούν επεξεργασία δεδομένων σε πραγματικό χρόνο.
5. Ασφάλεια: Παρέχει μηχανισμούς ασφάλειας, όπως πιστοποίηση και κρυπτογράφηση, για να προστατεύσει τα δεδομένα και την επικοινωνία μεταξύ του server και του client.

Κάποιοι από τους πιο δημοφιλείς Web servers είναι ο Apache HTTP Server, ο Nginx, ο Microsoft Internet Information Services (IIS) και ο LiteSpeed.

Ιστορικά στοιχεία για τους Web Servers:

1. Πρώτοι Web Servers:

- **CERN HTTPd:** Ο πρώτος web server δημιουργήθηκε από τον Tim Berners-Lee στο CERN το 1990. Ο CERN HTTPd ήταν ο διακομιστής που υποστήριξε την πρώτη ιστοσελίδα και χρησιμοποιήθηκε για να διανέμει περιεχόμενο στο αρχικό World Wide Web.
- **NCSA HTTPd:** Το 1993, το National Center for Supercomputing Applications (NCSA) ανέπτυξε το NCSA HTTPd, ένα από τα πρώτα ευρέως χρησιμοποιούμενα web servers. Ήταν η βάση για πολλές μελλοντικές εξελίξεις.

2. Εμφάνιση του Apache HTTP Server:

- **Apache HTTP Server:** Κυκλοφόρησε το 1995 και γρήγορα έγινε ο πιο δημοφιλής web server παγκοσμίως. Ο Apache είναι ένα open-source λογισμικό που παρέχει υψηλή απόδοση, σταθερότητα και ευελιξία, και έχει υποστήριξη

για διάφορα λειτουργικά συστήματα. Παραμένει ένας από τους πιο χρησιμοποιούμενους web servers μέχρι σήμερα.

3. Εμφάνιση του Microsoft Internet Information Services (IIS):

- **Microsoft IIS:** Το IIS κυκλοφόρησε το 1995 και ήταν ο web server που αναπτύχθηκε από τη Microsoft για τα λειτουργικά συστήματα Windows. Το IIS προσφέρει υποστήριξη για ASP.NET, PHP και άλλες τεχνολογίες ανάπτυξης ιστού.

4. Νέα Εξέλιξη και Εξυπηρετητές Υψηλής Απόδοσης:

- **Nginx:** Δημιουργήθηκε από τον Igor Sysoen και κυκλοφόρησε το 2004. Ο Nginx είναι γνωστός για την εξαιρετική του απόδοση και τη χαμηλή κατανάλωση πόρων, ειδικά για την εξυπηρέτηση μεγάλου αριθμού ταυτόχρονων συνδέσεων. Χρησιμοποιείται ευρέως για τη διαχείριση του HTTP traffic και ως reverse proxy server.

5. Εξέλιξη και Σύγχρονοι Web Servers:

- **LiteSpeed:** Κυκλοφόρησε το 2003 και προσφέρει βελτιωμένη απόδοση και ασφάλεια. Η εμπορική έκδοση του LiteSpeed παρέχει επίσης προηγμένα χαρακτηριστικά όπως HTTP/2 και QUIC.
- **Caddy:** Κυκλοφόρησε το 2015 και είναι γνωστός για τη εύκολη εγκατάσταση και την ενσωματωμένη υποστήριξη για HTTPS μέσω του Let's Encrypt.

Οι web servers είναι βασικά συστατικά της υποδομής του διαδικτύου, επιτρέποντας την αποθήκευση και την εξυπηρέτηση περιεχομένου ιστοσελίδων στους χρήστες. Από τις πρώτες υλοποιήσεις τους μέχρι τις σύγχρονες λύσεις, οι web servers έχουν εξελιχθεί για να υποστηρίζουν τις αυξανόμενες απαιτήσεις της τεχνολογίας του ιστού.

2.1.4 Διεπαφή Προγραμματισμού Εφαρμογών (API)

Το API (Application Programming Interface, Διεπαφή Προγραμματισμού Εφαρμογών) είναι ένα σύνολο κανόνων και πρωτοκόλλων που επιτρέπει σε διαφορετικά λογισμικά ή εφαρμογές να επικοινωνούν και να αλληλεπιδρούν μεταξύ τους. Το API καθορίζει τον τρόπο με τον οποίο οι διάφορες εφαρμογές μπορούν να ζητούν και να παρέχουν δεδομένα ή υπηρεσίες.

Συγκεκριμένα, το API περιλαμβάνει:

1. Μέθοδοι και Κλήσεις: Ορίζουν τις ενέργειες ή λειτουργίες που μπορεί να εκτελέσει το API. Αυτές οι μέθοδοι συνήθως αντιστοιχούν σε ενέργειες που μπορεί να εκτελέσει η εφαρμογή, όπως ανάκτηση δεδομένων, δημιουργία νέων εγγραφών ή ενημέρωση υπαρχόντων στοιχείων.
2. Δομή Δεδομένων: Καθορίζει πώς πρέπει να μορφοποιούνται τα δεδομένα που στέλνονται και λαμβάνονται μέσω του API. Αυτό περιλαμβάνει συνήθως τη μορφή δεδομένων (όπως JSON, XML) και τη δομή των αιτημάτων και των απαντήσεων.
3. Συμβόλαια: Περιγράφει τους κανόνες που πρέπει να τηρούνται για την αλληλεπίδραση με το API, όπως ποια είναι τα απαιτούμενα ή προαιρετικά δεδομένα και ποια είναι η αναμενόμενη μορφή της απάντησης.
4. Ασφάλεια: Ορίζει τους μηχανισμούς ασφαλείας που πρέπει να ακολουθούνται για να προστατευτούν οι πληροφορίες και οι υπηρεσίες, όπως η επαλήθευση ταυτότητας και η εξουσιοδότηση.

Τα APIs χρησιμοποιούνται για διάφορους σκοπούς, όπως η σύνδεση μεταξύ διαφορετικών υπηρεσιών διαδικτύου (π.χ., η ενσωμάτωσή τους με κοινωνικά δίκτυα), η αλληλεπίδραση με βάσεις δεδομένων, και η ανάπτυξη εφαρμογών που απαιτούν δεδομένα ή υπηρεσίες από τρίτους.

Για παράδειγμα, ένα API μπορεί να επιτρέπει σε μια εφαρμογή καιρού να ανακτά δεδομένα καιρικών συνθηκών από μια εξωτερική υπηρεσία και να τα εμφανίζει στους χρήστες της εφαρμογής.

Ιστορικά στοιχεία για το api:

1. Η Αρχή των API: 1940s-1960s

- Η ιδέα των API μπορεί να εντοπιστεί στις πρώτες ημέρες του προγραμματισμού. Κατά τη δεκαετία του 1940 και του 1950, όταν οι υπολογιστές και οι γλώσσες προγραμματισμού άρχισαν να εξελίσσονται, προέκυψε η ανάγκη για καθορισμένες διεπαφές που θα επέτρεπαν τη σύνδεση μεταξύ διαφορετικών κομματιών λογισμικού.
- Οι πρώτες μορφές API ήταν πολύ βασικές και χρησιμοποιήθηκαν κυρίως για την κλήση υπορουτινών (subroutines) και την επικοινωνία μεταξύ των προγραμμάτων μέσα σε ένα σύστημα.

2. Εξέλιξη και Διάδοση: 1960s-1980s

- Κατά τη δεκαετία του 1960 και του 1970, με την ανάπτυξη πιο εξελιγμένων λειτουργικών συστημάτων και γλωσσών προγραμματισμού, τα API άρχισαν να γίνονται πιο τυποποιημένα και σημαντικά. Το 1968, ο επικεφαλής μηχανικός της IBM, Fred Brooks, έκανε αναφορά στην έννοια του API στο βιβλίο του "The Mythical Man-Month".
- Κατά τη δεκαετία του 1980, τα API έγιναν πιο ευρέως χρησιμοποιούμενα με την άνοδο των λειτουργικών συστημάτων όπως τα Unix, τα οποία προσέφεραν τυποποιημένα API για την ανάπτυξη εφαρμογών.

3. Εξάπλωση στον Παγκόσμιο Ιστό: 1990s

- Με την έκρηξη του Διαδικτύου τη δεκαετία του 1990, τα API εξελίχθηκαν περαιτέρω για να υποστηρίξουν την επικοινωνία μεταξύ των web εφαρμογών. Ένας σημαντικός σταθμός ήταν η ανάπτυξη του **SOAP** (Simple Object Access Protocol) από τη Microsoft το 1998, το οποίο ήταν ένα από τα πρώτα πρωτόκολλα που χρησιμοποιήθηκε για την ανταλλαγή δομημένων πληροφοριών μέσω διαδικτύου.
- Τα API έγιναν σημαντικό εργαλείο για την ανάπτυξη και την ενσωμάτωση των web υπηρεσιών, επιτρέποντας την επικοινωνία μεταξύ διαφορετικών εφαρμογών και συστημάτων μέσω του Διαδικτύου.

4. Άνοδος του RESTful API: 2000s

- Στις αρχές του 2000, ο Roy Fielding δημοσίευσε τη διατριβή του για το REST (Representational State Transfer), το οποίο γρήγορα έγινε ο κυρίαρχος τρόπος σχεδιασμού API για το Διαδίκτυο. Τα RESTful API, τα οποία βασίζονται στο πρωτόκολλο HTTP, έγιναν δημοφιλή λόγω της απλότητας και της ευελιξίας τους.
- Πολλές μεγάλες εταιρείες, όπως η Amazon και η Google, άρχισαν να εκθέτουν τις υπηρεσίες τους μέσω RESTful API, διευκολύνοντας την ενσωμάτωση των υπηρεσιών τους σε εφαρμογές τρίτων.

5. Σύγχρονη Εποχή: 2010s και Έπειτα

- Σήμερα, τα API είναι αναπόσπαστο μέρος του οικοσυστήματος του λογισμικού. Οι μεγάλες πλατφόρμες όπως το Facebook, το Twitter και το Google χρησιμοποιούν API για να επιτρέπουν στους προγραμματιστές να δημιουργούν εφαρμογές που αλληλεπιδρούν με τις υπηρεσίες τους.
- Η πρόσφατη άνοδος του **GraphQL** από το Facebook προσφέρει μια πιο ευέλικτη και ισχυρή εναλλακτική λύση στα παραδοσιακά RESTful API.

Τρόπος λειτουργίας ενός api:

1. **Δομή και Επικοινωνία:**
 - Το API καθορίζει ένα σύνολο λειτουργιών (συνήθως μέσω εντολών και παραμέτρων) που οι προγραμματιστές μπορούν να χρησιμοποιήσουν για να ζητήσουν δεδομένα ή να εκτελέσουν λειτουργίες σε μια άλλη υπηρεσία. Αυτό επιτρέπει σε εφαρμογές να ζητούν πληροφορίες ή να εκτελούν ενέργειες σε έναν απομακρυσμένο διακομιστή.
2. **Αιτήματα και Απαντήσεις:**
 - Όταν μια εφαρμογή χρειάζεται δεδομένα ή λειτουργίες από άλλη υπηρεσία, στέλνει ένα αίτημα μέσω του API. Το αίτημα αυτό περιλαμβάνει την εντολή που θέλει να εκτελέσει και τα δεδομένα που απαιτούνται. Η υπηρεσία λαμβάνει το αίτημα, το επεξεργάζεται, και επιστρέφει μια απάντηση στην εφαρμογή, συνήθως σε μορφή JSON ή XML.
3. **Παραδείγματα API:**
 - Ένα κοινό παράδειγμα είναι το API των υπηρεσιών Google Maps, που επιτρέπει στους προγραμματιστές να ενσωματώνουν χάρτες, οδηγίες και τοποθεσίες στις εφαρμογές τους.
 - Τα API των κοινωνικών δικτύων (π.χ., Facebook, Twitter) επιτρέπουν την ενσωμάτωση λειτουργιών όπως η κοινή χρήση περιεχομένου ή η λήψη δεδομένων προφίλ σε άλλες εφαρμογές.

2.1.5 JSON Web Token

Το JWT (JSON Web Token) είναι ένα πρότυπο για την ασφαλή μετάδοση πληροφοριών μεταξύ μερών ως ένα αντικείμενο JSON, το οποίο είναι κωδικοποιημένο χρησιμοποιώντας μια ψηφιακή υπογραφή. Το JWT είναι ιδιαίτερα δημοφιλές στις σύγχρονες web εφαρμογές για την υλοποίηση αυθεντικοποίησης και εξουσιοδότησης, εξασφαλίζοντας ότι οι πληροφορίες που μεταφέρονται είναι αξιόπιστες και αμετάβλητες (wiki_jwt).

Πώς Λειτουργεί το JWT:

1. **Δομή του JWT:**
 - Ένα JWT αποτελείται από τρία μέρη: την επικεφαλίδα (header), το σώμα ή χρήσιμο φορτίο (payload), και την υπογραφή (signature). Αυτά τα τρία μέρη διαχωρίζονται από τελείες.
 - **Header:** Περιέχει τις πληροφορίες σχετικά με τον τύπο του token και τον αλγόριθμο κρυπτογράφησης.
 - **Payload:** Περιέχει τις αξιώσεις (claims), δηλαδή τις πληροφορίες που μεταφέρονται στο token, όπως τα δεδομένα του χρήστη ή ο χρόνος λήξης.
 - **Signature:** Υπογράφεται ψηφιακά για να διασφαλιστεί ότι το περιεχόμενο δεν έχει τροποποιηθεί.
2. **Χρήση:**
 - Το JWT χρησιμοποιείται συχνά για την αυθεντικοποίηση χρηστών. Όταν ένας χρήστης συνδέεται σε μια εφαρμογή, το σύστημα δημιουργεί ένα JWT που περιλαμβάνει τα στοιχεία του χρήστη και το στέλνει στον πελάτη (π.χ., το πρόγραμμα περιήγησης). Αυτό το token επισυνάπτεται στις μελλοντικές αιτήσεις προς τον server, και χρησιμοποιείται για την επαλήθευση της ταυτότητας του χρήστη χωρίς την ανάγκη να ζητηθούν εκ νέου τα διαπιστευτήρια του.

Ιστορικά στοιχεία του JWT:

- **Προέλευση και Ανάπτυξη:**
 - Το JWT αναπτύχθηκε αρχικά από τον *Auth0*, μια εταιρεία που παρέχει υπηρεσίες αυθεντικοποίησης και εξουσιοδότησης, και το πρότυπο καθιερώθηκε από την IETF (Internet Engineering Task Force) ως το RFC 7519 το 2015. Το

JWT σχεδιάστηκε για να επιλύσει προβλήματα ασφάλειας και αποδοτικότητας στη μεταφορά δεδομένων μεταξύ διαφορετικών συστημάτων.

- **Εξάπλωση και Χρήση:**
 - Με την αυξανόμενη δημοτικότητα των εφαρμογών μονού σελίδας (SPA) και των αρχιτεκτονικών *microservices*, το JWT έγινε ευρέως χρησιμοποιούμενο λόγω της ευκολίας του στη χρήση και της ανεξαρτησίας του από την πλατφόρμα. Έχει γίνει το *de facto* πρότυπο για την υλοποίηση *token-based authentication* σε *web* εφαρμογές και *API*.
- **Εξέλιξη:**
 - Κατά τα τελευταία χρόνια, έχουν αναπτυχθεί πολλές βιβλιοθήκες και εργαλεία που διευκολύνουν την υλοποίηση JWT σε διάφορες γλώσσες προγραμματισμού. Η ασφάλεια και η ευελιξία του το έχουν κάνει μία από τις κυρίαρχες μεθόδους για την ασφαλή διαχείριση των αυθεντικοποιημένων συνδέσεων στο διαδίκτυο.

Συνολικά, το JWT είναι ένα κρίσιμο εργαλείο για τις σύγχρονες *web* εφαρμογές, εξασφαλίζοντας την ασφαλή και αποδοτική διαχείριση των αυθεντικοποιημένων συνδέσεων και των δεδομένων που σχετίζονται με αυτές.

2.1.6 Βάσεις Δεδομένων (Databases)

Μια βάση δεδομένων (*database*) είναι μια οργανωμένη συλλογή δεδομένων που αποθηκεύεται και διαχειρίζεται με τρόπο που επιτρέπει την εύκολη πρόσβαση, αναζήτηση και τροποποίηση τους. Οι βάσεις δεδομένων χρησιμοποιούνται σε σχεδόν κάθε τομέα της τεχνολογίας, από τις επιχειρήσεις και την επιστήμη μέχρι τις εφαρμογές ιστού και κινητές εφαρμογές (*wiki_database*).

Οι βάσεις δεδομένων μπορούν να είναι απλές ή σύνθετες, και μπορούν να διαχειρίζονται τεράστιους όγκους δεδομένων με πολύπλοκες σχέσεις μεταξύ τους. Υπάρχουν διαφορετικοί τύποι βάσεων δεδομένων, όπως:

- **Σχεσιακές Βάσεις Δεδομένων (Relational Databases):** Χρησιμοποιούν πίνακες (*tables*) για να οργανώσουν τα δεδομένα σε σειρές και στήλες. Κάθε γραμμή αντιπροσωπεύει μια εγγραφή και κάθε στήλη αντιπροσωπεύει ένα πεδίο. Η *SQL* (*Structured Query Language*) είναι η γλώσσα που χρησιμοποιείται για την αλληλεπίδραση με αυτές τις βάσεις δεδομένων.
- **Μη Σχεσιακές Βάσεις Δεδομένων (NoSQL Databases):** Αυτές οι βάσεις δεδομένων δεν χρησιμοποιούν παραδοσιακές δομές πινάκων και είναι σχεδιασμένες για να διαχειρίζονται δεδομένα που δεν είναι αυστηρά δομημένα, όπως έγγραφα *JSON*, *key-value pairs*, γραφήματα, κ.λπ.
- **Βάσεις Δεδομένων Αντικειμένων (Object-Oriented Databases):** Αποθηκεύουν δεδομένα ως αντικείμενα, όπως στην αντικειμενοστραφή προγραμματιστική. Ιστορικά στοιχεία των βάσεων δεδομένων:

1. **1950s-1960s: Πρώιμα Συστήματα Διαχείρισης Δεδομένων**
 - Οι πρώτες βάσεις δεδομένων αναπτύχθηκαν για να ικανοποιήσουν την ανάγκη για αποθήκευση και διαχείριση μεγάλων ποσοτήτων δεδομένων που παράγονταν από επιχειρήσεις και κυβερνητικούς οργανισμούς. Αυτά τα πρώιμα συστήματα ήταν πολύ βασικά και συνήθως σχεδιασμένα για συγκεκριμένες εφαρμογές.
 - Η πρώτη γνωστή χρήση της βάσης δεδομένων ήταν το σύστημα **SABRE**, που αναπτύχθηκε τη δεκαετία του 1960 από την *IBM* για την *American Airlines*, το οποίο ήταν ένα από τα πρώτα ηλεκτρονικά συστήματα κρατήσεων πτήσεων.
2. **1970s: Εμφάνιση των Σχεσιακών Βάσεων Δεδομένων**
 - Η δεκαετία του 1970 ήταν καθοριστική για την ανάπτυξη της τεχνολογίας των βάσεων δεδομένων, με την εισαγωγή του μοντέλου των σχεσιακών βάσεων

δεδομένων από τον **Edgar F. Codd** το 1970. Το μοντέλο αυτό παρουσίασε την ιδέα της αποθήκευσης δεδομένων σε πίνακες και της χρήσης της SQL για τη διαχείρισή τους.

- Το μοντέλο αυτό καθιέρωσε την βάση για τις σημερινές σχεσιακές βάσεις δεδομένων και οδήγησε στην ανάπτυξη πολλών συστημάτων διαχείρισης βάσεων δεδομένων (DBMS), όπως το **IBM System R** και αργότερα το **Oracle**.
3. **1980s-1990s: Εξάπλωση και Τυποποίηση**
- Κατά τη δεκαετία του 1980, οι βάσεις δεδομένων έγιναν απαραίτητες για τις επιχειρήσεις και η SQL έγινε η τυποποιημένη γλώσσα για την αλληλεπίδραση με σχεσιακές βάσεις δεδομένων. Τα DBMS όπως το Oracle, το Microsoft SQL Server και το MySQL έγιναν δημοφιλή και κυριάρχησαν στην αγορά.
 - Η ανάπτυξη του Διαδικτύου τη δεκαετία του 1990 έφερε νέες απαιτήσεις για τις βάσεις δεδομένων, με τις διαδικτυακές εφαρμογές να απαιτούν αυξημένη κλίμακα και απόδοση.
4. **2000s και Μετά: Άνοδος των NoSQL και Big Data**
- Η άνοδος των μεγάλων δεδομένων (Big Data) και των πολύπλοκων, μη δομημένων δεδομένων οδήγησε στην ανάπτυξη των NoSQL βάσεων δεδομένων, όπως το **MongoDB** και το **Cassandra**. Αυτές οι βάσεις δεδομένων είναι σχεδιασμένες για να διαχειρίζονται μεγάλες ποσότητες δεδομένων που δεν μπορούν να αποθηκευτούν σε παραδοσιακές σχεσιακές βάσεις δεδομένων.
 - Η ανάπτυξη του cloud computing δημιούργησε επίσης νέες απαιτήσεις για ευέλικτες και επεκτάσιμες βάσεις δεδομένων, όπως το **Amazon DynamoDB** και το **Google Bigtable**.

Οι βάσεις δεδομένων είναι θεμελιώδεις για την αποθήκευση, την επεξεργασία και τη διαχείριση των πληροφοριών στον σύγχρονο κόσμο. Από τις πρώτες μορφές αποθήκευσης δεδομένων έως τις πιο προηγμένες και κλιμακούμενες λύσεις, οι βάσεις δεδομένων έχουν εξελιχθεί για να καλύπτουν τις συνεχώς αυξανόμενες ανάγκες της τεχνολογίας και της πληροφορικής.

2.1.7 Structured Query Language (SQL)

Η SQL (Structured Query Language) είναι μια γλώσσα προγραμματισμού που χρησιμοποιείται για τη διαχείριση και το χειρισμό σχεσιακών βάσεων δεδομένων. Με την SQL, οι χρήστες μπορούν να δημιουργούν, να τροποποιούν, να διαγράφουν και να αναζητούν δεδομένα σε μια βάση δεδομένων. Η SQL παρέχει μια δομημένη μέθοδο για την επικοινωνία με τις βάσεις δεδομένων, επιτρέποντας την εκτέλεση πολύπλοκων ερωτημάτων για την ανάκτηση συγκεκριμένων δεδομένων (wiki_sql).

Ιστορικά στοιχεία της sql:

1. **Αρχές και Δημιουργία: 1970s**
 - Η SQL αναπτύχθηκε από τους **Donald D. Chamberlin** και **Raymond F. Boyce** στα εργαστήρια της IBM στις αρχές της δεκαετίας του 1970. Η αρχική έκδοση της SQL ήταν γνωστή ως SEQUEL (Structured English Query Language) και σχεδιάστηκε για να διαχειρίζεται και να ανακτά δεδομένα από τα πρώιμα συστήματα σχεσιακών βάσεων δεδομένων της IBM.
 - Το σχεσιακό μοντέλο δεδομένων, το οποίο αποτέλεσε τη βάση για την SQL, προτάθηκε από τον **Edgar F. Codd** το 1970. Το μοντέλο αυτό ήταν μια επαναστατική προσέγγιση για την οργάνωση δεδομένων σε πίνακες, όπου οι σχέσεις μεταξύ των δεδομένων μπορούσαν να οριστούν με μαθηματική ακρίβεια.
2. **Διάδοση και Καθιέρωση: 1980s**
 - Η πρώτη εμπορική υλοποίηση της SQL έγινε από την IBM στο σύστημα R, το οποίο αργότερα εξελίχθηκε στο DB2. Στις αρχές της δεκαετίας του 1980, η SQL

υιοθετήθηκε από διάφορες άλλες εταιρείες λογισμικού βάσεων δεδομένων, όπως η Oracle, η Sybase και η Microsoft, καθιστώντας την SQL το πρότυπο για τη διαχείριση σχεσιακών βάσεων δεδομένων.

- Το 1986, η SQL αναγνωρίστηκε ως πρότυπο από το ANSI (American National Standards Institute) και το ISO (International Organization for Standardization), γεγονός που συνέβαλε στη διάδοσή της και την τυποποίησή της.

3. Εξέλιξη και Σημερινή Εποχή: 1990s και Μετά

- Στη δεκαετία του 1990 και του 2000, η SQL συνέχισε να εξελίσσεται με την προσθήκη νέων δυνατοτήτων, όπως οι αποθηκευμένες διαδικασίες, οι trigger, και η υποστήριξη για πιο πολύπλοκα ερωτήματα.
- Σήμερα, η SQL παραμένει η κυρίαρχη γλώσσα για τη διαχείριση σχεσιακών βάσεων δεδομένων, με διάφορες παραλλαγές και επεκτάσεις που υλοποιούνται από διαφορετικά συστήματα βάσεων δεδομένων (π.χ., MySQL, PostgreSQL, Microsoft SQL Server).

Η SQL είναι ένα από τα πιο σημαντικά εργαλεία στην ανάπτυξη λογισμικού και στη διαχείριση δεδομένων, και η επιρροή της στη βιομηχανία της πληροφορικής είναι βαθιά και εκτεταμένη.

2.1.8 Java

Η **Java** είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού που αναπτύχθηκε αρχικά από τη Sun Microsystems και κυκλοφόρησε το 1995. Η Java είναι σχεδιασμένη να είναι φορητή, επιτρέποντας στους προγραμματιστές να "γράψουν μία φορά, να τρέξουν παντού" (Write Once, Run Anywhere), κάτι που σημαίνει ότι κώδικας Java μπορεί να τρέξει σε οποιαδήποτε πλατφόρμα που υποστηρίζει τη Java χωρίς την ανάγκη επανασυμπλήρωσης (wiki_java).

Η Java είναι ευρέως χρησιμοποιούμενη για την ανάπτυξη διαφόρων τύπων εφαρμογών, από web και κινητές εφαρμογές έως μεγάλα συστήματα διαχείρισης επιχειρήσεων. Διαθέτει μια ισχυρή και εκτενή βιβλιοθήκη και υποστηρίζεται από μια μεγάλη κοινότητα προγραμματιστών.

Ιστορικά Στοιχεία της Java:

1. Προέλευση και Ανάπτυξη:

- Η Java ξεκίνησε το 1991 ως ένα εσωτερικό project της Sun Microsystems με το όνομα "Oak" υπό την καθοδήγηση του **James Gosling** και της ομάδας του. Η αρχική πρόθεση ήταν να δημιουργήσουν μια γλώσσα προγραμματισμού για έξυπνες συσκευές (όπως set-top boxes), αλλά σύντομα αναγνώρισαν το δυναμικό της γλώσσας για τον προγραμματισμό στο Διαδίκτυο.
- Το 1995, η γλώσσα μετονομάστηκε σε Java και κυκλοφόρησε επίσημα. Η Java αμέσως ξεχώρισε λόγω της δυνατότητάς της να εκτελείται σε διαφορετικές πλατφόρμες χωρίς τροποποιήσεις, κάτι που ήταν ιδιαίτερα σημαντικό για το ανερχόμενο περιβάλλον του Παγκόσμιου Ιστού (World Wide Web).

2. Η Άνοδος του Διαδικτύου:

- Η Java έγινε δημοφιλής χάρη στη χρήση της σε εφαρμογές Διαδικτύου, ειδικά με τη χρήση των **applets**, τα οποία ήταν μικρά προγράμματα Java που μπορούσαν να εκτελούνται σε προγράμματα περιήγησης. Παρόλο που τα applets αργότερα έπεσαν σε αχρηστία, η Java συνέχισε να αναπτύσσεται λόγω της ευρείας υιοθέτησής της σε επιχειρηματικές εφαρμογές και backend συστήματα.

3. Ανάπτυξη της Κοινότητας και των Εκδόσεων:

- Η Java συνεχώς εξελίχθηκε με την πάροδο των ετών. Μετά την εξαγορά της Sun Microsystems από την Oracle το 2010, η Oracle συνέχισε να αναπτύσσει και να βελτιώνει τη γλώσσα. Έχουν κυκλοφορήσει πολλές σημαντικές εκδόσεις της Java, με την Java SE 8 (κυκλοφόρησε το 2014) να φέρνει πολλές καινοτομίες, όπως τα Lambdas και το Stream API, που βελτίωσαν την παραγωγικότητα των προγραμματιστών.

4. Ευρεία Χρήση και Κληρονομιά:

- Η Java είναι μία από τις πιο διαδεδομένες γλώσσες προγραμματισμού παγκοσμίως, χρησιμοποιείται σε διάφορους τομείς όπως η ανάπτυξη Android εφαρμογών, web εφαρμογών, χρηματοοικονομικών συστημάτων και πολλά άλλα.
- Παρά την εμφάνιση νέων γλωσσών προγραμματισμού, η Java παραμένει σημαντική λόγω της σταθερότητάς της, της μεγάλης κοινότητας υποστήριξης και της εκτενούς χρήσης της σε επιχειρηματικά περιβάλλοντα.

2.1.9 Spring Boot

Το Spring Boot είναι ένα δημοφιλές πλαίσιο (framework) για την ανάπτυξη εφαρμογών στη γλώσσα προγραμματισμού Java. Το Spring Boot αναπτύχθηκε ως μια επέκταση του Spring Framework, που είναι ένα από τα πιο διαδεδομένα πλαίσια για τη δημιουργία εφαρμογών επιχειρησιακού επιπέδου. Το Spring Boot στοχεύει να απλοποιήσει τη διαδικασία ανάπτυξης εφαρμογών με το Spring, προσφέροντας προκαθορισμένες ρυθμίσεις (default configurations) και αυτοματοποίηση που μειώνει την πολυπλοκότητα και τον κώδικα που απαιτείται για την εκκίνηση μιας εφαρμογής (wiki_spring_boot).

Ιστορικά στοιχεία του Spring Boot:

1. Προέλευση:

- Το Spring Framework δημιουργήθηκε από τον **Rod Johnson** το 2003 για να διευκολύνει την ανάπτυξη εφαρμογών Java EE με ένα πιο ευέλικτο και ελαφρύ μοντέλο. Το Spring έγινε γρήγορα δημοφιλές λόγω της απλότητας και της επεκτασιμότητάς του.
- Καθώς το Spring Framework εξελισσόταν, έγινε όλο και πιο ισχυρό αλλά και πιο περίπλοκο στη χρήση του. Το **Spring Boot** κυκλοφόρησε επίσημα το 2014 από την εταιρεία Pivotal, που αργότερα έγινε μέρος της VMware. Το Spring Boot σχεδιάστηκε για να απλοποιήσει τη χρήση του Spring Framework, ειδικά για μικροϋπηρεσίες και σύγχρονες web εφαρμογές.

2. Άνοδος της Δημοτικότητας:

- Από την κυκλοφορία του, το Spring Boot γνώρισε ραγδαία αύξηση στη δημοτικότητά του. Η ευκολία με την οποία οι προγραμματιστές μπορούσαν να αναπτύξουν και να εκτελέσουν εφαρμογές, σε συνδυασμό με την ισχυρή υποστήριξη και την εκτενή κοινότητα του Spring, έκανε το Spring Boot την προτιμώμενη επιλογή για πολλές επιχειρησιακές εφαρμογές.
- Το Spring Boot ενσωματώθηκε στις τάσεις της βιομηχανίας, όπως η ανάπτυξη μικροϋπηρεσιών, και υποστηρίζει σύγχρονες αρχιτεκτονικές εφαρμογών, κάνοντας το ένα βασικό εργαλείο για πολλές εταιρείες τεχνολογίας.

3. Σημερινή Κατάσταση:

- Σήμερα, το Spring Boot συνεχίζει να εξελίσσεται με συχνές ενημερώσεις και νέες δυνατότητες. Είναι ευρέως χρησιμοποιούμενο σε εταιρείες τεχνολογίας για την ανάπτυξη εφαρμογών που απαιτούν ευελιξία, επεκτασιμότητα και απόδοση.

Το Spring Boot έχει αλλάξει δραματικά τον τρόπο με τον οποίο οι προγραμματιστές προσεγγίζουν την ανάπτυξη εφαρμογών με Java, προσφέροντας έναν πιο άμεσο και

απλοποιημένο τρόπο ανάπτυξης, με δυνατότητες που ανταποκρίνονται στις σύγχρονες ανάγκες των επιχειρήσεων και των τεχνολογικών λύσεων.

2.1.10 HTML (Hypertext Markup Language)

Η γλώσσα HTML (Hypertext Markup Language) είναι μια βασική γλώσσα προγραμματισμού που χρησιμοποιείται για την κατασκευή και την παρουσίαση ιστοσελίδων στο διαδίκτυο. Η HTML παρέχει τη δομή και τη μορφοποίηση του περιεχομένου σε μια ιστοσελίδα, επιτρέποντας στους προγραμματιστές να ορίζουν διάφορα στοιχεία της σελίδας (elements), όπως επικεφαλίδες, παραγράφους, εικόνες, συνδέσμους και πίνακες (wiki html, 1991).

Ιστορικά στοιχεία για την Html:

1. Δημιουργία της HTML:

- Η HTML αναπτύχθηκε από τον Tim Berners-Lee, τον ιδρυτή του World Wide Web, και παρουσιάστηκε για πρώτη φορά το 1991. Ο στόχος ήταν να δημιουργηθεί μια απλή γλώσσα που να επιτρέπει την ανταλλαγή και την παρουσίαση κειμένου και πολυμέσων μέσω του διαδικτύου.

2. Πρώτες Εκδόσεις:

- **HTML 1.0:** Η πρώτη έκδοση, γνωστή και ως HTML 1.0, δημοσιεύθηκε το 1991 και περιλάμβανε βασικά στοιχεία όπως επικεφαλίδες, παραγράφους, συνδέσμους και λίστες.
- **HTML 2.0:** Κυκλοφόρησε το 1995 και εισήγαγε νέες δυνατότητες, όπως πίνακες και φόρμες. Η HTML 2.0 έγινε το πρώτο πρότυπο HTML που εγκρίθηκε από την IETF (Internet Engineering Task Force).

3. HTML 3.2 και 4.0:

- **HTML 3.2:** Εισήχθη το 1997 και πρόσθεσε στοιχεία όπως CSS (Cascading Style Sheets) για τη διαχείριση της εμφάνισης των σελίδων, καθώς και υποστήριξη για δυναμικά περιεχόμενα και applets Java.
- **HTML 4.0:** Κυκλοφόρησε το 1997 και περιλάμβανε σημαντικές αλλαγές όπως η διαχωριστική ικανότητα της δομής της σελίδας από την παρουσίαση (μέσω CSS) και η υποστήριξη για διαδραστικά περιεχόμενα με τη χρήση του DOM (Document Object Model).

4. HTML5:

- **HTML5:** Ανακοινώθηκε το 2008 και έγινε το επίσημο πρότυπο το 2014. Η HTML5 προσφέρει πολλές βελτιώσεις και νέες δυνατότητες, όπως ενσωματωμένη υποστήριξη για πολυμέσα (βίντεο και ήχο), νέες ετικέτες (όπως <article>, <section>, <nav>), και APIs για διαδραστικές λειτουργίες (όπως Web Storage και Canvas).

5. Συνεχιζόμενη Εξέλιξη:

- Η HTML συνεχίζει να εξελίσσεται, με τις νέες εκδόσεις να προσφέρουν επιπλέον δυνατότητες και βελτιώσεις. Η HTML5+ και οι συνεχείς βελτιώσεις διασφαλίζουν τη συμβατότητα με τις σύγχρονες απαιτήσεις του διαδικτύου και τις εξελισσόμενες τεχνολογίες.

Βασικά στοιχεία και χαρακτηριστικά της Html:

1. Δομή Εγγράφου:

- Η HTML χρησιμοποιεί "tags" (ετικέτες) για να ορίσει τη δομή του περιεχομένου, το οποίο χωρίζεται σε στοιχεία (elements). Κάθε ετικέτα περιβάλλεται από αγκύλες, π.χ., <p> για παράγραφο, <h1> για επικεφαλίδα και δημιουργεί το αντίστοιχο element.

2. Στοιχεία και Ατρίβες (attributes):

- Τα στοιχεία HTML μπορεί να έχουν "attributes" (ατρίβες) που παρέχουν πρόσθετες πληροφορίες. Για παράδειγμα, η ετικέτα <a> για σύνδεσμο μπορεί να έχει την ατρίβα href για να καθορίσει τη διεύθυνση URL του συνδέσμου.

3. Διαχωρισμός Περιεχομένου και Παρουσίασης:

- Η HTML επικεντρώνεται στη δομή του περιεχομένου, ενώ η εμφάνιση και η διάταξη διαχειρίζονται συνήθως μέσω CSS (Cascading Style Sheets).

4. Ενσωμάτωση Πολυμέσων:

- Η HTML5 εισήγαγε στοιχεία όπως <video> και <audio> για την ενσωμάτωση πολυμέσων χωρίς την ανάγκη πρόσθετων plug-ins.

5. Διαδραστικά Στοιχεία:

- Εισήχθησαν στοιχεία για διαδραστικότητα, όπως φόρμες με <input>, <button>, και χαρακτηριστικά για επικοινωνία με JavaScript.

Η HTML είναι θεμελιώδης για την ανάπτυξη ιστοσελίδων και την παρουσίαση περιεχομένου στο διαδίκτυο. Από την αρχική της δημιουργία έως την σύγχρονη έκδοση HTML5, η γλώσσα έχει εξελιχθεί σημαντικά για να υποστηρίξει την ανάπτυξη πλούσιων και διαδραστικών ιστοσελίδων.

2.1.11 CSS (Cascading Style Sheets)

Το CSS (Cascading Style Sheets) είναι μια γλώσσα σχεδίασης που χρησιμοποιείται για να καθορίσει την εμφάνιση και τη μορφοποίηση ενός εγγράφου HTML ή XML (όπως XHTML). Το CSS επιτρέπει στους σχεδιαστές να εφαρμόζουν στυλ σε στοιχεία (elements) μιας ιστοσελίδας, όπως χρώματα, γραμματοσειρές, διαστήματα και διάταξη, αποχωρίζοντας έτσι τη δομή του περιεχομένου από την εμφάνιση (wiki css, 1996).

Ιστορικά στοιχεία για το CSS:

1. Αρχικές Σκέψεις και Σχέδιο:

- **Πρώτες Ιδέες:** Η ανάγκη για διαχωρισμό της εμφάνισης από τη δομή των εγγράφων προήλθε από την επιθυμία να διαχωριστούν τα στυλ από το περιεχόμενο. Πριν την εισαγωγή του CSS, η μορφοποίηση των ιστοσελίδων γινόταν κυρίως με HTML, χρησιμοποιώντας ετικέτες όπως και <center>, οι οποίες ήταν περιορισμένες και δύσχρηστες.

2. CSS 1.0:

- **Δημιουργία και Κυκλοφορία:** Το CSS εισήχθη για πρώτη φορά το 1996 από τον W3C (World Wide Web Consortium) υπό την καθοδήγηση του Håkon Wium Lie. Η πρώτη έκδοση, CSS 1.0, κυκλοφόρησε το Δεκέμβριο του 1996 και παρείχε βασικές δυνατότητες για την εφαρμογή στυλ σε έγγραφα HTML, όπως χρώματα, γραμματοσειρές, και περιθώρια.

3. CSS 2.0:

- **Κυκλοφορία και Νέες Δυνατότητες:** Το CSS 2.0 εγκρίθηκε το Μάιο του 1998 και εισήγαγε πολλές νέες δυνατότητες, όπως υποστήριξη για θέσεις, πίνακες, και διαχωριστικά στυλ. Επίσης, προστέθηκαν δυνατότητες για στυλ σε διαφορετικές οθόνες και μέσα (όπως εκτυπωτές και κινητές συσκευές).
4. **CSS 2.1:**
- **Αναθεώρηση και Βελτιώσεις:** Το CSS 2.1 κυκλοφόρησε το 2005 και περιλάμβανε διορθώσεις και αναθεωρήσεις της προηγούμενης έκδοσης. Στόχος ήταν η σταθεροποίηση του προτύπου και η βελτίωση της υποστήριξης από τους browsers.
5. **CSS 3.0:**
- **Εξέλιξη και Σημαντικές Προσθήκες:** Το CSS 3.0 δεν ήταν μια ενιαία έκδοση αλλά ανήκε σε πολλά "modules" που αναπτύχθηκαν ξεχωριστά. Οι σημαντικές νέες δυνατότητες περιλάμβαναν τα νέα χαρακτηριστικά για κινούμενα σχέδια (animations), μεταβάσεις (transitions), στυλ εφέ (effects), και καλύτερη υποστήριξη για διάταξη μέσω flexbox και grid layout.
 - **Μερικές Δημοφιλείς Προσθήκες:**
 - **@media Queries:** Εισάγουν την έννοια του responsive design, επιτρέποντας τη δημιουργία ιστοσελίδων που προσαρμόζονται σε διαφορετικά μεγέθη οθόνης.
 - **Flexbox και Grid Layout:** Επιτρέπουν πιο ευέλικτους και σύνθετους τρόπους διάταξης περιεχομένου.
6. **Σύγχρονη Εξέλιξη:**
- **CSS4 και Μελλοντικές Εκδόσεις:** Η ανάπτυξη του CSS συνεχίζεται με νέες προδιαγραφές και δυνατότητες. Οι εξελίξεις περιλαμβάνουν περαιτέρω ενσωματώσεις, υποστήριξη για νέες τεχνολογίες και συνεχιζόμενη βελτίωση της υποστήριξης από τους web browsers.

Βασικά στοιχεία και χαρακτηριστικά του CSS:

1. Επιλογείς (Selectors):

- Το CSS χρησιμοποιεί επιλογείς για να στοχεύσει συγκεκριμένα στοιχεία σε ένα έγγραφο HTML και να εφαρμόσει στυλ σε αυτά. Παράδειγμα: `p { color: red; }` θα αλλάξει το χρώμα όλων των παραγράφων σε κόκκινο.

2. Ιδιότητες και Τιμές:

- Οι ιδιότητες του CSS καθορίζουν τα χαρακτηριστικά που θέλουμε να εφαρμόσουμε (π.χ., color, font-size, margin). Κάθε ιδιότητα έχει μια ή περισσότερες τιμές που καθορίζουν την εμφάνιση του στοιχείου.

3. Συνδυασμός και Κληρονομικότητα:

- Το CSS επιτρέπει την κληρονομικότητα, όπου τα στυλ που εφαρμόζονται σε ένα γονικό στοιχείο επηρεάζουν τα παιδικά στοιχεία του. Επίσης, το CSS μπορεί να συνδυάζει πολλές δηλώσεις στυλ και κανόνες για πιο σύνθετα σχέδια.

4. Εξωτερικά, Εσωτερικά και Ενσωματωμένα Στυλ:

- **Εξωτερικά CSS:** Στυλ που εφαρμόζονται μέσω εξωτερικών αρχείων .css, που συνδέονται με το HTML έγγραφο μέσω της ετικέτας <link>.
- **Εσωτερικά CSS:** Στυλ που δηλώνονται μέσα στο <style> tag μέσα στο <head> τμήμα ενός HTML εγγράφου.
- **Ενσωματωμένα CSS:** Στυλ που δηλώνονται απευθείας μέσα στα HTML tags μέσω του attribute style.

Το CSS είναι κρίσιμο για την δημιουργία αισθητικά ευχάριστων και λειτουργικών ιστοσελίδων, παρέχοντας ένα ευέλικτο και ισχυρό εργαλείο για τη μορφοποίηση του περιεχομένου. Από τις πρώτες του εκδόσεις μέχρι τις σύγχρονες εξελίξεις, το CSS συνεχίζει να αναπτύσσεται για να υποστηρίξει τις συνεχώς αυξανόμενες απαιτήσεις του διαδικτύου.

2.1.12 Javascript

Η JavaScript είναι μια γλώσσα προγραμματισμού που χρησιμοποιείται κυρίως για την ανάπτυξη διαδραστικών και δυναμικών στοιχείων σε ιστοσελίδες και εφαρμογές ιστού. Ενσωματώνεται συνήθως σε HTML και CSS για να προσφέρει λειτουργικότητες όπως αλληλεπίδραση χρήστη, διαχείριση δεδομένων και αλλαγές στη σελίδα χωρίς την ανάγκη ανανέωσης (wiki javascript, 1995).

Ιστορικά στοιχεία για την Javascript:

1. Δημιουργία και Κυκλοφορία:

- **Αρχική Δημιουργία:** Η JavaScript αναπτύχθηκε από τον Brendan Eich της Netscape Communications. Η γλώσσα αρχικά δημιουργήθηκε με την ονομασία Mocha, στη συνέχεια μετονομάστηκε σε LiveScript, και τελικά έγινε γνωστή ως JavaScript.
- **Εκτόξευση:** Κυκλοφόρησε το Δεκέμβριο του 1995 με την Netscape Navigator 2.0. Η βασική ιδέα ήταν να παρέχει δυναμικές δυνατότητες στις ιστοσελίδες που μέχρι τότε ήταν κυρίως στατικές.

2. Εκδόσεις και Εξελίξεις:

- **ECMAScript:** Το 1997, η JavaScript υιοθετήθηκε από το ECMA (European Computer Manufacturers Association) και πήρε το επίσημο όνομα ECMAScript. Ο πρώτος επίσημος standard ήταν ECMAScript 1 (ES1).
- **ECMAScript 2 και 3:** Ενέκριναν διορθώσεις και πρόσθεσαν νέες δυνατότητες, όπως τα regular expressions και η υποστήριξη για καλύτερη διαχείριση των εξαιρέσεων.
- **ECMAScript 4:** Αν και ήταν πολύ φιλόδοξο project και επρόκειτο να περιλαμβάνει πολλές μεγάλες αλλαγές, δεν υιοθετήθηκε ποτέ πλήρως λόγω διαφωνιών στην κοινότητα.

3. ECMAScript 5 (ES5):

- **Κυκλοφορία:** Εγκρίθηκε το Δεκέμβριο του 2009. Εισήγαγε σημαντικές δυνατότητες όπως strict mode, getter/setter methods, JSON support, και βελτιώσεις στη διαχείριση εξαιρέσεων.

4. ECMAScript 6 (ES6):

- **Αναγνώριση:** Κυκλοφόρησε το Ιούνιο του 2015 και είναι επίσης γνωστό ως ECMAScript 2015. Αυτή η έκδοση εισήγαγε πολλές νέες δυνατότητες, όπως classes, modules, arrow functions, promises, και template literals.
- **Βελτιώσεις:** Έφερε σημαντικές αλλαγές στη γλώσσα, καθιστώντας την πιο ισχυρή και εκφραστική.

5. Σύγχρονες Εκδόσεις (ES7 και μεταγενέστερες):

- **ECMAScript 2016 (ES7):** Εισήγαγε τα features Array.prototype.includes και Exponentiation operator.
- **ECMAScript 2017 (ES8):** Προσέθεσε async/await, object.entries(), και object.values().
- **ECMAScript 2018 (ES9):** Περιλάμβανε νέες δυνατότητες όπως rest/spread properties και asynchronous iteration.
- **ECMAScript 2019 (ES10):** Εισήγαγε features όπως Array.prototype.flat(), Object.fromEntries(), και άλλες βελτιώσεις.

6. Εξελίξεις και Σύγχρονες Εφαρμογές:

- **ES11 και Επόμενες Εκδόσεις:** Εξελίξεις περιλαμβάνουν νέα χαρακτηριστικά όπως optional chaining και nullish coalescing operator.

Βασικά Χαρακτηριστικά της JavaScript:

1. Διαχείριση DOM (Document Object Model):

- Η JavaScript επιτρέπει την αλληλεπίδραση με το DOM της ιστοσελίδας, επιτρέποντας τη δυναμική αλλαγή του περιεχομένου, της δομής και της εμφάνισης της σελίδας.

2. Διαχείριση Εφαρμογών:

- Σημαντική για την ανάπτυξη client-side εφαρμογών, παρέχει τη δυνατότητα να δημιουργηθούν πλούσιες και διαδραστικές εφαρμογές ιστού, όπως διαχείριση φορμών, επικοινωνία με servers μέσω AJAX, και άλλα.

3. Συμβατότητα με Web APIs:

- Υποστηρίζει διάφορα Web APIs για πρόσβαση σε λειτουργίες του προγράμματος περιήγησης, όπως local storage, Geolocation, WebSocket κ.ά.

4. Σύγχρονα Frameworks και Βιβλιοθήκες:

- Η JavaScript έχει υποστηρίξει την ανάπτυξη δημοφιλών βιβλιοθηκών και frameworks όπως το React, Angular, Vue.js, που διευκολύνουν την ανάπτυξη διαδραστικών εφαρμογών ιστού.

5. Εκτέλεση στο Client Side:

- Συνήθως εκτελείται στον περιηγητή του χρήστη (client-side), αλλά με τη βοήθεια περιβαλλόντων όπως το Node.js, μπορεί να εκτελείται και στον server-side (τρέχει σε ηλεκτρονικό υπολογιστή ως desktop app και όχι σε πλοηγό ως web app).

Η JavaScript είναι μια κρίσιμη γλώσσα προγραμματισμού για την ανάπτυξη διαδραστικών και δυναμικών στοιχείων στο διαδίκτυο. Από την αρχική της δημιουργία μέχρι τις

σύγχρονες εκδόσεις, η γλώσσα έχει εξελιχθεί σημαντικά, προσφέροντας δυνατότητες που υποστηρίζουν τη σύγχρονη ανάπτυξη ιστού και εφαρμογών.

2.1.13 React.js

Το React είναι μια δημοφιλής βιβλιοθήκη JavaScript που χρησιμοποιείται για την κατασκευή διαδραστικών και δυναμικών διεπαφών χρήστη (UI). Αναπτύχθηκε από το Facebook και κυκλοφόρησε το 2013 ως ανοιχτού κώδικα εργαλείο. Το React επιτρέπει στους προγραμματιστές να δημιουργούν μεγάλες, single-page εφαρμογές (SPA) που μπορούν να ενημερώνουν τα δεδομένα τους χωρίς να χρειάζεται να φορτώνεται ξανά η σελίδα (wiki_react).

Το React χρησιμοποιεί μια προσέγγιση βασισμένη στα "components" (συστατικά), όπου κάθε μέρος της διεπαφής χρήστη είναι κατασκευασμένο ως ένα ανεξάρτητο και επαναχρησιμοποιήσιμο κομμάτι κώδικα. Ένα από τα κύρια χαρακτηριστικά του React είναι το "Virtual DOM", το οποίο βελτιστοποιεί την απόδοση της ενημέρωσης του UI, επιτρέποντας γρηγορότερες και πιο αποτελεσματικές αλλαγές.

Ιστορικά Στοιχεία του React:

1. Δημιουργία και Κυκλοφορία:

- Το React δημιουργήθηκε αρχικά από τον **Jordan Walke**, έναν μηχανικό λογισμικού του Facebook, το 2011. Στόχος του ήταν να απλοποιήσει τη διαδικασία κατασκευής σύνθετων διεπαφών χρήστη σε εφαρμογές ιστού.
- Το 2013, το React κυκλοφόρησε δημόσια ως ανοιχτού κώδικα από το Facebook, και σύντομα άρχισε να κερδίζει δημοτικότητα στην κοινότητα των προγραμματιστών.

2. Επέκταση και Υιοθέτηση:

- Αρχικά χρησιμοποιήθηκε εσωτερικά στο Facebook, ιδιαίτερα στο news feed του κοινωνικού δικτύου. Η επιτυχία του σε αυτό το περιβάλλον οδήγησε στην εξωτερική κυκλοφορία του και στην υιοθέτησή του από πολλούς άλλους οργανισμούς και προγραμματιστές.
- Το 2015, το Facebook κυκλοφόρησε το **React Native**, ένα παρακλάδι του React, που επιτρέπει στους προγραμματιστές να χρησιμοποιούν τη βιβλιοθήκη για την ανάπτυξη κινητών εφαρμογών που λειτουργούν σε iOS και Android.

3. Εξέλιξη και Κοινότητα:

- Από την κυκλοφορία του, το React έχει γνωρίσει μεγάλη εξέλιξη. Έχουν προστεθεί νέα χαρακτηριστικά, όπως τα "React Hooks" το 2018, τα οποία επιτρέπουν στους προγραμματιστές να χρησιμοποιούν τοπική κατάσταση (state) και άλλες δυνατότητες των React components χωρίς να χρειάζονται κλάσεις.
- Η κοινότητα γύρω από το React είναι τεράστια, και υπάρχει μια μεγάλη ποικιλία εργαλείων και βιβλιοθηκών που έχουν αναπτυχθεί για να συμπληρώσουν ή να βελτιώσουν τη χρήση του.

4. Σημερινή Κατάσταση:

- Το React είναι μία από τις πιο δημοφιλείς βιβλιοθήκες JavaScript και χρησιμοποιείται ευρέως σε εταιρείες τεχνολογίας παγκοσμίως,

συμπεριλαμβανομένων των Facebook, Instagram, Airbnb, Netflix, και πολλές άλλες. Το React έχει γίνει ουσιαστικά το de facto standard για την ανάπτυξη σύγχρονων διεπαφών χρήστη.

Το React έχει φέρει επανάσταση στην ανάπτυξη εφαρμογών ιστού και κινητών, καθιστώντας ευκολότερο για τους προγραμματιστές να δημιουργούν γρήγορα και αποτελεσματικά δυναμικές και αποδοτικές διεπαφές χρήστη.

2.2 Απαιτήσεις Συστήματος

Για την υλοποίηση του συστήματος πέρα από τις βασικές έννοιες απαιτούνται γνώσεις αρχιτεκτονικής ανάπτυξης εφαρμογών. Στην παρούσα χρησιμοποιήθηκε το αρχιτεκτονικό πρότυπο ανάπτυξης λογισμικού Μοντέλο – Προβολή – Ελεγκτής (Model-View-Controller ή MVC).

2.2.1 Model-View-Controller (MVC)

Το MVC (Model-View-Controller) είναι ένα αρχιτεκτονικό πρότυπο (design pattern) που χρησιμοποιείται για την ανάπτυξη εφαρμογών, ιδιαίτερα σε περιβάλλοντα διακομιστών (servers). Ο κύριος στόχος του MVC είναι ο διαχωρισμός της λογικής της εφαρμογής από την παρουσίαση και τη διαχείριση των δεδομένων, καθιστώντας την ανάπτυξη και συντήρηση πιο ευέλικτη και οργανωμένη.

Model (Μοντέλο): Διαχειρίζεται τα δεδομένα και τη λογική της εφαρμογής. Το μοντέλο είναι υπεύθυνο για την αλληλεπίδραση με τη βάση δεδομένων, την επεξεργασία των δεδομένων και την αποστολή τους στον ελεγκτή.

View (Προβολή): Είναι υπεύθυνη για την παρουσίαση των δεδομένων στον χρήστη. Περιλαμβάνει τη διεπαφή χρήστη και τον τρόπο εμφάνισης των δεδομένων που λαμβάνονται από το μοντέλο.

Controller (Ελεγκτής): Συντονίζει την αλληλεπίδραση μεταξύ του μοντέλου και της θέασης. Ο ελεγκτής δέχεται τις εισόδους από τον χρήστη, καλεί το κατάλληλο μοντέλο για την επεξεργασία των δεδομένων και μετά ενημερώνει τη θέαση για να παρουσιαστούν τα δεδομένα.

2.3 Διάγραμμα Δραστηριότητας

Πριν την εκκίνηση κατασκευής της παρούσας εργασίας δημιουργήθηκε διάγραμμα Δραστηριότητας (Activity Diagram) (Εικόνα 2.1) για την οργάνωση της, την κατανόηση της ροής των διαδικασιών αυτής, από την συλλογή δεδομένων έως την συγγραφή της. Το διάγραμμα τηρήθηκε πιστά από την αρχή έως το τέλος αυτής της εργασίας.

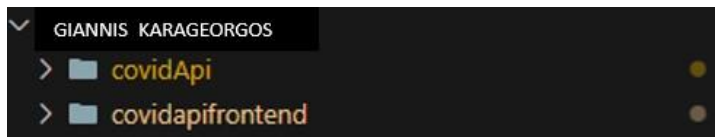


Εικόνα 2.1 Διάγραμμα δραστηριότητας παρούσας εργασίας

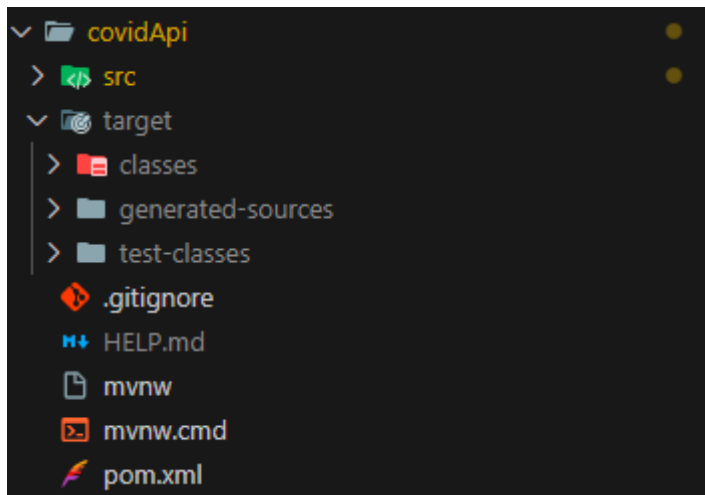
Κατά την διάρκεια κατασκευής της παρούσας εργασίας παρουσιάστηκε το πρόβλημα Κοινής Χρήση Πόρων Μεταξύ Τομέων (Cross-Origin Resource Sharing - CORS). Το πρόβλημα αυτό δεν είχε προβλεφθεί από τον συγγραφέα της εργασίας και για αυτό δεν υπάρχει στο διάγραμμα δραστηριότητας. Το πρόβλημα αυτό επιλύθηκε κατά την εμφάνιση του.

3. Το Σύστημα

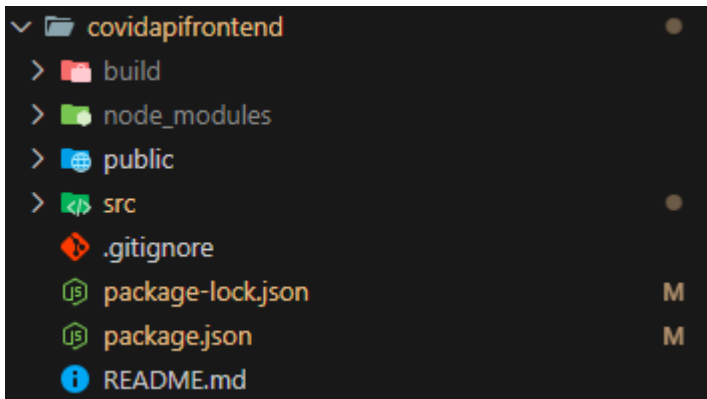
Η παρούσα εργασία στην τελική της μορφή περιέχει στον κεντρικό της φάκελο δύο υποφακέλους (Εικόνα 3.1). Ο υποφάκελος covidApi περιέχει τα αρχεία που δημιουργήθηκαν στην γλώσσα προγραμματισμού Java με χρήση του πλαισίου (Framework) αυτής που ονομάζεται Spring Boot (Εικόνα 3.2). Ο υποφάκελος covidapifrontend περιέχει τα αρχεία που δημιουργήθηκαν στην γλώσσα προγραμματισμού Javascript με χρήση της βιβλιοθήκης αυτής που ονομάζεται react.js (Εικόνα 3.3).



Εικόνα 3.1 Δομή υποφακέλων εργασίας στο VS Code



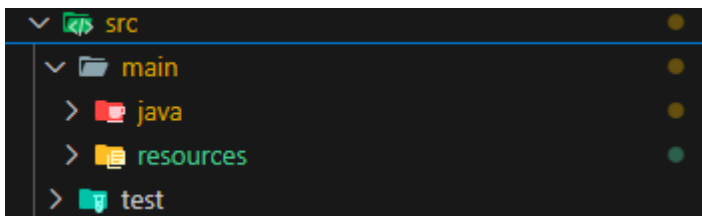
Εικόνα 3.2 Δομή υποφακέλων Spring Boot στο VS Code



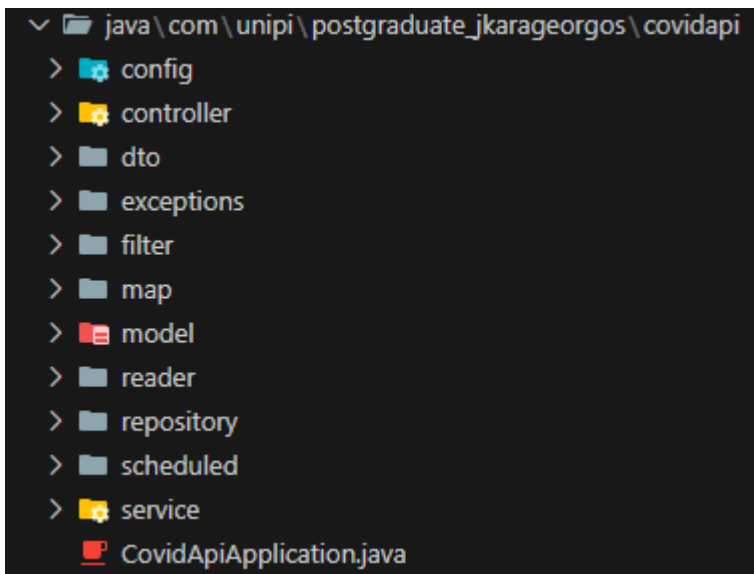
Εικόνα 3.3 Δομή υποφακέλων React.js στο VS Code

3.1 Υποφάκελος covidApi (Spring Boot)

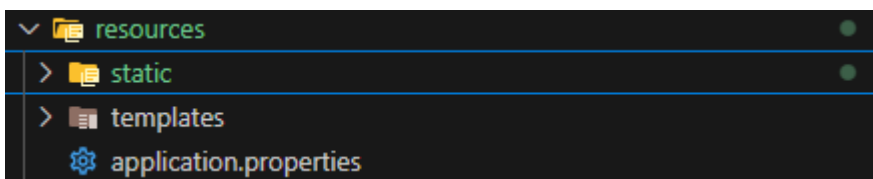
Ο υποφάκελος covidApi περιέχει τα αρχεία που δημιουργήθηκαν στην γλώσσα προγραμματισμού Java με χρήση του πλαισίου (Framework) αυτής που ονομάζεται Spring Boot. Στον υποφάκελο αυτό υπάρχουν δυο υποφάκελοι και πέντε αρχεία που δημιουργήθηκαν αυτόματα από το Spring Boot (Εικόνα 3.2). Ο πρώτος υποφάκελος με ονομασία src (Εικόνα 3.4) είναι αυτός που χρησιμοποιήθηκε από τον δημιουργό της παρούσας για την υλοποίηση του διακομιστή ιστού (web server) και της διεπαφής προγραμματισμού εφαρμογών (API). Ο υποφάκελος αυτός περιέχει υποφάκελο main με 2 υποφακέλους με ονόματα java και resources. Στον φάκελο java (Εικόνα 3.5) δημιουργήθηκαν όλα τα απαραίτητα αρχεία του διακομιστή ιστού και της διεπαφής προγραμματισμού εφαρμογών ενώ στον φάκελο resources(Εικόνα 3.6) τοποθέτηθηκαν όλα τα αρχεία που χτίστηκαν (build) από την βιβλιοθήκη της Javascript React.js.



Εικόνα 3.4 Δομή υποφακέλου src στο VS Code



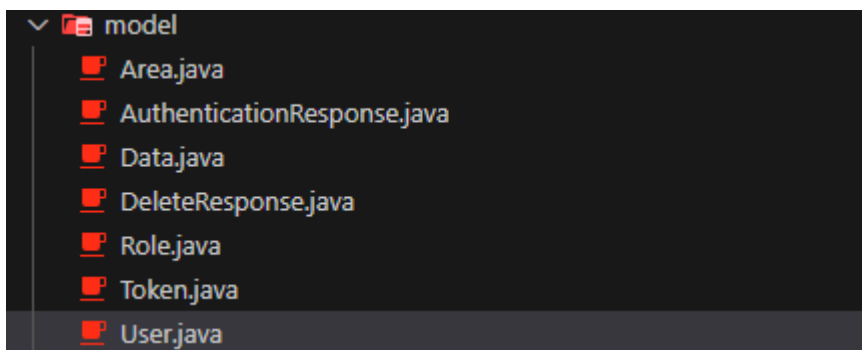
Εικόνα 3.5 Δομή υποφακέλου java στο VS Code



Εικόνα 3.6 Δομή υποφακέλου resources στο VS Code

3.1.1 Υποφάκελος model (Μοντέλο)

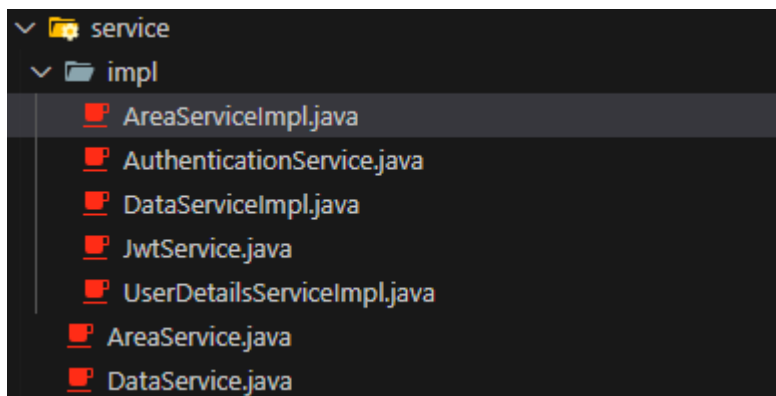
Για την υλοποίηση του αρχιτεκτονικού μοντέλου MVC αρχικά δημιουργήθηκαν τα μοντέλα των δεδομένων, δηλαδή οι κλάσεις της Java για τα δεδομένα (Εικόνα 3.7), με τους αντίστοιχους δομητές (constructors), τις αντίστοιχες μεθόδους ανάκτησης δεδομένων (getters) και τις αντίστοιχες μεθόδους ορισμού δεδομένων (setters). Σε αυτές περιλαμβάνονται οι κλάσεις για τα δεδομένα της κάθε περιοχής, για τα δεδομένα της ασθένειας COVID-19, για τα δεδομένα των χρηστών του συστήματος, ο ρόλος των χρηστών του συστήματος, όπως επίσης και οι βοηθητικές κλάσεις για την δημιουργία και έλεγχο της αυθεντικότητας του JWT (JSON Web Token) ώστε να επιτευχθεί η ασφαλής μετάδοση πληροφοριών μεταξύ των χρηστών και του συστήματος.



Εικόνα 3.7 Δομή υποφακέλου model στο VS Code

3.1.2 Υποφάκελος service

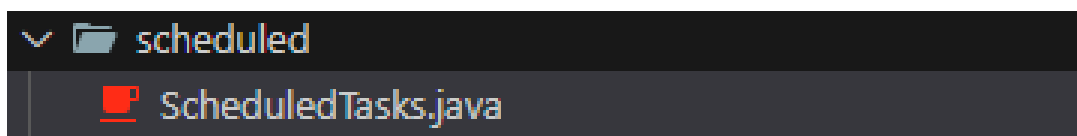
Για την αλληλεπίδραση των μοντέλων με την βάση δεδομένων, δημιουργήθηκαν οι απαραίτητες διεπαφές (interfaces) της java όπως και οι απαραίτητες κλάσεις που υλοποιούν της διεπαφές αυτές (Εικόνα 3.8). Οι κλάσεις αυτές περιέχουν τις κατάλληλες συναρτήσεις για την δημιουργία νέων δεδομένων, την ανάκτηση υπάρχοντων δεδομένων, την αναβάθμιση υπάρχοντων δεδομένων καθώς και την διαγραφή αυτών στην βάση δεδομένων.



Εικόνα 3.8 Δομή υποφακέλου service στο VS Code

3.1.3 Υποφάκελος scheduled

Αφού κατασκευάστηκαν τα απαραίτητα μοντέλα δεδομένων και οι αντίστοιχες κλάσεις ένωσης αυτών με την βάση δεδομένων, δημιουργήθηκε ο φάκελος scheduled που περιέχει την αντίστοιχη κλάση για την συλλογή των δεδομένων από το διαδίκτυο (Εικόνα 3.9). Τα δεδομένα αυτά συλλέγονται από την ιστοσελίδα github.com, από τον λογαριασμό της ιστοσελίδας ourworldindata.org. Τα δεδομένα αυτά παρέχονται σε αρχείο .csv (Comma Separated Values), το οποίο κατεβαίνει (download), διαβάζεται(read) και με χρήση των μοντέλων και των αντίστοιχων κλάσεων εισάγονται στην βάση δεδομένων. Η κλάση αυτή μπορεί να προγραμματιστεί να εκτελεί την παραπάνω διαδικασία ανά τακτά χρονικά διαστήματα χρησιμοποιώντας το σχόλιο (annotation) @Scheduled του πλαισίου Spring Boot που παρέχεται για αυτήν την δουλειά. Στην παρούσα εργασία η συλλογή δεδομένων προγραμματίστηκε να συλλέγει δεδομένα κάθε μια ώρα.

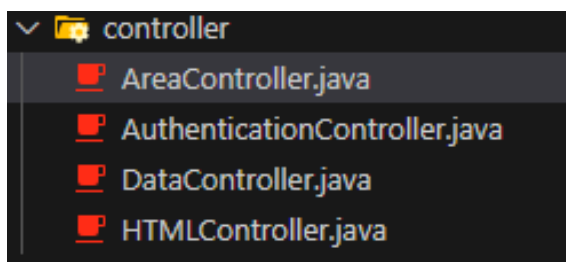


Εικόνα 3.9 Δομή συ scheduled στο VS Code

3.1.4 Υποφάκελος controller (Ελεγκτής)

Μετά την συλλογή δεδομένων, δημιουργήθηκαν οι απαραίτητες κλάσεις της Java που αντιπροσωπεύουν τον ελεγκτή του αρχιτεκτονικού μοντέλου MVC (Εικόνα 3.10). Αρχικά

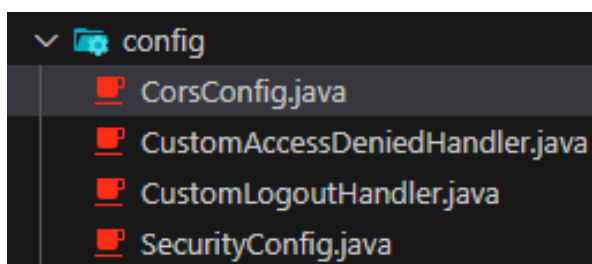
δημιουργήθηκαν οι κλάσεις της διεπαφής προγραμματισμού εφαρμογών (API), οι οποίες επιτρέπουν σε οποιονδήποτε χρήστη να αιτήσει (request) από την βάση δεδομένων τα διαθέσιμα δεδομένα, δημιουργήθηκε κλάση η οποία προστατεύει τα δεδομένα αυτά με χρήση του προτύπου για ασφαλή μετάδοση πληροφοριών JWT (JSON Web Token). Τέλος δημιουργήθηκε κλάση που θα αποστέλει τις ιστοσελίδες (View του αρχιτεκτονικού μοντέλου MVC) για την παρουσίαση των δεδομένων που δημιουργήθηκαν αργότερα με χρήση της γλώσσας προγραμματισμού Javascript.



Εικόνα 3.10 Δομή υποφακέλου controller στο VS Code

3.1.5 Υποφάκελος config - CORS

Περιέχει τα απαραίτητα αρχεία ρυθμίσεων (configuration files) για τη εύρυθμη λειτουργία του διακομιστή ιστού (web server) και της διεπαφής προγραμματισμού εφαρμογών (API). Κατά την διάρκεια κατασκευής των ιστοσελίδων, για την υλοποίηση της Προβολής (View) του αρχιτεκτονικού μοντέλου MVC, με χρήση της βιβλιοθήκης React.js της γλώσσας προγραμματισμού Javascript παρουσιάστηκε το πρόβλημα Κοινής Χρήση Πόρων Μεταξύ Τομέων (Cross-Origin Resource Sharing - CORS). Τα cors είναι ένας μηχανισμός ασφαλείας που επιτρέπει ή περιορίζει αιτήματα πόρων από έναν διαφορετικό τομέα (domain) από αυτόν στον οποίο βρίσκεται η ιστοσελίδα ή η εφαρμογή. Η διεπαφή προγραμματισμού εφαρμογών (API) που δημιουργήθηκε με χρήση της Java δεν επέτρεπε την αποστολή δεδομένων σε μια ιστοσελίδα η οποία διακομίζεται από κάποιον άλλο διακομιστή ιστού με διαφορετικό τομέα (domain) που δημιουργήθηκε αυτόματα από την React.js. Το πρόβλημα αυτό δεν είχε υπολογιστεί από τον δημιουργό της παρούσης κατά των σχεδιασμό αυτής και για τον λόγο αυτό δεν υπάρχει στο διάγραμμα δραστηριότητας. Για να επιλυθεί αυτό το πρόβλημα δημιουργήθηκε το αρχείο CorsConfig.java το οποίο περιέχει της απαραίτητες ρυθμίσεις (configurations) ώστε η διεπαφή προγραμματισμού εφαρμογών (API) να επιτρέπει αιτήματα (requests) από οποιονδήποτε τομέα(domain).

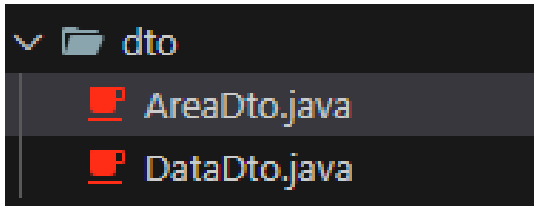


Εικόνα 3.11 Δομή υποφακέλου config στο VS Code

3.1.6 Υποφάκελος Dto

Περιέχει βοηθητικές βιβλιοθήκες για την επικοινωνία της Java με την βάση δεδομένων.

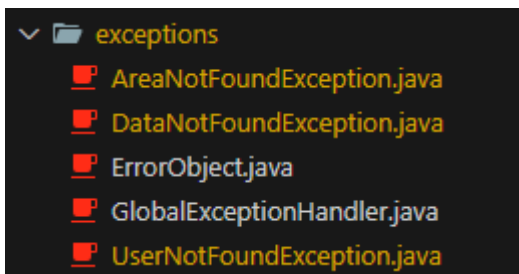
Περιέχει κλάσεις με δομητές (constructors), μεθόδους ανάκτησης δεδομένων (getters) και μεθόδους ορισμού δεδομένων (setters) (Εικόνα 3.12).



Εικόνα 3.12 Δομή υποφακέλου dto στο VS Code

3.1.7 Υποφάκελος exceptions

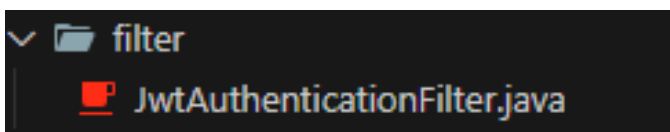
Περιέχει τις απαραίτητες κλάσεις για την διαχείριση των σφαλμάτων (errors) με χρήση προσαρμοσμένων εξαιρέσεων (exceptions) που κατασκευάστηκαν από τον δημιουργό της παρούσης (Εικόνα 3.13).



Εικόνα 3.13 Δομή υποφακέλου exceptions στο VS Code

3.1.8 Υποφάκελος filter

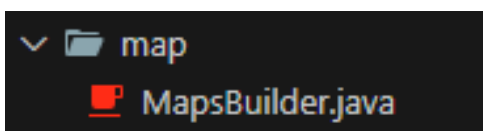
Περιέχει κλάση για τον έλεγχο των δεδομένων χρήστη κατά την διάρκεια σύνδεσης αυτού είτε στον διακομιστή ιστού (web server) είτε στην διεπαφή προγραμματισμού εφαρμογών (API) (Εικόνα 3.14). Η Κλάση αυτή ελέγχει τα δεδομένα του χρήστη (credentials) αποκρυπτογραφώντας το JWT που απέστειλε αυτός.



Εικόνα 3.14 Δομή υποφακέλου filter στο VS Code

3.1.9 Υποφάκελος map

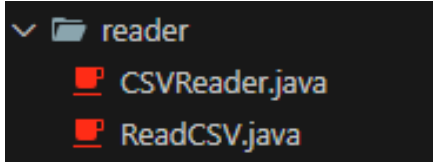
Περιέχει κλάση η οποία μετατρέπει τα δεδομένα ενός .csv αρχείου σε δομή δεδομένων json (HashMap της γλώσσας προγραμματισμού java) (Εικόνα 3.15).



Εικόνα 3.15 Δομή υποφακέλου map στο VS Code

11.1.1 Υποφάκελος reader

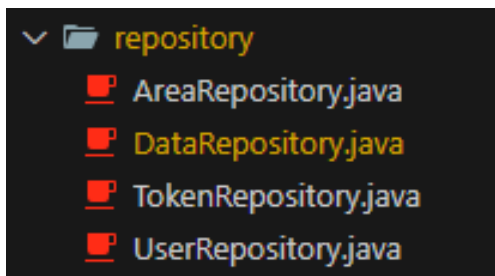
Περιέχει κλάσεις για το κατέβασμα (download) και το διάβασμα αρχείων .csv με τα δεδομένα της ασθένειας COVID-19 (Εικόνα 3.16).



Εικόνα 3.16 Δομή υποφακέλου reader στο VS Code

3.1.10 Υποφάκελος repository

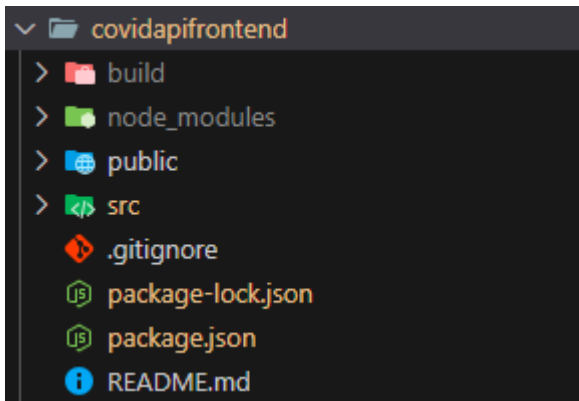
Περιέχει βοηθητικές κλάσεις για που περιέχουν ερωτήσεις (queries) για την διαχείριση των δεδομένων της βάσης δεδομένων (Εικόνα 3.17).



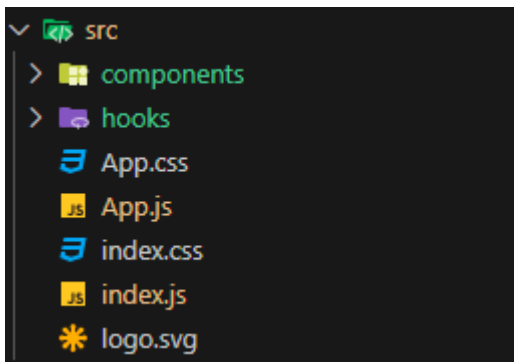
Εικόνα 3.17 Δομή υποφακέλου repository στο VS Code

3.2 Υποφάκελος covidapifrontend (React.js)

Ο υποφάκελος covidapifrontend περιέχει τα αρχεία και τους φακέλους που δημιουργήθηκαν αυτόματα στην γλώσσα προγραμματισμού Javascript με χρήση της βιβλιοθήκης αυτής, που ονομάζεται React.js. Για την αυτόματη δημιουργία των φακέλων αυτών εκτελέστηκε η εντολή της βιβλιοθήκης React.js `npm create-react-app covidapifrontend`. Στον υποφάκελο αυτό υπάρχουν τέσσερις υποφάκελοι και τέσσερα αρχεία (Εικόνα 3.18). Ο τελευταίος υποφάκελος με ονομασία src (Εικόνα 3.19) είναι αυτός που χρησιμοποιήθηκε από τον δημιουργό της παρούσας για την υλοποίηση και κατασκευή διαδραστικών και δυναμικών διεπαφών χρήστη (UI). Ο υποφάκελος αυτός περιέχει 2 υποφακέλους με ονόματα components και hooks τα οποία κατασκευάστηκαν από τον δημιουργό της παρούσας εργασίας. Στον φάκελο components δημιουργήθηκαν όλα τα απαραίτητα αρχεία της διεπαφής χρήστη ενώ στον φάκελο hooks τα απαραίτητα αρχεία για την επικοινωνία της διεπαφής χρήστη(UI) με τον διακομιστή ιστού(server).



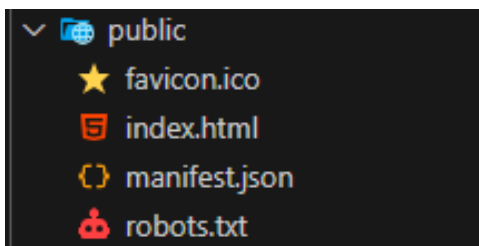
Εικόνα 3.18 Δομή υποφακέλου covidapifrontend στο VS Code



Εικόνα 3.19 Δομή υποφακέλου src στο VS Code

3.2.1 Υποφάκελος public

Ο φάκελος αυτός περιέχει τα απαραίτητα αρχεία της React.js που θα στείλει ο διακομιστής ιστού (server) στις ιστοσελίδες που δημιουργήθηκαν ως διεπαφή χρήστη (Εικόνα 3.20). Περιέχει το κεντρικό .html αρχείο της React.js, το εικονίδιο της ιστοσελίδας, όπως επίσης και τις γραμματοσειρές αυτής.



Εικόνα 3.20 Δομή υποφακέλου public στο VS Code

3.2.2 Υποφάκελος node_modules

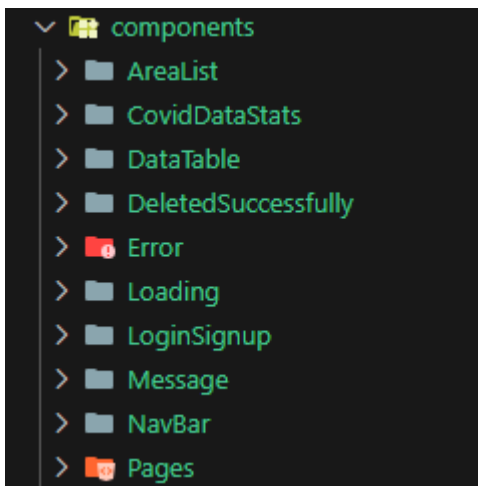
Ο φάκελος αυτός περιέχει όλες τις απαραίτητες βιβλιοθήκες της γλώσσας προγραμματισμού JavaScript για την εύρυθμη λειτουργία της βιβλιοθήκης React.js και κατέβηκαν (download) αυτόματα από αυτήν.

3.2.3 Αρχείο package.json

Το αρχείο αυτό περιέχει τα ονόματα και τις αντίστοιχες εκδόσεις των βιβλιοθηκών (dependencies) της γλώσσας προγραμματισμού JavaScript για την εύρυθμη λειτουργία της βιβλιοθήκης React.js σε μορφή json. Το αρχείο αυτό μπορεί να χρησιμοποιηθεί για την εγκατάσταση όλων των απαραίτητων βιβλιοθηκών της React.js σε οποιονδήποτε υπολογιστή που περιέχει την γλώσσα προγραμματισμού JavaScript (node.js) με εκτέλεση της εντολής **npm install** αυτής. Αυτόματα θα κατέβουν και θα εγκατασταθούν οι βιβλιοθήκες της React.js στον υποφάκελο node_modules.

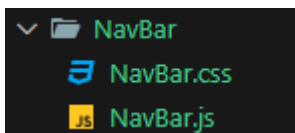
3.2.4 Υποφάκελος components

Ο φάκελος αυτός κατασκευάστηκε από τον δημιουργό της παρούσας και περιέχει όλα τα απαραίτητα αντικείμενα Html καθώς επίσης και της ιστοσελίδες που υλοποιούν την διεπαφή χρήστη (Εικόνα 3.21). Οι ιστοσελίδες αυτές συνθέτονται από τα αντικείμενα Html (components). Τόσο τα αντικείμενα Html, όσο και οι ίδιες οι ιστοσελίδες είναι συναρτήσεις της γλώσσας προγραμματισμού JavaScript, οι οποίες με χρήση της βιβλιοθήκης React.js επιστρέφουν αντικείμενα που συνδυάζουν HTML και την γλώσσα αυτή.



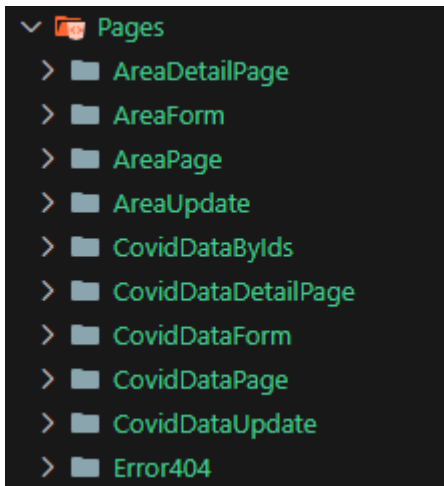
Εικόνα 3.21 Δομή υποφακέλου components στο VS Code.

Όλοι υποφάκελοι του φακέλου components περιέχουν δυο αρχεία (Εικόνα 3.22). Ένα αρχείο .css που περιέχει τις απαραίτητες οδηγίες για τον καλλωπισμό του αντικείμενου html και ένα .js που περιέχει την συνάρτηση που δημιουργεί και επιστρέφει το αντικείμενο αυτό.



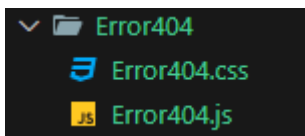
Εικόνα 3.22 Παράδειγμα δομής υποφακέλου ενός component στο VS Code.

Στον υποφάκελο Pages του φακέλου components (Εικόνα 3.23) περιέχονται όλες οι ιστοσελίδες που υλοποιούν την διεπαφή χρήστη της παρούσας εργασίας.



Εικόνα 3.23 Δομή υποφακέλου Pages στο VS Code.

Κάθε μια από αυτές τις ιστοσελίδες υλοποιείται μέσα σε ξεχωριστό φάκελο ο οποίος περιέχει δύο αρχεία (Εικόνα 3.24). Ένα αρχείο .css που περιέχει τις απαραίτητες οδηγίες για τον καλλωπισμό της ιστοσελίδας και ένα .js που περιέχει την συνάρτηση που δημιουργεί και επιστρέφει την ιστοσελίδα αυτή.



Εικόνα 3.24 Παράδειγμα δομής υποφακέλου ιστοσελίδας στο VS Code.

Οι ιστοσελίδες αυτές δημιουργούνται από σύνθεση html, javascript και αντικειμένων html (components) που δημιουργήθηκαν από τον δημιουργό της παρούσας. Τα αντικείμενα html (components) εισάγονται στην ιστοσελίδα με χρήση Javascript (Εικόνα 3.25) και συνδυάζονται με την γλώσσα αυτή (Εικόνα 3.26) για την δημιουργία της ιστοσελίδας.

```
import AreaList from "../../AreaList/AreaList"
```

Εικόνα 3.25 Παράδειγμα εισαγωγής αντικειμένου html (component) σε ιστοσελίδα.

```
{area_list && <AreaList area_list={area_list} />}
```

Εικόνα 3.26 Παράδειγμα συνδυασμού αντικειμένου html (component) και javascript.

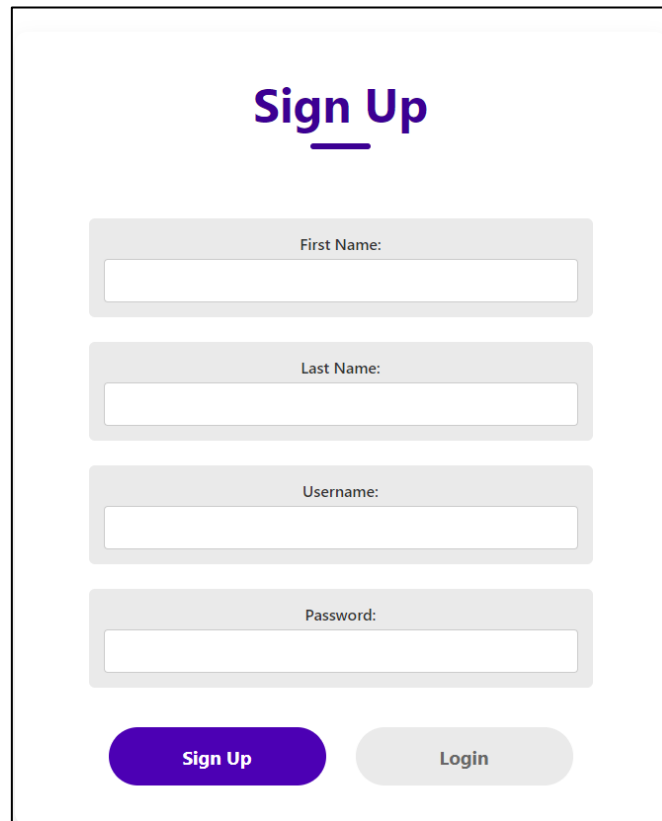
Σε όλες τις ιστοσελίδες έχει εισαχθεί το αντικείμενο html (component) NavBar (Εικόνα 3.27) για την σωστή πλοήγηση του χρήστη στις ιστοσελίδες



Εικόνα 3.27 Αντικείμενο html (component) NavBar σε πλοηγό (browser).

Όλες οι ιστοσελίδες όπως και η βάση δεδομένων προστατεύονται με σύστημα εισόδου (login system) (Εικόνα 3.28). Όταν ο χρήστης προσπαθήσει να προηγηθεί στο site της παρούσας εργασίας θα πρέπει οπωσδήποτε να συνδεθεί σε αυτό ώστε να αποκτήσει μοναδικό JWT (JSON Web Token) για να μπορεί να αλληλεπιδρά με τον διακομιστή ιστού (server). Αν ο χρήστης δεν είναι εγγεγραμμένος ώστε να έχει τους απαραίτητους κωδικούς, τότε μπορεί να εγγραφεί σε αυτό (Εικόνα 3.29).

Εικόνα 3.28 Σύστημα εισόδου (login system) σε πλοηγό (browser).

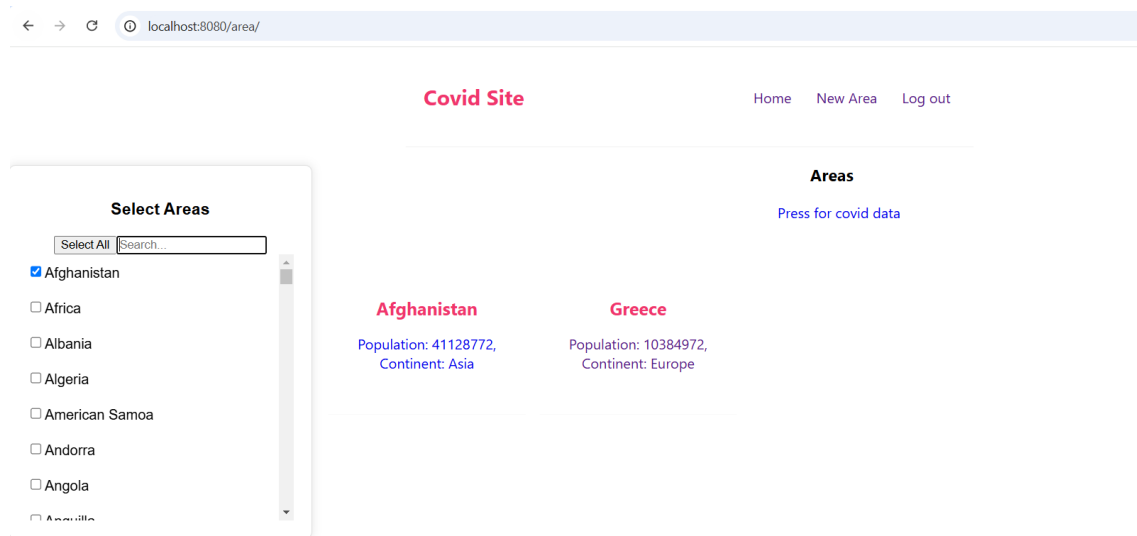


The image shows a web form titled "Sign Up" in a purple font. Below the title are four input fields, each with a label above it: "First Name:", "Last Name:", "Username:", and "Password:". At the bottom of the form, there are two buttons: a purple button labeled "Sign Up" and a light gray button labeled "Login".

Εικόνα 3.29 Εγγραφή στο σύστημα εισόδου (login system) σε πλοηγό (browser).

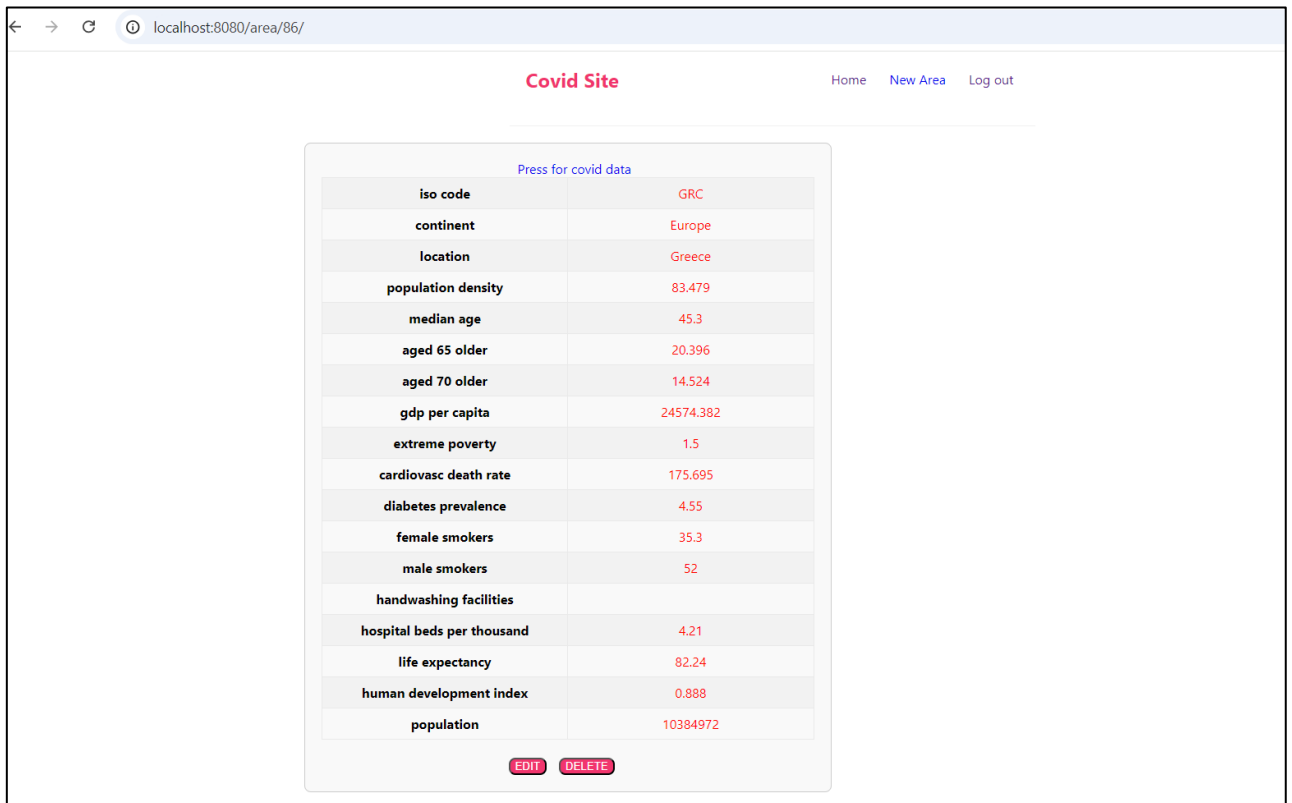
Το σύστημα εισόδου είναι κατασκευασμένο με τέτοιο τρόπο ώστε να παρέχει στους χρήστες δυο ρόλους, τον ρόλο του διαχειριστή (ADMIN) ή τον ρόλο του απλού χρήστη (USER). Μέσα από το σύστημα εγγραφής της διεπαφής χρήστη (UI) (Εικόνα 3.29), δίνεται η δυνατότητα εγγραφής μόνο ως απλός χρήστης, ενώ η δημιουργία καινούργιου διαχειριστή επιτυγχάνεται μόνο από άλλο διαχειριστή απευθείας στην βάση δεδομένων, χωρίς χρήση διεπαφής χρήστη. Οι δύο ρόλοι περιέχουν στον χρήστη διαφορετικές δυνατότητες. Στον ρόλο του απλού χρήστη δίνεται η δυνατότητα από το σύστημα μόνο να μπορεί να παρατηρεί (get) τα δεδομένα χωρίς να μπορεί να τα επεξεργαστεί, ενώ στον διαχειριστή πέρα από την παρατήρηση των δεδομένων επιτρέπεται και η επεξεργασία αυτών. Ο διαχειριστής μπορεί να προσθέτει δεδομένα (insert), μπορεί να διαγράφει δεδομένα (delete) ή μπορεί να αναβαθμίζει δεδομένα (update) στην βάση δεδομένων.

Αφότου ο χρήστης συνδεθεί στο σύστημα εισόδου έχει την δυνατότητα να πλοηγηθεί στις ιστοσελίδες αυτού. Αρχικά το σύστημα τον κατευθύνει στην ιστοσελίδα των περιοχών (Εικόνα 3.30) για τις οποίες περιέχονται δεδομένα για την ασθένεια COVID-19. Στην ιστοσελίδα αυτή δίνεται η δυνατότητα στον χρήστη να δει όλες τις περιοχές της βάσης δεδομένων, ενώ με την χρήση φίλτρων μπορεί να επιλέξει συγκεκριμένες περιοχές. Στην συγκεκριμένη ιστοσελίδα ο χρήστης μπορεί να δει βασικά στοιχεία της κάθε περιοχής (πληθυσμός, ήπειρος). Μετά την επιλογή περιοχών (όλων ή κάποιων) ο χρήστης μπορεί να ανακατευθυνθεί σε επόμενη ιστοσελίδα, στην οποία μπορεί να δει τα επιπλέον στοιχεία της κάθε περιοχής.



Εικόνα 3.30 Ιστοσελίδα περιοχών

Στην ιστοσελίδα της κάθε περιοχής παρέχονται τα επιπλέον δεδομένα της βάσης δεδομένων για την κάθε περιοχή (Εικόνα 3.31). Στην ιστοσελίδα αυτή δίνονται επιπλέον στον διαχειριστή οι επιλογές της διαγραφής μιας περιοχής και της επεξεργασίας αυτής. Μέσω αυτής της ιστοσελίδας ο χρήστης μπορεί να ανακατευθυνθεί σε επόμενη ιστοσελίδα, στην οποία μπορεί να δει δεδομένα της περιοχής αυτής για την ασθένεια COVID-19. Αν ο διαχειριστής επιλέξει να επεξεργαστεί τα δεδομένα μιας περιοχής θα ανακατευθυνθεί σε επόμενη ιστοσελίδα (Εικόνα 3.32).



localhost:8080/area/86/

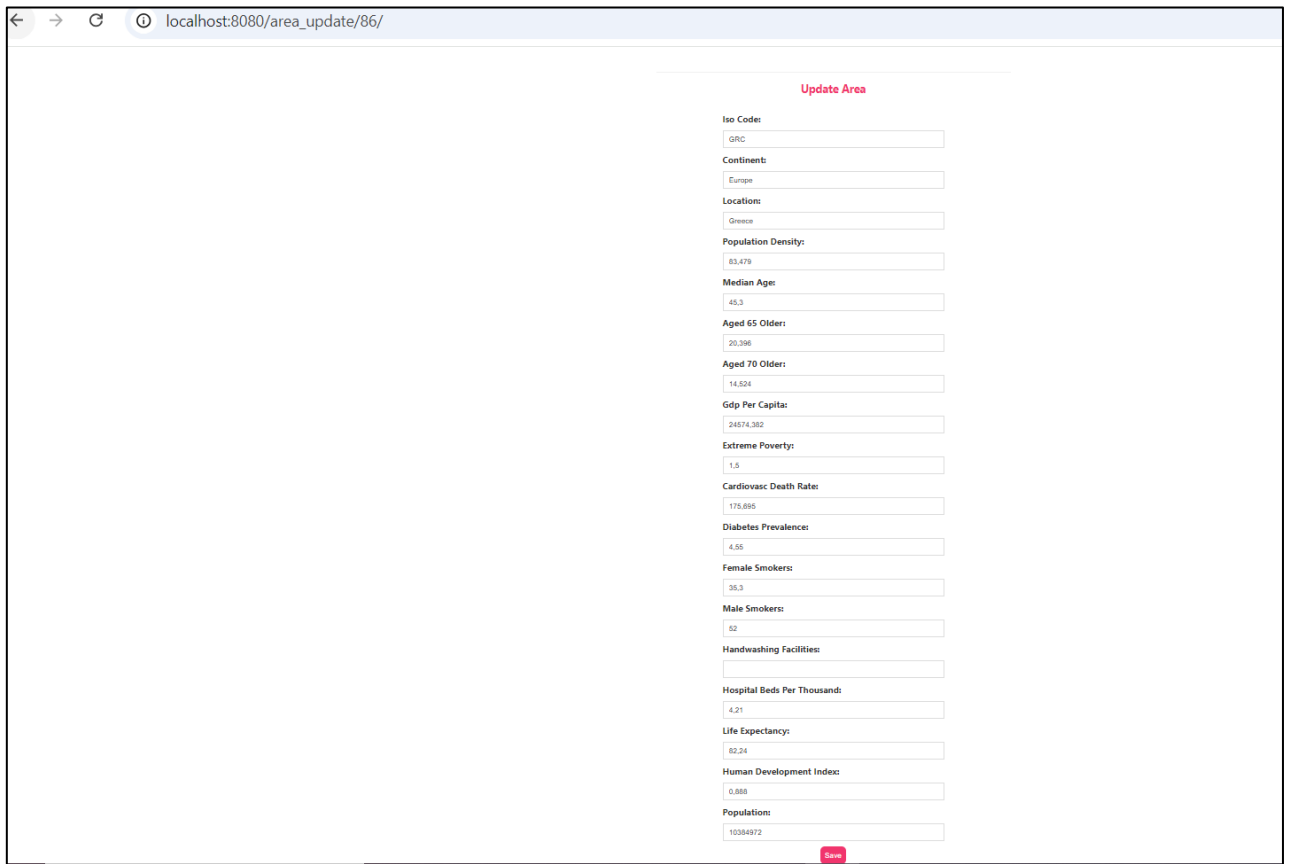
Covid Site Home New Area Log out

Press for covid data

iso code	GRC
continent	Europe
location	Greece
population density	83.479
median age	45.3
aged 65 older	20.396
aged 70 older	14.524
gdp per capita	24574.382
extreme poverty	1.5
cardiovasc death rate	175.695
diabetes prevalence	4.55
female smokers	35.3
male smokers	52
handwashing facilities	
hospital beds per thousand	4.21
life expectancy	82.24
human development index	0.888
population	10384972

EDIT DELETE

Εικόνα 3.31 Παράδειγμα ιστοσελίδας περιοχής για τον διαχειριστή του συστήματος.

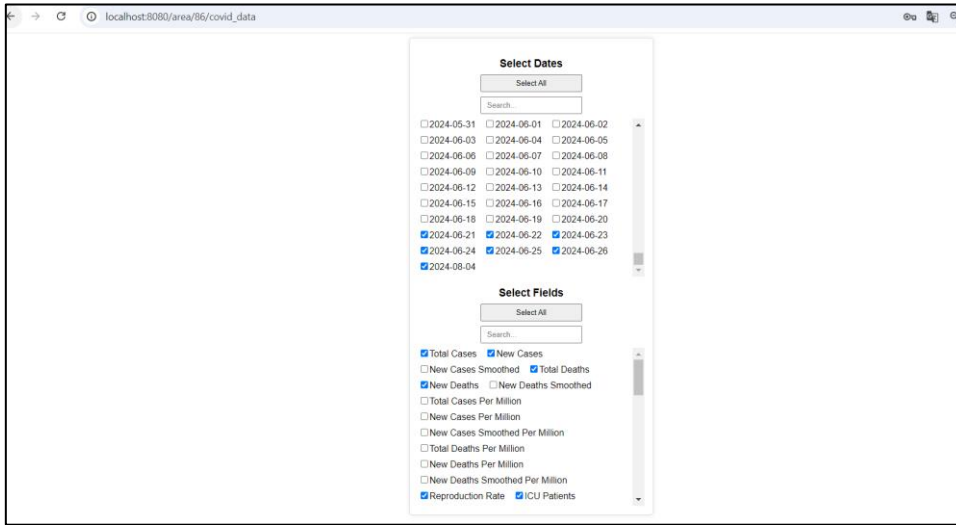


Εικόνα 3.32 Παράδειγμα ιστοσελίδας επεξεργασίας περιοχής για τον διαχειριστή του συστήματος.

Στην ιστοσελίδα των δεδομένων της ασθένειας COVID-19 για την κάθε περιοχή παρουσιάζονται αυτά σε μορφή πίνακα (Εικόνα 3.33). Επιπλέον δίνεται η δυνατότητα στον χρήστη με χρήση φίλτρων να επιλέξει τις ημερομηνίες των δεδομένων αυτών, όπως επίσης και ποια στοιχεία από αυτά θέλει να εμφανίζονται στον πίνακα (Εικόνα 3.34). Αν ο χρήστης έχει τον ρόλο του διαχειριστή μπορεί μέσω του πίνακα να ανακατευθυνθεί σε επόμενη ιστοσελίδα επεξεργασίας των δεδομένων (Εικόνα 3.35).

Greece																		
Create New Covid Data																		
Date	Total Cases	New Cases	Total Deaths	New Deaths	Reproduction Rate	ICU Patients	Hospital Patients	Weekly ICU Admissions	Weekly Hospital Admissions	Total Tests	New Tests	Total Vaccinations	People Vaccinated	People Fully Vaccinated	Total Boosters	New Vaccinations	Excess Mortality	Edit
2024-06-21												22467930	7938027	7647076	7461900	21		EDIT
2024-06-22																		EDIT
2024-06-23																		EDIT
2024-06-24																		EDIT
2024-06-25												22467970	7938030	7647076	7461937			EDIT
2024-06-26												22467978	7938031	7647076	7461944	8		EDIT
2024-08-04	5673681	5818	39258	40														EDIT

Εικόνα 3.33 Παράδειγμα ιστοσελίδας δεδομένων COVID-19 περιοχής για τον διαχειριστή του συστήματος.

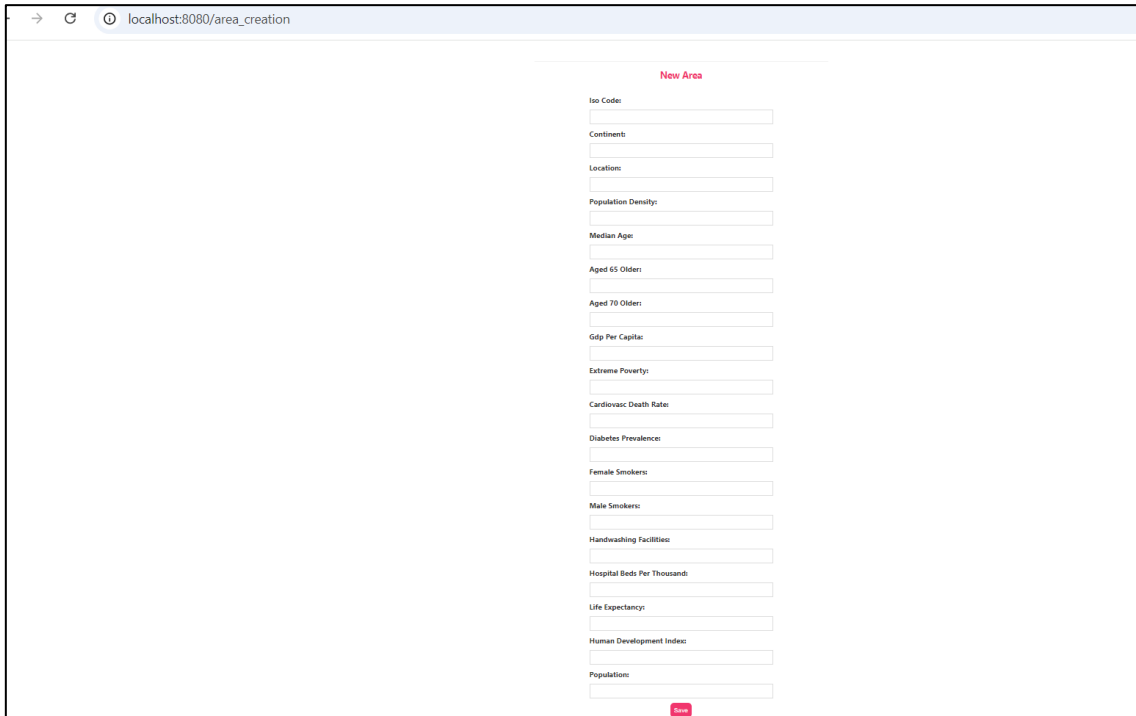


Εικόνα 3.34 Φίλτρα ιστοσελίδας δεδομένων COVID-19 περιοχής.

hosp patients	
hosp patients per million	
weekly icu admissions	
weekly icu admissions per million	
weekly hosp admissions	
weekly hosp admissions per million	
total tests	
new tests	
total tests per thousand	
new tests per thousand	
new tests smoothed	
new tests smoothed per thousand	
positive rate	
tests per case	
tests units	
total vaccinations	2247930
people vaccinated	7938027
people fully vaccinated	7647076
total boosters	7461900
new vaccinations	21
new vaccinations smoothed	14
total vaccinations per hundred	216.35
people vaccinated per hundred	76.44
people fully vaccinated per hundred	73.64
total boosters per hundred	71.85
new vaccinations smoothed per million	1
new people vaccinated smoothed	3
new people vaccinated smoothed per hundred	0
stringency index	
excess mortality cumulative absolute	
excess mortality cumulative	
excess mortality	
excess mortality cumulative per million	

Εικόνα 3.35 Παράδειγμα ιστοσελίδας επεξεργασίας δεδομένων COVID-19 περιοχής για τον διαχειριστή του συστήματος.

Τέλος ο διαχειριστής μέσω του αντικειμένου Html NavBar(Εικόνα 3.27) έχει την δυνατότητα να ανακατευθυνθεί σε ιστοσελίδα εισαγωγής καινούριας περιοχής και των δεδομένων αυτής (Εικόνα 3.36).



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/area_creation'. The main content area features a form titled 'New Area' with a red header. The form contains the following fields:

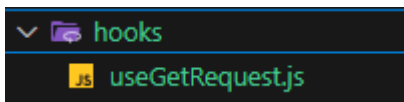
- Iso Codes:
- Continent:
- Locations:
- Population Density:
- Median Age:
- Aged 65 Older:
- Aged 70 Older:
- Gdp Per Capita:
- Extreme Poverty:
- Cardiovascular Death Rates:
- Diabetes Prevalence:
- Female Smokers:
- Male Smokers:
- Handwashing Facilities:
- Hospital Beds Per Thousand:
- Life Expectancy:
- Human Development Index:
- Population:

A red 'Save' button is located at the bottom right of the form.

Εικόνα 3.36 Παράδειγμα ιστοσελίδας εισαγωγής δεδομένων περιοχής για τον διαχειριστή του συστήματος.

3.2.5 Υποφάκελος hooks

Ο φάκελος αυτός περιέχει τα απαραίτητα «αγκίστρια» (hooks) της βιβλιοθήκης React.js για την διαχείριση των αιτημάτων για δεδομένα των ιστοσελίδων στον διακομιστή ιστού (server) (Εικόνα 3.37).



Εικόνα 3.37 Δομή υποφακέλου hooks στο VS Code.

3.2.6 Αρχείο App.js και δρομολόγηση (Routing)

Το αρχείο αυτό είναι το κεντρικό αρχείο της βιβλιοθήκης React.js. Η βιβλιοθήκη React.js αναλαμβάνει και την δρομολόγηση των ιστοσελίδων της παρούσας εργασίας. Η React.js δίνει την δυνατότητα αποστολής ταυτοχρόνως όλων των ιστοσελίδων στην πρώτη επίσκεψη του χρήστη και έπειτα η ίδια κάνει την δρομολόγηση χωρίς να χρειάζεται επικοινωνία των ιστοσελίδων με τον διακομιστή ιστού. Με τον τρόπο αυτό η ταχύτητα δρομολόγησης πέφτει στο ελάχιστο. Η

δρομολόγηση των ιστοσελίδων επιβάλλεται από την React.js να γίνει στο κεντρικό αντικείμενο html (component) αυτής(Εικόνα 3.38).

```
return (
  <div className="App">
    <Router>
      {token && <NavBar setToken={setToken}/>}
      <Routes>
        <Route path="/" element={<Navigate to="/login/">}/>/>
        <Route path="/login/" element={<LoginSignup setToken={setToken}/>}/>
        <Route path="/area/" element={<AreaPage token={token}/>}/>
        <Route path="/area/:area_id/" element={<AreaDetailPage token={token} />}/>
        <Route path="/area_creation/" element={<AreaForm area={null} token={token}/>}/>
        <Route path="/area_update/:area_id/" element={<AreaUpdate token={token}/>}/>
        <Route path="/area/:ids/covid_data/" element={<CovidDataByIds token={token} />}/>
        <Route path="/area/:area_id/covid_data/" element={<CovidDataPage token={token} />}/>
        <Route path="/area/:area_id/covid_data/:covid_data_id/" element={<CovidDataDetailPage token={token} />}/>
        <Route path="/area/:area_id/covid_data_creation/" element={<CovidDataForm covid_data={null} token={token}/>}/>
        <Route path="/area/:area_id/covid_data_update/:covid_data_id/" element={<CovidDataUpdate token={token} />}/>
      </Routes>
    </Router>
  </div>
)
```

Εικόνα 3.38 Κεντρικό αντικείμενο html (component) και δρομολόγηση ιστοσελίδων (routing) στο VS Code.

4. Συμπεράσματα και Προοπτικές

4.1 Συμπεράσματα επί των αποτελεσμάτων

Η παρούσα εργασία κατάφερε να συλλέξει και να αυτοματοποιήσει την συλλογή δεδομένων για την ασθένεια COVID-19 για διάφορες περιοχές του πλανήτη. Τα δεδομένα αυτά τοποθετήθηκαν σε βάση δεδομένων. Η βάση δεδομένων προστατεύτηκε με σύστημα εισόδου δύο ρόλων (διαχειριστή, απλού χρήστη) το οποίο υλοποιήθηκε με διεπαφή προγραμματισμού εφαρμογών (API). Στον διακομιστή ιστού μετέπειτα προστέθηκε διεπαφή χρήστη (UI) προστατευμένη από το σύστημα εισόδου για την ομορφότερη παρουσίαση των δεδομένων, καθώς και την ευκολότερη επεξεργασία αυτών από τους διαχειριστές του συστήματος.

4.2 Συμπεράσματα επί των εργαλείων

Πριν την εκκίνηση κατασκευής της παρούσας εργασίας ο δημιουργός της επέλεξε εργαλεία που το καθένα από αυτά είναι ένα από τα καλύτερα στο είδος του. Τα εργαλεία αυτά όχι απλά ανταποκρίθηκαν στις ανάγκες του δημιουργού, αλλά έχουν δυνατότητες για επίλυση περιπλοκότερων προβλημάτων από αυτά που επιλύθηκαν στην παρούσα.

4.3 Προοπτικές / Επόμενα βήματα

Το πρόγραμμα (διακομιστής ιστού) που δημιουργήθηκε στην παρούσα εργασία μπορεί να επαναχρησιμοποιηθεί από οποιονδήποτε, σε οποιονδήποτε ηλεκτρονικό υπολογιστή που έχει εγκατεστημένη την γλώσσα προγραμματισμού Java και τις αντίστοιχες βιβλιοθήκες αυτής.

Ο διακομιστής ιστού θα μπορούσε να μεταφορτωθεί (upload) στο υπολογιστικό νέφος (cloud), έτσι ώστε να έχει πρόσβαση στις ιστοσελίδες του, ο οποιοσδήποτε, από οποιοδήποτε μέρος του κόσμου.

Για την βελτίωση της παρουσίασης των δεδομένων θα μπορούσε να είχαν δημιουργηθεί διαγράμματα μέσα στις ιστοσελίδες της παρουσίασης, έτσι ώστε ο χρήστης να μπορεί να επιτύχει καλύτερη σύγκριση των δεδομένων αυτών.

Τέλος με χρήση με χρήση τεχνικών μηχανικής μάθησης (machine learning), τομέα της τεχνητής νοημοσύνης (AI), τα δεδομένα που συλλέγονται από την παρούσα θα μπορούσαν να χρησιμοποιηθούν για την πρόβλεψη της εξέλιξης της ασθένειας COVID-19, ανά περιοχή ή για ολόκληρο τον πλανήτη.

5. Βιβλιογραφία

- Duckett, J. (n.d.). *HTML and CSS: Design and Build Websites*. Wiley.
- Garsiel, T., & Irish, P. (n.d.). *How Browsers Work: Behind the scenes of modern web browsers*. Retrieved from <https://www.html5rocks.com/en/tutorials/internals/howbrowserswork/>
- Geewax, J. J. (2021). *API Design Patterns*. Manning Publications.
- Schade, M. H. W. (2020). *JWT Handbook*. Leanpub.
- Silberschatz, A., Korth, H., & Sudarshan, S. (2020). *Database System Concepts* (7th ed.). McGraw-Hill.
- Forta, B. (2012). *SQL in 10 Minutes, Sams Teach Yourself*. Sams Publishing.
- Bloch, J. (2018). *Effective Java* (3rd ed.). Addison-Wesley.
- Walls, C. (2015). *Spring Boot in Action*. Manning Publications.
- Duckett, J. (2011). *HTML & CSS: Design and Build Websites*. Wiley.
- McFarland, D. S. (2015). *CSS: The Missing Manual* (4th ed.). O'Reilly Media.
- Haverbeke, M. (2018). *Eloquent JavaScript* (3rd ed.). No Starch Press.
- Banks, A., & Porcello, E. (2020). *Learning React: Functional Web Development with React and Redux* (2nd ed.). O'Reilly Media.
- T. J. (n.d.). *Web Server Design*. Retrieved from <https://example.com>
- Wikipedia contributors. (1991). *HTML*. Wikipedia, The Free Encyclopedia. Retrieved from <https://en.wikipedia.org/wiki/HTML>
- Wikipedia contributors. (1995). *JavaScript*. Wikipedia, The Free Encyclopedia. Retrieved from <https://en.wikipedia.org/wiki/JavaScript>
- Wikipedia contributors. (1996). *CSS*. Wikipedia, The Free Encyclopedia. Retrieved from <https://en.wikipedia.org/wiki/CSS>
- Wikipedia contributors. (2013). *Our World in Data*. Wikipedia, The Free Encyclopedia. Retrieved from https://en.wikipedia.org/wiki/Our_World_in_Data

Ιστοσελίδα καταγραφής δεδομένων πανδημίας, υλοποιημένη με mvc αρχιτεκτονική και spring boot framework

- Wikipedia contributors. (n.d.). *Web browser*. Wikipedia, The Free Encyclopedia. Retrieved from https://en.wikipedia.org/wiki/Web_browser
- Wikipedia contributors. (n.d.). *Web page*. Wikipedia, The Free Encyclopedia. Retrieved from https://en.wikipedia.org/wiki/Web_page
- Wikipedia contributors. (n.d.). *API*. Wikipedia, The Free Encyclopedia. Retrieved from <https://en.wikipedia.org/wiki/API>
- Wikipedia contributors. (n.d.). *Database*. Wikipedia, The Free Encyclopedia. Retrieved from <https://en.wikipedia.org/wiki/Database>
- Wikipedia contributors. (n.d.). *Java (programming language)*. Wikipedia, The Free Encyclopedia. Retrieved from [https://en.wikipedia.org/wiki/Java_\(programming_language\)](https://en.wikipedia.org/wiki/Java_(programming_language))
- Wikipedia contributors. (n.d.). *JSON Web Token*. Wikipedia, The Free Encyclopedia. Retrieved from https://en.wikipedia.org/wiki/JSON_Web_Token
- Wikipedia contributors. (n.d.). *React (JavaScript library)*. Wikipedia, The Free Encyclopedia. Retrieved from [https://en.wikipedia.org/wiki/React_\(JavaScript_library\)](https://en.wikipedia.org/wiki/React_(JavaScript_library))
- Wikipedia contributors. (n.d.). *Spring Boot*. Wikipedia, The Free Encyclopedia. Retrieved from https://en.wikipedia.org/wiki/Spring_Boot
- Wikipedia contributors. (n.d.). *SQL*. Wikipedia, The Free Encyclopedia. Retrieved from <https://en.wikipedia.org/wiki/SQL>
- Douladiris, Anargyros, and Efthimios Alepis. 2023. "Covid-19 New Cases Correlation Analysis: Weather Conditions, Citizen Traffic and Vaccination Statistics Impact in NARX Estimated Regressions in Attica, Greece." In 2023 14th International Conference on Information, Intelligence, Systems & Applications (IISA), 1–7. IEEE
- Matzavela, Vasiliki, and Efthimios Alepis. 2021. "M-Learning in the COVID-19 Era: Physical Vs Digital Class." *Education and Information Technologies* 26 (6): 7183–203.
- Virvou, Maria, Efthimios Alepis, George A Tsihrintzis, and Lakhmi C Jain. 2020. "Machine Learning Paradigms: Advances in Learning Analytics." *Machine Learning Paradigms: Advances in Learning Analytics*, 1–5.
- Triantafyllou, Andreas M, George A Tsihrintzis, Maria Virvou, and Efthimios Alepis. 2021. "A Bimodal System for Emotion Recognition via Computer of Known or Unknown Persons in Normal or Fatigue Situations." In *Advances in Core Computer Science-Based Technologies*, 9–35. Springer, Cham.
- Maria Virvou, George A. Tsihrintzis, Dionisios N. Sotiropoulos, Konstantina Chrysafiadi, Evangelos Sakkopoulos, Evangelia-Aikaterini Tsihrintzi: "ChatGPT in Artificial Intelligence-Empowered E-Learning for Cultural Heritage: The case of Lyrics and Poems." In 2023 14th International Conference on Information, Intelligence, Systems & Applications (IISA), 1-9.
- Konstantina Chrysafiadi, Maria Virvou, Evangelos Sakkopoulos: "Optimizing Programming Language Learning Through Student Modeling in an Adaptive Web-Based Educational Environment." *Machine Learning Paradigms 2020*: 205-223
- Margaritis Kamitsios, Konstantina Chrysafiadi, Maria Virvou, Evangelos Sakkopoulos: "A Stereotype User Model for an Educational Game: Overcome the Difficulties in Game Playing and Focus on the Educational Goal." *IISA 2018*: 1-6