ALEXANDROS SAVVIDIS

6986159391 | alexandrossavvidis2@gmail.com | linkedin.com/in/asavvidi | github.com/asavvidi | asavvidi.netlify.app

EDUCATION

AUTH (Aristotle University of Thessaloniki)   Thessaloniki
Bachelor in Computer Science   *September 2018 – July 2024*

Bachelor Thesis   Thessaloniki
Dynamic Programming Algorithms and Applications in Bioinformatics   *February 2024 – July 2024*

TECHNICAL SKILLS

Languages: HTML, CSS, JavaScript, Java, Python, SQL

Frontend Development: React.js, React Router, Redux

Backend Development: Node.js, PostgreSQL, Spring Boot

Frameworks: React, Express, Spring Boot

Developer Tools: Git, VS Code, IntelliJ

PROJECTS

Stock Market Tracker App | *Node.js, Express, EJS, Axios, Chart.js*
- Developed a real-time stock market tracker app to display stock data using EJS for Server Side Rendering and Express for backend development.
- Used an external API to fetch stock and news data including open and close prices, percentage changes and financial news.
- Defined custom API endpoints using Express for frontend requests using Axios.

MyFinanceTracker | *React, Node.js, Express, Sequelize, PostgreSQL, JWT, Chart.js*
- Created a finance tracker app using React for the view layout, PostgreSQL for database storage, Sequelieze to connect the database with the server and JWT for authentication.
- Allow users to register and log in securely using JWT authentication and enable them to add, manage, and delete income and expense transactions and provide a graphical representation of financial activity using Chart.js.
- Defined custom API endpoints in Express to handle user authentication and financial transactions.

Brainaire Quiz App | *React, Context API, JSON Server*
- Built a quiz app using React to create the User Interface and Context API with useReducer for global state management.
- Fetch and display multiple-choice questions dynamically from a JSON server.
- Track scores, use countdown timer, and store high scores for competitive gameplay.

R-Tree Spatial Indexing System | *Java, R-tree*
- Developed a high-performance spatial indexing system using the R*-Tree algorithm to efficiently handle multidimensional range and nearest-neighbor queries.
- Implemented spatial data parsing from OpenStreetMap XML files, including data cleaning and transformation into the appropriate format for insertion into the R*-Tree structure.
- Developed multiple spatial query functions including range queries, k-nearest neighbor (kNN) queries, and skyline queries, supporting complex geographic searches.