OLENA DANYKH

Software Engineer | New York City| (929) 257-8739|olenadanykh@gmail.com | github.com/olenadanykh | linkedin.com/in/olena-danykh

TECHNICAL SKILLS

Strong: JavaScript (ES6+), React.js, React Router, Node.js, Express, SQL - PostgresQL, NoSQL - MongoDB, GraphDB - Neo4j, RESTful Architecture, Typescript, OOP & Functional Programming, HTML/CSS, Version Control (Git/GitHub), Webpack, Agile/Scrum Development, Cookies, Postman.
Experienced: Redux, Authentication (Auth/OAuth), Testing Frameworks (Jest, Enzyme, Supertest), GraphQL, Microservices architecture, GRPC.

EXPERIENCE

OSLabs/Trinity | Software Engineer

Neo4j VS Code Extension for Cypher Query Testing and Graph Structure Exploration *2019 - Present*

- Utilized VSCode API to develop an extension which allows developers perform real-time Cypher queries and quickly identify all graph relationships and properties using VSCode Explorer by removing the need for switching between VS Code and Neo4j Browser during the development process.
- Integrated TypeScript by enforcing type annotations optimize the debugging process, providing consistent cross-development team code generation.
- Wrote a recursive search algorithm to extract all Cypher queries, determining location and position of import declarations to identify possible errors.
- Constructed configuration file to store a user's Neo4j database credentials, persisting custom user views and eliminating the need of continuous login.
- Implemented Promises to write the queries output from VS Code Channel to a JSON file, providing intuitive, more readable code with modularity.
- Customized Webpack configurations to run all assets, bundling files and dependencies to create a fluid development experience and debugging process.
- Integrated Mocha testing into an existing codebase, ensuring a properly working VS Code extension, verifying correct information from the Neo4j database, displaying on the outline and sending to the output Channel to minimize the debugging of new features, and package functionality for engineers.

OPEN SOURCE

Budget Bulldog |*Expense Tracker app*

- Leveraged Redux-based architecture to maintain the state of each user's expense, improving scalability by dynamically updating components with predictable state mutations based on user actions and data flow, creating standardized structure across codebase.
- Implemented React-Router to improve user experience by engineering multiple front-end routes, allowing seamless navigation between views, making sure that only authenticated users have access to their saved content.
- Developed Node.js server with express middleware layer using RESTful architecture to process, store and display relevant user data from SQL database.
- Implemented Redux-Thunk middleware to handle asynchronous function calls, by dispatching actions and managing side-effects caused by asynchronous API calls in order to delay state changes and encourage application efficiency.
- Employed Bcrypt to encrypt user data and saved hashed information to an SQL database to enhance security and implement robust authentication.
-

Stock Visualizer | *GUI based stock visualizer for daily stock data*

- Leveraged React.js to design an interactive UI, maintaining DRY principles through unidirectional data flow and reusable component composition.
- Implemented Chart.js library to analyze and generate results, using component level state to incorporate interactive and animated graphs for application.
- Employed Webpack to bundle application, modularize code and implement efficient deployment via proxy, decreasing load times and improving UX.
- Designed a PostgresQL relational database to create a scalable data model for myriad of users, facilitating a persistent and scalable data store.
- Used Express library to design lightweight RESTful APIs by creating routes and middleware controllers that served as liaisons to CRUD operations.
-

CS Study | *Flash Card App*

- Implemented React Router to dynamically render card views, declaring static routes and components to minimize HTTP requests and page reloads.
- Created PostgresQL database schema, tables and associations to provide data integrity for complex, relational data model to ensure ACID compliance.
- Leveraged Node.js with Express framework, implementing router and middleware design to modularize routes that rendered pages and handle requests.
- Utilized Bcrypt encryption to secure client login and passwords and reduce the risk of unauthorized access to sensitive data containing user's details.
- Employed Jest and Supertest to ensure API and middleware endpoints behave as expected and minimize the debugging package functionality.

Book Finder| *Book finder App*

- Implemented React.js by creating reusable, composable components to handle complex DOM interactions and updates through component-level state.
- Implemented Mongoose ORM to generate NoSQL schema and created Express server routes to persist nested documents and enable CRUD operations.
- Configured Webpack and Babel to transpile ES6 to ES5, bundle assets and increase codebase modularity for maximum cross-browser compatibility.

PUBLIC TALKS

Microservices Architecture|*What Are They?* | *Build with code NYC* *March 2020*

EDUCATION

Finance and Economics University in Ukraine | Master of Management of Organization and Administration *2014*
IT Pass, Inc | Certified Quality Assurance Engineer *2018*

SPOKEN LANGUAGES : *English / Ukrainian / Russian*

INTERESTS

Traveling to foreign countries|Finding the best sushi in NYC|Cooking Ukrainian food |Listening to any music |Dancing|Taking pictures