COLE REDFEARN

(949) 922-1130 • coleredfearn@gmail.com • github.com/coleredfearn • linkedin.com/in/coleredfearn

SKILLS

Strong | JavaScript, React (Hooks, Router, Fiber), Node, Express, Redux, Recoil, React Flow, Context API, Webpack, Relational Databases (PostgresQL), Non-Relational Databases (MongoDB/Mongoose), REST, Agile/Scrum, Bcrypt, Cookies, HTML, CSS, SASS, Axios

Proficient | TypeScript, TDD (Jest, Enzyme, Supertest), AWS, Docker, Kubernetes, OAuth, GatsbyJS, Material-UI, CI/CD, Websockets

EXPERIENCE

Atomos (Open Source) | ***Software Engineer*** **2021 - Present**

- Created an open-source Chrome developer tool for React and Recoil to accurately track atom, selector, and component relationships and facilitate the debugging process of React applications built with the Recoil state management library
- Employed React on the front end to ensure a modular and reusable component architecture, consistency within the application codebase, adherence to DRY principles, and reduction of load times for the application's dynamic user interface
- Leveraged Chrome browser API and React DevTools to access the React Devtools Global Hook to find all relevant data regarding components, atoms, and selectors from the current tab's React application, streamlining the setup process for users
- Developed algorithms to recursively parse through the React Fiber tree and populate a new nested object with pertinent data regarding component sibling/child relationships, atom/selector relationships, data flow, and state management info
- Implemented React Flow node-based diagramming library to generate individual node elements and compose those nodes into a dynamically rendered, highly interactive component tree featuring atom/selector relationship visualization
- Utilized Agile methodology to efficiently organize development of the project, coordinate version control across the team, document commits and pull requests, and test new features by merging to staging branch before pushing to main branch
- Customized Webpack configuration to effectively bundle and transpile all application files into uglified, minified code in order to maintain modularity of the codebase and to further optimize the speed and efficiency of the application
- Product developed under tech accelerator OS Labs (opensourcelabs.io)

Operational Technical Services | ***Operations Analyst*** **2020**

- Designed client heat map to systematically track geographic sales data across California and identify macroscopic trends in order to deploy marketing resources efficiently and capitalize on regional markets with high potential value

OPEN SOURCE PROJECTS

LetsGo | ***Software Engineer*** **2021**

- Utilized SQL database with parameterized queries to validate and sanitize user input, ensuring valid and consistently formatted, ACID-compliant data, and protecting sensitive user information from Cross Site Scripting (XSS) and SQL Injection attacks
- Integrated BCrypt to hash user passwords before storing in the database, protecting sensitive user data from dictionary attacks
- Employed cookies to create user sessions, persist the current login credentials, and link database queries to specific users
- Configured Node and Express middleware to process HTTP and API requests in a systematic and modular structure, enable CRUD functionality, provide for future scalability, manage user authentication, and create user sessions and cookies

OS Cards | ***Software Engineer*** **2021**

- Implemented a MongoDB database to save user login credentials and previously created decks in a No-SQL, document-based system, improving the application's performance, scalability, flexibility, and protection from SQL injection attacks
- Conducted unit and integration tests using Jest and Supertest to test HTTP endpoints, accelerate the development cycle, ensure consistency and reliability of code, identify bugs and unexpected edge cases, and facilitate refactoring in the future
- Employed Sass to store variables, utilize nested syntax, increase readability, and provide a much-improved developer experience
- Utilized React Hooks to efficiently update state in specific components and eliminate the need for method bindings, resulting in a more performant and digestible codebase with an emphasis on functional components over class-based components

Movienite | ***Software Engineer*** **2021**

- Utilized Axios to efficiently process asynchronous HTTP requests using compact syntax and native error handling
- Incorporated React Router to build a single-page application (SPA), increasing responsiveness and enhancing user experience
- Harnessed React to modularize components, ensure one-way data flow, simplify state management, and build a maintainable, performant front-end codebase that will serve as a highly extensible framework for future iteration and expanded functionality

EDUCATION

University of Texas at Austin
BBA

TALKS AND ARTICLES

WebAssembly: The Future of High-Performance Web Applications | ***SingleSprout Speaker Series***
Atomos - A New Recoil Visualization Tool Powered by React Flow | ***Medium Article***

INTERESTS

Golf, Basketball, Surfing in Hermosa Beach, Hiking in Malibu and Palos Verdes, Riding my bike, Finding the best fried fish tacos in LA