

PRAISE EMMANUEL

Software Engineer | New York, NY

347-681-2884 | praisepemmanuel@gmail.com | github.com/praisepelumi | linkedin.com/in/praisееmmanuel/

TECHNICAL SKILLS

Languages and Technologies: TypeScript, JavaScript ES6+, React.js, Node.js, Next.js, Redux, Tailwind CSS, HTML5, SCSS, CSS, Express.js, NoSQL (MongoDB), SQL (PostgreSQL), Mongoose, React Testing Library, Jest, Supertest, Webpack, Docker, RESTful APIs, Git, Github.

EXPERIENCE

CS Engineering | *Software Engineer*

2023 - Present

- Utilized React's component-based architecture and Redux to develop and maintain Codesmith's public site and CSX learning platform, drawing over 165,000 annual users.
- Created Jest and React Testing Library unit tests alongside Cypress-end-to-end tests in the CI/CD pipeline, ensuring proper rendering of React/Redux components, intended middleware functionality, and early detection of unwanted side effects.
- Migrated the JavaScript codebase to TypeScript, ensuring type safety and code maintainability through comprehensive type checking and robust code structuring.
- Conducted technical training for junior engineers on topics such as React, Node/Express, TypeScript, Testing, and Ajax, to promote technical growth and foster engineering best practices.
- Performed technical interviews to assess candidates' JavaScript proficiency, problem-solving, technical communication, and debugging skills while evaluating their overall alignment with the company's culture and values.
- Provided mentorship to teams of engineers, coordinating 1:1 code reviews, overseeing the entire Software Development Life Cycle (SDLC) for full-stack open-source applications, facilitating regular stand-up meetings and sprints within an agile/scrum framework to ensure successful product launches and optimize workflow efficiency.

Kubernét (Open Source Labs) | *Software Engineer*

2023

- Collaborated with an agile team to develop an open-source product aimed at simplifying the visualization of Kubernetes cluster metrics such as CPU and memory utilization.
- Employed React for reusable front-end components, enforcing one-way data flow in line with Flux architecture, and efficiently managed state and asynchronous API calls using React Hooks.
- Built the backend of the application using Express for its ability to integrate npm packages, leverage routing, and set up middleware chains to perform different tasks such as user authentication and database queries.
- Utilized React Testing Library, Jest, and Supertest for comprehensive testing, verifying UI rendering, user interactions, backend functionality, and code scalability, ensuring codebase maintainability.
- Leveraged MongoDB's flexible and scalable data model for efficient storage and management of data, ensuring seamless retrieval and storage with horizontal scalability and high-performance capabilities.
- Utilized React Router to establish routes in the application, enabling smooth navigation across pages without requiring page refreshes, while optimizing rendering performance through minimal network requests.
- Used Webpack to efficiently bundle JavaScript and static assets, optimize performance, enable hot module replacement, transpile ES6+, and manage complex dependencies for seamless deployment of web applications.
- Integrated SCSS to modularize application code, eliminate the need for unique class names, streamline styling, and ensure consistent design patterns across pages and components.

OPEN SOURCE

[Code Snippets](#) | *Code Storage Tool*

2023

- Performed comprehensive unit tests utilizing Jest and React Testing Library to verify the correct rendering of components.
- Implemented server-side testing using Supertest to validate API endpoints and ensure robust functionality and reliability.
- Revamped the database architecture with MongoDB, optimizing horizontal scaling capabilities and enhancing overall efficiency.

[Mondinero](#) | *Expense and Budget Management Tool*

2023

- Built with React and Redux Toolkit to develop reusable front-end components and establish a centralized store for streamlined maintenance, enhanced modularity, and efficient state management.
- Retrieved user expenses and transactions utilizing the Plaid API.

[Portfolio Website](#)

2021 - Present

- Built my personal portfolio website utilizing React to create modular, reusable front-end components.
- Continuously updated and improved the website to stay current with the latest web technologies.

EDUCATION

Binghamton University

Bachelor's Degree

CompTIA Security+

ACTIVITIES AND INTERESTS

- I enjoy playing the guitar and piano, listening to live music, interior design, and cooking.