



ADRIAN RIVALDY

FRONTEND DEVELOPER

PROFILE

I have been working as a Front-end Developer for +5 years, specializing in Javascript ES6. My scope of expertise revolves around Stack technology such as React, Redux, and Next.js, which have been augmenting my proficiency in developing dynamic, scalable, and user-friendly web applications using modern JavaScript frameworks and libraries. I am also passionate about staying up to date with the latest trends and advancements in technology.

CONTACT

 adrianrivaldy@gmail.com

 0838-1142-5064

 Adrian Rivaldy

 github.com/adrianrivald

EDUCATION

- Universitas Pakuan - Bachelor of Computer Science
- SMK Negeri 3 Bogor - Computer and Network Engineering

SKILLS

- React js
- Vite js
- Next js
- Typescript
- Javascript
- HTML5
- CSS3
- ES6
- RESTAPI

WORK EXPERIENCES

GoodDreamer (October 2022 - Present)

- Utilized Next.js for server-side rendering (SSR) and static site generation (SSG), improving SEO and performance by delivering pre-rendered pages to users.
- Conducted unit and integration tests with Jest, React Testing Library, and Cypress to ensure code quality and prevent regressions, maintaining high test coverage.
- Implemented multi language website using i18n with Next.js
- Collaborated with backend in order to fetch data, handling API responses and errors efficiently
- Handled 4 sub-projects of the product using Vite.js, Next.js, and React.js
- Developed and maintained features of Content Management System using NX Dev with React.js
- Created efficient workflows by incorporating ES6 features into React.js Framework, writing high-quality code in a modular and maintainable way.
- Improved SEO needs such as open-graph, schema, meta, sitemap to make the best result on the Search Engine
- Used React Query's useQuery and useMutation hooks for data fetching and mutation, simplifying the management of async logic in React components while ensuring proper state transitions for success, loading, and error states.
- Utilized RTK Query's createApi to define endpoints for interacting with RESTful APIs, streamlining API calls and managing data fetching, caching, and error handling in a centralized manner.

DataOn (July 2021 - September 2022)

- Developed micro-services architecture using React, enabling independent deployment and scaling of specific features across large-scale applications, resulting in more modular and maintainable codebases.
- Developed ERP software features and pages using React with its Redux, Context API, and Redux Toolkit, handling complex data flows and providing a predictable state across large-scale applications.
- Worked with PM to keep on track with the project plan, timeline, and billing.
- Mentored 6 new internship front-end developers on team in expanding React micro-services web application.
- Integrated RESTful and GraphQL APIs to fetch data, handling API responses and errors efficiently, and ensuring smooth data flow between frontend and backend.

Minimi (July 2020 - June 2021)

- Built and maintained server-side rendering (SSR) and static site generation (SSG) applications with Next.js, enhancing SEO, performance, and user experience.
- Developed and optimized user interfaces for high-traffic applications using React, implementing component-based architectures for reusable, scalable codebases.
- Maintained mobile view interface using CSS Framework called Bulma
- Integrated RESTful APIs endpoints for dynamic data fetching, leveraging React's useEffect and Redux for effective state management and data flow.
- Worked closely with UI/UX designers to translate Figma and Sketch designs into fully functional, interactive, and accessible user interfaces, providing feedback to improve design feasibility and optimize user experience.

Treya (December 2019 - July 2020)

- Slicing layout of the pages and components using HTML, SASS with pixel-perfect designs ensuring seamless experience across various device and screen sizes
- Integrated RESTful APIs endpoints for dynamic data fetching using Axios
- Collaborated with UI/UX design by using Zeplin to ensure the styling code match with the mock-up design
- State management using useEffect and useContext hooks for effective state management and data flow.
- Collaborated with backend developers to define API requirements and implement end-to-end features, following Agile methodologies and frequent code reviews to ensure high-quality releases.