Prakash Paudel – Front-End Developer

React.js Developer — Performance-Driven UI Engineer — JavaScript Enthusiast

Portfolio: prakash-paudel-portfolio.netlify.app
GitHub: github.com/prakash22092002
LinkedIn: linkedin.com/in/prakash-paudel-226501251
Email: p.prakashpaudel35@gmail.com
Mobile: +91 63838 79770
Location: Chennai, India

Education

- **Masai** — India
  Full Stack Web Development — August 2020 - December 2024
  - **Courses:** Python Programming, Data Structures and Algorithms, Databases, React.js
- **Panimalar Engineering College** — India
  Bachelor of Technology in Information Technology; CGPA - 8.22 — July 2020 - June 2024
  - **Courses:** Operating Systems, Data Structures, Networking, Databases, Web Development Fundamentals

Skills Summary

- **Front-End**: HTML, CSS, JavaScript, Python, React.js
- **Libraries**: Zustand, React Query, Tailwind CSS
- **Programming Language**: JavaScript, Python
- **Tools & Technologies**: GitHub Actions, REST APIs, Figma, SEO, Debugging
- **Core CS Concepts**: Data Structures and Algorithms, Linux, Computer Networks

Experience

- **Pazago** — Mumbai
  Front-End Engineer (Full-Time) — December 2024 - July 2025
  - **Dashboard and Portal Development**: Engineered and maintained scalable front-end dashboards using React.js, ensuring pixel-perfect UI based on Figma designs. Delivered seamless UX and achieved 95% client satisfaction through iterative feedback and testing.
  - **Performance Optimization**: Optimized data-intensive dashboards by implementing lazy loading, code splitting, and virtualization. Achieved a 45% reduction in load time and enhanced interactivity for large datasets. Improved Core Web Vitals with CLS consistently below 0.1.
  - **API Optimization with React Query**: Utilized React Query's `useQuery` hook to cache and manage frequently accessed API data, eliminating unnecessary API calls. Reduced network requests by over 50%, resulting in a smoother user experience and a 30% improvement in perceived performance.
  - **Infrastructure Enhancements**: Reduced JavaScript bundle size by 25% through effective implementation of tree shaking and code splitting, resulting in faster initial load and better scalability.
  - **Technical SEO Optimization**: Collaborated on implementing SEO best practices to improve site visibility. Added meta tags, optimized title structures, and ensured semantic HTML. Implemented lazy loading for images and minimized CLS to enhance Core Web Vitals, contributing to a 20% improvement in organic page visits.
  - **Dynamic Component Architecture**: Developed modular and reusable React components to enforce the DRY principle. Reduced code duplication by 40% and accelerated feature delivery by 35%, improving maintainability and team productivity.

Projects

- **UI-Library**
  This project is a UI library website that provides a collection of reusable components to help developers build web pages faster. It is built using HTML, CSS, JavaScript, and Tailwind CSS, offering a variety of components with easy customization options.
  - **Deployed Link**: https://pixel-ui1.netlify.app/
  - **GitHub Repo**: https://github.com/prakash22092002/UI-Library-Website-
  - **Overview**: Developed a scalable UI component library using HTML, CSS, and Tailwind CSS, designed to accelerate web development with reusable and customizable components.
  - **Team Collaboration**: Worked closely with a cross-functional team of 4, ensuring seamless collaboration, task division, and timely delivery.
  - **Code Quality**: Increased code readability by 40% by implementing a modular component-based architecture and consistent naming conventions.
  - **Scalability**: Enhanced scalability by 60% through abstraction of design tokens, enabling easy theme and layout adjustments across multiple pages.
  - **UI & UX**: Led the creation of an engaging landing page, which improved initial user engagement by 25%, and ensured 100% responsiveness across all screen sizes with support for both light and dark modes.
  - **Documentation**: Authored clear, structured documentation and best-practice guides, reducing onboarding time for new developers by 50%.