N P SOORAJ

Portfolio|soorajnp.nelamangala@gmail.com| +91-8088234144 | github.com/Sooraj2003

Education

SJB Institute Of Technology,India **2022- 2025**
- Branch : **CSE** | CGPA: **8.68**

Narayana PU College,India **2020 - 2021**
- *PU (Class XII), Aggregate:**96.5***

St Mary's High School,India **2018 - 2019**
- *SSLC(Class X), Aggregate:**90.88***

Skills

| Java | JavaScript | React | TailwindCSS | Redux | Jest | Git | Docker | Jenkins | AWS EC2 |

Projects

Food Delivery App (using Swiggy APIs) foodieeapp

I developed a dynamic food delivery application using live Swiggy APIs, leveraging Redux for efficient state management. To enhance performance, I implemented lazy loading and used Parcel as a bundler for minification and optimized loading. I also ensured the reliability of the codebase by performing unit and integration testing with Jest.

- Utilized **Redux** for managing global state efficiently, ensuring smooth user interactions with real-time data from **Swiggy APIs**.
- Improved application performance with **lazy loading**, **code minification** via **Parcel**, and conducted **unit** and **integration testing** using **Jest**.
- Used `useState` and `useEffect` hooks to fetch data from the APIs, enabling efficient and reactive updates based on user actions.
- Implemented **React Router DOM** for seamless client-side routing, improving navigation between different sections of the app.

Netflix Clone netflix_clone

Build a Netflix clone by creating a React app with Firebase authentication, fetching movie data and search functionality from TMDB API, managing state with Redux, styling with Tailwind CSS, and deploying on Netlify with multilingual support for the search page.

- Build a Netflix clone using **Firebase** for **sign in**, **sign up**, and **sign out**, fetch movie data from TMDB, manage state with Redux, style with Tailwind CSS, and deploy on Netlify
- Build a Netflix clone with **multilingual support** in the search functionality .

Automated the deployment LinkToProjectRepo

*Implemented **CI/CD** automation using **Docker**, **Jenkins**, and **AWS**, streamlining the deployment process for faster, more reliable releases. Optimized build pipelines and infrastructure management, ensuring efficient and scalable application delivery.*

- Created and managed Docker containers for consistent development and deployment environments, using **Dockerfiles** to define and automate the **containerization** of applications.
- Automated CI/CD pipelines with Jenkins for efficient build, test, and deployment processes, and deployed scalable applications on **AWS EC2** for high performance and reliability.