**Tanguturu Venkata Sivanagi Reddy**

**Mobile:** +91 6303847478 ▪ **Mail ID:** [sivanagireddyrockstar@gmail.com](mailto:sivanagireddyrockstar@gmail.com)

LinkedIn Profile: [www.linkedin.com/in/sivanagireddytangutturu](http://www.linkedin.com/in/sivanagireddytangutturu)

GitHub Profile: [Sivanagi Reddy(github.com)](https://github.com/unvrknw881)

▪ React JS Developer

▪ Spring boot RESTful webservices ▪ API Development ▪ Troubleshooting

Communicatoin skills ▪ Collaborative and team worker

**Technical Expertise**

* **React JS**: React hooks, redux toolkit, API integration in react JS using useSWR.
* **React testing library**: Tested components to ensure 100% code coverage.
* **Spring Boot REST**: Performed CRUD operations.
* **Spring Boot**: Worked on Java projects.
* **Git/GitHub:** Managed code repositories.

**Skills**

* **Frontend Technologies:** React JS, Redux toolkit, React Hooks, Figma, Typescript
* **Programming Languages:** Java, JS
* **Frameworks:** Spring Boot
* **Web Services:** RESTful web services
* **Testing:** Jest, React Testing Library
* **Version Control:** Git, GitHub

**Work Experience**

* **Full Stack Java Development Internship** - Skill Safari Sep 2021 to Nov 2021
* **Full Stack Web Development Internship** - Cognizant Technology Solutions Jan 2022 to June 2022
* **React JS developer -** Cognizant Technology SolutionsAug 2022 to ---

**Project Experience:**

**America-GCS TestBed Fixed Capa:** This is an American express project which will display the card member details, In this project there were multiple screens that we had to develop, product manager and project manager will add the screens in figma, from figma we can check the UI/UX, element’s exact measurements like line spacing, font size, font style, icons andworked on react JS based one-app framework for the frontend, used react hooks, API integration using useSWR, created multiple components using one-app, written testcases using react testing library to make sure 100% code coverage, state management using context API, routing, I have used parrot middleware for checking all the scenarios from the UI perspective basically this is used for mocking the APIs, we can create scenarios like 404 error, bad request error, with this type of error scenarios we can create UIs without depending on the backend APIs. Deployed code from GitHub repositories to Jenkins to make sure code is working on different environments.

**TransMe(Transform Messages):** TransMe is message transformation application for converting MT type of messages into MX type, in this project I have worked as an front end developer, using react JS I have created search functionality using filter method and managing state using react hooks like useState(), useRef() hooks, and worked on creating dropdowns, modal, scroll bar. Integrating backend APIs with frontend using useSWR() hook and to avoid prop drilling I have implemented Context API, for complex state management in the application I have used redux tool kit.

**America-eApply Nextn-Gen Transf:** eApplyis part of American express credit card application, it is used to apply for credit cards through an online form, where all the details of the customers will be procured through this form, in this project I have contributed on Automation and manual testing, using selenium driver, cucumber JS, deployed the code to the Jenkins to check whether testcases are working in e2 environment.

Raised PRs and resolved merge conflicts.

**Key Deliverables:**

* State management using Redux toolkit, Context API.
* API integration using react hooks.
* Component testing using Jest and React Testing Library.
* React routing.
* Type safety using typescript.
* Created APIs using RESTful web services and implemented CRUD operations.
* Used Git and GitHub for code management, including creating branches, raising pull requests, resolving PR comments, and merging conflicts.
* **ReactJS Deliverables**
* **User Management System:** Developed a user management system with React JS as frontend and Spring Boot as backend. Implemented CRUD operations and integrated with a MySQL database.
* **Theme based on user preference(using local storage):** Background theme value is stored in local storage, so that, when the user revisits the website his preferred theme will be populated.
* **API integration:** Integrating backend APIs with frontend using useSWR react hook.
* **TypeScript:** Including typescript in the components for type safety.
* **Grid layout:** Created UI screens using grid layout.
* **Component Testing:** Utilized Jest and React Testing Library to ensure 100% code coverage for React components.
* **Spring boot Java Deliverables**
* Created REST APIs with CRUD operations and tested them using Postman.
* Secured APIs with JWT.
* Using microservices I have developed multiple services and made them communicate with each other using REST template & Open feign(feign client) and used Eureka server & client, created API gateways.