R Sowjanya
Phone: (314) 669-6284
Email: rangannagarisowjanya98@gmail.com
LinkedIn Profile

Summary

Frontend Developer skilled in building dynamic, scalable web applications using React.js and experienced with Node.js. Proficient in developing user interfaces, implementing state management with Redux, and integrating RESTful APIs for seamless frontend-backend communication. Knowledgeable in AWS services, including EC2, S3, and CloudFront, for deploying scalable applications and ensuring high availability. Adept at collaborating with cross-functional teams in Agile environments to deliver maintainable, responsive applications. Focused on applying best practices in performance optimization and security.

Technical Skills

Frontend Frameworks	React.js, Node.js
Programming Languages	JavaScript (ES6+), HTML5, CSS3
State Management	Redux
UI Libraries/Frameworks	Material-UI
API Integration	Axios, RESTful APIs
Version Control	Git
CI/CD Tools	Jenkins
Cloud Services	AWS EC2, AWS S3, AWS CloudFront
Testing Frameworks	Jest, Enzyme
Development Tools	Visual Studio Code, Webpack, Chrome Developer Tools
Security & Compliance	HIPAA, Role-Based Access Control (RBAC), TLS/SSL Encryption
Agile Practices	Scrum, Sprint Planning, Daily Stand-ups

Professional Experience

Senior React Developer **Jan 2024 – Present**
Johnson & Johnson

Project: Patient Support Program Portal (PSPP)

Project Description:

Built a secure, scalable healthcare portal focused on patient support, offering personalized treatment plans, medication reminders, and educational resources. The application follows a modular architecture integrating microfrontends with Spring Boot microservices, delivering responsive and accessible interfaces for patients, care coordinators, and healthcare providers.

Responsibilities:

Frontend

- Developed dynamic, responsive UI components using React 18, TypeScript, and Bootstrap, delivering dashboards, forms, and patient tracking modules.
- Used Redux Toolkit and Context API to manage application-wide state, supporting real-time updates and seamless navigation across modules.
- Implemented Axios for asynchronous REST API calls, enabling consistent data flow between frontend and Spring Boot microservices.
- Utilized React Router for client-side navigation, enabling user-specific views for patients, doctors, and admins.
- Employed Formik and Yup for form state management and validation, ensuring secure and accurate data capture for treatment inputs and monitoring.
- Structured UI using a component-based architecture, ensuring reusability and maintainability of form and dashboard modules.
- Ensured code quality and test coverage through React Testing Library, Jest, and Mock Service Worker (MSW) for API simulation.

Backend

- Designed and consumed RESTful APIs built on Spring Boot to support features like user onboarding, treatment scheduling, and health data integration.
- Utilized Java 11 features such as streams and Optionals to enhance backend code readability and efficiency.
- Integrated Hibernate ORM with Spring Data JPA for mapping entities to relational schemas in Oracle and PostgreSQL databases.
- Used Spring MVC for request handling, applying layered architecture principles to separate concerns across controllers, services, and repositories.
- Implemented robust exception handling, input validation, and secure logging for all backend services.
- Integrated Spring Security with OAuth2 and JWT for secure authentication and role-based access control.

Cloud

- Deployed frontend and backend components on AWS using EC2, S3, and PostgreSQL RDS for scalable infrastructure and persistent data storage.
- Implemented AWS Lambda for background processes like email reminders and batch updates, and used API Gateway for secure endpoint management.
- Containerized services using Docker and orchestrated deployments using Kubernetes to streamline environment consistency across dev and prod.
- Configured CI/CD pipelines with Jenkins and GitLab CI to automate build, test, and deployment workflows.
- Monitored system health using CloudWatch, setting up alarms and logs for proactive detection of issues.
- Managed access policies and service roles using AWS IAM to ensure HIPAA-compliant security and data protection.

Environment: React 18, TypeScript, Redux Toolkit, Axios, React Router, Formik, Yup, React Testing Library, Jest, MSW, HTML5, CSS3, Bootstrap, Java 11, Spring Boot, Spring MVC, Spring Data JPA, Hibernate, REST APIs, Spring Security, OAuth2, JWT, Oracle, PostgreSQL, AWS Lambda, S3, EC2, API Gateway, IAM, CloudWatch, Docker, Kubernetes, GitLab CI, Jenkins, Maven

Software Developer **Jan 2022 – Aug 2023**

JPMorgan Chase

Project: Cloud-Native Client Onboarding and Portfolio Management System

Project Description:
worked on building an enterprise-grade, cloud-native client onboarding and portfolio management platform tailored for internal advisors and client relationship teams. The application was designed as a microfrontend system integrated with microservices on the backend. It featured real-time data sync, responsive design, and seamless user interaction powered by modern JavaScript frameworks and AWS-native infrastructure.

Responsibilities:

Frontend

- Developed React 18-based UI components to create dynamic dashboards and onboarding workflows, ensuring a responsive and interactive experience for client relationship teams.
- Utilized Axios to consume RESTful and GraphQL APIs, facilitating secure data exchange with backend microservices for client portfolios, transactions, and KYC information.
- Implemented Redux Toolkit and Context API to manage application-wide state for modules like account summary, investment history, and real-time alerts, ensuring data consistency.
- Leveraged React Router for nested routing and protected views, enabling smooth navigation based on user roles and access levels.
- Used Formik with Yup to build complex, multi-step forms with custom validation logic for onboarding, risk profiling, and client preferences.
- Performed unit and integration testing using React Testing Library, Jest, and Mock Service Worker (MSW) to validate UI behaviors, simulate API responses, and maintain code reliability.
- Collaborated with UX and accessibility teams to ensure all interfaces met WCAG 2.1 compliance, delivering inclusive user experiences across browsers and devices.

Backend

- Collaborated with backend teams to integrate the frontend with RESTful services developed in Spring Boot, ensuring consistent API contracts for client, account, and transaction modules.
- Used Java 11 features such as functional programming and Stream APIs to simplify and enhance data processing within backend logic.
- Interacted with relational and NoSQL databases through Spring Data JPA and Hibernate, handling operations like client creation, profile updates, and data retrieval.
- Applied Spring Boot's annotation-driven configuration to structure services, repositories, and controller layers, enabling modular and maintainable codebases.
- Implemented centralized exception handling, input validation, and response wrapping to maintain a clean and consistent API interface for all consumers.

Cloud

- Containerized frontend applications using Docker and deployed them to Kubernetes clusters using Helm charts for versioned, scalable deployments on AWS.
- Hosted React application assets on Amazon S3 and configured CloudFront as the CDN to enable fast, secure delivery across geographies.
- Utilized AWS Lambda functions and API Gateway to expose serverless endpoints for low-latency event-driven services supporting the onboarding workflows.
- Implemented secure data storage using Amazon DynamoDB for storing user preferences and audit logs, and Amazon RDS for transactional data persistence.
- Managed CI/CD pipelines through GitLab CI and Jenkins, automating code linting, builds, tests, container image publishing, and deployments to development and staging environments.
- Enforced identity and access management using AWS IAM and integrated Cognito for federated authentication via enterprise identity providers, maintaining secure access controls.
- Monitored and logged cloud service performance using Amazon CloudWatch, setting up dashboards, log streams, and alerts for proactive issue resolution and uptime assurance.

Environment: React 18, TypeScript, TailwindCSS, Redux Toolkit, Context API, React Router, Axios, REST APIs, GraphQL, Formik, Yup, Docker, Kubernetes, AWS (Lambda, S3, DynamoDB), GitLab CI, Jenkins, React Testing Library, Jest, MSW, Spring Boot

Full Stack Developer **Oct 2019 – Dec 2021**
Regions Financial Corporation

Project: Financial Transaction Processing and Risk Management System

Project Description:
Led the development of a high-volume Financial Transaction Processing and Risk Management System at Regions Financial Corporation. This platform was essential for handling and processing large volumes of financial transactions while ensuring compliance with regulatory requirements and mitigating potential risks associated with banking operations.

Responsibilities:

Frontend
- Spearheaded the development of interactive user interfaces using React.js, enabling real-time transaction approvals and monitoring for internal financial teams to streamline high-volume workflows.
- Created modular, reusable React components tailored to transaction lifecycle needs, focusing on real-time data visualization and automated risk assessment features.
- Implemented Redux for state management, maintaining consistent application state across modules involved in transaction processing, compliance checks, and approval workflows.
- Integrated Axios for secure and efficient data exchange with backend systems, enabling seamless access to transaction records and risk indicators from DB2 and MongoDB.
- Developed a robust testing strategy using Jest and Enzyme to ensure frontend reliability and functional coverage of all transaction processing interfaces.

Backend
- Developed RESTful APIs using Spring Boot and Java 11 to handle transaction processing, approvals, and compliance validation workflows.
- Integrated Spring Data JPA with DB2 and MongoDB to support complex data operations and ensure consistency across high-volume financial records.
- Implemented service-layer logic to perform real-time risk evaluation and validation of financial transactions against regulatory rules.
- Secured backend endpoints using OAuth2 and implemented request validation to maintain system integrity and compliance.
- Created unit and integration tests using JUnit and Mockito, ensuring backend services met quality and performance benchmarks.

Cloud
- Deployed the transaction processing platform on AWS EC2 instances to ensure scalability, resilience, and continuous availability during periods of peak transaction loads.
- Used Amazon S3 for secure, compliant storage of transaction data and audit logs, enabling traceability and streamlined access during regulatory reviews.
- Integrated AWS Lambda to trigger real-time risk assessment functions, enhancing the platform's responsiveness and scalability for on-demand analysis.
- Implemented and maintained AWS API Gateway endpoints for secure routing of client-server communication.
- Worked with security engineers to implement TLS/SSL encryption and Two-Factor Authentication (2FA) to meet Basel III and Dodd-Frank compliance standards.
- Monitored cloud infrastructure using AWS CloudWatch for proactive performance tuning and system alerting.

Environment: React.js, Redux, JavaScript (ES6+), HTML5, CSS3, AWS EC2, AWS S3, AWS Lambda, Axios, RESTful APIs, Jest, Enzyme, Webpack, Git, Jenkins, DB2, MongoDB, TLS/SSL, 2FA, Basel III, Dodd-Frank.

Junior React Developer **Dec 2016 – Sep 2019**
Bigbasket

Project: Cloud-Based E-commerce Inventory and Order Management System

Project Description:
Contributed to the development of a cloud-based Inventory and Order Management System designed to streamline Big Basket's product catalog, order processing, and inventory tracking across multiple warehouses. The system was vital for ensuring real-time stock updates and smooth order fulfillment while supporting scalability during high demand periods.

Responsibilities:

Frontend
- Contributed to UI development using React.js, focusing on implementing layout updates and small components such as product cards, quantity selectors, and cart summary widgets based on senior developer designs.
- Performed maintenance on existing React components, resolving UI bugs and improving responsiveness for product listing and checkout pages.
- Assisted with integrating Axios to support basic API calls for fetching inventory and order data, ensuring proper data rendering across components.
- Helped apply responsive layout techniques using CSS3, Flexbox, and Grid, ensuring mobile-friendly views for shopping workflows.
- Collaborated with the frontend team to troubleshoot rendering issues and align UI behavior with business logic requirements.

Backend
- Assisted in developing RESTful API endpoints using Java and Spring Boot for handling product catalog and order data exchange between frontend and backend systems.
- Worked with Spring Data JPA to perform CRUD operations on backend databases, supporting functionality like product detail retrieval and cart updates.
- Helped implement basic validation and error handling for backend services to ensure proper data integrity and consistent response formatting.
- Supported integration with SQL-based data stores by writing simple queries and assisting in mapping backend entities to frontend models.

Cloud
- Assisted in deploying frontend builds to Amazon EC2 by monitoring deployment logs and supporting environment readiness checks.
- Managed basic uploads to Amazon S3 for storing product images and static assets, ensuring proper access control settings for public-facing content.
- Observed and supported serverless backend integrations using AWS Lambda for event-driven product updates.
- Participated in UI testing using Jest, contributing to test case creation and bug validation for shopping flows and data display scenarios.

Environment: React.js, JavaScript (ES6+), HTML5, CSS3, AWS EC2, AWS S3, Axios, Jest, Git.