Sai Suraj Nuthanakalva
Full Stack Java Developer
Email: nuthanakalvasaisuraj@gmail.com
Phone No: (669)999-7215
LinkedIn | Medium | Portfolio

Professional Summary:

- **AWS Certified Solution Architect** Full Stack Java Developer with over **8 years** of extensive experience in designing and building highly scalable, robust, and efficient web applications. Proven expertise in delivering full lifecycle solutions, from concept to deployment, across diverse industries to create dynamic, responsive web applications that provide seamless user experiences across devices.
- Developed reusable and modular **React components** with **React Hooks** and **Redux** for efficient state management and handling complex application states.
- Created dynamic user interfaces using **Angular**, **TypeScript**, and **RxJS**, implementing **Reactive Forms** and handling complex input validation to ensure data integrity.
- Integrated third-party services like **Microsoft Bot Framework** for chatbot functionality and **Chart.js** for data visualization, enabling real-time insights and improved decision-making.
- Used **CSS3**, **SASS**, and **Tailwind CSS** to ensure mobile-first designs, optimizing the user interface for accessibility and responsiveness across devices.
- Designed and built scalable backend services using **Java**, **Spring Boot**, and **Spring MVC**, enabling efficient processing of real-time data and ensuring the backend services remain performant under high load.
- Developed **RESTful APIs** using **Spring Boot** and **Spring WebFlux** for asynchronous processing, handling high-frequency requests from front-end systems with minimal latency.
- Utilized **Spring Security** and **OAuth2** to secure API endpoints and implement **JWT-based** authentication, ensuring role-based access control for various user roles like managers and analysts.
- Integrated with **ServiceNow** API to automate support workflows, and built microservices to handle business logic for various banking and telecom systems using **Java SE/EE** and **Spring Boot**.
- Leveraged **Java's CompletableFuture** and **Multithreading** to handle concurrent requests and real-time updates, reducing response times and improving system performance for large-scale distributed applications.
- Used **RabbitMQ** for managing asynchronous messaging between services, ensuring reliable communication and real-time data processing across microservices architecture.
- Leveraged **Apache Kafka** for high-throughput event streaming, enabling real-time transaction updates and customer behavior insights in financial applications.
- Worked extensively with **MongoDB** and **Cassandra** for scalable data storage and retrieval, optimizing queries and performance for large datasets in financial and sustainability tracking systems.
- Designed and optimized **CQL** and **PL/pgSQL** queries for fast data retrieval and efficient report generation in banking and retail environments.
- Deployed microservices and backend systems on **AWS** using **EC2**, **S3**, and **Lambda** for serverless task execution, ensuring scalability and fault tolerance in production environments.
- Managed containerized applications using **Docker** and **Kubernetes** on **AWS** and **Azure Kubernetes Service (AKS)**, automating deployment, scaling, and management of services.
- Utilized **Azure Blob Storage** for efficient data storage and retrieval of large reports, enabling easy access for audits and compliance purposes.
- Configured **Azure API Management** to handle routing and load balancing for financial services, ensuring high availability and resilience of microservices.
- Built and managed **CI/CD pipelines** using **Jenkins**, automating the build, test, and deployment workflows for both frontend and backend services, significantly reducing manual intervention and improving release cycles.
- Implemented **Docker** for containerization and **Kubernetes** for orchestration, ensuring high availability and reliability of services in distributed environments.
- Managed version control with **GitHub** and **GitLab**, collaborating through pull requests, conducting peer code reviews, and ensuring conflict-free integration of new features and bug fixes.
- Utilized **GitFlow** to maintain consistent branching strategies and ensure efficient collaboration across global teams.

- Developed comprehensive unit tests for both frontend and backend using **Jest**, **JUnit**, and **Mockito**, ensuring code quality and high-test coverage across all components.
- Implemented **integration tests** using **TestContainers** and **Selenium** to validate the seamless communication between services and ensure the stability of the system in production environments.

Technical Skills:

Category	Skills
Languages	Java, JavaScript, TypeScript, SQL, PL/SQL, CQL
Frontend	React, Angular, HTML5, CSS3, SASS, Bootstrap, Tailwind CSS
Backend	Spring Boot, Spring MVC, Node.js, Express.js, RESTful APIs, Microservices
Frameworks	Spring, Hibernate, Apache Kafka, RabbitMQ, OAuth2, JWT
Databases	MongoDB, Cassandra, MySQL, Oracle
Cloud	AWS, Azure, Google Cloud Platform (GCP), Docker, Kubernetes, Azure Kubernetes Service (AKS)
Tools	Jenkins, Git, GitHub, GitLab, Maven, Terraform, Jenkins CI/CD
Version Control	Git, GitHub, GitLab
Testing	JUnit, Mockito, Selenium, Jest, Jasmine, Karma, Test containers

Professional Experience:

Client: IBM **Feb 2023 – Till Date**
Location: Austin
Role: Sr. Full-stack Java Developer

Description: Developed an AI-powered employee self-service portal integrated with ServiceNow to streamline IT asset management and intelligent ticket routing. The portal used machine learning to predict and resolve routine IT issues through a chatbot, while employees could track and manage company-issued devices, request upgrades, and monitor the asset lifecycle. AI-driven ticket categorization and routing ensured faster resolution by assigning tickets to the appropriate teams, reducing manual intervention and improving efficiency. This solution enhanced IBM's IT service delivery by leveraging automation, real-time insights, and predictive maintenance capabilities.

Responsibilities:

- Developed a highly interactive portal interface using **React** and **Microsoft Bot Framework** to handle employee IT queries and automate routine technical issue resolution.
- Optimized the UI for responsiveness and accessibility using **React Bootstrap** and **CSS3 media queries**, ensuring seamless operation across mobile, tablet, and desktop devices.
- Utilized **React Hooks** and **Redux** for efficient state management, handling complex application states, API responses, and user interactions.
- Implemented **dynamic form validation** and **error handling** using React, improving user experience and reducing submission errors for IT service and hardware requests.
- Integrated **real-time chat functionality** using **Microsoft Bot Framework**, enabling employees to receive immediate support and status updates on IT and hardware requests.
- Designed and developed **RESTful APIs** using **Java** and **Spring Boot**, enabling real-time communication with ServiceNow for IT ticketing, hardware requests, and asset management.

- Implemented **asynchronous processing** using **Spring WebFlux**, reducing response time for high-frequency requests and ensuring minimal latency in IT support workflows.
- Enhanced backend services using **Java Multithreading** and **CompletableFuture** to handle concurrent ticket submissions, hardware provisioning, and asset tracking requests efficiently.
- Developed the backend logic with **Java SE/EE**, ensuring the smooth processing of employee queries, task generation, and interaction with external systems like ServiceNow and inventory management.
- Developed custom utility classes using **Java SE 8 features** such as **NIO (Non-blocking I/O)** for file operations, **ScheduledExecutorService** for task scheduling, and **Stream API** for efficient data parsing, which improved backend processing efficiency and simplified complex workflows.
- Leveraged **Java Collections** and **Streams API** for efficient data manipulation and transformation, contributing to the overall performance enhancement of the portal's backend.
- Used **Java Concurrency API** for handling multiple IT ticket submissions and hardware requests simultaneously, ensuring fast and reliable task execution in a high-traffic environment.
- Integrated **RabbitMQ** for managing asynchronous message queues, ensuring reliable real-time processing of employee IT support requests and backend tasks.
- Utilized **MongoDB** to store employee interactions, task history, and support tickets, optimizing queries for faster retrieval and ensuring high availability of data across distributed microservices.
- Designed database operations using **JPA/Hibernate** for smooth and optimized storage and retrieval of task history, hardware asset details, and ticket logs.
- Deployed the entire application on **AWS**, utilizing **Lambda** for serverless execution of specific tasks like ticket categorization and hardware provisioning, ensuring cost-effective cloud resource usage.
- Managed storage and retrieval of audit data and logs using **AWS S3**, ensuring scalability and reliability for compliance and tracking.
- Configured and managed containerized services using **Docker** and orchestrated deployments with **Kubernetes** on **AWS EC2**, ensuring high availability and fault tolerance.
- Set up **CI/CD pipelines** using **Jenkins**, automating the build, test, and deployment workflows for both frontend and backend services, significantly reducing manual intervention and improving release cycles.
- Integrated **Jenkins** with **SonarQube** for continuous code quality checks, ensuring code standards are maintained throughout the development process.
- Managed source code and version control with **GitHub**, collaborating through pull requests, conducting peer code reviews, and ensuring continuous integration with Jenkins for automated builds.
- Utilized **GitFlow** branching strategies to maintain code consistency and facilitate efficient collaboration across global development teams.
- Developed unit tests for the React frontend using **Jest**, ensuring component-level functionality and responsiveness across various devices.
- Wrote backend unit tests using **JUnit** and **Mockito**, adhering to **TDD** principles to ensure high code quality and comprehensive test coverage.
- Implemented **integration tests** to validate seamless interaction between the employee portal, ServiceNow, and the database layers, ensuring system reliability and stability in production environments.

Environment: Java, Spring Boot, Spring WebFlux, Spring MVC, Hibernate, JPA, React, Redux, JavaScript, HTML5, CSS3, Microsoft Bot Framework, MongoDB, PostgreSQL, RabbitMQ, Docker, Kubernetes, AWS (Lambda, EC2, S3), Fast API, Log4j, ServiceNow API, Git, Jenkins, JUnit, Mockito, Jest, SonarQube, REST, Agile, Microservices Architecture.

Client: Lactalis **Mar 2021 – Feb 2023**
Location: Chicago
Role: Full stack Java Developer

Description: Developed a sustainability tracking system for Lactalis to automate the monitoring of carbon emissions and energy consumption across multiple production sites. The solution provided real-time insights for sustainability reporting and compliance, improving operational transparency. This project streamlined data collection from various sources, automated reporting, and significantly enhanced Lactalis's ability to track and manage its environmental impact, supporting the company's long-term sustainability goals and reducing operational inefficiencies.

Responsibilities:

- Developed interactive dashboards using **Angular** and **Highcharts** for real-time visualization of key metrics such as production efficiency and energy consumption, ensuring user-friendly access to operational data.
- Designed responsive layouts using **Tailwind CSS** and **CSS Flexbox**, ensuring seamless accessibility across various devices, including desktops, tablets, and mobile phones.
- Utilized **RxJS** and **NgRx** for state management in Angular, enabling efficient real-time updates and handling of large datasets in the dashboards.
- Created and optimized **RESTful APIs** using **Java** and **Spring Boot**, facilitating real-time communication between the front-end dashboards and backend systems.
- Integrated **Cassandra** as the distributed NoSQL database to manage large-scale data storage for production statistics and operational metrics, ensuring high availability and fault tolerance.
- Used **Spring Data Cassandra** to simplify interaction with the database and optimized CQL queries for better performance in querying production and energy data across distributed nodes.
- Implemented **Java Streams API** and **Multithreading** to process large volumes of incoming data from various production sites, enabling real-time data aggregation for the dashboards.
- Utilized **RabbitMQ** to handle asynchronous data flow between distributed services, ensuring efficient data processing and real-time updates on the dashboard.
- Deployed microservices and backend systems using **Azure Kubernetes Service (AKS)** to manage container orchestration and automated scaling of services.
- Stored large operational reports and logs in **Azure Blob Storage**, providing cost-effective and scalable storage for audit and compliance purposes.
- Automated build, test, and deployment pipelines using **Jenkins**, ensuring continuous integration and delivery for rapid feature deployments and bug fixes.
- Managed source code and version control using **Bitbucket**, enabling collaborative development through pull requests and peer code reviews.
- Developed unit tests for the **Angular** frontend using **Jasmine** and **Karma**, ensuring UI components rendered correctly and handled user interactions properly.
- Wrote backend unit tests using **JUnit** and **Mockito** to validate the reliability of APIs and service layers, ensuring robust data processing workflows.
- Conducted integration tests with **Testcontainers** to verify seamless communication between microservices, Cassandra, and RabbitMQ, ensuring smooth system operations.

Environment: Angular 10, RxJS, NgRx, Highcharts, Tailwind CSS, Java, Spring Boot, Spring Data Cassandra, Cassandra, RabbitMQ, Azure Kubernetes Service (AKS), Azure Blob Storage, Jenkins, Bitbucket, Jasmine, Karma, JUnit, Mockito, Testcontainers, REST APIs, Agile.

Client: Tufts Health **Dec 2019 - Mar 2021**
Location: Watertown, MA
Role: Full stack Java Developer

Description: At Tufts Health, the healthcare dashboard project was developed to provide real-time insights into patient health data and insurance claims, addressing the need for faster and more accurate decision-making. By automating data collection and reporting, the system enabled healthcare providers to quickly access and monitor key metrics, improving efficiency and reducing manual errors. This project played a key role in enhancing patient care and streamlining insurance claim management through real-time data visualization and automation.

Responsibilities:

- Developed dynamic and interactive user interfaces using **React**, ensuring that healthcare providers could easily access real-time patient data and claim status.
- Created reusable React components for handling patient metrics and claim details, improving the maintainability and scalability of the dashboard.

- Implemented **React Hooks** and **Redux** for efficient state management, ensuring that data updates were reflected in real-time across the application.
- Built responsive and mobile-friendly designs using **CSS3** and **Flexbox**, ensuring the application worked seamlessly across different devices and screen sizes.
- Developed backend services using **Java** and **Spring Boot** to process patient health data and insurance claims, providing real-time updates to the dashboard.
- Designed and implemented **RESTful APIs** to ensure smooth communication between the React frontend and backend healthcare systems, handling high-frequency data requests efficiently.
- Utilized **Java Multithreading** to process multiple patient records and claims concurrently, ensuring the backend could handle high volumes of data without bottlenecks.
- Built core backend logic using **Java SE/EE** to manage business rules around insurance claims validation and patient data processing.
- Integrated **RabbitMQ** for managing real-time messaging between healthcare systems, ensuring patient data and claims were processed asynchronously.
- Used **MongoDB** to store patient records and insurance claim data, optimizing database queries to ensure fast data retrieval for real-time updates.
- Implemented **JPA/Hibernate** for smooth interaction with **MongoDB**, ensuring the secure storage and retrieval of sensitive patient information.
- Deployed the application on **AWS**, utilizing **EC2** for compute resources and **Lambda** for serverless execution of specific tasks like patient record updates.
- Managed storage of logs and audit data using **AWS S3**, ensuring scalable and secure cloud storage for healthcare-related information.
- Configured containerized services using **Docker** and orchestrated deployments with **Kubernetes** on **AWS EC2**, ensuring high availability and fault tolerance.
- Set up **CI/CD pipelines** using **Jenkins**, automating the build, test, and deployment workflows for both the React frontend and Spring Boot backend, reducing deployment time.
- Integrated **SonarQube** into the CI/CD pipeline to continuously monitor and ensure code quality throughout the project.
- Managed source control using **GitHub**, working collaboratively with the team through pull requests and peer reviews to ensure a clean, maintainable codebase.
- Utilized **GitFlow** for consistent branching strategies, making it easier to manage multiple features and bug fixes simultaneously.
- Developed unit tests for React components using **Jest**, ensuring that UI functionality was validated across different devices and browsers.
- Wrote backend unit tests using **JUnit** and **Mockito**, ensuring that Java services met performance and security standards for handling patient data.
- Conducted integration tests using **Testcontainers** to verify that the React frontend, backend, and MongoDB databases worked seamlessly together in production environments.

Environment: React, Redux, CSS3, Flexbox, Java, Spring Boot, Spring WebFlux, RabbitMQ, MongoDB, AWS (EC2, Lambda, S3), Docker, Kubernetes, Jenkins, GitHub, SonarQube, Jest, JUnit, Mockito, Testcontainers, REST APIs, Agile.

Client: City National Bank **Mar2018 – Dec 2019**
Location: Corsicana, Texas
Role: Jr Full stack Java Developer

Description: Developed a custom financial dashboard system for City National Bank to provide real-time insights into transaction trends, customer behaviour, and performance metrics. This project enhanced the ability of banking executives and analysts to make informed decisions by automating data collection and analysis. It also improved the speed and accuracy of financial reporting.

Responsibilities:

- Developed dynamic user interfaces for the financial dashboard using **Angular**, creating reusable components for transaction tracking, customer behavior insights, and performance analytics.

- Built responsive and mobile-friendly designs using Tailwind **CSS** and **CSS Flexbox**, ensuring seamless functionality on mobile, tablet, and desktop devices for executives and analysts.
- Implemented **Reactive Forms** in **Angular** to handle complex input validation for financial data entry, ensuring accurate submission of transaction reports and metrics.
- Integrated **Chart.j**s to visualize real-time financial data trends, such as customer transaction volumes and revenue growth, allowing banking staff to monitor performance effectively.
- Developed backend services using **Java and Spring Boot** to process large volumes of transaction data and track financial metrics, providing real-time updates to the dashboard.
- Designed and exposed **RESTful APIs** with **Spring WebFlux**, enabling asynchronous data communication between the Angular front-end and backend financial systems, ensuring efficient processing of high-frequency requests.
- Implemented **Java Multithreading** to handle concurrent transaction processing and data aggregation, ensuring backend services could scale efficiently without performance bottlenecks.
- Utilized **Java Streams API** for efficient data manipulation and processing of financial transactions, reducing complexity in data pipelines and improving backend performance.
- Built core backend logic using **Java SE 8** features like **CompletableFuture** and **ExecutorService** for non-blocking asynchronous operations, which improved the responsiveness of the financial dashboard.
- Integrated **RabbitMQ** for managing real-time messaging queues, ensuring reliable and fast data updates between the financial system and external services.
- Used **Apache Kafka** to manage real-time data streams for customer transactions, ensuring high-throughput and low-latency processing of financial events for instant dashboard updates.
- Leveraged **Cassandra** as the primary database to store large-scale transaction data and performance metrics, ensuring optimized data retrieval for generating reports and visualizing trends.
- Developed and optimized **CQL** queries to efficiently manage transaction data storage and retrieval, improving performance for real-time dashboard updates.
- Deployed microservices for financial data aggregation and report generation using **Azure Kubernetes Service (AKS),** providing scalable and reliable processing of financial metrics across distributed environments.
- Utilized **Azure Blob Storage** to store and retrieve large financial reports and transaction logs, ensuring data persistence and easy access for audit and compliance purposes.
- Configured **Azure API Management** to handle routing and load balancing for the financial data services, ensuring high availability and resilience for real-time data access across microservices.
- Built CI/CD pipelines in **Jenkins** to automate the testing, building, and deployment of **Angular** front-end and **Java Spring Boot microservices**, minimizing manual intervention and improving release cycles.
- Integrated **SonarQube** into **Jenkins** pipelines for continuous code quality checks, ensuring code met industry standards throughout the development lifecycle.
- Managed version control using **GitHub**, collaborating with the development team through pull requests and peer code reviews, ensuring a clean and maintainable codebase.
- Utilized **GitFlow** for consistent branching strategies, enabling efficient collaboration and feature development across the team.
- Developed unit tests for Angular components using **Jasmine and Karma**, ensuring that the UI rendered correctly and handled user inputs across various browsers and devices.
- Wrote backend unit tests using **JUnit and Mockito**, ensuring the Java services met performance and security standards for processing real-time financial transactions.
- Conducted integration tests using **Testcontainers** to verify seamless communication between the Angular front-end, Spring Boot microservices, and Cassandra, ensuring system reliability in production environments.

Environment: Angular, TypeScript, Tailwind CSS, Chart.js, Java, Spring Boot, Spring WebFlux, Java SE 8, RabbitMQ, Apache Kafka, Cassandra, CQL, Azure Kubernetes Service (AKS), Azure Blob Storage, Azure API Management, Jenkins, GitHub, SonarQube, Jasmine, Karma, JUnit, Mockito, Testcontainers, REST APIs, Agile.

Client : Pramati Technologies Pvt ltd **Apr 2016 - Mar 2018**
Location: Hyderabad, India
Role: Jr Full stack Java Developer

Description: Contributed to the development and optimization of enterprise-level applications for Pramati Technologies by working on both frontend and backend solutions. Improved system performance by implementing efficient UI components, integrating REST APIs, and building microservices to streamline business logic processing. Collaborated with cross-functional teams to ensure application stability, scalability, and efficient data handling.

Responsibilities:

- Participated in frontend development using **HTML5**, **CSS3**, and **JavaScript**, ensuring cross-platform compatibility and improving user experience.
- Enhanced interactivity by implementing dynamic UI components using **JavaScript** and **jQuery**, improving engagement across devices.
- Developed **RESTful APIs** for seamless data exchange between Aptiva's internal applications and external services, streamlining workflows.
- Improved backend performance by designing and optimizing **Spring Boot**-based microservices to handle business logic and increase modularity.
- Optimized database interactions using **Hibernate**, implementing efficient data mapping and lifecycle management for better performance.
- Conducted performance testing and optimization of **Java**-based applications to ensure scalability and stability under heavy load conditions.
- Improved query performance by optimizing database indexing and **SQL queries** in **MySQL** to enhance data fetching and update times.
- Developed automated test cases using **JUnit** and **Selenium**, increasing testing coverage and reducing manual testing efforts.
- Integrated **REST APIs** to facilitate seamless data flow between enterprise systems, ensuring efficient communication between external services.

Environment:
Java, Spring Framework, Spring Boot, Hibernate, REST APIs, JUnit, Selenium, HTML5, CSS3, JavaScript, jQuery, MySQL, Oracle, SQL, Jenkins, Maven, Git, Agile.

Education:

Texas A&M University | Corpus Christi | Master of Science, Computer Science | Jan 2023 – Aug 2024

Gitam University | Hyderabad | Bachelor of Technology, Computer Science | Jun 2012 – Apr 2016