Ramya Nuthakki

Phone no: 9016316121   Email: Rnthakki@gmail.com

PROFESSIONAL SUMMARY:

- 8+ years of experience in designing, developing, and maintaining enterprise-level web and cloud applications **using Java (Java 8/11), Spring Boot, React.js, Angular, and AWS** across Finance, Healthcare, Optical, and Banking industries.
- Proficient in full-stack development, specializing in **React.js, Angular, Spring Boot, and microservices** architecture with **AWS, Azure, and GCP** deployments.
- Expertise in building SPAs using **React.js and Angular**, implementing state management with **Redux and NGRX** for scalable applications.
- Skilled in **RESTful API** development, migrating legacy systems to **Spring Boot and GraphQL.**
- Hands-on cloud experience with **AWS (Lambda, EC2, RDS, S3, API Gateway, ECS)** and **Azure (App Services, AKS, SQL Database).**
- Strong database knowledge, working with **PostgreSQL, MongoDB, MySQL, AWS RDS, and DynamoDB,** optimizing **SQL queries** for performance.
- Implemented authentication & authorization using **OAuth 2.0, JWT, and Spring Security** for secure access control.
- Experienced in **DevOps & CI/CD pipelines**, utilizing **Jenkins, Git, Azure DevOps, Docker,** and **Kubernetes** for automation.
- In-depth knowledge of Java 8 features such as **lambda expressions, functional interfaces, and parallel streams** to improve code efficiency and performance.
- Proficient in integrating third-party libraries and tools like **NGRX, Material UI, Bootstrap, PrimeNG, and React Hooks** to streamline UI development and improve user experiences.
- Worked with event-driven architecture using **Apache Kafka, Spring Batch, and Azure Service Bus** for real-time processing.
- Optimized performance with caching (Redis), database indexing, and Java 8 features **(Lambda, Streams, Functional Interfaces).**
- Focused on scalable, high-performance solutions, leveraging serverless architectures (AWS Lambda, Azure Functions) and best practices in Agile environments.

Technical Skillset:

Languages	Java (J2SE 1.8/1.7/1.6/1.5/1.4, JEE 6 / 5), J2EE, JSP2.1/2.0/1.2, Servlets2.x/3.x, JDBC, JavaScript, TypeScript, Python, Go, C++, C
Frameworks	Spring, Spring Boot, Spring MVC, Spring Cloud, Spring Security, Spring Data JPA, Hibernate, Angular, React.js, Struts, Top Link, NGRX, Redux, Flask, Django Rest Framework, JPA, Spring Batch
Web Technologies	HTML5, DHTML, XHTML1.1, JavaScript, jQuery, CSS3, Bootstrap, Material UI, Angular Material, RxJS, WebSockets, XML1.0, Velocity Template, GraphQL, AJAX, RESTful Web Services
Databases	Oracle, DB2, MS SQL, MySQL, PostgreSQL, MongoDB, Redis, DynamoDB, AWS RDS, AWS DynamoDB
Version Control	GIT, CVS, SVN, Maven, JIRA
Methodologies	Agile (Scrum, TDD/BDD), Waterfall
Web /Application Servers	Apache Tomcat, JBOSS, Web Logic, WebSphere, Spring Boot
Other Tools	Docker, Kubernetes, Jenkins, Maven, Gradle, AWS (EC2, S3, RDS, Lambda, CloudWatch, API Gateway, SQS, SNS), Azure DevOps, Google Cloud Platform (GCP), Postman, Swagger, NGRX, RxJS, Selenium

WORK EXPERIECNE:

Client: US Bank, Minneapolis, MN   **Nov 2023**

- Present
Role: Sr Fullstack Developer
Responsibilities:

- Involved in all phases of the Software Development Life Cycle (SDLC), including requirement gathering, analysis, design, development, testing, release, and support for Bank's digital banking solutions.
- Experienced in Java 11 full-stack development using Spring Boot, Spring Cloud, Spring Security, Spring Data JPA, Spring AOP, Hibernate, and AWS services for secure and scalable banking applications.
- Utilized Java 8 features (Lambda expressions, Stream API) for efficient data processing with Spring MVC, Hibernate, and JUnit.
- Designed database solutions with AWS RDS and DynamoDB for secure financial data storage.
- Built Single Page Applications (SPA) using React.js, MVC architecture, components, and hooks.
- Developed and implemented REST APIs and migrated SOAP web services to Spring RESTful services using Spring Boot.
- Designed Java-based microservices with Spring Boot and integrated Apache Kafka for real-time event-driven communication.
- Implemented dynamic routing and navigation with React Router, including nested routes and route guards for secured pages to banking services.
- Implemented GraphQL endpoints and mutations for efficient data fetching and modification.
- Utilized NGRX store and RXJS for state management in a financial data banking application with React.
- Optimized React applications with React.memo, useCallback, and lazy loading to minimize unnecessary re-renders and improve performance.
- Developed and deployed backend services to AWS Cloud using EC2, Lambda, and RDS.
- Integrated DevOps processes using Docker, Kubernetes, Jenkins, and Gradle.
- Wrote unit and integration tests for React components using Jest and React Testing Library to ensure code quality and reliability.
- Worked on REST API development using Python (Flask, SQLAlchemy, Django Rest Framework) and Go.
- Developed and deployed reports using SSRS in MS SQL Server.
- Conducted parallel browser testing using Selenium Grid for cross-browser compatibility.
- Built Java desktop applications using Swing and JavaFX and Monitored RESTful services with Spring Boot Actuator for banking applications.
- Automated deployments and testing on AWS with Kubernetes, Docker, and Jenkins.
- Experienced in writing unit tests for Java, Go, and React.js for digital banking applications.
- Used React.js for dynamic views, model-binding to DOM, and data synchronization.
- Designed and developed user-defined functions (UDFs) in Redshift for custom financial data operations.
- Familiar with continuous integration and DevOps methodologies (CI/CD, Docker, Jenkins, Kubernetes).

Environment: Java 8, Java 11, Spring Boot, Spring Cloud, Kafka, React.js, AWS, JPA, Hibernate, MongoDB, Redis, PostgreSQL, REST API, JUnit, NGRX, Maven, Docker, Kubernetes, Jenkins, JAX-RS, AWS Lambda, EC2, DynamoDB, GraphQL, OAuth, Git, Postman.

Client: Fannie mae , Washington   **Dec 2021 – Oct 2023**
Role: Sr Fullstack Developer
Responsibilities:

- Developed responsive and scalable web applications using Angular for the frontend and .NET Core for the backend for mortgage and financial services.
- Implemented CRUD operations and integrated complex forms with Angular reactive forms for seamless data flow and validation for mortgage processing.
- Managed API calls, business logic, and data storage using Angular Services for mortgage application.
- Created reusable components and custom directives for consistency across the application.
- Improved user experience with Angular Router for navigation and dynamic component loading, reducing page reloads for mortgage apps.
- Enhanced UI with third-party libraries like Angular Material, Bootstrap, and PrimeNG.

- Built state management solutions using NGRX and Redux to ensure consistent app state.
- Utilized RxJS and Observables for efficient, asynchronous data handling.
- Implemented JWT and OAuth2 authentication for secure API access and role-based control.
- Developed RESTful APIs with ASP.NET Core Web API for seamless frontend-backend data exchange.
- Optimized Angular performance through lazy loading, AOT compilation, and change detection strategies.
- Integrated real-time data synchronization using WebSocket and HTTP with backend APIs.
- Leveraged Docker for application containerization, ensuring consistent and scalable deployments.
- Migrated applications from Azure to AWS, utilizing EC2, S3, RDS, and Lambda for enhanced scalability and performance.
- Implemented serverless architecture with AWS API Gateway and Lambda, reducing infrastructure overhead.
- Managed financial data storage and retrieval with AWS RDS and DynamoDB.
- Created CI/CD pipelines with AWS Code Pipeline and Code Build for automated deployment.
- Monitored application performance and errors using AWS CloudWatch and AWS X-Ray.
- Used AWS S3 for static file hosting, improving media storage and quick access.
- Deployed Angular apps on AWS S3 with CloudFront for enhanced performance and reduced load times.
- Automated resource deployment with AWS CloudFormation for consistent environments.
- Ensured secure communication between services through IAM roles, security groups, and VPC configurations.
- Deployed compute resources using AWS EC2, scaling applications with Auto Scaling groups based on traffic demand.
- Used Angular Router (RouterModule) to manage dynamic routing, guard routes with AuthGuard, and handle navigation efficiently.
- Implemented Azure DevOps for CI/CD pipelines, enabling automated testing and continuous deployment across multiple environments.
- Utilized Azure SQL Database and Azure Blob Storage for efficient data management and secure storage solutions.

Environment: Angular, TypeScript, HTML5, CSS3, Bootstrap, Angular CLI, RxJS, Angular Material, ASP.NET Core, .NET 6, RESTful API, JWT Authentication, AWS (EC2, Lambda, RDS, S3, CloudWatch, CloudFront, Elastic Beanstalk, ECS, Code Pipeline, CloudFormation), Azure (App Services, SQL Database, CLI), Postman, Swagger, Docker, AWS CLI, Git, VS Code.

Client: Cioxhealth, Alpharetta, GA **Sep 2019 - Nov 2021**
Role: Full stack Developer
Responsibilities:
- Developed dynamic, responsive web applications using React.js and Redux for efficient UI rendering and state management while preserving critical business logic for healthcare data management.
- Created reusable UI components with Material UI and Bootstrap, ensuring a consistent, responsive design across devices.
- Integrated RESTful APIs and GraphQL with React for real-time content loading and enhanced user interaction.
- Optimized React performance with Hooks (useState, useEffect, useReducer), lazy loading, code splitting, and React.memo improving user interactions for medical records processings.
- Developed full-stack solutions by combining Java (Spring Boot, J2EE) with React.js.
- Built and consumed RESTful APIs with Spring MVC, Spring Boot, and Spring Data JPA for secure data management.
- Deployed applications on Azure using Azure App Services and AKS, leveraging cloud scalability.
- Created a dashboard with interactive charts using React, Redux, and D3.js.
- Integrated MySQL, PostgreSQL, MongoDB, and Azure SQL Database with Java applications using JPA and Hibernate for high-traffic healthcare applications.
- Developed serverless applications with Azure Functions and containerized Java apps using Docker and AKS for modern healthcare standards.
- Implemented OAuth 2.0, JWT, and Azure AD for secure authentication and role-based access control.

- Managed asynchronous operations with Spring Batch, Kafka, and Azure Service Bus for event-driven architectures.
- Developed Microservices and APIs using Spring Cloud, Spring Security, Spring Boot, and Spring Integration.
- Optimized database interactions with Spring Data JPA and Hibernate for efficient queries and transactions.
- Used Oracle Coherence for data caching and Spring JDBC for managing data in Oracle databases.
- Developed and deployed scalable APIs and web applications on Microsoft Azure, leveraging Azure App Services and Azure Functions.
- Employed Apache Camel and IBM MQ for messaging, enhancing integration between distributed applications activities with healthcare IT standards.
- Worked with XML (DOM, SAX, XPath, XSLT) for data parsing and transformation.
- Enhanced real-time app scalability with Kafka 2.0 for distributed messaging.
- Managed version control with Git and Azure DevOps, supporting Agile practices and sprint activities.
- Contributed to Agile development processes, ensuring timely, high-quality deliveries.

Environment: React.js, Java, JPA, J2EE, Spring Boot, MySQL, PostgreSQL, MongoDB, Kafka 2.0, Oracle, Material UI, GraphQL, Docker, Web Services, JMS, XML (DOM, SAX, XPath, XSLT), IBM MQ, Apache Camel, Spring Batch, Azure (App Services, AKS, Functions), Azure DevOps, Maven.

Client: Lenskart, Bangalore, India   **Mar 2018- Aug 2019**
Role: Full stack Developer
Responsibilities:

- Developed UI components with AngularJS using MVC pattern and Dependency Injection, ensuring modular and scalable front-end development for e-commerce platform.
- Integrated Angular Data Binding and Reactive Forms to create a dynamic and interactive user experience seamless eyewear browsing and purchasing.
- Utilized Angular Directives (ngFor, ngIf, ngClass, ngStyle) to enhance UI flexibility and maintainability.
- Integrated AngularJS Data Binding in HTML5 for a rich, responsive user experience in shopping app.
- Used jQuery and the Sizzle selector engine to modify DOM content across browsers.
- Implemented MVC architecture with Spring Framework and used Spring IOC for lifecycle management of core objects.
- Applied TDD principles, writing unit tests and Grails Spock Test cases for controllers, services, and domain classes used Grails Pagination for data display in templates.
- Implemented Angular Services and HTTPClientModule for seamless API integration with backend services.
- Optimized application performance by implementing Lazy Loading, Ahead-of-Time (AOT) Compilation, and Change Detection Strategies.
- Developed SOAP Web Services with Spring-WS for user requests and inter-component communication.
- Configured AWS API Gateway for secure RESTful service management and exposed services efficiently.
- Designed and deployed Spring Boot Microservices on AWS Cloud, leveraging AWS Lambda for scalability.
- Created and managed Spring Boot Microservices for CRUD operations on assignment data.
- Utilized Angular Dependency Injection (DI) system to manage reusable services efficiently.
- Configured AWS services (API Gateway, RDS, S3) for secure API management, database storage,
- Managed containerized applications using AWS EC2, ECS, and EKS for deployment efficiency.
- Implemented secure authentication with AWS IAM roles and policies.
- Integrated AWS SNS and SQS for messaging and asynchronous processing.
- Developed personal projects showcasing expertise in Angular, such as a portfolio website, a task management app, or a real-time chat application.
- Parsed XML documents with SAX, DOM, and transformed using XSLT.
- Utilized AWS S3 for asset storage, log retrieval, and static file management.
- Developed stored procedures and triggers in MS SQL, writing complex SQL queries for data extraction.

Environment: JavaScript, Java/J2EE, JBoss, Spring Core, Spring MVC, JMeter, Hibernate, AngularJS,

Backbone.js, HTML5, CSS3, RESTful Web Services, Node.js, Bootstrap, JMS, jQuery, Jenkins, Oracle, AWS (EC2, ECS, EKS, Lambda, S3, RDS, API Gateway, IAM, SQS, SNS), Microservices.