Tejaswi Reddy Siddareddy

Full Stack Developer | Java, Spring Boot, React, AWS, SQL
Email: tejaswii913@gmail.com
Phone: 940-597-0948
Linkedln: www.linkedin.com/in/tejaswi-reddy-siddareddy-274614376

Summary

I am a full stack developer with 7 years of professional experience in designing, developing, and maintaining enterprise-scale web applications across financial, investment and multiple domains. My core competencies lie in Java, Spring Boot, React.js, AWS services, and SQL-based backend development. I specialize in building secure, scalable, and high-performance applications with end-to-end ownership from frontend components to backend APIs and cloud deployment. My expertise includes REST API design, CI/CD pipeline automation, infrastructure setup, and performance tuning. I am also experienced in integrating AI-based modules for intelligent decision making and automation.

Technical Skills

Programming Languages	Java 8+, JavaScript, TypeScript, SQL, C, C++, Python
Frameworks	Spring Boot, Spring MVC, Hibernate, JSF, JUnit, Mockito, TensorFlow, PyTorch, Scikit-learn, Keras
Web Technologies	HTML5, CSS3, JavaScript, jQuery, Bootstrap, React.js, AngularJS, Vue.js, AJAX, XML, JSON
Databases	Oracle, MySQL, MS SQL Server, PostgreSQL, MongoDB, Cassandra, DynamoDB, Couchbase, DB2, Hadoop
Architectures & Patterns	Microservices, Multi-Tier Architecture, DAO, DTO, MVC,
Cloud & DevOps	AWS (EC2, S3, RDS, Lambda, CloudWatch, Cognito, CloudFront, ECS, DynamoDB, IoT Core), Jenkins, Docker, Git
Web Services & APIs	REST, SOAP, JAX-RS, JAX-WS, JAXB, WSDL, Apache Axis, JSON-RPC, AI/ML model inference APIs

CI/CD & Build Tools	Maven, Gradle, Jenkins
Version Control	Git, GitHub
Testing Tools	JUnit, TestNG, Mockito, PowerMockito, Jasmine, Selenium
IDEs & Tools	IntelliJ, Eclipse, NetBeans, Spring Tool Suite, JBoss Developer Studio, Android Studio, Postman, JIRA, Jupyter Notebook, Google Colab
Methodologies	Agile (Scrum), Waterfall, Iterative development
AI/ML Tools	TensorFlow, PyTorch, Scikit-learn, Keras, Pandas, NumPy, OpenCV, NLTK, spaCy, Matplotlib, Seaborn, Jupyter, AWS SageMaker, MLflow, Hugging Face Transformers

Professional Experience

Full Stack Java Developer With AI/ML - Citibank
Mortgage Division – Citi Wealth Builder Platform
Duration: Jan 2024 - Present

Project Description: Worked on Citi Wealth Builder, a digital investing platform that automates investment based on user goals, risk profiles, and time horizons. The platform systematically monitors and rebalances portfolios. As a full stack developer with AI ML expertise, I contributed to every layer of application from UI to backend services and ensured AWS cloud deployment was secure and scalable and leveraged AI for risk scoring, portfolio forecasting, dynamic goal profiling, and smart recommendations to improve personalized investing.

Frontend:

- Designed modular and scalable React components using hooks, context API, and Redux to support dynamic user workflows.
- Created responsive and mobile-first UI with Tailwind CSS, ensuring compatibility across modern browsers and devices.

- Integrated investment analytics dashboards with D3.js to visualize real-time AI-enhanced portfolio performance and rebalancing forecasts.
- Used Formik for schema-based form validations across multi-step onboarding processes.
- Integrated RESTful endpoints and handled state management for asset allocation previews and ML-generated risk tolerance questionnaires.
- Built reusable notification components with dynamic rendering for alerts and investment events including AI-predicted risk events.
- Created accessibility-compliant interfaces adhering to WCAG standards, supporting keyboard navigation.
- Developed a guided tour for new users leveraging React Joyride and context-aware step tracking.
- Collaborated with UI/UX team to transform wireframes into production-grade screens with pixel-perfect accuracy.

Backend:

- Developed Java-based microservices using Spring Boot to handle goal matching, risk profiling, and asset distribution.
- Built dynamic rule engine logic for automated investment strategies based on time horizon, asset preferences, and ESG filters.
- Integrated portfolio rebalance logic using Spring Batch jobs that trigger based on user-defined conditions and ML anomaly detection.
- Secured APIs with Spring Security and JWT token mechanisms, ensuring role-based access control.
- Created caching layers with Redis to enhance performance of frequently accessed investment data.
- Applied Java 11 features such as var, Streams, and Optional to improve code readability and maintainability.
- Implemented audit trails for transaction and portfolio modifications using Hibernate event listeners.
- Integrated with internal customer service APIs to allow human override and monitoring of AI recommendations.
- Utilized AI modules for dynamic risk scoring, behavioral clustering, and portfolio forecasting, using externally trained ML models.

Cloud:

- Automated provisioning and deployment of services on AWS EC2 and ECS using CloudFormation templates.
- Configured S3 buckets for encrypted data storage of investment documents and user audit logs.
- Used AWS Lambda for serverless execution of asynchronous ML-triggered portfolio rebalance checks.
- Integrated SNS for user alerts, notifications, and AI-detected anomaly reports.
- Applied IAM policies to secure access between services and roles.
- Integrated AWS Secrets Manager to securely manage DB credentials and API keys.
- Enabled SQS queues for asynchronous processing between user events and backend triggers including AI pipeline outputs.

- Collaborated with DevOps team to maintain CI/CD pipelines via Jenkins and GitHub Actions.

Environment: Java, Spring Boot, React, AWS (EC2, ECS, Lambda, SQS, SNS, S3, Secrets Manager), MySQL, Redis, Docker, Jenkins, GitHub, IntelliJ, Maven, D3.js, Tailwind CSS, Python (for ML model integration), Scikit-learn, TensorFlow, AWS SageMaker (external model deployment), RESTful APIs for AI inference

---

Full Stack Developer - TruStage
Digital Signature Compliance Platform
Duration: Nov 2021 - Sep 2023

Project Description: Involved in building a compliance-focused digital signature platform for financial services. The application provided secure, identity-verified e-signatures integrated with fraud detection modules and AI-assisted anomaly alerts. The focus was on secure document handling, identity verification, and ensuring traceability across the signing process.

Frontend:

- Developed React UI components to manage multi-party signing flows, document previews, and signer role-specific screens.
- Used state machines for controlling multi-step e-signature processes and transitions.
- Implemented drag-and-drop components for placing signature fields dynamically.
- Built step indicators and validation triggers for guided signing experiences.
- Incorporated AI-based UI hints for new users based on document behavior analysis.
- Developed real-time progress indicators showing signing completion status.
- Created responsive design with styled components for adaptive rendering on mobile and tablets.
- Optimized performance by lazy loading and React Suspense in key modules.
- Used analytics dashboards to track drop-off rates and UI engagement metrics.

Backend:

- Developed AI integration module using pre-trained models to detect forged IDs and anomalies in signature timing.
- Built REST APIs using Spring Boot to manage digital certificates, signer flows, and document metadata.
- Implemented digital hash verification for e-signed documents with secure audit logs.
- Supported OAuth2 flows for financial institutions and enterprise clients with scoped access.
- Enabled multi-factor authentication with Spring Security and custom token generators.
- Supported cross-border compliance by incorporating region-specific digital signing standards.
- Built robust exception handling and fallback mechanisms for signer workflow resilience.

Cloud:

- Designed serverless document processing with AWS Lambda to handle upload and parsing workflows.
- Used AWS Textract for extracting fields from uploaded documents.
- Managed storage and version control of signed documents in AWS S3 with bucket-level encryption.
- Configured SQS to queue signing events and trigger status updates.
- Enabled CloudWatch for API latency tracking, signature timeout alerts, and debugging logs.
- Used AWS Certificate Manager (ACM) for managing document signing certs.
- Implemented continuous integration and deployment with Jenkins and CodePipeline.
- Utilized Amazon Rekognition APIs for facial verification modules.

Environment: Java, Spring Boot, React, AWS, PostgreSQL, Lambda, Jenkins, Git, Gradle

---

Full Stack Developer - Simmons Bank

Health Savings Platform – Tax-Advantaged Savings for Medical Expenses

Duration: Jan 2020 - Oct 2021

Project Description: At Simmons Bank, I contributed to the development of a secure, FDIC-insured Health Savings Account (HSA) web application, empowering users to manage tax-beneficial savings for qualified healthcare expenses. The platform offered streamlined account management, transaction visibility, tax summary generation, and eligibility-based recommendations—designed for usability, compliance, and performance.

Frontend:

- Built React-based UI for displaying HSA balances, tax benefits, and contribution history with real-time updates.
- Developed eligibility checker and personalized health expense dashboards using interactive forms and REST APIs
- Implemented secure digital check ordering (no-fee standard checks) and contribution tracking components.
- Created mobile-responsive UI with accessibility features and offline caching (PWA features).
- Integrated real-time transaction alerts and account activity monitoring.
- Designed modular A/B testable components to analyze engagement with savings tools.
- Added hooks to track user interaction for insights into most-used features.
- Incorporated multi-device sync for wishlist medical services and recurring contributions.

Backend:

- Developed Spring Boot-based microservices for secure HSA account access, contribution limits, and tax-deduction eligibility.
- Integrated Redis for session management and caching of frequent queries like expense categories and limits.
- Created services for monthly service fee application, transaction history storage, and interest accrual logic.

- Designed endpoints to retrieve and store healthcare expense logs for IRS documentation.
- Enabled JWT-based authentication, rate-limiting, and audit logging for financial compliance.
- Tuned JPQL queries and added indexing for rapid retrieval of transaction summaries and tax reports.
- Integrated feedback loop for tax-optimized saving recommendations based on user behavior.

Cloud:

- Deployed services to AWS ECS, scaled based on login traffic and peak healthcare periods (e.g., open enrollment).
- Used AWS DynamoDB for secure storage of session tokens, contribution metadata, and tax benefit flags
- Wrote AWS Lambda functions to asynchronously process incoming contributions and log tax impact in near real-time.
- Delivered static content (statements, documents) via S3 and CloudFront for secure, low-latency access.
- Automated deployment pipelines using Jenkins, supporting rollbacks and staging-to-prod promotions.
- Monitored and enforced monthly service charges ($3) through automated ledger functions.

Environment: Java, Spring Boot, React, AWS (ECS, S3, CloudFront, DynamoDB, Lambda), Redis, Jenkins, Docker, Maven, Git

---

Full Stack Developer - Actics
Integrated Medical Information Platform – Commercial Medical Affairs Division
Duration: Apr 2018 - Dec 2019

Project Description: At Actics, I worked on a global Medical Information (MI) platform that supports pharmaceutical and life sciences clients in delivering accurate, timely, and personalized medical content to healthcare professionals, patients, and caregivers. The system modernized the legacy contact center model by enabling scalable, multi-channel access- including live chat, chatbot, self-service portals, and adverse event reporting—while adhering to strict regulatory standards across different regions

Frontend:

- Developed responsive self-service medical information portals in React, enabling HCPs and patients to search drug FAQs, safety profiles, and support documents.
- Created reusable components for contact center scheduling, inquiry tracking, and case lifecycle visualization.

- Built adverse event and product complaint submission workflows with guided input validation for regulatory compliance.
- Integrated live chat interfaces and chatbot components powered by conversational AI to address frequent inquiries.
- Designed dynamic dashboards for MI specialists to review and respond to incoming medical content requests.
- Implemented multilingual UI support with locale-based routing and fallback strategies.
- Integrated accessibility and compliance features to meet HIPAA and GDPR standards.

Backend:

- Built microservices using Spring Boot to handle MI content delivery, structured response documents (SRDs), and real-time triage of incoming medical inquiries.
- Integrated with a configurable and validated MI database to manage therapeutic-specific data and FAQs.
- Developed secure endpoints for intake of adverse event reports, product complaints, and clinical trial information requests.
- Implemented flexible staffing model logic (Shared/Dedicated/Hybrid) for routing incoming queries to the right specialist or global hub.
- Created services to sync feedback data into analytics pipelines for improving chatbot and knowledge base performance.
- Enabled region-specific logic to comply with local laws and practices for different geographies.
- Developed audit logging and access control layers to ensure traceability and security of medical data.

Cloud:

- Deployed services on AWS ECS with intelligent autoscaling to accommodate regional traffic surges (e.g., product launches).
- Used AWS S3 and CloudFront to serve downloadable content like prescribing information and medical literature.
- Integrated AWS Comprehend Medical to extract clinical insights from inquiry text and improve classification accuracy.
- Built Lambda-based event-driven processes to monitor incoming MI queries and route urgent cases.
- Automated deployments with Jenkins CI/CD pipelines; included failover routing and staged rollouts for new features.
- Applied WAF configurations and API Gateway throttling to prevent misuse of patient-facing services.
- Used CloudWatch and custom alerts to monitor compliance metrics, SLA breaches, and system usage trends.

Environment: Java, Spring Boot, React, AWS (ECS, Lambda, S3, CloudFront), Jenkins, Docker, Redis, Git, DynamoDB

Education

Master’s in Computer Science (MS)
University of North Texas – GPA (4.0/4.0)

Bachelor’s in Computer Science (B.E)
Anna University - R.M.D Engineering College- GPA (9.36/10)