

# Teja Dasi

Bharath.Y@sbasotech.net | 972-945-5769

## PROFESSIONAL SUMMARY

- Full Stack Java Developer with **7+ years** of experience designing and developing scalable enterprise web applications using **Java, Spring Boot, Spring MVC**, and **React.js** across **healthcare** and **fintech** domains through the complete **SDLC**.
- Proficient in building **RESTful microservices** and **APIs** using **Spring Boot**, integrating with **React.js** frontends, and deploying **cloud-native** applications on **AWS, GCP**, and **Azure** platforms.
- Strong experience with the **React.js** ecosystem including **React Hooks, Redux Toolkit, RTK Query, React Router, Context API**, and component libraries such as **Material-UI** and **Ant Design**.
- Hands-on expertise in **Spring Security, OAuth 2.0, JWT** authentication, and implementing **role-based access control (RBAC)** in enterprise-grade healthcare and fintech applications.
- Skilled in relational and NoSQL databases including **PostgreSQL, MySQL, Oracle, MongoDB**, and **Redis** with deep knowledge of **JPA** and **Hibernate ORM** frameworks.
- Experience with **microservices architecture** patterns, service discovery using **Eureka, API Gateway**, load balancing, and inter-service communication via **REST** and **Apache Kafka**.
- Proficient in containerization using **Docker** and orchestration with **Kubernetes**, implementing **CI/CD pipelines** using **Jenkins, GitHub Actions**, and **GitLab CI** for automated delivery.
- Experienced in performance optimization of Java backend services using profiling tools, query tuning, caching strategies with **Redis** and **EhCache**, and **HikariCP** connection pool configuration.
- Hands-on with unit and integration testing using **JUnit 5, Mockito, Testcontainers**, and **React Testing Library**, consistently achieving over **80%** test coverage across backend and frontend modules.
- Knowledgeable in cloud-native development on **AWS** using **EC2, ECS, EKS, Lambda, S3, RDS**, and **CloudWatch** for deploying and managing scalable **Java Spring Boot** microservices.
- Experienced in data visualization and dashboard development using **Chart.js** and **Recharts** integrated with **React.js** components consuming Java REST APIs in analytics-heavy applications.
- Strong understanding of software design patterns including **Singleton, Factory, Builder**, and **Observer** applied consistently across **Java Spring Boot** enterprise applications.

## TECHNICAL SKILLS

<b>Programming Languages</b>	Java 17/11/8, JavaScript (ES6+), TypeScript, SQL, HTML5, CSS3
<b>Frameworks &amp; Libraries</b>	Spring Boot, Spring MVC, Spring Security, Spring Cloud, React.js, Redux Toolkit, Next.js
<b>ORM &amp; Persistence</b>	JPA, Hibernate, Spring Data JPA, MyBatis
<b>Frontend Technologies</b>	React Hooks, React Router, RTK Query, Axios, Material-UI, Webpack, Babel
<b>Databases</b>	PostgreSQL, MySQL, Oracle, MongoDB, Redis, DynamoDB, H2
<b>Web Services</b>	REST, SOAP, Microservices, GraphQL, WebSockets
<b>Streaming / Messaging</b>	Apache Kafka, RabbitMQ, AWS SQS, AWS SNS
<b>AWS Cloud</b>	EC2, ECS, EKS, Lambda, S3, RDS, VPC, IAM, CloudWatch, SNS, SQS, Step Functions, EMR, Secrets Manager, KMS
<b>GCP</b>	BigQuery, GCS Bucket, Cloud Functions, Cloud Dataflow, Pub/Sub, Dataproc
<b>Azure</b>	App Services, Azure SQL Database, Virtual Machines, Azure CDN, Notification Hub
<b>DevOps / CI/CD</b>	Docker, Kubernetes, Terraform, Jenkins, GitHub Actions, GitLab CI, Maven, Gradle, SonarQube
<b>Testing</b>	JUnit 5, Mockito, Testcontainers, React Testing Library, Cypress, Jest, Pact
<b>Practices &amp; Tools</b>	Agile/Scrum, JIRA, Confluence, Bitbucket, OpenAPI 3.0/Swagger, Spring Batch, Flyway

## WORK EXPERIENCE

**Komodo Health (Remote, San Francisco, CA)**

Sep 2025 - Present

**Senior Software Engineer** (*React.js, Java/Spring Boot, PostgreSQL, AWS*)

- Architected and delivered a unified clinical analytics portal using **12 Java Spring Boot microservices** and **React.js**, consolidating **5** disparate provider dashboards into a single platform and reducing clinician context-switching by **80%**.

- Designed and implemented **RESTful APIs** with **Spring Boot** and **JWT**-based authentication via **Spring Security**, enabling secure **role-based access control (RBAC)** for **500+** daily active healthcare professionals.
- Built a reusable **React.js** component library with **40+** components using **Hooks** and **Redux Toolkit** for state management, reducing frontend feature development time by **50%** across teams.
- Configured **AWS ECS** blue-green deployments for all **Spring Boot** microservices, achieving zero-downtime releases and eliminating deployment-related production incidents.
- Integrated **Apache Kafka** for asynchronous event-driven communication between **8 microservices**, decoupling provider data ingestion pipelines from the **React** analytics presentation layer.
- Applied **React.lazy** and **Suspense** for route-level code splitting and **Brotli** compression on **Spring Boot** server, reducing initial bundle size from **3.8 MB to 940 KB** and Time-to-Interactive by **65%**.
- Implemented **Hibernate** second-level caching with **Redis** for frequently queried provider records, reducing database load by **60%** and average API response time from **2.1s to 450ms**.
- Developed **Spring Batch** partitioned jobs for nightly provider-data reconciliation processing **15M+** records, replacing ad-hoc scripts and achieving **100%** SLA adherence.
- Set up **GitHub Actions CI/CD** with **Maven** build, **JaCoCo** 80% coverage gate, **SonarQube** quality gate, **Docker** image push to **ECR**, and **ECS** deployment, cutting deployment time from **45 min to 11 min**.
- Implemented **Spring Cloud Sleuth** and **Zipkin** for distributed tracing, enabling cross-service log correlation and reducing mean time to diagnose production issues from **4 hours to 30 minutes**.
- Implemented field-level **PHI encryption** in **Spring Boot** service layer using **AES-256**, ensuring **HIPAA** technical safeguard compliance across all provider and patient data APIs.
- Built **WebSocket** support using **Spring WebSocket** and **STOMP** protocol for real-time dashboard updates, providing clinicians live data refresh without manual page reload.
- Designed **GraphQL** endpoints using **Spring for GraphQL** as an alternative query interface for the **React** frontend, reducing over-fetching by **75%** for complex provider cohort views.
- Configured **Resilience4j** circuit breakers, retry, and rate limiter patterns across all external API integrations, improving overall system resilience and preventing cascade failures.
- Deployed **Spring Boot** services to **AWS EKS** with **HPA** autoscaling and **Helm** charts, enabling scaling from **3 to 20 pods** during peak provider data ingestion without manual intervention.
- Led **API-first** development using **OpenAPI 3.0** specifications, generating **TypeScript** client SDKs for **React.js** via **openapi-generator** and preventing **12** breaking changes from reaching production.
- Mentored **3 junior engineers** on **Spring Boot** microservices and **React.js** best practices through pair programming and technical documentation, accelerating team onboarding by **50%**.

**Environment:** Java 17, Spring Boot, Spring Security, Spring Cloud, React.js, Redux Toolkit, TypeScript, PostgreSQL, Redis, Kafka, AWS (ECS, EKS, S3, RDS, ECR, CloudWatch), Docker, Kubernetes, GitHub Actions, JUnit 5, Mockito, Testcontainers, SonarQube.

## Codal (Chicago, IL)

Sep 2024 - Apr 2025

**Software Engineer** (*React.js, Java/Spring Boot, PostgreSQL, Redis, AWS*)

- Developed patient scheduling and appointment management module using **Spring Boot REST APIs** and **React.js**, handling **500+** concurrent users with **99.9%** uptime during peak enrollment periods.
- Refactored **20+** class-based **React** components to functional components with **Hooks**, reducing codebase size by **30%** and eliminating state management inconsistencies across the portal UI.
- Implemented **Redis** caching in **Spring Boot** for lab-result API responses with **5-minute TTL**, reducing average response time from **3.2s to 0.9s** and improving clinician satisfaction score from **3.1 to 4.6**.
- Built a centralized **Redux** store with **RTK Query** for server-state caching, eliminating redundant API calls and reducing total data-fetching requests per clinician session by **55%**.
- Integrated **JWT** refresh token rotation in **Spring Security**, meeting **HIPAA** session management requirements without degrading the clinician user experience.
- Built a custom **React** hook library (**useForm**, **useValidation**, **useAuth**) with **Jest** unit tests published to an internal npm registry, eliminating **90%** of duplicated form logic across the application.
- Authored **OpenAPI 3.0** specifications for all **Spring Boot** endpoints and auto-generated **TypeScript** client SDKs via **openapi-generator**, enabling independent frontend development against mocked contracts.
- Configured **SonarQube** quality gates in **Jenkins** gating PRs below **80%** Java coverage and **75%** JavaScript coverage, eliminating low-quality merges into the main branch.
- Set up **Cypress** end-to-end test suite covering **8** critical patient workflows, catching **3** UI regressions per sprint before reaching production environments.
- Used **Flyway** for versioned database schema migrations in **Spring Boot**, eliminating manual deployment scripts and resolving environment schema drift across dev, staging, and production.
- Optimized **JPA** entity relationships with **JOIN FETCH** and batch fetching to eliminate **N+1 queries** in appointment-listing APIs, reducing database round trips by **80%** for clinical workflows.

- Containerized **Spring Boot** services with multi-stage **Docker** builds, reducing image size by **55%** and improving local developer startup time from **3 minutes to 40 seconds**.
- Built **Spring Batch** reconciliation job for weekly billing processing **200K** records with skip and retry policies, replacing manual Excel process and reducing billing errors from **3% to 0.02%**.
- Migrated **AngularJS** legacy components to **React.js** using an incremental approach, resolving cross-framework state conflicts and reducing bundle size by **45%** post-migration.

**Environment:** Java 17, Spring Boot, Spring Security, React.js, Redux Toolkit, TypeScript, PostgreSQL, Redis, MySQL, Docker, AWS (ECS, S3, RDS), Jenkins, GitHub Actions, JUnit 5, Mockito, Cypress, Flyway.

### Accenture (Hyderabad, India)

Jan 2022 - Aug 2023

**Software Engineer** (*React.js, Java/Spring Boot, Oracle, PostgreSQL, AWS, Azure*)

- Modernized a fintech client's loan-origination system from a **Struts 1** monolith to **Spring Boot** microservices using the **strangler-fig pattern**, migrating **100%** of traffic within **6 months** with zero downtime.
- Developed **React.js SPA** consuming **Spring Boot APIs** for loan application and approval workflows, replacing JSP pages and reducing average page load time from **18s to 1.4s**.
- Implemented **OAuth 2.0** with **Spring Authorization Server** and **PKCE** flow in **React.js** for a fintech client, meeting RBI compliance requirements and enabling **SSO** across **4** banking applications.
- Built automated payment reconciliation dashboard with **Spring Batch** querying **Oracle DB** and **React.js Chart.js** visualizations with **Apache POI** Excel export, reducing analyst effort from **8 hours to 12 minutes** daily.
- Optimized **Oracle** trading-ledger queries by adding composite indexes, rewriting **N+1 Hibernate** queries with **JOIN FETCH**, and introducing **EhCache**, reducing query latency from **6s to 0.4s**.
- Implemented idempotency controls in **Spring Boot** for fund-transfer APIs using **Redis**-backed request deduplication, eliminating duplicate transaction incidents and meeting banking compliance standards.
- Integrated **AWS KMS** server-side encryption for **S3**-stored customer KYC documents and enforced HTTPS via **Spring Security HSTS**, passing **ISO 27001** security assessment with full marks.
- Implemented audit logging using a **Spring Boot** audit microservice publishing events to **AWS SQS** persisted to **DynamoDB** with **7-year** retention for regulatory compliance.
- Designed **Kafka**-based event-driven notification pipeline for trade workflows, processing **50,000+** daily trade events with guaranteed delivery via dead-letter topic handling.
- Configured **Azure App Service** with **Azure CDN** for **React.js** static asset delivery, reducing global asset load time by **60%** for relationship managers across **8** regional offices.
- Led **API contract-first** development using **OpenAPI 3.0** specifications, preventing **12** breaking changes from reaching production through **Pact** consumer-driven contract test validation.
- Implemented **Spring Cloud Config Server** for centralized configuration management across **6 microservices**, enabling environment-specific changes without redeployment cycles.
- Built **Kafka** consumer groups with dynamic partition assignment for high-volume market-data ingestion, processing **500,000 records per minute** during peak market-hours load.
- Implemented **Spring Boot Actuator** Prometheus endpoint integrated with **Grafana** dashboards for API latency, error rates, and Kafka consumer lag metrics for operations team visibility.
- Conducted knowledge transfer sessions on **Spring Boot** microservice patterns for **8 team members**, reducing onboarding time for new engineers from **3 weeks to 5 days**.

**Environment:** Java 11, Spring Boot, Spring Security, Spring Cloud, React.js, Redux, Oracle, PostgreSQL, Kafka, AWS (ECS, S3, KMS, SQS, DynamoDB), Redis, Docker, Jenkins, Maven, JUnit 5, Mockito, JIRA.

### Archents (Hyderabad, India)

May 2018 - Dec 2021

**Software Engineer** (*React.js, Java/Spring Boot, MySQL, PostgreSQL, AWS*)

- Developed customer-facing web banking application using **Spring Boot REST APIs** and **React.js**, serving **50,000+** daily active users across **3** digital banking product lines in the **fintech** domain.
- Implemented **JPA** optimistic locking with **@Version** for concurrent loan-approval transactions and **Testcontainers**-based tests simulating **200** concurrent users, eliminating double-booking race conditions.
- Built multi-step loan application wizard in **React.js** with **Redux** state persistence and **Spring Boot** session-store API auto-saving every **30 seconds**, reducing form abandonment rate from **45% to 8%**.
- Implemented **Spring Security OAuth 2.0** Resource Server with **RSA**-signed **JWTs** for stateless authentication, removing sticky-session dependency and enabling horizontal scaling by **4x** during peak load.
- Built reusable **React** component design system with **25+** UI components documented in **Storybook**, enabling consistent UI delivery across **3** banking product teams.
- Implemented **Spring Boot Kafka** consumers for real-time loan-disbursement event processing, replacing polling-based batch jobs and reducing disbursement notification latency from **2 hours to 3 minutes**.

- Created customer notification service using **Spring @Async** backed by **AWS SQS** with dead-letter queue and Slack alerting on **3** consecutive failures, improving notification delivery reliability to **99.8%**.
- Integrated **AWS Secrets Manager** in **Spring Boot** for database credentials and third-party API keys, replacing hard-coded application properties and achieving zero credentials in version control.
- Built **React.js** data visualization dashboards with **Chart.js** for loan portfolio analytics, providing relationship managers real-time portfolio performance views refreshed every **5 minutes**.
- Configured **AWS EKS** with **HPA** autoscaling for loan-processing microservices, enabling scaling from **2 to 10 pods** based on CPU utilization during month-end repayment surges without SLA breach.
- Implemented cursor-based pagination in **Spring Boot REST APIs** with **React InfiniteScroll**, replacing offset pagination that caused performance degradation beyond **100K** records in the loan table.
- Built **Spring Cloud Gateway** with **Resilience4j** circuit breakers and rate limiting, protecting backend microservices from traffic spikes and transient dependency failures.
- Built **RESTful** endpoints for customer **KYC** document management with **AES-256** field-level encryption in **Spring Boot**, ensuring compliance with RBI data protection guidelines.
- Authored **250+ JUnit 5** and **Mockito** unit tests for loan calculation and repayment schedule modules, discovering and fixing **2** critical calculation bugs before production deployment.
- Set up **Jenkins** pipeline with **Maven** multi-module build, **JaCoCo** 80% coverage gate, **SonarQube** quality gate, and blue-green deployment to **AWS ECS**, compressing release cycle from monthly to bi-weekly.

**Environment:** Java 8/11, Spring Boot, Spring Security, React.js, Redux, MySQL, PostgreSQL, MongoDB, Redis, Kafka, AWS (EC2, ECS, EKS, SQS, Secrets Manager), Docker, Kubernetes, Jenkins, Maven, JUnit 5.

---

## PROJECTS

---

**ResearchGPT Pro** | *TypeScript, LangGraph, FastAPI, Pinecone, PostgreSQL, Next.js* **Jan 2026 – Feb 2026**

- Orchestrated an **Agentic AI/RAG** pipeline with **LangGraph** multi-agent workflows, **LLM** prompt engineering, and a 7-stage retrieval system spanning **25K+ papers** at **sub-1.2s latency**.
- Built a **React** portal with **Python/FastAPI** backend, SSE streaming, and **GenAI** integration over a **102K-node** knowledge graph on **AWS**, shipping to production in **under 2 weeks**.

**Sidekick, AI-Powered Indoor Navigation** | *React Native, Gemini AI, WebSockets, TTS, Sensor Fusion* **Dec 2025 – Feb 2026**

- Built a **real-time** indoor navigation system using **React Native** fused with **Gemini AI** vision inference and **BLE/Wi-Fi** fingerprinting, achieving **1 to 3m** positioning accuracy without GPS across multi-floor environments.
- Implemented **voice-guided navigation** with **TTS** and **WebSocket** streaming, integrating **60Hz IMU** sensor fusion and **Kalman** filtering, reducing position uncertainty by **70%** across 3 indoor test environments.

**Cloud Infrastructure Suite** | *Node.js, TypeScript, Terraform, Docker, AWS, GCP* **Feb 2024 – Apr 2024**

- Provisioned secure cloud infrastructure on **GCP** and **AWS** using **Terraform** for IaC and immutable VM images, reducing deployment time from **45 minutes to 8 minutes**.
- Configured **CI/CD** with rolling updates, **Pub/Sub** messaging workflows, load balancing, **SSL/TLS**, and **CMEK encryption** for compliance, improving release reliability by **30%**.

**PanoramaEd** | *React JS, Node.js, MongoDB, Redux, WebSockets, PWA, i18n, MVC* **Oct 2023 – Dec 2023**

- Built a **global abroad education platform** serving students across **15+ countries**, architected in **MVC** with a **React JS** frontend and **Node.js** backend, delivering localized experiences via **i18n** across **8 languages**.
- Implemented **real-time** event-driven messaging using **WebSockets**, reducing student inquiry response time by **65%** and increasing mentor engagement by **3x** over asynchronous workflows.
- Designed a **relational-style** schema in **MongoDB** supporting multi-entity relationships across universities, programs, and student profiles, enabling complex filtered search across **500+** institutions in under **10 seconds**.

---

## EDUCATION

---

**NORTHEASTERN UNIVERSITY**  
Master's in Computer Software Engineering

**Boston, MA**  
**Sep 2023 to Aug 2025**

**NATIONAL INSTITUTE OF TECHNOLOGY**  
Bachelor's in Computer Science

**Nagpur, MH**  
**Jun 2015 to Jul 2019**