Charan Tej B

+1 (469) 945-7475|ext.1025| [ctej9908@gmail.com](mailto:charanteja.bala16@gmail.com)

**Summary:**

* Accrued over 8+ years of robust experience as a **full-stack** engineer, showcasing a deep understanding of front-end technologies, **MVC** frameworks, **Restful** web services, and **database** design.
* Fluent in the **MERN** Stack, exhibiting vast experience in crafting Web Applications. Employed **React.js**, **Vue.js** for the client side, **Node.js/Express** for server-side logic, and harnessed **MongoDB**, along with **SQL** Server for efficient database operations.
* Highly durable messaging systems like **Apache Kafka, IBM MQ, RabbitMQ**, and **Active MQ** are used for asynchronous exchange of vital information between multiple business applications.
* Extensively utilized **Java 17** advanced features including **Sealed** Classes, Pattern Matching, Switch Expressions, Text Blocks, **Stream API**, Time API, Functional Interfaces, Multithreading, and Collection API.
* Containerized **Spring Boot** application using **Docker** and deployed it to **Pivotal Cloud Foundry PAAS** (Platform as a service).
* Developed robust and highly scalable applications based on **Microservice** architecture using **Spring Boot** and **Spring Rest** for REST APIs and utilizing **Kafka** for pub/sub model, integrated with **Spring Boot** using bootstrap properties and **Kafka listener** annotation.
* Worked with various frameworks in **Spring** such as **Spring IOC, AOP**, **Spring MVC, Spring Data JPA, Spring JDBC, Spring Batch, Spring JTA, Spring IO,** and **Spring Security.**
* Implemented authentication and authorization for web services using **OIDC (OpenID Connect)** on top of **OAuth 2.0**, as well as **SAML** (Security Assertion Markup Language) and **JWT** (JSON Web Tokens) for secure communication.
* Implemented **Spring Boot Netflix OSS** stack for efficient microservices deployment and auto-configurations, incorporating **Ribbon** for client-side Load Balancing, **Hystrix** for latency and fault tolerance, and **Zuul** proxy for client-side load balancing with Eureka service registry and **Spring Cloud Config Server** for managing properties across microservices
* Applied advanced routing techniques for seamless navigation and navigated through component lifecycle methods, including **componentDidMount**, **shouldComponentUpdate**, and **componentWillUnmount.**
* Revamped front-end Single Page Application (**SPA**) applications using contemporary component patterns. Transitioned from class components to functional ones, leveraging Hooks such as **useState, useEffect, useReducer, and useRef** for **streamlined state management**.
* Extensively worked using ORM frameworks like **Hibernate, Spring Data JPA, MyBatis, IBatis** and Java Persistence API (JPA), with a specialization in **Hibernate's** cache - Ehcache.
* Conducted comprehensive UI testing utilising **Karma, Jasmine,** and **Protractor** for **Angular, Mocha,** and **Chai** for **Vue.js** and **Backbone JS.**
* Collaborated with **Angular, Backbone JS,** and **Ext JS**, tailoring and developing front-end applications in adherence to the MVC design pattern.
* Led end-to-end SDLC for enterprise apps with **Java, Spring, SOAP, and REST,** reducing deployment time and costs by 25%. Pioneered on-premises to **Azure** migration, realising 25% performance improvement and 30% cost reduction, showcasing cloud architecture expertise
* Leveraged cloud solutions, including **AWS CloudWatch, Amazon RDS, Amazon EC2, Amazon S3, Amazon Glue,** and **Access Managemen**t.
* Executed configuration, deployment, and support of cloud services on **Amazon Web Services** (AWS). Launched diverse **AWS EC2** instances for efficient application deployment.
* Diverse experience working with SQL databases like **PostgreSQL, MySQL, DB2, Oracle 13c/11g, Microsoft SQL Server**, NoSQL databases like **MongoDB, Cassandra, Couch DB, Redis, AWS DynamoDB, Azure CosmosDB** and good experience with **stored procedures, functions**, and **triggers**, **Views, Materialized Views** using **PL/SQL**.
* Building real-time dashboard applications using **ElasticSearch, Logstash,** and **Kibana (ELK)**, as well as **Splunk** for visualizing, analyzing, and exploring data on Apache Web servers.
* Utilized **Axios/Fetch** to orchestrate **Web API** calls, fetching data and updating global state in **Redux**.
* Experience with story/ticket tracking tools like **JIRA, Rally, and ServiceNow** and version control tools like **SVN, GIT, Bitbucket, and Aws Codecommit**.
* Implemented **Test Driven Development (TDD)** using **JUnit4/5, Mockito, PowerMock, EasyMock, SpringJUnitRunner,** and **TestNG** for reporting generation and **Behaviour Driven Development** using **Cucumber** with **Gherkin** syntax.

**Technical Skills:**

|  |  |
| --- | --- |
| **Technologies** | HTML/HTML5, CSS2/CSS3, Bootstrap, DHTML, XML, XHTML, XSLT, JavaScript, AJAX, JQuery, JSON, Java j2EE, Spring, Spring boot, Hibernate. |
| **JavaScript Libraries** | Backbone.js, React.js, Node.js, VUE.js, Handlebars.js Angular.JS, Angular 2/4/6/8. |
| **Database** | MySQL, Oracle, MongoDB, PostgreSQL |
| **Version control** | GIT, SVN, CVS |
| **Built tools** | Maven, Gradle, Ant |
| **Database** | MySQL, Oracle, MongoDB, PostgreSQL |
| **Web Servers** | Web sphere, Web logic, Apache Tomcat, JBoss |
| **Methodologies** | Agile, Scrum, Waterfall Model |
| **Project Management Tools** | GitHub, Bitbucket, Confluence, JIRA, ServiceNow |
| **CI/CD Tools** | Jenkins, Bamboo |
| **IDE & Build Tools** | Eclipse, IntelliJ, SQL Developer, Dbeaver, Vscode |
| **Cloud Environments** | AWS, Azure |
| **Testing Tools** | Junit, Jasmine, Mockito, Protractor |

**Work Experience:**

**Sr Full Stack Java Developer**

**Cleveland Clinic | Dallas, Texas | May 2023 – present**

**Key Contributions:**

* Orchestrated end-to-end Software Development Lifecycle (SDLC) for **microservices**, utilizing **Spring Boot** with **Spring MVC, Spring Data JPA,** and **Spring Security** to ensure seamless development, deployment, and security standards.
* Employed Java 17 advanced features like **Sealed Classes, Pattern Matching,** and **Stream API**.
* Configured **Spring Cloud-Netflix** OSS stack for microservices deployment, incorporating services like **Eureka** for service registry, **Ribbon** for client-side Load Balancing, and **Zuul** for API Gateway.
* Implemented authentication and authorization for microservices using **OpenID Connect (OIDC)**, **Security Assertion Markup Language (SAML),** using **JSON Web Tokens (JWT)**.
* Developed and deployed **Microservices** using **Spring Boot** and **Spring Cloud Config** for dynamic properties and implemented **resilience 4j** circuit breaker pattern for **resilience**.
* Transitioned from class components to functional components, leveraging **React Hooks** like **useState, useEffect, useReducer**, and **useRef** for optimized stateful management.
* Experienced in **React JS**, utilizing concepts such as One-way data flow, **Virtual DOM, JSX,** and **React Native** for creating interactive UIs.
* Deployed **Spring Boot**-based **microservices** **Docker** containers using **AWS EKS** (Elastic Kubernetes Service).
* ​​Worked with **Kafka Zookeeper** to manage the cluster with broker/cluster technology. Implemented **Kafka Avro** schema validation and topic partitioning in conjunction with **Spring Boot** and on the consumer side used annotations like **KafkaListener**.
* Integrating **Elasticsearch** into the **Spring Boot** backend, empowering users with fast and accurate full-text search capabilities.
* Configured monitoring solutions, such as **Grafana**, to track the performance and health of microservices.
* Utilized **Kubernetes** features within **AWS EKS,** including **Horizontal Pod Autoscaler (HPA)** and **Helm**, to orchestrate and efficiently manage containerized Spring boot microservices.
* Integrated essential cloud services, including **AWS CloudWatch** for monitoring, **Amazon RDS** for relational databases, **EC2** for scalable virtual servers, and **S3** for secure object storage.
* Implemented **CI/CD** pipelines using tools like **Jenkins** or **GitLab** **CI**, ensuring automated testing and seamless deployment of microservices.

**Environment:** Spring Boot 3.X, Spring Data JPA, Node, Express, React, Redux, Jest, Jenkins, Docker, AWS EKS, Swagger, Kafka, Elastic Search.

**Java Full Stack Developer**

**DXC Technologies | Atlanta, GA | Aug 2022 – April 2023**

**Key Contributions:**

* Built applications from scratch using **Micro Service** Architecture, utilizing **Spring Boot** with **Kafka**, and the **MEAN** stack **(MongoDB, Express, Angular, and Node.js)**.
* Utilized the **Spring** framework, incorporating **Spring Boot** for rapid application development and **Spring MVC** for building **RESTful** APIs.
* Employed **Spring Data JPA** for seamless integration with databases and **Spring Security** for robust authentication and authorization.
* Implemented microservices architecture with **Spring Cloud**, using **Azure App Configuration, Azure Event Hubs, Azure Service Bus, and Azure Storage Queue**.
* Using the latest **Java 1.8** features, such as functional interfaces and lambda expressions, along with the **Optional** class for effective null value handling, and the **Stream API** for improved code structure
* Used **Maven** for **Spring Boot** and **Yarn** for **Vue.js**, ensuring smooth integration and efficient dependency management throughout the SDLC.
* Deployed microservices using **Kubernetes** for container orchestration and **Istio** as a service mesh for enhanced microservices management.
* Integrated Reactive **Spring Data** for **NoSQL** databases like **MongoDB** or **Cassandra**, providing flexibility in handling unstructured data.
* Utilized **PostgreSQL** for structured data storage, ensuring optimal performance and reliability.
* Incorporated **Apache** **Kafka** for event-driven architecture, enabling seamless communication between microservices.
* Utilizing the **Angular** framework with **NGRX**, experienced in achieving **MVVM** design flow.
* Implemented **Angular** **canActivateChild** interface to guard child routes within a parent route, ensuring consistent access control across nested routes.
* Implemented custom **Directives, Dependency Injection, Services, Routing, Pipes, Animations, Forms(Reactive & Template Driven), AOT (Ahead of Time compilation), Lazy Loading,** and **RXJS** with expertise in **Angular**.
* Implemented **Cypress** for end-to-end testing, ensuring robust and reliable user interfaces.
* Containerized applications using **Docker** for consistency and portability across different environments.
* Spearheaded the successful on-premises to Azure cloud migration utilizing services like Azure Virtual Machines, Azure SQL Database, and Azure Active, resulting in a 25% performance improvement and a 30% reduction in maintenance costs, and provided crucial backend application support for seamless operations.
* Implemented the **ELK Stack** (Elasticsearch, Logstash, Kibana) for centralized logging and monitoring, ensuring efficient tracking of application health and performance.
* Automated testing processes by incorporating tools such as **Mockito** and **PowerMock** for mocking and isolating components during testing, ensuring accurate and isolated test results.
* Integrated **TDD** into the **CI/CD** pipeline, where tests were executed automatically upon code changes to maintain code quality and identify issues early in the development lifecycle.
* Established **CI/CD** pipelines using **GitLab**, **Jenkins and Azure DevOps**, enabling automated testing and seamless deployment of application updates.

**Environment:** Java 1.8, Spring 4.x, Hibernate, Node, Express, Jasmine, Karma, Jenkins, Docker, OAuth, GitHub, Swagger.

**Full Stack Developer**

**Humana | Louisville Kentucky | Nov 2019 – July 2022**

**Key Contributions:**

* Analyzed business requirements to translate functional requirements into technical requirements using MVC-based **AngularJS,** migrated **JavaScript to AngularJS, and** replaced **JavaScript objects** with **AngularJS modules, views, and controllers.**
* Worked with **AngularJS** directives, components, filters and services to customize controllers.
* Built **Angular Controllers** to control the data flow and handle user-raised events. Built Angular factories and services to facilitate communication across independent controllers.
* Effectively used **Angular** **Directives, filters, declarative templates, service providers, and context-aware communication**.
* Implemented **Java8** features like **Lambda expressions, Streams, filters, optional, Completable Features, and Functional Interfaces**.
* Worked on **Spring MVC** Transaction Management, **Spring/Hibernate** Batch Transactions and Hibernate cache and developed the application using design patterns like **Singleton, Factory, Delegator, Strategy, DTO prototype**.
* Implemented the **ORM** framework using **Hibernate** created entity mapping, Criteria API for dynamic queries, Association Mapping for performance tuning, and **HQL** used to communicate with databases to access data.
* Used **Jasmine** on Karma to test cases of **AngularJS** code and interacted with the QA team to fix issues.
* Added interactive charts using **AngularJS** Directive, implemented logic for dynamically changing data in scope and watched the chart update automatically.
* Orchestrated the deployment of microservices to a cloud platform, Pivotal Cloud Foundry for its cloud-native capabilities or AWS for its comprehensive service offerings.
* Configured IBM MQ connection, implemented a message receiver with **Spring** @JmsListener, and developed a parser to extract and store relevant data into an **Oracle** database, enhancing data accessibility and reducing reliance on mainframe systems.
* Implemented **Spring Batch** for processing large volumes of records in the application, utilizing **Quartz** scheduler for job scheduling and management
* Configured and managed public/private cloud infrastructure utilizing Amazon Web Services like **Amazon Ec2, and Amazon S3**.
* Configured **Jenkins** to handle application deployment in the Pivotal Cloud Foundry (PCF) server and integrate with GitHub version control.

**Environment:** Java, Oracle, AWS S3, Spring Batch, Hibernate, Node, Express, Jasmine, Karma, Jenkins, Docker, GitHub, IBM MQ.

**Java Developer**

**Micro Pyramid Informatics Pvt Ltd| India | June 2017 – Oct 2019**

**Key Contributions:**

* Developed the backend using **Node.js**, leveraging its event-driven, non-blocking I/O model for scalable and high-performance server-side applications.
* Implemented the server-side logic using **Express.js**, a minimalist and flexible Node.js web application framework.
* Utilized **Express** middleware for **handling requests**, **routing**, and managing **HTTP** **responses** efficiently.
* Designed and implemented **RESTful** APIs to facilitate communication between the client and server.
* Ensured adherence to **REST** principles for resourceful and stateless communication.
* Interfaced with databases, such as **MongoDB**, using appropriate **Node.js** libraries **Mongoose** for MongoDB.
* Utilized **callbacks**, **Promises**, or **async/await** syntax for managing asynchronous tasks.
* Implemented unit testing using frameworks like **Mocha,** and **Jest** to ensure the reliability and correctness of the code.
* Utilizing features of the **Spring** framework such as **Spring Dependency Injection, Spring Security, Spring JPA,** developed an application using the **MVC** architecture.
* ​​To distinguish business logic from cross-cutting concerns, used **Spring AOP** in creating custom aspects like logging, auditing, security, and transaction management
* Integrated monitoring tools **New Relic**, and **PM2** to track application performance and identify potential bottlenecks.
* Worked on  **MySQL DB** normalizing data and created **Stored Procedure**, **Trigger** and Materialized Views for query performance improvements.
* Implemented Multithreading using **Executor Service** for handling multiple requests and also worked on **CompletableFuture** in making async API calls.
* Containerized the **Node.js** application using **Docker** for consistent deployment across different environments.
* Established **CI/CD** pipelines using platforms like **Jenkins**, and **GitHub** Actions for automated testing and deployment.
* Implemented security best practices, including input validation, parameterized queries to prevent **SQL injection**, and measures against common vulnerabilities.
* Maintained comprehensive documentation for the project, including API documentation using tools like **Swagger** or **API Blueprint**.
* Implemented user authentication using technologies like **OAuth**, or **session-based** authentication.
* Automated recurring tasks and processes using **task runners** like **Grunt** for enhanced development workflow.

**Environment:** Java 1.8, Spring 4.x, Hibernate, Node, Express, Jasmine, Karma, Jenkins, Docker, OAuth, GitHub, Swagger.

**Java/J2EE Engineer**

**Sparkzen Technologies, India | Oct 2015 – May 2017**

**Key Contributions:**

* Engineered the complete Software Development Lifecycle (SDLC) for enterprise applications, leveraging **Java, Spring, SOAP, and REST**; implemented robust processes that drove a 25% reduction in deployment time and maintenance costs for clients.
* Developed **SOAP**-based service functionalities to enable seamless Check-in, Refund, Booking and cancellations, Flight Operations, and other features; resulting in a 40% reduction in processing time.
* Executed automation script integration for **CI/CD** pipelines using **Jenkins and Git**, resulting in a 50% increase in deployment frequency and a 40% reduction in merge conflicts, ensuring streamlined version control and integration processes.
* **REST API** development for key airport functionalities like Display Booking, Display Ticket, and FLIFO, serving over 100,000 requests daily with 99.9% uptime using **Java, and Spring**.
* Implemented **object-relational mapping** (ORM) to optimize data storage by utilising **Java Persistence API** and **Hibernate**, resulting in a reduction in database query times and a 25% increase in overall system performance.
* Engineered cost-efficient solutions in the AIS application, reducing **mainframe** system dependencies by 40%, leading to an annual 30% reduction in operational costs.
* Created SHADOW app with **Java 8 & Spring Boot**, enhancing SHARES incident documentation efficiency, achieving a 30% reduction in issue resolution time.

**Environment:** Java, J2EE, Spring 3.2, Spring MVC, Hibernate, Angular, RESTful Service, Tomcat 7, JPA, Apache CXF Soap.