**Bhargav Ram**

**Senior Full Stack Java Developer**

[**bhargavram7788@gmail.com**](mailto:bhargavram7788@gmail.com)

**+1 3318066776**

**ProfessionalSummary: -**

* **10+ years** of strong software experience working with different **SDLC (Software Development Life Cycle)** process including Requirement gathering, Analysis, Design, Coding, Testing, and Implementation like **AGILE (SCRUM), TDD** of web-based Client-Server business applications,
* Having good knowledge in front end technologies like **HTML5, CSS4, Android, AJAX, JSON, DOM, TypeScript, Bootstrap, XHTML, DHTML, JavaScript, jQuery, node JS, Tag libraries, Angular**
* Designed pages using **Angular 8 Validations**, **Modules, Events, Expressions, Directives, Controllers and Filters.**
* Knowledge in developing applications using **JAVA/J2EE technologies Hibernate, Servlets, JSP, JSF, AJAX, JDBC, JNDI, GWT JavaBeans, EJBs, JSTL, Swing, Struts, XML, Web Services, JMS.**
* Created **Spring Boot with Microservices utilizing RESTful API.**
* Implemented **Microservices architecture using Spring Boot for all the Orchestrators to consumer Adapters**
* Experience on **Spring Framework-Spring** **Core, Spring Observer design patterns, Spring IOC, Spring MVC, Spring ORM, Spring AOP, Spring Boot, Spring** **Web flow** and hands on experience in working with **Spring Batch** for monitoring, logging into the application, tracing, application support, transaction, and resource management.
* Experience in implementing and deploying **JAVA EE design** patterns such as **MVC, Singleton, Factory, Observer design patterns** in the development of multi-Tier distributed Enterprise **Microservices** **Applications**.
* Good knowledge in core **JAVA** concepts like **Collection frameworks, Interfaces, Synchronization, Serialization, Thread pools, JavaBeans, Externalization and Generics**
* Experience in designing and implementing scalable API management solutions using **API Gateways** like **Apigee, AWS API Gateway**, and **Kong** to ensure high performance and security.
* Proficient in identifying and resolving issues related to **API Gateway** configurations, policies, and performance bottlenecks.
* Knowledgeable in adaptive **MFA** solutions that dynamically adjust authentication requirements based on user behaviour and context, enhancing security.
* Strong working understanding of relational database systems such as **MySQL, MSSQL, PL/SQL, Oracle, DB2, and SQL, as well as NOSQL database systems such as Redis, PostgreSQL, MongoDB, CouchDB, and Cassandra.**
* Experienced in **time-based token authentication methods, including One-Time Passwords (OTP)** and **Google Authenticator,** for **secure access**.
* Proficient in developing **IAM** strategies to control access, strengthen security, and maintain compliance with industry standards and regulations.
* Experience in database development of **PL/SQL** stored procedures, and packages in relational databases like **10g / 11g, SQL Server, DB2, MySQL**
* Hands-on experience in implementing **NoSQL Databases like MongoDB, Apache Cassandra and CouchDB.**
* Experience in **Amazon Web Services (AWS) and Pivotal Cloud Foundry (PCF)**
* Design, Developed and Deployed enterprise advice applications using **spring boot for REST services in EC2, Fargate**
* Experience on **AWS platforms ECS, ELB, ROUTE 53, Dynamo DB, S3, Lambda, Cloud Watch, Kubernetes, IAM Role and CloudFormation Stacks**
* Building **Microservices using Golang, JSON, Docker, MongoDB and Redis.**
* Designed and implemented **RESTful APIs and REST Easy**, adhering to best practices and industry standards.
* Building **Microservices** using spring Framework and cloud-based architectures and deploying in **EC2, Fargate**
* Ability to identify and gather requirements to define a solution to be built and operated on **AWS**.
* Deploying and maintaining production environment using **AWS EC2 instances and ECS/EKS with Docker.**
* Used AWS SDK for connecting to **Amazon S3Buckets and EC2** to deploy on Docker Container
* Expertise in relational databases such as **Oracle, PostgreSQL, SQL server, and MYSQL, as well as PL/SQL.**
* Created **CI/CD pipeline** by using Build and Deploy plan spec template for **Services Batch Projects in ECS and EKS**
* Experience in Continuous Integration **(CI) like Jenkins, Bamboo, TeamCity, and Continuous Deployment (CD) tools like EC2 and Spinnaker**.
* Integrated cryptographic solutions for securing sensitive data in RESTful services.
* Implemented a distributed messaging queue to integrate with **MongoDB using Kafka and Zookeeper.**
* Experienced in managing **Terraform** **configurations** at scale, including managing multiple environments (e.g., development, staging, production) and handling complex dependency graphs between resources.
* Experience in setting up **Kafka cluster** setup on the Non-Prod and Production environments.
* Experience in using **Kafka** for Publish/Subscribe pattern in application dealing with messaging.
* Experience in configuration and deployment of multi-tier applications using different web servers like **Apache Tomcat, WebSphere, Apache Axis, Apache HTTP Server, JBoss and WebLogic.**
* Familiarity with **OpenShift** monitoring and logging tools such as **Prometheus, Grafana, and Elasticsearch/Fluentd/Kibana (EFK),** enabling observability and troubleshooting of applications and infrastructure components
* Experience in working with different **IDEs like Eclipse, NetBeans, IntelliJ, STS, Visual Studio, Web Strom**
* Experience with project build/test/deployment tools like **Maven, Gradle and Ant.**
* Experience in using code repository tools - **Tortoise SVN, GitHub, and Visual Source Safe.**
* Worked with **Log4j2** for logging and used **JUnit and Mockito** for unit testing and integration testing.

**Technical Skills: -**

|  |  |
| --- | --- |
| **Programming Languages** | **JAVA**, J2EE, Python, SQL, PL/SQL |
| **Web Technologies** | HTML5, CSS3/4, **JAVA**Script, jQuery, AngularJS, Angular2, TypeScript, Ajax, Node JS, Express JS, JSON, Swagger UI, XML, Angular 8/11/14 |
| **Spring Frameworks** | Spring Boot, Spring MVC, AOP, IOC, Spring Security, Spring Batch |
| **JAVA/J2EE Technologies** | Servlets, JSP, JSTL, EJB, JDBC, RMI, XML, JSF, Struts2, Swings, AWT, Applets. |
| **Application/web Server** | Apache Tomcat, WebSphere, JBoss, WebLogic Server |
| **Data base** | Oracle, DB2, MySQL, MongoDB, DynamoDB, Cassandra |
| **O-R MAPPING** | Hibernate, Spring data, JPA |
| **Methodologies** | Agile, Scrum, waterfall model. |
| **Cloud Environments** | AWS, Pivotal Cloud Foundry, Azure |
| **Version Control** | GIT, CVS, SVN |
| **Web Services** | RESTful – JAX-RS, Jersey, SOAP – WSDL, JAX-WS |
| **Reporting Tools** | Crystal Reports XI, Jasper Reports 6.0, SSRS 5.0 |
| **Testing Tools** | Junit, Jasmine, TestNG, Mockito, Protractor, Selenium, Cucumber, Log4j2 |
| **Tools** | Kubernetes, Docker, Ansible, Apache Camel, |
| **IDE ad Build Tools** | Eclipse, IntelliJ, NetBeans, Toad, SQL Navigator, Ant, Gradle, Maven, Jenkins, |

**Professional Experience: -**

**Verizon, NJ**

**Senior Full Stack JAVA Developer**   **Jun 2023- Present**

**Responsibilities:**

* Designed dynamic and multi-browser compatible pages using **HTML5, CSS3, jQuery** and **JAVAScript**.
* Built **React** application utilizing Redux for state management.
* Worked in **React JS** for creating interactive UI using One-way data flow, Virtual DOM, JSX, and React concepts.
* Used **React-router** to route the pages on the client-side and built **SPA**.
* Developed a Single page application and structured the **JavaScript** code using the **React.JS** framework including **MVC** **architectures, different modules, specific controllers, templates, custom directives, and custom filters.**
* Incorporated **react-redux** for global state management, for the **authentication process**, and added **redux-thunk** for **asynchronous handling**.
* Oversaw migration of the monolithic application to cloud-hosted **microservices** using **Spring Boot** and **Rest APIs.**
* Successfully migrated data from **MySQL/PostgreSQL** databases to **Amazon Aurora DB**, ensuring data integrity and minimal downtime.
* Created the **React JS** components and triggered **Angular 14** code to render the **React** components using life cycle hooks.
* Developed and maintained automated **Cypress** test suites for comprehensive **UI testing**, **enhancing the reliability** and user experience of **web applications.**
* Developed **JAVA** applications using **JAVA 11/17,** leveraging new features like **Lambda expressions** and the **Date-Time API**, **Sealed classes, Pattern Matching for ‘switch’** to enhance code efficiency and readability.
* Used **Microservices architecture**, with **Spring Boot-based services** interacting through a combination of **REST** and **Apache** **Kafka** message brokers. Scaled **Microservices**, distributed systems, and **Serverless** applications using Simple Queue Service.
* Proficient in identifying and resolving issues related to **AWS API Gateway** and **Apigee API Gateway** configurations, including Lambda integration, OAuth policies, throttling, quota enforcement, and response caching.
* Developed **REST APIs** using **Spring Boot**, leveraging its robust ecosystem for dependency injection, security, and integration with Cassandra database for scalable data management.
* Developed and deployed messaging solutions using **Apache Kafka APIs** in Java, enabling real-time data streaming and event-driven architecture with high throughput and fault tolerance.
* Used **Spring Data JPA** to develop **JPA Repository** for accessing the database to **CRUD (Create, Read, Update, and Delete) data**.
* Authenticating Users and providing security to **Restful** **web services** using **OAUTH 2.0.**
* Worked on Developing the Application using **Spring MVC** Framework by implementing **Controller, and Service classes.**
* Designed and developed Business Services using **Spring Framework (Dependency Injection), Business Delegate, and DAO Design Patterns.**
* Experience with **Spring Data** **JPA** for seamless integration with **PostgreSQL**.
* All the functionality is implemented using **Spring** **Boot** and **Hibernate ORM**. Implemented **JAVA EE** components using **Spring MVC, Spring IOC, Spring transactions, and Spring security modules.**
* Hands-on experience in using message brokers such as **ActiveMQ** and **RabbitMQ**.
* Participated in the design and construction of **PostgreSQL** database tables.
* Developed **Spring Annotations** to simplify the business logic for creating **Spring MVC RESTful web services**.
* Developed end-to-end data processing pipelines that begin with receiving data using distributed messaging systems **Kafka** through the persistence of data into **HBase**
* Worked on **MongoDB** database concepts such as locking, transactions, indexes, Sharing, replication, and schema design.
* Created multiple databases with shared collections and chose shard key based on the requirements.
* Used Swagger Framework for **Restful API** documentation as **Swagger** gives the rich **UI** to test the **API**.
* Implementing a Continuous Delivery framework using **Jenkins, CHEF, and Maven in a Linux environment.**
* Set up the **Continuous Integration (CI) and Continuous Deployment** (**CD**) process for the application using **Jenkins**.
* Designed and developed the **REST-based** **Microservices** using **Spring Boot, Spring Data with JPA** (**JAVA Persistence API**), and deployed **Microservices** onto **Docker**.
* Used **Kafka** messaging with spring integration and **Kafka Clusters**.
* Integrated **Datadog** with **CI/CD** pipelines, automating deployment checks and monitoring.
* Conducted in-depth data analysis using **Datadog** to generate performance reports.
* Worked on **AWS Security Groups** which acted as virtual firewalls that controlled the traffic allowed to reach one or more **AWS EC2 instances.**
* Developed unit tests and integration tests using **Junit** and **Mockito** frameworks and tested components during development.
* Wrote the **SQL queries**, view and materialized views, sequences, index, and triggers.

**Environment: JAVA** 11/17, Microservices, Spring Boot 3.2, Apache Kafka**,** **Vue.js,** REST APIs, Spring RESTful, Spring cloud, Spring Data JPA, Spring MVC, Hibernate 4.0, Api Gateway, Kafka, Mongo DB, Swagger, Jenkins, OpenShift, Maven, Docker, Log4J, AWS Cloud**,** Junit, SQL, JSON.

**Sabre, TX**  **Oct 2021 to May 2023**

**JAVA** Full Stack Developer

**Roles & Responsibilities:**

* Involved in development and testing of Streamlets module. Actively involved in Analysis, Design, Development, System Testing and User Acceptance Testing.
* Followed **Agile** **methodology** in the Scrum Cycle model.
* Migrated existing **UI application** with **HTML5**, **CSS4** and **Bootstrap** with **Angular** interaction.
* Create responsive, user-friendly web interfaces with **Angular 15/12** and **TypeScript**.
* Developed directives, Factories, Services in **Angular** and created Controller to isolate the scopes to perform operations.
* Used **JAVAScript, jQuery** for user interactions in pages .
* Create responsive, user-friendly web interfaces with **Angular 14/12** and **TypeScript**.
* Worked on **Angular** **14/11/8, Angular JS** framework to augment browser-based applications with MVC Capability, built dynamically generated dropdown lists using Ajax, **jQuery**.
* Proficiency in using tools like **Terraform** and **Ansible** for automating the provisioning and management of **ROSA clusters**.
* Implemented the function to send and receive **RabbitMQ** synchronously and asynchronously and send **AMQP** messages on MS message to Apache **ActiveMQ** on the edge device.
* Implemented **JAVA8** features like **Lambda** **expressions, Streams, filters, pipelines, Optional Interfaces etc**
* Development of **microservices** with **JAVA8, Spring Boot**, and deployment in Cloud Platforms.
* Utilized **AWS** **CloudWatch** and Performance Insights to monitor **Aurora DB** performance metrics and optimize resource usage.
* Used Executor Service Interfaces, Thread Pool Executor class for multi-threading thread resource pooling.
* Used **Multithreading and collections** to improve performance of applications.
* Developed classes using **Core JAVA/Executor Service (multithreading, concurrency, memory management).**
* Used **J2EE** design patterns like **Adapter, Service locator, Session facade, Singleton, Factory and Observer design patterns.**
* Created **Spring Boot** with **Microservices** utilizing **NoSQL** **RESTful** API.
* Designed and developed **Spring Batch Jobs** using **Spring Framework** and used Cron expression to trigger the job.
* Consumed **REST Web Services** for fetching the accounts information
* Developed **Restful Web Services** to interact with consumers and secured the service with **Spring- Security Oauth-2.0**
* Building **microservice** using Spring Framework with cloud-based architectures and deploying in **EC2 Fargate**
* Implemented **Microservices** architecture with **Netflix, Spring Eureka** and making calls to consumer applications
* Developed **Session Beans** for necessary Transactions like fetching the data and updating the customer information.
* Utilized **Spring core, AOP** and **security** to build robust business operations with transactional support
* Extensively used **Spring Batch** which uses **Spring IOC dependencies**, this is useful for developing **batch processing**.
* Design, Developed and Deployed enterprise advice applications using spring boot for **REST** **services** and deployed in **EC2 Fargate** in **AWS**
* Successfully integrated **Appium** tests into **Continuous Integration/Continuous Deployment** (**CI/CD**) pipelines, automating the testing process for seamless releases.
* Deployed and managed **Docker** containers using Rancher for various environments **(development, staging, production)**
* Orchestrated **Kubernetes** **clusters** with Rancher, ensuring high availability and scalability of applications.
* Configured **AWS Identity Access Management (IAM)** Group and users for improved login authentication.
* Worked extensively **with AWS services like EC2, S3, VPC, ELB, Autoscaling Groups, Route 53, IAM, CloudTrail, CloudWatch, CloudFormation, CloudFront, SNS,** and **RDS.**
* Set up and built **AWS** infrastructure with various services available by writing cloud formation templates in **JSON**
* Experience on relational database design and writing **SQL** queries, stored procedures, views, and functions with Oracle.
* Proficient in integrating **Terraform** with **CI/CD** pipelines using tools like **Jenkins, GitLab CI/CD, or GitHub Actions to automate infrastructure provisioning and deployment workflows.**
* Worked in a **CI/CD** environment with **Jenkins, Maven, Gradle, Ansible, AWS and Terraform**.
* Built and maintained **CI/CD** pipelines using **Jenkins**/**ArgoCD**, integrating Helm for Kubernetes deployments.
* Closely worked with the **Kafka** **Admin** team to set up **Kafka** **cluster** setup on the Non-Prod and Production environments.
* Configured and optimized **Elasticsearch**, **Logstash**, and **Kibana** for scalable data ingestion and visualization, enhancing operational insights for **JAVA**-based applications. Running applications on a cutting-edge cloud native infrastructure using **Kubernetes**, **Docker**.
* Developed unit and integration tests for existing **microservices** using **JUnit**, **Mockito**.
* Used **JMeter** to test performance of API.

**Environment: JAVA** 8, J2EE, Git, Spring Boot, Rest, Web Services, Eclipse, **JIRA**, AWS (ECS, EKS, S3, EC2, ELB, Autoscaling Groups, DynamoDB, CloudFormation, CloudWatch, IAM, Route 53), Spring batch, Oracle DB Apache JMeter, jQuery, JUnit, Docker, Terraform, Maven, Bamboo.

**Kroger, Charlotte, NC Dec 2019 to Sep 2021**

**JAVA** **Full Stack Developer**

**Roles & Responsibilities:**

* Involved in development and testing of **Microservices** with **Restful API**. Actively involved in Analysis, Design, Development and System Testing.
* Followed **Agile** methodology in the **Scrum** **Cycle** model.
* Implemented **JAVA**8 features like **Lambda** **expressions**, **Streams, filters etc**.
* Used Executor Service framework to implement multithreading in programming to improve overall performance.
* Developed classes using **Core** **JAVA**/Executor Service (multithreading, concurrency, memory management).
* Involved in developing the **Rest APIs** for backend.
* Created Spring Boot with **Microservices** utilizing **RESTful** **API**.
* Developing hybrid cloud solutions for Azure, Google, Soft Layer, and other platforms.
* Utilized Spring core, AOP and security to build robust business operations with transactional support.
* Extensively used Spring Batch for developing batch processing.
* Implementing or exposing the **Microservice** architecture with Spring Boot based services interacting through a combination of REST.
* Used WebSphere for Application server implementation, server setup configuration, support, performance.
* Experience in building **microservice** and cloud-based architectures.
* Monitor **Azure** Pipeline deployments using **CI/CD**.
* Integrated **Azure** **Active** **Directory** for **secure authentication** in **JAVA applications**, ensuring compliance with enterprise security standards.
* Designed and managed **Azure SQL** and Cosmos DB for **JAVA** applications, ensuring high availability and data consistency across cloud environments.
* Used apache camel to integrate the different applications using protocols like **http, Active MQ and JMS.**
* Developed **spring boot** applications for **REST** **services** and deployed in **Pivotal Cloud Foundry (PCF**)
* Developed **SQL** stored procedures and prepared statements for updating and accessing data from databases.
* Developed database objects-Stored Procedures, Tables, Views, Sequences and Triggers.
* Wrote **SQL queries** using stored procedures and enhanced performance.
* Cassandra, Docker, Azure, Spring Boot, and **JBOSS** were all used.
* Developed and maintained **Azure DevOps pipelines**, improving **CI/CD** efficiency.
* Set up and managed **Azure Kubernetes** **Service (AKS)** to **orchestrate scalable** **JAVA** microservices, ensuring high availability and efficient resource utilization.
* Developed comprehensive **unit and integration tests** for existing **JAVA** microservices using **JUnit** and **Mockito**, improving code coverage and reliability.Used **Log4j2** for tracking errors and debugging the code and resolved bugs
* Used Git as Version Control System for the application.

**Environment: JAVA** 1.7, J2EE, Git, Spring Boot, Hibernate, Core **JAVA**, SOAP/Rest, Web Services, CSS, Eclipse, WSDL, JIRA, JMS, NOSQL(MongoDB), ETL, Spring batch, PCF, Apache JMeter, Azure, jQuery, JUnit, Docker, Maven, Jenkins.

**Client: Humana, Louisville, KY Feb 2017 to Nov 2019**

**Role: Java Engineer**

**Responsibilities:**

* Responsible for requirements analysis, **technical design, implementation, testing** and **documentation.**
* Implemented **Spring application architectures** based on the **Model View Controller (MVC)** design pattern and **Spring framework** for **front-end development** and **presentation logic.**
* Implemented service layer using **Spring IOC** and annotations and Controllers using **Spring MVC**.
* Provided authentication and authorization by using **Spring Security and OAuth**.
* Implemented an **express-GraphQL library** to mount a **GraphQL API server** on the user-configured **HTTP** endpoint.
* Performed server migration using cloud services like **AWS** from physical to cloud environment.
* Utilized **AWS Lambda** platform to upload data into **AWS S3 buckets** and to **trigger** other **Lambda functions** developed drag and drop document editing.
* Used **Spring boot** created **REST APl's** per the application business to perform search using **elastic** **java sdk.**
* Worked in container-based technologies like **Docker**, **Kubernetes** and **OpenShift**.
* Designed **SOAP** based **Spring** Webservices and built user interfaces using **Java/Spring** technologies.
* Worked on **API Implementation** using **Restful** and **SOAP Webservices**
* Used **Hibernate** **persistence framework** on the backend against **Oracle database.**
* Created persistence layer using **Hibernate ORM framework**. Developed mapping files and **Hibernate** (**POJOs/DAOs**).
* Configured **Spring** to manage actions as **beans** and set their dependencies in spring context file and integrated middle tier with **Spring Web Layer**.
* Used **Spring Core** to define **beans** for **Services, Entity services** and corresponding depended services.
* Developed web pages **HTML, JSP, Portlets, AJAX, JSTL, JavaScript, ReactJS, Bootstrap, jQuery and CSS**.
* Implemented functionality like searching, filtering, sorting, validating using **ReactJS and JavaScript**.
* Performed and developed **REST API**-CRUD Operations using **GET, PUT, POST, DELETE** operations.
* Implemented **JSON Web Token (JWT)** to securely transfer claims between two microservices applications.
* Integrated **JMS** with Spring Boot by providing an instance which is embed with **ActiveMQ**.
* Used **PL/SQL** stored procedures for applications needed to execute as part of **scheduling mechanisms.**
* Implemented **SAML** for identity federation and secure exchange of authentication and authorization data.
* Developed the code environment using **IDE** as **Eclipse** and handled multiple **UNIX Shell Scripts** and **WebLogic Server** maintenance at the production servers and test environments for application maintenance**.**
* Troubleshoot the production incidents, analyse and provide the fixes using Java tools like **XSLT, SOAP, and PL/SQL and XML configurations**.
* Used **XML Web** **Services** for transferring/retrieving data between different providers.
* Used **Mocha** test runner and **Chai** test framework to test **ReactJS** components.
* Implemented all the components following test-driven development methodology using **JUnit**.
* Utilized **Log4j** to debug and record the logs in application.
* Provided end to end support for the testing activities during **Unit Testing, System Testing** and **UAT**.
* Used **JIRA** as an issue tracker tool to collaborate with testers and functional team as well as to track defects and change requests.

**Environment:** Java 8, Spring MVC, Spring Security, Hibernate, Restful Webservices, HTML5, CSS3, Bootstrap, ReactJS, Ajax, jQuery, Tomcat Server, XML, JSON, SAML, SQL, Oracle, Eclipse, GIT, Jenkins, Maven, Log4j, Mocha, Chai, Junit, Jira

**Sun Software technologies - India.**

**JAVA Developer Jan 2014 -Dec 2016**

**Responsibilities:**

* Involved in analysis, specification, design, and implementation and testing phases of **Software Development Life Cycle (SDLC)** and used **agile methodology (SCRUM)** for developing application.
* Created responsive **UI modules** using **Angular JS** with hard coded dependency of Underscore JS in combination with Handlebar JS for tinplating.
* Used **Angular JS** to utilize its rendering **functions** and create **controllers** to handle events triggered by clients and send requests to the server.
* Used **Spring AOP** Module to implement logging in the application to know the application status.
* Utilized features of the **Spring Core** and **Spring DAO** support layers to develop the application.
* Implemented the project structure based on the **Spring MVC** pattern using **Spring.**
* Implemented **authentication and authorization** for **Rest API** using **Spring Security**.
* Implemented **Spring Validators, persistence layer, and service layer components using Spring/Hibernate API and Spring/Hibernate annotations**.
* Leveraged **Hibernate Query Language (HQL)** extensively for efficient data retrieval and manipulation.
* Created a persistence layer using Hibernate for sending and retrieving data from the database.
* Used Spring AOP module to handle transaction management services for objects in any Spring-based application.
* Developed **SOAP** web services using **Apache-CXF** which would be available for other local OSGi bundles.
* Developed Several **SOAP-based JAX-WS Web Service** interfaces for integrating multiple systems and code reusability.
* Used **SOA Architecture to Develop REST Web services**, generated **WADL**, and consumed the services.
* Wrote **stored procedures**, and complex queries using **PL/SQL** to extract data from the database, delete data and reload data on **Oracle DB.**
* Worked on building **Docker** **Images** and running them on a **Docker** **container**.
* Used Jenkins as the integration tool and improved the scalability of applications on cross-platforms.
* Used **JIRA** to keep track of bugs and issues.
* Using **JUnit** test in Eclipse, developed Unit test cases and test suite. Curl was also implemented to test.
* Used **Maven** for build and deployment to **JBoss** server and **Splunk** as log monitoring.
* Used Git for version control and **JIRA** for bug Tracking and Project Management.

**Environment:** **JAVA** Server Faces, Spring IOC, Spring AOP, SOAP, Apache-CXF, JAX-WS, WADL, PL/SQL, Oracle DB, Hibernate, Maven, JBoss, **Junit**, Selenium.