

Name: Manoj Kumar
Contact at : +1 (206) 333-7804
Email: manojks8790@gmail.com

JAVA FULL STACK DEVELOPER

PROFESSIONAL SUMMARY:

- Acquired 8+ years of Full Stack development experience across all stages of **Software Development Life Cycle (SDLC)** including Analysis, Design, Development, Implementation and Testing of web based, enterprise business applications using different application architectures like **Monolithic, Service Oriented Architecture, Microservices, N -tier and Event-driven**.
- Worked with various **SDLC** methodologies including Requirement gathering, Analysis, Design, Coding, Testing, and Implementation like **AGILE (SCRUM), TDD, and Waterfall Models**.
- Well-versed in implementing and deploying **Java EE** design patterns such as **Singleton, Abstract Factory, DAO, DTO, and Business Delegate** for **Multi-Tier distributed Enterprise Applications**.
- Possess expertise in using **Core Java SE 11** concepts including **Streams API, Lambda functions, Functional programming, and Generics**.
- Experienced in constructing **SOA (Service Oriented Architectures)** and **Microservice architectures using J2EE web services** based on **Restful (JAX-RS), SOAP (JAX_WS), JAX-RPC, and JAXR (Java API for XML Registries)** utilizing frameworks like **Apache CXF, Jersey, Apache Axis**.
- Performed unit testing of web services using **Junit, Mockito Easy mock, Spring Junit Runner, Spock, TestNG** thereby accomplishing **Test-Driven Development (TDD)**.
- Worked extensively with **JAVA 11** features including **Time API, Functional Interfaces, Completable Future, Optionals, Multithreading, Transaction Management, Exception Handling, Collection API**.
- Possess vast expertise in **Spring** framework modules like: **Spring MVC, Spring Boot 2.7, Spring IOC, Spring IO, Spring AOP, Spring Security, and Spring Data**.
- Good knowledge in developing real-time dashboard applications leveraging **Splunk, ElasticSearch, and Logstash (ELK)** for profiling and monitoring streaming application data.
- Hands-on experience with **Spring Security** Framework for User Access Management, **JWT** token-based authentication, and **OAuth 2.0** configuration for authentication and authorization.
- Extensively utilized **Spring IOC** for injecting objects based on selecting components like **setter injection and Interface injection** to manage object references.
- Good grasp of Server-side **JavaScript** frameworks including **NodeJS 16, HapiJS, ExpressJS 6**.
- Hands-on experience in server-side rendering of Java content structures like **ReactJS 18** and rendering webpages on the server-side using **Redux** and **Flux** to achieve unidirectional data flow.
- Working knowledge of integration tools like **Spring Integration, MuleESB, Apache CXF, and Apache Camel** to integrate enterprise applications with existing **JVM** environments.
- Substantial experience in building server-side JavaScript frameworks with **NodeJS 16 (HapiJS, ExpressJS 6)** to construct event-driven, non-blocking i/o model web applications.
- Strong experience in developing Dynamic Single Page Applications (SPA) using **MEAN (MongoDB, Express, Angular 15, and NodeJS 16) full-stack development toolkit**.
- Proven expertise working with Application Servers including **IBM WebSphere, Oracle WebLogic, JBoss, and Tomcat**.
- Worked with various **Message Queue (MQ)** services such as **Apache Kafka, RabbitMQ, ActiveMQ, and IBM MQ**.
- Extensively involved in **UNIX Shell** scripting to automate build processes.
- Hands-on experience in integration testing tools including **Selenium Web Driver, Cucumber (Behavior Driven Development testing)**.
- Used **JUnit** Framework along with **Mockito and EasyMock, Spring Junit Runner, JMeter** for testing.
- Worked on testing **JavaScript** applications using **Mocha, Chai, Jasmine, and Karma**.
- Expertise in Various **Agile** methodologies like **SCRUM, Test Driven Development (TDD), Incremental and Iterative methodology, Pair Programming, Agile Development & Testing** using **Software Development Life Cycle**.
- Good knowledge of **Amazon Web Services (AWS)** including **EC2, VPC, IAM, Elastic Beanstalk, Lambda, S3, CloudFront, Glacier, SQS, SNS, RDS, DynamoDB, Route53, CloudWatch, CloudTrail, CloudFormation, Security Groups, etc.**

- Involved in operations and maintenance of **AWS cloud resources** including launching, troubleshooting, and maintaining **EC2 instances, S3 buckets, Auto Scaling, DynamoDB, AWS IAM, Elastic Load Balancers (ELB) and Relational Database Services (RDS)**.
- Designed and built highly scalable, available, fault-tolerant and secured distributed systems on **AWS** using services like **EC2, S3, EBS, API Gateway, Elastic Beanstalk, ECS**. Used **configuration management tools** like **Puppet, Chef, Docker, and AWS SDK**.
- Working knowledge of **Kubernetes** for deploying and scaling containerized applications.
- Good grasp of **Drools Rule Engine**, used for validating **BRMS (Business Rule Management System)**.
- Experience in using **NPM** and **BOWER** for package and dependency management.
- Involved in database design and development of **PL/SQL** stored procedures and packages in **RDBMS** like **Oracle, SQL Server, MySQL**, and **DB2**. Also possess experience in designing and building **NoSQL** Databases like **MongoDB**.
- Expertise in leveraging **JavaScript** build tools and task runners including **Webpack** for **React 18** and **Angular 15**, **Grunt, Gulp** for **Angular**, **Browserify** for packaging NodeJS modules.
- Hands-on expertise in build tools **ANT, Maven 3.8, Gradle 8** for Java projects. Experience integrating CI/CD tools like **Jenkins, Bamboo**.
- Worked on **JavaScript MVC** and **MVVM** frameworks like **Angular 15, AngularJS, Backbone.Js, Ember.JS, Handlebar.JS** for creating dynamic web applications.
- Developed various **Angular** artifacts like components, services, directives, and pipes following best practices, components for **UI**, services for business logic, directives for **DOM** manipulation and pipes for data transformations. Performed **unit** and **e2e** testing using **Jasmine, Karma**, and **Protractor** to achieve over 90% test coverage.
- Extensive experience in developing **ReactJS 18** components, forms, events, routers, and **Redux** asynchronous functions. Implemented **React - Flux pattern**.
- Developed **Typescripts (ES6)** components, **ngModules**, Providers, Services, classes leveraging arrow functions and optional types.
- Working knowledge of source code version control tools including **SVN, Git, BitBucket**.
- Involved in developing scalable, reliable, and efficient web and enterprise applications using **JSP, EJB, JNDI, JMS, Struts, JSF (PrimeFaces)**.
- Built multiple applications using **Spring** framework - **Spring MVC, Spring Boot, Spring ORM, Spring JDBC, Spring transaction Management, Spring Data, Spring Web Services, Spring JMS, Spring AOP**. Used **Spring** and **EJB** containers for managing bean lifecycle.
- Experience in developing **RESTful** services (**JAX-RS, Spring Boot**), **SOAP (JAX-WS, Apache CXF, WSDL, Apache Axis2)** and automated API documentation using **Swagger** and **WADL**.
- Expertise in interacting with **RDBMS** using **ORM** frameworks such as **Hibernate, Spring Data, JPA, MyBatis**, Java Persistence API (**JPA**), **Hibernate's** second level cache - **Ehcache**.
- Worked with **ODM** frameworks like **Mongoose Connector** and **Cassandra Node Driver** to connect asynchronous environments.
- Involved in designing and building Microservices based applications leveraging **Spring Boot** and **Spring Cloud**.
- Good experience in using **Spring Data JPA** for automatic generation of Data Access Objects (**DAOs**) at compile time.
- Knowledge of securing web services using **OIDC (OpenID Connect)** on top of **OAuth 2.0** for authentication and authorization.
- Hands-on experience in creating access tokens with **JWT (JSON Web Token)** to assert claims.
- Good understanding of **XML** languages including **SAX, XSD, DOM, XPath, DTD, JAXB, JAXP, XSL/XSLT**.
- Worked with asynchronous message queues like **JMS, RabbitMQ** and **Apache Kafka**.
- Used integration tools such as **Spring Integration, Apache CXF** and **Apache Kafka** for integrating enterprise applications with **JVM** environments.
- Involved in developing real-time dashboard applications using **Elasticsearch, Logstash** and **Kibana (ELK Stack)**, **Splunk** on **Apache** Web servers for data visualization, analytics, and exploration.
- Working knowledge of database caching techniques using **Redis, MongoDB Elastic Search**.
- Hands-on experience with **Pivotal Cloud Foundry (PCF)** for continuous delivery of applications to the cloud.
- Extensively worked with various **AWS (Amazon Web Services)** offerings including **EC2, S3, Glacier, EBS, Elastic Cache, DynamoDB, SQS, RDS, Lambda**.

- Good understanding of **Spring Cloud–Netflix OSS** components for microservices deployment including **Ribbon** (client-side Load Balancer), **Hystrix** (latency and fault tolerance library), **Zuul** (routing, monitoring, security) integrated with **Eureka** service registry and **ELK** for log management.
- Implemented **Docker** container service, **Jenkins**, **Bamboo** and **GoCD** pipelines for Continuous Integration and Continuous Deployment on cloud and on-premises environments.
- Good hands-on experience with **Java** logging framework - **log4j2**.
- Used **Kubernetes** for deploying and scaling containerized applications.
- Strong experience in automated testing of Java applications using **Selenium** and **Cucumber**.

TECHNICAL SKILLS:

Programming Languages	Java, JavaScript, C, SQL, PL/SQL
Java/J2EE Technologies	JSP, JSTL, JDBC, JPA, EJB, JMS, Servlets, JAX-WS, JAX-RS, JNDI, RMI, JAXB, JAXP, JAXR
Frameworks	Spring, Hibernate, Struts, Spring Boot, Spring Cloud, Spring Batch, Spring Security, Spring AOP, Spring Data, Spring ORM, Spring MVC, Spring Integration, Spring JTA
Web Technologies	HTML, HTML5, CSS, CSS3, XML, jQuery, Angular, ReactJS, AJAX, JSON, JSF, Bootstrap
Web Servers	Tomcat, WebLogic, WebSphere, JBoss, Apache HTTP Server
Databases	Oracle, MySQL, SQL Server, PostgreSQL, MongoDB, Cassandra, Redis, DynamoDB
Middleware	Web Services, SOAP, REST, Apache Kafka, RabbitMQ, JMS, ActiveMQ, IBM MQ
Methodologies	Agile, SCRUM, TDD, Waterfall, OOAD
Testing Tools	JUnit, Mockito, Selenium, Cucumber, JMeter, Jasmine, Karma, Mocha, Chai, TestNG, JIRA X-Ray
Build Tools	Maven, Gradle, ANT, Webpack, npm, bower, gulp, grunt
Version Control	Git, SVN, BitBucket
IDE Tools	Eclipse, IntelliJ IDEA, Spring Tool Suite, Visual Studio, NetBeans
OS & Scripting	Windows, Linux, UNIX, Shell Scripting
Cloud Technologies	AWS, Docker, Kubernetes, PCF
Monitoring Tools	ELK Stack, Splunk, Kibana, New Relic, Graphana, Datadog
Security	Spring Security, OAuth 2.0, OIDC, JWT

PROFESSIONAL EXPERIENCE:

Client: Capital One

Dec 2022 -present

Location: McLean, VA

Role: Sr. Full Stack Java Developer

Project: Developed an innovative digital banking platform, "Capital One 360", to deliver personalized and intelligent financial services through immersive digital experiences. The platform provided a unified suite of products including checking & savings accounts, credit cards, auto loans, and investments.

Responsibilities

- Developed Involved in various phases of Software Development Life Cycle including requirements gathering, analysis, solution design, development, testing, deployment, and maintenance by using **Agile** principles and values.
- Developed the single-page application following **Agile Scrum** methodology, participated in daily standups, sprint reviews, sprint retrospectives, backlog grooming sessions and used **Jira** for user story creation, assignment, sprint planning and project management.
- Configured **Docker** container service and build pipeline **Jenkins CI/CD** tool for continuous integrating and deploying applications on servers.

- Developed back-end interfaces using **SQL** packages, stored procedures, Functions, Procedure, Anonymous **SQL** programs, Exception Handling in **PL/SQL** programs.
- Developed unit test cases using **Mocha** and **Chai** as unit test runners to test various **React JS** components.
- Worked on the **Splunk** architecture and various components (indexer, forwarder, search head, deployment server), Universal and Heavy forwarder.
- The application is built in **Micro Service** Oriented Architecture using **MEAN** stack (**MongoDB 5, ExpressJS 5, Angular 14, Node.js 16**) to develop a set of small, autonomous services and deployed them independently on **AWS EC2** instances.
- Designed and implemented **Angular 14** components, ensuring compatibility with the most recent versions of **TypeScript (4.5)** and **RxJS (7.4)** through meticulous module management.
- Created reusable **Angular CLI 13**, utilizing its powerful capabilities to efficiently manage project workflows and streamline the development process.
- Designed responsive user interaction screens using **HTML5, CSS3, Bootstrap** framework by using grid system, media queries, mixins and other features for optimized experience across desktop and mobile.
- Implemented **MongoDB NoSQL** database for its dynamic schema design, horizontal scalability through sharding and built-in replication for high availability. Wrote aggregation pipelines, map-reduce queries for faster processing.
- Used advanced **Java 8** features like **Lambda** expressions, **Streams API, CompletableFuture** for asynchronous programming and new **Date/Time API** for robust date handling.
- Leveraged **Object-Document Mapper (ODM)** frameworks like **Mongoose** to define schemas for **MongoDB** documents and **Cassandra Node Driver** for interacting with **Cassandra** from **Node.js** asynchronously.
- Created **Kafka** producers and consumers using **Zookeeper** for coordination, managing offsets, and streaming large volumes of data by creating topics and partitions.
- Configured **JBoss Wildfly** application server in high availability mode and utilized **Hibernate ORM** with **Oracle Database** for object-relational mapping along with 2nd level caching using **EHCache** for better performance.
- Built scalable microservices architecture using **Spring Cloud Netflix** components like **Eureka** for service discovery, **Ribbon** for client-side load balancing and **Hystrix** circuit breaker for fault tolerance.
- Developed and deployed **Microservices** based applications using **Spring Boot** and **Spring Cloud** and implemented **Zuul** to enable routing and load balancing the requests to back-end web services.
- Created Logging aspects for auditing and logging purposes using **Spring AOP** and implemented security for the application using **Spring Security**.
- For better understanding about end points developed by backend developers. We used **Swagger** in the project. Which helps frontend developers to view end points and their documentation.
- Deployed **Spring Boot** based microservices **Docker** container using **Amazon EC2** container services.
- Handled Http Requests using **Axios** which supports **Promise API** functionality native to **JZ** oracle.
- Developed responsive applications using **Bootstrap.js, Require.js, Node.js, D3.js** which allowed the same application to be optimized for both mobile and desktop resolutions.
- Created and consumed **RESTful API** using **Express.js** to provide centralized logic and reusable data retrieval build on top of **Node.js**.
- Implemented **Node.js** router to make cross-domain requests to **API** services like **GitHub API, MongoDB**.
- Used **Node.js** to run **Grunt** tasks and compile, minify, concat and build the project properly.
- Worked with build tools like **NPM, Bower** and **Grunt**.
- Used continuous build using **Jenkins** and supported the Application for production deployment and post-production.
- Performed configuration, deployment and support of cloud services including **Amazon Web Services (AWS)**. Launch various **AWS EC2** instances to deploy the application.
- Performed queuing services for the cloud data processing for certain tasks like file upload, data processing, notice to the On-Premises application using the **AWS service SQS (AWS Queuing System)**.
- Implemented **OAuth 2.0, Basic AUTH**, and **API Key** policies for **API Security**.
- **Gradle** build tool is used to deploy the application on the web server.
- Designed, Developed, and maintained **PostgreSQL** Database architecture.

Environment: Agile Scrum, Jira, Jenkins, Docker, AWS, Git, Swagger, Grunt, Splunk, MongoDB, Express.js, HTML5, CSS3, Bootstrap, Kafka, Zookeeper, Node.js, Angular (14, CLI 13), TypeScript (4.5.2), RxJS (7.4.0), Java 8, Hibernate ORM (5.5.6), Spring Cloud Netflix (Eureka, Ribbon, Hystrix), Zuul, Spring Boot, Spring Security, OAuth 2.0, Basic Auth, API Key, PostgreSQL, SQL, PL/SQL, Mocha, Chai, React JS, Bootstrap.js, Require.js, D3.js, NPM, Bower, Gradle, Postman, ELK Stack.

Client: Nokia of America Corporation.

Aug 2021 – Nov 2022

Location: Dallas, Texas

Role: Full Stack Java Developer

Project: Developed a cloud-based platform for Nokia of America Corporation, aimed at streamlining and centralizing the management of Nokia's extensive range of products and services. This unified platform enables efficient integration and seamless handling of diverse functionalities, enhancing operational agility and customer satisfaction. The architecture emphasizes modularity and scalability, ensuring adaptability to evolving business needs.

Responsibilities:

- Developed a **Micro Service Oriented** application with **React JS (v16.14)** in the front end and **Spring Boot services**.
- Involved in development, design, and implementation of the front-end part of the widget-based application using **HTML5, CSS3-LESS, AJAX, Bootstrap 4.0**.
- Configured **React-Redux** for state management and **React-Router (v5.2)** for programmatic navigation.
- Implemented **Store-Redux** for holding the whole state tree of application and **Reducer-Redux** for describing actions.
- Used **Axios AJAX** request to data from back-end based on user interface input and post updated data back to backend.
- Used **PostgreSQL** open-source relational database for storing application data. Performed advanced operations like indexing, cursors, triggers, stored procedures for faster data retrieval.
- Implemented **Hibernate ORM** for managing object-relational mapping between Java objects and **PostgreSQL** database. Used **Hibernate Query Language (HQL)** for simplified database communication.
- Developed robust server-side application using **Spring framework** components like **Spring MVC** for REST API development, **Spring Security** for authentication and authorization, **Spring JPA** for persistence layer, **Spring Batch** for transactional data processing, **Spring AOP** for managing cross-cutting concerns.
- Leveraged declarative **Spring Transactions** for effective handling of ACID transactions and used **Spring Beans** to encapsulate business logic implementation.
- Used **Spring Batch** with **Quartz Scheduler** for developing job processing system to run batch processes on scheduled time intervals.
- Developed asynchronous messaging system using **Java Message Service (JMS)** API. Used **RabbitMQ** message broker for decoupled communication between microservices. Interacted with server using **Java Naming and Directory Interface (JNDI)**.
- Configured **Spring Cloud Netflix** components like **Eureka** for service registration and discovery, **Ribbon** for client-side load balancing to build fault-tolerant microservices architecture.
- Containerized microservices and applications using **Docker** containers and **Kubernetes** for automated container deployment, scaling, and management.
- Implemented scalable and lightweight API backend with **Node.js** runtime and **Express.js** framework. Built robust RESTful APIs and connected them with frontend.
- Used **React (v16.14)** library with **Redux** state container for building modular front-end components and managing application state across components in a centralized store.
- Performed API testing using **Postman** tool. Implemented unit testing by creating test suites with **Jest** and **Enzyme**.
- Created **CI/CD** pipelines using **Jenkins** on **AWS EC2** instances for automating build, test, and deployment processes.
- Provisioned and configured various managed **AWS services** like **EC2** for compute, **RDS** for relational databases, **S3** for storage, **CloudWatch** for monitoring and logging.
- Developed serverless applications using **AWS Lambda** functions and **DynamoDB** NoSQL database.
- Implemented centralized logging and real-time application monitoring using **ELK stack - Elasticsearch, Logstash, Kibana**.
- Containerized microservices and applications using **Docker** for development and **Kubernetes** for container orchestration to achieve portability and scale.
- Followed Agile practices including daily standup meetings, sprint retrospectives, design reviews and solution discussions. Used **JIRA** for user story and defect tracking and **Bitbucket** for source code version control.

Environment: React, Redux, Spring Boot, Spring MVC, Spring Security, Spring Data JPA, Spring Batch, PostgreSQL, Hibernate ORM, Java Message Service (JMS), RabbitMQ, Spring Cloud Netflix (Eureka, Ribbon), Docker, Kubernetes, Node.js, Express.js, Jest, Enzyme, Jenkins, AWS (EC2, RDS, S3, DynamoDB, Lambda, CloudWatch), ELK Stack (Elasticsearch, Logstash, Kibana), Bitbucket, JIRA.

Client: CREXi

Oct 2019- July 2021

Location: Marina del Rey, CA

Role: Java/J2EEE Developer

Project: Revitalized the CREXi platform, refining user experiences through an interactive UI and optimized backend operations. Implemented RESTful services, streamlined messaging, and ensured robust testing. Orchestrated containerized services and enabled precise cloud deployment, bolstered by vigilant monitoring. Collaborated cross-functionally and upheld stringent security measures for seamless, high-performance operations.

Responsibilities:

- Developed and enhanced various modules and features of the CREXi platform following agile development principles. Worked closely with product owners and engineering team to deliver user stories and epics.
- Created an interactive and responsive UI using **AngularJS, HTML5, CSS3, JavaScript** and **Bootstrap framework**. Developed reusable UI components and ensured consistency across the application. Optimized page load time through lazy loading.
- Implemented core business logic and CRUD operations using **Spring MVC, Spring Boot, Spring Data JPA and Hibernate**. Developed data models, repositories, services, and controllers. Used Spring declarative transaction management.
- Developed RESTful web services using **Spring Boot** and **JAX-RS** to expose backend APIs. Created API documentation using Swagger. Added validation, security, caching, and versioning.
- Integrated **Kafka and ActiveMQ** for asynchronous messaging between services. Configured topics, producers, and consumers. Handled serialization and deserialization of messages.
- Used **JUnit, Mockito** and **Power Mock** for unit testing the REST APIs and core components. Achieved over 80% code coverage for critical components.
- Containerized services using **Docker** and managed them with **Kubernetes**. Created Docker files, optimized image sizes and configured Kubernetes deployments.
- Leveraged **AWS EC2, S3, RDS, CloudWatch** and other services to build and deploy the application on cloud. Automated provisioning using CloudFormation and Ansible.
- Implemented monitoring, logging and analytics using **New Relic, ELK stack**. Enabled tracking of key business metrics and system health.
- Developed automation test framework using **Selenium and Cucumber**. Executed regression testing on every build.
- Used **Git, Jenkins and Bamboo** for version control, continuous integration, and deployment. Setup branch policies and CI/CD pipelines.
- Collaborated with cross-functional teams including product, DevOps and QA following agile methodology. Participated in daily standups, plannings, retrospectives and code reviews.
- Configured application security and access control using **Spring Security** and **OAuth2**.
- Developed reproducible builds across environments using **Maven** and followed semantic versioning.
- Implemented **Redux** for predictable state management on the frontend.
- Leveraged **Postgres** and **MongoDB** based on data access patterns. Designed normalized schemas optimizing for performance.

Environment: Java 8, J2EE, Spring Boot, Spring MVC, Spring Data, JPA, Hibernate, AngularJS, Kafka, ActiveMQ, REST, SOAP, JUnit, Mockito, Selenium, Cucumber, Docker, Kubernetes, AWS, Git, Jenkins, Spring Security, OAuth2, Maven, Redux, Postgres, MongoDB

Client: Aam Health
Location: Gurugram, India
Role: J2EE Developer

Apr 2017- Sep 2019

Project: Aam Health is a healthcare startup focused on providing affordable and quality healthcare services. This project involved developing a web platform and mobile apps to enable patients to consult doctors online, book appointments, order medicines and access health records.

Responsibilities:

- Developed various features and enhancements for the web platform and Android/iOS apps using **Java, J2EE, Spring MVC, Hibernate, RESTful web services**. Worked on implementing appointment booking, online consultation, medicine ordering and health records modules.
- Built responsive and interactive UI for the web platform using **HTML5, CSS3, JavaScript, jQuery, Bootstrap**. Focused on performance, accessibility, and responsive design for supporting multiple devices.
- Designed and implemented the persistence layer and data access components using **Hibernate, JPA, Spring Data**. Defined entities, repositories, and custom **ORM** mappings.
- Developed scalable and resilient REST APIs and microservices architecture using **Spring Boot**. Documented APIs using **Swagger** to make it easy for consumers.
- Integrated asynchronous **Kafka and RabbitMQ** message brokers for building event-driven architecture. Configured topics, queues, producers, and consumers.
- Containerized microservices using **Docker** and orchestrated containers using **Kubernetes and OpenShift** for high availability and autoscaling. Optimized **Docker** images for size and security.
- Leveraged managed **AWS EC2, S3, RDS, CloudWatch** and other services for hosting the platform on cloud. Automated infrastructure provisioning using CloudFormation.
- Implemented advanced application monitoring, logging and analytics using **ELK stack, Grafana**. Tracked KPIs like user signups, retention etc.
- Performed comprehensive unit and integration testing using **JUnit, Mockito and Selenium**. Achieved over 85% code coverage for critical components.
- Used **Git, Jenkins and Maven** for version control, continuous integration, and delivery with zero downtime deployments.
- Collaborated extensively with designers, QA engineers and DevOps team in an agile setup. Participated in daily standups, plannings and retrospectives.
- Developed **ETL** pipelines for analytics using **Spring Batch**. Handled large **CSV** datasets for reporting.
- Implemented caching using **Redis** and **Hazelcast** data grids to optimize performance and reduce database loads.
- Secured **REST APIs** using **OAuth2** and handled encryption of sensitive data using **Java Cryptography Architecture**.
- Defined optimal database schemas and queries using **MySQL** and **MongoDB** based on data access patterns and relationships.
- Handled cluster management, deployments and monitoring using **Apache Ambari**. Automated Hadoop stack management.

Environment: Core Java, J2EE, Spring Boot, Spring MVC, REST, Microservices, HTML5, CSS3, JavaScript, Kafka, Docker, Kubernetes, AWS, JUnit, Selenium, Git, Jenkins, MySQL, MongoDB, Hazelcast, Redis, Ambari

Client: Infrassist Technologies Pvt. Ltd.,
Location: Ahmedabad, India
Role: Java Developer

May 2015 - Mar 2016

Project: Developed an enterprise web application for managing employee payroll and leave management. The application digitized the manual processes and provided self-service portal for employees.

Responsibilities:

- Designed responsive user interfaces using **HTML5, CSS3, JavaScript, Bootstrap4** framework for consistent user experience across devices. Developed mobile-first layouts using media queries.
- Developed **RESTful APIs** using **Spring Boot** framework for seamless integration with front-end. Used **Spring Data JPA** for interacting with **MySQL** database. Wrote SQL queries, stored procedures using **JDBC** for complex database operations.
- Developed back-end request handling using **Core Java multithreading** and exception handling. Optimized application performance by implementing caching, multi-threading, and pooling.
- Implemented **Model-View-Controller architecture** using **Spring MVC** and **Struts** frameworks. Developed Action classes and mapped requests to appropriate controllers.
- Developed persistence layer using **Hibernate ORM**. Performed object-relational mappings to map domain model objects to database tables. Used **Hibernate API** for database interactions.
- Developed and exposed **SOAP and REST web services** using **JAX-WS** and **JAX-RS** specifications for integration with external systems. Created **WSDL** file and wrote automated test cases using **JUnit**.
- Batch processed large volume of data using **Spring Batch** framework. Used **Quartz Scheduler** to create cron jobs and schedule tasks.
- Enabled asynchronous communication using **JMS API** and **RabbitMQ** message broker. Configured topics, queues, and listeners to publish and consume messages.
- **Containerized** application using **Docker** to enable portability across environments. Leveraged **Kubernetes** for container orchestration and deployment automation.
- Used **Jenkins CI/CD** pipelines to implement continuous integration and deployment. Performed automated UI testing using **Selenium** and load testing using **JMeter**.
- Used **GIT** for version control and **Maven** and **ANT** build tools for build automation and dependency management.
- Followed **Agile** methodology using **Scrum** framework. Used **JIRA** and **Confluence** tools for project management.

Environment: J2EE, Spring Boot, Spring Data, JPA, Hibernate, REST, SOAP, JavaScript, Bootstrap, JUnit, Selenium, Docker, Kubernetes, Jenkins, Maven, Jira