Name: Hruday Kumar Email: hrudayb333@gmail.com

Java full stack developer Phone:4706326833

**Professional Summary:**

* Having Tangible engagement as a Java full stack developer, I specialize in front-end UI development using various JavaScript frameworks and libraries. I possess a strong understanding of the Software Development Life Cycle (SDLC), including requirements analysis, design, development, testing, and documentation.
* Proficient in using Client-side Development Frameworks like **Angular, React.JS, ECMAScript 6, and Typescript** to develop responsive Single Page Applications (SPA) and Dynamic Web Applications using MVC Architecture.
* Used **React JS** to build dynamic, interactive, and responsive user interfaces to create complex and interactive UIs with ease.
* Used **Angular** to connect client and server end of an application, making it easier to develop, test, and deploy code.
* Good knowledge of core Java with a strong understanding of **Multithreading, Collections**, **Exception handling,** and **Garbage Collector**.
* Developed core modules in large cross-platform applications using **Java, J2EE, Spring, JSF, Hibernate, JSP, Servlets, EJB, JDBC, Web Services (SOAP, REST), E-Commerce, Microservices**.
* To improve flexibility, reliability, and agility of the application I used **Microservices** to break down a monolithic application into smaller, independent services that can be developed, deployed and scaled independently.
* Used **Spring Boot** to manage and update my project’s dependencies and I used Spring Boot to build the production ready applications such as logging.
* Used **Spring Security** to provide authentication, authorization like security features in order to allow users to log in securely and also have access to different resources in a web application.
* Used **Postman** to make HTTP request to test and debug their APIs. It is easy with this to create and save HTTP requests, to view server responses, and test their APIs for different situations and edge cases.
* Used **Swagger** for documenting RESTFUL APIs, because of the way it documents API’s including the endpoints, testers, and stakeholders to understand the APIs functionality and usage.
* Proficient in **RDBMS** concepts, queries, **PL-SQL**, stored procedures, and functions using **Oracle.**
* Written and executed automated tests to ensure the quality of the software that is developed using **Testing, Jasmine, Protractor,** and **Karma.**
* Extensively Used Jenkins as a Continuous Integration tool to deploy Spring Boot Microservices to Pivotal Cloud Foundry (PCF) using build pack.
* Extensive experience with databases such as **Oracle** and **MySQL, MongoDB.**
* Experienced in deploying J2EE applications on Apache Tomcat web server and **WebLogic**, **WebSphere**, and **JBoss application server.**
* Strong Experience in **RDBMS** using **PostgreSQL, PL/SQL** to write Stored Procedures, Functions, **Triggers**, and Prepared Statements.
* Used **Maven** and **Jenkins** as build tools and deployed the application using **Kafka.**
* Used Spring config server for centralized configuration and **Splunk** for centralized logging.
* Expertise in developing **SQL Queries**, **Stored Procedures, and Triggers in databases viz. Oracle, SQL Server, My SQL,** and **DB2.**
* Focused on architecting NoSQL databases like **MongoDB, Cassandra,** and Cache database.
* Developed and maintained code for various applications using **PHP, Java, and MySQL**.
* Experience in **Amazon Web Services** (Amazon EC2, Amazon S3, Amazon Simple DB, Amazon RDS, Amazon Elastic Load Balancing, Amazon SQS, AWS Identity and access).
* Developed unit and integration tests for existing microservices using **JUnit**, **Mockito,** and **Docker.**

**Technical Summary:**

|  |  |
| --- | --- |
| Languages  | Java, SQL, and PL/SQL. |
| Java Script Technology  | Angular, React JS, Node.js. |
| App/Web servers  | Apache Tomcat, WebLogic, Apache camel, WebSphere, JBoss |
| Messaging Tools | Rabbit MQ, Apache Kafka, Apache Spark, Scala |
| Applications Frameworks  | Spring, Hibernate, Struts, JSF. |
| Build Tools | Ant, Maven, Gradle |
| Databases | Oracle, MySQL, SQL Server, PostgreSQL, DB2, MongoDB, Cassandra. |
| Tools  | ANT, JUNIT, Log4j, TOAD, Maven, Rational Rose. |
| Testing | JUnit, Mockito, SOAP UI, Selenium, Cucumber, Log4J |

**Work Experience:**

**Client: Verizon, FL.**

 **Date: April 2022 - Current**

**Java Full Stack Developer**

**Responsibilities:**

* Implemented **JAVA/J2EE** design patterns such as **Factory, DAO, Session Facade, and Singleton**.
* Used **React JS** to improve the overall performance of the application because it provides a fast and smooth user experience by allowing the creation of reusable UI components, reducing the need for full page reloads.
* Used **React JS** to connect the front- end and back -end of an application making it easier to develop, test, and deploy code.
* Used **React JS** to access a wealth of resources, libraries, and tools so that we could use the tools for development, testing, and debugging.
* Used **React JS** in conjunction with other front end and back-end technologies, including java frameworks such as spring and hibernate.
* Utilized several Java 8 concepts like **Stream API, Time API, Collection API,** and **lambda expressions to migrate the existing application.**
* Streamlined the server Implementation by moving to **JSON** and **MongoDB** as the backend data model.
* Created various **MongoDB** collections and wrote services to store and retrieve user data from the application.
* Implemented **REST API** as **Spring Boot Microservice** to perform modular functionality as per **Microservices** architecture.
* Used **JWT (**JSON Web Token) to implement single sign-on (SSO) across multiple applications. By sharing a common JWT token between the different applications, users can log in once and access all the authorized resources across the different applications without needing to log in again.
* Introduced **Hibernate** in this Project for the first time in place of **EJB** and made a successful implementation of **POJO (Plain Old Java Objects)**.
* Used **Hibernate** to query the **DB2 database** and return results to the **User Interface.**
* Developed and executed Unit Test plans using **JUnit**, ensuring that results are documented and reviewed with Quality Assurance teams responsible for integrated testing.
* Designed and developed **Web Services** to provide services to the various clients using **Restful**.
* Worked on **Swagger** API and auto-generated documentation for all **REST calls**.
* Resources were exchanged using **RESTful** web services with the data passed around in **JSON** format.
* Implemented **Microservices** architecture to make application smaller and independent.
* Worked with **microservices** related frameworks **Ribbon** and **Eureka** for **client-side** load balancer and network discovery.
* Using **REST API’S,** I created endpoints that can be accessed by the front-end components to perform various tasks, such as retrieving data from a database, updating data, or deleting data.
* Using **Spring Boot,** I built the back-end components of web applications because it simplifies the development process, allowing me to focus on the logic rather than infrastructure, and helps me to reduce code I need to write.
* Used **Oracle** database by leveraging JDBC (Java database connectivity) to connect to the database, execute queries and manage data.
* Used **Hibernate** to write database queries and manipulate data so that it could save me time rather than writing SQL statements directly.
* Used Databases to store data so that retrieval and manipulation of data would be easy for me. I used to store sensitive information such as user credentials using encryption.
* Implemented Core Java Concepts such as Multi-Threading, Exception Handling and Collections when needed.
* Involved in writing **SQL** queries, **PL/SQL** programming and creating new packages and procedures and modified and tuned existing procedure and queries using **TOAD**.
* Data modeling and development experience in **Cassandra** or other **NoSQL databases.**
* Implemented data ingestion and handling clusters in real time processing using **Kafka.**
* Developing scripts for build, deployment, maintenance, and related tasks using **Jenkins** and **Maven.**
* Deployed the **Application EAR** and **RESTful Services** on **JBoss** Application Server.
* Used **AWS** services such as S3(Simple storage service), EC2(Elastic Compute Cloud), and EBS (Elastic Beanstalk). Utilized Amazon S3 to store Images, videos, CSS files and JavaScript files used in Web Applications.
* Used EC2 to host the backend services and APIs of the applications. Used Elastic Beanstalk to easily deploy and to manage the java applications.

**Environment**: Java, J2EE, React, Angular, JSON, REST API, JWT, Spring Boot, Mongo-DB, Spring AOP, Hibernate 4, REST, JUnit, Jenkins, Gradle, XML, Selenium, Kubernetes, Groovy, Grunt, PostgreSQL, SQL, TOAD. Cassandra, AWS, Kafka, Maven, JBoss, NoSQL.

**Client: Sumitomo Mitsui banking corporation, Jersey City.**

**Date: Jan 2021 – March 2022**

**Java Full Stack Developer**

**Responsibilities**:

* I have experience in working with **Agile scrum methodology** in developing business applications within an object-oriented environment.
* Used **Angular** for **data binding** which is connecting the data in the component with the template such as one-way data binding, two-way data binding and event binding.
* Used **Angular** to create custom pipes with a set of built in pipes. I also used Angular for **routing** which is **handling navigation**.
* Used Angular to create and manage **Forms** (which are used to collect data from user). Used **Components**, **Modules**, and **Templates** for controlling the portion of user interface, to organize components into logical units and to define the structure and layout of the user interface.
* Used Angular for **Dependency Injection** which is a design pattern used to manage dependencies between different parts of the application so that it would be easy to create and manage services.
* Used **Directives** (built in Angular) to extend the functionality of HTML and to create custom directives.
* Extensively working on Java persistence layer in application migration to Cassandra using Spark to load data to and from Cassandra Cluster.
* Experience in implementing and managing Docker-based infrastructure using Kubernetes.
* Integrated Single sign on using **LDAP** and **Active directory** on Spring security layer, successfully maintained users and groups integrity in the application.
* Used Hibernate to improve the performance of web applications by reducing the number of database queries importantly in applications with complex queries.
* Extensively involved in writing **PL/SQL** code for stored Procedures, Functions, Packages, Libraries, and other database triggers etc. Written interfaces to transfer/interact data with various external systems.
* Design, implement, test, and deploy services in a Microservice-oriented architecture and has Implemented the REST based Microservices using the Spring Boot, Spring Data with JPA.
* Created single page applications with nested and multiple views using **Angular UI-router,** custom directives for reusable components used across the application.
* Used **HTTP** and **resource** service for retrieving data from the server via **XML Http** request object in **Angular,** framework.
* Used **MySQL** in the server-side development of Java full Stack development. I used to write Java code to handle business logic, process data, and interact with the MySQL database executing SQL queries to retrieve data from the database performing data manipulations and updating or inserting data as required.
* Involved in Production Support and Maintenance of the application and troubleshooting of Issues on **JBoss** by checking logs, payments etc.
* Extensively Implemented **Spring boot, dependency Injection (DI)** and **aspect-oriented programming (AOP)** features along with **hibernate.**
* Used **JMS (Java Mailing Service)** API and **Kafka** to mail notifications upon the success or failure when the backend processes are completed and to send mail notifications regarding any system related problems.
* Implemented data ingestion and handling clusters in real time processing using **Kafka.**
* Wrote **Hibernate** Criteria queries for data retrievals and performed the **ORM mappings** to map java entities to database tables.
* Implemented **RESTFUL web services** using Spring Boot and consumed using spring REST services. Published Rest services for performing **CRUD** operations from database.
* Used **AWS** to deploy, scale, and manage web applications and also to run the services run back-end services because it provides me a range of services that allows me to build and manage the infrastructure.
* I used **AWS** to build and deploy web applications and back-end services so that it allows me to handle high traffic and provide reliable services.
* Focused on **Test Driven Development (TDD)** thereby creating detailed **JUnit** tests for every single piece of functionality before writing the actual functionality.
* Used **Log4J** to capture the logging information and **JUnit** to test the application classes.

**Environment**:

MEAN Stack (HTML5, CSS, jQuery, JavaScript, MongoDB, React.js, Node.js, Express JS), Angular, Databinding, Multi-threading, Bootstrap, Spring, Maven, Spring Boot, PCF, Kubernetes, Struts, Selenium, JSP, React JS,, PostgreSQL, REST& SOAP webservices, Spark, Scala, Kafka, log4j, Junit, Docker, Eclipse, GitHub.

**Client: Xansa, Hyderabad.**

**Date: Nov 2018 – Dec 2020**

**Role: Java Developer**

**Responsibilities:**

* Followed Agile software methodology for project development.
* Designed the architecture based on the popular **J2EE Model 2** approach of **MVC** design paradigm.
* Developed application on Spring framework by utilizing its features like **Spring Dependency injection, Spring Security, Spring Web flow with Spring MVC.**
* Created **web services** using **SOAP,** to allow application and services to connect them over the Internet.
* Designed **WSDL's** and elaborated type definitions schemas based on corporate standards, integration, and security specifications.
* Developed unit and integration tests for existing microservices using **JUnit, Mockito** environments for development and QA teams.
* Used **Pivotal Cloud Foundry (PCF)** to role-based access controls, applying and enforcing roles and permissions to ensure that users can only view and affect the spaces for which they have been granted access.
* Implemented **MongoDB** queries for reporting and analyzing production data and used various concepts such as locking, transactions, indexes, sharing, replication, and schema design.
* Involved in implementing various screens for the front end using **React-JS** and used various predefined components from **NPM (Node Package Manager)** and **redux library**.
* Developed code using various patterns like **Singleton, Front Controller and Value Objects, Data Access Object**.
* Designed and implemented scalable, secure cloud architecture based on **Amazon Web Services**.
* Used **Log4J** for logging and tracing messages.
* Hands on knowledge of **Git, Jenkins, AWS Cloud Technologies, RDBMS (PostgreSQL)**
* Developed **JUnit** test cases for **Unit Testing** and **functional testing** for various modules and prepared Code Documentation for future reference and upgrades.
* Used **JIRA** tracking tool for assigning and defect management. Used **Maven** to build and deploy applications onto **JBOSS** Application Server and **LARA** to deploy code onto server.
* Used **GIT HUB** for maintaining the component and to release version management.
* Used **Jenkins** to build the War file.
* Used Confluence page to update the services information and the project related information. Deployed web server components using **Apache Tomcat** and **application server** with **WebLogic.**
* Developed Application using **Spring MVC** Architecture.
* Developed Validations for **Client-side** using **JavaScript.**
* Developed browser compatibility and performance using **AJAX.**
* Developed backend logic using Core Java including **Collections** and **multi-threading.**

**Environment**:

Java, Spring MVC frameworks, Servlets, JSP, MongoDB, Struts, HTML, JavaScript, jQuery, AJAX, Webservices, Angular 4&6, SOAP and REST web services, Jenkins, Micro services, PCF, React Js, Hibernate, Selenium, PostgreSQL, Cassandra Eclipse, Kafka, Apache Tomcat.

**Client: Lumen Technologies, Hyderabad.**

**Date: May 2017 – Oct 2018**

**Role: Software developer**

**Responsibilities:**

* As a java developer I have experience working with **HTML5** in defining the content and structure of web pages allowing us to create meaningful and accessible web pages.
* Used to use **CSS3** for styling the layout and the appearances so that we could create beautiful and interactive user interfaces that are consistent across multiple devices and platforms.
* Used **AJAX** because it enables web applications to communicate with servers in the background so that the user doesn’t have to reload the page.
* Used to create mobile first web applications using **Bootstrap,** and to design and develop the user interface of web applications.
* Used **jQuery** to create dynamic web applications efficiently so that it meets the needs of modern users. I have experience in jQuery plugins and widgets for additional functionality. I have experience in jQuery’s event handling and animation capabilities.
* Used **JavaScript** to manipulate the Document Object Model to add, remove, or modify elements in the web page. I have experience in JavaScript events and event handling to respond to user interactions, such as clicks or keyboard input.
* Implemented **SOAP** to exchange structured data between applications. I used SOAP standards to make sure that applications are compatible with other systems.
* Utilized **Spring MVC** because it provides flexible configuration approach, which allowed me to configure the framework using XML, java annotations, or a combination of both. I also used its testing framework that allowed me to write unit tests for controllers, services, and other components.
* Used **Hibernate** to persist java objects to a database. I mapped java classes to databases tables so that I can store and retrieve data from a database without writing SQL code. I used Hibernate because it supports transactions ensuring database operations are atomic and consistent.
* Used **MySQL** to store and retrieve large amounts of data efficiently. I integrated MySQL with java frameworks such as spring and hibernate to build web applications. I used it to back up and recover data so that we could maintain the integrity of data in the web application.

**Environment:** Core Java, Spring MVC, Hibernate, SQL, HTML, CSS, AJAX, JavaScript, Hibernate, Angular, SOAP, MySQL, jQuery.