**Harini**

**Java Full Stack Developer**

**mounika@sptecinc.com**

**5128663151 EXT:901**

**SUMMARY:**

* Having 8 years of experience in **Java/ J2EE** Professional with proficiency in Analysis, Design**,** Development, Architecture, Coding, Testing, and implementation of Client/Server, Internet, and Legacy Applications.
* Expertise in **Java Technology (J2SE and J2EE)** with a strong foundation of **object-oriented** principles.
* Expertise in creating **web applications** using **HTML**/**HTML5, CSS/CSS3(SASS, LESS), Bootstrap, JavaScript (ES5), jQuery, Angular 9/11, AJAX, and JSON, Node.js, and ReactJS.**
* Hands-on experience in creating **Angular 9/11** factories for using **angular** services like $http and $resource to make **RESTful API** calls to the **Java-based** backend.
* Used Go Programming Language (GoLang) and Scala in the development of the application.
* Worked closely with the DevOps team to set up CI/CD pipeline using **Jenkins, Ansible, AWS, Maven, and GIT**.
* Used **React JS** to create views to hook up models to the DOM and synchronize data with a server as a Single Page Application (**SPA**) and React-**Redux** **thunk** to make **REST API calls**.
* Experienced in Full stack UI technologies using **Grunt.js**, **Gulp Server**, and **Yoman** generator for **Angular** **8/9.**
* Experienced in **Client-side validations** using Ajax in **Node.js**, **Angular**.js, and Dashboards reports using Backbone. JS.
* Worked and learned a great deal from **Amazon Web Services** (**AWS**) Cloud services like **EC2, S3, EBS, RDS, VPC, and IAM.**
* Explored Spark, **Kafka**, and Storm along with other open-source projects to create a real-time analytics framework.
* Deployed the **Cassandra cluster** in a cloud (amazon **AWS**) environment with scalable nodes as per the business requirements
* Deployed **Spring Boot-based** microservices **Docker** container using **Amazon EC2** container services.
* Experience with container-based deployments using **Docker**, working with **Docker images, Docker Hub**, **Docker registries,** and **Kubernetes.**
* Experience in publishing and subscribing **streams** of records in enterprise message systems and in designing and building real-time streaming **data pipelines** using **Apache Kafka**.
* Involved in UI Designing, Development, and Testing of Web Forms **using ASP.Net, C#, XML, ADO.Net,** and **.NET Framework.**
* Wrote **Kafka** producers to stream the data from external rest APIs to **Kafka** topics.
* Expertise in designing, developing, and deploying applications using **J2EE technologies** including Servlets, **JSP, EJB, JSTL, Struts, Spring, JMS, AJAX, JDBC, JNDI,** **Java** Beans **Hibernate,** and **Web Services**.
* Developed and Implemented Web Services using Quarkus.
* Good working knowledge on reactive programming and the spring webflux module
* Worked with **Jenkins** in setting up **CI/CD** pipeline for the dev environment and was involved in setting up GIT webhooks.
* Hands on experience using and writing scripts for**Autosys scheduling tool.**
* Scheduling Batch Jobs and Autosys Scheduling.
* Involved in setting up Amazon EC2 instances, virtual private cloud (VPCs), and security groups. Setting up databases in AWS using RDS, storage using S3 bucket and configuring instance backups to S3 bucket.
* Experience in working with other **iOS and Android** developers to complete detailed projects.
* Experience in all phases of usability testing including user profiles and personals, task analysis, contextual research, test moderation, and reporting using**Jasmine.**
* Expertise in creating Workflow based on Functional Specification documents. Good knowledge of OTLP system(s) and involved in creating facts, dimensions, and Star schema representation for the Datamart.

**TECHNICAL SKILLS:**

|  |  |
| --- | --- |
| **Languages** | Java, Java Script, JSP, UML,Go lang, WSDL, HTML, CSS, jQuery, PL/SQL, Perl,  Python and Shell Scripting. |
| **Java Technologies** | Java, J2EE, JDBC, Hibernate, JSF, XML Parsers, JSP, Servlets, Portals,  EJB, JMS, Java Swing, Struts, Spring Framework, Java Beans, Web  Services. |
| **Web Technologies** | JavaScript, HTML, CSS, JavaScript, XML, XSL,ECMA script, XSLT, SOAP, AngularJS,  Angular 9/10, Node JS, TypeScript, JSON, Restful web services (API), Junit. |
| **Database** | Oracle, SQL Server, My SQL, PL/SQL, PostgreSQL, NoSQL, Cassandra, Mongo DB. |
| **Tools** | Tortoise SVN, TOAD, SQL Developer, Sub Version, SOAP UI, GitHub,  Splunk, Jfrog, UDeploy, Fortify Scan, and Black duck Scan |
| **Lifecycle Methodologies** | OOAD, Waterfall Model, Agile, Test-Driven Development (TDD) |
| **Defect Tracking Tools** | JIRA, Bugzilla, Intercept, Jasper. |
| **Frameworks** | Spring, Spring MVC, Spring Boot, Spring Batch, Spring JDBC, Spring  Security, Hibernate, Struts, EJB |
| **IDE** | Eclipse, WebStorm, IntelliJ, NetBeans, Visual Studio. |
| **Build Tools** | Maven, Jenkins, Docker, Apache Ant. |
| **Web/App Servers** | Apache Tomcat, JBoss, Web logic, WebSphere, Glassfish. |
| **Cloud Services** | Amazon Web Services (AWS) (Cloud Formation, Cloud Watch, Code Build, Code Commit, Code Deploy, Code Pipeline, EC2, EC2 Container Service, EBS, Elastic Beanstalk, EKS cluster, IAM, Security Groups, Route 53, S3, CloudFront, Glacier, SQS, SNS, VPCs, DynamoDB, Elastic Cache, Lambda). |

**EDUCATION DETAILS:**

* Master’s in computer science from the University of Central Missouri- 2018
* Bachelor of Technology in Electronics and Communication Engineering from MGIT - 2014

**PROFESSIONAL EXPERIENCE:**

**Client:** **World Fuel Services, NC Aug 2020 - TILL DATE**

**Role: Sr. Java Full Stack Developer**

**Responsibilities:**

* Involved in various phases of the **Software Development Life Cycle (SDLC)** as requirement gathering, modeling, analysis, architecture design & development, and worked with the business team to review the requirements.
* Developed static pages using **HTML5, CSS3, and JSP** with **Ajax** calls in **jQuery** to retrieve data as **JSON** Object from back-end **RESTful** Server, then display the well-organized result in web pages by writing **Angular 9/10** controller, directives, services, and route providers. Also, used **Angular 9/10** filter to make the data searchable and sortable.
* Created Directives, Services, Filters using Angular JS with Typescript.
* Strong Understanding of **Docker** and **Kubernetes** to deploy applications quickly and set up the CICD pipeline.
* Implemented **Kafka** producer and consumer applications with help of the zookeeper and retrieved log messages using **Kafka**.
* Developed responsive web product features using React JS, SASS, on GoLang with REST.
* Wrote a prototype on GoLang to improve performance of report generator.
* Created a single-page application with loading multiple views using route services and added more user experiences to make it more dynamic by using the **AngularJS** framework.
* Developed complex **Multithreaded Java** applications for processing large volumes of data to achieve Multitasking.
* Configured and tuned the **AWS** EC2 and worked on ELB, and Route53 for high availability.
* Created **TypeScript** reusable components and services to consume REST APIs using component-based architecture provided by **Angular**.
* Developed Single Page applications using AngularJS using components, directives using TypeScript.
* Used **Spring Kafka** API calls to process the messages smoothly on **Kafka** Cluster setup.
* Analysed the **Node.JS** server structure in the legacy project, then mimicked the **REST service**by using **java** **JAX-WS API,**and did the corresponding configurations.
* Used **Java 1.8** **Lambda** expressions along with **Streams** for creating internal iteration and performing chain operations such as **Filter, Map, and Collect on a collection**.
* Worked on developing **APIs** using **Kubernetes** to manage and specify the copies of the containers to run the actual servers in the cloud environment.
* Worked with **Node.JS** framework including authoring **complex directives, services, controllers, and filters**.
* Generalized JPA Toplink mapping template for the application so that top link related code could be separated from the actual DAO implementation. Combined imperative code and non-blocking reactive style using Quarkus.
* Implemented a metrics collection mechanism for the server by utilizing **Apache Kafka**to stream server-side events.
* Worked with **MongoDB** to store non-relational data in collection and retrieve them whenever required.
* Involved in writing **SQL** and **Stored Procedures** for handling complex queries with help of **TOAD** and accessing them through **Java** Programs from **Oracle Database**.
* Used **REST API** calls from **Angular** **8/9** and parsed the data to the client and handled the security and REST services authorization using **Spring** Security.
* Developed **REST** APIs using **Spring MVC** and **Spring boot**, hosted all **microservices** on **AWS,** and used **Elastic Beanstalk** to set up the application and configured environments.
* Designed and developed **Micro Services Architecture** to divide the application into business components using **Spring Boot**.
* Experienced in working on Spring Modules (DAO, MVC, Securities, AOP, ORM data), Quarkus, Micronaut, JWT.
* Used Quarkus for building lightweight, container-based microservices and serverless applications.
* Worked with **JUnit** Regression Test Framework to implement unit testing, accelerate programming speed, and increase the quality of code.
* Involved in creating resources stack on **AWS EC2** using **Cloud Formation** by writing **Template** files.
* **AWS Lambda**was utilized for serverless implementations, where the functions were triggered when the database tables were updated and AWS **Cloud Watch**was utilized for alarms.
* Developed API for using AWS **Lambda**to manage the servers and run the code in the DB.
* Used the **WebLogic Server** Administration Console to control production-specific deployment details.
* Worked on integrating **Spring** with the **Elastic search** and developed APIs to read and publish data from the **Elastic search** cluster using a native **Java** transport client as well as a **REST** client.
* Worked on developing **Restful** endpoints to cache application-specific data in in-memory data clusters like **REDIS** and exposed them with **Restful** endpoints.
* Used datatypes like **Mono** and **Flux** from Spring WebFlux to create a **non-blocking asynchronous reactive** API.
* Experience in writing unit tests on both client and JS server side using **Jasmine framework.**
* Worked in an **Agile TDD** environment using SDLC tools like **SVN, Jira, Jenkins**, and **Maven** for build packaging and deployments of the application components.
* Used **Spring Tool Suite (STS)** as an IDE for Application Development

**Environment**: Agile methodology, Java, HTML5, CSS3, JavaScript, AngularJS, Angular 9/10, STS, J2EE, Hibernate, Spring framework modules, JSP, Oracle, Cassandra, MVC, Jira, Jenkins, JUnit, Maven, JAX-RS,ECMA script, Log4J, Apache Tomcat, Spring Boot, Spring webflux, Web Socket Microservices, AWS, Elastic Bean stack, Quarkus, Spring Batch, Docker, WebLogic Server, Elastic Search, STS, Tortoise SVN.

**Client:** **John Deere, Moline-IL Jan 2019 – July 2020**

**Role: Java Full Stack Developer**

**Responsibilities:**

* Responsible for system analysis, design, and development using **Java** J2EE architecture.
* Developed the application use cases using the open-source framework such as **Spring** and **Hibernate**.
* Used multi-tiered **Java** design and coding utilizing **Spring** **Inversion of Control** (IOC), **spring** **boot,** and **spring** **security**.
* Setting up and configuring **Kafka** Environment in Linux from the scratch and monitoring it.
* Worked in using **React.js** Router, Forms, Animations, Keys, components, Events, and Flux concepts.
* Responsible for React UI and architecture. Including Tree, Building components library, Table Grid, and Slide-View.
* Developed, configured, and maintained a new **GraphQL** service using **Spring Boot**, **Spring WebFlux**, **Reactor**, that was responsible for communicating with the SOAP/REST endpoints in our existing microservices to fetch the required data
* Set up **CI/CD pipeline** process for the application using **Jenkins**.
* Experience in working with messaging systems like **Kafka,** and **Kafka** Event Sourcing.
* Experience in **Docker** **Engine** and **Docker** **Machine environment**, to deploy the **micro services-oriented environments** for a scalable application.
* Created multiple **Kubernetes** Clusters in **AWS** and deployed a **Docker** Image on top of the **Kubernetes** Cluster for testing purposes.
* Designed and developed the login portal using a framework along with **HTML5, CSS3,** and **TypeScript.**
* Implemented stable React components and stand-alone functions to be added to any future pages.
* Developed a dashboard based on Ajax calls with Business Customers Count, Flags, and Real-time graphs for Analytical Reporting using **Node.js**.
* Experience in Message Oriented Middleware implementation using **JMS**and **Apache Kafka.**
* Used Express.js which is an assertive, fast, essential, and moderate web framework of **Node.js** that helps manage a server and routes.
* Developed readable and writable streams of **Node JS** to process the data from external means of the application.
* Developed REST based Microservices using Quarkus and Spring Boot. using Azure Devops Pipeline to support weekly releases and out-of-cycle releases based on business needs.
* Working as a Java full time developer in developing the application using REST API Services, Implementing security, integration with quarkus.
* Used a microservice architecture, designed and developed RESTful web services using Apache CXF and **Spring** Boot framework.
* Experienced in working on Spring Modules (DAO, MVC, Securities, AOP, ORM data), Quarkus, Micronaut, JWT.
* Changing the **AWS** infrastructure Elastic Beanstalk to **Docker** with **Kubernetes**.
* Used **React.js** library functions for the logical implementation part on the client side for all the applications.
* Involved in the process of **Cassandra** data modeling for efficient querying.
* Worked in a CI/CD environment with Jenkins, Maven, Ansible, AWS, and Terraform.
* Moving existing monolithic architecture to microservices and migrating them to the **AWS** cloud.
* Used the **AWS** Services platform to upload data into **AWS,** S3 buckets and create EC2 instances.
* Used **spring** core annotations for dependency injection and utilized **spring** boot for developing microservices and **spring** batch for batch processing.
* Worked in using **React JS** components, Forms, Events, Keys, Routers, Animations, and Flux concepts.
* Implemented Web Services (WSDLs) using JAX - WS and also implemented REST services by Golang with microservices architecture.
* Implemented and configured the business objects of an application by using **spring** beans.
* Written mapping files for mapping **java** objects with columns in database tables.
* Developed Restful Web Services using **JAX-RS** which were exposed to other applications.
* Write unit test cases by using **JUnit** and **Mockito** framework to perform the unit testing.
* Used Git as a source version control tool and used Log4j for log.

**Environment:** Java, J2EE, Angular, Restful Web Services, AWS, ECMA script ,Oracle, GIT, Hibernate, Spring Boot, Spring-MVC, Spring-REST, Soap UI, PL-SQL, JSON, Cassandra, Quarkus, jQuery, JavaScript, Maven, JBoss, Tomcat, Windows.

**Client: Farmers Insurance, Houston (TX). Dec 2017 – Dec 2018**

**Role: Java /J2EE Developer**

**Responsibilities:**

* Used **Agile** methodology throughout the software development lifecycle (SDLC).
* Used **Java 8** features like **lambda expressions** for communicating between the business layer and database, and **stream API** for Bulk Data Operations on **Collections** using both streams and parallel streams.
* Defined and developed the application’s presentation layer in UI development with technologies like **HTML 5**, **CSS3**, **JavaScript,** and **Angular 8.**
* **Apache Kafka** is used to sending message objects to client queues and topics.
* Defined **Angular** routes using the Router module for components at the module level to enable lazy loading for modules and components.
* Utilized **Kubernetes** and **Docker** for the runtime environment of the CI/CD system to build, test, deploy.
* Created internal diagnostic tools using Golang and AngularJS in order to assist with customer issues. Golang and Java were used as a backend REST service while AngularJS was used for the user interface.
* Worked on writing chain code using GOlang.
* Developed data transformer for converting legacy data from Source tables to avro format and fed into **Kafka**using **Camel.**
* Used **Node.js** for non-blocking design operation for server-side code.
* Working on **Micro Services** with **Spring Cloud** using **Netflix Eureka**, **REST,** and **Netflix Feign Client**.
* Implemented **REST-based** web services using **JAX-RS** annotations, Jersey provider
* Used **jQuery/JQLite** to create dropdowns, status, progress and menu bar, and other drag-and-drop components.
* Apache Mesos and **Kubernetes** were used as the best ecosystem to deploy and manage **Micro-services**.
* Used **Node JS** for server-side rendering. Implemented modules into **Node JS** to integrate with designs and requirements.
* Experience in creating connections for HTTP requests using Websockets.
* Used Node.js to create powerful web-based dashboards that checks the services’ statuses in an asynchronous manner and pushes data to clients using websockets.
* Experience in Building **Web Service** Contracts in **SOA Service Manager, Orchestration, and performance tuning**.
* Implemented **Spring Circuit breaker** pattern, integrated **Hystrix dashboard** to monitor **Spring microservices**, configured **Swagger-UI** to registered Microservices with **Eureka server,** and monitored service health check from **Spring Boot** admin console and **Spring Ribbon** for load balancing.
* Used **Multi-threading** to overcome the errors in the process of transactions, and handle all threads by the main thread process.
* Implemented **Hysterix** a Latency and Fault Tolerance system for Distributed Systems. Wrote Controller, Services, and directive code using **Angular 2/4** for a POC for converting CIM UI.
* Used **Hibernate** extensively to have a Database access mechanism with complex queries through **HQL** and worked on **Hibernate Cache, Criteria, and Transactions**.
* Autosys support: Create scripts to add/delete jobs to/from Autosys using JIL processor
* Autosys and Batch Remediation, Scheduling and Support.
* Develop source code for processing of Confidential files through backend jobs scheduled by autosys.
* Worked on **Camel-based integration** middle-ware solution for Provisioning Services by design and the Implementation of business logic and data processing routes using **Apache Camel**.
* Experience in end-to-end development and migrating the application to the Cloud using the **Amazon web services** (**AWS**) availing services of **AWS** like **EC2,** **Elastic Beanstalk, Elastic Search, Cloud Watch, Cloud Front, Cloud Formation, IAM, S3, RDS, Elastic Cache, SNS, SQS, AMI, and Dynamo DB**
* Worked in a CI/CD environment with Jenkins, Maven, AWS, Ansible, and Terraform.
* Created the **Docker** containers and **Docker** consoles for managing the application life cycle.
* Hands-on experience with **MongoDB** Installation, Configuring, Monitoring, and Designing Sharded Environment.
* Developed test cases and performed unit testing using **Junit** and **Karma.**

**Environment:** Web Services, Spring, Hibernate, ECMA script ,HTML5, JavaScript, jQuery, AJAX, GWT, CSS3, Angular JS, SOAP UI, XML, WSDL, Java, J2EE, Tomcat, Scala, Kafka, AWS, MongoDB, Agile Methodology, Autosys, Spring Boot, Spring Cloud, Spring Ribbon, Netflix Eureka, JIRA processes, GIT, Design Patterns, PL/SQL, JUnit, Maven, Eclipse IDE, CVS, and Oracle.

**Client: DSQ Software, Hyderabad, India  July 2014 – Dec 2016**

**Role: Java Developer**

**Responsibilities:**

* Working as a **full-stack senior Java developer** for this project. Implemented change requests from start to end touching all the layers.
* Used **Struts framework** for the **MVC** implementation with **spring for DI** and **Hibernate as ORM tool** for **database** communication.
* Designed **UI screen** for Insurance Quote0
* to get User information with **JSP, Struts tag library.**
* Implemented the **business logic** in **EJB 2.0.**
* Developed **Action Classes, and Service Classes** for the Resident Insurance premium payment module.
* Involved in Coding **JavaScript code** for UI validation and worked on **Struts validation frameworks.**
* Involved in the development of both the **back end** and the **front end** of the application using Struts, **Hibernate,** and Web Services**.**
* Developed **JavaScript** and **JS frameworks** like **J-Query**.
* Develop integration techniques using the Apache camel and Apache **Active MQ** technologies and experience using Fuse/Apache **Kafka** as an integration platform and Apache **Active MQ** as messaging platform.
* Used **Java** to design and develop this application, **DOM** parser to parse an **XML** document
* Developed **Session** and **Entity beans** for insurance Quotes.
* Developed **web services** using the Apache Axis web service engine. Used **SOAP** request-response exchange pattern to exchange business **XMLs**.
* Responsible for Writing **JSP** for Online Quote for Resident Insurance.
* Used **Web Services** for interacting with a remote client to access data.
* Developed **CMP** read-only Entity beans to get Insurance Quotes.
* Strong Understanding of **Docker** and **Kubernetes to** deploy applications quickly and set up the CICD pipeline.
* Built enterprise wide DNS historical data trending analysis tools in Go (Golang), Languages with REST API for user queries as microservices.
* Wrote **J-Unit Test cases** for Unit testing in Pension Manager Module, performed automated acceptance tests written in **TDD (Test Driven Development)** using **cucumber.**
* Documented **CR impact** analysis and prepared implementation documents.

**Environment:** Java, J2EE, EJB, JSP, Servlets, Struts, XML, SVN, ANT, J-Unit, HTML, CSS, Eclipse, Apache AXIS, Web-logic, SOAP, WSDL, Oracle.