**Name**: **Ramya**

**Email**: archerc@softworldusa.com

**Contact**: **(913) 945-1166 | ext. 1004**

**PROFESSIONAL SUMMARY:**

* Over **8 years of** professionalexperienceas a **Full Stack Developer,** experiencedin **implementing** and **developing complex** applications**.** Involvedin **Software Development Life Cycle (SDLC)** whichembraces **Design, Development, Implementation** and **Testing** of **enterprise business applications** using **Java.**
* I have gained extensive experience in configuring the application context of a **Spring framework application**, utilizing **dependency injection, Spring Boot,** and **Spring JPA**.
* I possess proficiency in implementing various **Spring frameworks**, including **Spring MVC, Spring IOC/DI, Spring REST, Spring AOP, Spring Transaction Management, Spring Data, Spring DAO, Spring Batch,** and **Spring Boot**.
* I have acquired considerable expertise in utilizing **JAVA11** features, including a strong command over **Functional Interfaces, Lambda functions, Stream API, Time API, Transaction Management, Exception Handling, and Collection API.**
* **Experience in developing enterprise level solution using batch processing (using Apache Pig) and streaming framework (using Streaming, apache Kafka & Apache Flink).**
* Have experience developing web applications using **HTML, DHTML, CSS, XML, UML, JavaScript, Ajax, Spring, Hibernate, Struts, and J2EE (JSP, JMS, Servlets, JDBC, Java Beans, EJB)**.
* Having hands-on experience, I am well-versed in utilizing a variety of **ORM frameworks, such as Hibernate ORM Hibernate, Spring Data, JPA, IBatis, MyBatis, Java Persistence API (JPA), Hibernates second-level cache – Ehcache and Mongoose ORM**, for effectively mapping objects to databases.
* With proficiency in utilizing **Spring JTA (Java Transaction API)** for **transaction management** across multiple **databases**, as well as **Spring Batch** for **automated handling** of large-scale, complex data processing, I have demonstrated expertise in these areas.
* **Created reusable elements and attributes in Angular 2/4/8/9 directives**, along with easy to test and maintain client-side interactivity in web applications.
* To develop the rendering view used **React JS components, forms, events, routers, and Redux asynchronous functions and implemented React - Flux pattern for core dependency**.
* **Angular** applications, some of them are **Lodash**, **ngrx (redux architecture for state management)**.
* Created **SPA** project using various **JavaScript** frameworks like **SAAS**, **Angular, Backbone.js, Node JS, Vue JS, TypeScript, Express JS.**
* Created applications and declarative templates with data binding using **Vue JS, Forms, Validations, Modules, Components, Directives, Controllers and HTTP requests**.
* Focusing on databases, I have worked on diverse databases, encompassing **PostgreSQL**, **MySQL**, **DB2**, **Oracle** **13c/11g, Microsoft SQL Server, and NoSQL** databases such as MongoDB, Cassandra, CouchDB, Redis, and **DynamoDB**. Additionally, I have gained considerable experience in writing **stored procedures, functions, and triggers** using **PL/SQL**. These skills have been applied effectively in **JDBC, ODBC, and WebLogic server** environments.
* Used highly durable messaging systems like **Apache Kafka, IBM MQ, RabbitMQ and Active MQ**, for asynchronous exchange of vital information between multiple business applications.
* Worked on a wide range of **AWS services**, including **EC2**, **VPC (NAT, VPC Peering, and VPN), IAM (Identity and Access Management), Elastic Beanstalk, Lambda, S3, CloudFront, Glacier, SQS, SNS, RDS, DynamoDB, Route53, CloudWatch, CloudTrail, CloudFormation, Security Groups,** and more. This extensive experience enables me to effectively leverage the full capabilities of the **AWS** platform.
* Dockerized Spring Boot application to Pivotal Cloud Foundry **PAAS** (Platform as a service) by container-based application deployments using **Docker** with **Kubernetes** and building **continuous integration (CI)** and **continuous deployment (CD)** pipeline using **Jenkins** with **GitHub**.

**PROFESSIONAL EXPERIENCE:**

**Client:  Aetna**

**[Dec 2022-Present]**

**Location: Hartford, CT**

**Role: Sr Full Stack Developer**

**Responsibilities:**

* Made use of **Reactive Forms** using **Angular 12.0** where most of the validation is done using components rather than using template-driven which is provided in HTML.
* Designed Forms, Modules, Views and Templates using Django and Python.
* Used **rxjs 7.x of Angular 12** with the new **methods** and the new **importing style** according to the package structure change in **Angular 12**.
* Managed the **Angular** state using **rxjs** powered **state management** solution **NGRX**.
* Coding the core modules of the application compliant with the **Java/J2EE** coding standards and Design Patterns.
* Used various spring framework features including **Spring IOC, Spring AOP and Spring Templates**.
* Created Spring **MVC Controller** module for better handling of **client requests, consuming Restful data service** and sending response back to client.
* Developed a Micro Service Oriented application with **angular** in the front end and **Spring Boot** services in backend.
* Designed and developed the **REST** based **Microservices** using Spring **Boot, Spring Data with JPA (Java Persistence API)** and Used **Swagger** to interact with API endpoints of the **Microservices**.
* Experience in **JAVA 1.8** features like **Parallel Streams, Lambdas, functional interfaces and filters.**
* Created core **Java interfaces** and **Abstract classes** in **Java 1.8** for different functionalities and implemented **Multi Thread** concepts in **Java** classes to avoid **deadlocking**.
* Implemented **Spring Validators, persistence layer, and service layer components using Spring/Hibernate API and Spring/Hibernate annotations**, extensively used **Hibernate QL**.
* For design and development used **Spring Boot** and **Apache CXF** framework, **Microservice architecture**.
* To make internal security models with exciting features, it integrated it into new projects with **Spring Batch, Spring Microservices, Spring Boot, Pivotal Cloud Foundry and Netflix Eureka.**
* Implemented **OAuth 2.0**, Basic **AUTH** and **API Key policies** for API Security.
* Configured **OIDC** (OpenID Connect) on top of **OAUTH** 2.0 authentication and authorization framework to provide security to rest web services.
* Created Logging aspects for auditing and logging purposes using **Spring AOP** and implemented security for the application using **Spring Security**.
* Used Complete annotation driven approach to develop the application using **Spring Boot** and was also involved in setting up the **Spring Boot** configuration.
* Deployed **Spring Boot** based **Microservices** **Docker** container using **Amazon EC2** container services and using **AWS admin console**.
* Designed, configured, and deployed **Amazon Web Services (AWS)** for multiple applications utilizing the **AWS** stack (Including EC2, S3), focusing on high-availability, fault tolerance, and auto scaling.
* Developed backend interfaces using SQL **packages, stored procedures, Functions, Procedure,** Performed queuing services for the cloud data processing for certain tasks like file upload, data processing, notice to the On-Premise application using the AWS service **SQS (AWS Queuing System**).
* Done **Java** based integration testing and behavioral testing of the application using **Selenium (connects you to your browser), Unit testing, Junit4, TestNg, (organizes the tests) Mockito,** (**power mock for mock) and** for **rest testing mockmvc** and fluent **API, mockers service server for spring rest template template testing.**

**Environment:** HTML5, CSS, Angular 12,Angular 8 JIRA, GITLAB, Jenkins, HTTP/HTTPS, Firebug, Jasmine, Mocha, Agile Methodology, Oracle, PL/SQL, Apigee Gateway, Microsoft Azure, Java 1.8, NoSQL, Hibernate, Jira, AWS, GitHub.,AWS.

**Client: Kemper**

**[Aug 2021- Nov 2022]**

**Location: Birmingham, AL**

**Role: Full Stack Developer**

**Responsibilities:**

* Implemented **Spring Boot** to create a **CRUD** backend for our **React** frontend application.
* For client-side view rendering services, **React** **JS** typically used various predefined **Redux** libraries.
* Implemented **Store-Redux** for holding the whole state tree of application and **Reducer-Redux** for describing actions.
* Developed user interface by using **React JS**, **Redux** for SPA development and implemented client-side Interface using **React JS**.
* Worked with **React**’s lifecycle methods including **componentDidMount**, **shouldComponentUpdate**, **componentWillUnmount** and **React-Router** techniques.
* Wrote Client-Side code using **React JS** and used **NPM** & gulp directories to generate the server environments used browser, **flux** to manage with targeted URLs and to monitor the application.
* Adopted **ES6 Generators** for managing the **concurrency** in Angular Components in the application.
* Bottle micro-framework implemented with **RESTful service** and used **WADL** (Web Application Description Language) for modeling the resources provided by the services and relationships between them.
* To modularize code integrated **Spring Boot** and implemented Rest API’s using **Spring Rest** and integrated **Swagger API** for documentation.
* Functionality is implemented by developing **RESTful** web services using **Spring Boot**. Designing new components to fit in the already existing architecture using core **Java 17**, **multithreading**, design patterns like the **factory method, Abstract factory pattern, Singleton**, while developing web-based applications using **REST Web Services**.
* Built enterprise level **Java** applications to apply **OOP** and **AOP** language concepts using frameworks such as **Struts, Spring, Hibernate and JSF**.
* As a member of the team my contribution also involved migrating existing applications to robust **Java** Portal technologies.
* For logging, Security, Error handling mechanism used **Core Java, Spring Aspect Oriented programming concepts** and for implementation of **MVC** **architecture** using **Spring framework** and **customized user interfaces**.
* To build concurrent product models, the upload/download process used **Java Multi-threading programming, Synchronization, Java Mail API,** and **Collections API**.
* After gathering the business requirement, to encapsulate business logic designed and implemented most of the **Java** related portions of the application including **EJBs**.
* Used Micro service architecture with **Spring Boot** based services interacting through a combination of Rest and **Apache Kafka** message brokers and worked with **Kafka** **Cluster** using **Zookeeper**.
* Using **Spring 5.x** managed beans to use windows session state framework to persist sessions to the database.
* Implemented persistence layer using **Hibernate** that uses the POJOs (which do not have the business processes) to represent the persistence database tuples.
* Deployed and monitored web services using **Spring Cloud Foundry** Managed Domains and Routes with **Spring Cloud Service Registry.**
* Built real time pipeline for streaming data using **Kafka** and **Spark Streaming**.
* **Gradle** build tool is used to deploy the application on the web servers i.e., **JBoss**.
* Developed **Spring Batch** processes for financial reporting applications and modules using shell scripts on **Oracle database**, with partitions and subpartitions.
* Created automation framework and test scripts using **Selenium Web-Driver** for multiple platforms and browsers testing.
* Managed **Docker orchestration and Docker containerization** using **Kubernetes**.
* Implemented unit test cases using **Karma** and **Jasmine** on client side.
* Used **Kubernetes** to create a new cluster and container management tool.

**Environment:** AWS, AWS Load Balancer,EC2,Amazon DynamoDB, Swagger, Node JS, React JS, Web pack, ECMA6, Karma and jasmine, Spring 5.x, Spring Boot, Hystrix, Netflix ZUUL, Micro services architecture, Docker, Cassandra DB, OAUTH, Apache Kafka, Splunk, Kubernetes, Oracle, Kafka.

**Client: One Main Financials**

**[Oct 2019 – July 2021]**

**Location: Miami, FL**

**Role: Full Stack Java Developer**

**Responsibilities:**

* Implemented platform modules through **spring Java micro** and **Node.js services** by employing **Hapi.js**, as well as **front-end Angular JS.**
* For an angular project I used **the angular-cli** tool and built it for development of **Angular** applications using **Angular 8**.
* Created two way binding AngularJS Components and provide access to the server-side through Backbone's API from Backbone Model and Backbone Collection.
* Used **Angular**  to associate **HTML** elements to models, to provide dynamic user validations on the client side.
* UI components have been built using **Angular Dependency Injection, Models, data binding** and controllers.
* To prevent commonly found scope conflict implemented **Angular** service calls using **Angular Factory** with **Dependency Injection**. And to maintain each view data implemented **AngularJS Controllers**.
* Created a user interface with various screens by using React JS, Redux and various predefined components from NPM package manager.
* For making future pages involved in building stable React components and stand-alone functions.
* Implemented Spring Circuit breaker pattern, integrated Hystrix dashboard to monitor spring micro services, writing server-side request handling and automated functional test cases to validate UI functionality using Node JS.
* Developed the business logic code in Core Java/J2EE, Struts, JDBC.
* **Spring** **AOP** is used for logging, auditing, and security and transaction management to distinguish business logic from cross-cutting concerns.
* Developed **Java Servlets** and objects using **J2EE Standards**.
* Used **Spring** **Core** for **IOC** implemented for developing **Restful** **web services**.
* Project structure implementation was made based on **Spring MVC** pattern using **Spring Boot** and persisted data to **MongoDB** using **Spring Data**.
* Implemented **Micro-services** using **Pivotal Cloud Foundry (PCF)** platform built upon **Spring Boot Services**.
* Developed **MongoDB** Data Access Layer implementing **Mongo Repository** interface from **Spring Data** and Created various **MongoDB** collections and wrote services to store and retrieve user data from for the application on devices and used **Mongoose Connector** to access the **MongoDB.**
* Created dynamic schema design through **NoSQL MongoDB** by writing rich queries and achieved data in a collection to be distributed across multiple systems for **horizontal scalability** as data volumes increase through **Shading**.
* Also used **MongoDB** internal tools like **Mongo Compass, Mongo Atlas Manager & Ops Manager, Cloud Manager** etc. and concepts like **locking, indexes, replication, schema design.**
* Performed unit testing using **JUNIT** framework and tested **DAO’s** and business services**.**
* Used **SOAPUI** to test sending and receiving **XML** data and worked with **JMS** **Queues** for sending messages in point-to-point mode communication.
* Used **Jenkins** for building pipelines and **GIT** as version control.

**Environment:** React JS**,** Angular**,** Redux, MongoDB, Mongoose, Git, Jenkins, Spring Boot, Spring MVC, Structs**,** JAVA, J2EE, Core Java, JSF, AJAX, JavaScript, Apache CXF, Rest Service, Jax-RS, WSDL, JQuery, Karma and jasmine, Multi-Threading, AWS, Jquery, HTML5, CSS3, Ajax, ORACLE, IBatis, Docker, Mockito.

**Client: Commvault**

**[Jul 2017 – Sep 2019]**

**Location: Hyderabad, India**

**Role: Full Stack Developer**

**Responsibilities:**

* Developed UI pages using **HTML5, CSS3, JavaScript, jQuery, AJAX and JSON**.
* Effectively migrated the **Angular 2.0** application to **Angular 4.0** to use the enhanced features of **typescript, Angular Routers** and **Angular Components.**
* Used **Angular 4.0** to create views to hook up models to the **DOM** and synchronize data with the server as Single **Page Applications.**
* Effectively used **Angular Directives, filters, declarative templates, service providers** and **context aware communication.**
* Functionality is implemented by developing **RESTful** web services using **Spring Boot** and integrated **Spring Security** with JDBC to manage user registration and login.
* Migrating existing applications into **micro services architecture using Rest APIs, spring boot.**
* Developed **Java** Utility classes using servlets that were used by other modules for formatting, excel downloads and other purposes.
* Used various **JAVA**, **J2EE** design patterns like **DAO**, **DTO**, **Singleton** etc.
* Used **Spring Data JPA** with **hibernate** to interact with **oracle** using DAO patterns for accessing Database Objects.
* Used **Spring Boot, Spring MVC** to create new REST web services which interact with CICS Transaction Gateway (CTG) services to access mainframe programs.
* Developed **SOAP web services** (JAX-WS) for sending and getting data from different applications, **WSDL** generation and exposed the service at server-side over the Internet and tested the services using **JUnit**.
* Worked on securing the user’s account pages using **SSL**.
* Designed and documented **REST** **APIs**, including data formats and API versioning.
* **Unit, functional, end-to-end testing** is achieved using the TestNgframework.
* Loading data from Text Files to Database by using **SQL Loader.**
* **JDBC** was used to make connections to **MYSQL** and retrieve necessary data from it.
* Tested and deployed the application on **Tomcat Application Server**.
* Experienced in using Editors like **Visual Studio Code, Visual Studio, Eclipse, IntelliJ, Dreamweaver**.

**Environment:** HTML5, CSS3, JSP, JDK 1.5, Hibernate 2.5, Spring, Spring Boot, JMS, JavaScript, JQuery, MVC Architectural, STRUTS 2.x, SQL Loader, IBM WebSphere, Log4J.

**Client:  CIEL Services Pvt ltd**

**[May 2015 – Jun 2017]**

**Location: Hyderabad, India**

**Role: Java Developer**

**Responsibilities:**

* Developed backend logic using Core Java including **Collections, Multithreading and Exception Handling.**
* Developed/Customized **Java Server Pages (JSP)** for Customer User Interface (UI).
* Developed a **Monolithic** application and generated UI screens using **JSP, HTML, CSS, JavaScript**.
* Designed and developed **Message Driven Beans** (MDB) that consumed the messages from the **JMS**.
* Involved in development of **Action Forms, Action Servlets**, **Action,** and **Validating Action Forms** in **Struts framework.**
* Responsible in creating **Servlets**, which route submittals to appropriate Enterprise Java Bean (**EJB**) components and render, retrieved information and **EJB** Session Beans were used to process request from user interface using OSS.
* Worked on **JSP, Servlets, Struts framework** and production support issues of the existing applications.
* Schema design for **database** and created various **triggers, cursors, Oracle sequences, Oracle PL/SQL packages** in the Oracle database.
* Designed and Developed **JDBC** connections for data retrieval.
* Worked with **JPA and Hibernate 5.0** for developing persistence layers and have written custom DAOs to merge and retrieve records.
* Involved in writing complex queries in **Oracle**, to get the coupons for products considering different factors like time schedule, special dates, special clients etc.
* Developed **SOAP web services**(JAX-WS) for sending and getting data from different applications **WSDL** generation and exposed the service at server-side over the Internet and tested the services using **JUnit**.
* Consumed **SOAP** and **REST** web services to get the persisted data from different databases.
* Build and maintain **SQL** scripts, indexes, and complex queries for data analysis and extraction.
* Develop a business continuity plan for the **SQL Server Databases** using **JDBC** drivers.
* Developed web components using JSP, Servlets and JDBC
* Deployed application on **Client Linux server**.
* Used **ANT** build tool for compiling and generating war files.

**Environment:** Core Java, J2EE, JSP, HTML, MVC, Servlets, Struts, MyBatis, JMS, MDB, EJB3.1, JUnit, PL/SQL, DAO, Web Logic Server, Ant, SOAP, JAX-WS, WSDL, Junit, SQL.