**Self Introduction**

* Hi, I’m Raguram.
* I have a 5 years of work experience in a web application development.
* I have a good work experience in front-end, back-end and datatbases.
* Front-end technologies like Angular, HTML, CSS, Bootstrap, Javascript, JQuery and Ajax.
* Back-end technologies like Java, Spring, Spring boot and Microservices.
* Databases like Postgres SQL and MYSQL.
* In additionally I have good working knowledge in Agile, Git, Bit-bucket, Jenkins, Jira and good exposure in AWS.

**Project:**

1. **DWM Finance Collection System:**

* My current project was DWM Finance Collection System, DWM means Daily, Weekly and Monthly Collection System.
* It helps for optimize the daily, weekly and monthly cash flow and reporting details.
* It mainly used for the finance user don’t need to write the customer account details in notes and don’t need to emi calculation. It over all takecare of it.

**Roles and Responsbilities:**

* This project I’m as a web and API developer.
* I’m actively involved in Agile methodology for two weeks sprint.
* In this project I have used Angular for front-end design and back-end development to using with spring boot microservices.
* In databases wise we are used PostgresSQL.
* Used Spring Security and JWT for Authentication and authorization extensively.
* In this project I have implemented to both front-end & back-end the modules like User OTP, Membership, Agent user create and updated form, Profile module.
* In additionaly fron-end side I had implemented in SEO concept.
* In back-end side I had implemented the User OTP to connect with third party API using with rest template.
* Finally this project we deployed in AWS.

1. **Texpack: Client -** Woven Global Soft

* Texpack is the Textile management web Application.
* It gives some real time funtionalities for organization about the production and supply chain like these…
* It mainly focus on improving to maintain all textile business core task in a single platform.

**Roles and Responsbilities:**

* This project I’m as a web and API developer.
* I’m actively involved in Agile methodology for two weeks sprint.
* In this project I have used Angular for front-end design and back-end development to using with spring boot microservices.
* In databases wise we are used MySQL.
* Used Spring Security for Authentication and authorization extensively.
* In this project I have implemented to both front-end & back-end the modules like Department, Yarn-count, Yarn-variety.
* In additionaly fron-end side I had implemented in SEO concept.
* Finally this project we deployed in AWS.

1. **Policy Input: Client** **-** SFG, SD, USA

* Policy input is a intranet web site.
* It mainly used for the agent can easily enter the policy information details like such as Children purposed insurance and Beneficiary details.

**Roles and Responsbilities:**

* This project I’m as a web and API developer.
* I’m actively involved in Agile methodology for two weeks sprint.
* In this project I have used Angular for front-end design and back-end development to using with spring boot microservices.
* In databases wise we are used PostgresSQL.
* Used Spring Security and Okta for Authentication and authorization extensively.
* In this project I have implemented to both front-end & back-end the modules like Children purposed insurance and Beneficiary details.
* Finally this project we deployed in Jenkins.

**Angular:**

It is a front-end web framework based on typescript which is used to developing the enterprise web applications.

**HTML: (Hyper Text Markup Language)**

HTML describes the structure of a Web page.

**CSS: (Cascading Style Sheets)**

It is a style sheet language which is used to describe the look and formatting of a html document.

**Tailwind CSS:**

 Tailwind CSS is basically a utility-first CSS framework for **rapidly building custom user interfaces**.

**Bootstrap:**

It is most popular css framework for developing responsive and mobile-first website. Which is used to develop our web pages in responsively.

**Java Script:**

* It is a scripting language.
* Used to create a dynamically updating content.

**JQuery:**

* It is open source and Javascript library.
* It is used to handle very much easier than JS.
* Designed to Manipulation, CSS animation, AJAX.

**Ajax: (Asynchronous Javascript)**

It is used to fetch the data from database without reloading the webpage.

**JSP:**

* Used to create the web pages.
* It contains HTML tags and JSP tags.
* It is used to write a java code using scriptlet tag.

**Servlet:**

* It is a server side java program.
* Used to create web apps.
* It handle client request.

**JSON:**

* Lightweight format for storing and transferring the data.
* It is used to send the data from the server to web page.

**Java:**

* Java is a object-oriented programming language.
* Which is used to build web application.
* It have some features like platform independent, securied, multithreaded, garbage collected.

**Spring:**

It is a **lightweight**, **loosely coupled** and integrated framework for developing **enterprise application** in java.

**Springboot:**

Spring boot is a **spring module** and it used to **create stand alone spring based application** that you can run because it needs very little spring configuration.

**Spring Security:**

* It is a framework. It provides many features like authentication, authorization.
* Spring Security is **a powerful and highly customizable authentication and access-control framework**.

**Microservice:**

Each application build as set of services and each services run own processes and communicate through API.

**MySQL:**

MySQL is a relational database management system, like SQL Server, Oracle, that is used to manage SQL databases.

**PostgresSQL:**

It is a object-oriented relational database management system, that is used to manage SQL databases.

**JDBC: (Java Database Connectivity)**

Used to connect the Java code to Database.

**JPA:**

It is act as a bridge between object-oriented domain model & relational database system.

**SQL: (Structured Query Language)**

* Standard language dealing with relational database.
* Used for CRUD databased record. (or) used to handle CRUD process in a database records.
* It help optimizing and maintanence the DB.

**Database:**

Used to store collection of data.

**Agile:**

* It is the one of the famous Methodology for project Management.
* In this methodology, we have followed daily standup meeting, story grooming & reviewing the acceptance creteria.

**Bitbucket** **&** **Git:**

* It is a open-source tool.
* Used for source code management.

**Jenkins:**

* It is a open-source tool.
* It is used to implement Continous Integration/Continous Deployment (CI/CD) workflows called pipelin.

**Workflow:**

1. I have done my code, I commited in my feature brance, then my teammates are to review my source code, once they approved it will merge to the master branch.

2. In jenkins side, I created one job name and to configured the our project repository url, then jenkins to clone our project through that url.

3. Simultaneously, jenkins to interact with maven, so maven create the war or jar file & it given to the jenkins.

4. Then, jenkins interact with tomcat server, to deploy the jar or war file into tomcat server.

* Jenkins is going to Git Repo. (through git project URL)
* Jenkins is cloning from the project repo.
* Jenkins is integrated with maven to perform build process.
* The maven create the war or jar file & it given to the jenkins.
* Them jenkins deploy the jar or war file into tomcat server.
* Jenkins Job started, it has taken the project from repository. (Jenkins to monitoring the git repository)
* Then, jenkins to integrated with maven to perform build process.
* Then, jenkins integrated with the tomcat container.

**Jira:**

JIRA is **a software development tool used for project management.**

**Apache Tomcat:**

* It is a web-container.
* It allows user to run JSP/Servlet pages that are based on web-apps.
* It is used to as the HTTP Server.

**SonarQube:**

* It is a Open-Source & Quality management tool.
* The developer can easily find out the code velunerabilities, code smells and debugs.

**Log4j:**

To log regular debug and Exception statement.

**Rest Template (or) Rest API:**

Used to connect one microservie to another microservice.

**Workflow: (User OTP) –** Third party API: Smart Enterprise Solutions

* They give API.
* Then I configured the API parameters in applicatioin.yml file like username, password, from, to, country code, text.
* Then I get these values in service class by using @Value.
* Then use rest template (to use exchange method), to pass the api, then return to the response entity.

**Junit:**

Overall, Junit is an essential tool for testing Spring Boot applications.

Example: @SpringBootTest, @Test

**Batch Jobs:**

A **lightweight**, comprehensive **batch framework** designed to **enable the development of** robust **batch applications** vital **for the daily operations of enterprise systems.**

* We are implemented our project in membership concept.
* It means in which customer to get membership expired date, so we change their roles, so this scenario we need to check 2times per day.
* This concept we implement our project by using spring batch concept.
* 1st we declare @EnableSheduling in project level.
* Then using @Scheduled annotation with cron job conept.
* Example: @Scheduled(cron = “0 \*/1 \* \* \* \* \*”)

**AWS:**

* It provides on-demand cloud computing platforms.
* This platforms provided the
* Infrastructure as a Service (Iaas):

Amazon EC2

* Platform as a Service (Paas):

Elastic Beanstalk

* Software as a Service (Saas):

Google Apps (Gmail, Zoom)

**Maven:**

Maven is a build automation tool that manages a project's build process.

**STS (Spring Tool Suite):**

Spring Tool Suite (STS) is an Integrated Development Environment (IDE) based on Eclipse and specifically designed for developing Spring Framework-based applications.

**Spring Boot MVC Framework:**

Spring MVC is a Model, View, and Controller based framework used to build web applications. Spring Boot is one of the most widely used frameworks in the REST API development field. It is developed on top of the conventional spring framework. It is used to build stand-alone web spring applications.

* Model: Pojo class
* View: User Interface
* Controller: It is a business logic (Combination of Model & view)

**JWT (Json Web Token):**

**JSON Web Token or JWT**, as it is more commonly called, is an open Internet standard (RFC 7519) for securely transmitting trusted information between parties in a compact way. The tokens contain claims that are encoded as a JSON object and are digitally signed using a private secret or a public key/private key pair.  **(or)**

Basically the identity provider (IdP) generates a JWT certifying user identity and Resource server decodes and verifies the authenticity of the token using secret salt / public key.

* User sign-in using username and password or google/Facebook.
* Authentication server verifies the credentials and issues a JWT signed using either a secret salt or a private key.
* User’s Client uses the JWT to access protected resources by passing the JWT in HTTP Authorization header.
* Resource server then verifies the authenticity of the token using the secret salt/ public key.

**Okta:**

Okta provides features like authentication, authorization, and social login for web, mobile, or API services. Additionally, it has robust support for the Spring Framework to make integrations quite straightforward.

**SEO:**

SEO means **Search Engine Optimization**. It is used to get your Web application on top of the user’s search list on search engines.

**Workflow:**

1. Add the new package for angular universal.

* ng add @nguniversal/express-engine@14.1.0 (version **\***important)

2. Then, you have some default files

* server.ts
* main.server.ts
* app.server.module.ts

3. Then, we can Setting the Titles and Metadata in component.ts file

* Title and Metadata comes under the @angular/platform-browser.
* Angular provides a Meta service for changing titles and metadata on web applications.
* updateTag()
* addTag()
* removeTag()
* getTag()
* Inside these tages we added the content, description etc…

4. After Run the Application

* ng serve or npm run dev:ssr

**(or)**

4. Create a build(Server Side Render) with the following command:

* npm run build:ssr ( ssr: server side render)
* npm run serve:ssr
* then, you have localhot:4000

5. Then, you can see your project website Source page.