Objectives

To create a dynamic and responsive Java travel website for booking travel tickets for all types of transport mediums.

Problem Statement and Motivation

Problem Statement:

In this project, you should be able to develop a front-end web app using Angular and a Java backend using Spring Boot, configure the applications using Docker containers, deploy the project on AWS using CI CD Pipeline

Real-World Scenario:

TravelBuddy is a company that provides travel solutions for your travel bookings. They found out that online booking with companies, such as Ixigo and Ibibo are gaining more profits by eliminating middlemen from the equation. As a result, the team decided to hire a Full Stack developer to develop a online booking portal with a rich and user-friendly interface.

Skills used in the project and their usage in the industry are given below:

• Angular-Platform for building mobile and desktop web applications

• Java and SpringBoot-Leading technology to develop backend

• Docker-Deliver Software as Containers

• AWS-Host and Deploy your Apps in World's leading Cloud Platform

• Jenkins-Use Jenkins to build CI CD Pipelines

Tasks

The following tasks outline the CI/CD pipeline creation process:

1. Develop the Front End with Angular for the Admin to add flight details, cities, airline brands and fairs from the Interface with authentication for the Admin User.
2. Develop an End User Web Application listing the airlines based on the data provided by user with respect to the source and destination cities and date of travel.
3. Define the structure of a database and create necessary tables using SQL in MySQL Database or MongoDB as per your preference.
4. Develop a Java Backend using Spring Boot containing various microservices for various functionalities like login, register, booking flights, searching relevant flights, filtering search based on different parameters like airlines and duration of journey, dummy payment gateway, tracking flight status.
5. Perform the Front-End and Back-End communication using HTTP Client. Define Jenkinsfile for both Angular and Java Projects for the automated builds.

6. Define Dockerfile for both Angular and Java Projects to develop images and run them as containers.
7. Develop a CI CD Pipeline in Jenkins for both Angular and Java Backend Projects.
8. Using AWS Launch EC2 Instances and configure other required Services. Deploy the Projects on EC2 Instance

1. Task 1, 2:Angular Components, Routing, Services and AuthGuard, Forms
2. Task 3:SQL CRUD Commands, Primary and Foreign Key Relationship
3. Task 4:Spring Boot Web Dependency, RestController, RequestMapping, Post and Get Requests
4. Task 5: Jenkinsfile stages and step declarations
5. Task 6:Dockerfile and commands to assemble an image
6. Task 7: Jenkins Pipeline Project Creation with git SCM
7. Task 8: AWS, EC2, SSH/CloudShell Connection, Tool Configuration