

Travel Agency

A Leading Travel Agency in India has decided to diversify its business into ecommerce way. It has hired a technology partner to put in place, the necessary infrastructure required including creating a website and host it online for its customers. As part of requirements, the below mentioned specification has been given to the partner to implement.

Requirements Specification:

1. Home Page should provision for booking Flight Tickets, Cabs & Deals. It should also have a Contact Us feature
2. Flight & Cab ticket Pages should have provision to book only a one-way ticket
3. The Deals Page can show the deals available in a simple textual form
4. The Contact Us Page should display a Phone Number, Email ID and availability timing of the Customer care team

Note: Make your own assumptions while developing Flight, Cab, Deals and Contact Us pages.

As a DevOps Professional in the project, you are expected to do the following tasks:

1. Create a Git Repository Project in GitLab (Project Name should be DevOps-ILP1-<EmpID>)
2. Create a Maven Project & construct necessary class files as per your design and do an initial commit in Git
3. Develop in an incremental approach the
 - a. code implementing above requirements
 - b. unit test code (preferably in Junit) to test the implementation code
 - c. Upon successful test, do a commit with suitable message
4. Perform functional testing using selenium. Upon successful completion, do a commit with suitable message
5. Create a Jenkins job with Github Actions to perform build process automatically for every change made into the git

The partner organization has selected

- GIT as its SCM tool
- Maven as a build process tool to build and test the application using Junit framework in an incremental fashion
- Selenium to test the application developed
- Jfrog Artifactory to store binary to be deployed.
- Code Quality report from SonarQube
- Jenkins for orchestration

Note: During integrated learning project execution, you may use any alternative tools.

Project Inputs:

- Application Source code
- Web configuration files & UI pages
- Junit framework based unit test code s
- Functional Testing scripts using selenium
- Required dependency details if any. (pom.xml)
- Read Me (file name: readme.txt => Refer to know services details)

Note: Use of give sample code is optional. You may develop your own code for above requirements and tool integration. Use above code for learning/reference only.

Project Tasks

As a DevOps Professional in the project, you are expected to do the following tasks:

1. Create a Git Repository Project in GitLab – Use below mentioned naming convention
 - a. <Your ADID>/DevOpsProfessional/ ILP_CI-CD/<projectTitle>
2. Create a Project structure as per the Build tool and do an initial commit /Push to Gitlab
3. Experiment to validate / verify the execution of solution as per the requirements.
 - Example: Run any/all Maven goals
4. If step 3 is successful then configure and create Jenkins job to perform build process automation
 - Ensure automated build trigger and notification
5. Take necessary/suitable snapshots as listed below.
 - a. Gitlab Commit history
 - b. Jenkins Global Tool Configuration
 - c. Jenkins Configure System
 - d. Jenkins Project Configuration
 - e. Jenkins Build History
 - f. Application execution copy

General Instructions

Gitlab repository should have the following artefacts

- Structured code base & Build process configuration file (pom.xml)
- Snapshots listed under project tasks (Step 5 as mentioned under Project tasks)
- Jenkins triggered email upon successful completion of below listed phases to avinash.patel@wipro.com and raghavendran.sethumadhavan1@wipro.com (An email should be triggered to yourself, upon failure at any phase)

NOTE: While sending email, please follow below convention

- ✓ Subject: <Your ADID>/DevOpsProfessional/ ILP_CI-CD/<projectTitle>
 - Please ensure email triggered from your email id only
 - For all failures emails should be send back to you only
 - After deliverables are pushed to Gitlab, do post your gitlab URL on suggested MS Teams link

Integrated Learning Project Objective & Deliverables

1. Continuous Integration (CI)

Objective:

- Construct project structure as per recommendation of build processing tool
- Create application source codes (Business Logic, Unit Test code, other scripts /automation scripts –if required)
- Integrate necessary tools with CI server to perform Continuous integration
 - SCM : Gitlab
 - Build Tool : Maven
 - Unit Testing : Junit
 - Artifactory : Jfrog
 - CI Server : Jenkins
- Automate CI process to produce artifact (war/jar) and notify with suitable response

Deliverables:

- a. Project Structure with sources
 - i. Application Source Code
 - ii. Application Unit Testing Source Code
 - iii. UI & web configurations sources
 - iv. Any automation scripts /other scripts (if applicable)

- v. Read Me (readme.txt) having configuration and usage details
- b. Snapshots
 - i. Gitlab
 - 1. Project structure
 - 2. History (commit details, tags, branches etc.,)
 - ii. Jenkins
 - 1. Project Configuration (SCM, Build Trigger, Steps, Build and Post build actions)
 - 2. Build History
 - 3. Other Configuration (Artifactory, SonarQube and other suitable setup/Configuration)
 - iii. Artifacts details from Jfrog Artifactory
 - iv. Auto triggered mail (as mentioned above)

2. Continuous Delivery (CI-CD)

Objective:

- Construct project structure as per recommendation of build processing tool
 - Create application source codes (Business Logic, Unit Test code, other scripts /automation scripts –if required)
 - Integrate necessary tools with CI server to perform Continuous integration and Continuous Delivery
 - SCM : Gitlab
 - Build Tool : Maven
 - Unit Testing : Junit
 - Artifactory : Jfrog
 - Code Review : SonarQube
 - Testing : Functional test using Selenium
 - Test Environment : server-Apache Tomcat, browser - Chrome/Firefox/chromium
 - CI Server : Jenkins
 - Automate CI process to produce artifact (war/jar) and notify with suitable response
- a. Snapshots
 - i. Jenkins Configuration – Artifactory, SonarQube and WebServer
 - ii. Jenkins Project Configuration & Build History
 - iii. Artifacts details from Jfrog Artifactory
 - iv. Code quality report from SonarQube
 - v. WebServer UI Snapshot
 - vi. Application UI responses