|  |  |  |  |
| --- | --- | --- | --- |
| Flight Ticket Booking System | | | |
| **Problem Statement**:  An efficient web application is needed through which users can view available flights, from one destination to another, and can book tickets. | | | |
| **Requirement Specifications**:   * Users can login after registering to the website. * Once logged in, users can search, view available flights from one destination to another, can view route, and can book tickets. * Users can also change password for their accounts. | | | |
| Assumptions (if any): | | | |
| Deliverables: (Version Controlled in Source Code Repository).  Build Automation   * Project Structure with source * Snapshots   + Gitlab History   + Jenkins Project Configuration (Auto build trigger)   + Jenkins Build History   Deployment Automation ( After Build Automation)   * Snapshots   + Jenkins Configuration – Artifactory, SonarQube and WebServer   + Jenkins Project Configuration   + Jenkins Build History   Release Automation (Build + Deployment Automation)   * Configuration Scripts * Gitlab History * Snapshots   + Jenkins Configuration – Ansible and Docker   + Jenkins Project Configuration   + Jenkins Build History | | | | Deliverables: (Version Controlled in Source Code Repository).  Build Automation   * Project Structure with source * Snapshots * Gitlab History * Jenkins Project Configuration (Auto build trigger) * Jenkins Build History * Auto triggered mail   Deployment Automation ( After Build Automation)   * Snapshots * Jenkins Configuration – Artifactory, SonarQube and WebServer * Jenkins Project Configuration * Jenkins Build History * Auto triggered mail   Release Automation (Build + Deployment Automation)   * Configuration Scripts * Gitlab History * Snapshots * Jenkins Configuration – Ansible and Docker * Jenkins Project Configuration * Jenkins Build History * Auto triggered mail |
| Purpose | Description | Tool Used | Remark |
| SCM | Project Management: Complete Project Folder structure with Source and Read Me document giving brief description for repository. | GIT | (Provide complete Path in SCM) |
| DevOps Pipeline | Give brief description about how following DevOps Principles are considered in the project at each stage:  Continuous development  Continuous Build  Continuous Integration  Continuous test  Continuous release  Continuous deployment  Continuous monitoring | Gitea  Jenkins  Maven  Junit  Selenium  JFrog Artifactory  Docker Swarm/Docker  SonarQube | (Environment - Tools used) |
| Environment / Configuration Automation | * Whether application is deployed on server or on container environment * Environment may involve multiple binaries, libraries, app/web/database servers.   + List the packages to be installed.   + List of configuration files to be modified   + List the services & any other relevant info | Ansible  Docker |  |

**Project Guidelines:**  You may develop your own code for above requirements and tool integration.

1. Perform Build Automation
   * Scope for managing multiple branches / modules
   * Scope for automating execution of test scripts
   * Scope for automating customized notification based on build process (optional)
2. Perform Deployment Automation (After Build Automation)
   * Scope for Automated deploy and test different artifacts
   * Scope for Automated environment configuration
3. Perform Release Automation (Build + Deployment Automation)
   * Scope for Automated Server orchestration
   * Scope for Automated Container Orchestration
   * Scope for Integration of Monitoring tools

**Project Tasks/Deliverables:**

1. Create a Git Repository Project in GitLab – Use below mentioned naming convention
   1. <Your ADID>/DevOpsProfessional/<batch#>/Capstone/<projectTitle>
2. Structured code base & Build process configuration details (example: pom.xml)
3. Few Snapshots to review Build/Deploy/release/Notification status