**LAB EXERCISE – 8**

**Applying CI/CD Principles to Web Development Using Jenkins, Git, and Local HTTP Server**

**Task : Set up a CI/CD pipeline for a web development project using Jenkins, Git, and webhooks, without the need for a Jenkinsfile.**

**CI/CD Workflow:**

* Code Changes: Developers make changes to the web application's source code locally.
* Git Repository: Developers push their code changes to a Git repository, such as GitHub or Bitbucket.
* Webhook: A webhook is configured in the Git repository to notify Jenkins whenever changes are pushed.
* Jenkins Job: Jenkins is set up to listen for webhook triggers. When a trigger occurs, Jenkins initiates a CI/CD pipeline.
* Build and Test: Jenkins executes a series of predefined steps, which may include building the application, running tests, and generating artifacts.
* Deployment: If all previous steps are successful, Jenkins deploys the application to a local HTTP server for testing.
* Verification: The deployed application is tested locally to ensure it functions as expected.
* Optional Staging: For more complex setups, there might be a staging environment where the application undergoes further testing before reaching production.
* Production Deployment: If the application passes all tests, it can be deployed to the production server.

PROCEDURE:

* **Set Up the Web Application and Local HTTP Server using Apache or any other you comfortable with. Make sure it is configured right way, and its application is in running condition**
* **Set Up a Git Repository**
* **Install and Configure Jenkins**
* **Create a Jenkins Job using “Freestyle Project” type. Configure the job to use a webhook trigger. (Use "GitHub hook trigger for GITScm polling")**
* **Set Up the Jenkins Job (Using Execute Shell)**
  + In the job configuration, go to the "Build" section.
  + Add a build step of type "Execute shell."
  + In the "Command" field, define the build and deployment steps using shell commands. For example:

**# Checkout code from Git**

**# Build your web application (e.g., npm install, npm run build)**

**# Copy the build artefacts to the local HTTP server directory**

*rm -rf /var/www/html/webdirectory/\**

*cp -r \* /var/www/html/webdirectory/*

* **Set Up a Webhook in Git Repository**

1. In your Git repository (e.g., on GitHub), go to "Settings" and then "Webhooks."
2. Create a new webhook, and configure it to send a payload to the Jenkins webhook URL (usually http://jenkins-server/github-webhook/). (Make sure to set the content type to "application/json.")
3. OR use “GitHub hook trigger for GITScm polling?” Under Build Trigger

* **Trigger the CI/CD Pipeline**
  1. Push changes to your Git repository.
  2. The webhook should trigger the Jenkins job automatically, executing the build and deployment steps defined in the "Execute Shell" build step.
  3. Monitor the Jenkins job's progress in the Jenkins web interface.

* **Verify the CI/CD Pipeline**

Visit the URL of your local HTTP server to verify that the web application has been updated with the latest changes.