**Experiment No. 2**

1. What is version control?

Version control is a system that records changes to files over time.  
It helps track revisions and collaborate on code or documents efficiently.

1. What is staging area?

The staging area in Git is a place where changes are gathered before committing, allowing you to control and review what goes into the final snapshot.

1. What care is to be taken when merging two branches?

When merging two branches, ensure no conflicts exist, and carefully review changes.  
Test the merged code thoroughly to avoid breaking functionality.

4.Commands

* git –version
* git config --global user.name "pujagit"
* git config --global user.email “[padiya.puja@rait.ac.in](mailto:padiya.puja@rait.ac.in)”
* mkdir makes a new directory
* cd changes the current working directory.
* gitinit
* $ git clone [project url]
* $ git status
* $ git add [file]
* $ git diff [file] Show changes between working directory and staging area.
* $ git diff --staged [file] Shows any changes between the staging area and the repository.
* $ git checkout -- [file] Discard changes in working directory. This operation is unrecoverable.
* $ git reset [file] Revert your repository to a previous known working state.
* $ git commit Create a new commit from changes added to the staging area. The commit must have a message!
* $ git rm [file] Remove file from working directory and staging area.
* $ git branch [-a] List all local branches in repository. With -a: show all branches (with remote).
* $ git branch [branch\_name] Create new branch, referencing the current HEAD.
* $ git checkout [-b][branch\_name] Switch working directory to the specified branch. With -b: Git will create the specified branch if it does not exist.
* $ git merge [from name] Join specified [from name] branch into your current branch (the one you are on currently).
* $ git branch -d [name] Remove selected branch, if it is already merged into any other. -D instead of -d forces deletion.
* $ git log [-n count] List commit history of current branch. -n count limits list to last n commits.

**Experiment No.: 3**

1. What is difference between git and github?

Git is a version control system for tracking changes in code, used locally, while GitHub is a cloud-based platform for hosting Git repositories, enabling remote collaboration and sharing.

2. How do you push your project into remote repository?

To push a project to a remote repository, use git push <remote> <branch>, where <remote> is typically origin, and <branch> is your branch name.Ensure you’ve committed your changes before pushing.

3. Is it possible to revert changes after commit? Is so, how?

Yes, you can revert changes after a commit using git revert <commit> to create a new commit that undoes the changes.  
Alternatively, git reset --hard <commit> can reset the branch to a previous state, discarding later commits.

4.Steps

Step 1: Create a GitHub account

Step 2: Create a new repository

Step 3: Create a file

Open the Terminal program on your computer by using open with git bash

In the terminal, type: mkdir Demo

cd Demo

Then enter: echo "#Demo" >> README.md

To check that the file was created successfully, enter: cat README.md

Step 4: Make a commit

To make a commit, enter: git commit -m "first commit"

**B.Synchronizing repositories:**

git remote add origin <https://github.com//Demo.git>

* $ git push [--tags] [remote] Push local changes to the remote. Use --tags to push tags.
* $ git push -u [remote] [branch] Push local branch to remote repository. Set its copy as an upstream
* $ git fetch [remote] Fetch changes from the remote, but not update tracking branches.
* $ git fetch --prune [remote] Delete remote Refs that were removed from the remote repository.
* $ git pull [remote] Fetch changes from the remote and merge current branch with its upstream.

**C. Tagging known commits:**

* **$ git tag List all tags.**

**D.** **Reverting changes**

* **$ git reset [--hard] [target reference] Switches the current branch to the target reference, leaving a difference as an uncommitted change. When --hard is used, all changes are discarded.**
* **$ git revert [commit sha] Create a new commit, reverting changes from the specified commit. It generates an inversion of changes.**

**Experiment No.: 4**

**Installation of Jenkins To install and configure Jenkins to setup a build Job.**

1.What is Jenkins?

Jenkins is a Java-based open-source automation platform with built-in continuous integration plugins. Jenkins is used to continuously build and test your software projects, making it simpler for developers to incorporate changes to the project and for users to get a new build.

2. Advantages of Jenkins include:

• It is an open-source tool with great community support.

• It is easy to install.

• It has 1000+ plugins to ease your work.

• It is free of cost.

• It is built with Java and hence, it is portable to all the major platforms.

1. Jenkins Features
2. Continuous Integration and Continuous Delivery:
3. Easy configuration
4. Easy installation
5. Plugins
6. Extensible
7. Steps for Installation of Jenkins:
8. Installation of JAVA
9. Download Jenkins from the official site, run the installer, and follow the setup wizard
10. Choose install folder, enter user credentials or select LocalSystem, set Jenkins port, and specify Java directory
11. Select features to install, click Install, and finish the installation process.

**Unblock Jenkins**

1. Navigate to http://localhost:[port number] and locate the initialAdminPassword file.
2. Open the file, copy the password, paste it on the Unblock Jenkins page, and click Continue.

**Customize Jenkins**

1. Click the Install suggested plugins button to install frequently used plugins automatically.
2. After plugin installation, create the first admin user and click Save and Continue.
3. Confirm the port number on the Instance Configuration page and click Save and Finish.
4. Click Start using Jenkins to access the dashboard.
5. In the Jenkins dashboard, click Create a job to build your first software project.

7.What is Continuous Integration?

Continuous Integration (CI) is the practice of frequently merging code changes into a shared repository.It automates testing and building to detect issues early and improve code quality.

8. What is CI/CD?

CI/CD stands for Continuous Integration and Continuous Delivery/Deployment.  
CI automates code integration and testing, while CD automates deployment to production.

9. Which tools can be plugged with Jenkins?

Jenkins can integrate with tools like Git, Docker, Maven, Gradle, Kubernetes, Ansible, and Selenium. It also supports integration with Jira, Slack, and other collaboration tools.

Experiment No.: 5

To build the pipeline of jobs in Jenkins, create a pipeline script to deploy an application overServer.

1. How to create Jenkins Scripted Pipeline?

* Firstly, from the Jenkins dashboard, click on New Item on the left panel.
* Secondly, enter the name for your pipeline, select Pipeline from the list. After that, click OK.
* After that, go to the Pipeline tab, and from the Definition, the dropdown selects the Pipeline script
* The next step is to write your pipeline code in the web UI provided by Jenkins. Let us see a sample pipeline example as available in Jenkins-
* You need to copy and paste the same in UI.
* After that, click on Save. Conclusively, this finishes the process.

2. How to create a Declarative Jenkins Pipeline?

* To create a declarative pipeline, you need to have a Jenkinsfile in place. Since I will be using the project from my Github account, I have already placed the Jenkinsfile in my project.
* For creating a Declarative Pipeline, you may follow step#1 and Step#2 from the scripted pipeline creation steps stated above and then follow the below steps- Go to the Pipeline tab, and from the Definition, the dropdown selects the Pipeline script from SCM
* Go to the Pipeline tab, and from the Definition, the dropdown selects the Pipeline script from SCM
* Now, you will get an option to input your Repository URL and credentials.
* Next, you may set the branch or let it be blank for any branch. In the script path, you need to write the Jenkinsfile name that exists in your repository. Click on Save, and there you go, your declarative pipeline is ready for use.
* Now that you are all set with your pipelines, you can execute the same from your Jenkins UI. All you need to do is select your pipeline and click on Build Now link on the left panel.
* 7.Then run the pipeline and results will be shown

3. What is pipeline?

A pipeline in Jenkins is an automated workflow that defines the process of building, testing, and deploying applications.  
It is defined using a Jenkinsfile, which specifies the stages and steps involved in the CI/CD process.

4. What is declarative pipeline?

A declarative pipeline in Jenkins is a simplified, structured way to define a pipeline using the pipeline block.  
It focuses on clarity and readability, with predefined stages like stages, steps, and post.

5.What is scripted pipeline?

A scripted pipeline in Jenkins uses Groovy scripting to define the pipeline.  
It provides more flexibility and control but is less structured compared to the declarative pipeline.

6. What is Tomcat server?

Tomcat is an open-source Java servlet container developed by Apache.  
It is used to deploy and run Java-based web applications and supports Java EE components.

7. What is a web server?

A web server is software that delivers web content (like HTML pages) to users over HTTP/HTTPS.  
It handles requests from clients (browsers) and serves the requested web pages or resources.

8. What are the other web servers available for deployment?

Other web servers include Apache HTTP Server, Nginx, Microsoft IIS, Lighttpd, and Caddy.  
They are used for hosting and serving web applications across different environments.

**Experiment No.: 6**

To Setup and Run Selenium Tests in Jenkins Using Maven

1. Selenium WebDriver?

Selenium WebDriver is a tool that allows interaction with web pages using various programming languages, including Python. It enables the automation of web browsers, testing web applications, and performing repetitive tasks.

2. Pytest

Pytest is a testing framework for Python that allows the creation of simple and scalable tests. It provides various features like fixtures, parameterization, and plugins for writing complex tests.

3. Jenkins

Jenkins is a popular open-source automation server used for continuous integration and continuous delivery (CI/CD) of software applications. It enables the automation of building, testing, and deploying software applications

4. What is Selenium, and what are its components?

Selenium is an open-source automation testing tool with components: Selenium IDE, RC, WebDriver, and Grid.

5. **What is WebDriver in Selenium?**  
WebDriver is a core Selenium component that automates browser actions by directly communicating with the browser.

1. **How do you set up a Selenium project using Python?**  
   Install Selenium via pip, set up a virtual environment, download the browser driver, and write test scripts using WebDriver.
2. **What is Maven and how is it used in Jenkins?**  
   Maven is a Java-based build automation tool used in Jenkins to manage dependencies and automate builds/tests.

8.H**ow to run Selenium tests using Jenkins?**  
Create a Jenkins job, configure environment and commands to run the Selenium test scripts using Python.

9. **What is Pytest and how is it used with Selenium?**  
Pytest is a Python testing framework used with Selenium to write structured, scalable, and readable automation tests.

10. **How to install and configure Selenium in a Python project?**  
Install with pip install selenium, download browser driver, set up virtual environment, and write test scripts

**Experiment No.: 7**

Docker containerization and installation

1. Container?

Docker Container is a standardized unit which can be created on the fly to deploy a particular application or environment. It could be an Ubuntu container, CentOs container, etc. to full-fill the requirement from an operating system point of view.

2. Docker Image

Docker Image can be compared to a template which is used to create Docker Containers. They are the building blocks of a Docker Container. These Docker Images are created using the build command.

3. Docker Container

Docker Containers are the ready applications created from Docker Images. Or you can say they are running instances of the Images and they hold the entire package needed to run the application. This happens to be the ultimate utility of the technology.

4. Docker Registry

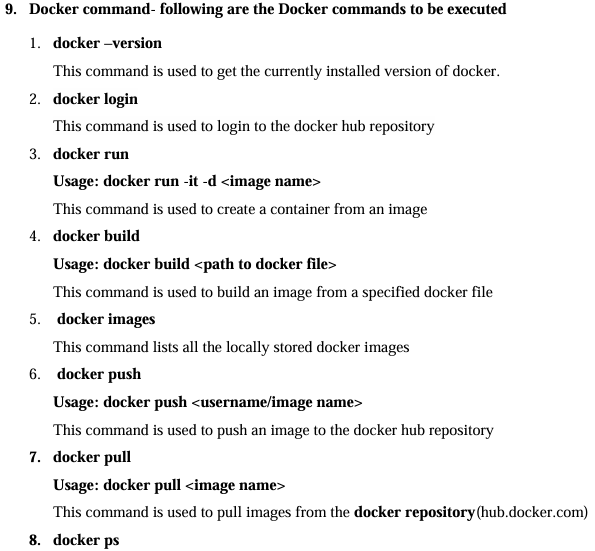
Docker Registry is where the Docker Images are stored.

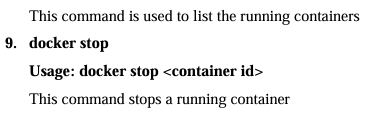
5. Docker Architecture?

Docker Architecture includes a Docker client – used to trigger Docker commands, a Docker Host – running the Docker Daemon and a Docker Registry – storing Docker Images. The Docker Daemon running within Docker Host is responsible for the images and containers.

6. Why Learn Docker?

Docker helps in consistent app deployment, simplifies environment setup, improves scalability, and speeds up development workflows.





7. What is a dockerfile?  
A Dockerfile is a text file with instructions to build a Docker image, specifying the base image, dependencies, and commands to run.

8. What is Docker hub?

Docker Hub is a cloud-based registry where Docker users can find, share, and store container images.

9. How do you create a docker container from an image?

You can create a Docker container from an image using the following command:  
docker run <image\_name>

10. Steps for Building an Image for a Sample Web Application on Docker Engine

1. Create a folder named "PHP Hello Docker" in Visual Studio Code.
2. Add a hello.php file with basic HTML/PHP code.
3. Create a Dockerfile with FROM php, COPY, EXPOSE, and CMD instructions.
4. Build the Docker image using docker build . -t dockerhubaccountname/foldername.
5. Run the container with docker run --name=php -p=3000:3000 dockerhubaccountname/foldername.
6. Access the web app at localhost:3000 in a browser.