Exercise 4

**Module name:**

Implementation of CICD with Java and open source stack Configure the web application and Version control using Git using Git commands and version control operations.

**Repository:**

* Centralised location to store a common data where sharing given to multiple people.
* Hosting service where we can save and share data across multiple users.

**Version Control System(VCS):**

* Version control systems are software that help track changes make in code over time.
* As a developer edits code, the version control system takes a snapshot of the files.
* It then saves that snapshot permanently so it can be recalled later if needed.
* Without version control, developers are tempted to keep multiple copies of code on their computer. This is dangerous because it's easy to change or delete a file in the wrong copy of code, potentially losing work.
* Version control systems solve this problem by managing all versions of the code, but presenting the team with a single version at a time.
* Git provides Version Control

**Git :**

* Git is a **distributed version control system** that allows developers to track versions of files and collaborate on software projects.
* It was originally developed by Linus Torvalds in 2005 and is widely used for both open source and commercial software development.
* GitHub is a developer platform that allows developers to create, store, manage and share their code.
* It uses Git software, providing the distributed version control of Git plus access control, bug tracking, software feature requests, task management, continuous integration, and wikis for every project.

**Git Installation:**

* Open your browser and and search for git download or go to [Git - Downloads (git-scm.com)](https://git-scm.com/downloads)
* Choose your operating system and then click on download
* Git is available in 2 types i.e, (CLI,GUI), we use Command line interface(CLI)
* After downloading, install it.

**Git Bash:**

Git is divided into:

1. Working directory
2. Staging area
3. Local Repository
4. Remote Repository

**Working directory:**

* The working directory in Git is the current state of your project files on your local machine.
* It contains the actual files that you are editing and working on, reflecting the most recent changes made.
* When you clone a repository, the working directory is created with the files from the remote repository. As you modify these files, Git tracks the changes, but they remain untracked until you add them to the staging area.
* The working directory allows you to make adjustments and test features before committing those changes to the local repository.
* This setup provides a dynamic environment for development.

**Staging Area:**

* The staging area in Git is a crucial intermediary space where changes to files are prepared before being committed to the repository.
* It allows you to selectively stage individual files or specific changes, enabling you to organize your commits meaningfully.
* By staging, you can review and control what gets included in your next commit, ensuring that only relevant updates are recorded.
* Once you're satisfied with the staged changes, you can move them to your local repository with git commit and then push them to the remote repository.
* This process helps maintain a clean and coherent project history, making collaboration more efficient.

**Local Repository:**

* A local repository in Git is a version of your project that exists on your own computer.
* It allows you to track changes, create branches, and manage version history independently from any remote repositories.
* When you clone a remote repository, you create a local copy that you can modify without affecting others.
* You can commit changes, switch branches, and test features locally before pushing your updates to the remote repository.
* This setup enables a flexible and efficient workflow for individual developers.

**Remote Repository**

* A remote repository in Git is a version of your project that is hosted on the internet or another network, allowing multiple users to collaborate.
* It serves as a central location where changes made by different contributors can be shared and managed.
* Users can clone the remote repository to their local machines, make changes, and then push those changes back to the remote repository.
* This facilitates version control and collaboration across teams and projects.

**Commit:**

* A commit in Git is a snapshot of your project's changes at a specific point in time, capturing the current state of the working directory and staging area.
* Used to move from staging area to Local repository

**Syntax:**

git commit -m "Your commit message here"

**Example:**

git commit -m "Fix bug in user authentication"

**Push:**

* In Git, a push is the process of transferring local commits from your local repository to a remote repository, making your changes available to others.
* This command updates the remote repository with your latest commits from the specified branch.
* Used to move from Local repository to Remote repository

**Syntax:**

git push <remote> <branch>

**Example:**

to push your changes to the main branch of the remote repository named origin,

”git push origin main”

**Git add:**

* This command adds a file to the staging area

Git add [file] —adds a single file

Git add \* – adds all files

**Steps:**

* Create a Directory
* Make the Directory Associated to Git using command “git init”
* Navigate Browser from google to [github.com](http://github.com) and create an account and an Repository
* It provide a static URL and it is considered as Remote

**Working directory (gitinit → create files)**

**|**

**|**

**Staging area (“Git Add” send required file only)**

**|**

**|**

**Local Repository (“Git Commit” the files which are in Local repository)**

**|**

**|**

**Remote Repository (“Git Push” the files which are in Local repository)**

**Example:**







