T.Y.B.Tech CSE 2023-24

Name : Parimal Shrikant Kolhe **Sem :** 5
Roll No : PB_01 **Panel :** B
Sub : FSD **Batch :** B1

FSD Laboratory 01

Problem Statement: Create a public git repository for your team and submit the repo URL as a solution to this assignment, Learn Git concept of Local and Remote Repository, Push, Pull, Merge and Branch.

Aim : Version control with Git.

Objectives:

1. Introduce the concepts and software behind version control, using the example of Git.
2. Understand the use of 'version control' in the context of a coding project.
3. Learn Git version control with Clone, commit to, and push, pull from a git repository.

Theory:

What is Git? What is Version Control?

- Git is a distributed version control system that allows you to track changes in your files and collaborate with others efficiently.
- Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later.

How to use Git for version controlling?

1. Initializing a Git Repository:
   - Use git init to create a new Git repository in a directory.
2. Cloning a Repository:
   - Use git clone <repository_url> to create a local copy of a remote Git repository.
3. Adding and Committing Changes:
   - Use git add <file> to stage changes for commit.
   - Use git commit -m "Commit message" to commit staged changes with a descriptive message.
4. Pushing and Pulling:
   - Use git push to upload your local commits to a remote repository.
   - Use git pull to fetch changes from a remote repository and merge them into your local branch.

5. Checking Status and History:
   - Use git status to see the status of your working directory.
   - Use git log to view the commit history of the current branch.

FAQ :

1. What is branching in Git?

Branching in Git is the practice of creating divergent lines of development. It allows you to work on new features or bug fixes without affecting the main codebase.

2. How to create and merge branches in Git? Write the commands used.

1) Creating a Branch:

Use git branch <branch_name> to create a new branch.

Use git checkout -b <branch_name> to create and switch to a new branch in one step.

2) Switching Between Branches:

Use git checkout <branch_name> to switch to an existing branch.

3) Merging Branches:

To merge changes from one branch into another, use git merge <branch_name>.

For example, if you're on the target branch, use git merge feature_branch to merge changes from feature_branch into the target branch.

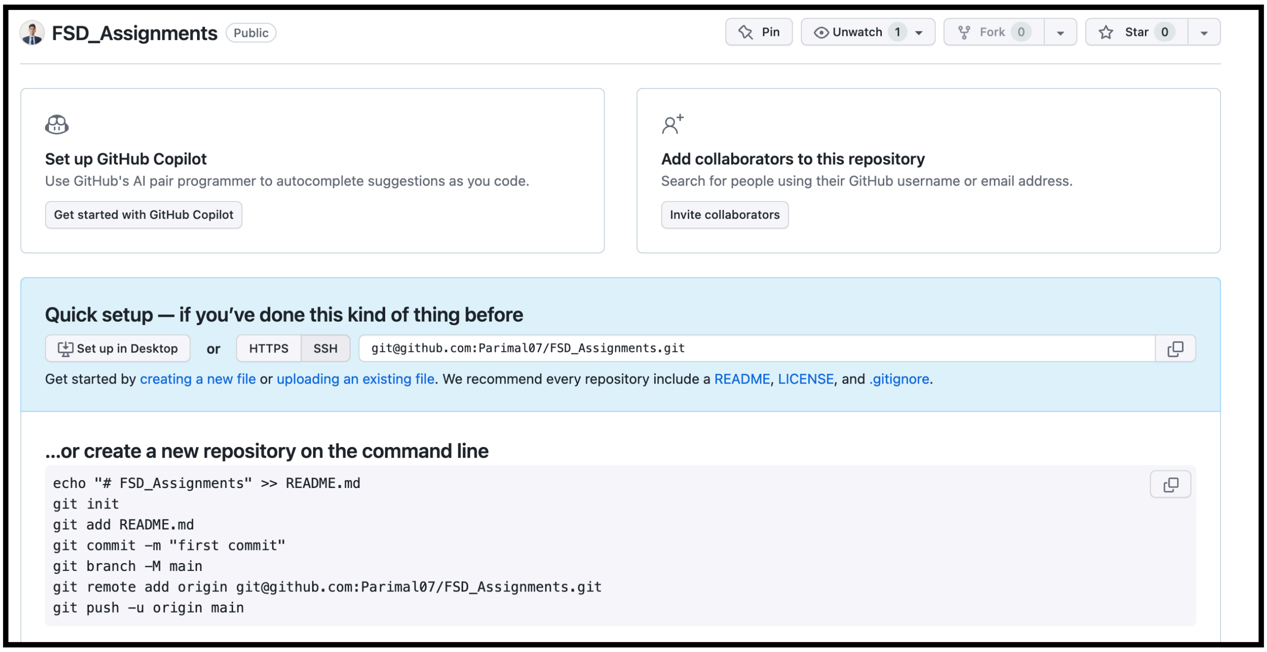
4) Resolving Conflicts:

If there are conflicts during a merge, Git will pause the process and allow you to manually resolve conflicts in the affected files.

After resolving conflicts, use git add <resolved_file> to stage the changes, then git commit to complete the merge.

Output :

Github Link - https://github.com/Parimal07/FSD_Assignments



Set up GitHub Copilot

Use GitHub's AI pair programmer to autocomplete suggestions as you code.

Get started with GitHub Copilot

Add collaborators to this repository

Search for people using their GitHub username or email address.

Invite collaborators

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH git@github.com:Parimal07/FSD_Assignments.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

```
echo "# FSD_Assignments" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin git@github.com:Parimal07/FSD_Assignments.git
git push -u origin main
```