

A
PROJECT
REPORT ON
Git Version Control Workflow Documentation

Project Overview

In this project, you will learn the basics of Git version control by creating and managing repositories locally and on GitHub. You will perform essential Git operations, including staging, committing, branching, and merging. By the end of this project, you will have a clear understanding of how to work with Git and GitHub to manage code collaboratively and efficiently.

Step-by-Step Process

1. Initialize Local Repository

Task: Create a local folder, add files, and initialize a Git repository.

1. Create a Local Folder

- Create a folder on your computer (e.g., GitProject) and add some files inside it (e.g., index.html, style.css).

2. Navigate to the Folder in Git Bash

Open Git Bash and use the following command to go to your folder:

```
cd desktop gitproject
```

3. Initialize a Git Repository

Convert the folder into a Git repository by running:

```
git init
```

2. Stage and Commit Changes

Task: Stage and commit the changes to your local repository.

1. Check the Current Status

See the status of the repository by running:

git status

2. Stage Files

Add the files to the staging area:

git add .

3. Commit the Changes

Commit the changes with a meaningful message:

git commit -m "Initial commit - Added basic project files"

3. Setup .gitignore

Task: Add files to the .gitignore to exclude unnecessary files.

1. Create .gitignore File

Create a .gitignore file in the root of your project folder and add files/folders you want to ignore (e.g., node_modules, .vscode, .gitconfig).

Example content for .gitignore:

txt

CopyEdit

.anaconda/

.conda/

.gitconfig

.ipynb_checkpoints/

.vscode/

node_modules/

*.log

2. Stage and Commit .gitignore

After creating the .gitignore, stage and commit it:

```
git add .gitignore
```

```
git commit -m "Added .gitignore to exclude unnecessary files"
```

4. Push to GitHub

Task: Push your local repository to GitHub.

- 1. Create a Repository on GitHub**

Go to [GitHub](https://github.com), log in, and create a new repository (e.g., git-project).

- 2. Link Local Repository to GitHub**

Run the following command to link your local repository to GitHub:

```
git remote add origin https://github.com/Ejalandhar/git-project.git
```

- 3. Push to GitHub**

Push the changes to GitHub's main branch:

```
git branch -M main
```

```
git push -u origin main
```

5. Create and Work on a New Branch

Task: Create a develop branch, make changes, and push it to GitHub.

- 1. Create a New Branch**

Create and switch to a new branch named develop:

```
git checkout -b develop
```

- 2. Make Changes**

Edit files (e.g., index.html) and save the changes.

- 3. Stage and Commit Changes**

Stage and commit your changes:

git add .

git commit -m "Updated index.html in develop branch"

4. Push the develop Branch to GitHub

Push the develop branch to GitHub:

git push -u origin develop

6. Merge Branches

Task: Merge the develop branch into the main branch.

1. Switch to main Branch

Switch to the main branch:

git checkout main

2. Merge develop into main

Merge the develop branch into the main branch:

git merge develop

3. Push the Updated main Branch

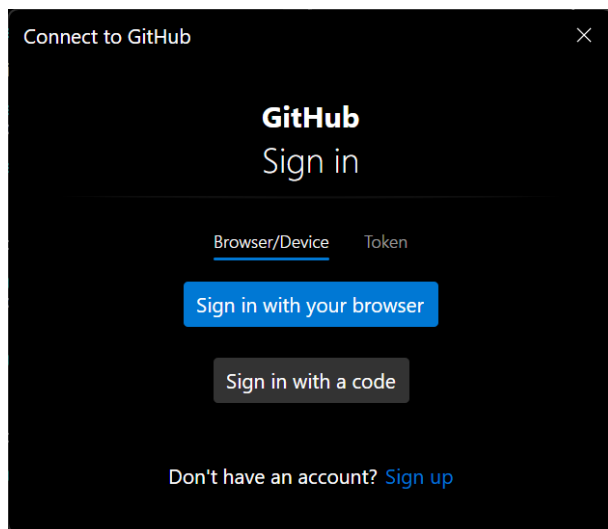
Push the merged main branch to GitHub:

git push origin main

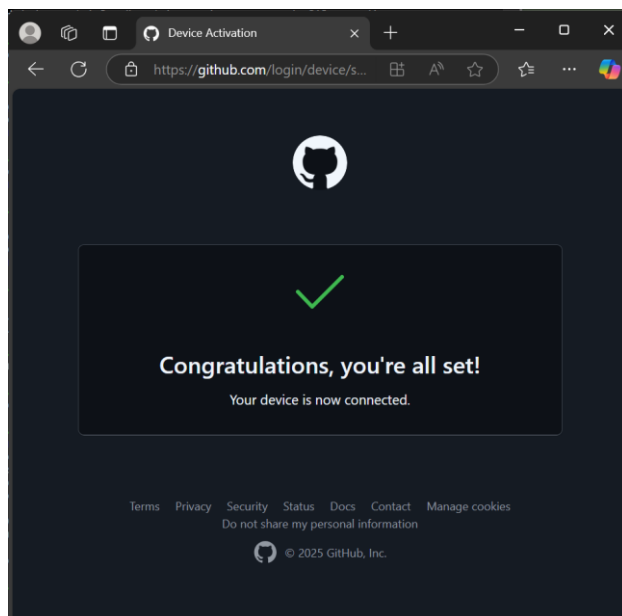
Output:-

GitHub Repository

- Your GitHub repository should now contain two branches: main and develop.
 - Repository URL: <https://github.com/Ejalandhar/git-project>



Connect to git hub



Git bash :-

```

DELL@Ejjigiri-Jalandhar MINGW64 ~ (master)
$ cd desktop

DELL@Ejjigiri-Jalandhar MINGW64 ~/desktop (master)
$ cd gitproject
bash: cd: gitproject: No such file or directory

DELL@Ejjigiri-Jalandhar MINGW64 ~/desktop (master)
$ cd Gitproject
bash: cd: Gitproject: No such file or directory

DELL@Ejjigiri-Jalandhar MINGW64 ~/desktop (master)
$ cd gitproject

DELL@Ejjigiri-Jalandhar MINGW64 ~/desktop/gitproject (master)
$ git init
Initialized empty Git repository in C:/Users/DELL/Desktop/gitproject/.git/

DELL@Ejjigiri-Jalandhar MINGW64 ~/desktop/gitproject (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        index.html
        style.css

nothing added to commit but untracked files present (use "git add" to track)

DELL@Ejjigiri-Jalandhar MINGW64 ~/desktop/gitproject (master)
$ git add .

DELL@Ejjigiri-Jalandhar MINGW64 ~/desktop/gitproject (master)
$ git commit -m "Initial commit - Added basic project files"
[master (root-commit) afdc777] Initial commit - Added basic project files
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 index.html
 create mode 100644 style.css

DELL@Ejjigiri-Jalandhar MINGW64 ~/desktop/gitproject (master)
$ node_modules/
*.log
secret.txt
bash: node_modules/: No such file or directory

```

```

bash: *.log: command not found
bash: secret.txt: command not found

DELL@Ejjigiri-Jalandhar MINGW64 ~/desktop/gitproject (master)
$ git add .gitignore
git commit -m "Added .gitignore to exclude unnecessary files"
fatal: pathspec '.gitignore' did not match any files
On branch master
nothing to commit, working tree clean

DELL@Ejjigiri-Jalandhar MINGW64 ~/desktop/gitproject (master)
$ https://github.com/Ejalandhar/git-project
bash: https://github.com/Ejalandhar/git-project: No such file or directory

DELL@Ejjigiri-Jalandhar MINGW64 ~/desktop/gitproject (master)
$ git remote add origin https://github.com/Ejalandhar/git-project-

DELL@Ejjigiri-Jalandhar MINGW64 ~/desktop/gitproject (master)
$ git branch -M main
git push -u origin main
remote: Repository not found.
fatal: repository 'https://github.com/Ejalandhar/git-project-/' not found

DELL@Ejjigiri-Jalandhar MINGW64 ~/desktop/gitproject (main)
$ git remote add origin https://github.com/Ejalandhar/git-project
error: remote origin already exists.

DELL@Ejjigiri-Jalandhar MINGW64 ~/desktop/gitproject (main)
$ git branch -M main
git push -u origin main
remote: Repository not found.
fatal: repository 'https://github.com/Ejalandhar/git-project-/' not found

DELL@Ejjigiri-Jalandhar MINGW64 ~/desktop/gitproject (main)
$ git remote -v
origin https://github.com/Ejalandhar/git-project- (fetch)
origin https://github.com/Ejalandhar/git-project- (push)

DELL@Ejjigiri-Jalandhar MINGW64 ~/desktop/gitproject (main)
$ git remote remove origin

DELL@Ejjigiri-Jalandhar MINGW64 ~/desktop/gitproject (main)
$ git remote add origin https://github.com/Ejalandhar/git-project.git

```

```

DELL@Ejjigiri-Jalandhar MINGW64 ~/desktop/gitproject (main)
$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 251 bytes | 251.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Ejalandhar/git-project.git
 * [new branch]    main -> main
branch 'main' set up to track 'origin/main'.

DELL@Ejjigiri-Jalandhar MINGW64 ~/desktop/gitproject (main)
$ git branch -M main
git push -u origin main
branch 'main' set up to track 'origin/main'.
Everything up-to-date

DELL@Ejjigiri-Jalandhar MINGW64 ~/desktop/gitproject (main)
$ git checkout -b develop
Switched to a new branch 'develop'

DELL@Ejjigiri-Jalandhar MINGW64 ~/desktop/gitproject (develop)
$ git add .
git commit -m "Updated index.html in develop branch"
On branch develop
nothing to commit, working tree clean

DELL@Ejjigiri-Jalandhar MINGW64 ~/desktop/gitproject (develop)
$ git push -u origin develop
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:   https://github.com/Ejalandhar/git-project/pull/new/develop
remote:
To https://github.com/Ejalandhar/git-project.git
 * [new branch]    develop -> develop
branch 'develop' set up to track 'origin/develop'.

DELL@Ejjigiri-Jalandhar MINGW64 ~/desktop/gitproject (develop)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

DELL@Ejjigiri-Jalandhar MINGW64 ~/desktop/gitproject (main)

```

```

DELL@Ejjigiri-Jalandhar MINGW64 ~/desktop/gitproject (develop)
$ git push -u origin develop
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:   https://github.com/Ejalandhar/git-project/pull/new/develop
remote:
To https://github.com/Ejalandhar/git-project.git
 * [new branch]    develop -> develop
branch 'develop' set up to track 'origin/develop'.

DELL@Ejjigiri-Jalandhar MINGW64 ~/desktop/gitproject (develop)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

DELL@Ejjigiri-Jalandhar MINGW64 ~/desktop/gitproject (main)
$ git merge develop
Already up to date.

DELL@Ejjigiri-Jalandhar MINGW64 ~/desktop/gitproject (main)
$ git push origin main
Everything up-to-date

```

Conclusion

By following these steps, you will successfully create and manage a Git repository, push it to GitHub, and work with branching and merging. This will help you understand the basics of Git and GitHub for version control and collaborative coding.