

Git Project Documentation

Project Title:

Learning Git Version Control

Objective:

To understand and practice the basics of Git and GitHub, including repository creation, essential Git operations (staging, committing, branching, merging), and collaboration with GitHub.

Tools Used

- **Git Bash**
- **GitHub**

Tasks Performed

1. Initialize a Local Repository

1. Created a local folder Mygitproject.
2. Initialized Git in the folder:

`git init`

3. Added a README file:

`echo "# My Git Project" > README.md`

2. Stage and Commit Changes

1. Staged all changes:

`git add .`

2. Committed the changes:

```
git commit -m "Initial commit with README.md"
```

3. Configure GitHub Remote

1. Checked the existing remote (if any):

```
git remote -v
```

2. Removed and re-added the origin remote (if necessary):

```
git remote remove origin
```

```
git remote add origin https://github.com/venky7234/My-Git-Project.git
```

4. Push to GitHub

1. Pushed the main branch to GitHub:

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

2. Resolved the error ! [rejected]:

- Pulled remote changes with rebase:

```
git pull origin main --rebase
```

- Resolved conflicts (if any) and pushed again:

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

5. Create and Work on a New Branch

1. Created a branch develop:

```
git branch develop
```

```
git checkout develop
```

2. Made updates to the project in the develop branch and committed changes:

`git add .`

`git commit -m "Updated files in the develop branch"`

3. Pushed the develop branch to GitHub:

`git push -u origin develop`

6. Merge Branches

1. Merged the develop branch into the main branch:

`git checkout main`

`git merge develop`

2. Pushed the updated main branch to GitHub:

`git push origin main`

7. Git Ignore Setup

1. Created a .gitignore file to exclude unnecessary files and folders:

plaintext

*.log

node_modules/

.vscode/

2. Staged and committed the .gitignore file:

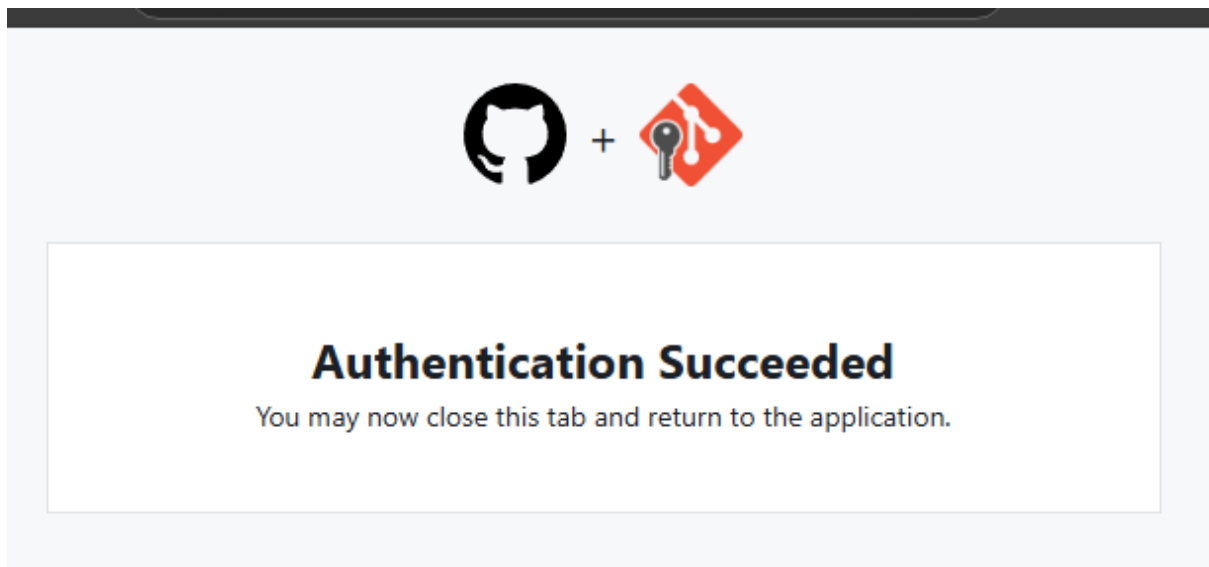
`git add .gitignore`

`git commit -m "Added .gitignore file"`

GitHub Repository

- **Repository Name:** <https://github.com/venky7234/My-Git-Project.git>
-
- **Branches:**
 - main
 - develop

Screenshots



```
hp@DESKTOP-1MAPRG3 MINGW64 ~/desktop/Mygitproject (main)
$ git branch develop

hp@DESKTOP-1MAPRG3 MINGW64 ~/desktop/Mygitproject (main)
$ git checkout develop
Switched to branch 'develop'

hp@DESKTOP-1MAPRG3 MINGW64 ~/desktop/Mygitproject (develop)
$ git add .
git commit -m "Update files in develop branch"
On branch develop
nothing to commit, working tree clean

hp@DESKTOP-1MAPRG3 MINGW64 ~/desktop/Mygitproject (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/venky7234/My-Git-Project/pull/new/develop
remote:
To https://github.com/venky7234/My-Git-Project.git
 * [new branch]      develop -> develop
branch 'develop' set up to track 'origin/develop'.

hp@DESKTOP-1MAPRG3 MINGW64 ~/desktop/Mygitproject (develop)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

hp@DESKTOP-1MAPRG3 MINGW64 ~/desktop/Mygitproject (main)
$ git push origin main
Everything up-to-date

hp@DESKTOP-1MAPRG3 MINGW64 ~/desktop/Mygitproject (main)
$ git branch -r
  origin/develop
  origin/main

hp@DESKTOP-1MAPRG3 MINGW64 ~/desktop/Mygitproject (main)
$
```

```
hp@DESKTOP-1MAPRG3 MINGW64 ~/desktop/Mygitproject (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/venky7234/My-Git-Project/pull/new/develop
remote:
To https://github.com/venky7234/My-Git-Project.git
 * [new branch]      develop -> develop
branch 'develop' set up to track 'origin/develop'.

hp@DESKTOP-1MAPRG3 MINGW64 ~/desktop/Mygitproject (develop)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

hp@DESKTOP-1MAPRG3 MINGW64 ~/desktop/Mygitproject (main)
$ git push origin main
Everything up-to-date

hp@DESKTOP-1MAPRG3 MINGW64 ~/desktop/Mygitproject (main)
$ git branch -r
  origin/develop
  origin/main

hp@DESKTOP-1MAPRG3 MINGW64 ~/desktop/Mygitproject (main)
$ git pull origin main --allow-unrelated-histories
From https://github.com/venky7234/My-Git-Project
 * branch      main      -> FETCH_HEAD
Already up to date.

hp@DESKTOP-1MAPRG3 MINGW64 ~/desktop/Mygitproject (main)
$ git add .
git commit -m "Resolve merge conflicts"
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

hp@DESKTOP-1MAPRG3 MINGW64 ~/desktop/Mygitproject (main)
$ git push -u origin main
branch 'main' set up to track 'origin/main'.
Everything up-to-date
```

Learnings

- How to initialize a Git repository locally.
- How to stage, commit, and push changes to GitHub.
- How to resolve common Git errors (e.g., rejected pushes).
- Working with branches for feature development and merging changes.
- Using .gitignore to exclude unnecessary files.