**PURBANCHAL UNIVERSITY**

****

**KHWOPA ENGINEERING COLLEGE**

**LIBALI-08, BHAKTAPUR**

LAB REPORT ON .NET

LAB NO. 01

**SUBMITTED BY: SUBMITTED TO:**

Name: Yogyata Neupane Department of Computer Engineering

Roll No. : 770347

Submission: 2081/12/08

**Theory:**

### **Git:**

### Git is a **distributed version control system** that helps track modifications in source code throughout the software development process. It enables multiple developers to collaborate efficiently by managing various versions of a project.

With Git, developers can create branches, merge updates, and revert changes, making code management more streamlined. It is extensively used in both **open-source and commercial** software projects. Platforms like **GitHub, GitLab, and Bitbucket** offer remote repositories for seamless collaboration.

### **2. GitHub:**

GitHub is an **online platform** that leverages Git for **version control and team collaboration**. It provides developers with a space to store, organize, and share code repositories effectively.

Key features of GitHub include **branching, pull requests, issue tracking, and CI/CD integration**, making development workflows more efficient. It is widely adopted for both **open-source and private projects**, promoting effortless teamwork. Additionally, GitHub offers **cloud-based hosting**, allowing access to repositories from any location.

**General Git and GitHub Commands:**

**Git Configuration**

*git config --global user.name “Your Name”*

This command sets the global username for the Git commits.

*git config --global user.email “your\_email@example.com”*

This command sets the global email associated with Git commits.

**Initializing**

*git init*

initializes a new Git repository in the current directory.

**Staging and Commits**

*git add .*

It stages all changes and new files for commit.

*git commit -m “Your commit message”*

Saves the staged changes with a descriptive message.

**Branching and Merging**

*git branch*Lists all the branches in the repository.

*git branch <branch\_name>*

Creates a new branch for separate development.

*git checkout <branch\_name> / Git switch <branch\_name>*

Switches to the specified branch

*git merge <branch\_name>*

Merges changes from the specified branch into the current branch.

**Pushing and Pulling**

*git push -u origin <branch\_name>*

Uploads the local changes to the remote repository.

*git pull origin <branch\_name>*

Fetches and merge the latest changes from the remote repository.

**Status and Logs**

*git status*

Show the current state of the files in the working directory (modified, staged or untracked).

*git log*

Displays the commit history of the repository.

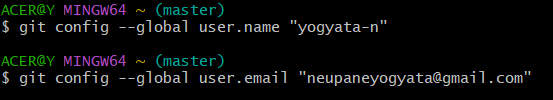
**GitHub Specific**

*git remote add origin <repo\_url>*

Links the local repository to a remote repository on GitHub.

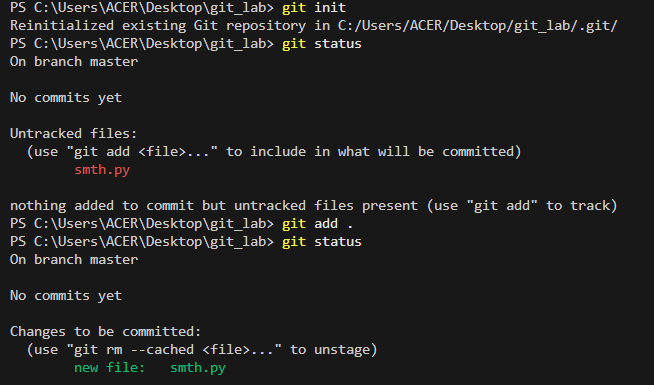
**Lab Works**

First set the global username and email of the GitHub.

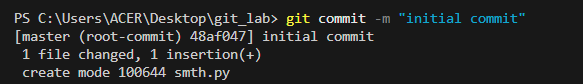


Create a folder and inside it files as per the user desire so that we can identify the changes inside the file using the version control (Git).

On creating the new files, initially the files are in the untracked stage so sent the untracked files to the staging stage. To do so first initialize the directory and staged the files.

**

Now commit the files such that the files are stored in the local repository.

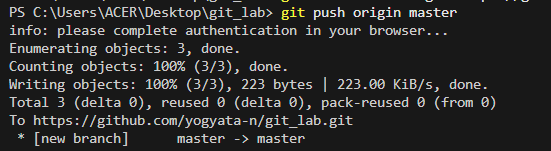


After changing the contents in the file **“test.py”** add the file and commit it.

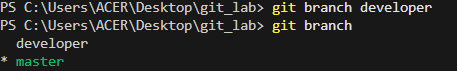
All of these files are saved in the local repository. Now to add these files in the remote repository create the repository in the GitHub and copy the url of the repo and use the following code.

Screenshot 2025-03-22 214549

Now push the files in the repository created.

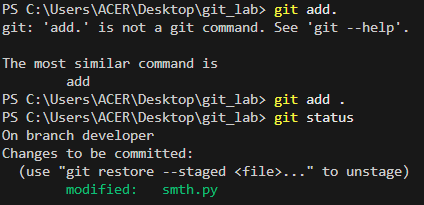


Now creating branches, allowing the work on different version of a project without affecting the main codebase.

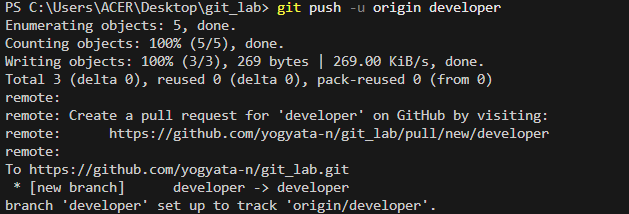


Moving on to the recently created branch to modify the contents in the file without affecting the main codebase.

Screenshot 2025-03-22 214726

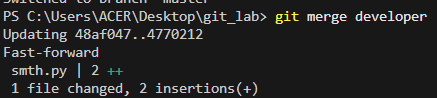
Screenshot 2025-03-22 214726

To change the branch, we can use the command *“git switch developer”*. To make sure the branch is visible to other users of the repository push the branch in the GitHub.

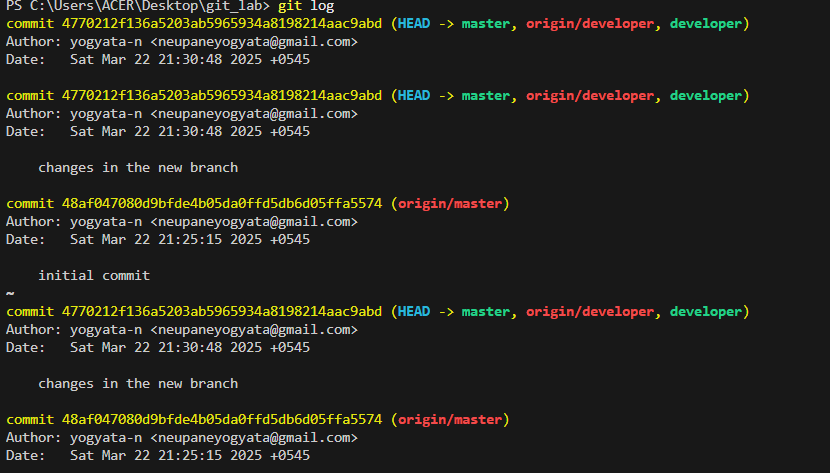


Merging the branches such that the changes in the new branch or new features added in the new branch is added to

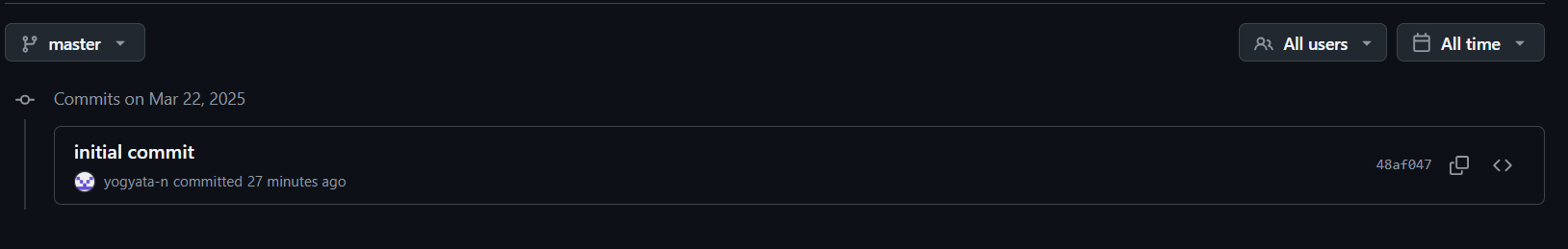
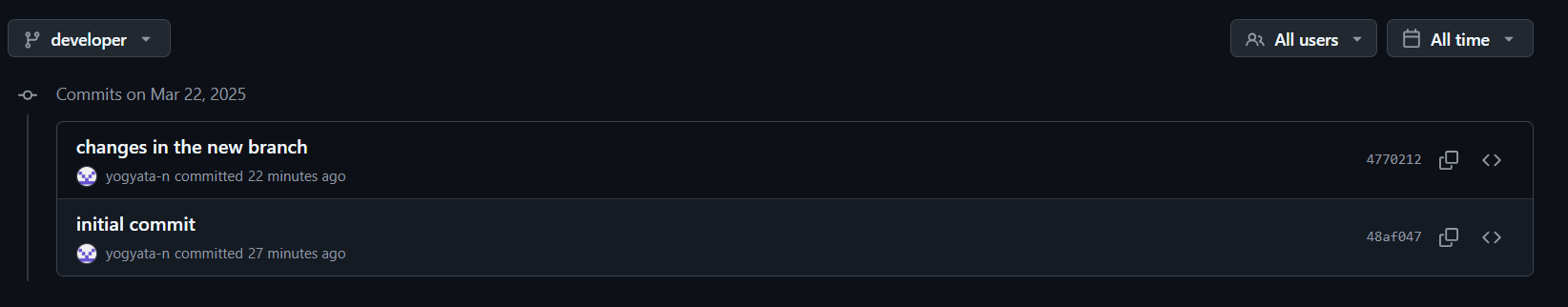
the main code base.



To check the commits performed in the past



Merging the branch in the GUI GitHub (Web)



**Conclusion:**

In this lab, we learn about the basics of the Git and GitHub. We perform initialization, branching, merging, pushing and commit.