Subject Name: **Source Code Management**

Subject Code: **24CSE0106**

Cluster: **Alpha**

Department: **DCSE**

Submitted By:
Kiratbir Singh
2410990383
G5

Submitted To:
Dr. Chetna Sharma

Department of Computer Science & Engineering
Chitkara University Institute of
Engineering and Technology, Rajpura,
Patiala, Punjab

Index Task 1.1#

Practical No. : 1
Setting Up Git Client

Aim: To install and configure Git Client on your local system.

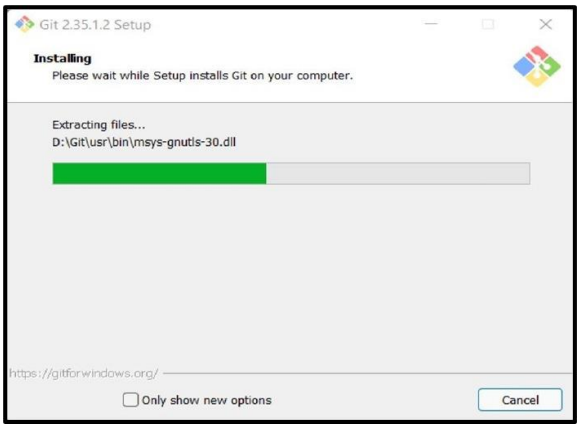
Theory: Git is a distributed version control system used to track changes in source code. Created by Linus Torvalds in 2005, it allows developers to work offline, create branches for new features, and merge them into the main codebase, facilitating collaborative work. Each developer has a local repository with a complete project history.

Git also supports collaboration through remote repositories, enabling multiple developers to work on the same project simultaneously. Its powerful features make Git an indispensable tool for modern software development, ensuring efficient and reliable version control.

Procedure:

1. Download Git from git-scm.com.
2. Install Git by following the setup wizard.
3. Open Git Bash and verify installation using the command: git --version.
4. Configure user details using the commands:.
5. git config --global user.name "Your Name".
6. git config --global user.email "Your Email".

Screenshots:



- This image guides you through the steps to install Git Bash, an essential tool for efficient version control and seamless collaboration.
- Follow these instructions to get Git Bash up and running, making sure you're ready to streamline your development workflow.
- With Git Bash installed, you're all set to manage your projects effectively and collaborate with others.

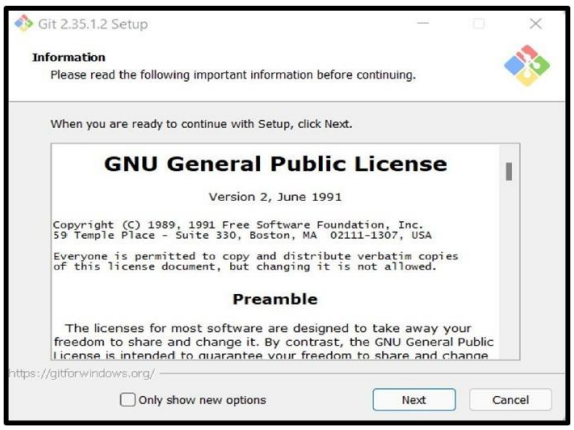


Fig: Git installation wizard

- The GNU General Public License (GPL) for Git Bash ensures that the software is free to use, modify, and distribute, promoting software freedom and collaboration among developers.
- It requires that any modified versions of the software also be distributed under the same GPL license, ensuring the source code remains available to everyone.

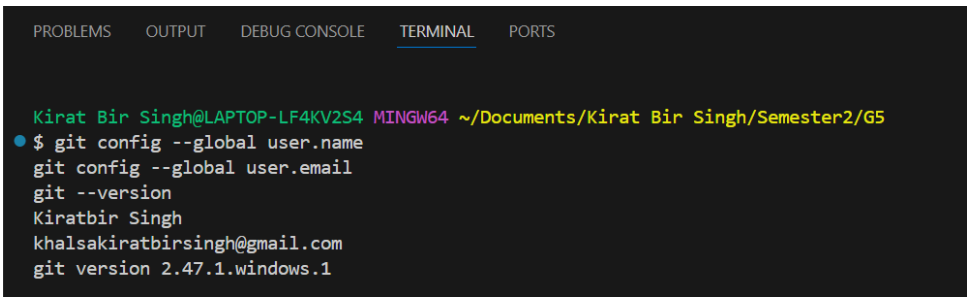


Fig: git --version in Git Bash

- Set your Git username with the command git config --global user.name "Your Name".
- Configure your Git email address using git config --global user.email "youremail@example.com".
- To check the installed Git version, use the command git –version.

Practical No. : 2

Setting up GitHub Account and Adding Collaborators on GitHub Repository

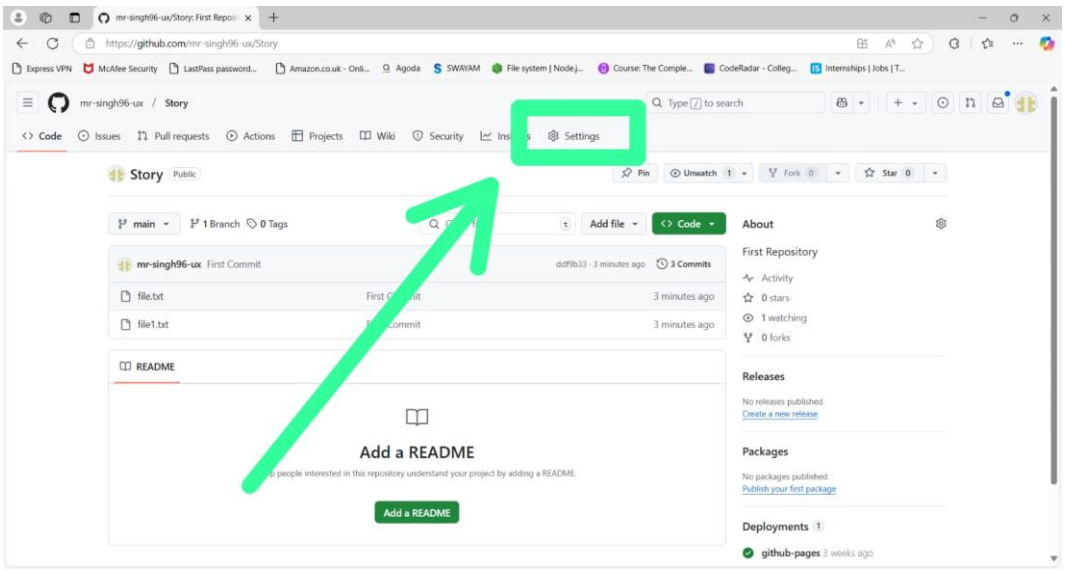
Aim: To add collaborators to a GitHub repository for collaborative work.

Theory: Collaborators are individuals with write access to a repository. They can contribute to the project by pushing changes and merging pull requests. They are essential for teamwork and project development, enabling multiple people to work on the same codebase simultaneously. Effective collaboration ensures that projects progress smoothly and efficiently.

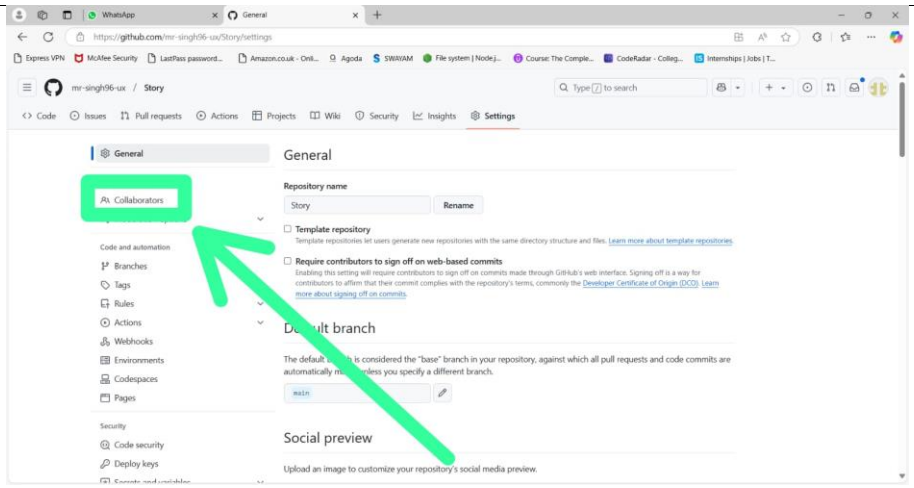
Procedure:

1. Log in to your GitHub account and create a new repository.
2. Navigate to Settings > Manage Access in the repository.
3. Add collaborators by their GitHub usernames.
4. Collaborators will receive an invitation email, which they must accept.

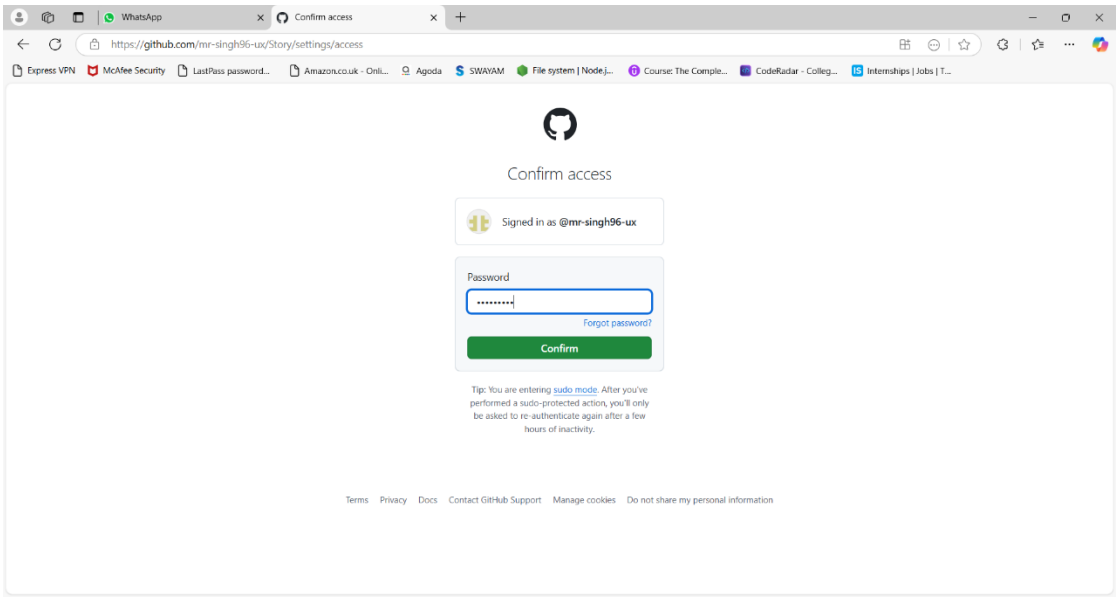
Screenshots:



- Navigate to the main page of your repository on GitHub and click on the "Settings" tab.



- In the "Access" section of the sidebar, click "Collaborators" and then "Add people."



- Before that add password of your GitHub account.

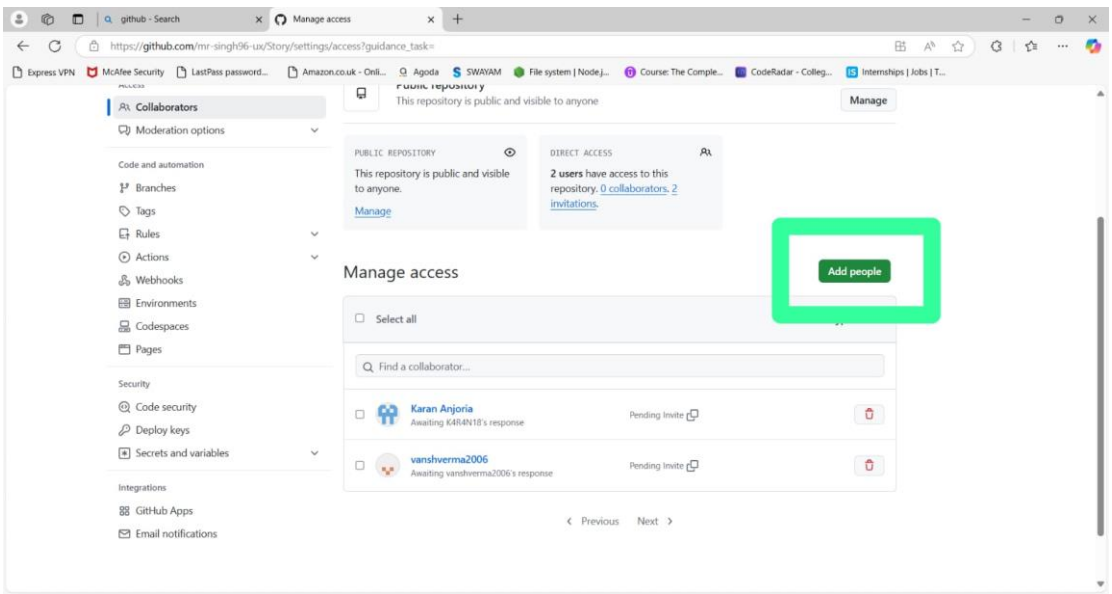


Fig: Collaborator Invitation

- Enter the username or email of the person you want to invite, and click "Add NAME to REPOSITORY."
- The user will receive an email invitation to collaborate on your repository. Once they accept, they will have collaborator access.

Practical No. : 3
Merging Two Branches

Aim: To merge two branches within a Git repository.

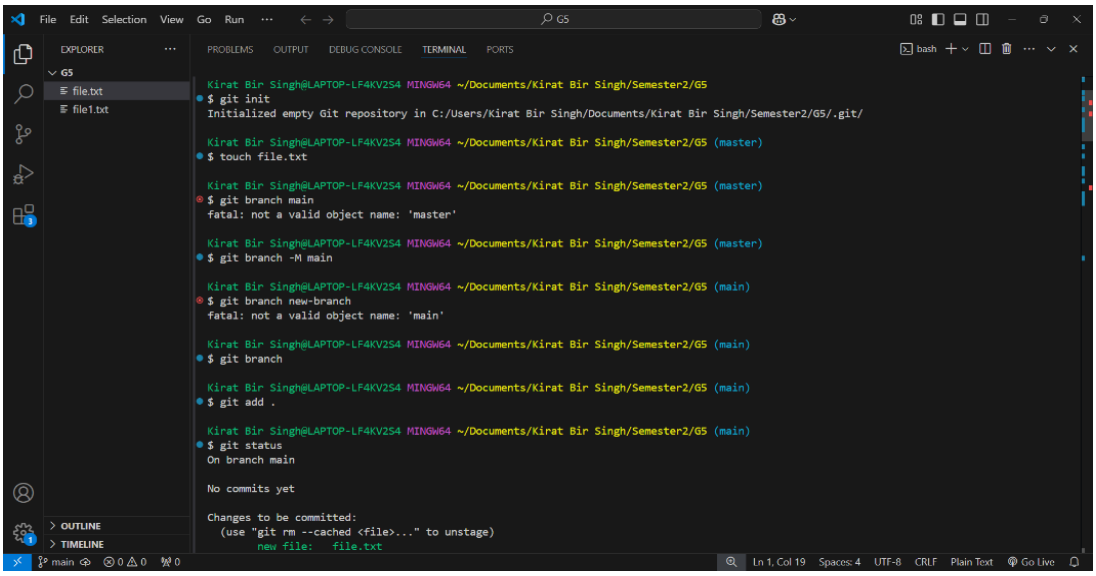
Theory: Merging branches in Git allows you to combine changes from one branch into another. It is a fundamental process in collaborative workflows, ensuring all contributions are integrated into a single codebase.

This practice helps prevent conflicts and ensures that all team members are working with the most up-to-date code. Regular merging can also make it easier to track changes and maintain a clean project history.

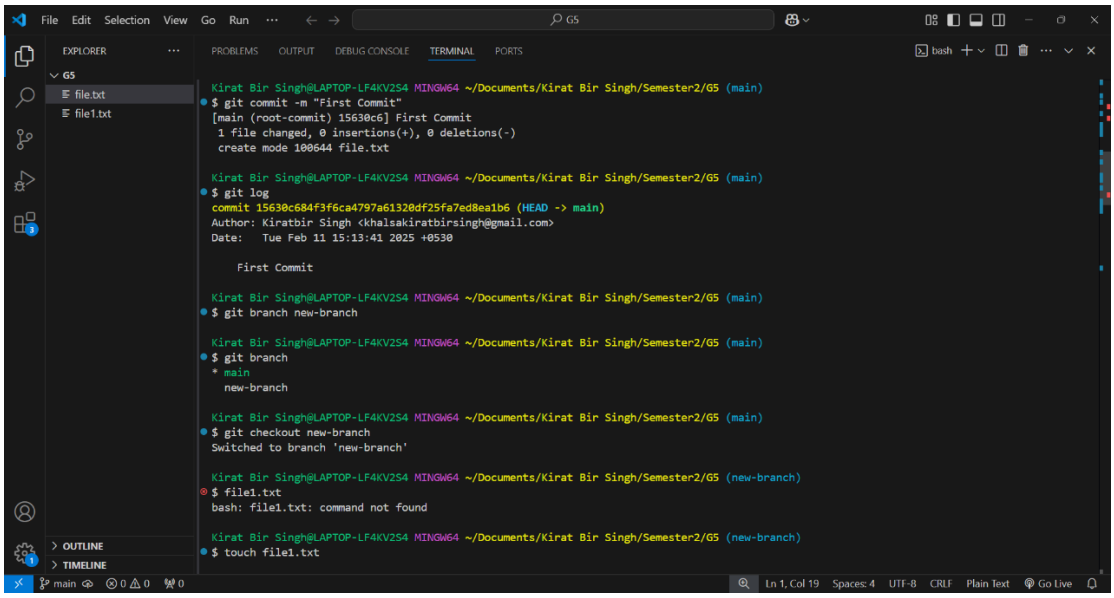
Procedure:

- Create a new branch and switch to it:
  git checkout -b new-branch
- Make changes to a file in the new branch and commit them:
  echo "New content" > file.txt
  git add file.txt
  git commit -m "Add changes in new branch"
- Switch back to the main branch:
  git checkout main
- Modify another file in the main branch and commit the changes:
  echo "Main branch changes" > another-file.txt
  git add another-file.txt
  git commit -m "Modify file in main branch"
- Merge the new branch into the main branch:
  git merge new-branch

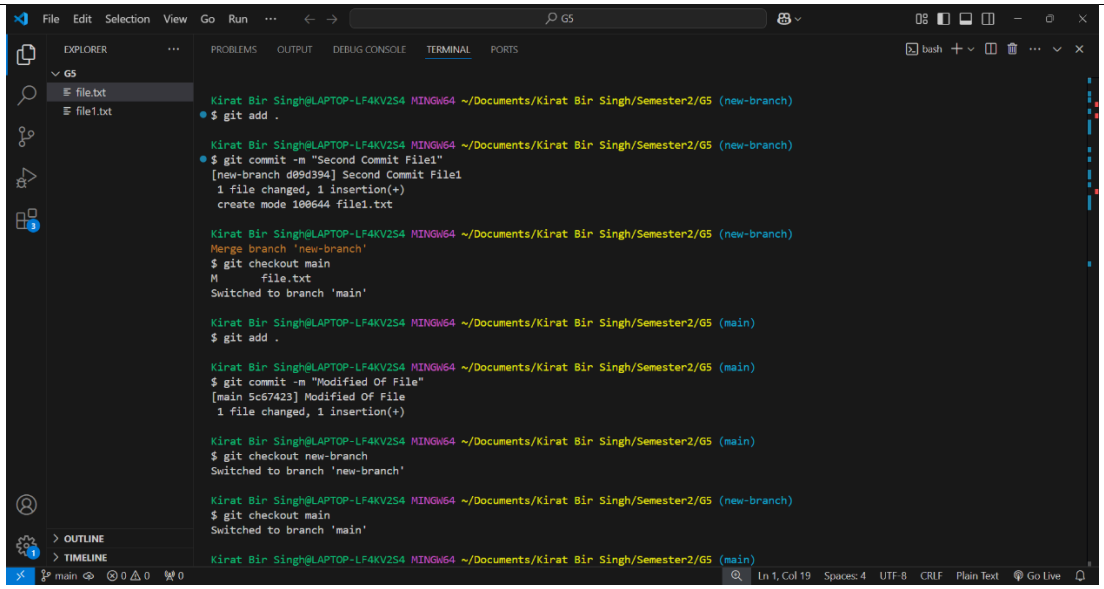
Screenshots:



- Initializes a new Git repository using “**git init**”.
- Create an empty file named “file.txt” using “**touch file.txt**”.
- Rename the master branch by “**git branch -M main**”.
- Stages all changes for the next commit using “**git add .**”.



- Shows the working directory status.
- Commits the staged changes with the message "First commit".
- Displays the commit history.
- Creates a new branch named new-branch.



- Lists all branches in the repository using "**git branch**".
- Switches to the "**new-branch**" branch.
- Creates a new empty file named file1.txt.
- Stages all changes in the working directory.
- Commits the staged changes with the message "Second commit".

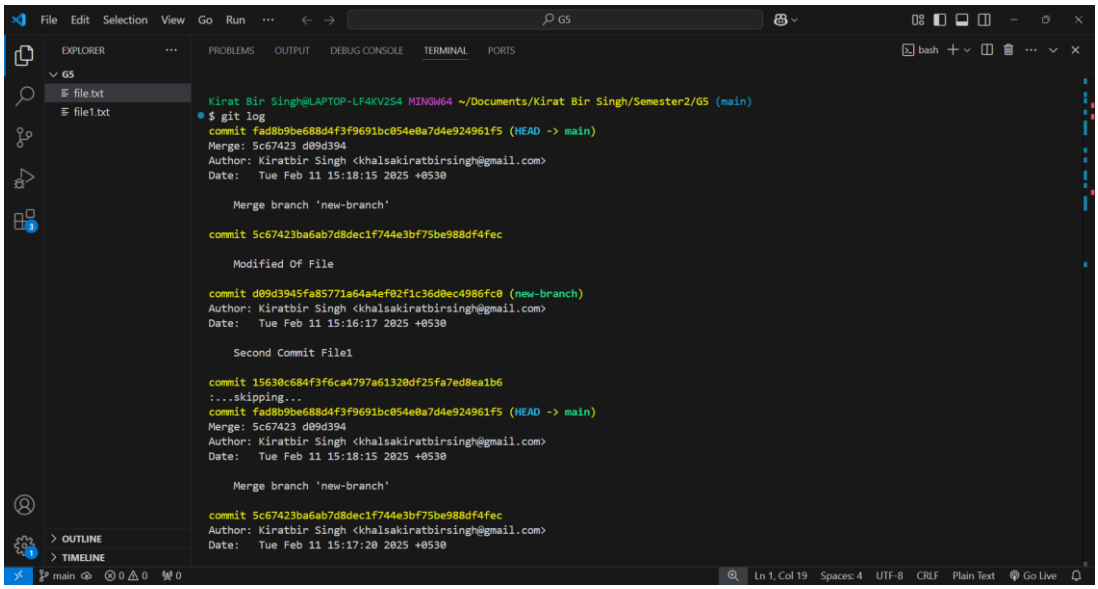


Fig: Merging of Branches

- Switches to the main branch.
- Merges the new-branch into the main branch using "**git merge new-branch**".

Practical No. : 4
Push/Pull Using Git

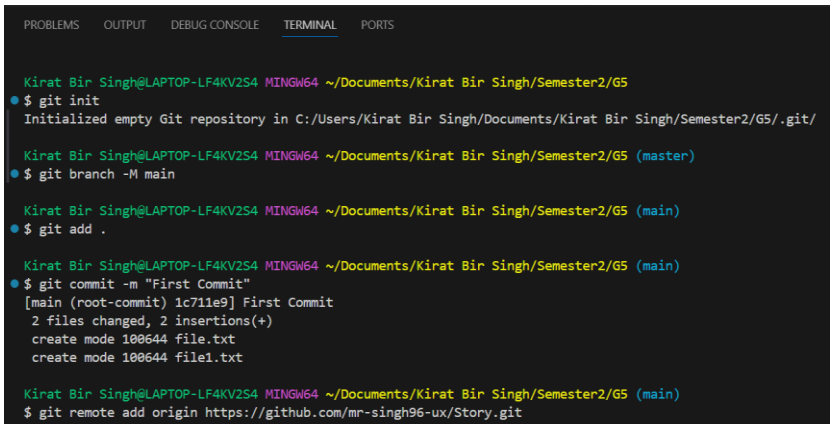
Aim: To demonstrate push and pull operations in Git.

Theory: Push transfers committed changes from the local repository to the remote repository, while pull retrieves updates from the remote repository. This process ensures that the local and remote repositories are synchronized, allowing team members to access the latest code changes. Regularly using push and pull commands helps maintain consistency and collaboration within a project.

Procedure:

- Make changes in the local repository and commit them.
- Push the changes to the remote repository using git push.
- Make changes directly on the remote repository (e.g., via GitHub interface).
- Pull the changes to the local repository using git pull.

Screenshots:



- After initializing the git, commit all the changes to the local repository by using "**git commit -m "First Commit**".

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Kirat Bir Singh@LAPTOP-LF4KV2S4 MINGW64 ~/Documents/Kirat Bir Singh/Semester2/G5 (main)
$ git pull origin main --rebase
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 10 (delta 0), reused 10 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (10/10), 10.00 KiB | 200.00 KiB/s, done.
From https://github.com/mr-singh96-ux/Story
 * branch            main       -> FETCH_HEAD
 * [new branch]      main       -> origin/main
Successfully rebased and updated refs/heads/main.

Kirat Bir Singh@LAPTOP-LF4KV2S4 MINGW64 ~/Documents/Kirat Bir Singh/Semester2/G5 (main)
$ git push -u origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 335 bytes | 167.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/mr-singh96-ux/Story.git
   f645e9b..ddf9b33  main -> main
branch 'main' set up to track 'origin/main'.
```

- Adds a remote repository with the URL https://github.com/mr-singh96-ux/Story.git.
    - Pulls the latest changes from the remote repository's main branch and rebases your local changes on top of them (if the repository is already created).
    - Then, pushes your local commits to the remote repository's main branch.

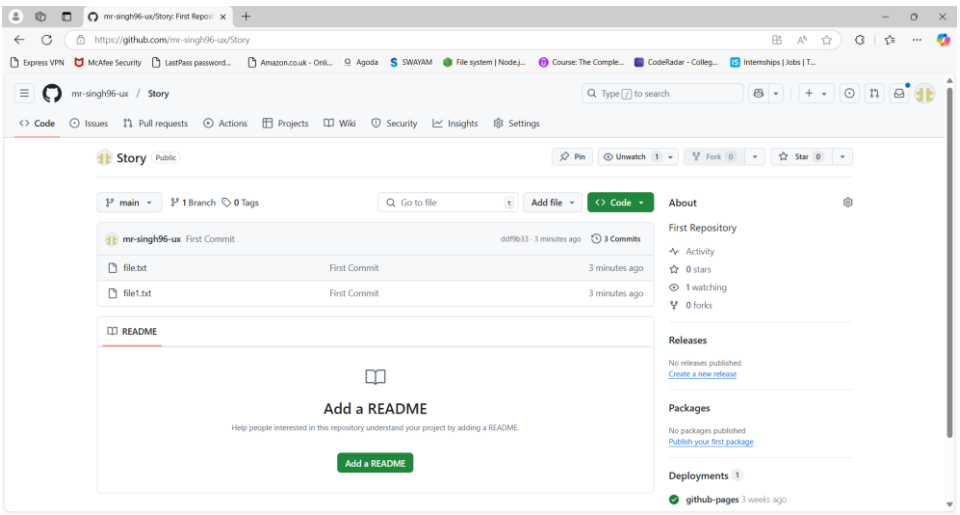


Fig: Push/Pull using git