

2. Demonstrate to use of Version Control System (Git offline and connect to github account).

Create multiple users and usage with team leader role and coder roles.

Demonstrate code pull and push, fork (branching) amongst the users

Compare it with hg. (on answer sheet)

Procedure:

Create 3 users - 1 leader, 2 coders (user1 and user2)

The leader creates GitHub repository and adds 2 collaborators (coders).

Rules are created for coders by leader e.g. coder cannot directly push to main branch without Pull Request.

To do this, log onto leader account.

Go to Settings -> Rules -> Ruleset

Create new rule

The screenshot shows the GitHub Settings interface for a repository named 'aaditya-user3 / test-project'. The 'Rulesets / push' tab is selected. A new ruleset is being created with the name 'push'. The 'Enforcement status' is set to 'Disabled'. The 'Bypass list' section indicates that the list is empty. The 'Targets' section shows a placeholder for selecting branches. The sidebar on the left lists various settings categories like General, Access, Code and automation, and Security.

Set target branch -> by pattern .e.g main.

Targets

Which branches do you want to make a ruleset for?

Target branches

Branch targeting criteria

Add target

main

Applies to 1 target: main.

Rules

Which rules should be applied?

Branch rules

Restrict creations
Only allow users with bypass permission to create matching refs.

Restrict updates
Only allow users with bypass permission to update matching refs.

Restrict deletions
Only allow users with bypass permissions to delete matching refs.

Require linear history
Prevent merge commits from being pushed to matching refs.

Add rules e.g. Require a pull request before merging
Save changes.

Workflow:

The leader creates a main branch and adds some files. E.g. file.txt

test-project Public

Pin Unwatch 1

main 1 Branch 0 Tags

Go to file Add file Code

aaditya-user3 Create file.txt 78294de · 5 minutes ago 1 Commit

file.txt Create file.txt 5 minutes ago

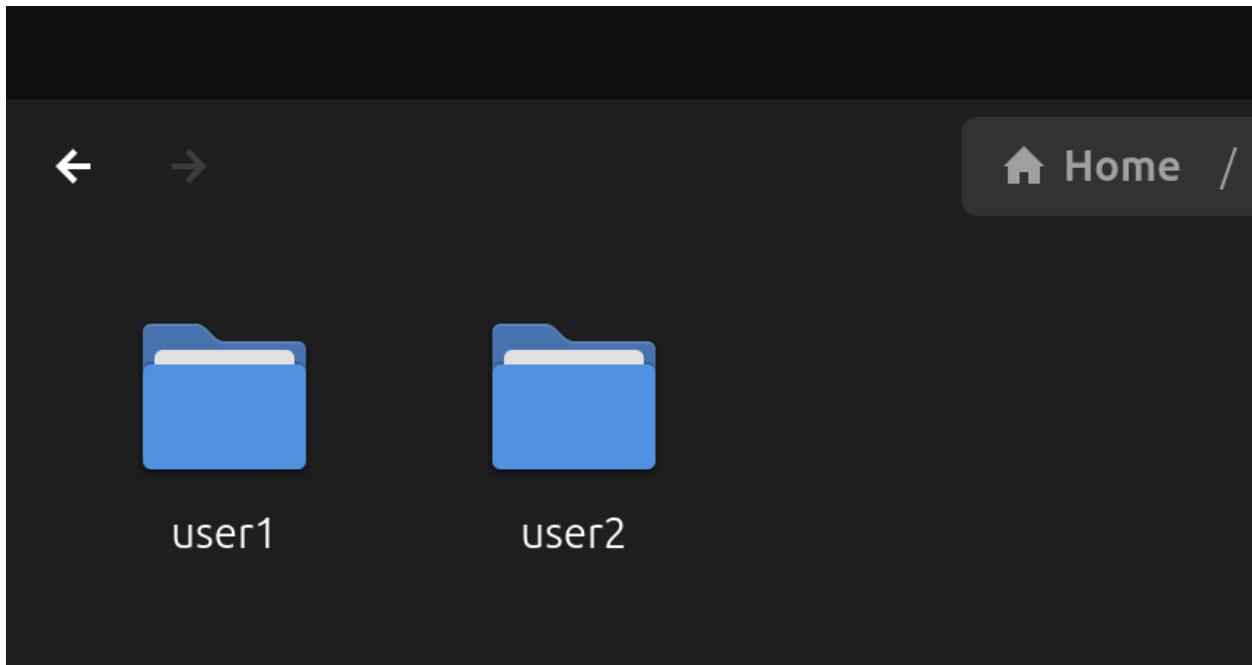
README

Add a README

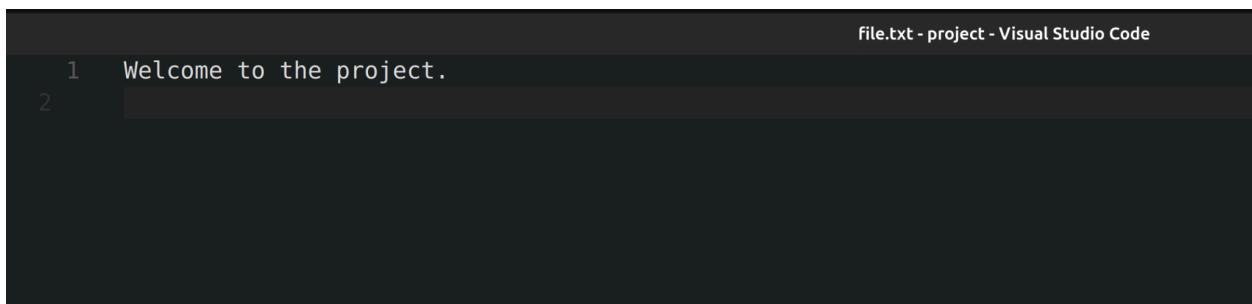
Now coders - user1 and user2 can clone this repository locally using:

git clone <repo url>

For demonstration, consider user1 clones it to folder 'user1' and user2 clones it to folder 'user2'.



Consider this file.txt from main branch.



To start working, user1 will create a feature branch e.g. feature-a, and add some code.

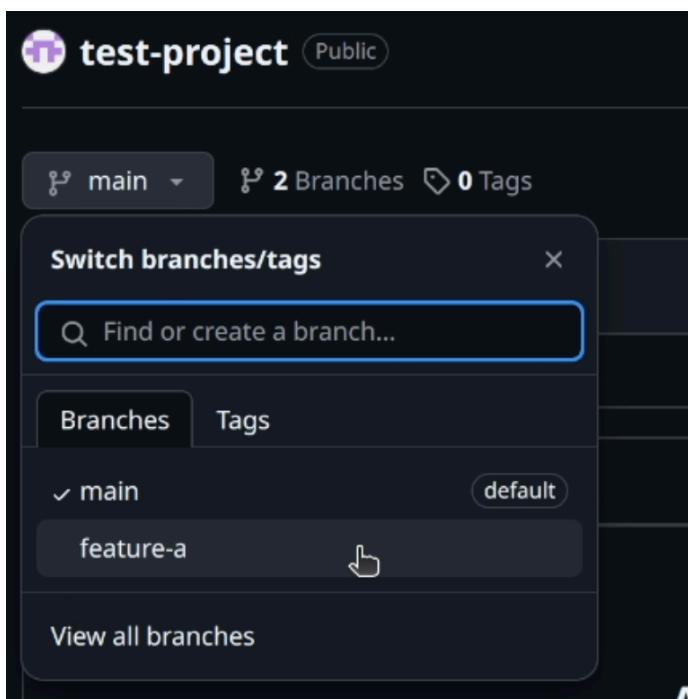
```
PROBLEMS      OUTPUT      DEBUG CONSOLE      TERMINAL      PORTS

● aaditya@g14:~/project/user1$ git branch
  feature-a
* main
● aaditya@g14:~/project/user1$ git checkout feature-a
Switched to branch 'feature-a'
○ aaditya@g14:~/project/user1$ █
```

```
3  Welcome to the project.  
2  
1  ===  
4  Feature A code written by user1
```

Now user1 will push his/her changes onto the remote feature-branch as shown:

```
aaditya@g14:~/project/user1$ git push origin feature-a  
Enumerating objects: 5, done.  
Counting objects: 100% (5/5), done.  
Delta compression using up to 16 threads  
Compressing objects: 100% (2/2), done.  
Writing objects: 100% (3/3), 316 bytes | 316.00 KiB/s, done.  
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0  
remote:  
remote: Create a pull request for 'feature-a' on GitHub by visiting:  
remote:     https://github.com/aaditya-user3/test-project/pull/new/feature-a  
remote:  
To https://github.com/aaditya-user3/test-project.git  
 * [new branch]      feature-a -> feature-a  
aaditya@g14:~/project/user1$
```



Note: If you get error like this...

```
❶ aaditya@g14:~/project/user1$ git push origin feature-a
remote: Permission to Aaditya-User3/test-project.git denied to Aaditya-1219.
fatal: unable to access 'https://github.com/Aaditya-User3/test-project.git/': The requested URL returned error: 403
❷ aaditya@g14:~/project/user1$ █
```

Refer this youtube video to fix it.

<https://youtu.be/dtZIEQc0J1w>

Moving on...

The screenshot shows a GitHub repository named 'test-project'. The 'feature-a' branch is currently selected. The repository has 2 branches and 0 tags. A message indicates that the 'feature-a' branch is 1 commit ahead of the 'main' branch. A recent commit by user 'aaditya-1219' titled 'Added code for feature-a' was made 25 minutes ago, with a file named 'file.txt' attached. The 'README' file is also visible.

To merge these changes, user1 will create a Pull Request. The leader will approve this later.

To create a Pull Request, go to Pull Requests -> set compare: feature-a

The screenshot shows the 'Compare changes' interface on GitHub. It displays a dropdown menu for 'base: main' and 'compare: main'. A modal window is open, prompting the user to 'Choose a head ref' and showing 'feature-a' as the selected branch. Below the modal, a section titled 'Compare and review just about anything' lists example comparisons between 'feature-a' and 'main' at different times, such as '25 minutes ago' and '24 hours ago'.

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#) or [learn more about diff comparisons](#).

base: main ▾ ← compare: feature-a ▾ ✓ **Able to merge.** These branches can be automatically merged.

Discuss and review the changes in this comparison with others. [Learn about pull requests](#) Create pull request

→ 1 commit 1 file changed 1 contributor

Commits on Nov 8, 2024

Added code for feature-a
aditya-1219 committed 25 minutes ago

Showing 1 changed file with 3 additions and 0 deletions. Split Unified

3 file.txt

```
... ...
1 1 Welcome to the project.
2 +
3 + ===
4 + Feature A code written by user1
```

Now the leader will approve this Pull Request:

Added code for feature-a #1

Open aditya-user2 wants to merge 1 commit into `main` from `feature-a`

Conversation 0 Commits 1 Checks 0 Files changed 1

aditya-user2 commented now
No description provided.

Added code for feature-a f8f7856

Require approval from specific reviewers before merging [Rulesets](#) ensure specific people approve pull requests before they're merged. Add rule

Continuous integration has not been set up [GitHub Actions](#) and [several other apps](#) can be used to automatically catch bugs and enforce style.

This branch has no conflicts with the base branch Merging can be performed automatically.

Merge pull request or view [command line instructions](#).

Reviewers: No reviews Still in progress? [Convert to draft](#)

Assignees: No one—assign yourself

Labels: None yet

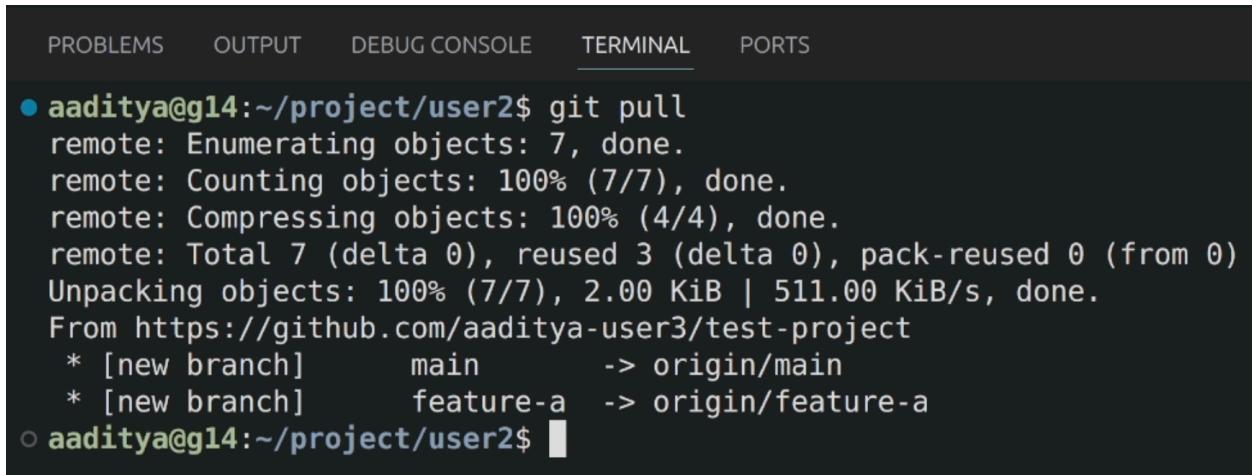
Projects: None yet

Milestone: No milestone

Development: Successfully merging this pull request may close these issues

The 'feature-a' branch will now be merged with the main branch.

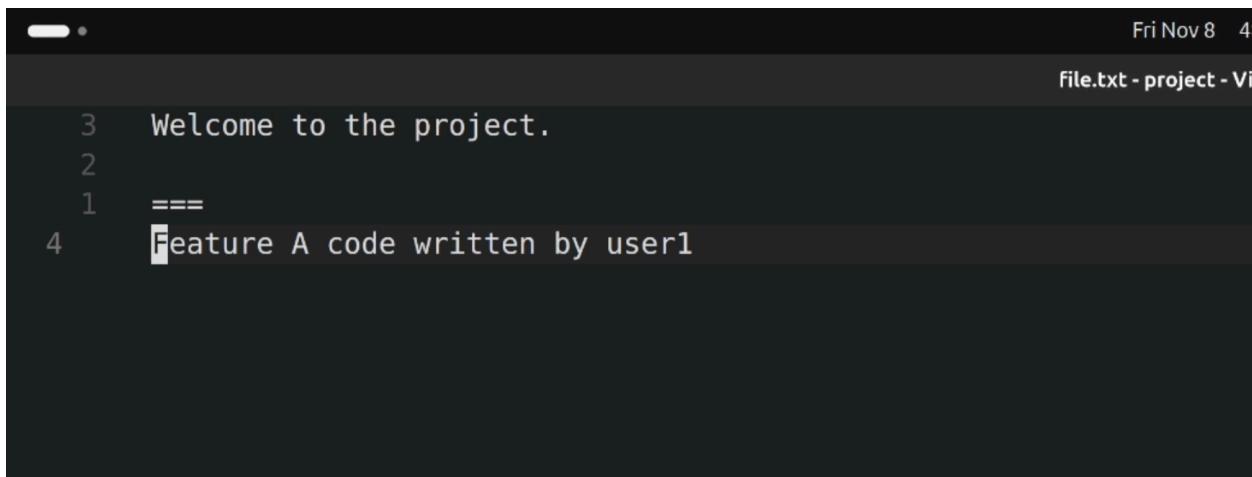
At this point, user2 does not have the changes made to the main branch.
So user2 will use **git pull** command to get the latest changes from main.
Command: **git pull**



The screenshot shows a terminal window with tabs: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is selected), and PORTS. The terminal output is as follows:

```
● aaditya@g14:~/project/user2$ git pull
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 7 (delta 0), reused 3 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (7/7), 2.00 KiB | 511.00 KiB/s, done.
From https://github.com/aaditya-user3/test-project
 * [new branch]      main      -> origin/main
 * [new branch]      feature-a -> origin/feature-a
○ aaditya@g14:~/project/user2$ █
```

Now user2 has changes from main.



The screenshot shows a vi editor window titled "File.txt - project - Vi". The content of the file is:

```
3 Welcome to the project.
2
1 ===
4 Feature A code written by user1
```

Similarly, user2 will create a feature branch 'feature-b' as shown earlier.
Now user2 will add some code, commit the changes then push.

The screenshot shows a dark-themed interface of Visual Studio Code. At the top right, it displays the date and time: Fri Nov 8 5:03:14 PM. Below that, it says "File.txt - project - Visual Studio Cod". The main area contains a code editor with the following content:

```
6 Welcome to the project.  
5  
4 ===  
3 Feature A code written by user1  
2  
1 ===  
7 Feature B code written by user2
```

Below the code editor, there is a navigation bar with tabs: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined), and PORTS.

The terminal pane at the bottom shows a git session:

- **aaditya@g14:~/project/user2\$ git status**
On branch feature-b
Changes not staged for commit:
(use "git add <file>..." to update what will be committed)
(use "git restore <file>..." to discard changes in working directory)
modified: file.txt
- no changes added to commit (use "git add" and/or "git commit -a")
- **aaditya@g14:~/project/user2\$ git add file.txt**
- **aaditya@g14:~/project/user2\$ git commit -m "Added feature-b code"**
[feature-b 70e20d8] Added feature-b code
1 file changed, 4 insertions(+), 1 deletion(-)
- **aaditya@g14:~/project/user2\$**

"file.txt" 7L 97C written

As shown earlier, user2 will also create a pull request, which will be approved by the leader.

Added feature-b code #2

aaditya-user2 wants to merge 1 commit into `main` from `feature-b`

Conversation 0 Commits 1 Checks 0 Files changed 1

aaditya-user2 commented now
No description provided.

Added Feature-b code

This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request or view command line instructions.

Add a comment

Write Preview

Reviewers
No reviews
Still in progress? Convert to draft

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Development
Successfully merging this pull request may close these issues.

Final result:

test-project / file.txt

aaditya-1219 Added feature-b code

Code Blame 7 lines (5 loc) · 97 Bytes Code 55% faster with GitHub Copilot

```
1 Welcome to the project.
2
3 ===
4 Feature A code written by user1
5
6 ===
7 Feature B code written by user2
```