VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Machhe, Belagavi, Karnataka-590018

Lab Experiment Record

Project Management with Git [BCSL358C]

Submitted in partial fulfillment towards AEC of 3rd semester of

Bachelor of Engineering
in
Computer Science and Engineering
(Artificial Intelligence & Machine Learning)

Submitted by

CHINMAYI MOHAN
4GW24CI010

DEPARTMENT OF CSE (Artificial Intelligence & Machine Learning)

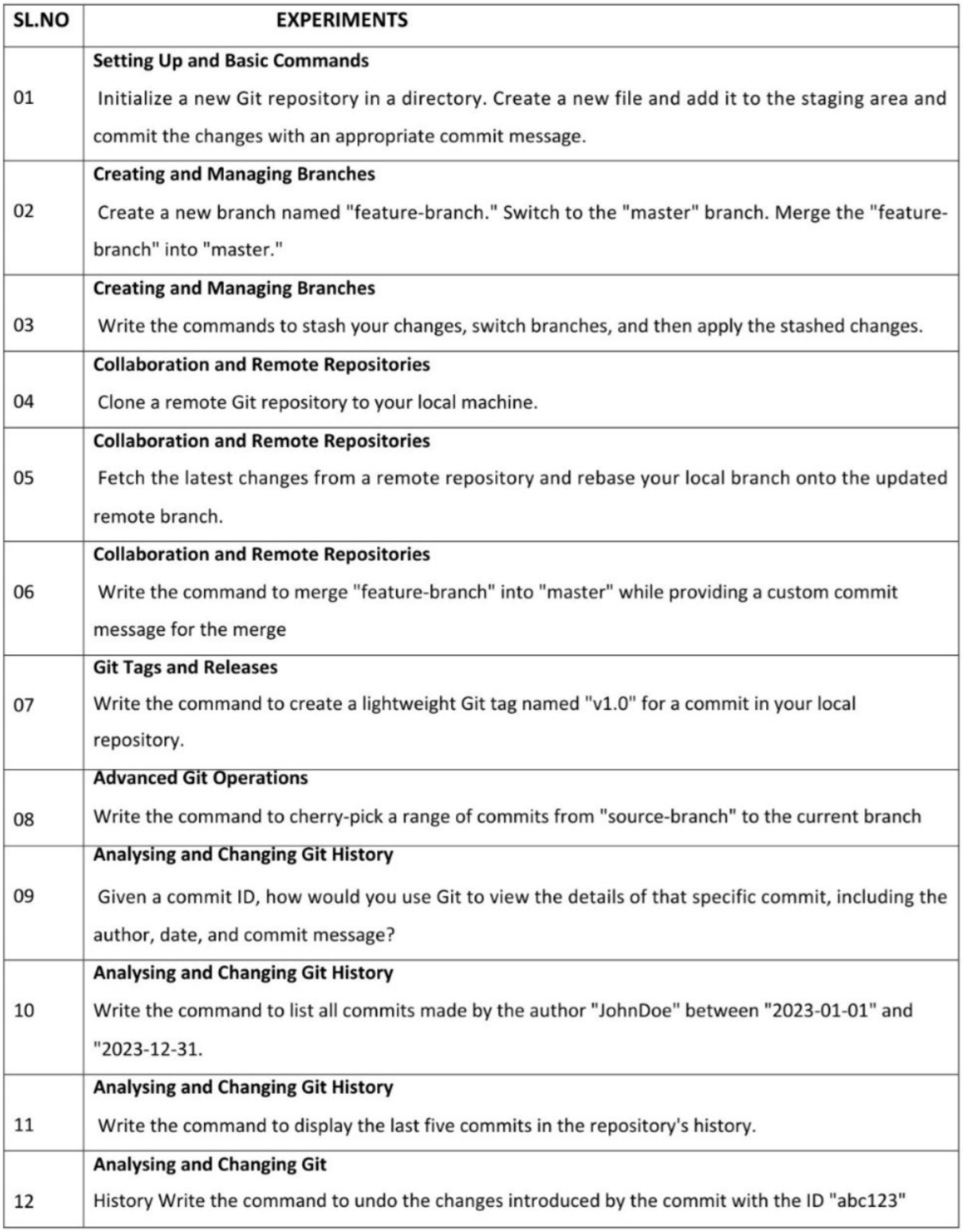
GSSS INSTITUTE OF ENGINEERING & TECHNOLOGY FOR WOMEN

(Affiliated to VTU, Belagavi, Approved by AICTE, New Delhi & Govt. of Karnataka)

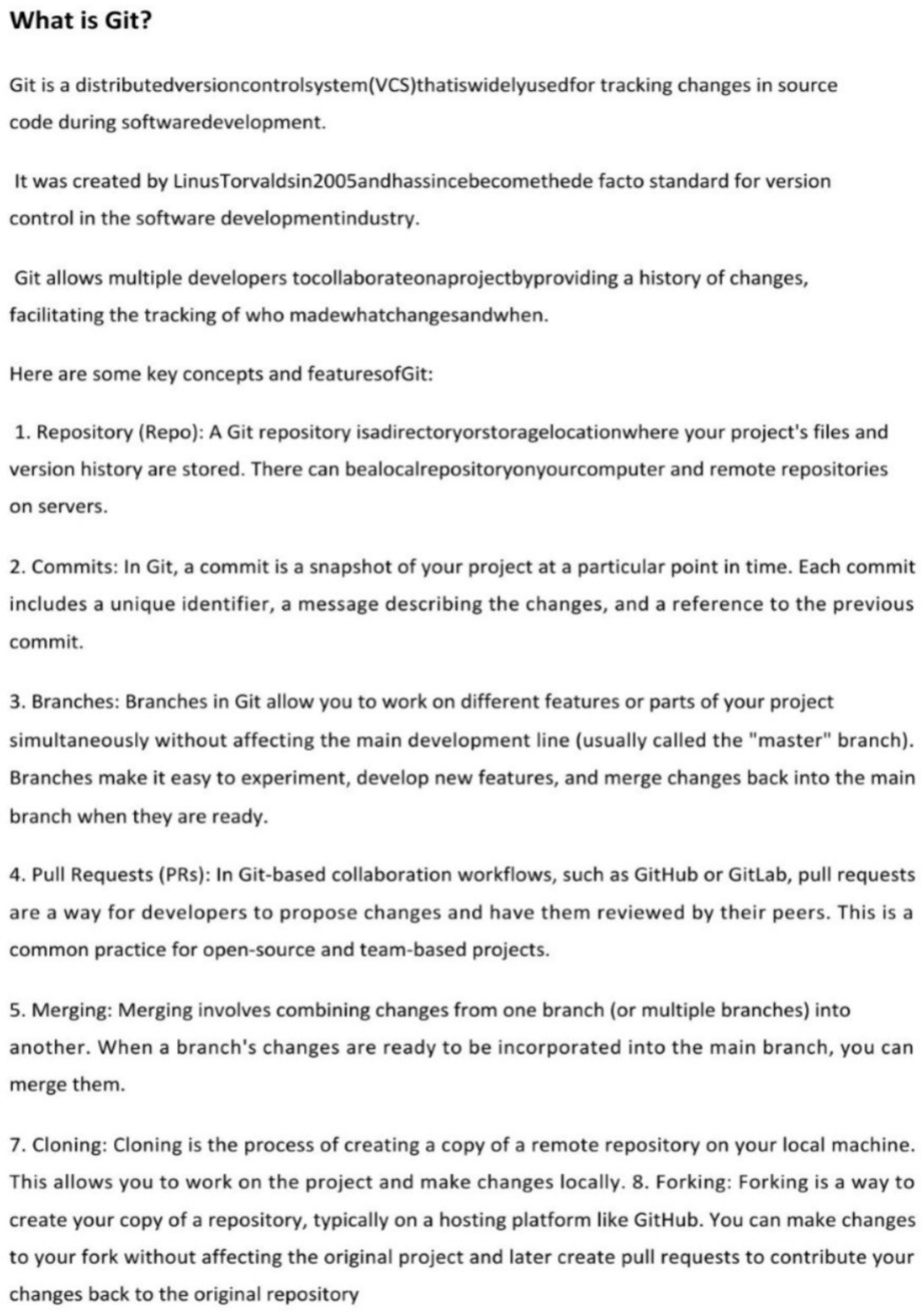
K.R.S ROAD, METAGALLI, MYSURU-570016, KARNATAKA

(Accredited by NAAC)

2025-2026



SL.NO	EXPERIMENTS
01	**Setting Up and Basic Commands** Initialize a new Git repository in a directory. Create a new file and add it to the staging area and commit the changes with an appropriate commit message.
02	**Creating and Managing Branches** Create a new branch named "feature-branch." Switch to the "master" branch. Merge the "feature-branch" into "master."
03	**Creating and Managing Branches** Write the commands to stash your changes, switch branches, and then apply the stashed changes.
04	**Collaboration and Remote Repositories** Clone a remote Git repository to your local machine.
05	**Collaboration and Remote Repositories** Fetch the latest changes from a remote repository and rebase your local branch onto the updated remote branch.
06	**Collaboration and Remote Repositories** Write the command to merge "feature-branch" into "master" while providing a custom commit message for the merge
07	**Git Tags and Releases** Write the command to create a lightweight Git tag named "v1.0" for a commit in your local repository.
08	**Advanced Git Operations** Write the command to cherry-pick a range of commits from "source-branch" to the current branch
09	**Analysing and Changing Git History** Given a commit ID, how would you use Git to view the details of that specific commit, including the author, date, and commit message?
10	**Analysing and Changing Git History** Write the command to list all commits made by the author "JohnDoe" between "2023-01-01" and "2023-12-31.
11	**Analysing and Changing Git History** Write the command to display the last five commits in the repository's history.
12	**Analysing and Changing Git** History Write the command to undo the changes introduced by the commit with the ID "abc123"



What is Git?

Git is a distributedversioncontrolsystem(VCS)thatiswidelyusedfor tracking changes in source code during softwaredevelopment.

It was created by LinusTorvaldsin2005andhassincebecomethede facto standard for version control in the software developmentindustry.

Git allows multiple developers tocollaborateonaprojectbyproviding a history of changes, facilitating the tracking of who madewhatchangesandwhen.

Here are some key concepts and featuresofGit:

1. Repository (Repo): A Git repository isadirectoryorstoragelocationwhere your project's files and version history are stored. There can bealocalrepositoryonyourcomputer and remote repositories on servers.

2. Commits: In Git, a commit is a snapshot of your project at a particular point in time. Each commit includes a unique identifier, a message describing the changes, and a reference to the previous commit.

3. Branches: Branches in Git allow you to work on different features or parts of your project simultaneously without affecting the main development line (usually called the "master" branch). Branches make it easy to experiment, develop new features, and merge changes back into the main branch when they are ready.

4. Pull Requests (PRs): In Git-based collaboration workflows, such as GitHub or GitLab, pull requests are a way for developers to propose changes and have them reviewed by their peers. This is a common practice for open-source and team-based projects.

5. Merging: Merging involves combining changes from one branch (or multiple branches) into another. When a branch's changes are ready to be incorporated into the main branch, you can merge them.

7. Cloning: Cloning is the process of creating a copy of a remote repository on your local machine. This allows you to work on the project and make changes locally. 8. Forking: Forking is a way to create your copy of a repository, typically on a hosting platform like GitHub. You can make changes to your fork without affecting the original project and later create pull requests to contribute your changes back to the original repository

EXPERIMENT 1: SETTING UP AND BASIC COMMANDS

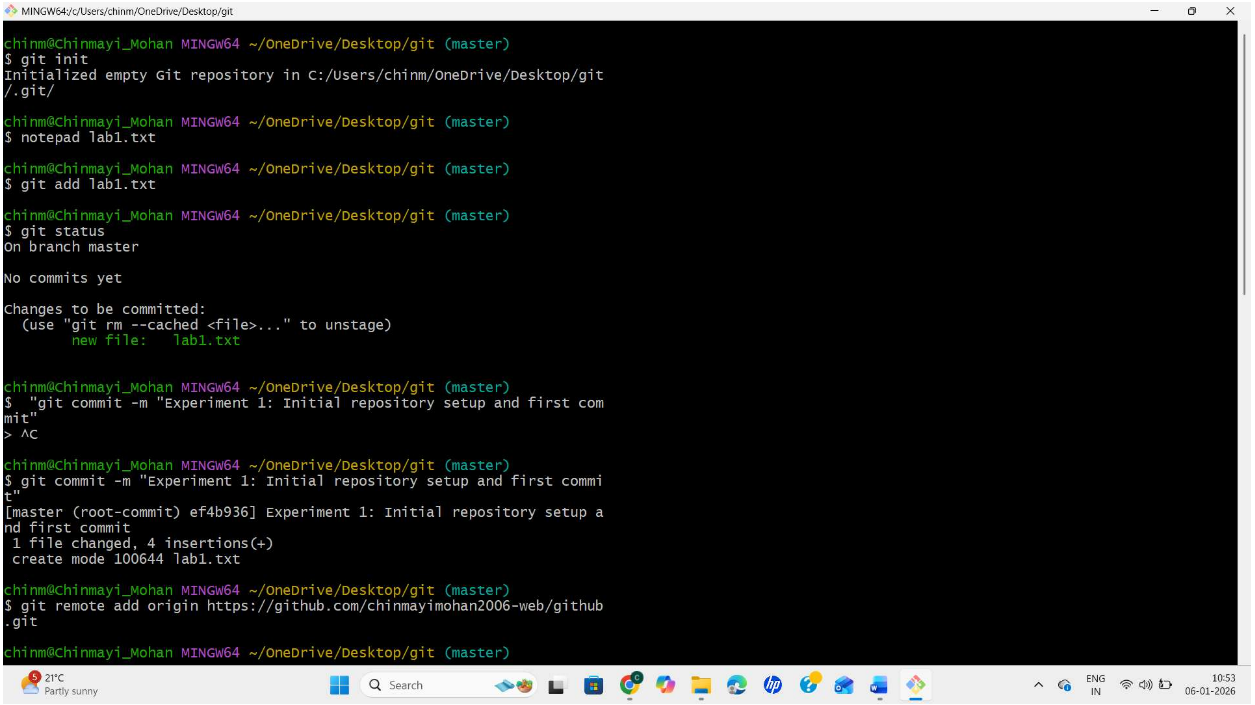
PROCEDURE

1. Open Git Bash and move to the project folder created on the Desktop using the cd command.

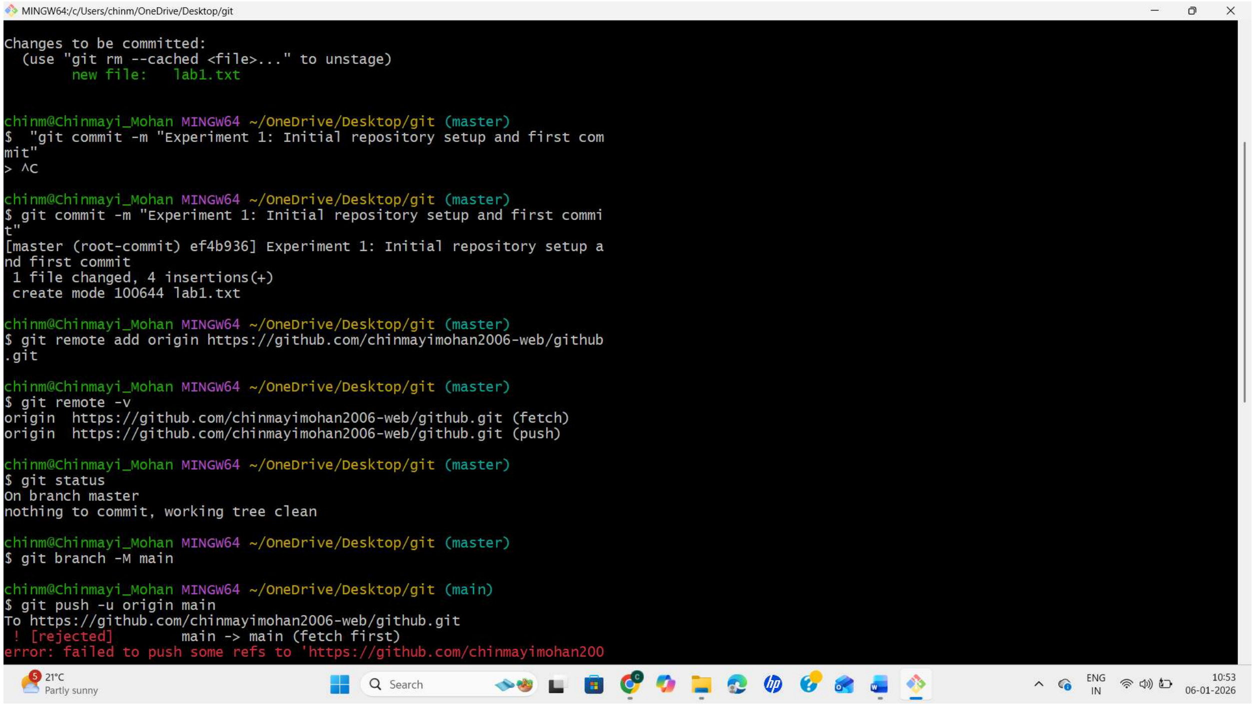
2. Initialize the folder as a Git repository using the git init command.
3. Create a new text file using Notepad to store the experiment content.
4. Add the created file to the staging area using the git add command.
5. Check the status of the repository using the git status command.
6. Commit the staged file to the local repository using the git commit command.
7. Link the local repository to the remote GitHub repository using the git remote add origin command.
8. Rename the default branch to main using the git branch -M main command.
9. Synchronize the local repository with the remote repository using the git pull --rebase command.
10. Push the committed files from the local repository to the GitHub repository using the git push command.

    GIT REPO URL :

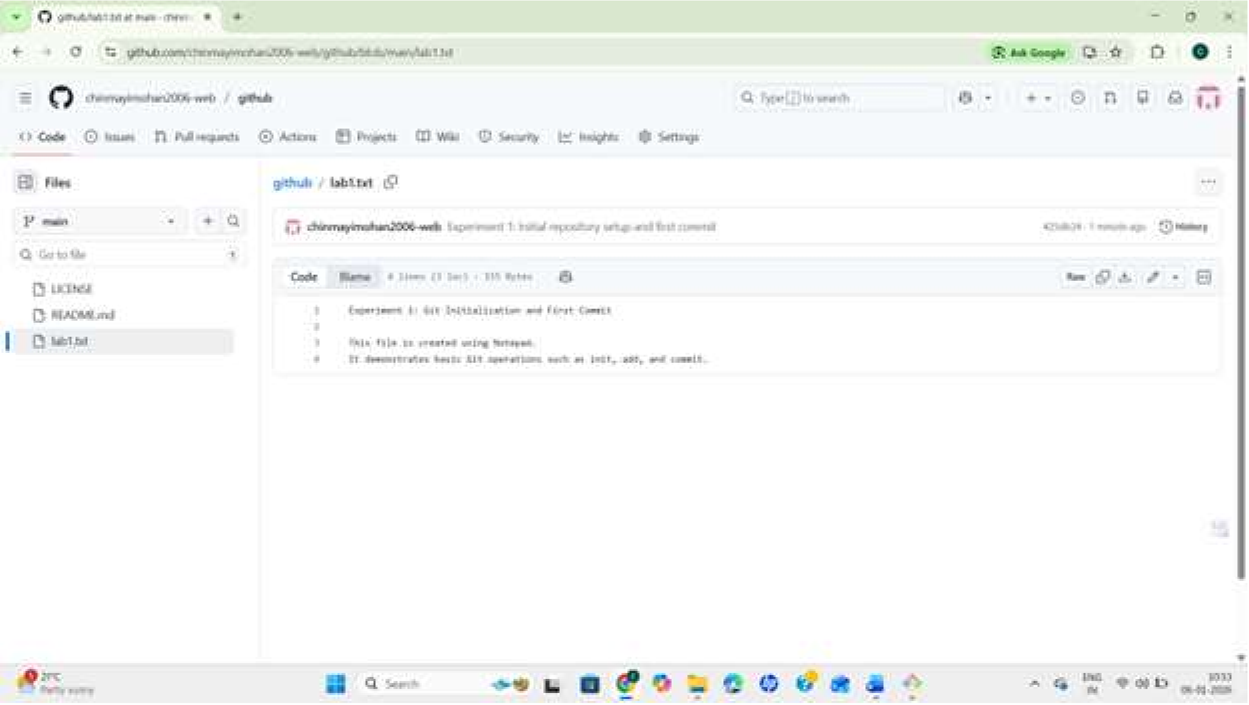
    https://github.com/chinmayimohan2006-web/4GW24CI010.git



```
chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/git (master)
$ git init
Initialized empty Git repository in C:/Users/chinm/OneDrive/Desktop/git
/.git/

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/git (master)
$ notepad lab1.txt

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/git (master)
$ git add lab1.txt

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/git (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   lab1.txt


chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/git (master)
$  "git commit -m "Experiment 1: Initial repository setup and first com
mit"
> ^C

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/git (master)
$ git commit -m "Experiment 1: Initial repository setup and first commi
t"
[master (root-commit) ef4b936] Experiment 1: Initial repository setup a
nd first commit
 1 file changed, 4 insertions(+)
 create mode 100644 lab1.txt

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/git (master)
$ git remote add origin https://github.com/chinmayimohan2006-web/github
.git

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/git (master)
```



```
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   lab1.txt


chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/git (master)
$  "git commit -m "Experiment 1: Initial repository setup and first com
mit"
> ^C

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/git (master)
$ git commit -m "Experiment 1: Initial repository setup and first commi
t"
[master (root-commit) ef4b936] Experiment 1: Initial repository setup a
nd first commit
 1 file changed, 4 insertions(+)
 create mode 100644 lab1.txt

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/git (master)
$ git remote add origin https://github.com/chinmayimohan2006-web/github
.git

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/git (master)
$ git remote -v
origin  https://github.com/chinmayimohan2006-web/github.git (fetch)
origin  https://github.com/chinmayimohan2006-web/github.git (push)

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/git (master)
$ git status
On branch master
nothing to commit, working tree clean

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/git (master)
$ git branch -M main

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/git (main)
$ git push -u origin main
To https://github.com/chinmayimohan2006-web/github.git
 ! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'https://github.com/chinmayimohan200
```



```
Experiment 1: Git Initialization and First Commit

This file is created using Notepad.
It demonstrates basic Git operations such as init, add, and commit.
```

COMMANDS USED:

```
git init
notepad lab1.txt
git add lab1.txt
git status
git commit -m "Experiment 1: Initial repository setup and first commit"
git remote add origin https://github.com/chinmayimohan2006-web/github.git
git branch -M main
git pull --rebase origin main
git push -u origin main
```

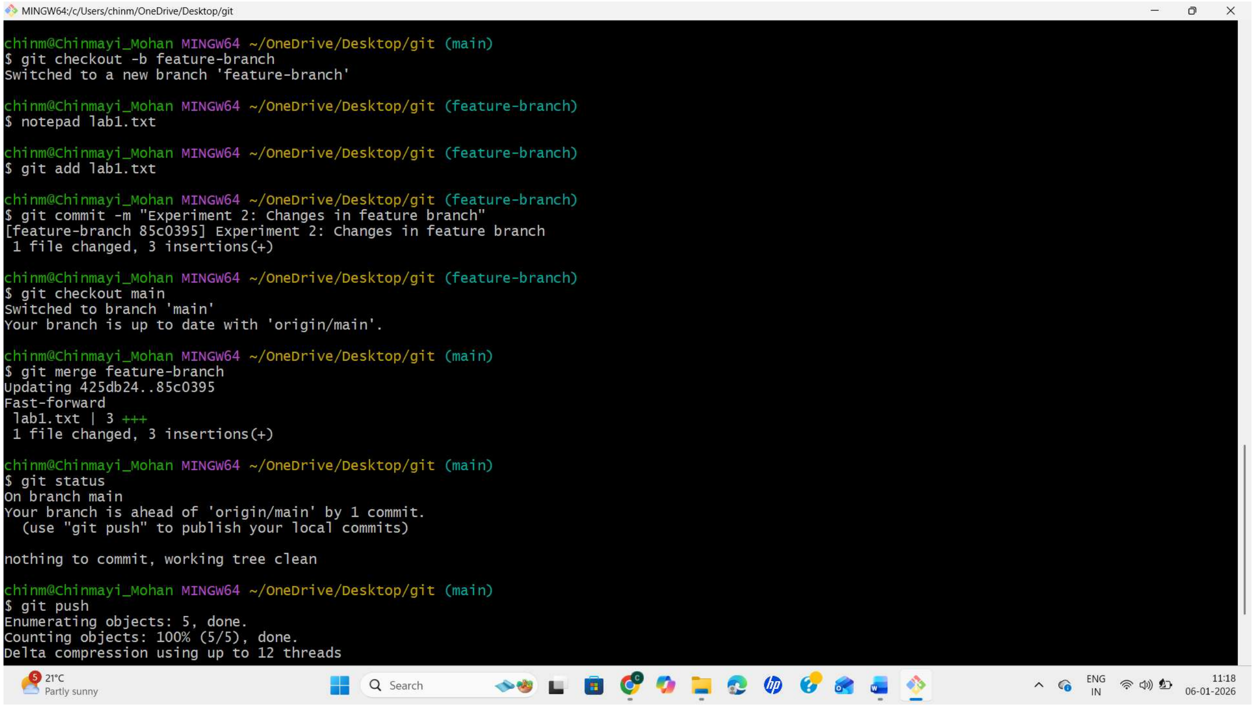
EXPERIMENT 2: CREATING AND MERGING BRANCHES

PROCEDURE

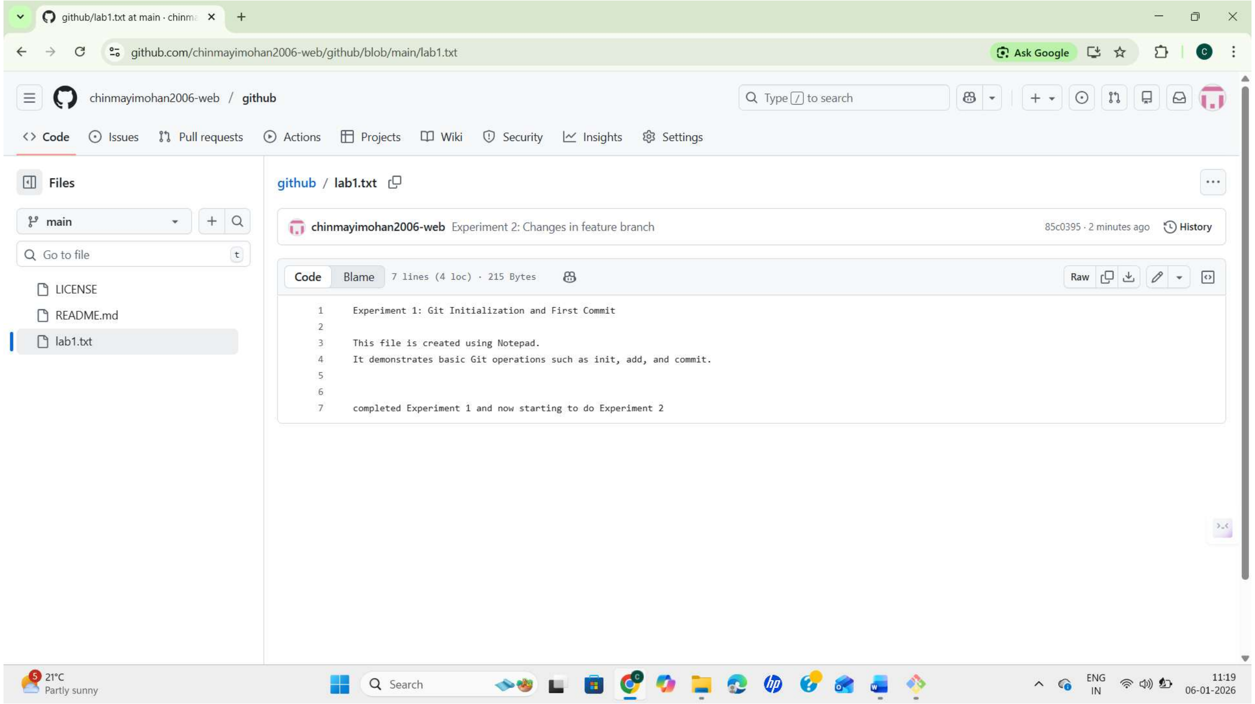
1. Open Git Bash and move to the project directory.
2. Switch to the main branch using the git checkout command.
3. Create a new branch named feature-branch and switch to it.
4. Modify an existing file using Notepad.
5. Add the modified file to the staging area.
6. Commit the changes in the feature branch.
7. Switch back to the main branch.
8. Merge the feature-branch into the main branch.

COMMANDS USED:

```
git checkout main

git checkout -b feature-branch

notepad lab1.txt

git add lab1.txt

git commit -m "Experiment 2: Changes in feature branch"

git checkout main

git merge feature-branch
```



```
chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/git (main)
$ git checkout -b feature-branch
Switched to a new branch 'feature-branch'

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/git (feature-branch)
$ notepad lab1.txt

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/git (feature-branch)
$ git add lab1.txt

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/git (feature-branch)
$ git commit -m "Experiment 2: Changes in feature branch"
[feature-branch 85c0395] Experiment 2: Changes in feature branch
 1 file changed, 3 insertions(+)

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/git (feature-branch)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/git (main)
$ git merge feature-branch
Updating 425db24..85c0395
Fast-forward
 lab1.txt | 3 +++
 1 file changed, 3 insertions(+)

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/git (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/git (main)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
```



```
Experiment 1: Git Initialization and First Commit

This file is created using Notepad.
It demonstrates basic Git operations such as init, add, and commit.



completed Experiment 1 and now starting to do Experiment 2
```

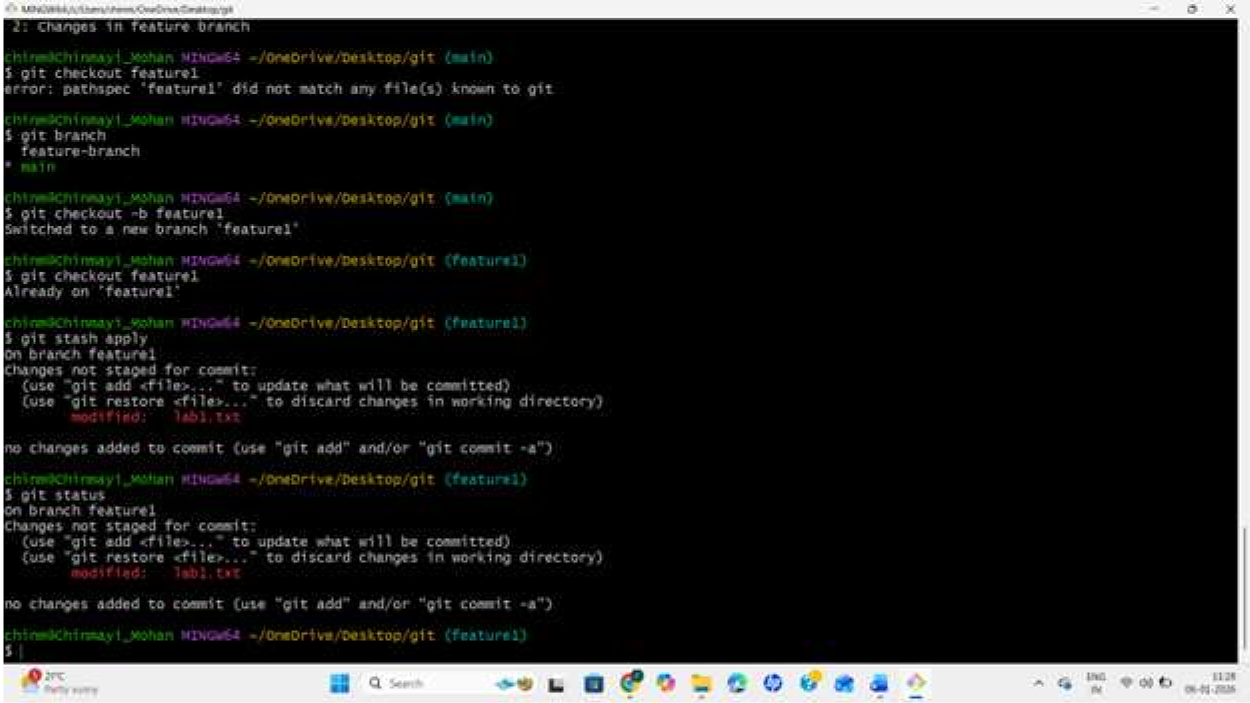
EXPERIMENT 3: CREATING AND MERGING BRANCHES

PROCEDURE:

1. Open Git Bash and move to the project directory.
2. Modify an existing file using Notepad.
3. Check the repository status using the git status command.
4. Save the uncommitted changes using the git stash command.
5. Verify that the working directory is clean.
6. Switch to another branch.
7. Apply the stashed changes using the git stash apply command.
8. Check the repository status again.

COMMANDS USED:

notepad lab1.txt , git status , git stash , git checkout feature1 , git stash apply , git status



```
 2: Changes in feature branch

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/git (main)
$ git checkout feature1
error: pathspec 'feature1' did not match any file(s) known to git

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/git (main)
$ git branch
  feature-branch
* main

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/git (main)
$ git checkout -b feature1
Switched to a new branch 'feature1'

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/git (feature1)
$ git checkout feature1
Already on 'feature1'

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/git (feature1)
$ git stash apply
On branch feature1
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:   lab1.txt

no changes added to commit (use "git add" and/or "git commit -a")

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/git (feature1)
$ git status
On branch feature1
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:   lab1.txt

no changes added to commit (use "git add" and/or "git commit -a")

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/git (feature1)
$ |
```

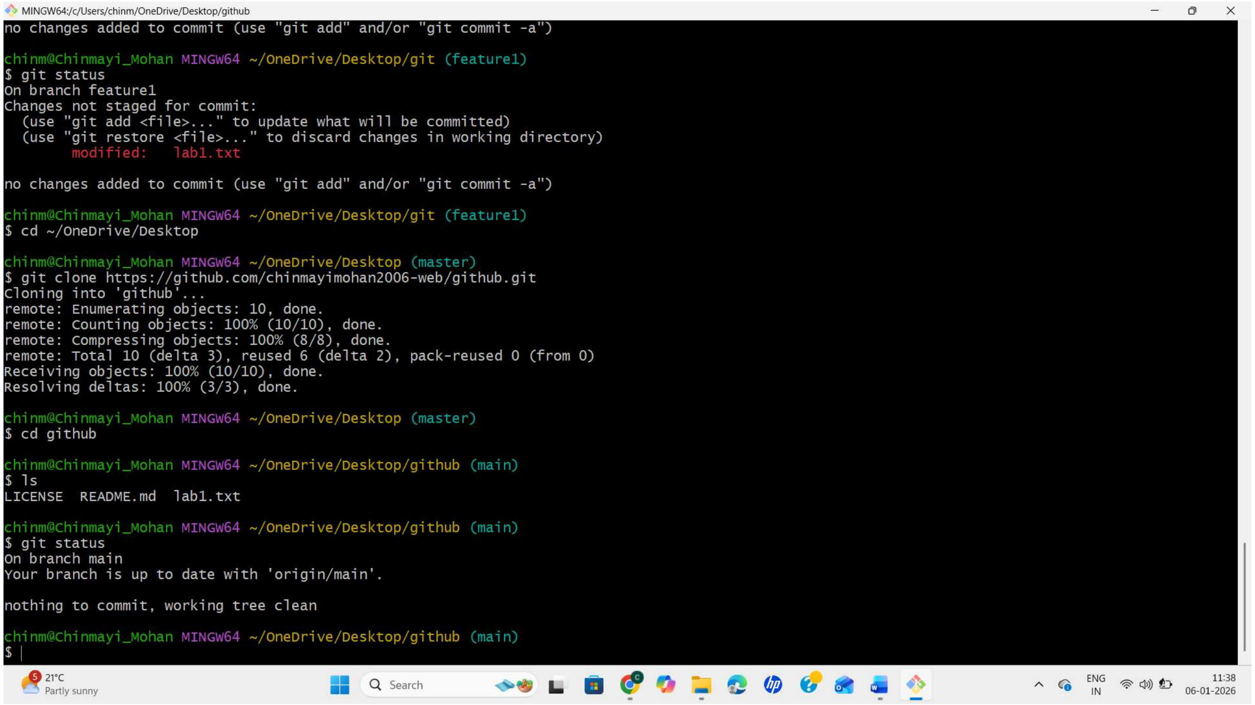
EXPERIMENT 4: COLLABORATIONS AND REPORT REPOSITORIES

PROCEDURE:

1. The terminal was navigated to the Desktop using the cd command.
2. The remote GitHub repository was copied to the local system using the git clone command.
3. The cloned repository folder was accessed using the cd command.
4. The files present in the repository were displayed using the ls command.
5. The current status of the repository was checked using the git status command.

COMMANDS USED:

```
 cd ~/OneDrive/Desktop
git clone <repository-URL>
cd <repository-folder>
ls
git status
```



MINGW64:/c/Users/chinm/OneDrive/Desktop/github

```
no changes added to commit (use "git add" and/or "git commit -a")

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/git (feature1)
$ git status
On branch feature1
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:   lab1.txt

no changes added to commit (use "git add" and/or "git commit -a")

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/git (feature1)
$ cd ~/OneDrive/Desktop

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop (master)
$ git clone https://github.com/chinmayimohan2006-web/github.git
Cloning into 'github'...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 10 (delta 3), reused 6 (delta 2), pack-reused 0 (from 0)
Receiving objects: 100% (10/10), done.
Resolving deltas: 100% (3/3), done.

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop (master)
$ cd github

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$ ls
LICENSE  README.md  lab1.txt

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$ |
```

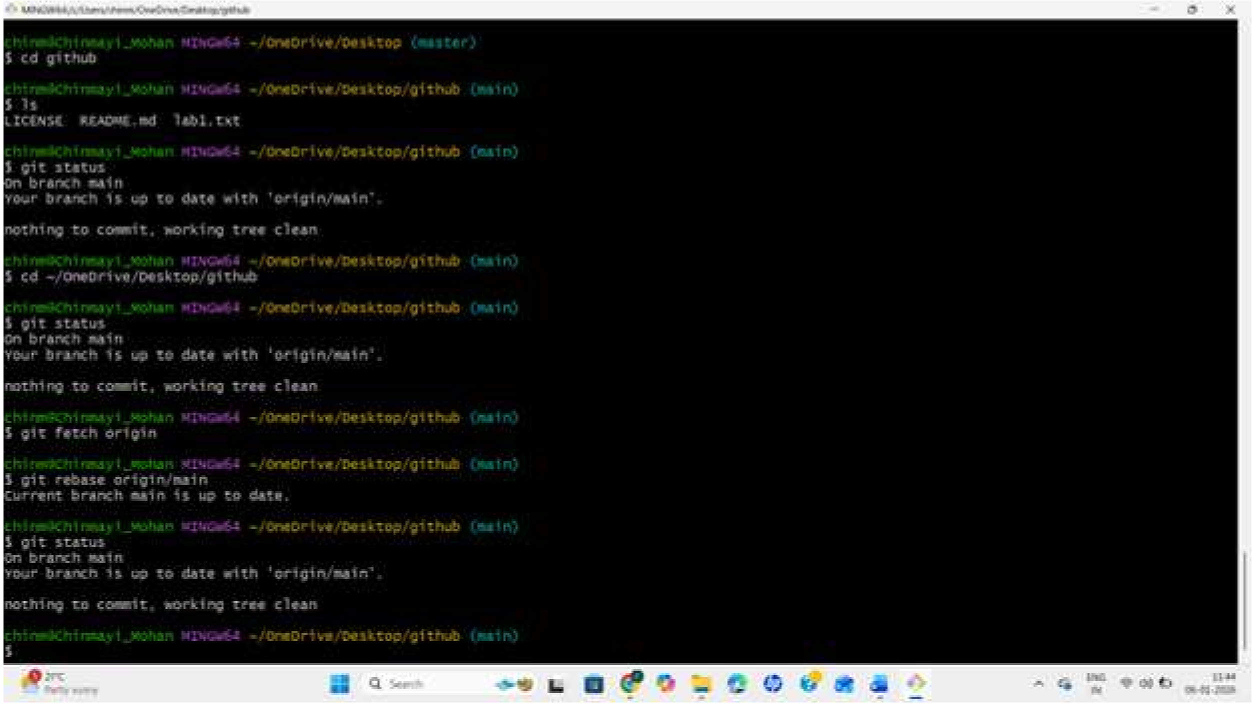
EXPERIMENT 5: COLLABORATIONS AND REPORT REPOSITORIES

PROCEDURE:

1. The terminal was navigated to the local GitHub repository directory.
2. The current branch and working tree status were verified using the git status command.
3. The latest changes from the remote repository were fetched using the git fetch origin command.
4. The local branch was synchronized with the remote main branch using the git rebase origin/main command.
5. The repository status was checked again to ensure that the working tree was clean and up to date.

COMMANDS USED:

cd ~/OneDrive/Desktop/github , git status , git status , git fetch origin, git rebase origin/main



```
chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop (master)
$ cd github

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$ ls
LICENSE  README.md  lab1.txt

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$ cd ~/OneDrive/Desktop/github

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$ git fetch origin

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$ git rebase origin/main
Current branch main is up to date.

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$
```

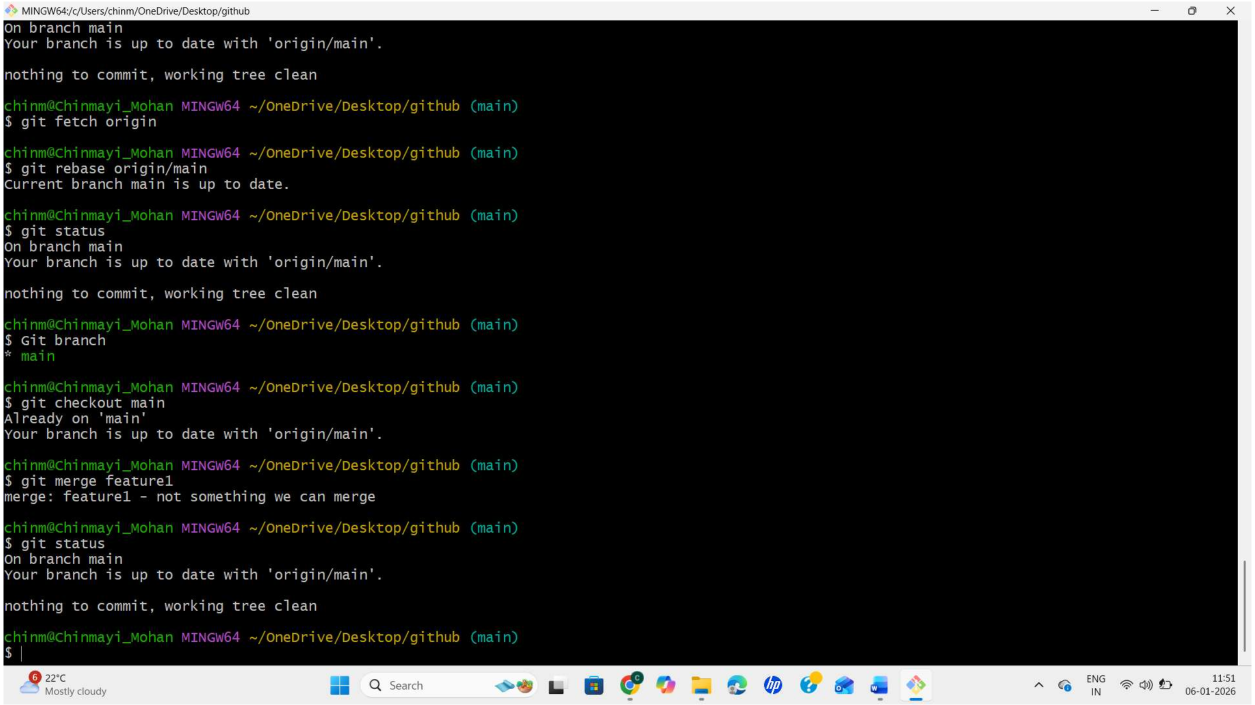
EXPERIMENT 6: COLLABORATION AND REMOTE REPOSITORIES

PROCIDURE:

1. The list of available branches was displayed using the git branch command.
2. The current working branch was switched to the main branch using git checkout main.
3. The changes from the feature branch were merged into the main branch using the git merge feature1 command.
4. The status of the repository was verified using the git status command to ensure successful merge.

COMMANDS USED:

git branch, git checkout main, git , git merge feature1, git status



```
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$ git fetch origin

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$ git rebase origin/main
Current branch main is up to date.

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$ Git branch
* main

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$ git checkout main
Already on 'main'
Your branch is up to date with 'origin/main'.

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$ git merge feature1
merge: feature1 - not something we can merge

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$ |
```

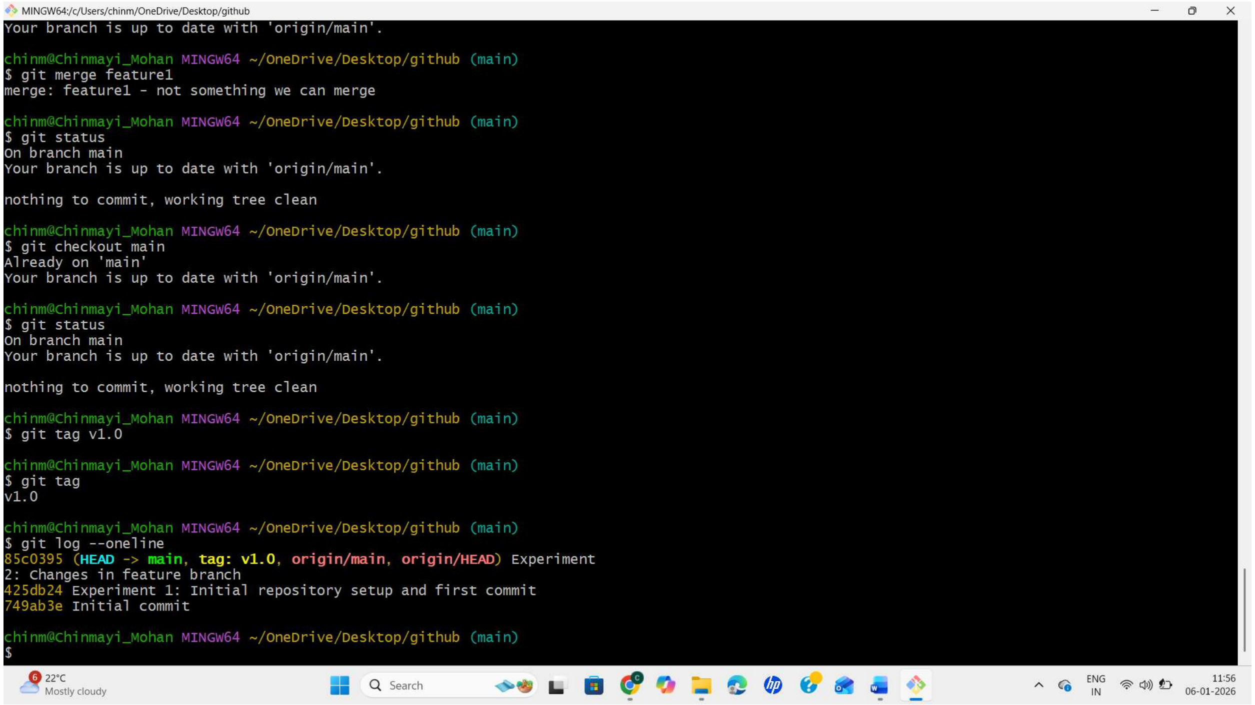
EXPERIMENT 7: GIT TAGS AND RELEASES

PROCEDURE:

1. The working branch was verified and switched to the main branch using the git checkout main command.
2. The current repository status was checked using git status.
3. A tag named v1.0 was created using the git tag command to mark an important version of the project.
4. The list of available tags was displayed using the git tag command.
5. The commit history was viewed using git log --oneline to confirm that the tag was attached to the correct commit.

COMMANDS USED:

git checkout main , git status, git tag v1.0, git tag , git log –oneline



```
Your branch is up to date with 'origin/main'.

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$ git merge feature1
merge: feature1 - not something we can merge

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$ git checkout main
Already on 'main'
Your branch is up to date with 'origin/main'.

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$ git tag v1.0

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$ git tag
v1.0

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$ git log --oneline
85c0395 (HEAD -> main, tag: v1.0, origin/main, origin/HEAD) Experiment
2: Changes in feature branch
425db24 Experiment 1: Initial repository setup and first commit
749ab3e Initial commit

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$
```

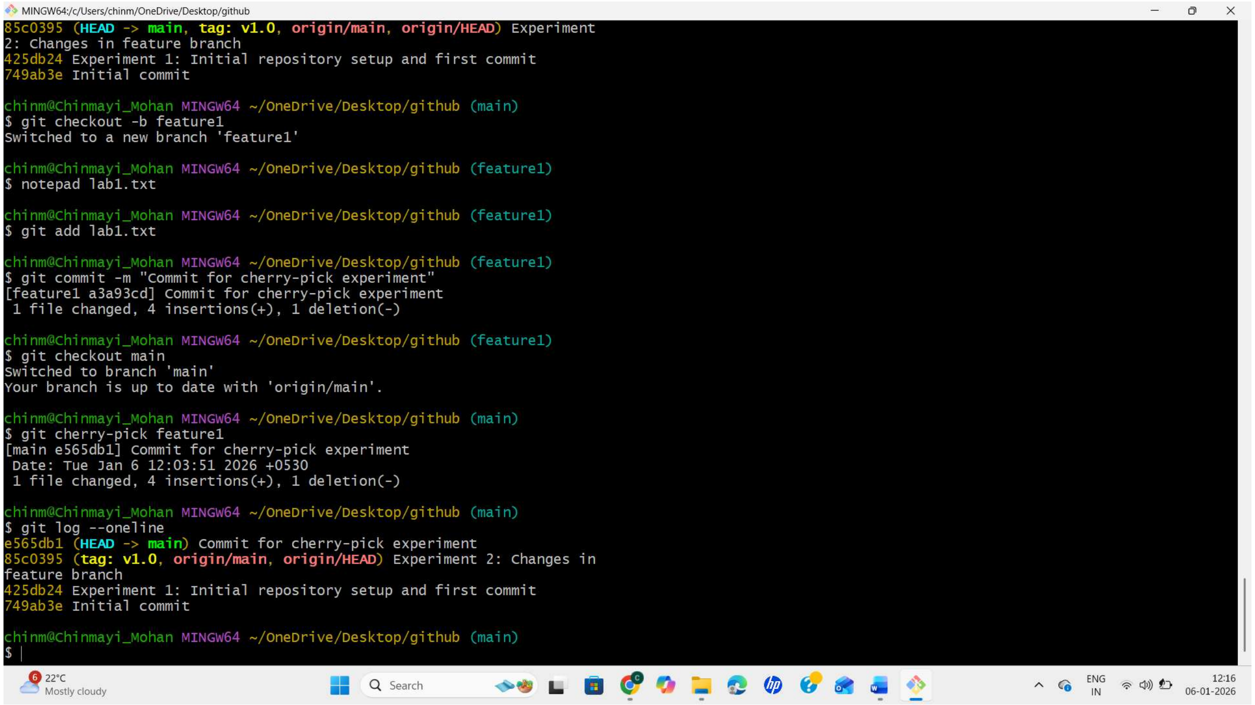
EXPERIMENT 8: ADVANCED GIT OPERATIONS

PROCEDURE:

1. The commit history was viewed using the git log --oneline command.
2. A new branch named feature1 was created using git checkout -b feature1.
3. The file lab1.txt was modified using Notepad.
4. The modified file was staged using the git add command.
5. The changes were committed using the git commit command.
6. The working branch was switched back to main using git checkout main.
7. The commit from the feature branch was applied to the main branch using the git cherry-pick command.
8. The commit history was verified to confirm successful cherry-pick.

COMMANDS USED:

```
git log –oneline ,
git checkout -b feature1
notepad lab1.txt
git add lab1.txt
git commit -m "Commit for cherry-pick experiment"
git checkout main
git cherry-pick feature1
git log --oneline
```



```
85c0395 (HEAD -> main, tag: v1.0, origin/main, origin/HEAD) Experiment
2: Changes in feature branch
425db24 Experiment 1: Initial repository setup and first commit
749ab3e Initial commit

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$ git checkout -b feature1
Switched to a new branch 'feature1'

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (feature1)
$ notepad lab1.txt

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (feature1)
$ git add lab1.txt

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (feature1)
$ git commit -m "Commit for cherry-pick experiment"
[feature1 a3a93cd] Commit for cherry-pick experiment
 1 file changed, 4 insertions(+), 1 deletion(-)

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (feature1)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$ git cherry-pick feature1
[main e565db1] Commit for cherry-pick experiment
 Date: Tue Jan 6 12:03:51 2026 +0530
 1 file changed, 4 insertions(+), 1 deletion(-)

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$ git log --oneline
e565db1 (HEAD -> main) Commit for cherry-pick experiment
85c0395 (tag: v1.0, origin/main, origin/HEAD) Experiment 2: Changes in
feature branch
425db24 Experiment 1: Initial repository setup and first commit
749ab3e Initial commit

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$ |
```

EXPERIMENT 9: ANALYSING AND CHANGING GIT HISTORY

PROCEDURE:

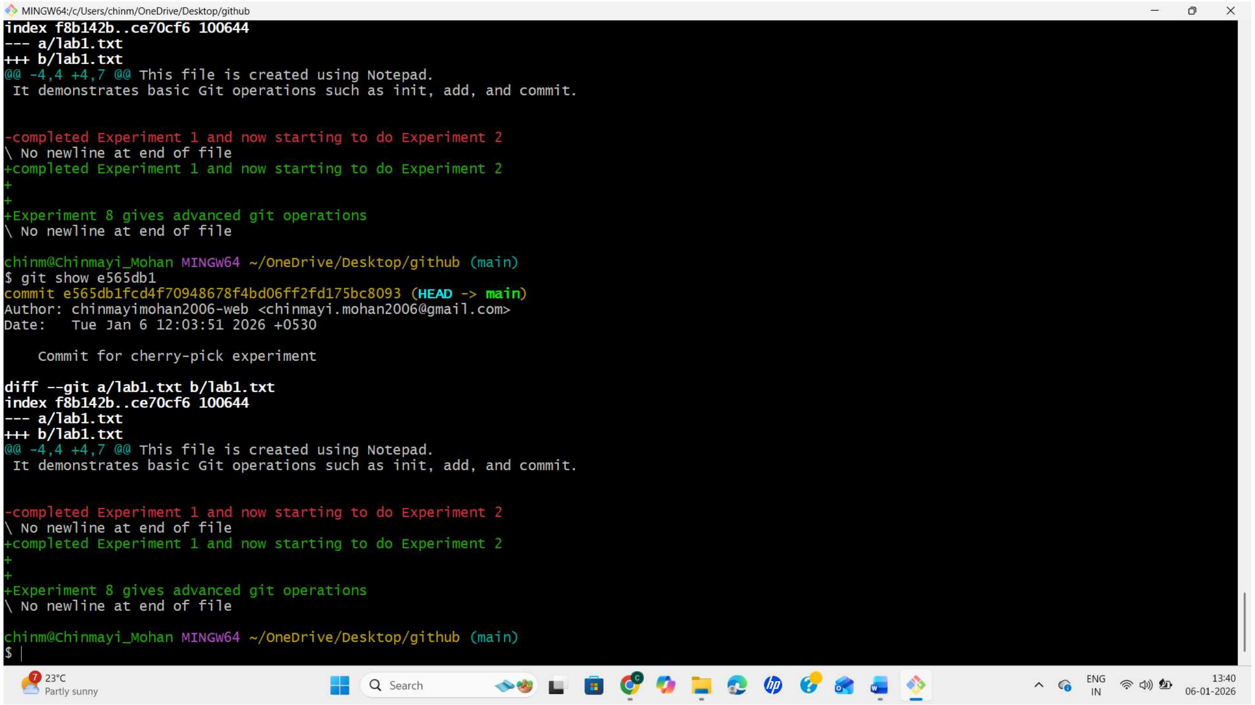
1. The commit history of the repository was viewed using the git log --oneline command to display concise commit information.
2. The complete commit history with author name, date, and commit message was viewed using the git log command.
3. A specific commit was selected from the history and its detailed information was displayed using the git show command.
4. The changes made in the selected commit were analyzed using the diff output.

COMMANDS USED:

git log –oneline ,

git log ,

git show <commit-id>



```
index f8b142b..ce70cf6 100644
--- a/lab1.txt
+++ b/lab1.txt
@@ -4,4 +4,7 @@ This file is created using Notepad.
 It demonstrates basic Git operations such as init, add, and commit.


-completed Experiment 1 and now starting to do Experiment 2
\ No newline at end of file
+completed Experiment 1 and now starting to do Experiment 2
+
+
+Experiment 8 gives advanced git operations
\ No newline at end of file

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$ git show e565db1
commit e565db1fcd4f70948678f4bd06ff2fd175bc8093 (HEAD -> main)
Author: chinmayimohan2006-web <chinmayi.mohan2006@gmail.com>
Date:   Tue Jan 6 12:03:51 2026 +0530

    Commit for cherry-pick experiment

diff --git a/lab1.txt b/lab1.txt
index f8b142b..ce70cf6 100644
--- a/lab1.txt
+++ b/lab1.txt
@@ -4,4 +4,7 @@ This file is created using Notepad.
 It demonstrates basic Git operations such as init, add, and commit.


-completed Experiment 1 and now starting to do Experiment 2
\ No newline at end of file
+completed Experiment 1 and now starting to do Experiment 2
+
+
+Experiment 8 gives advanced git operations
\ No newline at end of file

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$ 
```

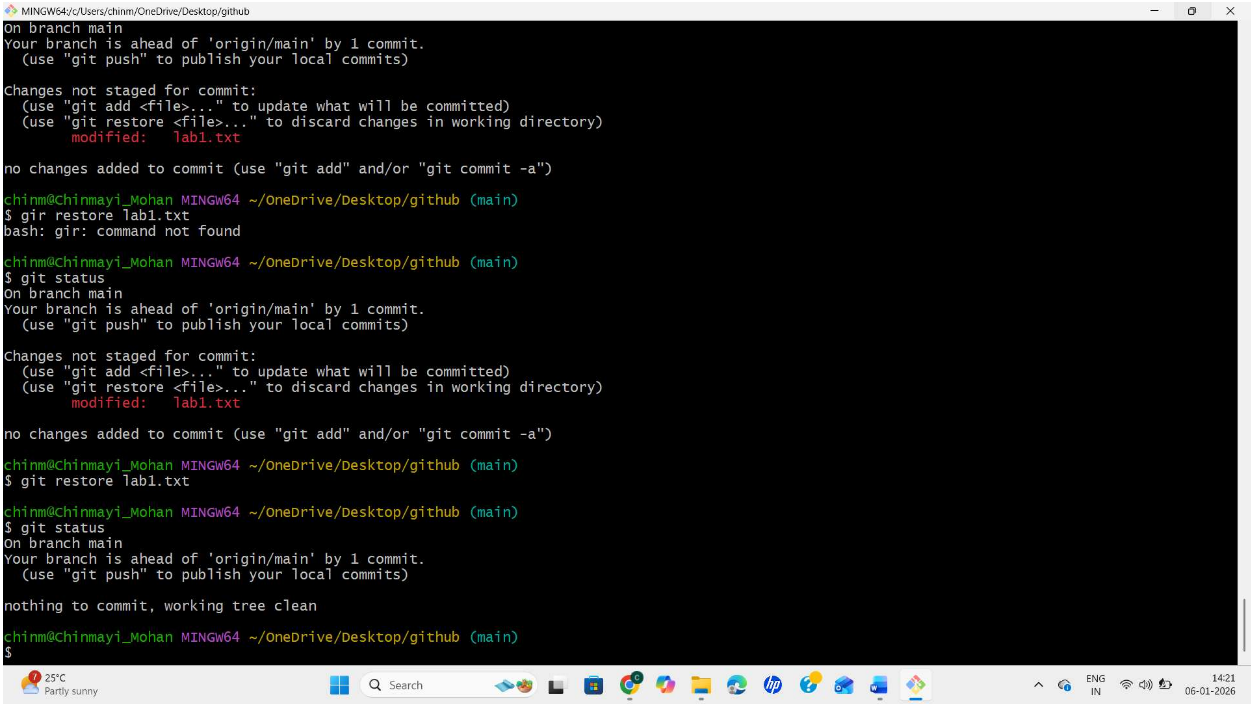
EXPERIMENT 10: ANALYSING AND CHANGING GIT HISTORY

PROCEDURE:

1. The current status of the repository was checked using the git status command.
2. The file lab1.txt was modified using Notepad.
3. The repository status was checked again to confirm that the file was modified.
4. The modified file was restored to its last committed state using the git restore lab1.txt command.
5. The repository status was verified to ensure that the working tree was clean.

COMMANDS USED:

```
git status

notepad lab1.txt

git status
```

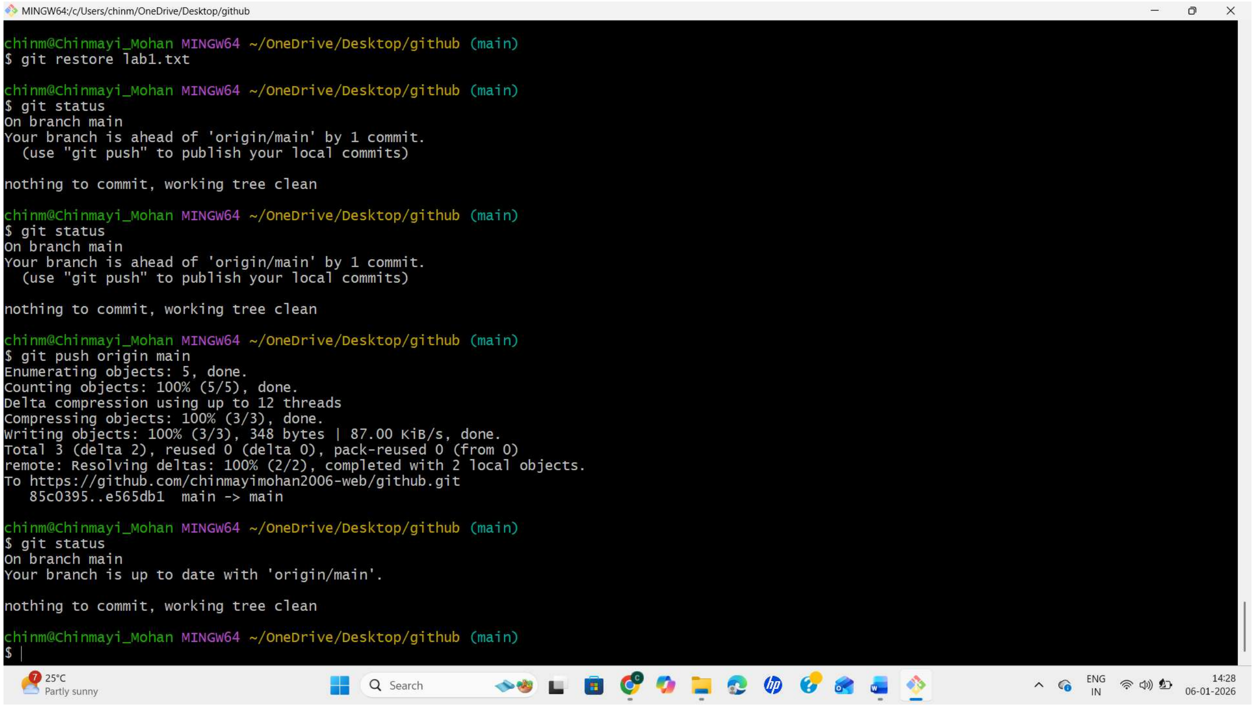


```
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

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:   lab1.txt

no changes added to commit (use "git add" and/or "git commit -a")

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$ gir restore lab1.txt
bash: gir: command not found

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

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:   lab1.txt

no changes added to commit (use "git add" and/or "git commit -a")

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$ git restore lab1.txt

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$
```

EXPERIMENT 11: ANALYSING AND CHANGING GIT HISTORY

PROCEDURE:

1. The current status of the local repository was checked using the git status command.
2. The committed changes were pushed to the remote GitHub repository using the git push origin main command.
3. After pushing, the status was verified to ensure the local and remote repositories were synchronized.
4. The updated files and commits were verified on the GitHub repository.



```
chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$ git restore lab1.txt

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 348 bytes | 87.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/chinmayimohan2006-web/github.git
   85c0395..e565db1  main -> main

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$ |
```

COMMAND USERS:

git status, git push origin

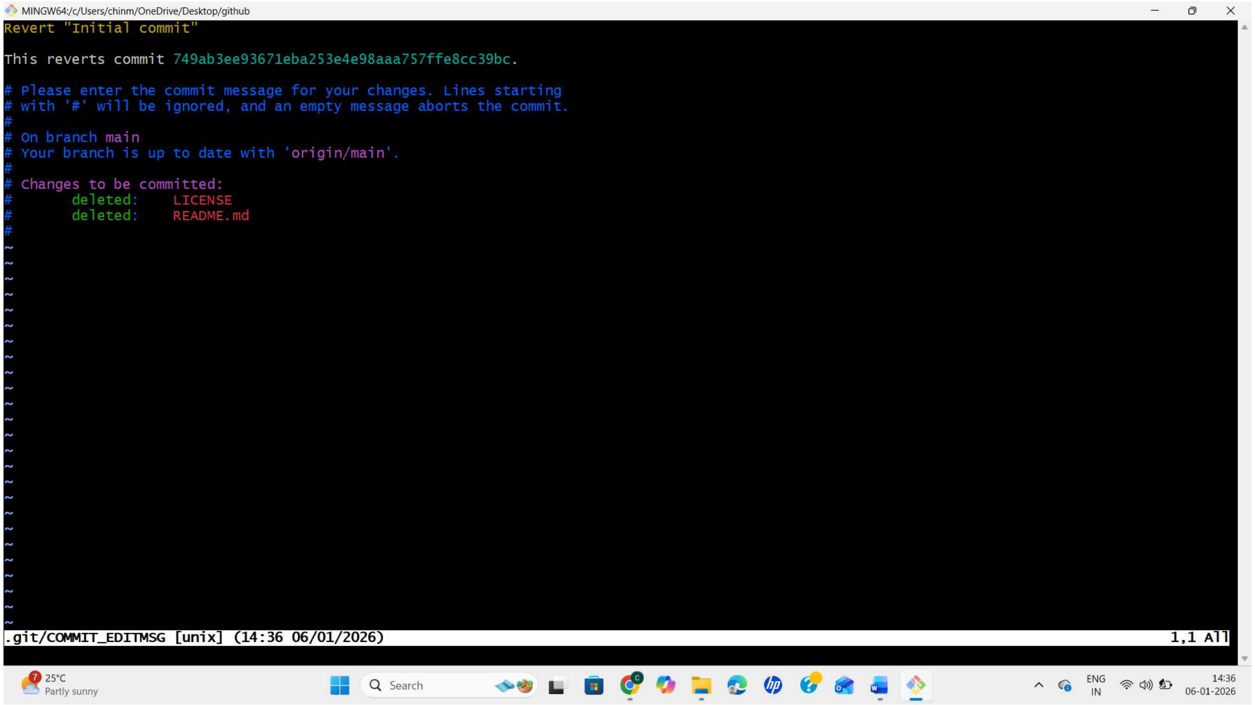
EXPERIMENT 12: ANALYSING AND CHANGING GIT HISTORY

PROCEDURE:

1. The commit history was viewed using the git log --oneline command.
2. The commit that needed to be undone was identified.
3. The git revert command was used to reverse the changes made by the selected commit.
4. A new revert commit was created without deleting previous commits.
5. The repository status and commit history were verified.
6. The revert commit was pushed to the remote GitHub repository

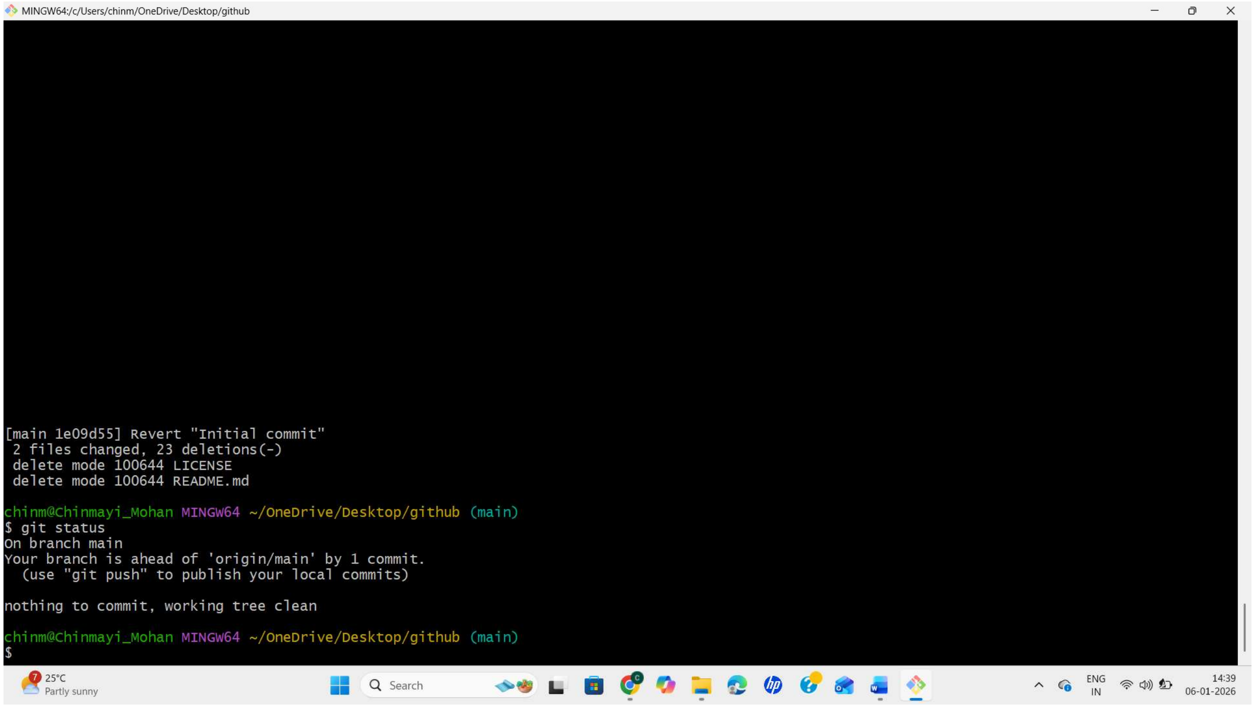
COMMANDS USED:

git log –oneline, git revert <commit-id>, git status, git pus git push origin main

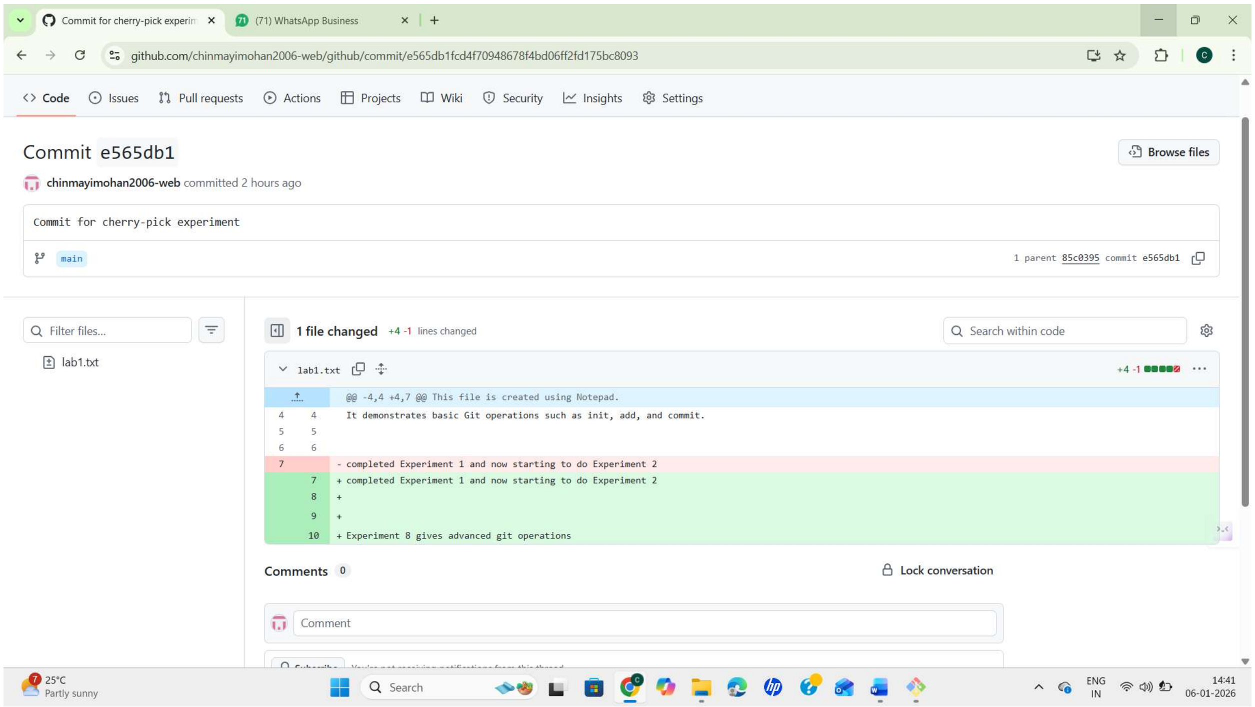


MINGW64:/c/Users/chinm/OneDrive/Desktop/github

```
Revert "Initial commit"

This reverts commit 749ab3ee93671eba253e4e98aaa757ffe8cc39bc.

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch main
# Your branch is up to date with 'origin/main'.
#
# Changes to be committed:
#       deleted:    LICENSE
#       deleted:    README.md
#
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
.git/COMMIT_EDITMSG [unix] (14:36 06/01/2026)                                1,1 All
```



```
[main 1e09d55] Revert "Initial commit"
 2 files changed, 23 deletions(-)
 delete mode 100644 LICENSE
 delete mode 100644 README.md

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

chinm@Chinmayi_Mohan MINGW64 ~/OneDrive/Desktop/github (main)
$
```



Commit for cherry-pick experim
(71) WhatsApp Business
github.com/chinmayimohan2006-web/github/commit/e565db1fcd4f70948678f4bd06ff2fd175bc8093
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
Browse files

Commit e565db1

chinmayimohan2006-web committed 2 hours ago

```
Commit for cherry-pick experiment
```

main 1 parent 85c0395 commit e565db1

lab1.txt

1 file changed +4 -1 lines changed

lab1.txt +4 -1

```
@@ -4,4 +4,7 @@ This file is created using Notepad.
 It demonstrates basic Git operations such as init, add, and commit.


- completed Experiment 1 and now starting to do Experiment 2
+ completed Experiment 1 and now starting to do Experiment 2
+
+
+ Experiment 8 gives advanced git operations
```

Comments 0