**Assignment-11:**

1. **Create a project under git repository and record multiple versions/branches and practice merging of branches**

To demonstrate creating a Git project, recording multiple versions/branches, and practicing merging, let's go through a simple example.

First, let's create a Git repository locally:

mkdir git\_practice

cd git\_practice

git init

Now, let's create a simple text file named example.txt and add some content to it:

echo "This is the initial content of example.txt" > example.txt

Add and commit this file to the main branch:

git add example.txt

git commit -m "Initial commit: Added example.txt"

Now, let's create a new branch named feature\_branch:

git checkout -b feature\_branch

Add some changes to example.txt:

echo "This is an additional line from feature\_branch" >> example.txt

Add and commit these changes:

git add example.txt

git commit -m "Added a line from feature\_branch"

Now, let's switch back to the main branch:

git checkout main

Make some changes to example.txt on the main branch:

echo "This is a change from the main branch" >> example.txt

Add and commit these changes:

git add example.txt

git commit -m "Made a change from main branch"

Now, let's merge the feature\_branch into main:

git merge feature\_branch

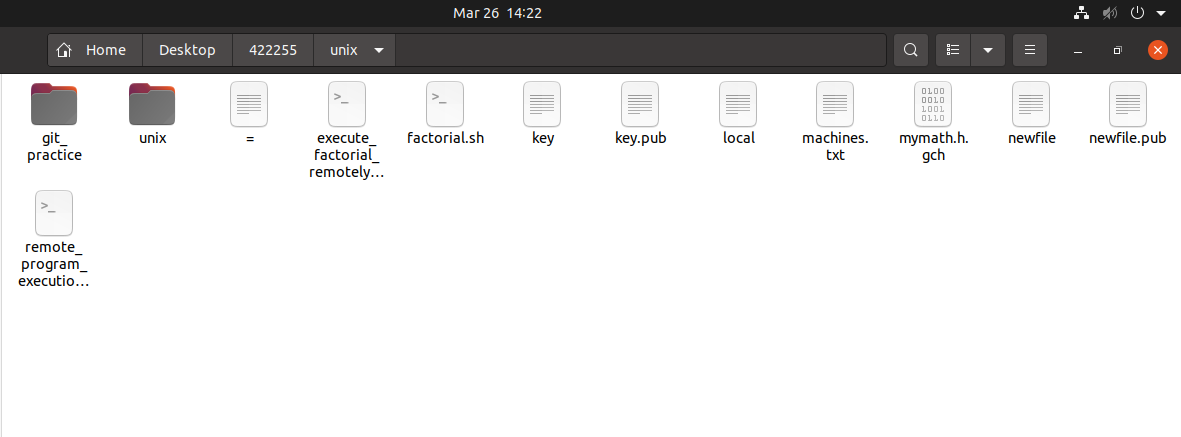
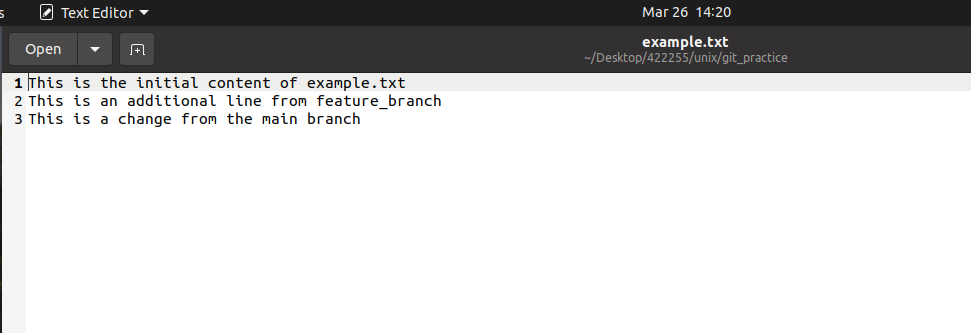
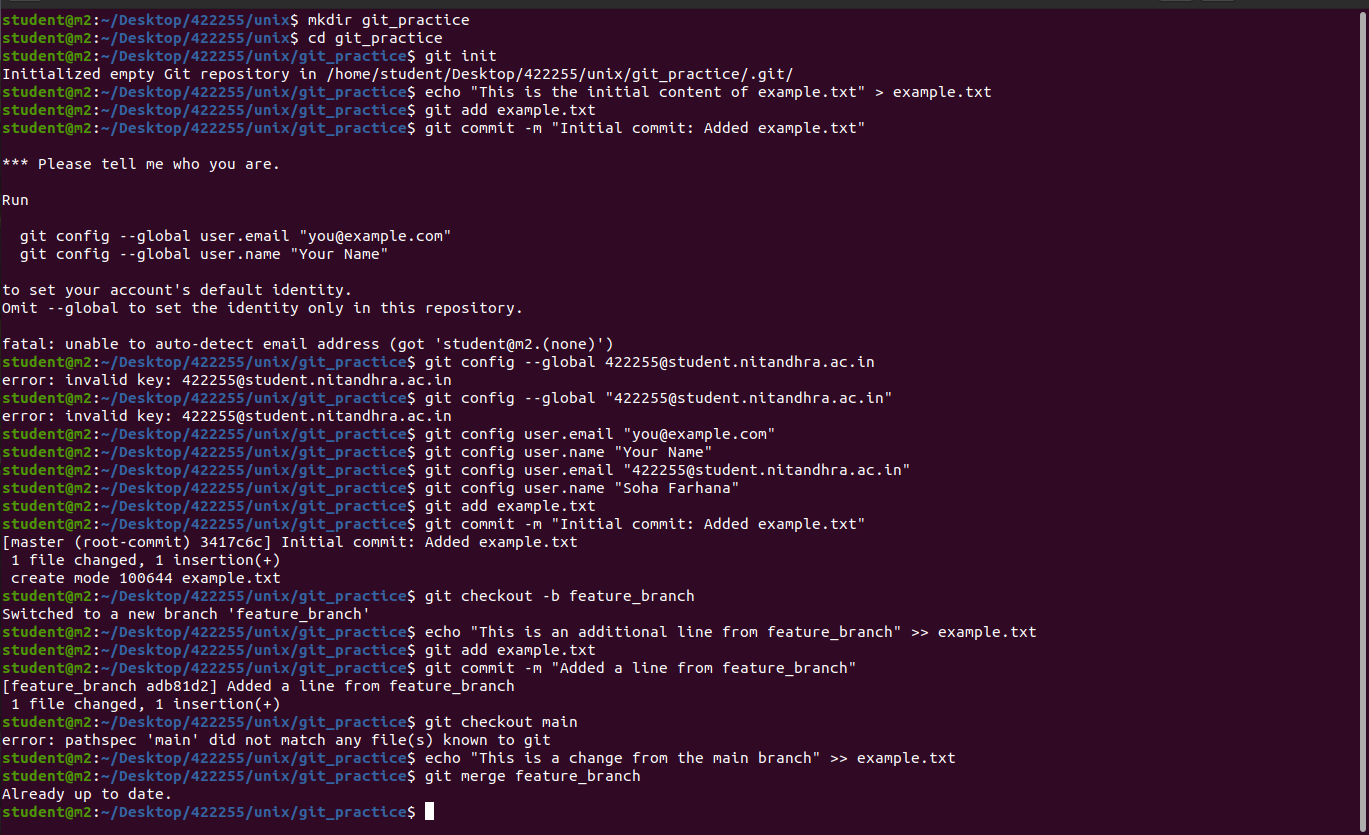
Git will try to merge the changes automatically. If there are no conflicts, Git will perform a fast-forward merge. If there are conflicts, you'll need to resolve them manually.

After merging, you can view the history of commits:

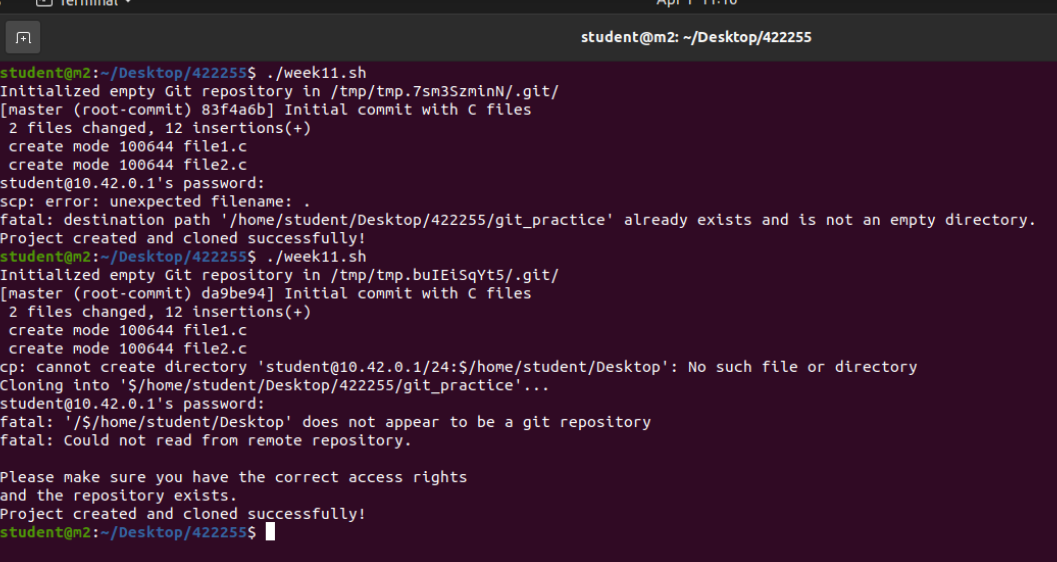
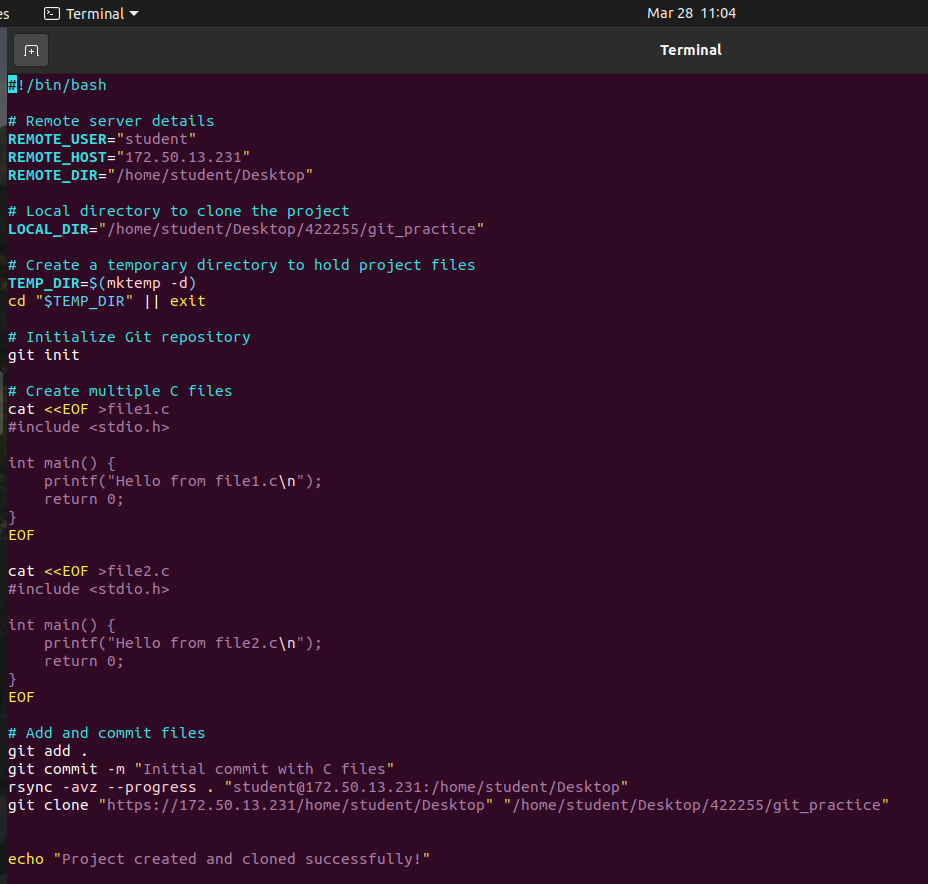
git log --oneline --decorate --graph --all



This will show you a graphical representation of the commit history, including all branches.

****

1. **Create a shell script that creates a git project with different c files in a remote server and clone the project into the local machine.**

****