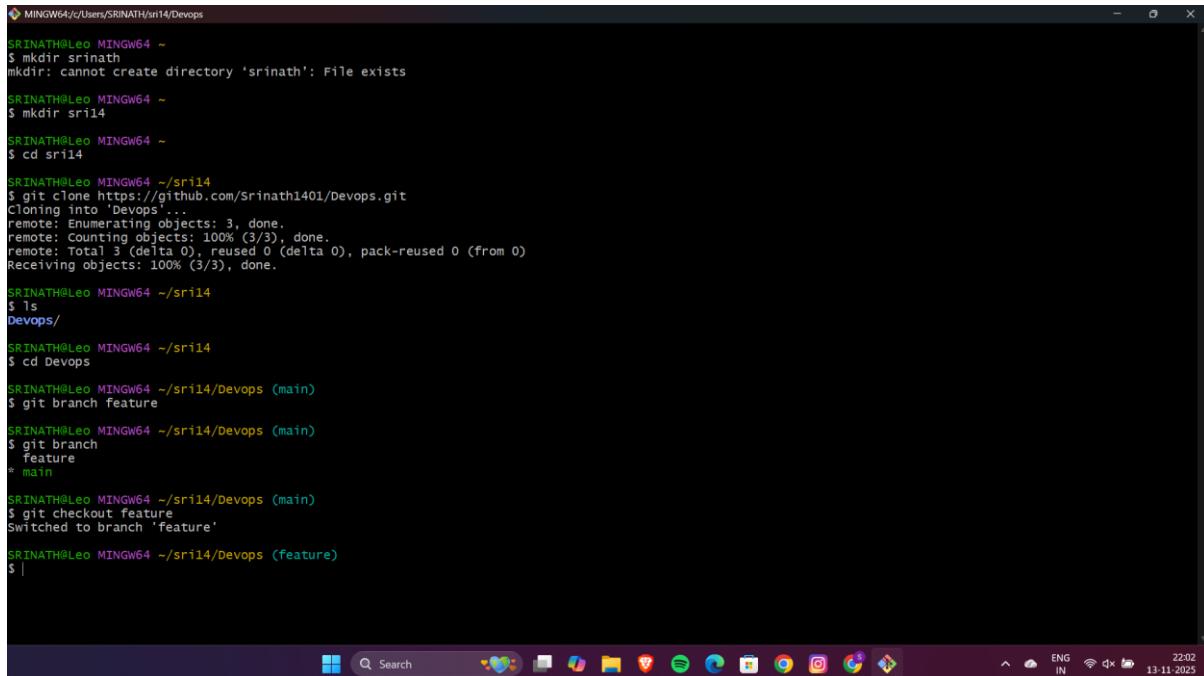


Task 2

Git Branch:

In Git, a branch is like a separate workspace where you can make changes and try new ideas without affecting the main project.



```
MINGW64:/c/Users/SRINATH/sri14/Devops
$ mkdir srinath
mkdir: cannot create directory 'srinath': File exists

SRINATH@Leo MINGW64 ~
$ mkdir sri14

SRINATH@Leo MINGW64 ~
$ cd sri14

SRINATH@Leo MINGW64 ~/sri14
$ git clone https://github.com/Srinath1401/Devops.git
Cloning into 'Devops'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.

SRINATH@Leo MINGW64 ~/sri14
$ ls
Devops/

SRINATH@Leo MINGW64 ~/sri14
$ cd Devops

SRINATH@Leo MINGW64 ~/sri14/Devops (main)
$ git branch feature
* main

SRINATH@Leo MINGW64 ~/sri14/Devops (main)
$ git checkout feature
Switched to branch 'feature'

SRINATH@Leo MINGW64 ~/sri14/Devops (feature)
$ |
```

Git Checkout:

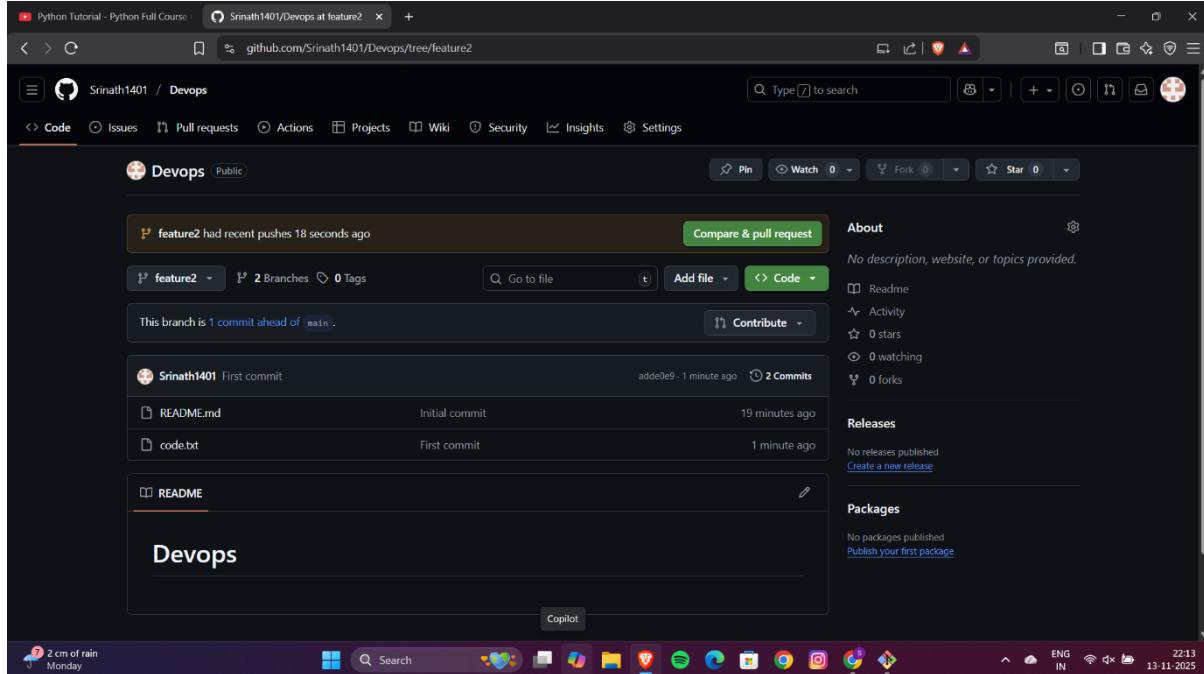
```
SRINATH@Leo MINGW64 ~/sri14/Devops (main)
$ git checkout feature
Switched to branch 'feature'

SRINATH@Leo MINGW64 ~/sri14/Devops (feature)
$ git checkout -b feature2
Switched to a new branch 'feature2'

SRINATH@Leo MINGW64 ~/sri14/Devops (feature2)
$ git branch
  feature
* feature2
  main
```

Git Commit -am:

Before



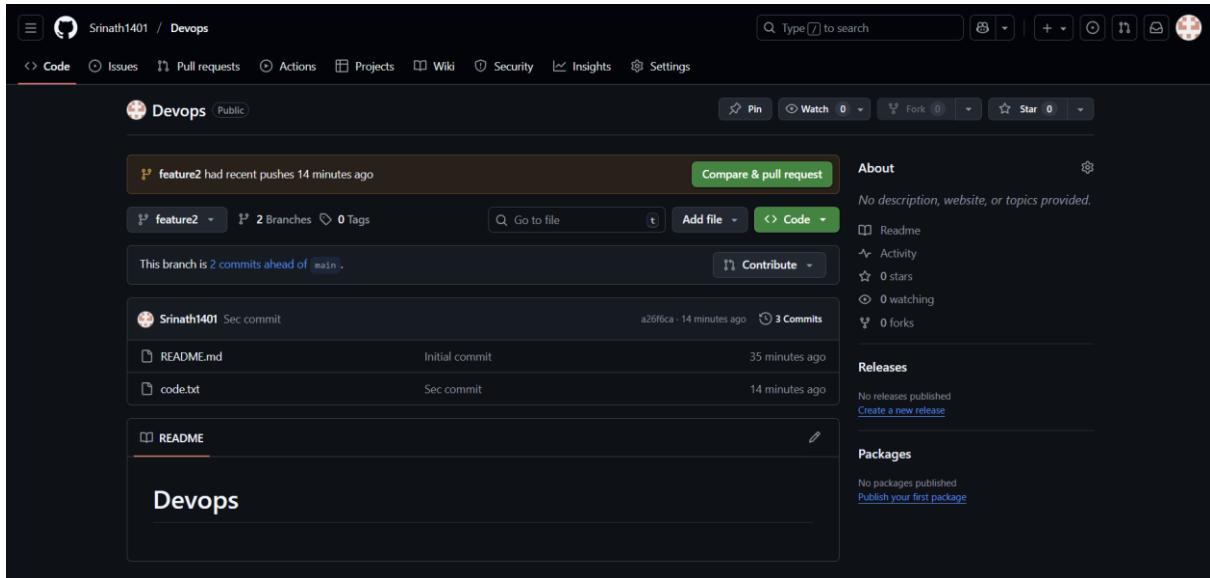
Commands:

```
SRINATH@Leo MINGW64 ~/sri14/Devops (feature2)
$ vi code.txt

SRINATH@Leo MINGW64 ~/sri14/Devops (feature2)
$ git commit -am "Sec commit"
warning: in the working copy of 'code.txt', LF will be replaced by CRLF the next time Git touches it
[feature2 a26f6ca] Sec commit
 1 file changed, 1 insertion(+), 1 deletion(-)

SRINATH@Leo MINGW64 ~/sri14/Devops (feature2)
$ git push origin feature2
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% (3/3), 293 bytes | 293.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Srinath1401/Devops.git
  addde0e9..a26f6ca  feature2 -> feature2
```

After



Git Merge:

Git Merge is a command used to combine the changes from two branches into one. It integrates work from different branches into a single unified history without losing progress. For example, you can merge a feature branch into the main branch to include all recent updates.

```
SRINATH@Leo MINGW64 ~/sri14/Devops (feature)
$ git log --oneline
8edefb5 (HEAD -> feature, origin/main, origin/HEAD, main) Initial commit

SRINATH@Leo MINGW64 ~/sri14/Devops (feature)
$ git merge feature2
Updating 8edefb5..a26f6ca
Fast-forward
 code.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 code.txt

SRINATH@Leo MINGW64 ~/sri14/Devops (feature)
$ git log --oneline
a26f6ca (HEAD -> feature, origin/feature2, feature2) Sec commit
adde0e9 First commit
8edefb5 (origin/main, origin/HEAD, main) Initial commit

SRINATH@Leo MINGW64 ~/sri14/Devops (feature)
$ git log --oneline
a26f6ca (HEAD -> feature, origin/feature2, feature2) Sec commit
adde0e9 First commit
8edefb5 (origin/main, origin/HEAD, main) Initial commit
```