PROJECT MANAGEMENT WITH GIT :

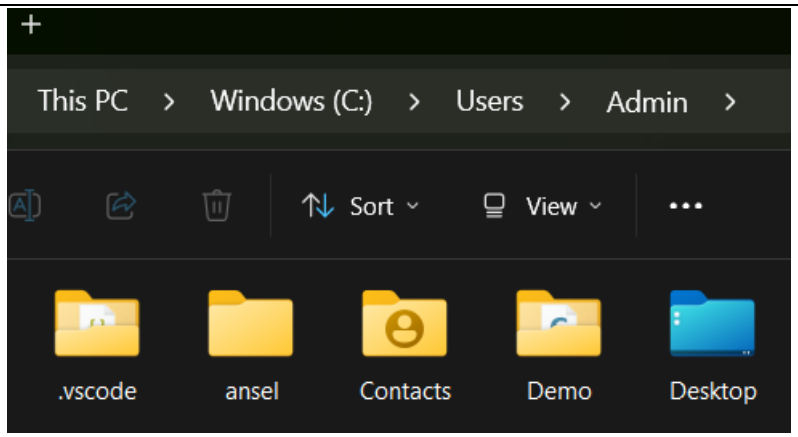
- **Initializing a GIT repository command :**

  Step 1:

  Creating a folder using command:

$mkdir Demo;cd Demo

```
sinch@Sinchana MINGW64 ~
$ mkdir Demo;cd Demo
mkdir: cannot create directory ‘Demo’: File exists
```



Step 2:

$git init

```
sinch@Sinchana MINGW64 ~/Demo
$ git init
Initialized empty Git repository in C:/Users/sinch/Demo/.git/
```

- git init initializes a empty git repository.

2)Creating a file and adding it to the staging area:

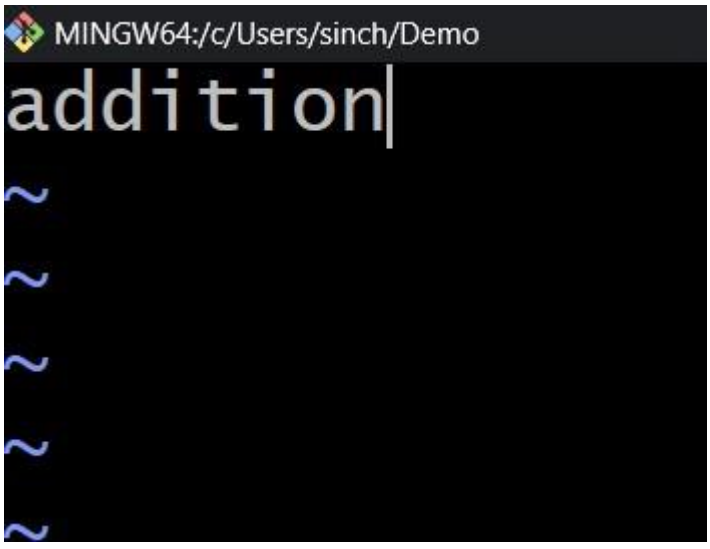
Command:

---

git add<file_name.>

```
sinch@Sinchana MINGW64 ~/Demo (main)
$ vi lord.c

sinch@Sinchana MINGW64 ~/Demo (main)
$ git add lord.c
warning: in the working copy of 'lord.c', LF will be replaced by CRLF the next time Git touche
s it
```

- first we create file and added to the git
- 

- git add . will stage the whole files in the current directory and are ready to be commited

---

3)Checking status of repository:

Command:

git status

- This command tells current state of the repository

4)Commiting Changes to a repository:

Command**:**

Git commit -m <message>

```
sinch@Sinchana MINGW64 ~/Demo (main)
$ git commit -m"ganesh"
[main (root-commit) bfa249e] ganesh
 1 file changed, 1 insertion(+)
 create mode 100644 lord.c

sinch@Sinchana MINGW64 ~/Demo (main)
$ git config --global user.name"sinchana"

sinch@Sinchana MINGW64 ~/Demo (main)
$ git config --global user.email"sinchu@123.com"
```

- It makes sure that changes are saved to lacal repository

5)Creating a new branch:

Command:

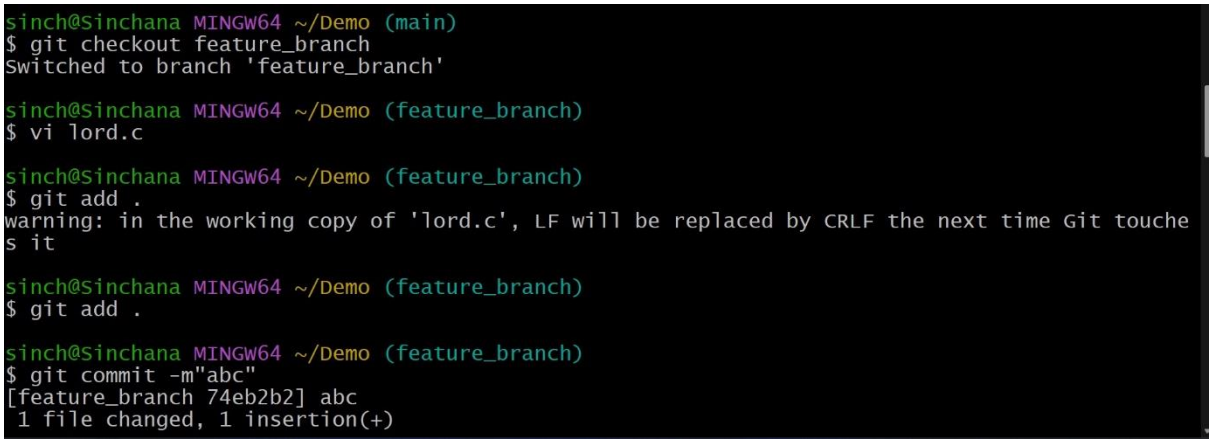
git branch <new_branch_name>

```
sinch@Sinchana MINGW64 ~/Demo (main)
$ git branch feature_branch

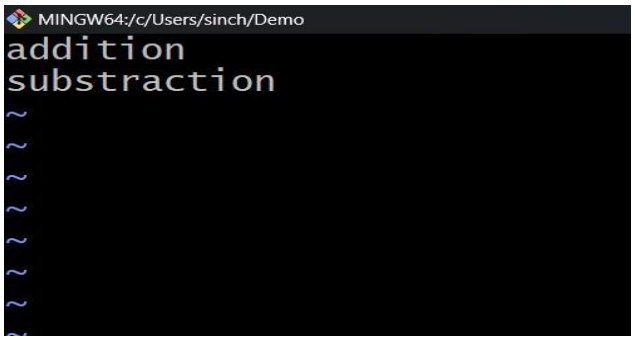
sinch@Sinchana MINGW64 ~/Demo (main)
$ git branch
  feature_branch
* main
```

- git branch feature_branch this will create a new branch of the main master branch in which the contents and files are copied from the master branch .
- git branch → this command will show all the branches we have made and the current branch will be in green colour.

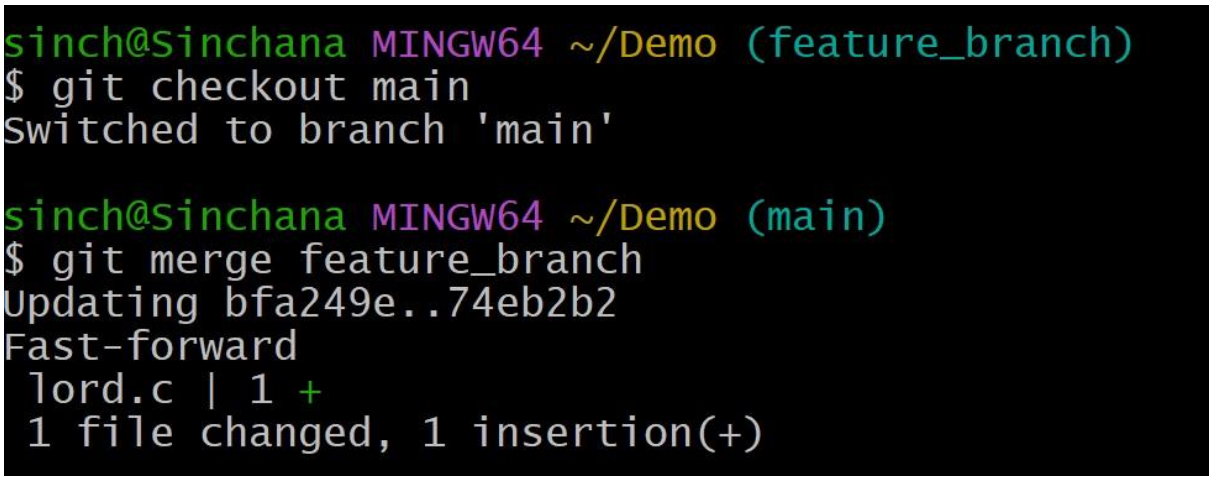
Checkout branch:



Making changes in the new branch and commiting it:



- git checkout :it is used to switch one branch to other branch ,here we are moving from branch master to the feature-branch.
- And also we have edited the file text.txt.
- Here in the feature-branch we staged the file and commited with message saying “xyz”.
- So here the file has been changed ,but in the master branch it will be as it is until we merge the feature-branch with the master branch



- For merging the file to the master branch we first need to move to the main/master branch using “git checkout master” command.
- Then we can merge the branch with “git merge feature-branch” command

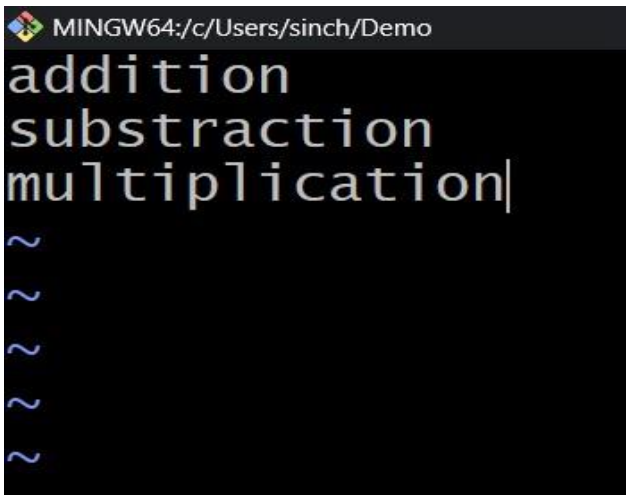
- So ,now the files will be meged ,the changes or the edits in the feature-branch will be merged.

Moving back to master branch and making separate changes:

```
sinch@Sinchana MINGW64 ~/Demo (main)
$ vi lord.c

sinch@Sinchana MINGW64 ~/Demo (main)
$ git add .

sinch@Sinchana MINGW64 ~/Demo (main)
$ git commit -m"xyz"
[main d479082] xyz
 1 file changed, 1 insertion(+)
```



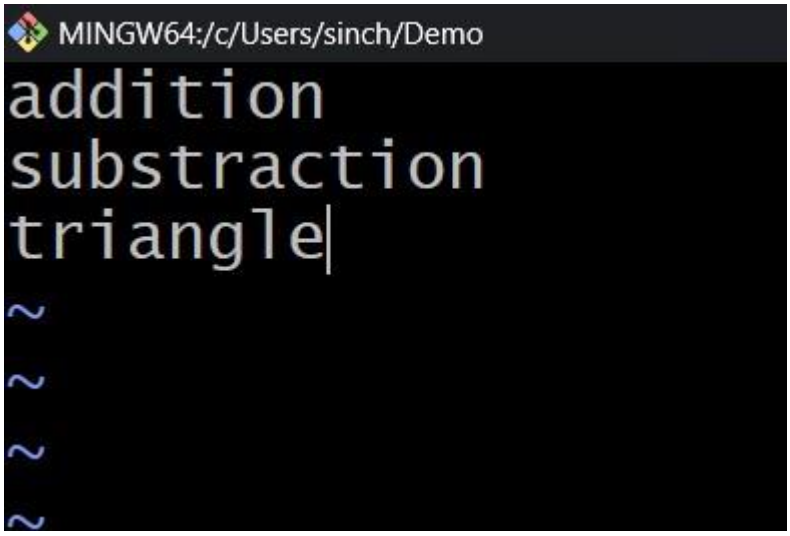
Moving back to feature_branch and making separate changes:

```
sinch@Sinchana MINGW64 ~/Demo (feature_branch)
$ git checkout feature_branch
Already on 'feature_branch'

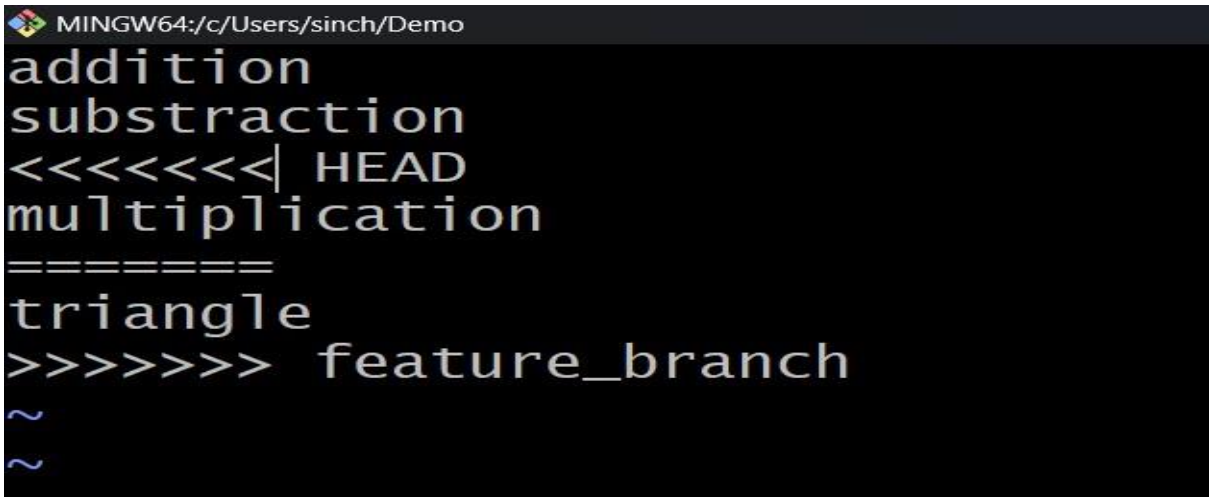
sinch@Sinchana MINGW64 ~/Demo (feature_branch)
$ vi lord.c

sinch@Sinchana MINGW64 ~/Demo (feature_branch)
$ git add .

sinch@Sinchana MINGW64 ~/Demo (feature_branch)
$ git commit -m"ccc"
[feature_branch a63db8a] ccc
 1 file changed, 1 insertion(+)
```



Merging the two branches:




```
sinch@Sinchana MINGW64 ~/Demo (main|MERGING)
$ vi lord.c

sinch@Sinchana MINGW64 ~/Demo (main|MERGING)
$ git add .

sinch@Sinchana MINGW64 ~/Demo (main|MERGING)
$ git commit -m"aaa"
[main 25688f1] aaa

sinch@Sinchana MINGW64 ~/Demo (main)
$ git commit -m"merged branches"
On branch main
nothing to commit, working tree clean
```

git log:

- Shows the commit history

```
sinch@Sinchana MINGW64 ~/Demo (main)
$ git log
commit 25688f142bfc92c46cb23f7cc46342b6c934109e (HEAD -> main)
Merge: d479082 a63db8a
Author: sinchu <sinchu@sinchu.com>
Date:   Mon Feb 26 23:25:16 2024 +0530

    aaa

commit a63db8a309165af2fed80707d70d926c2eebe9e4 (feature_branch)
Author: sinchu <sinchu@sinchu.com>
Date:   Mon Feb 26 23:23:06 2024 +0530

    ccc

commit d479082285955f7e87adfc367d3b07850d48a685
Author: sinchu <sinchu@sinchu.com>
Date:   Mon Feb 26 23:10:32 2024 +0530

    xyz
```

- ```
  commit 74eb2b2d3985bd3fa37397160d8ff5f30ba9d6ca
  Author: sinchu <sinchu@sinchu.com>
  Date:   Mon Feb 26 23:07:27 2024 +0530

      abc

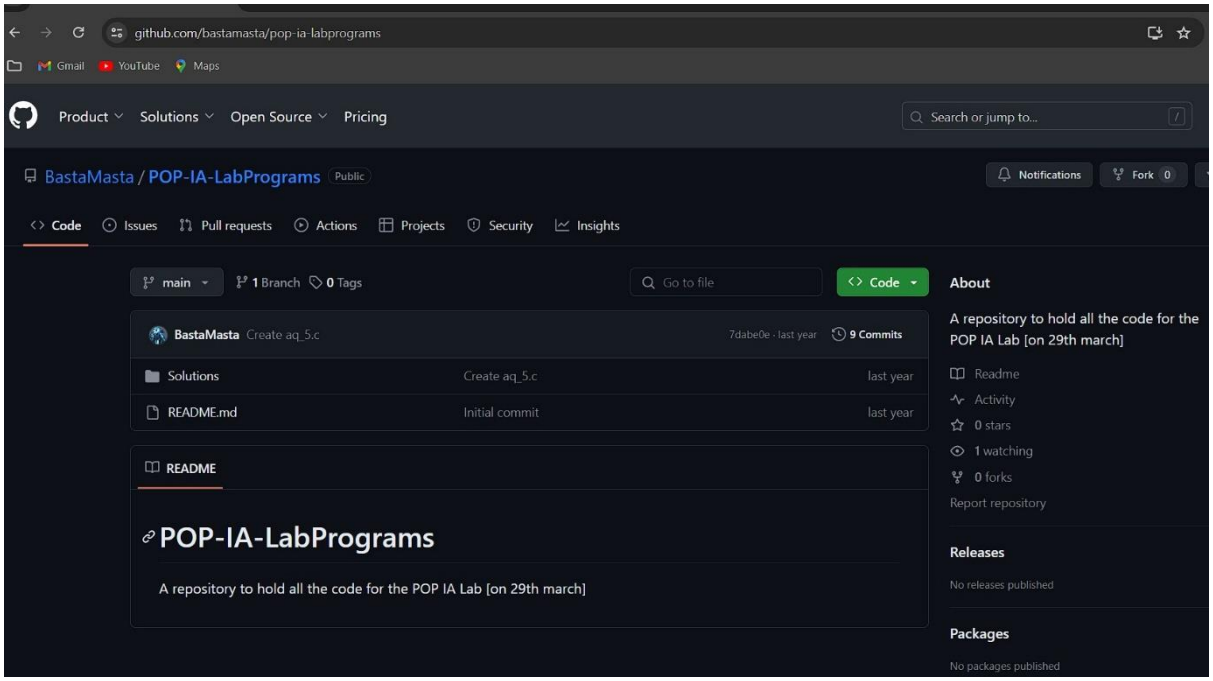
  commit bfa249e0770ed26bb99917f7a2ca9a53e72da8cb
  Author: sinchu <sinchu@sinchu.com>
  Date:   Mon Feb 26 23:03:12 2024 +0530

      ganesh
  ```
- (END)

Clonig a remote repository:

Command:

git clone <remote repo url>
  ```




```
sinch@Sinchana MINGW64 ~
$ git clone https://github.com/BastaMasta/POP-IA-LabPrograms.git
Cloning into 'POP-IA-LabPrograms'...
remote: Enumerating objects: 35, done.
remote: Counting objects: 100% (35/35), done.
remote: Compressing objects: 100% (33/33), done.
remote: Total 35 (delta 5), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (35/35), 9.40 KiB | 3.13 MiB/s, done.
Resolving deltas: 100% (5/5), done.

sinch@Sinchana MINGW64 ~
$ cd pop-IA-Labprograms
```

Fetch latest chages from clong repository:

Command:

git fetch

OR

git fetch origin

```
sinch@Sinchana MINGW64 ~/pop-IA-Labprograms (main)
$ git fetch
```

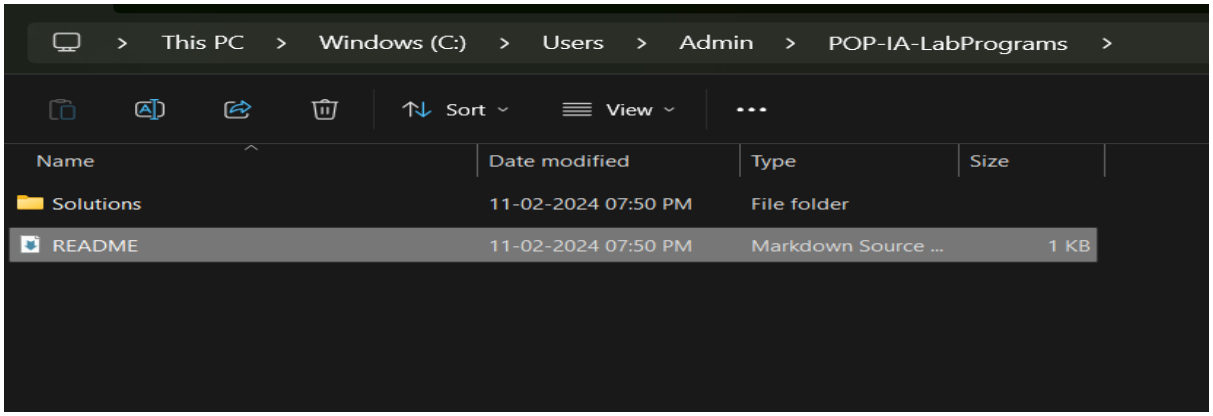
Pull changes from remote repo and rebase the local branch:

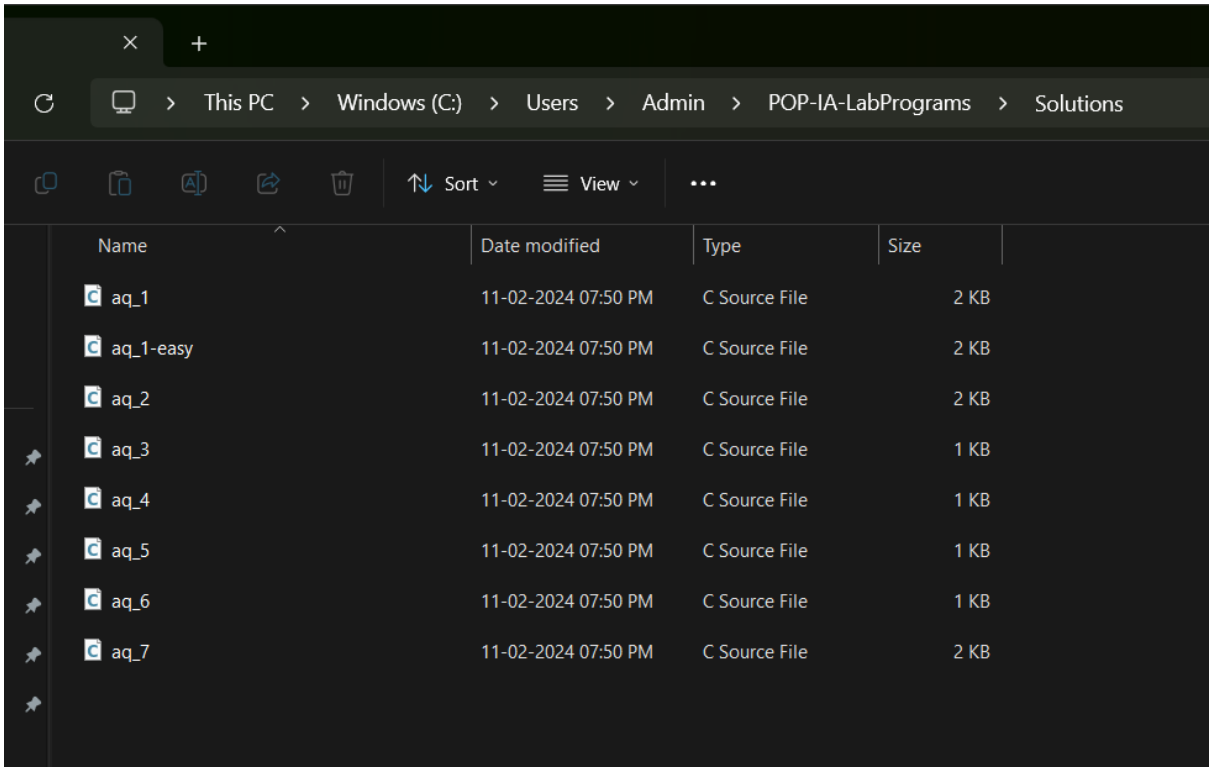
Command:

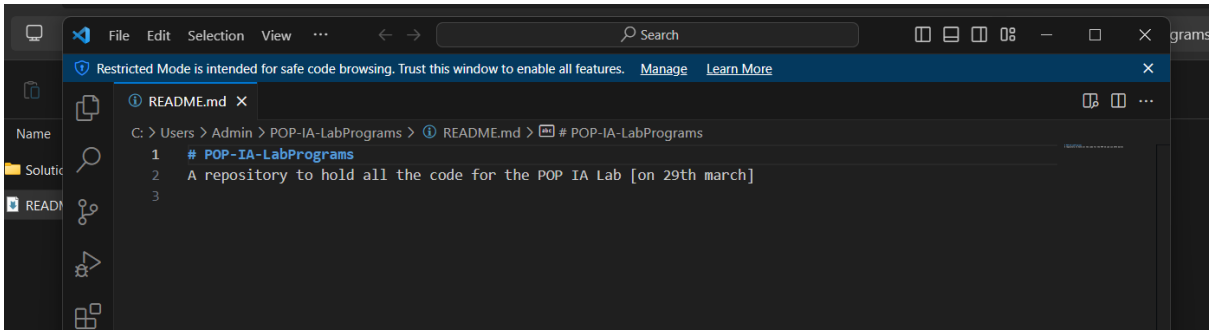
git config –global pull.rebase “true”

```
sinch@Sinchana MINGW64 ~/pop-IA-Labprograms (main)
$ git config --global pull.rebase"true"

sinch@Sinchana MINGW64 ~/pop-IA-Labprograms (main)
$ git pull
Already up to date.
```




This PC
Windows (C:)
Users
Admin
POP-IA-LabPrograms
Solutions
Sort
View
Name
Date modified
Type
Size
aq_1 11-02-2024 07:50 PM C Source File 2 KB
aq_1-easy 11-02-2024 07:50 PM C Source File 2 KB
aq_2 11-02-2024 07:50 PM C Source File 2 KB
aq_3 11-02-2024 07:50 PM C Source File 1 KB
aq_4 11-02-2024 07:50 PM C Source File 1 KB
aq_5 11-02-2024 07:50 PM C Source File 1 KB
aq_6 11-02-2024 07:50 PM C Source File 1 KB
aq_7 11-02-2024 07:50 PM C Source File 2 KB


File Edit Selection View
Search
Restricted Mode is intended for safe code browsing. Trust this window to enable all features.
Manage
Learn More
README.md
C: > Users > Admin > POP-IA-LabPrograms > README.md > # POP-IA-LabPrograms
1 # POP-IA-LabPrograms
2 A repository to hold all the code for the POP IA Lab [on 29th march]
3