PURBANCHAL UNIVERSITY

KHWOPA ENGINEERING COLLEGE

LIBALI-08, BHAKTAPUR

LAB REPORT ON: .NET

LAB NO: 01

SUBMITTED BY:

Name: Arbin Thaku

Roll No.: 770305

SUBMITTED TO:

Department of Computer Engineering

Submission: 2081/12/10

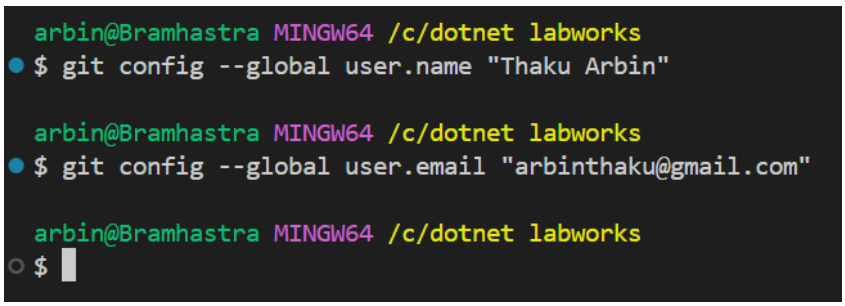
Theory:

1. **Git:**
   Git is a distributed version control system used for tracking the changes in the source code during software development. It allows multiple developers to collaborate efficiently by managing different versions of projects. Git enables branching, merging and reverting changes, making code management easier. It is widely used in open-source and commercial projects. Popular platforms like GitHub, GitLab, and Bitbucket provide remote repositories for Git-based collaboration.

2. **GitHub**
   GitHub is a web-based platform for version control and collaboration using Git. It allows developers to store, manage, and share code repositories efficiently. GitHub supports features like branching, pull requests, issue tracking, and CI/CD integration. It is widely used for open-source and private projects, enabling seamless teamwork. GitHub also provides cloud-based hosting, making it accessible from anywhere.

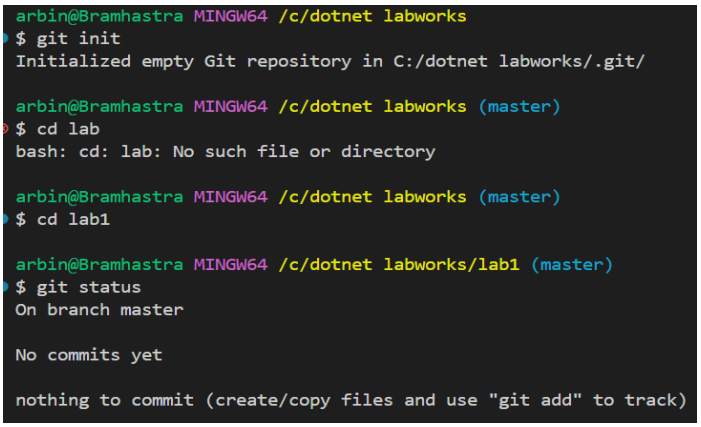
Lab Works

First set the global username and email of the GitHub.



Create a folder and inside it files as per the user desire so that we can identify the changes inside the file using the version control (Git).

On creating the new files, initially the files are in the untracked stage so send the untracked files to the staging stage. To do so first initialize the directory and stage the files.



```
arbin@Bramhastra MINGW64 /c/dotnet labworks/lab1 (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        ./

nothing added to commit but untracked files present (use "git add" to
 track)
```

Now commit the files such that the files are stored in the local repository.

```
arbin@Bramhastra MINGW64 /c/dotnet labworks/lab1 (master)
$ git add .

arbin@Bramhastra MINGW64 /c/dotnet labworks/lab1 (master)
$ git commit -m "initial commit"
[master (root-commit) 0e045a3] initial commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 lab1/test.py
```

Make certain changes inside the file to see the changes in the file status.

```
arbin@Bramhastra MINGW64 /c/dotnet labworks/lab1 (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working director
y)
        modified:   test.py

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

After changing the contents in the file **"test.py"** add the file and commit it.
All these files are saved in the local repository. Now to add these files in the remote repository create the repository in the GitHub and copy the url of the repo and use the following code.

```
arbin@Bramhastra MINGW64 /c/dotnet labworks/lab1 (master)
$ git remote add origin https://github.com/arbinthaku5/dotnet-lab.git
```

Now push the files in the repository created.

```
arbin@Bramhastra MINGW64 /c/dotnet labworks/lab1 (master)
$ git push -u origin master
info: please complete authentication in your browser...
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (8/8), 547 bytes | 273.00 KiB/s, done.
Total 8 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/arbinthaku5/dotnet-lab.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

Now creating branches, allowing the work on different versions of a project without affecting the main codebase.

```
arbin@Bramhastra MINGW64 /c/dotnet labworks/lab1 (master)
$ git branch feature

arbin@Bramhastra MINGW64 /c/dotnet labworks/lab1 (master)
$ git checkout feature
Switched to branch 'feature'

arbin@Bramhastra MINGW64 /c/dotnet labworks/lab1 (feature)
$ git branch
* feature
  master
```

Moving on to the recently created branch to modify the contents in the file without affecting the main codebase.

```
arbin@Bramhastra MINGW64 /c/dotnet labworks/lab1 (feature)
$ git status
On branch feature
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working director
y)
        modified:   test.py

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

```
arbin@Bramhastra MINGW64 /c/dotnet labworks/lab1 (feature)
$ git add .

arbin@Bramhastra MINGW64 /c/dotnet labworks/lab1 (feature)
$ git commit -m "included substraction in feature branch"
[feature b6c8337] included substraction in feature branch
 1 file changed, 2 insertions(+), 1 deletion(-)
```

```
arbin@Bramhastra MINGW64 /c/dotnet labworks/lab1 (feature)
$ git push -u origin feature
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 347 bytes | 347.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'feature' on GitHub by visiting:
remote:      https://github.com/arbinthaku5/dotnet-lab/pull/new/featu
re
remote:
To https://github.com/arbinthaku5/dotnet-lab.git
 * [new branch]      feature -> feature
branch 'feature' set up to track 'origin/feature'.
```

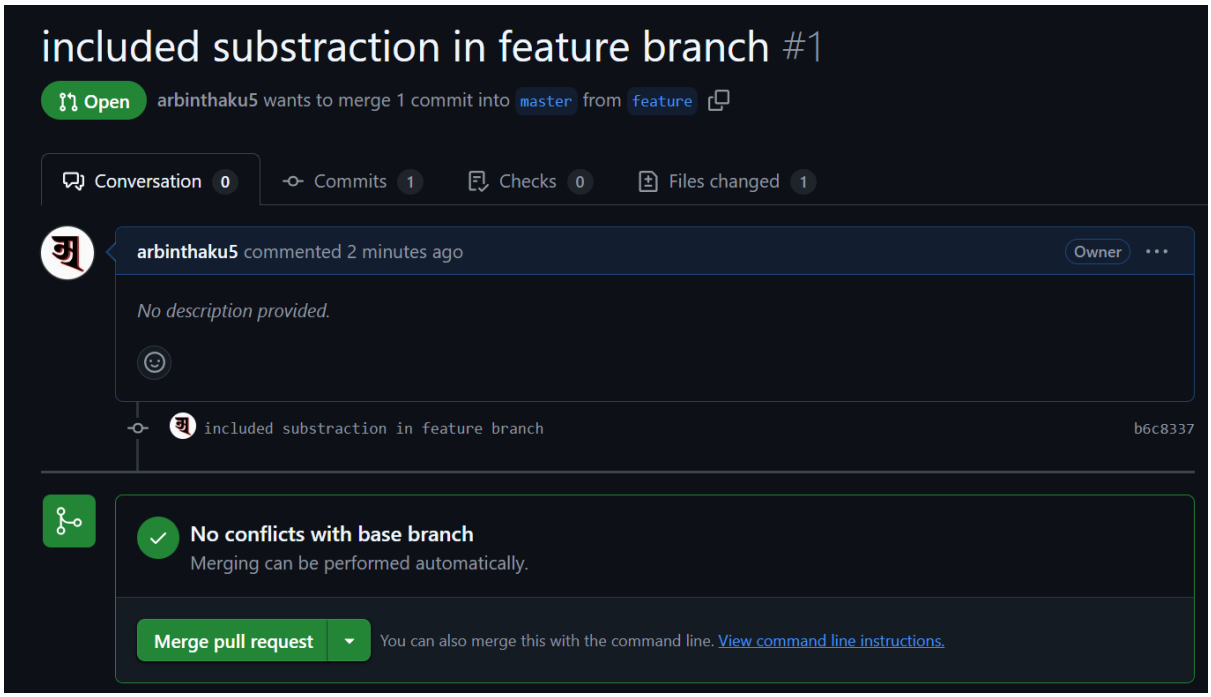
To change the branch, we can use the command *"git checkout master"*. To make sure the branch is visible to other users of the repository push the branch into the GitHub.
Merging the branches such that the changes in the new branch is added to the main code base.

```
arbin@Bramhastra MINGW64 /c/dotnet labworks/lab1 (feature)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

arbin@Bramhastra MINGW64 /c/dotnet labworks/lab1 (master)
$ git merge feature
Updating 323a20e..b6c8337
Fast-forward
 lab1/test.py | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)
```

To check the commits performed in the past

```
arbin@Bramhastra MINGW64 /c/dotnet labworks/lab1 (master)
$ git log
commit b6c833714e4f567fef6208834542440fa4c7f09d (HEAD -> master, orig
in/feature, feature)
Author: Thaku Arbin <arbinthaku@gmail.com>
Date:   Sat Mar 22 21:08:15 2025 +0545

    included substraction in feature branch

commit 323a20ed9cc5273b9b2c41cf09657e4bb532bccd (origin/master)
Author: Thaku Arbin <arbinthaku@gmail.com>
Date:   Sat Mar 22 21:03:02 2025 +0545

    added/multiply

commit 0e045a32862c913fc95483a6ef877e377500a854
Author: Thaku Arbin <arbinthaku@gmail.com>
Date:   Sat Mar 22 20:58:05 2025 +0545

    initial commit
```

Merging the branch in the GUI GitHub (Web)



Conclusion:

In this lab, we learn about the basics of Git and GitHub. We perform initialization, branching, merging, pushing and committing and are hosted in this repo.