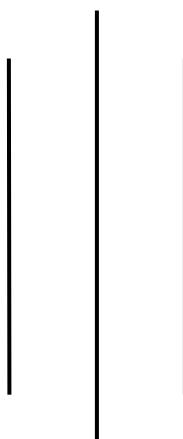


PURBANCHAL UNIVERSITY



KHWOPA ENGINEERING COLLEGE

LIBALI-08, BHAKTAPUR



LAB REPORT ON: .NET

LAB NO: 01

SUBMITTED BY:

Name: Buddha Tamang

Roll No.: 770308

SUBMITTED TO:

Department of Computer Engineering

Submission: 2081/12/10

Theory:

1. Git:

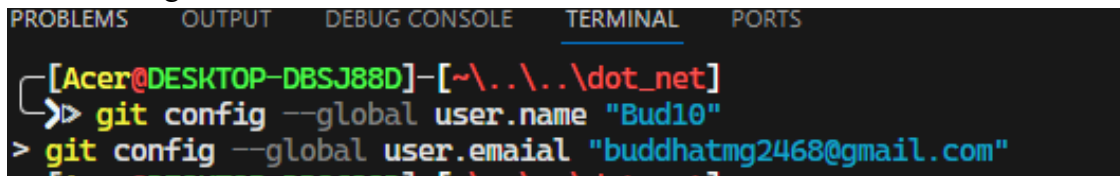
Git is a distributed version control system used for tracking the changes in the source code during software development. It allows multiple developers to collaborate efficiently by managing different versions of projects. Git enables branching, merging and reverting changes, making code management easier. It is widely used in open-source and commercial projects. Popular platforms like GitHub, GitLab, and Bitbucket provide remote repositories for Git-based collaboration. It is a free and open-source software that is available for Windows, macOS, and Linux. GIT is a software and can be installed on your computer.

2. GitHub

GitHub is a web-based hosting platform for Git repositories 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.



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
[Acer@DESKTOP-DBSJ88D]~[~\..\..\dot_net]
>> git config --global user.name "Bud10"
> git config --global user.email "buddhatmg2468@gmail.com"
```

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.



```
>> git init
Initialized empty Git repository in C:/Users/Acer/OneDrive/Desktop/dot_net/.git/
[Acer@DESKTOP-DBSJ88D]~[~\..\..\dot_net]
[y master • ?u]-> git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
        README.md
        add.py
        test.txt

nothing added to commit but untracked files present (use "git add" to track)
[Acer@DESKTOP-DBSJ88D]~[~\..\..\dot_net]
[y master • ?u]-> █
```

Now add the files to place it into staging area and then commit the files such that the files are stored in the local repository.

```
[Acer@DESKTOP-DBSJ88D] ~\..\..\dot_net
[~ master ● +4] -> git add .
[~ master ● +4] -> git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   .gitignore
    new file:   Readme.md
    new file:   add.py
    new file:   test.txt

[Acer@DESKTOP-DBSJ88D] ~\..\..\dot_net
[~ master ● +4] -> git commit -m "Initial files commit"
[master (root-commit) e688b35] Initial files commit
 4 files changed, 14 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 Readme.md
 create mode 100644 add.py
 create mode 100644 test.txt
```

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

```
> 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 directory)
    modified:   test.py

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

After changing the contents in the file “**app.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.

```
[Acer@DESKTOP-DBSJ88D] ~\..\..\dot_net
[~ master] -> git remote add origin https://github.com/Bud10/Dot_net_Lab.git
[Acer@DESKTOP-DBSJ88D] ~\..\..\dot_net
```

Now push the files in the repository created.

```
[Acer@DESKTOP-DBSJ88D] ~\..\..\dot_net
[~ master] -> git push -u origin master
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 12 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 940 bytes | 470.00 KiB/s, done.
Total 9 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Bud10/Dot_net_Lab.git
 * [new branch] master -> master
branch 'master' set up to track 'origin/master'.
[Acer@DESKTOP-DBSJ88D] ~\..\..\dot_net
[~ master] ->
```

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

```
[Acer@DESKTOP-DBSJ88D] ~\..\..\dot_net
[~ dev] -> git branch
* dev
master
```

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

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
[Acert@DESKTOP-DBSJ88D] [-\.\.\dot_net]
[P master]-> git checkout -b dev
Switched to a new branch 'dev'
[Acert@DESKTOP-DBSJ88D] [-\.\.\dot_net]
[P dev]-> git add .
[Acert@DESKTOP-DBSJ88D] [-\.\.\dot_net]
[P dev]-> git commit -m "Modified text file in dev branch"
[dev 81c4129] Modified text file in dev branch
1 file changed, 0 insertions(+), 0 deletions(-)
[Acert@DESKTOP-DBSJ88D] [-\.\.\dot_net]
```

To change the branch, we can use the command “*git switch master*” or “*git checkout master*”. To make sure the branch is visible to other users of the repository push the branch into the GitHub.

```
[P dev]-> git push -u origin dev
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 391 bytes | 391.00 KiB/s, done.
Total 4 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
remote:
remote: Create a pull request for 'dev' on GitHub by visiting:
remote:   https://github.com/Bud10/dot_net_lab/pull/new/dev
remote:
To https://github.com/Bud10/dot_net_lab.git
```

Merging the branches such that the changes in the new branch is added to the main code base. To do this you must be in master branch.

```
[Acert@DESKTOP-DBSJ88D] [-\.\.\dot_net]
[P dev]-> git switch master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
[Acert@DESKTOP-DBSJ88D] [-\.\.\dot_net]
[P master]-> git merge dev
Updating 2c7780c..81c4129
Fast-forward
 lab1/test.txt | Bin 156 -> 154 bytes
1 file changed, 0 insertions(+), 0 deletions(-)
[Acert@DESKTOP-DBSJ88D] [-\.\.\dot_net]
```

To check the commits performed in the past use git log or git log --oneline

```
[Acert@DESKTOP-DBSJ88D] [-\.\.\dot_net]
[P master]-> git log
commit 81c4129d181916b868b4e8732c5669df3da156df (HEAD -> master, origin/dev, dev)
Author: Bud10 <buddhatmg2468@gmail.com>
Date: Sat Mar 22 14:11:26 2025 +0545

    Modified text file in dev branch

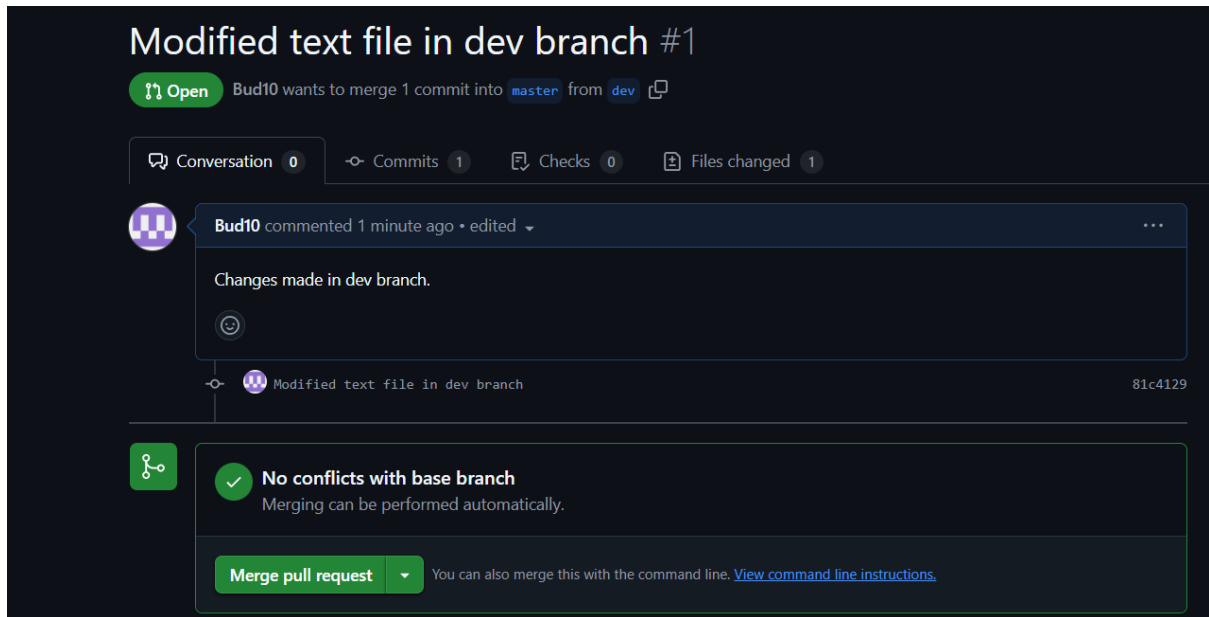
commit 2c7780c5c14b66aa4bec31bb85565a5342eab990 (origin/master)
Author: Bud10 <buddhatmg2468@gmail.com>
Date: Sat Mar 22 14:06:44 2025 +0545

    Created a lab1 file

commit e688b357ac476a1e045f6516be1935fe53b59e9
Author: Bud10 <buddhatmg2468@gmail.com>
Date: Sat Mar 22 13:58:41 2025 +0545

    Initial files commit
```

Merging the branch in the GUI GitHub (Web)



In case of merge conflict, the owner of the repo is responsible for selecting the changes to apply themselves.

Conclusion:

In this lab, we learned about the basics of Git and GitHub. We performed various practical implementation such as initialization, branching, merging, pushing and committing and are hosted in [this repo](#).