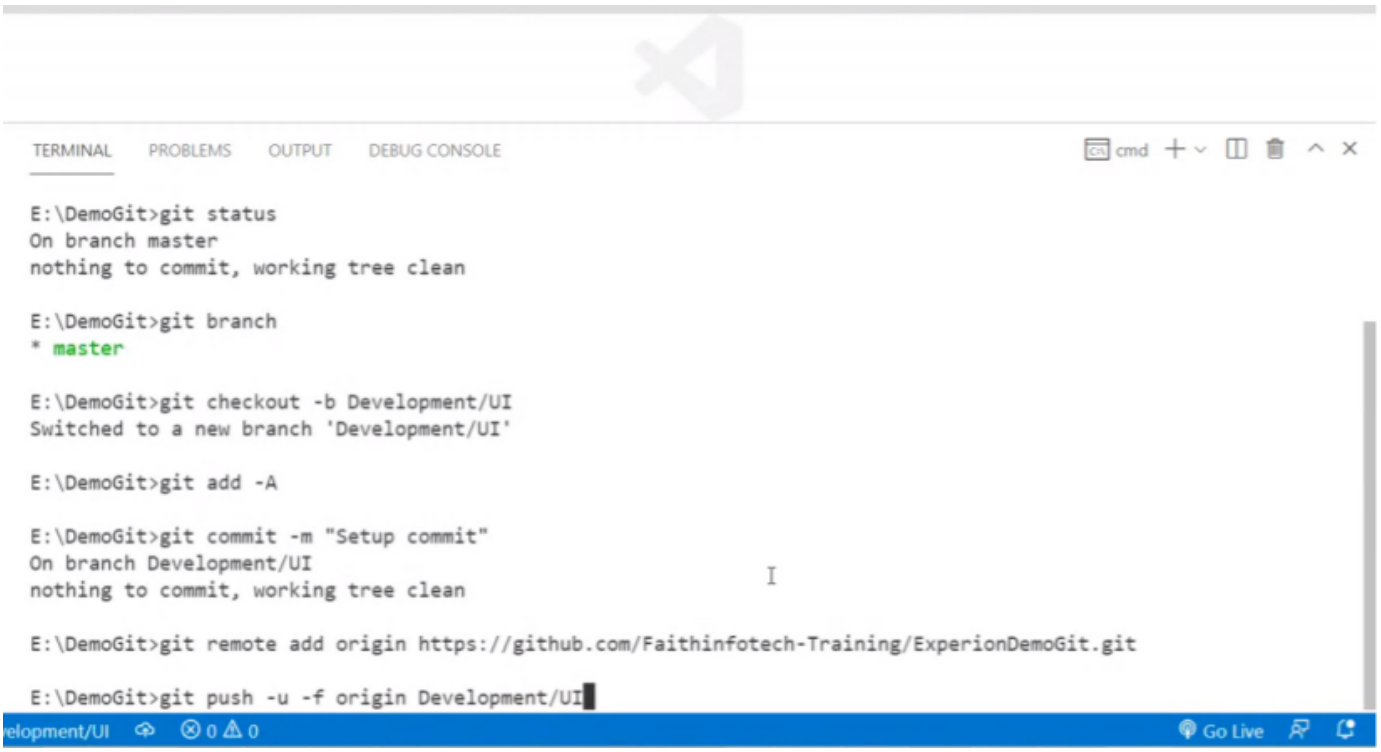
GIT REPORT

Collaboration of Git Repository in the project can be done by following steps:

1. Create a Git Repository
    a. In the upper-right corner of any page, use the drop-down menu, and select New repository.
    b. Give a name for the repository.
    c. Optionally, add a description of the repository.
    d. Choose a repository visibility.
    e. Select Initialize this repository with a README.
    f. Click Create repository.

2. Initializing git Repository
    a. Create a project on the local device.
    b. Run Git init command to make the project as Git project
    c. Run Git branch command to know the current working branch
    d. To make new branch run,
        git checkout -b <branch name>
    e. For making initial commit run,
        git add -A
    f. For committing the changes run,
        git commit -m “message”
    g. To connect local project folder to remote git repository run,
        git add remote <remote name> <git URL>
    h. To push the local changes to remote repository run
        git push -u -f <remote name> <branch name>



3. Clone the Existing Git Repository

   To clone existing remote git repository and use it in local system execute commands:

   a. Git clone <URL>
   b. Then to checkout to the existing branch

      git checkout -t <remote name>/<branch name>
   c. After making changes in the file add the file using

      git add .
   d. Then commit the changes by executing

      git commit -m “message”
   e. Then push to remote repository

      git push

4. Pull and Push into the remote Repository when a developer makes changes
   a. Checkout to the working branch

      git checkout-t <remote name>/<branch name>
   b. Then add the changes using

      git add .
   c. Commit the changes made in the local repository

git commit -m “message”

d. Then pull the remote repository

git pull

```
roshni.at@ROSHNIAT MINGW64 /d/Clinic Management/C#/Experion-.Net-Team2 (C#|MERGING)
$ git add .

roshni.at@ROSHNIAT MINGW64 /d/Clinic Management/C#/Experion-.Net-Team2 (C#|MERGING)
$ git commit -m "id changed"
[C# fbe7ec3] id changed
 Committer: Roshni A T <roshni.at@experionglobal.local>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author


roshni.at@ROSHNIAT MINGW64 /d/Clinic Management/C#/Experion-.Net-Team2 (C#)
$ git pull
Already up to date.

roshni.at@ROSHNIAT MINGW64 /d/Clinic Management/C#/Experion-.Net-Team2 (C#)
$ git push
Enumerating objects: 18, done.
Counting objects: 100% (18/18), done.
Delta compression using up to 4 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (10/10), 910 bytes | 303.00 KiB/s, done.
Total 10 (delta 6), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (6/6), completed with 5 local objects.
To https://github.com/Faithinfotech-Training/Experion-.Net-Team2.git
   b9b0c6c..fbe7ec3  C# -> C#

roshni.at@ROSHNIAT MINGW64 /d/Clinic Management/C#/Experion-.Net-Team2 (C#)
$
```

e. Git pull will result in conflicts and the developers have to resolve the conflicts that occur in the file. There are three options for resolving conflicts
   i. Accept incoming changes
   ii. Accept Current changes
   iii. Accept Both changes

   A developer could choose any of the three options according to the scenario.

f. After resolving conflicts add the file using

   git add .

g. Then commit the changes using

   git commit -m “message”

h. Then push the changes to the remote repository

   git push