# Git Repository:

## Link: <https://github.com/nehabaddam/ICE6.git>

Firstly, we download git. Then in the terminal, we make sure that it is installed.

Text

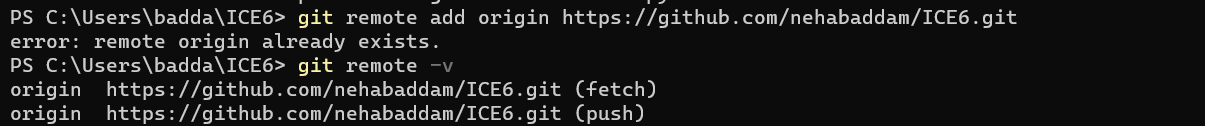
Description automatically generated

Now we configure the user name and email address for the git local repository.

Text

Description automatically generated

On the remote git server, we must create a new repository and link it to the local repository.



# Tasks:

1. **First create a new repository on the Git server.**

We use the cd (change directory) command to change the directory to the directory where the project code is saved. Then we use the “git init” command to create a new repository on the git server.

Text

Description automatically generated

**2) Initialize the local repository and then stage and commit both ipynb files to the local repository.**

We first use the “git status” command to get the status of the main branch. It displays the state of the working directory and the staging area. Here it shows that 2 untracked files need committing. Then we use the command “git add” command to add the files to the staging area.

Text

Description automatically generated

Then we use the “git commit -m” to commit the ipynb files to the local git repository.

Text

Description automatically generated

**3) Now, push all changes of the local repository to the remote repository in the master(main) branch**

Now we use the “git push origin main” command to push the changes to the remote repository in the main branch. Here, the term **origin stands** for the remote repository, and the main is the branch.

Text

Description automatically generated

Once the above command is executed, the files are pushed to the remote repository as shown below.

Graphical user interface, text, application, email

Description automatically generated

**4) Modify one of the files**

I have modified the “UnsupervisedImageClassification.ipynb” file by adding comments to the code and saving it.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

**5) Now, create a new branch in Git and switch to that new branch**

Here we have created a new branch using the command “git branch newbranch”. The “newbranch” is the branch name. The “git branch” command is used to create, list, rename and delete branches.

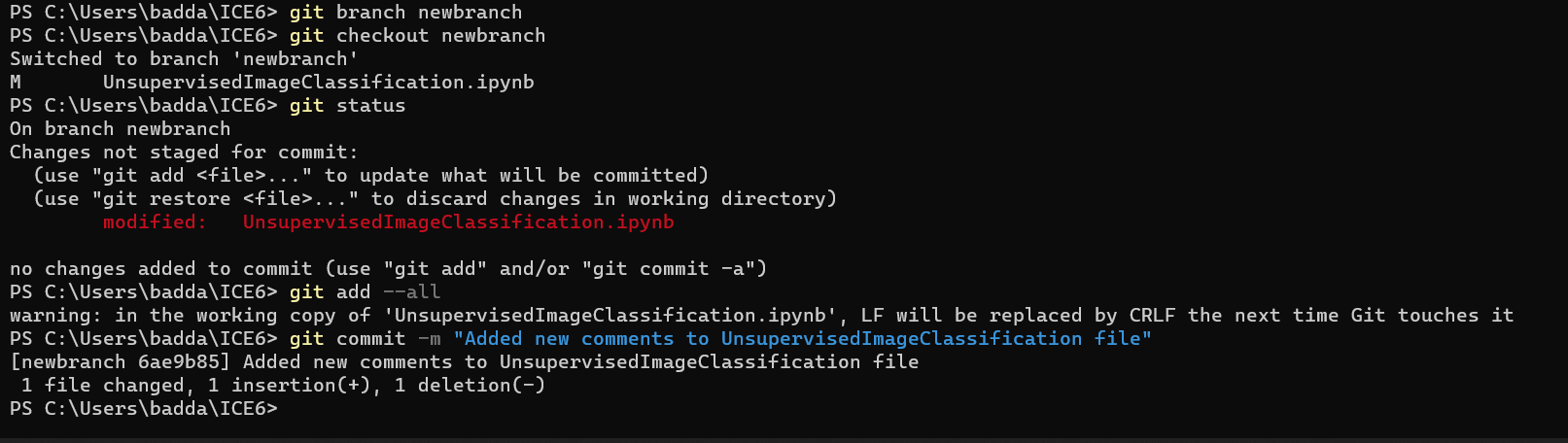
We use the command “git checkout newbranch” to switch to the new branch. The “git branch” command helps to navigate between the branches created by the git branch.

Text

Description automatically generated

**5) Now, stage and commit all changes to the local repository.**

After creating the new branch and switching to that branch, we execute the “git status” command to the status of the working repositories. The “git add -all” command adds all changed files in the working directory to the staging area. Then, we use the “git commit -m” command to push all changes that have been made locally to a remote repository.



**7) Now, push that change to a new branch.**

Now we use the command “git push origin newbranch” to push the changes to the new branch of the remote repository.

Text

Description automatically generated

**8) Switch to the master(main) branch and merge changes in the new branch to the master(main) branch.**

Now, we use the command “git checkout main” to switch from the “newbranch” to the main branch. Then we use the “git merge newbranch” to merge the changes in the “newbranch” to the main branch.

Graphical user interface, text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

**9) push changes to the master(main) branch**

Finally, we push the above committed changes to the main branch using the command “git push origin main”.

Text

Description automatically generated

# Graphical user interface, text, application, email Description automatically generated