CAN-ID : CAN\_33858087  
NAME: Harsh Sulakhe   
College : BLDEA's V P Dr PG Halakatti College of Engineering & Technology

College in Bijapur, Karnataka

Solutions :   
1. Configure your Git with your name and email:

To configure your Git with your name and email as globally, you would use the commands below:

git config --global user.name "Your Name"

git config --global user.email "your.email@example.com"

These commands set up your Git with your name and email so that all of your commits are tied to this information globally throughout your system.

2. Clone the project repository from GitHub

To clone the repository using the URL https://github.com/example/repo.git, use the following command

git clone https://github.com/example/repo.git

It will make a local copy of the remote repository on your machine.

3. Create a new feature branch:

To create and switch to a new branch named new-feature, you can use the following command:

git checkout -b new-feature

The checkout -b option creates the branch and switches to it immediately, preparing it for your new feature development.

4. In your new branch, add new file called feature.py

After creating the file feature.py you can stage it for commit using the following command:

git add feature.py

This command will stage the new file. That is, Git will now track this file to be included in the next commit.

5. Commit your changes with a message:

To commit changes with the message "Add new feature implementation" enter the following command in the terminal:

git commit -m "Add new feature implementation"

The -m flag lets you append a commit message which, for your own documentation of the commit purpose, is really useful.

6. Push changes in your remote repository:

To push changes in your new-feature branch into the remote repository in GitHub, issue the bash command in your git command line or terminal application:

git push origin new-feature

This will push your new-feature branch into the remote repository in GitHub under the new-feature branch.

7. Pull the latest changes in the main branch:

To pull the last changes from the main branch of your remote repository, you could use this command as follows

git pull origin main

That will fetch the latest changes and merge them into your current working branch.

8. Remove the feature.py file from your branch:

To remove a feature.py file from the repository, you could use the following command

git rm feature.py

That will delete the file in your working directory and Git index (or staging area).

9. Commit the removal of the file and push the changes:

To commit the removal of the file, you need to use this command:

git commit -m "Remove feature.py file"

Then you push the changes to the remote repository with this command:

git push origin new-feature

This sequence of commands deletes the file, commits it, and then pushes those changes to the new-feature branch of the remote repository.  
  
  
repository link :