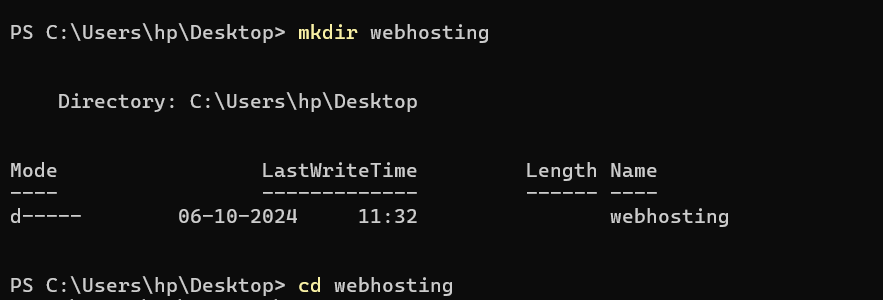
**VCS Task**

**Task Description:**

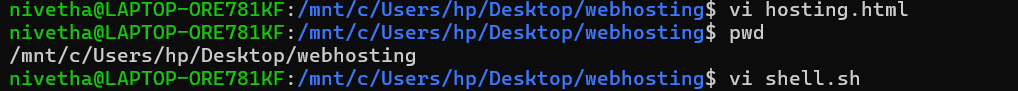
Establish a new directory, populate it with script files, initiate an empty repository on GitHub, convert the local directory into a Git repository, and link it to GitHub for pushing the code into the repository.Perform merge, rebase, stash commands in following github repo.

Explanation:

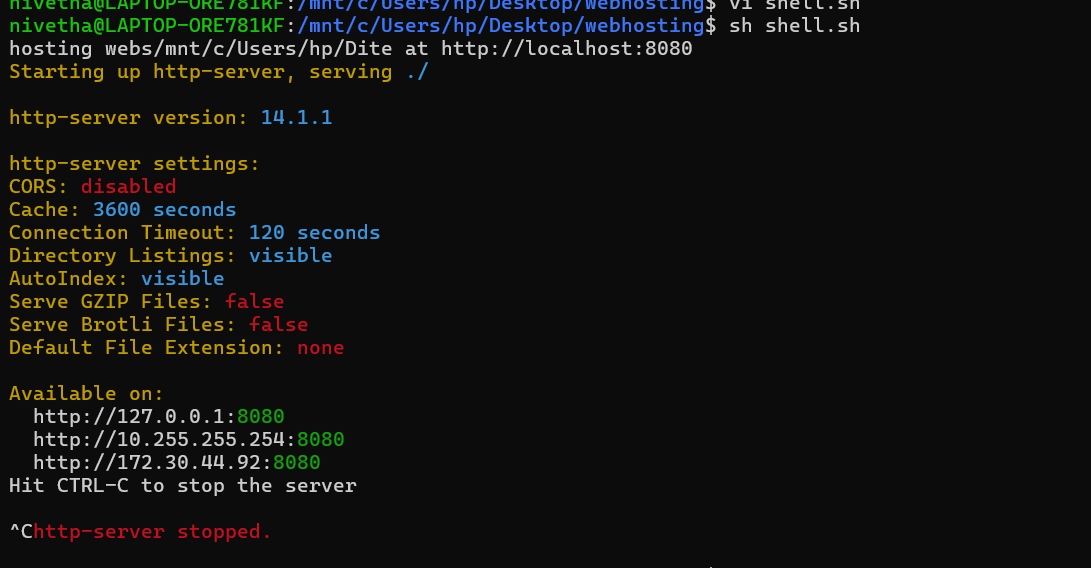
1. Launched the new directory using mkdir and moved to the directory using cd.



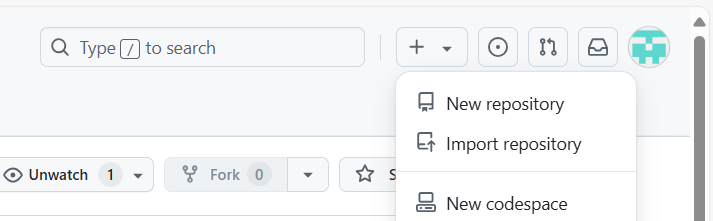
1. Created the new files using vi and populated the scripts it.



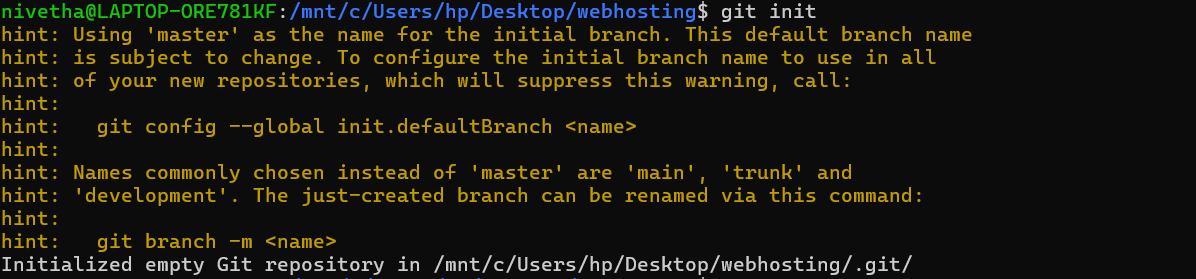
1. Executed the scripts using sh and the below output is displayed.



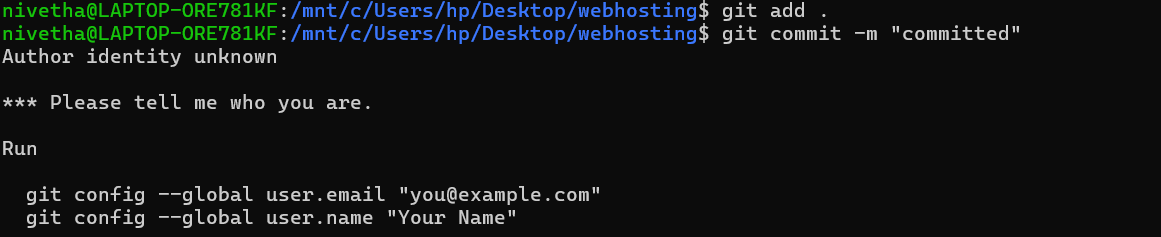
1. Initiated the empty repository on github by login to the github account and by clicking the + icon as mentioned below. Named the repositoty as sample



1. Initialize the git repository using git init.



1. Add the files using git add and commit the changes using git commit -m “<Message>”.

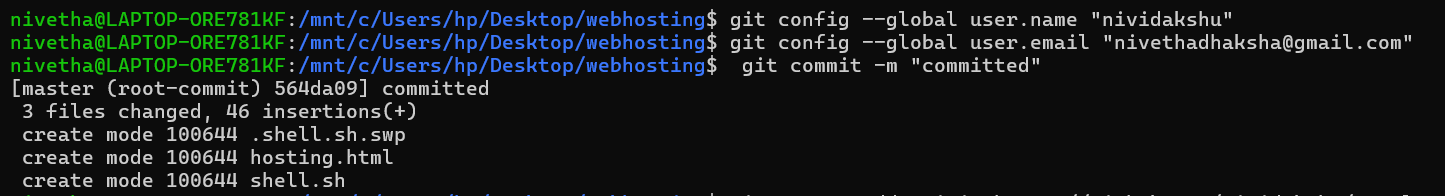


Since, it is not configured use

git config --global user.name "<user name>"

git config --global user.email "<user email>"

git commit -m "committed"



1. Add the github repository as remote using

git remote add origin <https://github.com/nividakshu/sample.git>

copy the url of the repository from github.



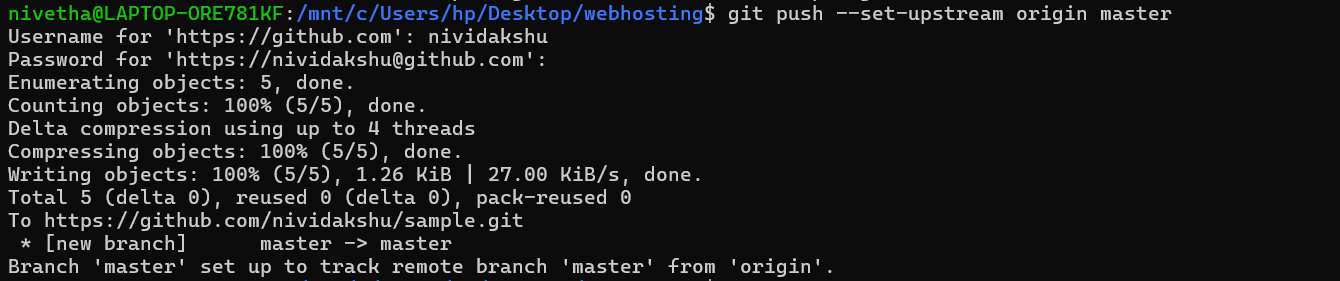
Push the code to github using

git push --set-upstream origin master

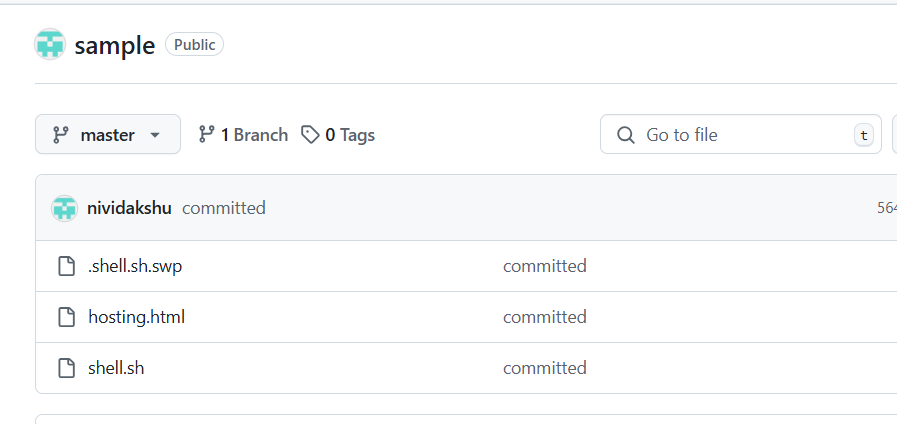
Then, give the user name and password. For password, create token from github and copy paste it.

To create token , navigate to the following path in github account.

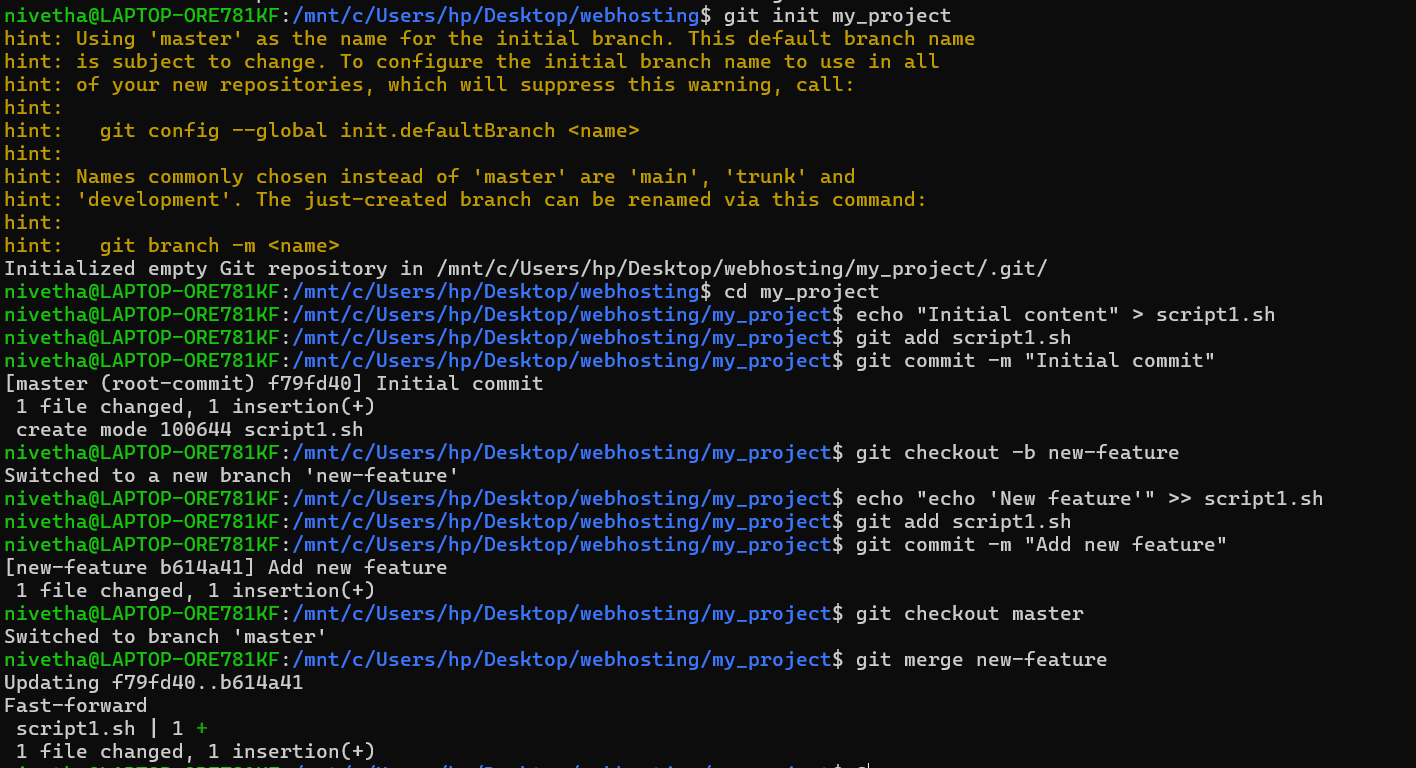
Settings > Developer Settings > Personal access tokens > token (classic)>generate new token classic



1. Now the codes are pushed into git repository.



1. Performing Merge



Explanation :

1. Git repository with initial commit

git init my\_project

cd my\_project

echo "Initial content" > script1.sh

git add script1.sh

git commit -m "Initial commit"

1. Create a New Branch and Make Changes

git checkout -b new-feature

echo "echo 'New feature'" >> script1.sh

git add script1.sh

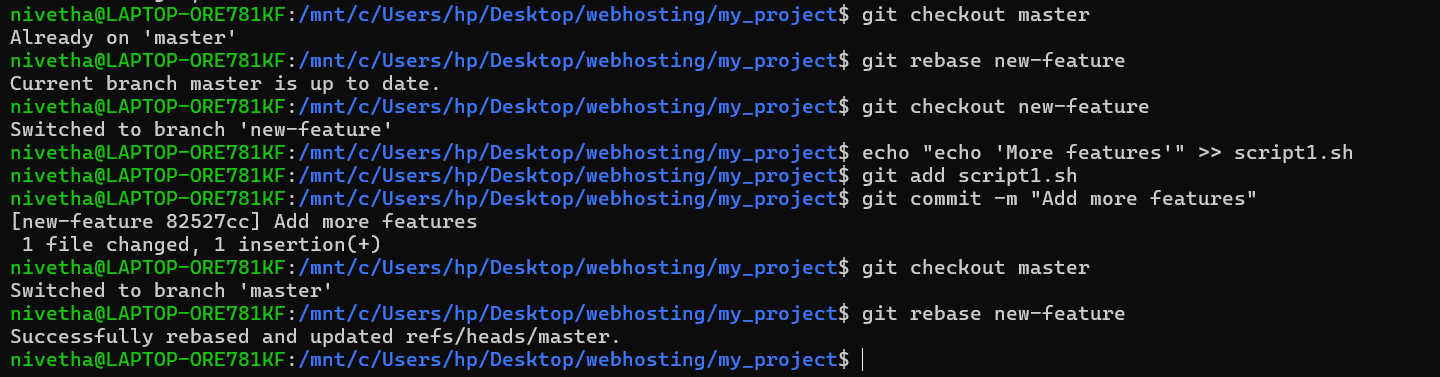
git commit -m "Add new feature"

1. Switch back to master branch and merge the new feature

git checkout master

git merge new-feature

1. Rebase



Explanation:

1. Since already some changes done in the new feature branch, commiting some more feature.

git checkout new-feature

echo "echo 'More features'" >> script1.sh

git add script1.sh

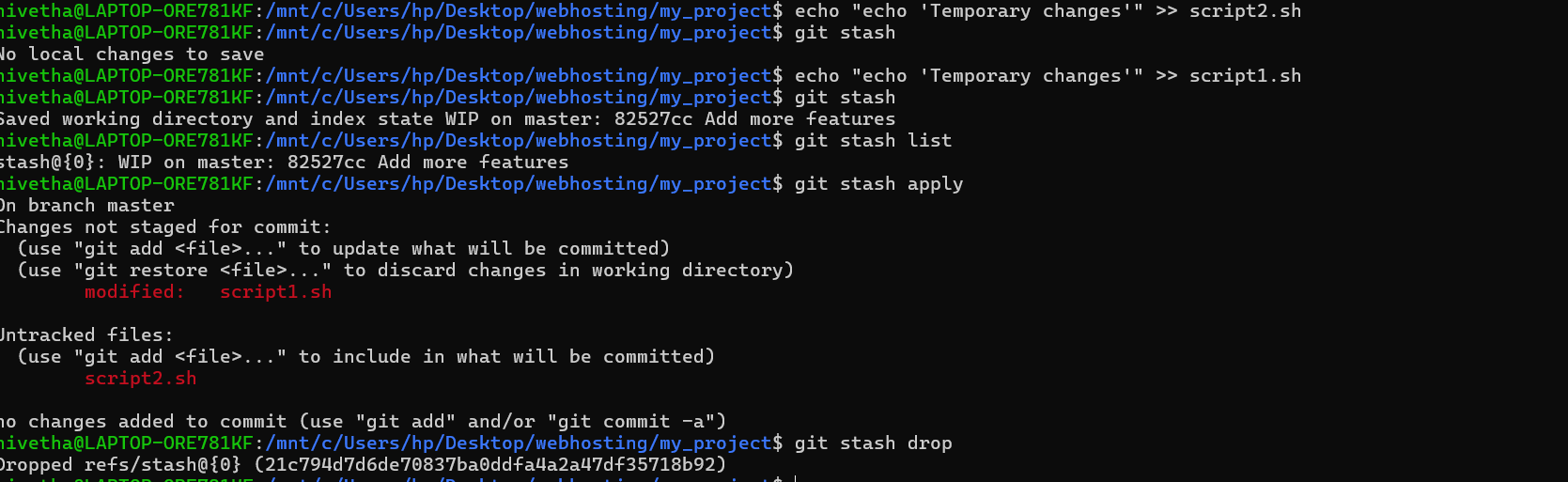
git commit -m "Add more features"

1. Back to main branch and rebase the new feature

git checkout master

git rebase new-feature

1. Performing Stash



Explanation:

1. Edit script1.sh and stash the changes

echo "echo 'Temporary changes'" >> script1.sh

git stash

1. View, apply,drop the stashed changes

git stash list

git stash apply

git stash drop