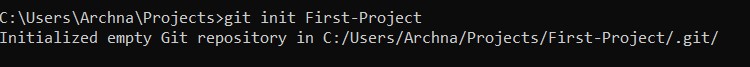
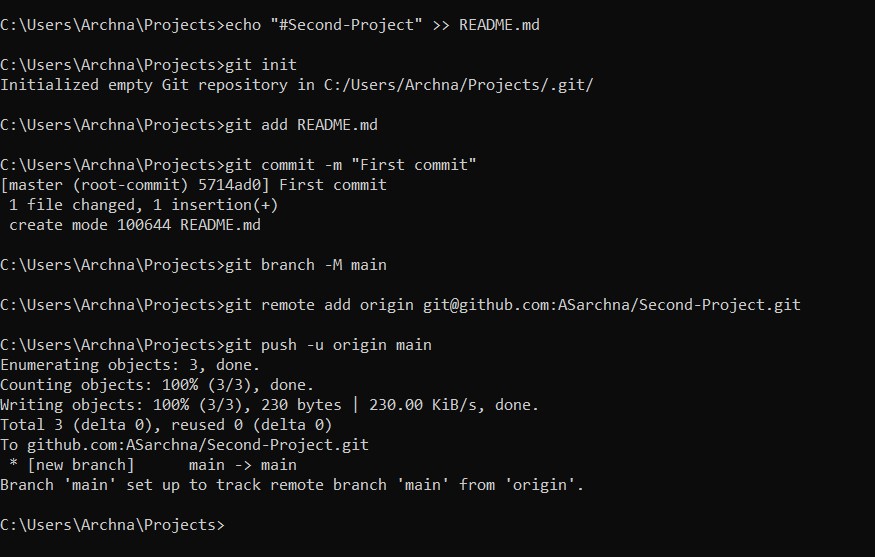
Week-1 Assignment (solved by- Archna Bharti) (bhartiarchna4@gmail.com)

1. Git-hub
2. Basic usage using CLI
3. Setup a local repository

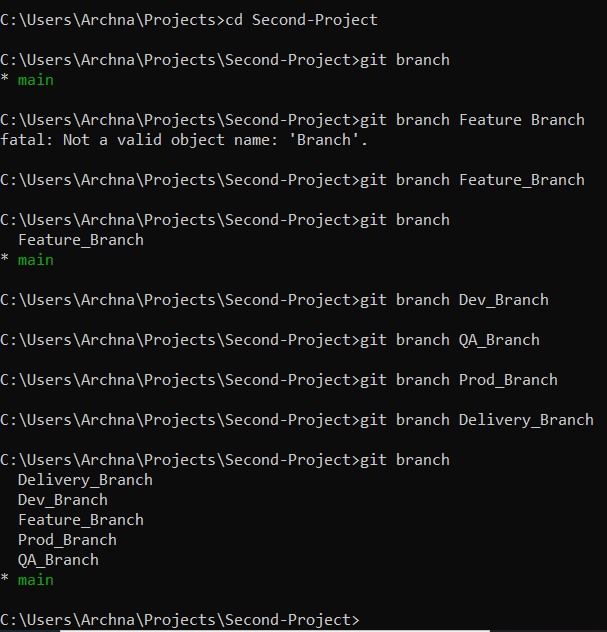
Steps : run command (git init Project\_name)



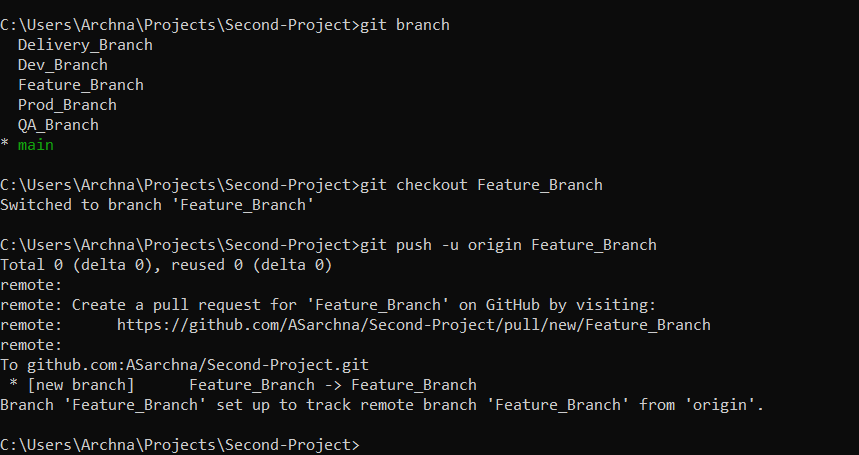
1. Setup a remote repository



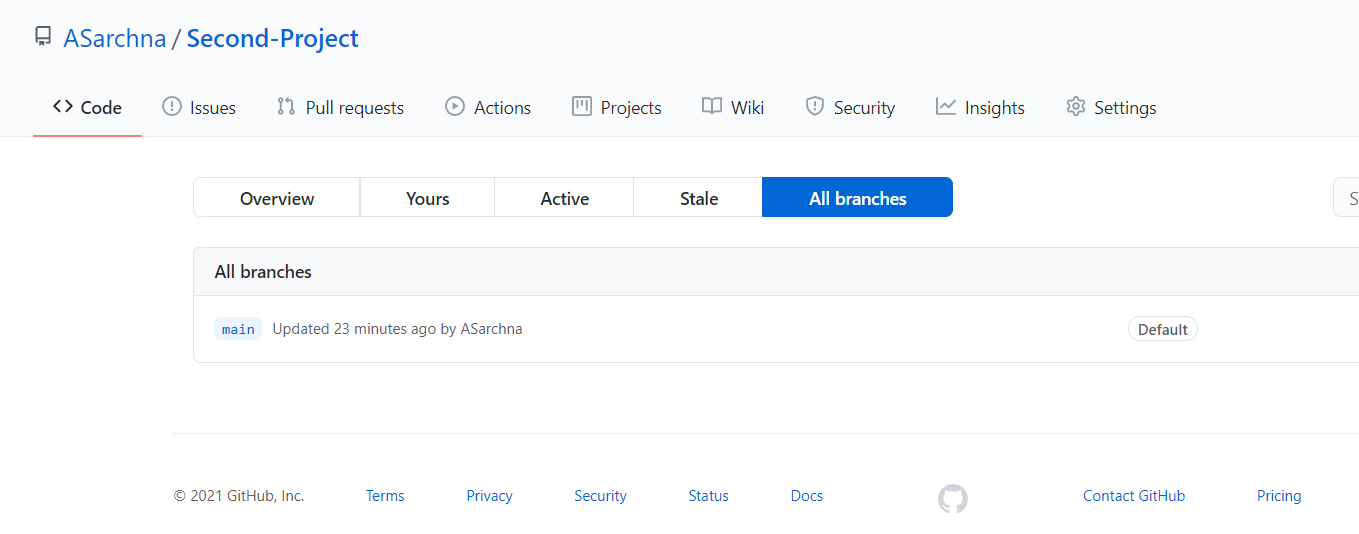
1. Create local branches (Feature Branch, Dev. Branch, QA Branch, Delivery Branch, Prod. Branch)



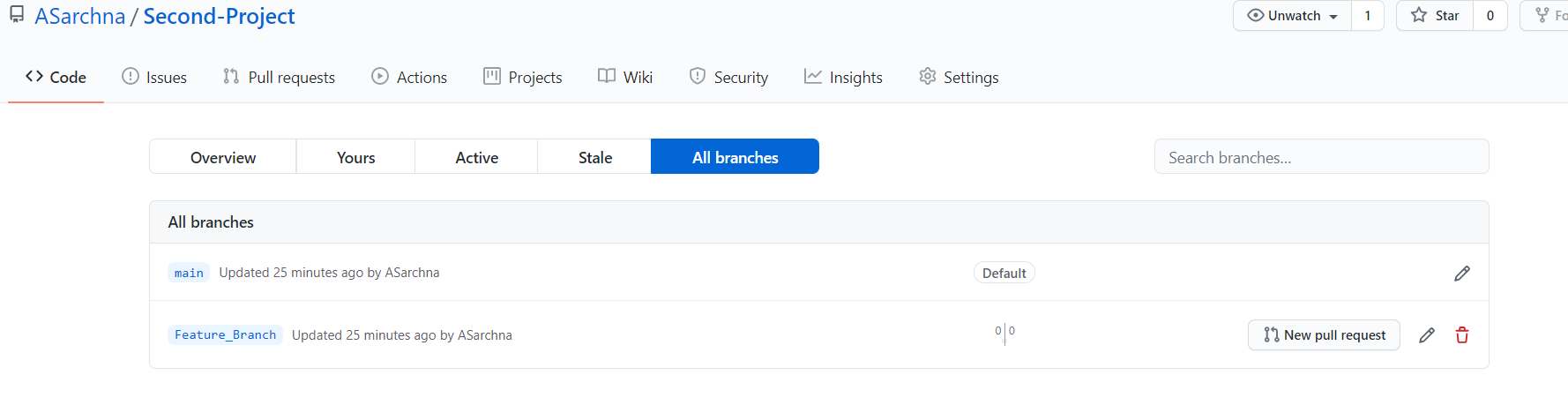
1. Create Remote Branches



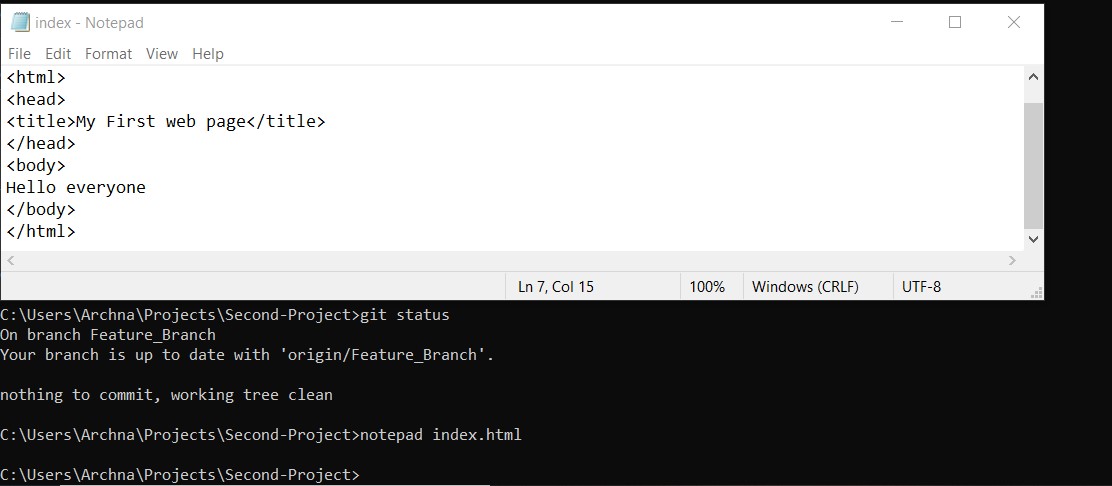
Before Creating Remote Branch



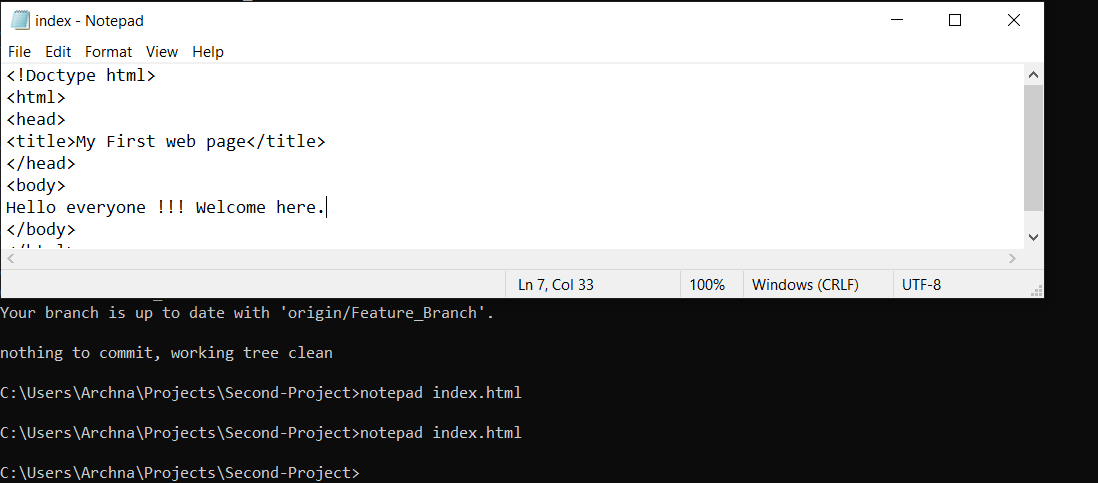
After Creating Remote Branch



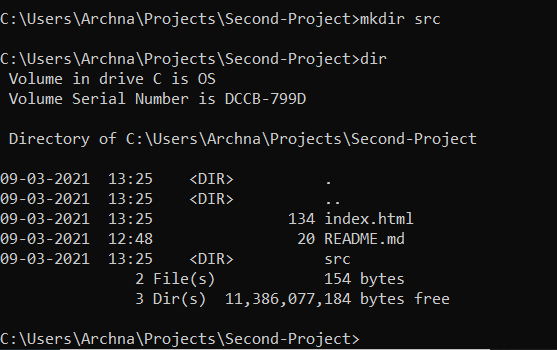
1. Add files, Make changes to exiting files, Add folders, Delete Folders, delete files.



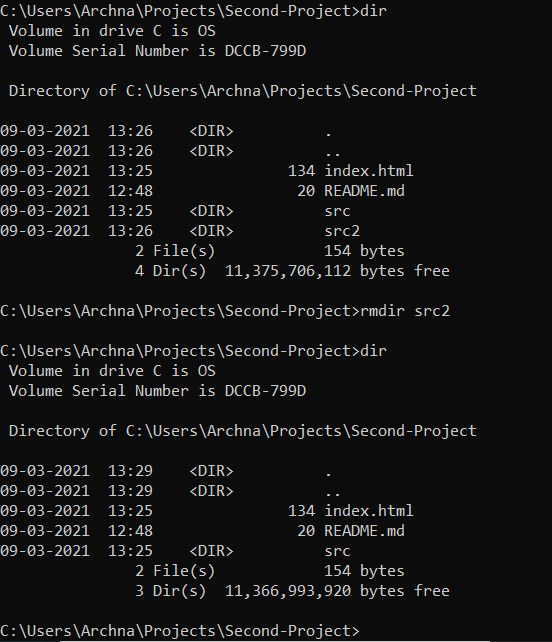
File (index.html) added in Feature Branch



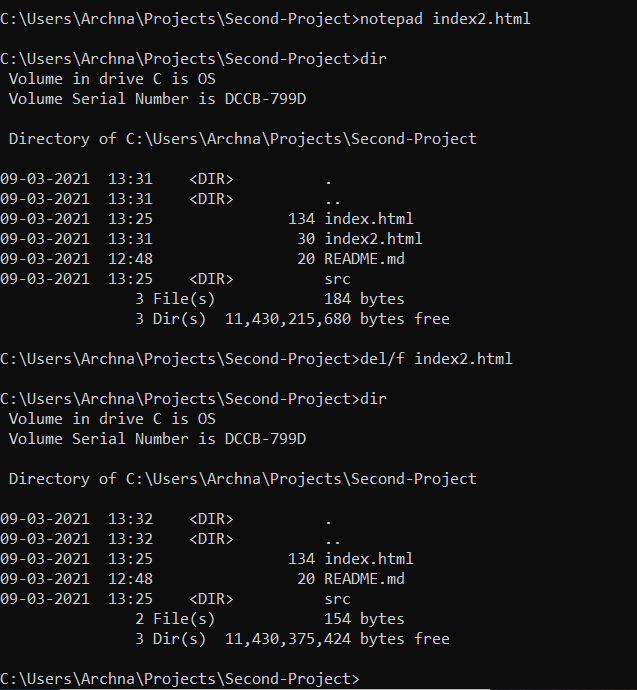
File (Index.html) edited.



**src** Folder/Directory added in Feature Branch

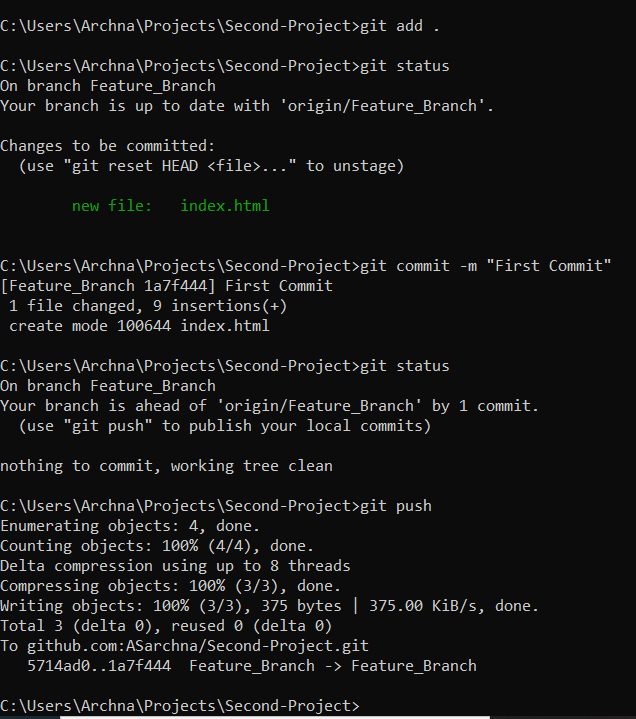


Src2 folder is added and deleted from Feature Branch.

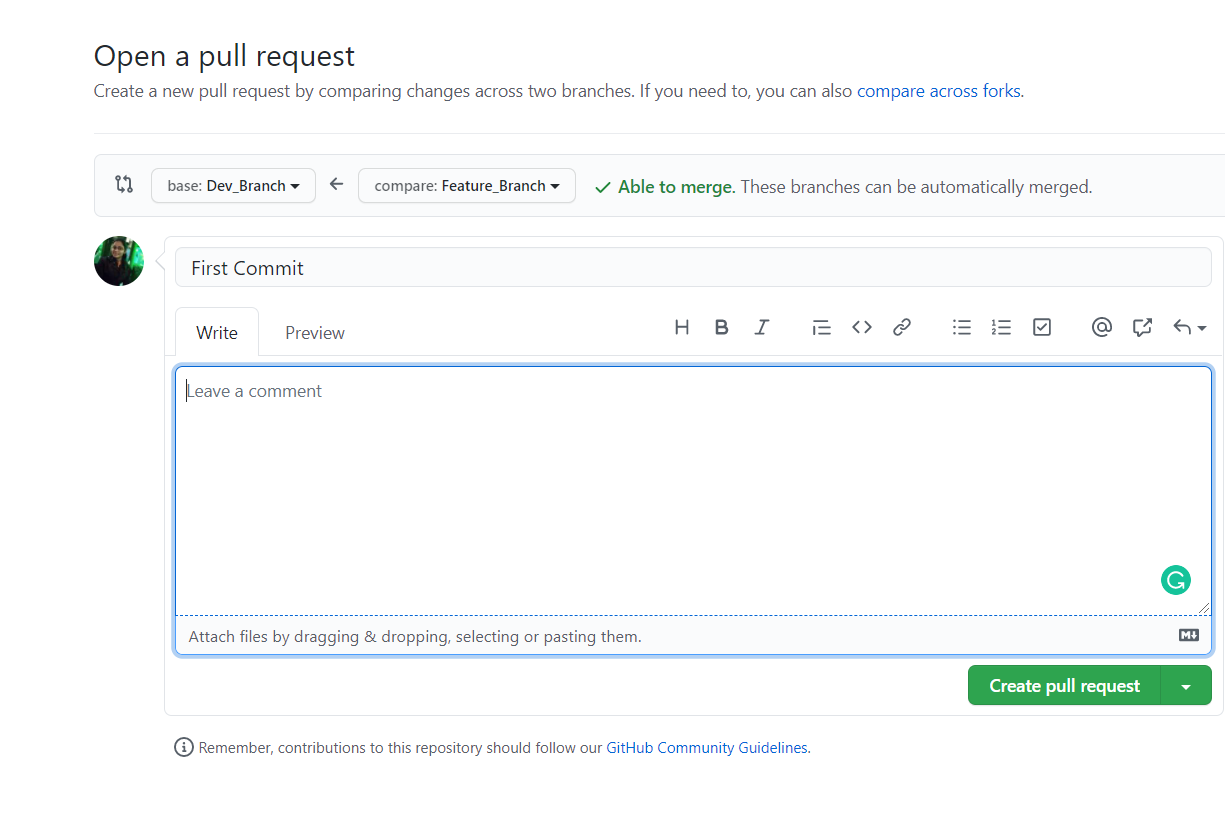


Index2.html file is added and deleted from Feature\_Branch

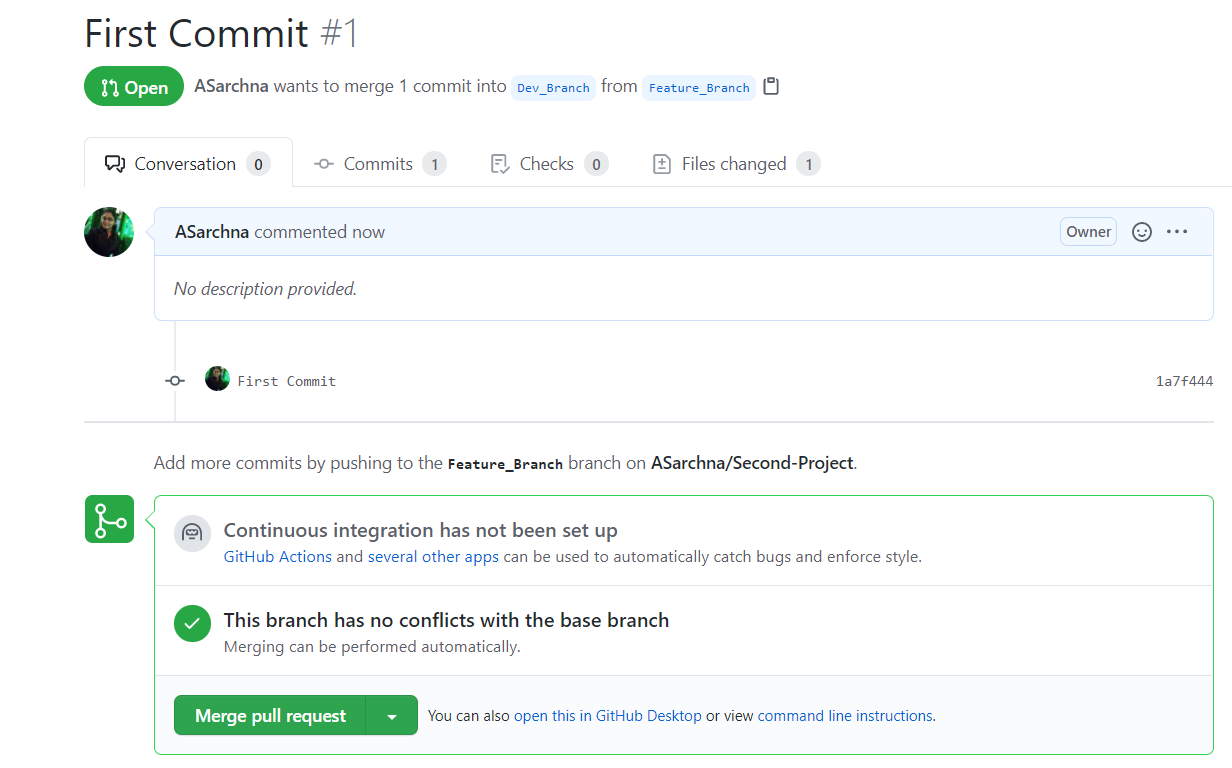
1. Check-in , stage, Commit, Push File into Feature Branch



1. Promote code from Feature Branch to Dev. Branch via Pull Request

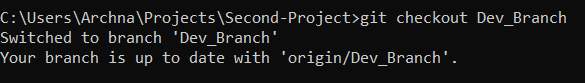


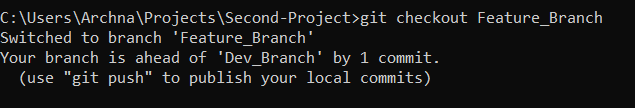
Made a pull request in Feature\_Branch to merge it into Dev\_Branch



No conflict found. Then I merged pull request.

1. Check-out the latest code from remote branch to local branch.





1. Explore the difference between checkout vs pull

Ans : git pull contacts the remote repository identified by origin and looks for updates. It fetches any updates and then merges the changes into the target branch.

Checkout only check the latest code from remote repository to local repository. It does not merge code in local repository.

1. Ans :

Ans: If we want to merge same file from two person. We will have to follow following steps

1. Clone the project into both person system.
2. Both of them work in Branch other then main to reduce merge conflict.
3. Make changes to same file.
4. First person will see status and if code is up-to-date from remote repository then that person will push the changes to remote repository.
5. Then second person will see status. First they have to pull code from remote repository. After that they will add file and commit changes. Then push the code into other branch. If everything looks fine then make pull request and merge changes to main branch.
6. Ensure that all branches is in sync.

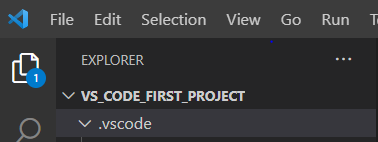
Ans : Make pull request to all branches and make all of them updated. we can also do it by pushing our code to origin branch.

1. Basic Usage using the GUI Client (VS Code or Git-hub Client)

<Repeat all above steps using a GUI>

Ans:

1. Make local repository
2. Steps : make local folder in system like I made folder on desktop VS\_CODE\_FIRST\_PROJECT
3. Open VS code
4. Go to file section and open new folder.
5. Open folder from your system.

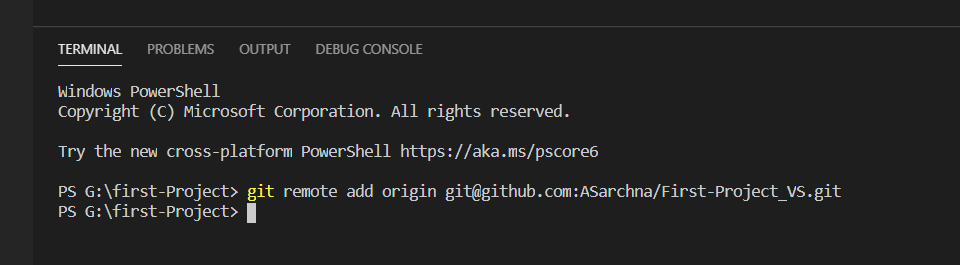


And then initialize the repository in VS code

1. Setup a remote repository

Steps:

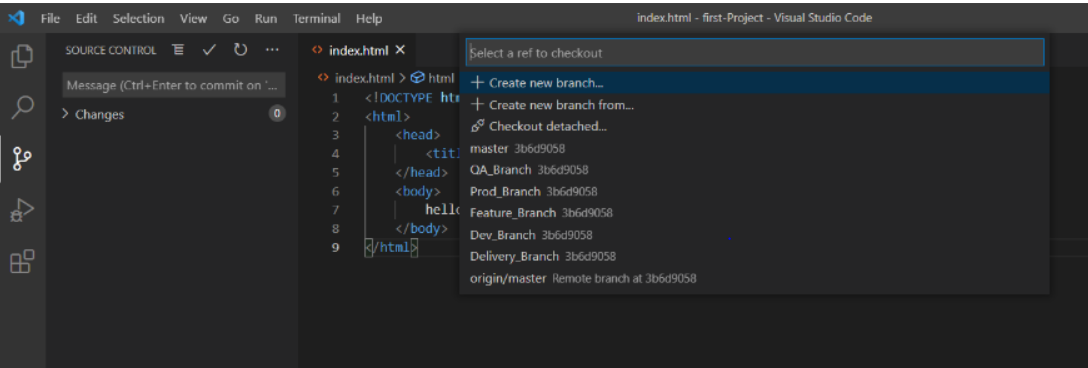
Make new repository in Github without adding README.md file.

And then run following command in terminal of VS code

1. Create local branches (Feature Branch, Dev. Branch, QA Branch, Delivery Branch, Prod. Branch)

Steps:

Go to Source Contro🡪Branch🡪create Branch🡪branch name



Here I created local branch

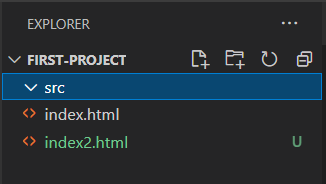
1. And one remote (master) branch , because I made remote repository from here, we can add remote branch same as local with one more step to connect to github .
2. Add files, Make changes to exiting files, Add folders, Delete Folders, delete files.

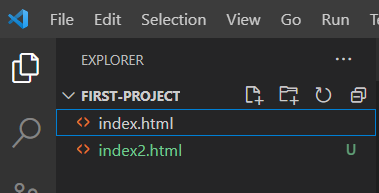
Click on add file and give name to file and add them.

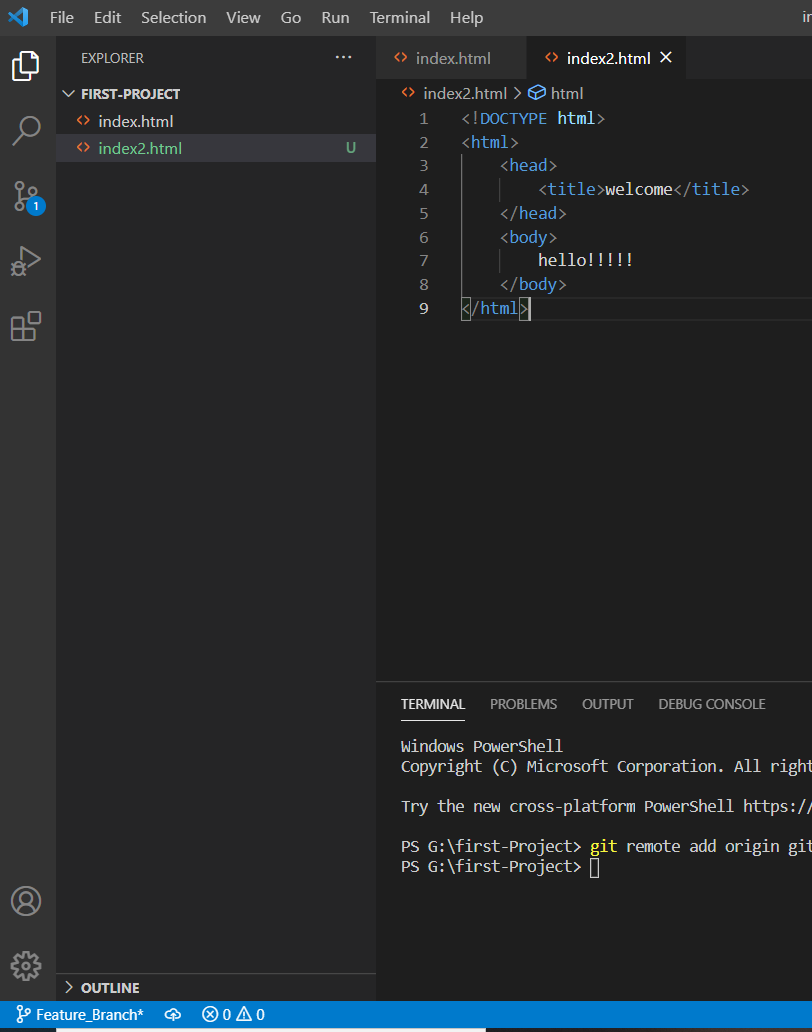
Edit the file and save the changes.

Click on new folder and give name .

If want to delete the folder, right click on folder and click on delete.







Here we can see that I am working in Feature\_branch

1. Check-in , stage, Commit, Push File into Feature Branch

In VS code go to source control and click on push .Write comment and make commit. After commit push into Feature\_Branch

1. Promote code from Feature Branch to Dev. Branch via Pull Request

Ans: this can be done same as in part-a.

If we want to promote code in local reposiroty then merge the code from Feature Branch to Dev\_Branch by clicking merge

1. Check-out the latest code from remote branch to local branch.

Go to checkout to remote branch

And see code in that branch. We can compare code from remote to local branch.

9,10 and 11 is same as part-a

1. On the ground day to day scenarios

Ans:

1. Reset/revert one or more file to previous state and ignore the local changes.

Steps in VS code:

Make changes in files. Commit it . all changes will be saved first. Then go to commit🡪undo last commit. It will remove all local changes.

1. Stash and local changes during merge conflict.

Go to source control🡪stash🡪then many option will come . we can see stash. We can see conflict in code and mere then. Then we can make changes in local repository.

1. Rebasing with all options (reword, edit, squash, fixup, exec, drop).

Rebasing is changing the base of your branch from one commit to another making it appear as if you’d created your branch from a different commit. Internally , Git accomplishes this by creating new commits and applying them to the specified base.

1. Reword : it is used to make changes in comment which we have weitten during commit.
2. Edit : it is used to edit the comit.
3. Squash: it is used to make single buildable commit.
4. Fixup: it is associates a new commit with an existing commit so that when we do an interactive rebase, we don’t have to change any commit messages.
5. Exec : it execute a command after each commit in our interactive rebase.If the command fails, it will stop the rebash, allow to fix whatever needs to be fixed.
6. Drop: it is used to remove the commit from the interactive rebase file.
7. Git log, status & reflog

Ans:

Git log is used to review and read history of everything happens to repository.

Git status is used to see status of all files in repository. It will show changes made to files in working repository.

Git reflog is used recode every action we perform inside of Git where data is stored.