KARANAM YESWANTH TEJA

EMP-ID-289224

BEGINNER PROJECT 2

TO DO LIST WEB APPLICATION

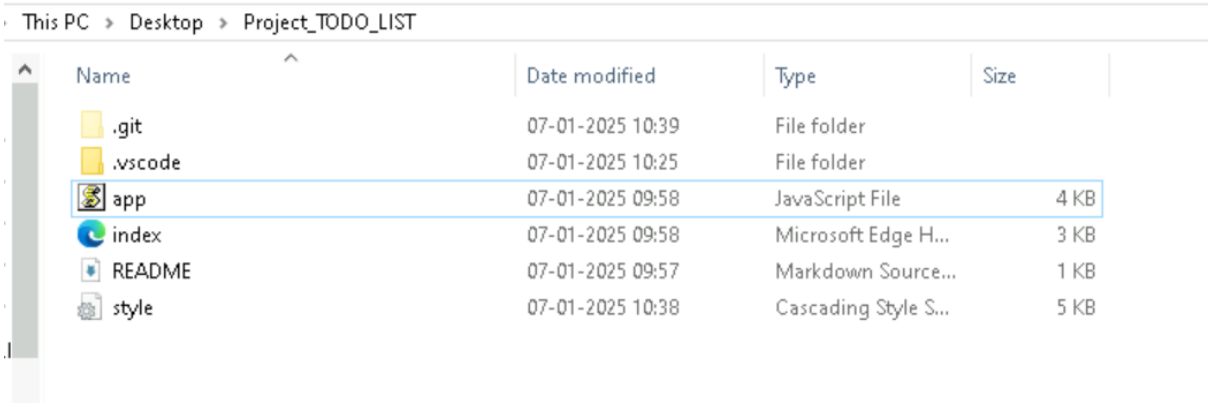
Step-1: Create a repository in GitHub naming Project_TODO_LIST and include readme file.

Step-2: Initialize the git using git Init command in git bash.

```
Administrator@334a7a24d0695Oc MINGW64 ~/Desktop/Project_TODO_LIST (master)
$ git init
Initialized empty Git repository in C:/Users/Administrator/Desktop/Project_TODO_
LIST/.git/
```

Step-3: Clone a project from GitHub in a terminal from visual studio code using git clone "link".

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS                    powershell

PS C:\Users\Administrator\Desktop\Project_TODO_LIST> git clone https://github.com/Jomon-J/Todo-App.git
Cloning into 'Todo-App'...
remote: Enumerating objects: 20, done.
remote: Counting objects: 100% (20/20), done.
remote: Compressing objects: 100% (20/20), done.
remote: Total 20 (delta 6), reused 5 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (20/20), 60.54 KiB | 12.11 MiB/s, done.
Resolving deltas: 100% (6/6), done.
PS C:\Users\Administrator\Desktop\Project_TODO_LIST>
```



Step-4: Connect the remote repo on GitHub to the git by using the command git remote add origin “link”.

```
Administrator@334a7a24d06950c MINGW64 ~/Desktop/Project_TODO_LIST (master)
$ git remote add origin "https://github.com/Yeswanthteja1010/TO-DO-LIST.git"
```

Step-5: Use git pull origin main command to pull the files (like readme) from the remote repo to local repo.


```
Administrator@334a7a24d06950c MINGW64 ~/Desktop/Project_TODO_LIST (master)
$ git remote add origin "https://github.com/Yeswanthteja1010/TO-DO-LIST.git"
```


Step-6: Add the files that are present in the folder(index.html,style.css and app.js) into the staging area by using git add .

Step-7: To know the status of the files that are added to staging area type the git status command.

```
Administrator@334a7a24d06950c MINGW64 ~/Desktop/Project_TODO_LIST (master)
$ git add .

Administrator@334a7a24d06950c MINGW64 ~/Desktop/Project_TODO_LIST (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   app.js
        new file:   index.html
        new file:   style.css
```

Step-8: Commit the changes by using git commit -m “Some description here”.

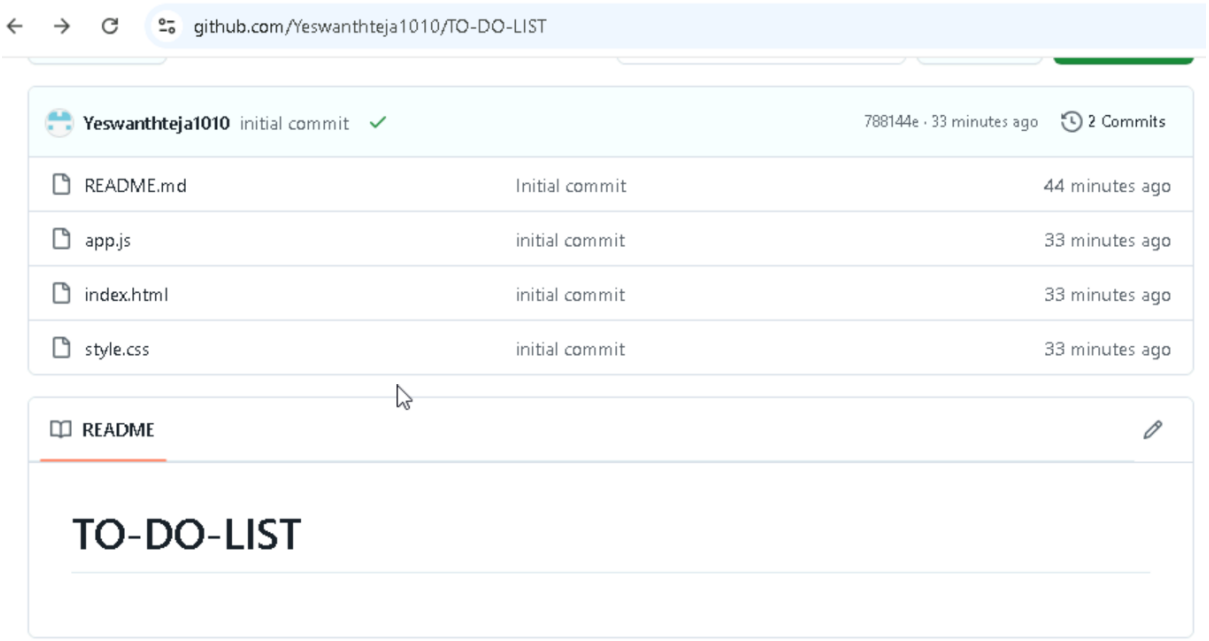
```
dministrator@334a7a24d06950c MINGW64 ~/Desktop/Project_TODO_LIST (master)
 git commit -m "initial commit"
master 788144e] initial commit
3 files changed, 411 insertions(+)
create mode 100644 app.js
create mode 100644 index.html
create mode 100644 style.css

dministrator@334a7a24d06950c MINGW64 ~/Desktop/Project_TODO_LIST (master)
 git branch -a
 master
 remotes/origin/main
```

Step-9: Change the name of master branch to main by using the command “git branch -m master main”.

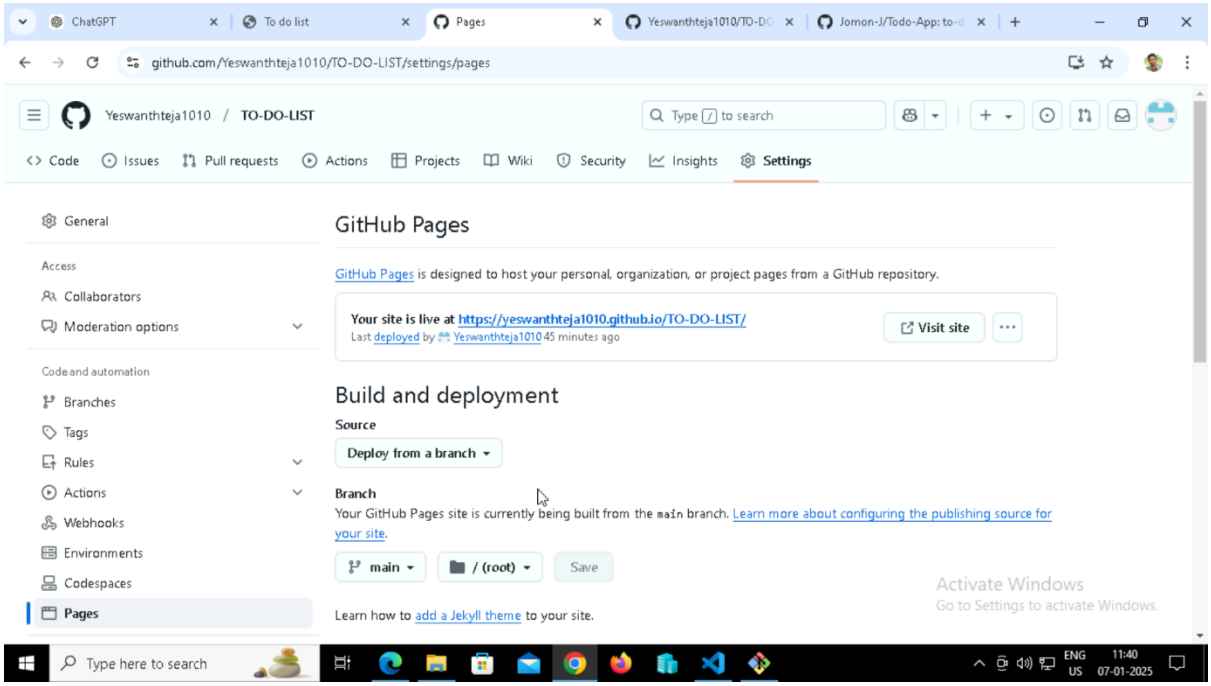
```
Administrator@334a7a24d06950c MINGW64 ~/Desktop/Project_TODO_LIST (master)
$ git branch -m master main

Administrator@334a7a24d06950c MINGW64 ~/Desktop/Project_TODO_LIST (main)
$ git branch -a
* main
  remotes/origin/main
```

Step-10: Now push the files that are committed (saved) previously by using git push origin main. Check whether the files have came to GitHub if any error occurs follow the previous steps.

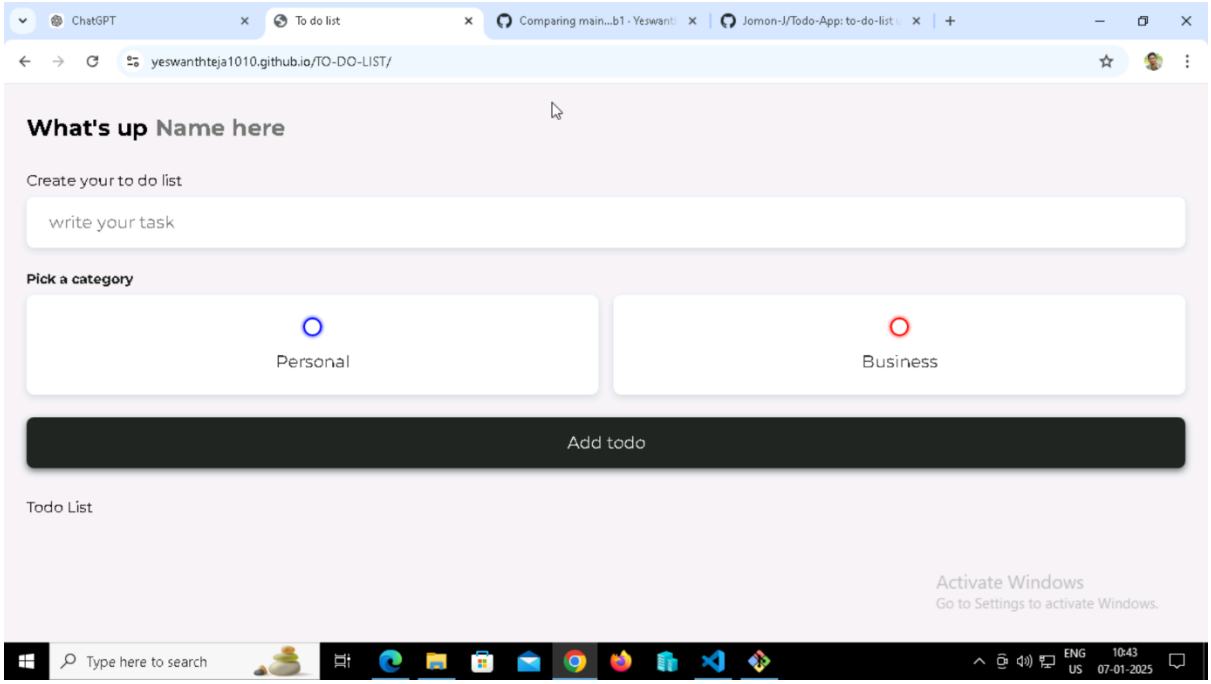
```
Administrator@334a7a24d06950c MINGW64 ~/Desktop/Project_TODO_LIST (main)
$ git push origin main
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 4 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 2.96 KiB | 2.96 MiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Yeswanthteja1010/TO-DO-LIST.git
   3e8323d..788144e  main -> main
```



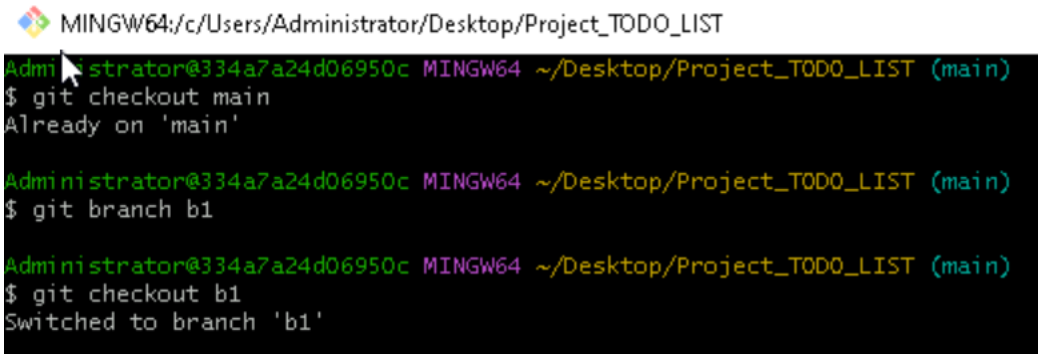
Step-11: Go to settings -> pages and select the main branch and click save to deploy and after the completion of deployment click on the link provided by GitHub.



STEP 12: Click on the link to get the webpage



STEP 13: Check for current branch name and create another branch for features and activate that branch using git checkout command



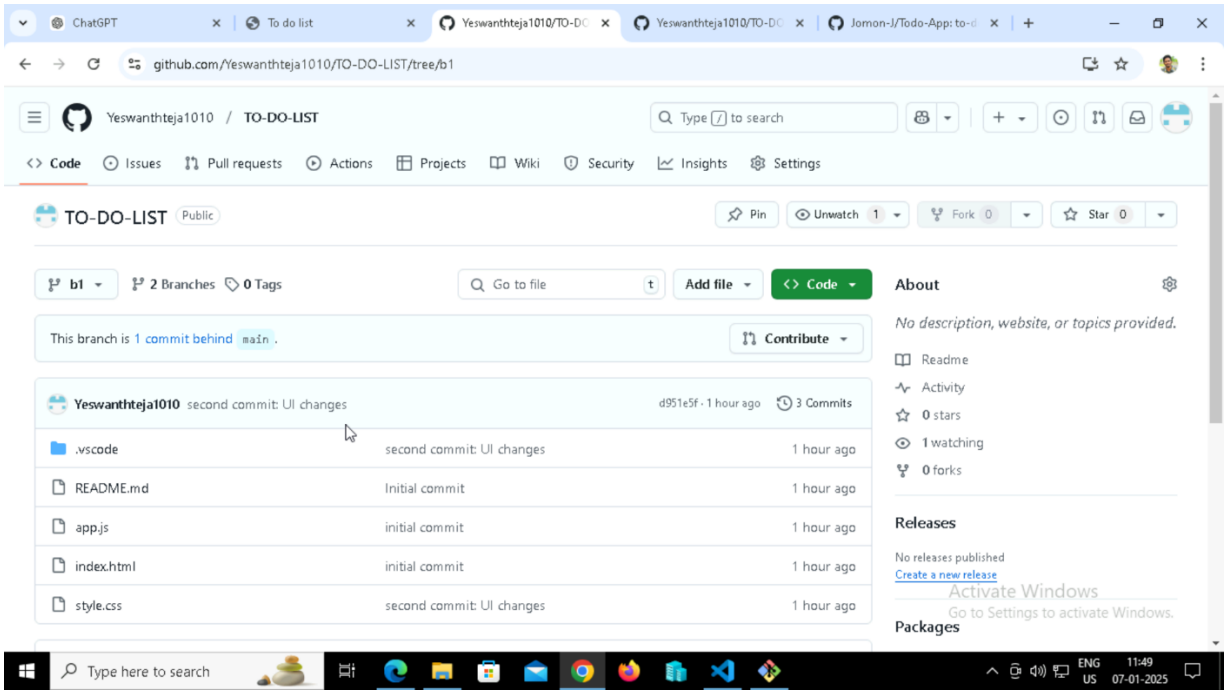
STEP 14 : Modify your project is VScode to add new features

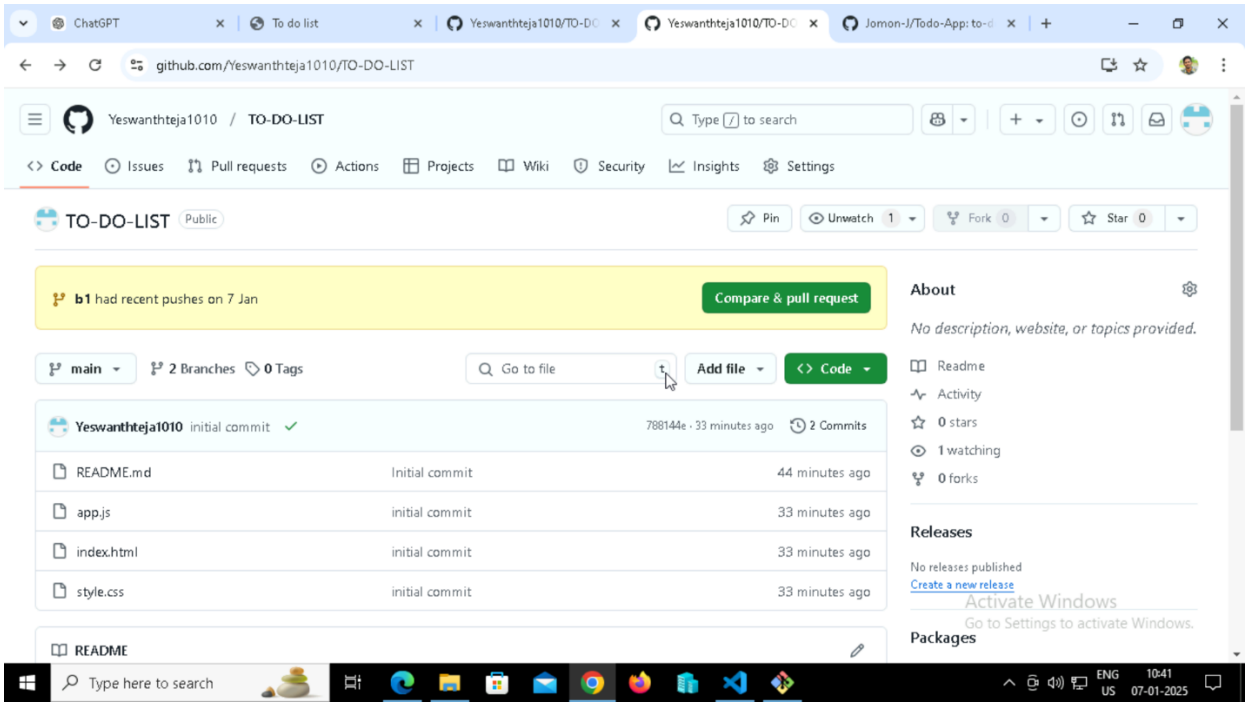
Step-15: Add the files that are present in the folder(index.html,style.css and app.js) into the staging area by using git add .

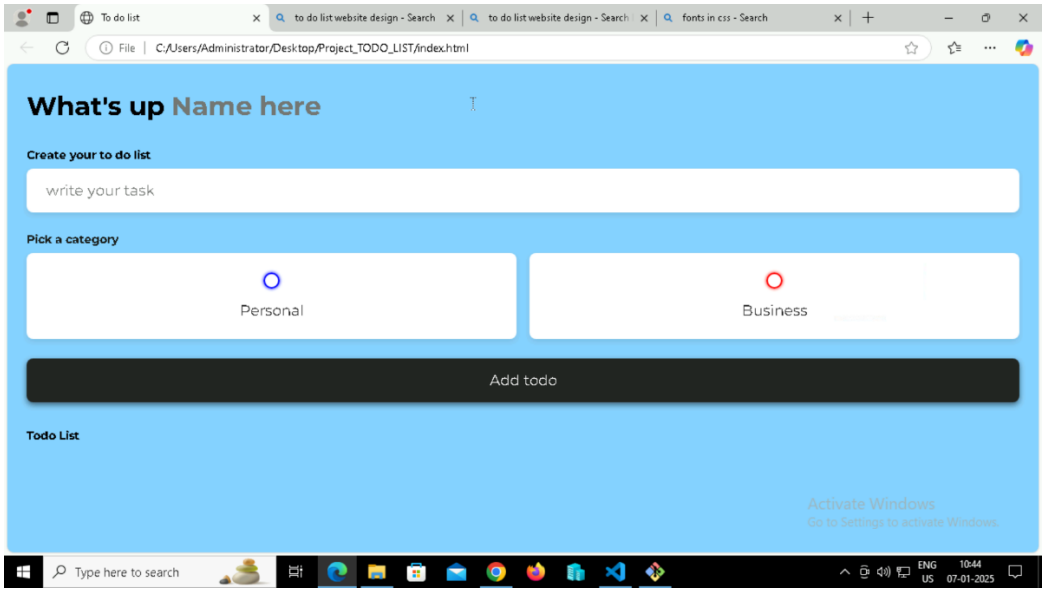
Step-16: To know the status of the files that are added to staging area type the git status command.

Step 17 : Commit the new changes and push those to the new branch

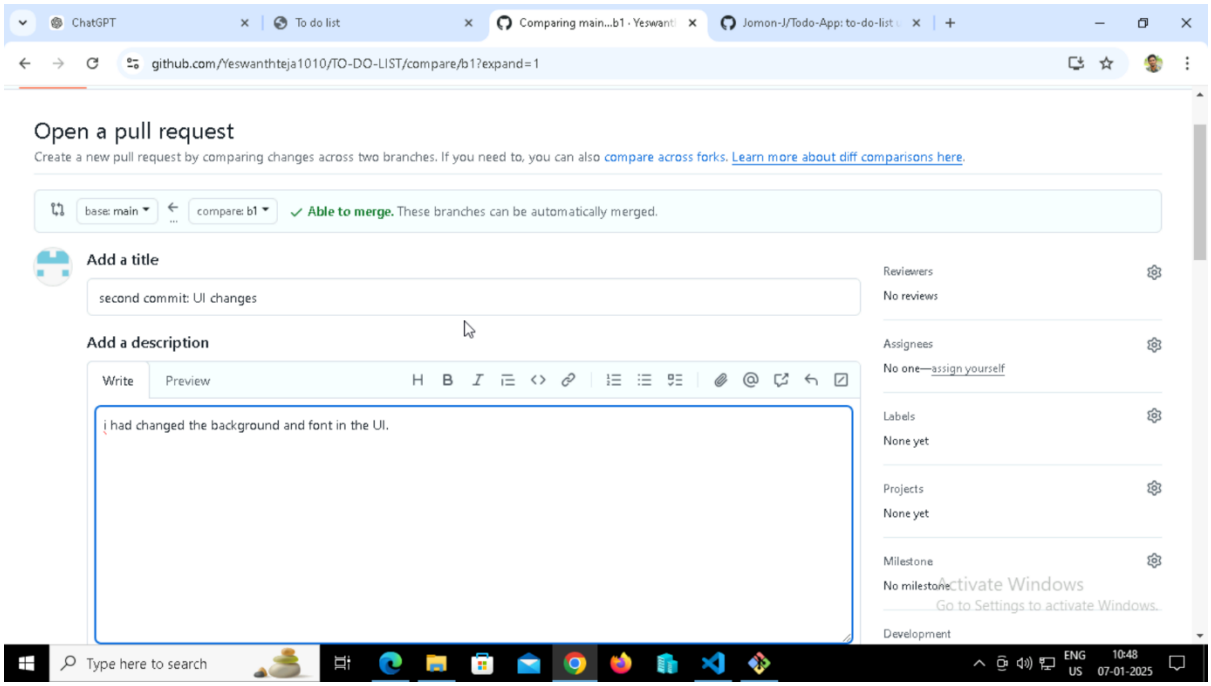
Step 18 : Verify the changes in your github



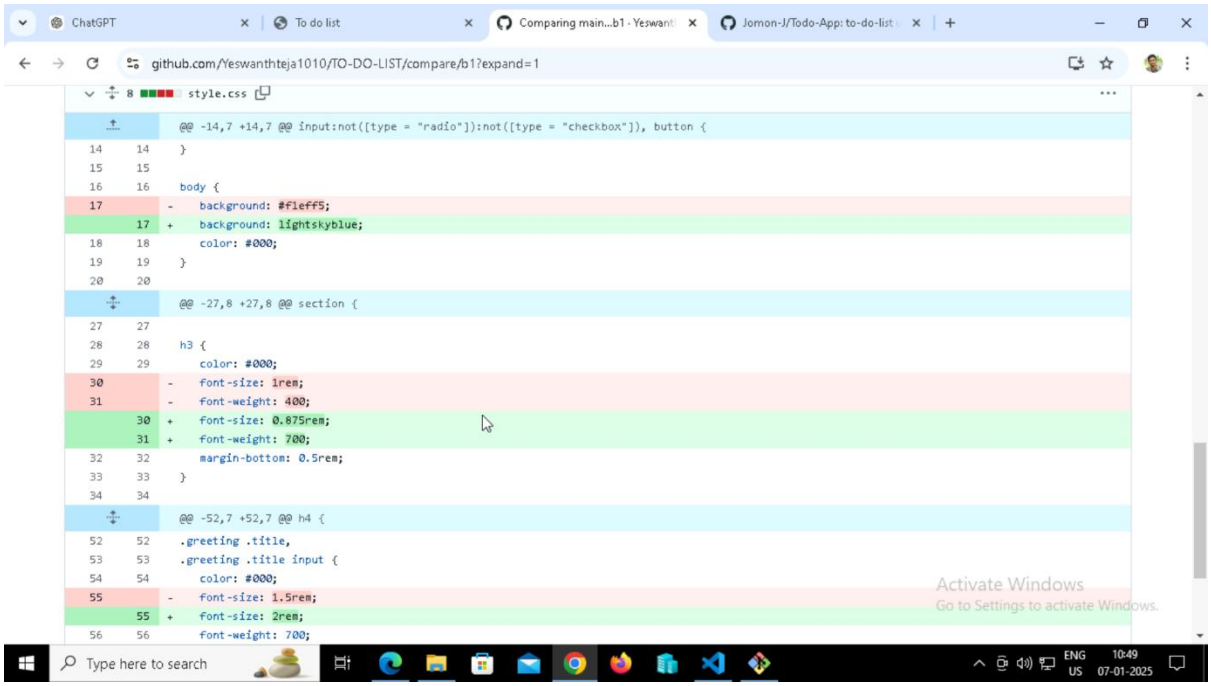




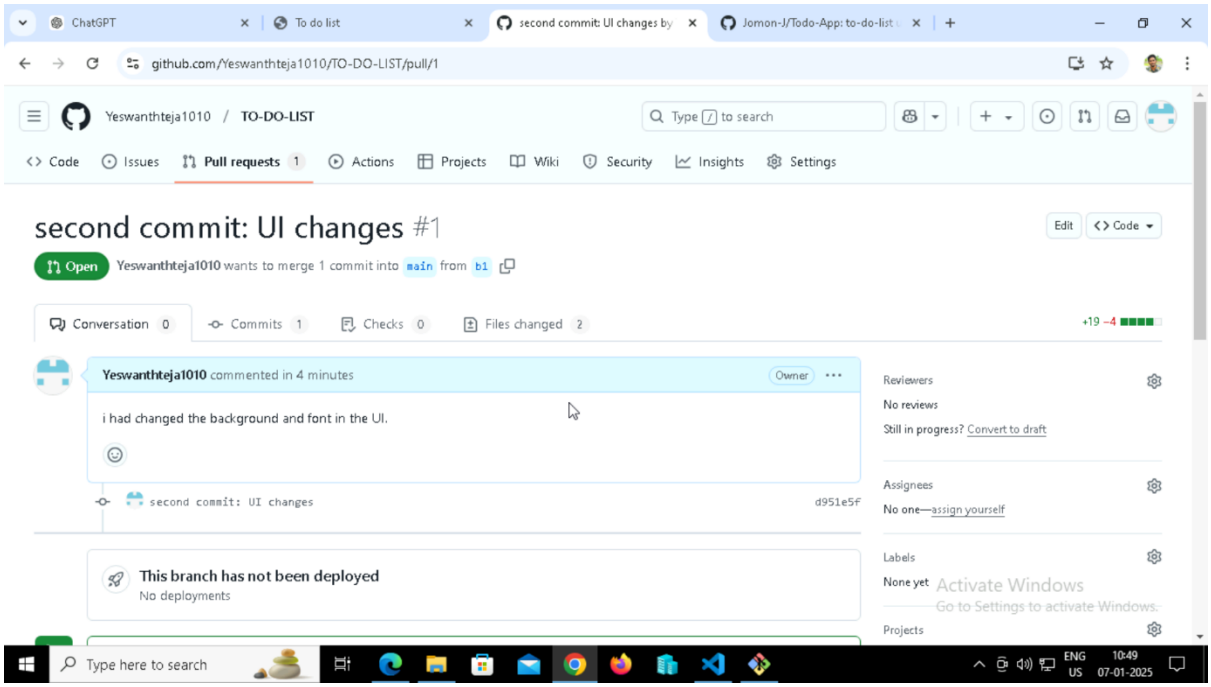
Step 21 : Click on compare and pull button to compare changes make pull request and merge changes.

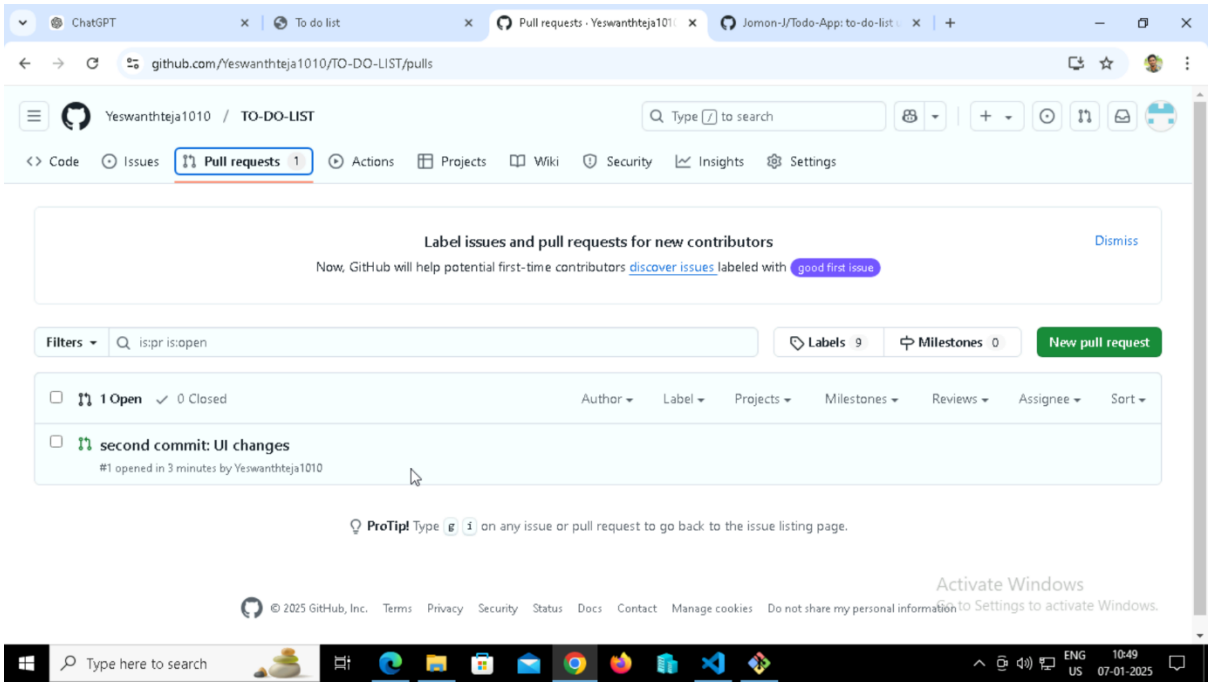


Step 22 : Add title and description for your changes

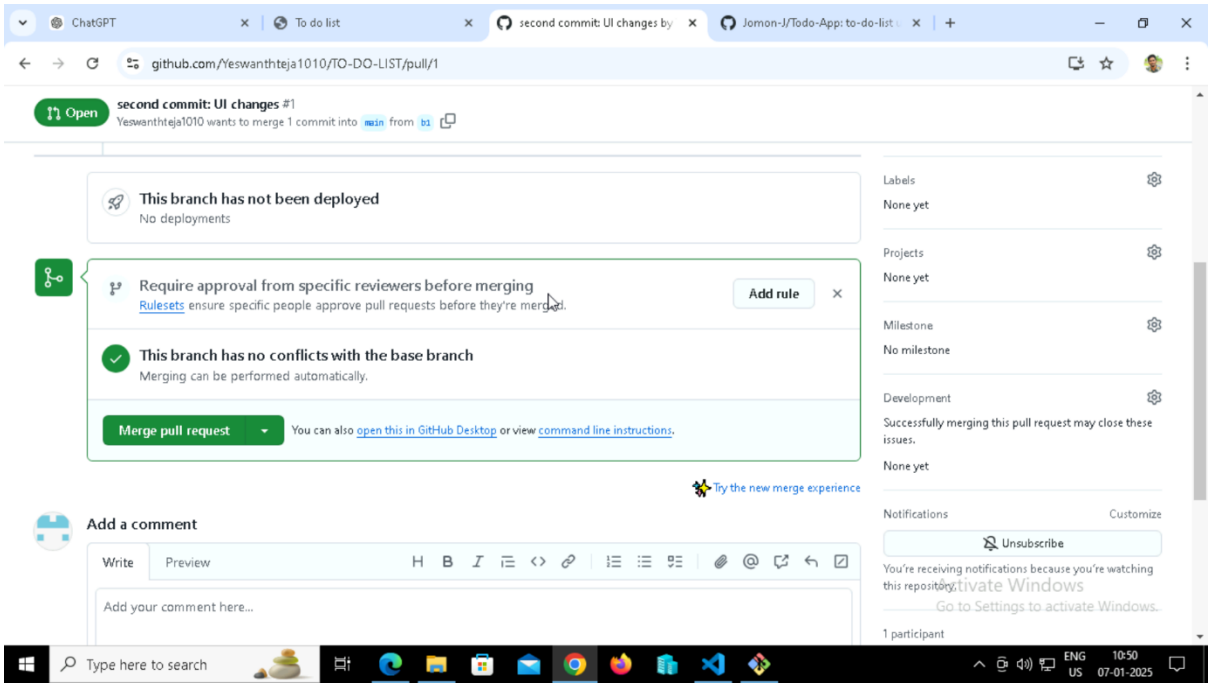


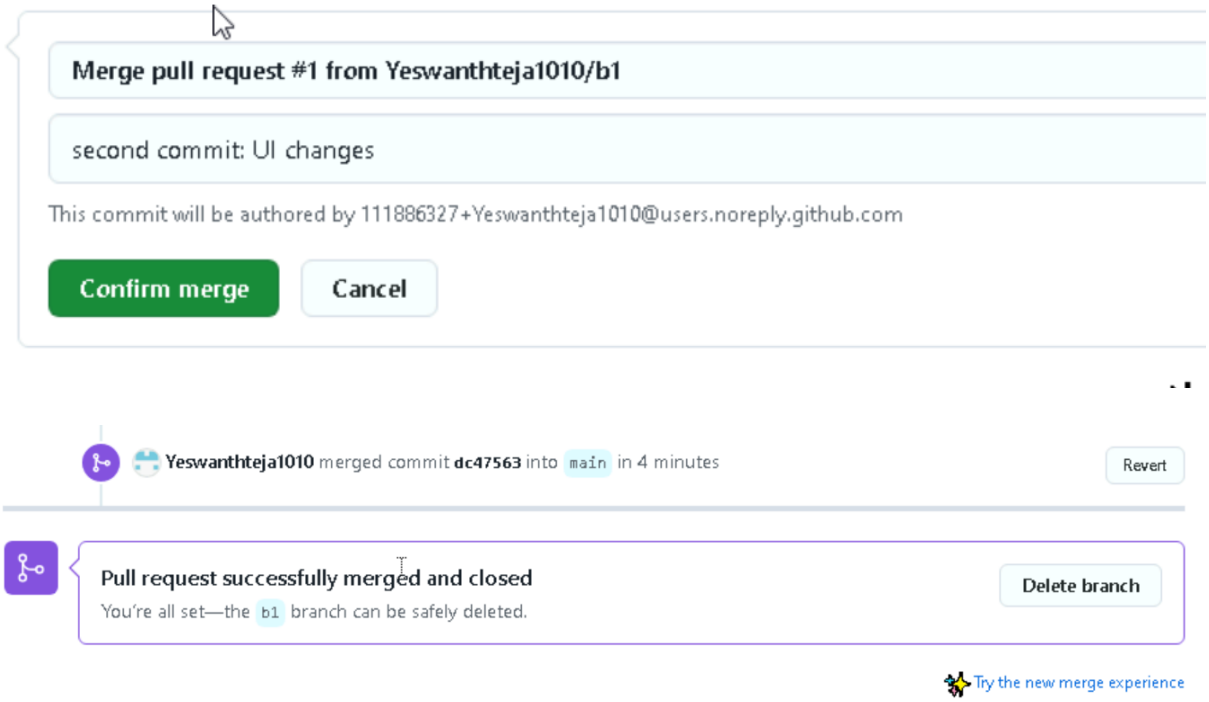
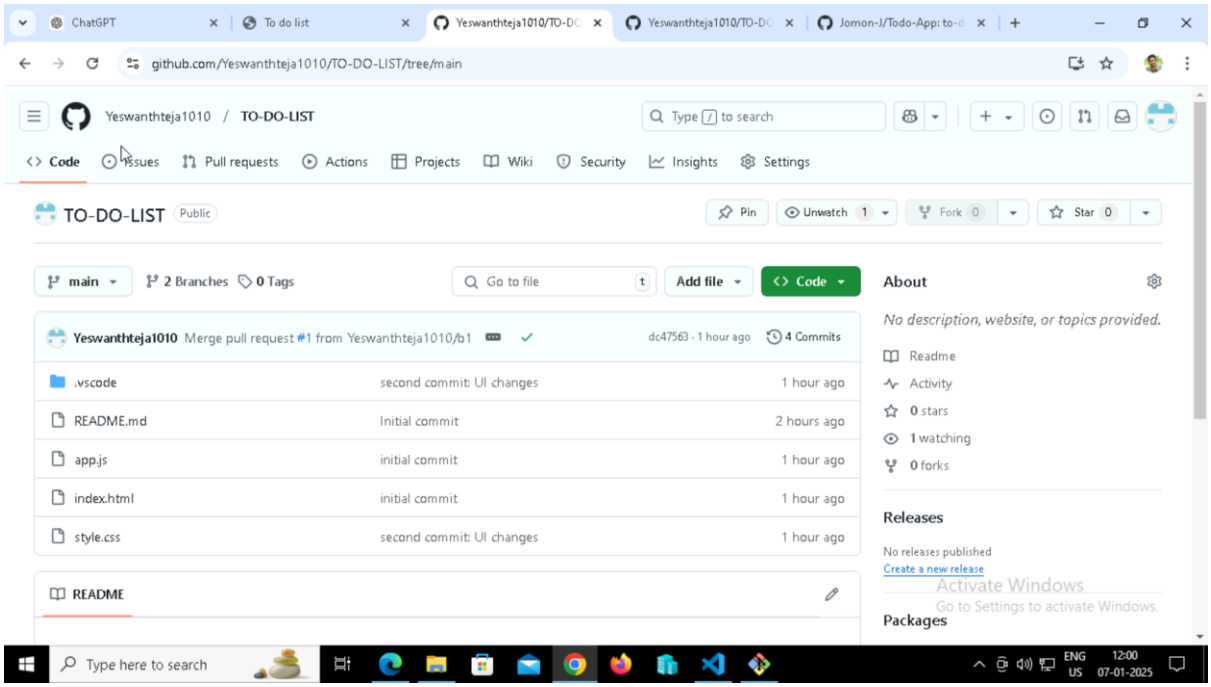
Step 23 : Checkout all the changes you made and click on create request. This will create a request with a title and description.





Step 24 : Now go to that request and click on merge request to merge our branch data with main





Step 25 : Now we can see merge is successful and all the files are present in main branch