Task 1:

```
$ git init
Initialized empty Git repository in C:/Users/dell/Desktop/git_repo_prac/.git/
```

Initialization of git repository

```
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        file1.txt

nothing added to commit but untracked files present (use "git add" to track)
```

To see state of the Directory and Staging Area. Also useful to find untracked files.

```
dell@DESKTOP-1QUPSPC MINGW64 ~/Desktop/git_repo_prac (master)
$ git add .

dell@DESKTOP-1QUPSPC MINGW64 ~/Desktop/git_repo_prac (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   file1.txt
```

To make the git track the untracked files. Which means we are adding the files to the staging area.

```
dell@DESKTOP-1QUPSPC MINGW64 ~/Desktop/git_repo_prac (master)
$ git rm --cached file1.txt
rm 'file1.txt'

dell@DESKTOP-1QUPSPC MINGW64 ~/Desktop/git_repo_prac (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        file1.txt

nothing added to commit but untracked files present (use "git add" to track)
```

To make the tracked files untracked.

```
$ git commit -m "First Commit"
[master (root-commit) b362800] First Commit
 1 file changed, 1 insertion(+)
 create mode 100644 file1.txt
```

Commiting the files to the repository from the files in the staging area. This is a saving point for the git.

```
$ git branch
* master
```

We can see the list of branches and also the present branch. Branch -> New separate version of main repo.

```
$ git branch -M main

dell@DESKTOP-1QUPSPC MINGW64 ~/Desktop/
$ git branch
* main
```

Change the name of the branch.

```
dell@DESKTOP-1QUPSPC MINGW64 ~/Desktop/git_repo_prac (main)
$ git remote add origin https://github.com/sraghu1234/PracGit.git

dell@DESKTOP-1QUPSPC MINGW64 ~/Desktop/git_repo_prac (main)
$ git remote
origin

dell@DESKTOP-1QUPSPC MINGW64 ~/Desktop/git_repo_prac (main)
$ git remote -v
origin  https://github.com/sraghu1234/PracGit.git (fetch)
origin  https://github.com/sraghu1234/PracGit.git (push)
```

Local Repository is connected with Remote repository.

```
$ git remote rm origin

dell@DESKTOP-1QUPSPC MINGW64 ~/Desktop/git_repo_prac (main)
$ git remote -v
```

Removing the connection.

```
$ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Writing objects: 100% (3/3), 252 bytes | 126.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/RaguTeja/gitpractice.git
   b362800..d22b3a7  main -> main
```

Pushing the changes to the remote repositoy of main branch.

```
$ git branch main2

dell@DESKTOP-1QUPSPC MINGW64 ~/Desktop/git_repo_prac (main)
$ git branch
* main
  main2

dell@DESKTOP-1QUPSPC MINGW64 ~/Desktop/git_repo_prac (main)
$ git checkout main2
Switched to branch 'main2'

dell@DESKTOP-1QUPSPC MINGW64 ~/Desktop/git_repo_prac (main2)
$ git branch
  main
* main2
```

Switching from one branch to other.

```
$ git ls-files
file 2.txt
file1.txt
```

List of files in that branch.

```
$ git branch -d main3
Deleted branch main3 (was 05bf6ea).

dell@DESKTOP-1QUPSPC MINGW64 ~/Desktop/git_repo_prac (mai
$ git branch
* main
  main2
```

Branch Deletion.

```
$ git merge main3
Updating d22b3a7..05bf6ea
Fast-forward
 file 2.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 file 2.txt
```

Merging of Branches.

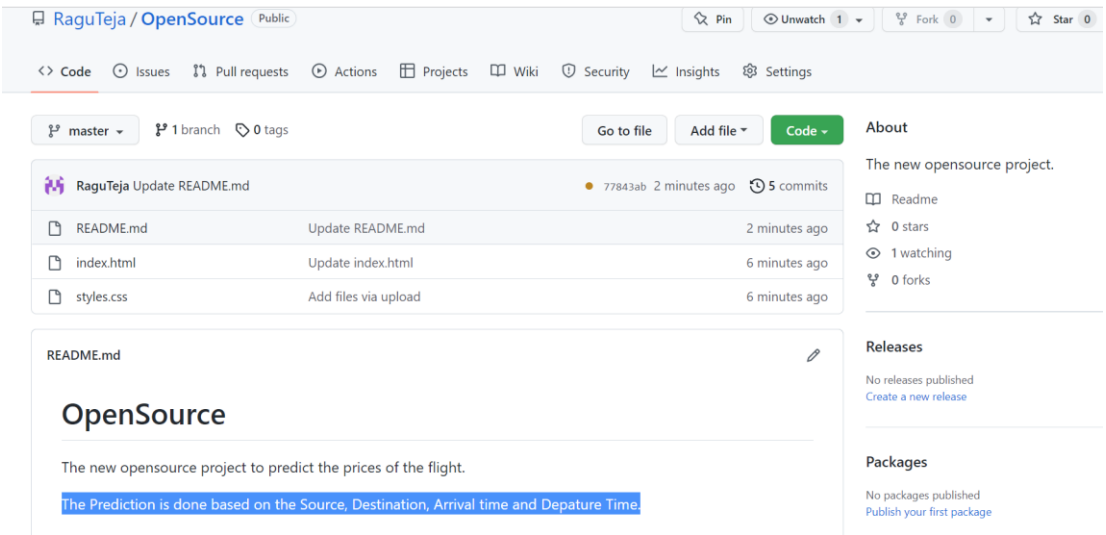
```
$ git push origin main
To https://github.com/RaguTeja/gitpractice.git
 ! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'https://github.com/RaguTeja/gitpractice.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.

dell@DESKTOP-1QUPSPC MINGW64 ~/Desktop/git_repo_prac (main)
$ git pull origin main
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 666 bytes | 55.00 KiB/s, done.
From https://github.com/RaguTeja/gitpractice
 * branch            main       -> FETCH_HEAD
   05bf6ea..a306a23  main       -> origin/main
Updating 05bf6ea..a306a23
Fast-forward
```

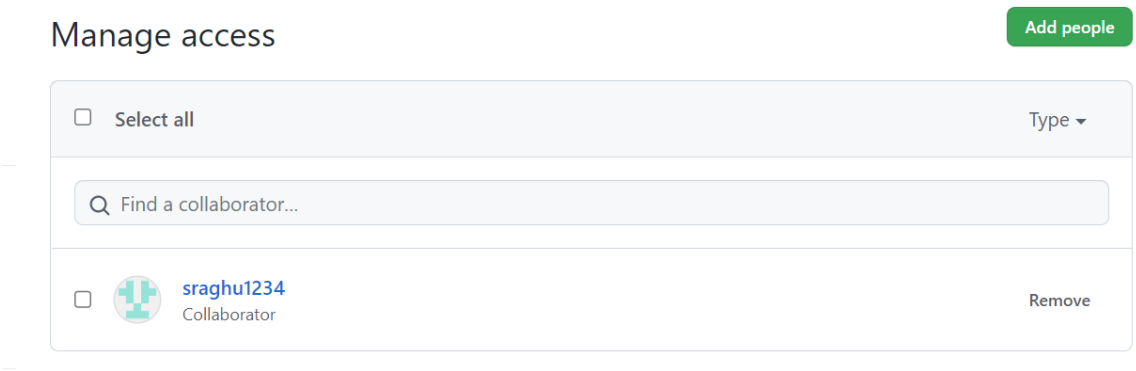
When there is a conflict, we need to pull the remote repository to local to make them having same data.

==========

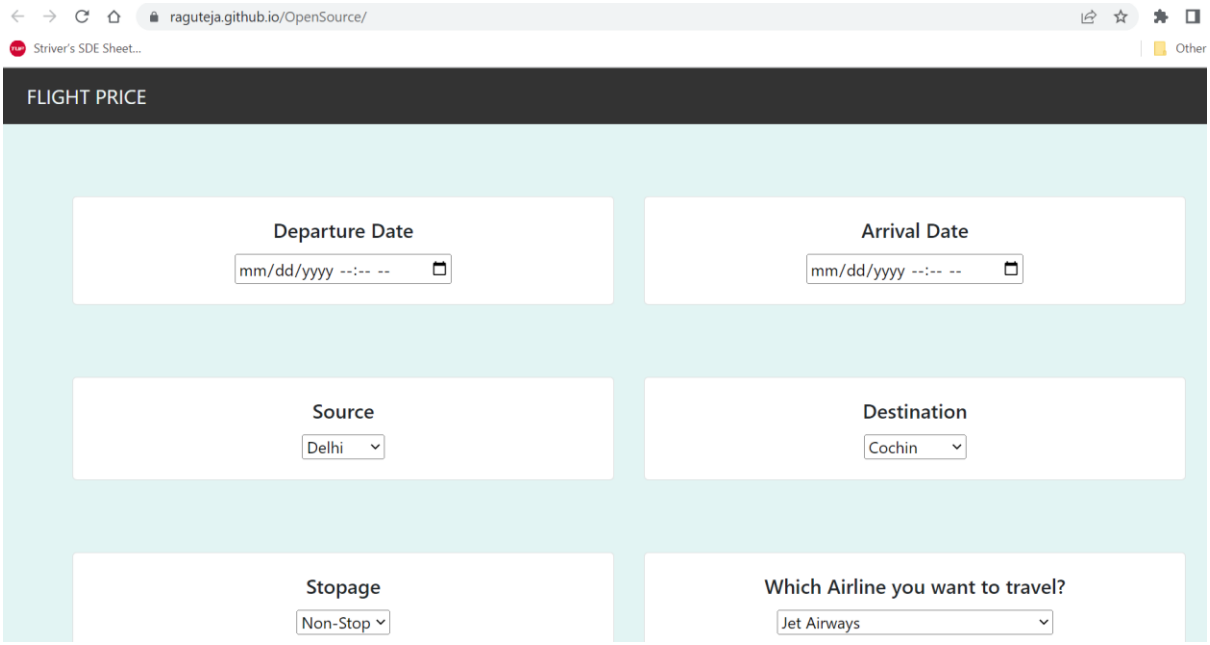
Task 2:



Created an opensource Public repository with README.md file.



Added a Collaborator.



Hosted a github page successfully.

==========

Task 3:

All steps are done successfully – please check my github.

Git stash : if you want to save your uncommitted changes for later use then git stash is useful.

```
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   main.py

no changes added to commit (use "git add" and/or "git commit -a")
(base) PS C:\Users\dell\Desktop\OpenSource> git stash
Saved working directory and index state WIP on main: 92de816 Update main.py
(base) PS C:\Users\dell\Desktop\OpenSource> git status
On branch main
Your branch is up to date with 'origin/master'.
```

If you want to work again on stashed changes. Then we should use git stash pop.

```
(base) PS C:\Users\dell\Desktop\OpenSource> git stash pop
On branch main
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   main.py

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (9bcd7e644b054f9a9fe320ae8f7f29c37566e9cb)
(base) PS C:\Users\dell\Desktop\OpenSource> git status
On branch main
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   main.py
```