1- Tasks to Be Performed:

A:- Based on what you have learnt in the class, do the following steps:

a. Create a new folder :-

```
PANKAJ@DESKTOP-JUGM1PP MINGW64 /d
$ mkdir Git_Assignment

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d
$ ls -a
'$RECYCLE.BIN'/       Developer/              'System Volume Information'/
 ./                   Git_Assignment/         'Update With PAW'/
 ../                 'Java Program'/          html/
 AWS/                 Movies/                 nadim/
'AWS Internship'/     MyConsoleApp/           'programs in c'/
 Bhushan/            'PW Skills'/             python_env/
 CSharp/              Python/
'DevOps Internship'/  Python_Assigmnent_1.docx
```

Make the folder name — Git_Assignment

b. Put the following files in the folder
   Code.txt
   Log.txt
   Output.txt

```
PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/Git_Assignment
$ touch code.txt output.txt log.txt

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/Git_Assignment
$ ls -a
./  ../  code.txt  log.txt  output.txt
```

Putting the file inside the folder Git_Assignment

c. Stage the Code.txt and Output.txt files

```
PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/Git_Assignment (master)
$ git add code.txt output.txt

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/Git_Assignment (master)
$ git status
On branch master

No commits yet

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

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

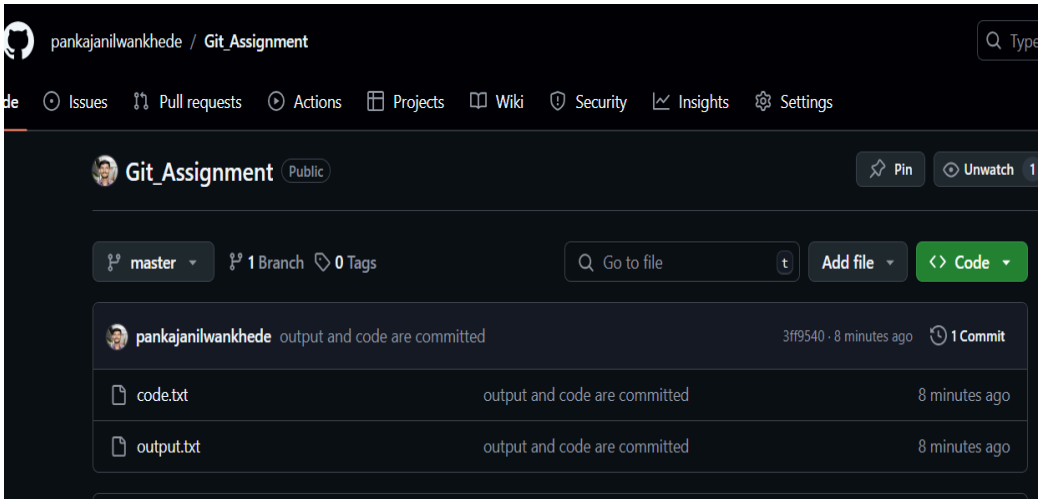
d. Commit them

```
PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/Git_Assignment (master)
$ git commit -m "output and code are committed"
[master (root-commit) 3ff9540] output and code are committed
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 code.txt
 create mode 100644 output.txt

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/Git_Assignment (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        log.txt

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

e. And finally push them to GitHub

```
PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/Git_Assignment (master)
$ git push -u origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 242 bytes | 242.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/pankajanilwankhede/Git_Assignment.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

Code.txt and Output.txt are successfully shown on the github after pushing them



B:- Commands that are present on the above points –

1:- mkdir to make the directory

2:- touch to make the file

3:- git init to initialize the git

4 :- git add to add the files into staging area
5:- git commit
6 :- git push

2:- Tasks to Be Performed:

1.Create a Git working directory with feature1.txt and feature2.txt in the master branch

```
PANKAJ@DESKTOP-JUGM1PP MINGW64 /d
$ mkdir my_project

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d
$ cd my_project

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project
$ touch feature1.txt feature2.txt

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project
$ ls -a
./  ../  feature1.txt  feature2.txt

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project
$ git init
Initialized empty Git repository in D:/my_project/.git/

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project (master)
$ git add .

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   feature1.txt
        new file:   feature2.txt


PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project (master)
$ git commit -m "Added feature1.txt and feature2.txt"
[master (root-commit) b0ed775] Added feature1.txt and feature2.txt
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 feature1.txt
 create mode 100644 feature2.txt
```

2. Create 3 branches develop, feature1 and feature2

```
PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project (master)
$ git branch develop

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project (master)
$ git branch
  develop
* master

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project (master)
$ git branch feature1

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project (master)
$ git branch feature2

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project (master)
$ git branch
  develop
  feature1
  feature2
* master
```

3. In develope branch create develop.txt, do not stage or commit it

```
PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project (master)
$ git checkout develop
Switched to branch 'develop'

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project (develop)
$ git branch
* develop
  feature1
  feature2
  master

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project (develop)
$ touch develop.txt
```

4. Stash this file and checkout to feature1 branch.

```
PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project (develop)
$ git add develop.txt
warning: in the working copy of 'develop.txt', LF will be replaced by CRLF the next time Git touches it

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project (develop)
$ git stash
Saved working directory and index state WIP on develop: b0ed775 Added feature1.txt and feature2.txt

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project (develop)
$ git stash list
stash@{0}: WIP on develop: b0ed775 Added feature1.txt and feature2.txt

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project (develop)
$ ls -a
./  ../  .git/  feature1.txt  feature2.txt

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project (develop)
$ git checkout feature1
Switched to branch 'feature1'

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project (feature1)
$ git branch
  develop
* feature1
  feature2
  master
```

5.Create new.txt file in feature1 branch, stage and commit this file

```
PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project (feature1)
$ git branch
  develop
* feature1
  feature2
  master

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project (feature1)
$ touch new.txt

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project (feature1)
$ git add new.txt

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project (feature1)
$ git commit -m "Commited new.txt"
[feature1 42ba84e] Commited new.txt
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 new.txt

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project (feature1)
$ git status
On branch feature1
nothing to commit, working tree clean
```

6. Checkout to develop, unstash this file and commit

```
PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project (feature1)
$ git checkout develop
Switched to branch 'develop'

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project (develop)
$ git branch
* develop
  feature1
  feature2
  master

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project (develop)
$ ls -a
./  ../  .git/  feature1.txt  feature2.txt

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project (develop)
$ git stash apply
On branch develop
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   develop.txt


PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project (develop)
$ git stash pop
On branch develop
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   develop.txt

Dropped refs/stash@{0} (747c83c268a0fb16c49040bcfa91127f06ef5236)

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project (develop)
$ ls -a
./  ../  .git/  develop.txt  feature1.txt  feature2.txt
```

7. Please submit all the Git commands used to do the above step.
   Mkdir,cd,touch,git init,git commit,git branch,git checkout,git stash,git stash pop

3:-Tasks to Be Performed:

1. Create a Git working directory, with the following branches:
   Develop
   f1
   f2

```
PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project2 (master)
$ ls -a
./  ../  .git/  pankaj.js

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project2 (master)
$ git branch
  develop
* master

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project2 (master)
$ git branch f1

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project2 (master)
$ git branch f2

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project2 (master)
$ git branch
  develop
  f1
  f2
* master
```

2. In the master branch, commit main.txt file

```
PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project2 (master)
$ touch main.txt

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project2 (master)
$ git add .

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project2 (master)
$ git commit -m "commited master.txt"
[master 83e696f] commited master.txt
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 main.txt
```

3. Put develop.txt in develop branch, f1.txt and f2.txt in f1 and f2 respectively

```
PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project2 (master)
$ git checkout develop
Switched to branch 'develop'

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project2 (develop)
$ touch develop.txt

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project2 (develop)
$ git checkout f1
Switched to branch 'f1'

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project2 (f1)
$ touch f1.txt

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project2 (f1)
$ git checkout f2
Switched to branch 'f2'

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project2 (f2)
$ touch f2.txt
```

4. Push all these branches to GitHub

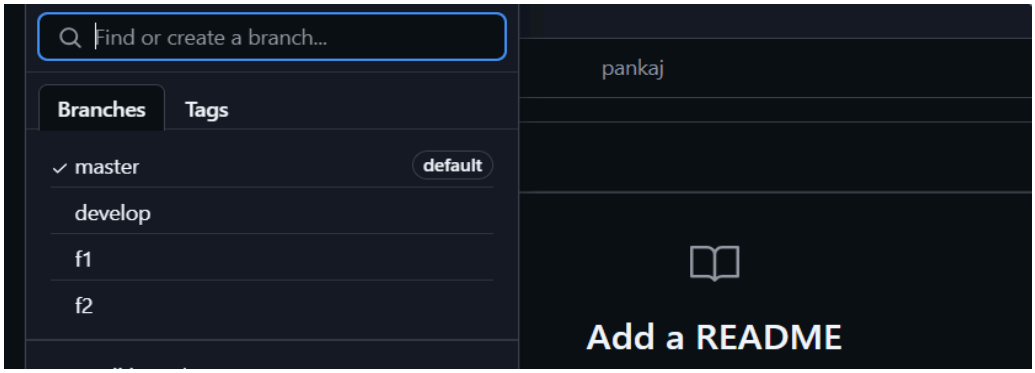
```
PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project2 (f2)
$ git branch
  develop
  f1
* f2
  master

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project2 (f2)
$ git remote -v
origin  https://github.com/pankajanilwankhede/Git_Assignment.git (fetch)
origin  https://github.com/pankajanilwankhede/Git_Assignment.git (push)

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project2 (f2)
$ git push --all origin
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (9/9), 660 bytes | 660.00 KiB/s, done.
Total 9 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), done.
To https://github.com/pankajanilwankhede/Git_Assignment.git
   535904b..0bf350c  master -> master
 * [new branch]      develop -> develop
 * [new branch]      f1 -> f1
 * [new branch]      f2 -> f2
```

5. On local delete f2 branch

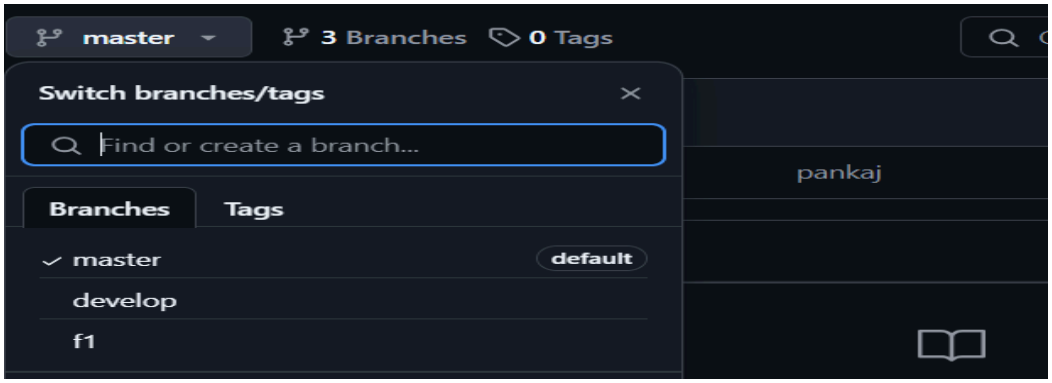
```
PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project2 (master)
$ git branch -D f2
Deleted branch f2 (was 3be6b80).

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project2 (master)
$ git branch
  develop
  f1
* master
```

6. Delete the same branch on GitHub as well

Before deleting the f2 is present in github



After deleting the f2 is not present




```
PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project2 (master)
$ git push origin --delete f2
To https://github.com/pankajanilwankhede/Git_Assignment.git
 - [deleted]         f2
```

4:-Tasks to Be Performed:

1. Put master.txt on master branch, stage and commit

```
PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project3 (master)
$ touch master.txt

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project3 (master)
$ git add .

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project3 (master)
$ git commit -m "master commit"
[master (root-commit) 852c0be] master commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 master.txt
```

2. Create 3 branches: public 1, public 2 and private

```
PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project3 (master)
$ git branch public1

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project3 (master)
$ git branch public2

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project3 (master)
$ git branch private

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project3 (master)
$ git branch
* master
  private
  public1
  public2
```

3. Put public1.txt on public 1 branch, stage and commit

```
PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project3 (master)
$ git checkout public1
Switched to branch 'public1'

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project3 (public1)
$ touch public1.txt

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project3 (public1)
$ git add .

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project3 (public1)
$ git commit -m "public 1 commit"
[public1 b28a394] public 1 commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 public1.txt
```

4. Merge public 1 on master branch

```
PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project3 (master)
$ git merge public1
Updating 852c0be..b28a394
Fast-forward
 public1.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 public1.txt

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project3 (master)
$ ls -l
total 0
-rw-r--r-- 1 PANKAJ 197121 0 Oct  7 19:44 master.txt
-rw-r--r-- 1 PANKAJ 197121 0 Oct  7 20:37 public1.txt
```

5. Merge public 2 on master branch

```
PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project3 (master)
$ git merge public2
Merge made by the 'ort' strategy.
 public2.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 public2.txt
```

6. Edit master.txt on private branch, stage and commit

```
PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project3 (private)
$ vim master.txt

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project3 (private)
$ git add .
warning: in the working copy of 'master.txt', LF will be replaced by CRLF the next time Git touches it

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project3 (private)
$ git commit -m "Edit master.txt"
[private 57318af] Edit master.txt
 1 file changed, 1 insertion(+)
```

7. Now update branch public 1 and public 2 with new master code in private

```
PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project3 (master)
$ git checkout public1
Switched to branch 'public1'

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project3 (public1)
$ git merge private
Merge made by the 'ort' strategy.
 master.txt | 1 +
 1 file changed, 1 insertion(+)

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project3 (public1)
$ ls -l
total 1
-rw-r--r-- 1 PANKAJ 197121 22 Oct  7 20:54 master.txt
-rw-r--r-- 1 PANKAJ 197121  0 Oct  7 20:49 public1.txt

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project3 (public1)
$ vim master.txt
```

```
PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project3 (public1)
$ git checkout public2
Switched to branch 'public2'

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project3 (public2)
$ ls -l
total 0
-rw-r--r-- 1 PANKAJ 197121 0 Oct  7 20:55 master.txt
-rw-r--r-- 1 PANKAJ 197121 0 Oct  7 20:55 public2.txt

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project3 (public2)
$ vim master.txt

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project3 (public2)
$ git merge private
Merge made by the 'ort' strategy.
 master.txt | 1 +
 1 file changed, 1 insertion(+)
```

8. Also update new master code on master

```
PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project3 (master)
$ git merge private
Merge made by the 'ort' strategy.
 master.txt | 1 +
 1 file changed, 1 insertion(+)
```

9. Finally update all the code on the private branch

```
PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project3 (master)
$ git checkout private
Switched to branch 'private'

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project3 (private)
$ git merge public1
Updating 57318af..99f871b
Fast-forward
 public1.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 public1.txt

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project3 (private)
$ git merge public2
Merge made by the 'ort' strategy.
 public2.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 public2.txt

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project3 (private)
$ git merge master
Merge made by the 'ort' strategy.

PANKAJ@DESKTOP-JUGM1PP MINGW64 /d/my_project3 (private)
$ ls -l
total 1
-rw-r--r-- 1 PANKAJ 197121 22 Oct  7 20:57 master.txt
-rw-r--r-- 1 PANKAJ 197121  0 Oct  7 21:47 public1.txt
-rw-r--r-- 1 PANKAJ 197121  0 Oct  7 21:47 public2.txt
```

5:-Tasks to Be Performed:

1. Create a Git Flow workflow architecture on Git
2. Create all the required branches
3. starting from e feature branch, push the branch to the master, following the architecture
4. Push an urgent.txt on master using hotfix

I can’t understand this topic because it is not in the Lecture uploaded by PWSKILL.