

ASSIGNMENT 1 (GITHUB)

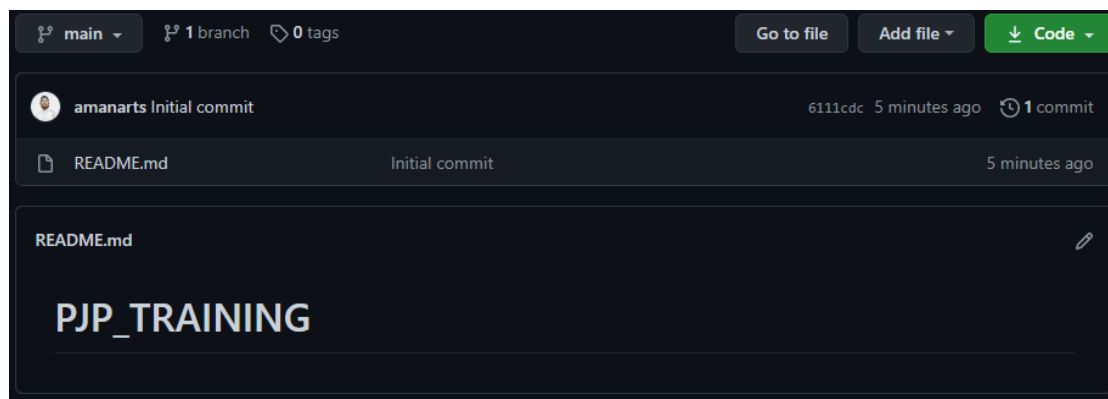
AMAN GUPTA (amanarts1.0@gmail.com)

A: BASIC USAGE OF THE CLI

1: Setup a local repository

```
amanarts@LAPTOP-VFDANDVM MINGW64 ~/PJP_TRAINING/week1
$ git init
Initialized empty Git repository in C:/Users/amanarts/PJP_TRAINING/week1/.git/
```

2: Setup a remote repository



```
amanarts@LAPTOP-VFDANDVM MINGW64 ~/PJP_TRAINING/week1 (master)
$ git remote add origin https://github.com/amanarts/PJP_TRAINING.git

amanarts@LAPTOP-VFDANDVM MINGW64 ~/PJP_TRAINING/week1 (master)
$ git push origin master
fatal: credential-cache unavailable; no unix socket support
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 231 bytes | 231.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/amanarts/PJP_TRAINING/pull/new/master
remote:
To https://github.com/amanarts/PJP_TRAINING.git
 * [new branch]      master -> master
```

3: Create a local branches

```

amanarts@LAPTOP-VFDANDVM MINGW64 ~/PJP_TRAINING/week1 (master)
$ git checkout -b feature
Switched to a new branch 'feature'

amanarts@LAPTOP-VFDANDVM MINGW64 ~/PJP_TRAINING/week1 (feature)
$ git checkout -b dev
Switched to a new branch 'dev'

amanarts@LAPTOP-VFDANDVM MINGW64 ~/PJP_TRAINING/week1 (dev)
$ git checkout -b QA
Switched to a new branch 'QA'

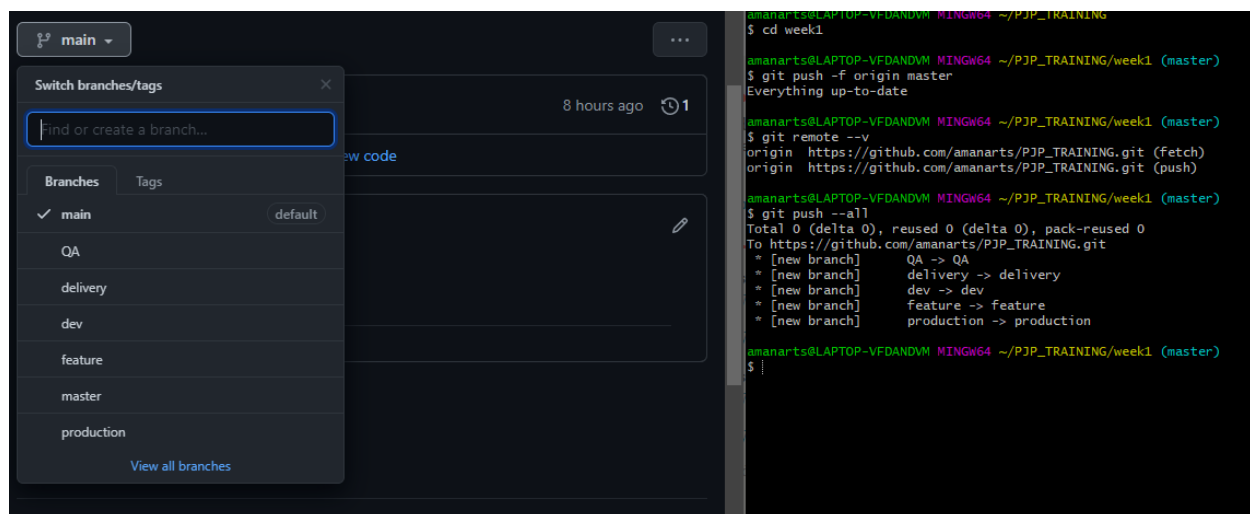
amanarts@LAPTOP-VFDANDVM MINGW64 ~/PJP_TRAINING/week1 (QA)
$ git checkout -b production
Switched to a new branch 'production'

amanarts@LAPTOP-VFDANDVM MINGW64 ~/PJP_TRAINING/week1 (production)
$ git checkout -b delivery
Switched to a new branch 'delivery'

amanarts@LAPTOP-VFDANDVM MINGW64 ~/PJP_TRAINING/week1 (delivery)
$ git branch
  QA
* delivery
  dev
  feature
  master
  production

```

4: Create remote branches



The screenshot shows a code editor interface. On the left, a 'Switch branches/tags' dialog is open, displaying a list of local branches: main (checked), QA, delivery, dev, feature, master, and production. The 'main' branch is marked as the default. On the right, a terminal window shows the following commands and output:

```

amanarts@LAPTOP-VFDANDVM MINGW64 ~/PJP_TRAINING
$ cd week1

amanarts@LAPTOP-VFDANDVM MINGW64 ~/PJP_TRAINING/week1 (master)
$ git push -f origin master
Everything up-to-date

amanarts@LAPTOP-VFDANDVM MINGW64 ~/PJP_TRAINING/week1 (master)
$ git remote -v
origin https://github.com/amanarts/PJP_TRAINING.git (fetch)
origin https://github.com/amanarts/PJP_TRAINING.git (push)

amanarts@LAPTOP-VFDANDVM MINGW64 ~/PJP_TRAINING/week1 (master)
$ git push --all
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/amanarts/PJP_TRAINING.git
 * [new branch]    QA -> QA
 * [new branch]    delivery -> delivery
 * [new branch]    dev -> dev
 * [new branch]    feature -> feature
 * [new branch]    production -> production

amanarts@LAPTOP-VFDANDVM MINGW64 ~/PJP_TRAINING/week1 (master)
$

```

5: Add update and delete files/folder

```
amanarts@LAPTOP-VFDANDVM MINGW64 ~/PJP_TRAINING/week1 (master)
$ git checkout feature
Switched to branch 'feature'

amanarts@LAPTOP-VFDANDVM MINGW64 ~/PJP_TRAINING/week1 (feature)
$ notepad doc.txt

amanarts@LAPTOP-VFDANDVM MINGW64 ~/PJP_TRAINING/week1 (feature)
$ git status
On branch feature
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    doc.txt

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

amanarts@LAPTOP-VFDANDVM MINGW64 ~/PJP_TRAINING/week1 (feature)
$ git add .

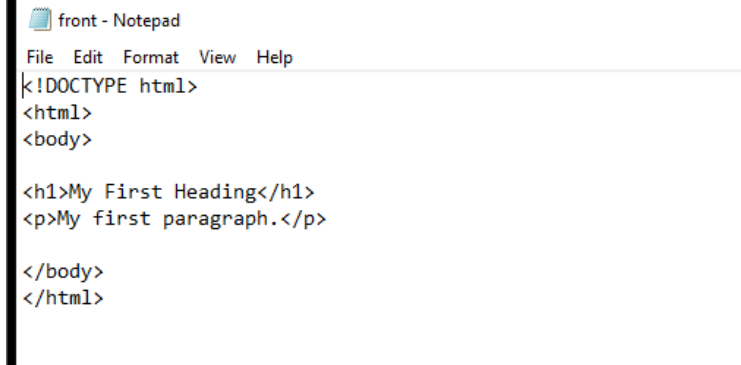
amanarts@LAPTOP-VFDANDVM MINGW64 ~/PJP_TRAINING/week1 (feature)
$ git commit -m "added document"
[feature dc71cda] added document
1 file changed, 1 insertion(+)
create mode 100644 doc.txt

amanarts@LAPTOP-VFDANDVM MINGW64 ~/PJP_TRAINING/week1 (feature)
$ git push origin feature
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 280 bytes | 280.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/amanarts/PJP_TRAINING.git
46de85c..dc71cda feature -> feature
```

Modification of files/folders

```
amanarts@LAPTOP-VFDANDVM MINGW64 ~/PJP_TRAINING/week1 (feature)
$ notepad front.html

amanarts@LAPTOP-VFDANDVM MINGW64 ~/PJP_TRAINING/week1 (feature)
$ notepad front.html
```



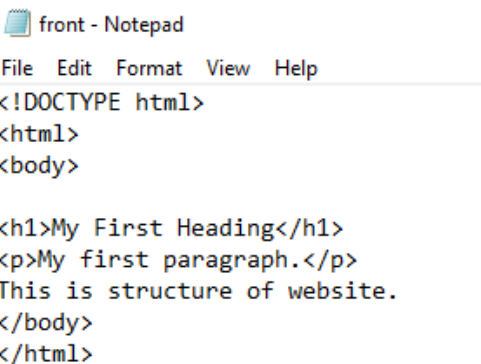
The screenshot shows a Notepad window titled "front - Notepad". The menu bar includes File, Edit, Format, View, and Help. The text content is as follows:

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

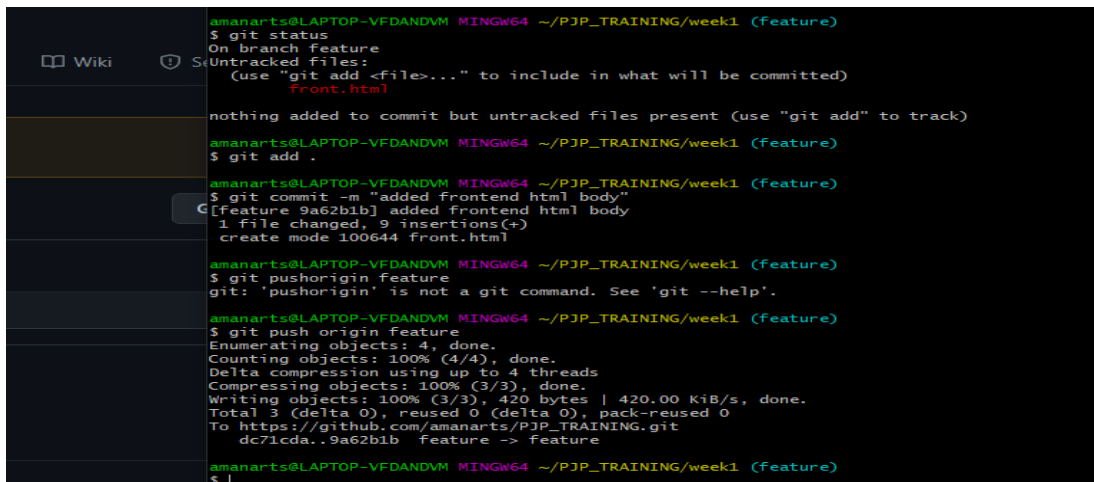
File front.html edited



The screenshot shows the same Notepad window with the following updated content:

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>
This is structure of website.
</body>
</html>
```



```
amanarts@LAPTOP-VFDANDVM MINGW64 ~/PJP_TRAINING/week1 (feature)
$ git status
On branch feature
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    front.html

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

amanarts@LAPTOP-VFDANDVM MINGW64 ~/PJP_TRAINING/week1 (feature)
$ git add .

amanarts@LAPTOP-VFDANDVM MINGW64 ~/PJP_TRAINING/week1 (feature)
$ git commit -m "added frontend html body"
[feature 9a62b1b] added frontend html body
1 file changed, 9 insertions(+)
create mode 100644 front.html

amanarts@LAPTOP-VFDANDVM MINGW64 ~/PJP_TRAINING/week1 (feature)
$ git pushorigin feature
git: 'pushorigin' is not a git command. See 'git --help'.

amanarts@LAPTOP-VFDANDVM MINGW64 ~/PJP_TRAINING/week1 (feature)
$ git push origin feature
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 420 bytes | 420.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/amanarts/PJP_TRAINING.git
dc71cda..9a62b1b feature -> feature

amanarts@LAPTOP-VFDANDVM MINGW64 ~/PJP_TRAINING/week1 (feature)
$
```

6: Check in, stage, commit, push changes into feature branch

```
amanarts@LAPTOP-VFDANDVM MINGW64 ~/PJP_TRAINING/week1 (feature)
$ notepad front.html

amanarts@LAPTOP-VFDANDVM MINGW64 ~/PJP_TRAINING/week1 (feature)
$ git status
On branch feature
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:   front.html

no changes added to commit (use "git add" and/or "git commit -a")

amanarts@LAPTOP-VFDANDVM MINGW64 ~/PJP_TRAINING/week1 (feature)
$ git commit -m "modified html file"
On branch feature
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:   front.html

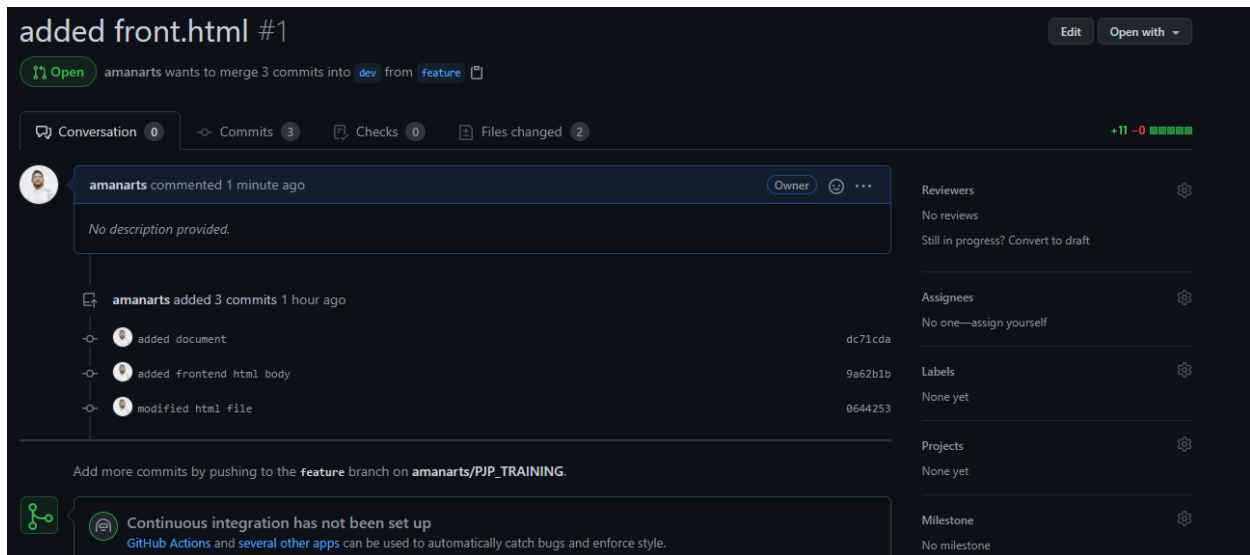
no changes added to commit (use "git add" and/or "git commit -a")

amanarts@LAPTOP-VFDANDVM MINGW64 ~/PJP_TRAINING/week1 (feature)
$ git add .

amanarts@LAPTOP-VFDANDVM MINGW64 ~/PJP_TRAINING/week1 (feature)
$ git commit -m "modified html file"
[feature 0644253] modified html file
1 file changed, 1 insertion(+)

amanarts@LAPTOP-VFDANDVM MINGW64 ~/PJP_TRAINING/week1 (feature)
$ git push origin feature
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 294 bytes | 294.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/amanarts/PJP_TRAINING.git
9a62b1b..0644253 feature -> feature
```

7: Promote code from feature branch to dev branch via pull requests.



Checkout code from remote to local branch

```
amanarts@LAPTOP-VFDANDVM MINGW64 ~/PJP_TRAINING/week1 (dev)
$ git fetch origin dev
remote: Enumerating objects: 1, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (1/1), 628 bytes | 11.00 KiB/s, done.
From https://github.com/amanarts/PJP_TRAINING
* branch      dev      -> FETCH_HEAD
   46de85c..7653b79 dev    -> origin/dev

amanarts@LAPTOP-VFDANDVM MINGW64 ~/PJP_TRAINING/week1 (dev)
$ git log
commit 46de85c1ba9ee291ea5cc6aea4b0d25ec2adedef (HEAD -> dev, origin/production, origin/master
, origin/delivery, origin/QA, production, master, delivery, QA)
Author: Amanarts <amanarts007@gmail.com>
Date:   Sat Mar 13 20:46:29 2021 +0530

    first commit

amanarts@LAPTOP-VFDANDVM MINGW64 ~/PJP_TRAINING/week1 (dev)
$ git checkout/dev
git: 'checkout/dev' is not a git command. See 'git --help'.

amanarts@LAPTOP-VFDANDVM MINGW64 ~/PJP_TRAINING/week1 (dev)
$ git checkout origin/dev
Note: switching to 'origin/dev'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

    git switch -c <new-branch-name>

Or undo this operation with:

    git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at 7653b79 Merge pull request #1 from amanarts/feature
```

```

amanarts@LAPTOP-VFDANDVM MINGW64 ~/PJP_TRAINING/week1 ((7653b79...))
$ git log
commit 7653b796ab0122a576c4059d8b6a4a996aabdd82 (HEAD, origin/dev)
Merge: 46de85c 0644253
Author: Aman Gupta <30577538+amanarts@users.noreply.github.com>
Date: Sun Mar 14 05:37:46 2021 +0530

    Merge pull request #1 from amanarts/feature

    added front.html

commit 064425308e85e08965d2574ffa859df8156af354 (origin/feature, feature)
Author: amanarts <amanarts007@gmail.com>
Date: Sun Mar 14 05:23:43 2021 +0530

    modified html file

commit 9a62b1ba1dcb1db370881c3a27ebb1883e2009cc
Author: amanarts <amanarts007@gmail.com>
Date: Sun Mar 14 05:17:31 2021 +0530

    added frontend html body

commit dc71cda670e2344a034ace804b5f7852f5e3f8b6
Author: amanarts <amanarts007@gmail.com>
Date: Sun Mar 14 04:46:10 2021 +0530

    added document

commit 46de85c1ba9ee291ea5cc6aea4b0d25ec2adedef (origin/production, origin/master, origin/deli
very, origin/QA, production, master, dev, delivery, QA)
Author: Amanarts <amanarts007@gmail.com>
Date: Sat Mar 13 20:46:29 2021 +0530

    first commit

amanarts@LAPTOP-VFDANDVM MINGW64 ~/PJP_TRAINING/week1 ((7653b79...))
$

```

Difference between Checkout vs Pull

checkout on a remote branch lets the user explore the branch on the local machine. The changes are not merged in the local repo.

pull fetches the changes from the remote repo and also merges them locally, effectively changing the local repo.

```

amanarts@LAPTOP-VFDANDVM MINGW64 ~/PJP_TRAINING/week1 ((7653b79...))
$ git pull origin dev
From https://github.com/amanarts/PJP_TRAINING
 * branch      dev          -> FETCH_HEAD
Already up to date.

amanarts@LAPTOP-VFDANDVM MINGW64 ~/PJP_TRAINING/week1 ((7653b79...))
$ |

```

Get two people to make changes to the same file, check in and handle merge conflicts

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.

Ensure code is in sync with the latest changes across all branches

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

B: Basic usage using the GUI client

Repeat the above steps using a GUI (VS code/GitHub client)

VS code has different extensions for stage, commit, merging and pushing to see changes done by each person.

1. Make local repository
 - a. Steps : make local folder in system like I made folder on desktop
VS_CODE_FIRST_PROJECT
 - b. Open VS code
 - c. Go to file section and open new folder.
 - d. Open folder from your system.

And then initialize the repository in VS code

2. 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.

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

Steps: Go to Source Contro -Branch -create Branch -branch name