**Lab 11**

**Branching & Merging in Git**

**Note**: No upload required for this exercise, it is a practice exercise in the use of Git.

This exercise will demonstrate the use of branching and merging in git and will need to be done in teams of two to see it working.

**Team Member 1**

Create a new folder called **demolab**, create & activate a virtual environment inside this folder and install Django into this environment.

Create a new Django project called **shop\_project** inside this folder.

(env) django-admin startproject shop\_project .

Open the project in VS Code and add in a .gitignore file and copy-paste the following contents:

# Environments

.env

.venv

env/

venv/

ENV/

env.bak/

venv.bak/

Go to GitHub and create a new repo called **TeamProject** and invite team member 2 as a collaborator.

In Windows Command Line launch the server to make sure the project is running and go to the web browser to open the web page to view the rocket launching.

In Windows Command Line, run the following git commands:

**Command Line**

git init

git add -A

git commit -m “first commit”

git branch -M main

git remote add origin ………..your repo…………

git push -u origin main

Go to GitHub and the project should be there:

Graphical user interface, text, application, email

Description automatically generated

**Team member 1**

You will create a new branch on which to work.

**Command Line**

Use the following command to create a new branch tm1branch

**git branch tm1branch**

Use the following command to switch to the tm1branch

**git checkout tm1branch**

Use the following command to see that you are now on the tm1branch (note the \* beside the current branch)

**git branch**

Create an **accounts** app and register this app in **settings.py**.

Use the following commands to stage your changes, commit them, and push your branch to the remote repository:

**Command Line**

git add -A

git commit -m “lab11part1”

Use the following command to push your branch up to the remote repository

git push -u origin tm1branch

Go to GitHub and you will see that the project now has two branches. When a team member completes a feature, they shouldn’t merge it into main straight away. The best approach is to push the feature branch which in this case is the **accounts** app to the central repo on GitHub and file a pull request asking to merge their additions into main. This gives other team members a chance to review the changes before they become a part of the main repo.

Graphical user interface

Description automatically generated

Now you want to create a pull request which is a way of asking the other team members for permission to merge your new feature into main.

Click on the branches button to view the different branches:



You should see a screen like the following. Click the button “New pull request”.

Graphical user interface, text, application, email

Description automatically generated

When you click the button to create a new pull request, you will see a screen like the following:

Graphical user interface, text, application, email

Description automatically generated

Click on the **Create pull request** button which will notify your team members that you have issues a pull request. In the screen shot above you will also see a green tick with the message “Able to merge” which means that GitHub hasn’t detected any merge issues.

As this branch feature is ok and has no merge conflicts we can go ahead and merge it into main.

When you look at the repo in GitHub you will see that there is now a pull request:

Graphical user interface, text, application, chat or text message

Description automatically generated

Click on the pull request and you will see an option to merge the branch into main:

Graphical user interface, text, application, email

Description automatically generated

You will then see an option to **confirm** the merge. Click this button & if you look at the repository you will see that main contains the updated files from the branch:

Graphical user interface, text, application, email

Description automatically generated

On your local computer there are a couple of ways for you to merge your branch into main. The following is the easiest way:

Use the following command to switch to the main branch

(env) TeamProject >git checkout main

Use the following command to merge the tm1branch into main

(env) TeamProject >git merge tm1branch

**Team member 2**

Go to your email and accept the invite to collaborate on the TeamProject repo, log into GitHub and you should see the repo.

**Team Member 2**

Use the following command to clone the TeamProject repo:

**Command Line**

**git clone …..repo url from Github……**

Move into the TeamProject folder and create the virtual environment

**py -m venv env**

**env\scripts\activate.bat**

Run the following command to install Django:

**pip install django**

Open the project in VS Code and you will see the **accounts** app there that was created by Team member 1.

You will create a **shop** app and to do this, you will create a new branch on which to work

**Command Line**

Use the following command to create a new branch tm2branch

git branch tm2branch

Use the following command to switch to the tm1branch

git checkout tm2branch

Use the following command to see that you are now on the tm2branch(note the \*)

git branch

In Windows Command Line create a new app called **shop** and in VS code register this app in **settings.py**.

Use the following commands to stage your changes, commit them and push your branch to the repo:

**Command Line**

git add -A

git commit -m “lab11part2”

Use the following command to push your branch up to the remote repository

git push -u origin tm2branch

If you look at the repository on GitHub you will see that there are 3 branches:

Graphical user interface, text, application

Description automatically generated

Click on the button called “3 branches”. You should see a screen like the following. Click the button “New pull request”.

Graphical user interface, text, application, email

Description automatically generated

When you click the button to create a new pull request, you will see a screen like the following:

Graphical user interface, text, application, email

Description automatically generated

Click on the **Create pull request** button which will notify your team members that you have issues a pull request. In the screen shot below you will also see a green tick with the message that GitHub hasn’t detected any merge conflicts.

Graphical user interface, text, application, email

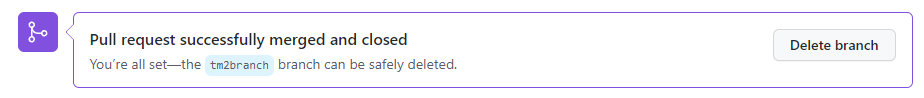
Description automatically generated

You can go ahead and merge this branch by clicking on the Merge pull request button highlighted in the screenshot above. You then confirm the merge by clicking the **Confirm merge** button:

Graphical user interface, text, application, email

Description automatically generated

Once the merge happens you also get an option to delete the branch if you wish. We will leave it there for the moment:



On your local computer you need to make sure that your main is up to date:

Use the following commands to switch to the main branch and merge the changes into main on your local computer:

git checkout main

git merge tm2branch

**Team member 1**

Your project contains the work done in Lab 11 Part 1. The remote repo contains the work from Team member 2 (Lab 11 Part 2). Before you start work on Lab 11 Part 3 you need to update your local repo with any changes from the remote repo.

Type the following command to fetch information about the remote repo (it does not update your local working files)

git fetch

Type the following command which will tell you that your branch is behind 'origin/main':

git status

Type the following command to pull the changes to the local repo:

git pull

In VS Code, you will see that the **shop** app is now there.

Now you can create a new branch and continue working on Lab 11 Part 3 to create a **cart** app.

**Command Line**

Use the following command to create a new branch part3branch

git branch part3branch

Use the following command to switch to the part3branch

git checkout part3branch

Use the following command to see that you are now on the part3branch (note the \* beside the current branch)

git branch

In Windows Command Line, create an app called **cart** and in VS Code register it in **settings.py**

**Command Line**

git add -A

git commit -m “lab 11 part 3”

git push -u origin part3branch

Go to the GitHub repository and view the branches and create a new pull request for this branch. When you do this, you should see the following indicating that it is ok to do the merge. Click the button “New pull request” and follow the steps.

Graphical user interface, text, application, email

Description automatically generated

On your local computer you need to make sure that your main is up to date:

Use the following commands to switch to the main branch and merge the changes from the branch into main.

git checkout main

git merge part3branch

**Team member 2**

The remote repo contains the work from team member 1. Before you start work on Lab 11 Part 4 you need to update your local repo with any changes from the remote repo.

Type the following command to fetch information about the remote repo (it does not update your local working files)

git fetch

Type the following command which will tell you that your branch is behind 'origin/main'

git status

Type the following command to pull the changes the local repo:

git pull

In VS Code, you will see that the **cart** app is now there.