Activity 09

Basic operations with Git

CSC 2310

In this lab you will perform basic operations using git. You will use the files in the repository to perform these operations.

Pre-work

Download the lab source files using the following command:

```
git clone https://gitlab.csc.tntech.edu/%userid%/act09_git.git
```

replacing `%userid%` with your own TNTech issued userid.

Concept

You are working on a project with many files. There are also many developers working on the project. Your goal is to maintain a productive and collaborative environment where multiple developers can submit their changes without affecting each other. In this exercise, you will clone a remote repository, create a branch, and perform other basic git operations.

Section 1

Step 1. Create TWO branches

Once you have checked out the project, you need to create two branchs. Name these branches as follows replacing `%userid%` with your own TNTech issued userid.

```
git checkout -b "<%userid%-personal-branch>"
```

```
git checkout -b "<%userid%-branch-for-merge>"
```

Example names would be: testuser-personal-branch, testuser-branch-for-merge.

Step 2. Make sure to switch to the first branch

```
git checkout "<%userid%-personal-branch>"
git branch
```

It should show a * next to your branch name.

Step 3. Change and add a file

Once you have created a branch, it's time to make changes.

- change the "Assignment" file. Include the following details: your name, change date, and the branch name
- create a new file named "alice_in_wonderland.pdf". You can use this publicly available file. If you can not download, create an empty file.

Step 4. Check the status of the new file

Check and verify the file is not added to the repository. You can do this using "git status" command. The file shoud be shown in red.

```
git status
```

Step 5. Add the file to them

- Add this new file to git.

```
git add alice_in_wonderland.pdf
```

Step 6. Check the status of the new file

Check and verify the file is now in the repository. You can do this using "git status" command. The file should be green.

```
git status
```

Step 7. Commit the changes

Once you have added the file and verified it, commit the change.

```
git commit -m 'changed README, added alice in wonderland'
```

Step 8. Push to the remote repository After committing, push the changes and the new branch to the remote repository.

```
git push --set-upstream origin <%your branch name>
```

You can find your branch name using

```
git branch
```

Section 2

In this section, we learn to merge different branches.

Step 1: Checkout the branch for merge After pushing the new branch, checkout the other branch: "<%userid%-branch-for-merge>

```
git checkout "<%userid%-branch-for-merge>"
```

An example name would be: testuser-branch-for-merge.

Step 2: Modify the file Once you have created this branch,

- change the "README" file. Include the following details: your name, change date, and the branch name

Step 3: Commit the changes and push Once you have changed the file, commit the change.

```
git commit -m 'changed README'
git push --set-upstream origin "<%userid%-branch-for-merge>"
```

Step 4: Merge this branch with master Merge this branch with the master branch.

```
git checkout main
git merge "<%userid%-branch-for-merge>"
```

Step 5: Push to the remote repository After merging, push the changes to the remote repository.

```
git push origin main
```

Step 6: Remove the branch you were working on First, remove the local branch

```
git branch -d "<%userid%-branch-for-merge>"
```

Then, remove the remote branch

```
git push origin :"<%userid%-branch-for-merge>"
```

Turn-in

By the end of the exercise, your gitlab repository will have two branches. One will be a branch with the README file and "alice_in_wonderland.pdf" in the assignment directory. The other branch will have just a README file in that directory.

Take the following screenshots and include them in *one* pdf file.

- Screenshot 1: In your gitlab repository, go to "Repository" -> "Graph" on the left hand menu. Take a screenshot, and attach to the submission. It should show three commits and a graph showing the branches.
- Screenshot 2: In your gitlab repository, go to "Repository" -> "Branches" on the left hand menu. It should show only two active branches.
- Screenshot 3: In your gitlab repository, go to "Repository" -> "Compare" on the left hand menu. Select the source as ""<%userid%-personal-branch>" and the target as "Main". DO NOT reverse the source and the target. Press the compare button, take a screenshot, and attach it to your submission.

Put all three screenshots in a pdf file, name it according to the follwing format (%username%-screenshot.pdf) and submit to iLearn for the Activity 09 assignment.