Submission

For this assignment, you need to

- submit the screenshot for the final question to the Lab assignment 7 in Brightspace.
- submit the link to *your branch* to the Lab assignment 5b in Brightspace. (You will know what *your branch* is later in this lab assingment.)

Install Git

Follow instructions from this link (https://git-scm.com/book/en/v2/Getting-Started-Installing-Git) to install Git.
Important note: (For Windows users), if asked during the installation, select the option that add git to the PATH environment so that you can use it from the command prompt.

In this lab assignment, you will be practicing fundamental git commands. For extensive usage of git, a graphical user interface (GUI) could be convenient. Source (https://git-scm.com/downloads/guis/) is here. Note that, we will not be using any GUI today.

Quick reference to git commands can be found from a cheatsheet. An example is from this link (https://dev.to/doabledanny/git-cheat-sheet-50-commands-free-pdf-and-poster-4gcn)

Setup your Github account

You should by now have an account on Github. If not, please follow the instructions provided in brightspace.

PART 1 - WORK AS A GROUP

A Github repository for your group

For your group, pick an account (amongst yours), and create a repository named `TIL6010-LabAssignments` . For the settings, use the following:

- Set the repository as Public.
- Select 'Add a README file'

Add collaborators

At your repository page, go to Settings. In the menu on the left, you will find Collaborators under Access. From there, invite your groupmates as collaborators for the repository

The other members need to accept invitations and should have access to the repository afterwards.

Clone the repository

Now, go to the main page of your repository (which is the `<> Code` page). You can notice that the repository has a branch main, a README.md file.

Now let's clone this repository to your local computer. Navigate to the `Code` button (highlighted in green) and click on that.

You will be provided with 3 different ways to clone a repository. Copy the link under HTTPS.
Now, open you terminal, change directory to location (in your computer) where you want to store the repository. (Use the `cd` command).
Clone the repository using the git clone command. Here is an example

```
git clone https://github.com/nguyenthientin/TIL6022-LabAssignments.git
```

If successful, you will have the folder TIL6022-LabAssignments created. Inside, there should be a README.md file.

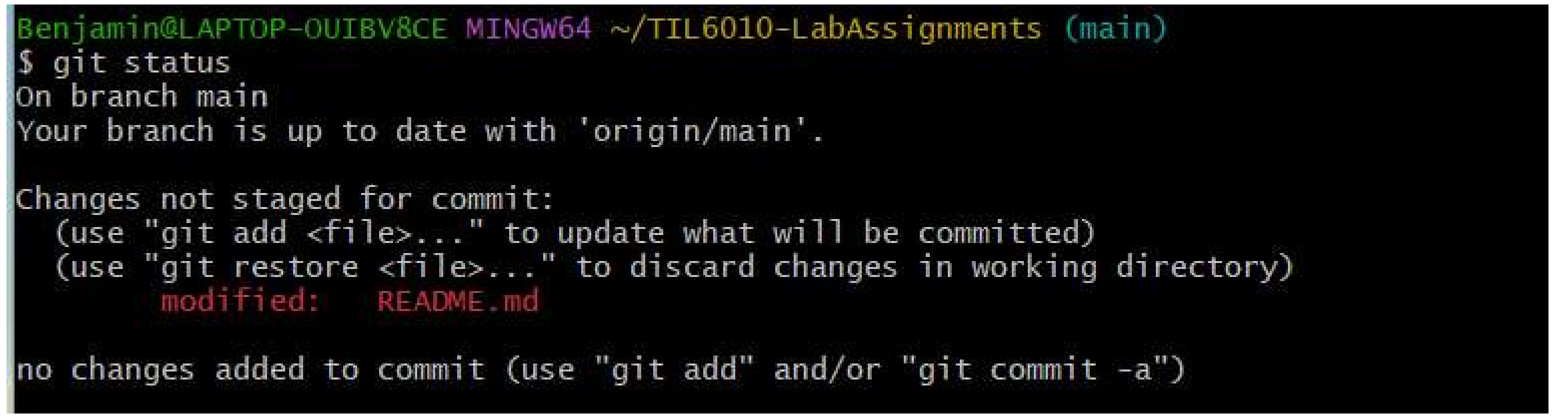
Your first commit

Now, pick one member of the group,

- Open the README.md file, add some text (whatever you like), save and close the file.
- Create a folder named `Lab5`. Then create a file `lab5.txt` under this folder.

Question 1: What is the current status of the repository? A screenshot is sufficient enough.

Use `git status`



```
Benjamin@LAPTOP-OUIBV8CE MINGW64 ~/TIL6010-LabAssignments (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

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:   README.md

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

Question 2: Commit the changes you have made to the README.md file. For this, you need 2 steps.

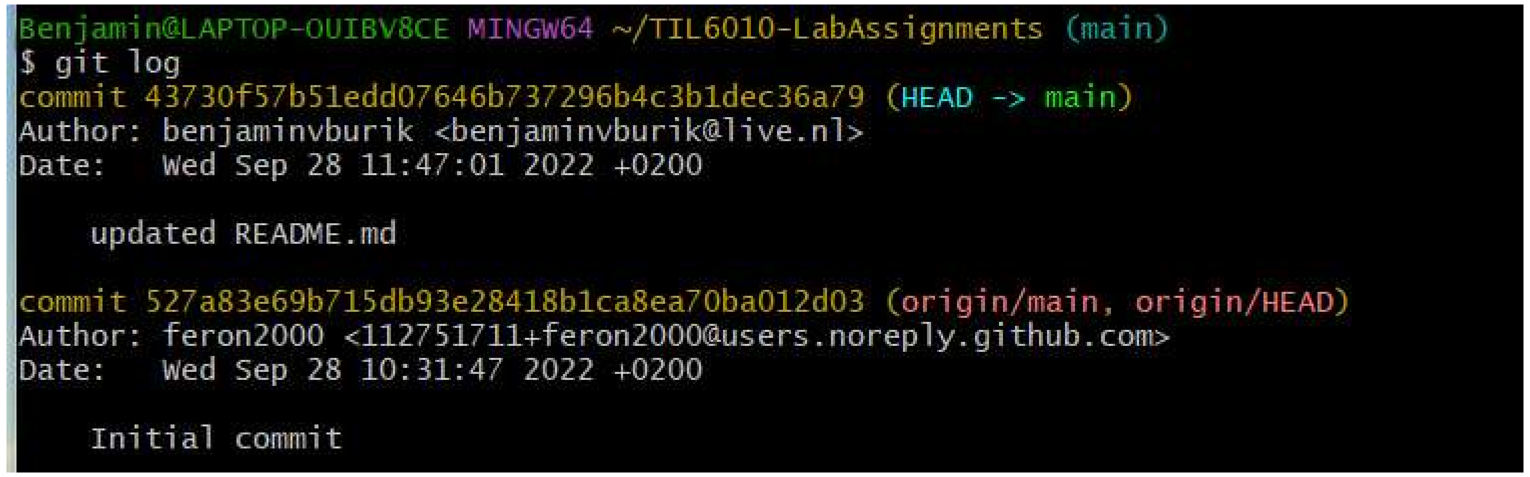
First, stage the file using `git add`. Observe the status of the repository afterwards.

For example, to stage the file README.md: `git add README.md`. If you want to add all files, use `git add .`

Second, commit the changes using `git commit`. Observe the status of the repository afterwards.
Usage as `git commit -m "message"`. A good `message` should describe the changes well.
An example: `git commit -m "updated README.md"`

Question 3: Show the commit tree of the repository. Take a screenshot of the output

Use `git log`



```
Benjamin@LAPTOP-OUIBV8CE MINGW64 ~/TIL6010-LabAssignments (main)
$ git log
commit 43730f57b51edd07646b737296b4c3b1dec36a79 (HEAD -> main)
Author: benjaminvburik <benjaminvburik@live.nl>
Date:   Wed Sep 28 11:47:01 2022 +0200

    updated README.md

commit 527a83e69b715db93e28418b1ca8ea70ba012d03 (origin/main, origin/HEAD)
Author: feron2000 <112751711+feron2000@users.noreply.github.com>
Date:   Wed Sep 28 10:31:47 2022 +0200

    Initial commit
```

If you have not commit `lab5.txt` , please do so with similar steps.

Update local changes to remote repository

Question 4: At the moment, do the local repository and the remote repository have the same content? (i.e. are the two README.md files the same?)

Before syncing any local changes, it's good practice to check if there are changes made to the remote repository. (We know that there has not been any for our case).
Run `git pull` to pull changes from the remote repository.
Then run `git push` to push local changes to the remote repository.

Authentication error

When running `git push` for the first time, you might be asked to enter username and password of your Github account.
In the newest policy of Github, account password cannot be used here. One alternative option is to create a Personal Access Token.
You can follow the instruction (https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/creating-a-personal-access-token) from Github official document to generate one token.
Afterwards, you should be able to push your code.

Now, other member of your groups can pull the update from the remote repository.

PART 2 - WORK INDIVIDUALLY

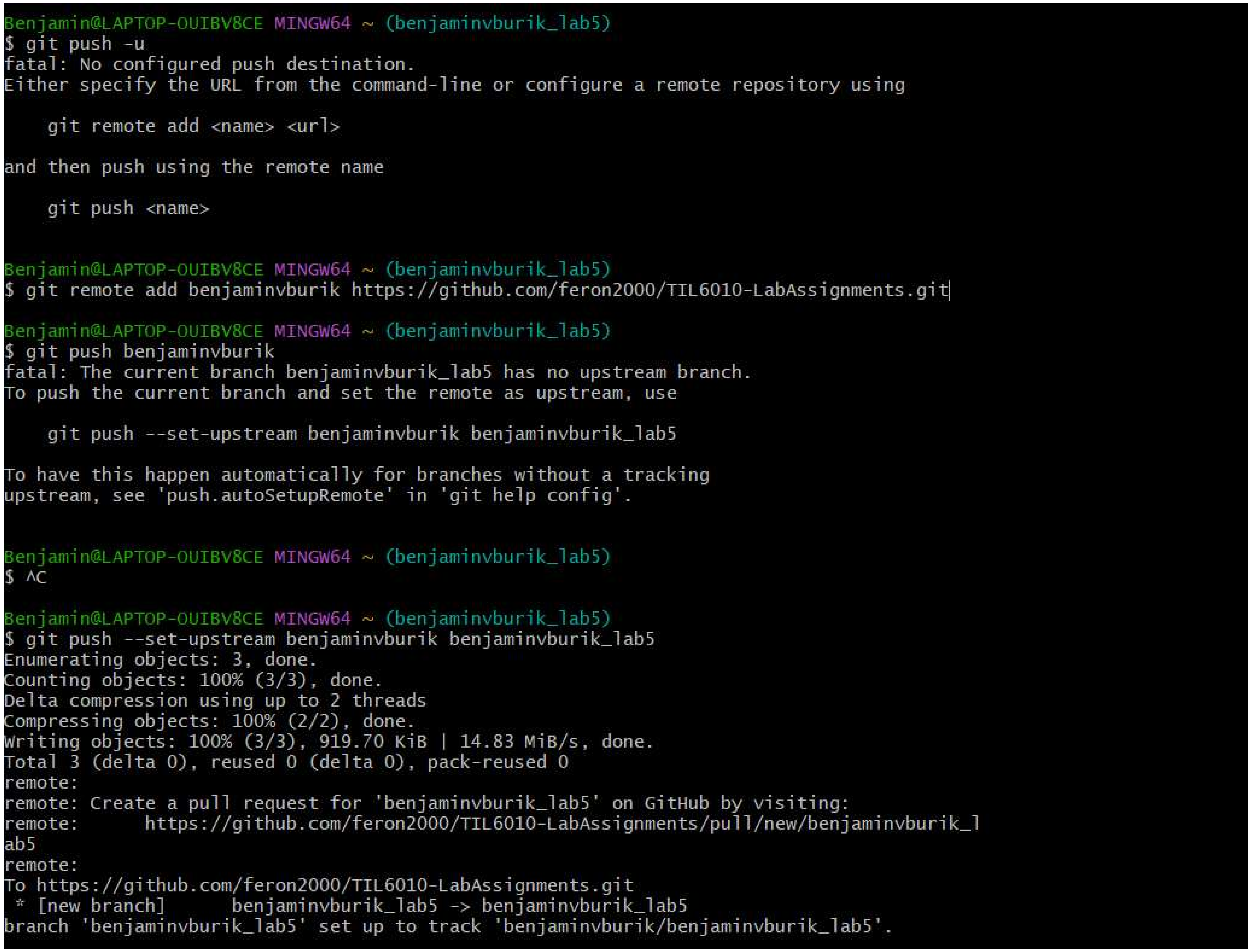
Branch

For each member, create a new branch `yourname_lab5` . Change `yourname` according to your name.
Usage `git checkout -b branchname` . Option `-b` will create a branch if it does not exist.
Example: `git checkout -b Peter_lab5`

Run git status to see if you are in the new branch

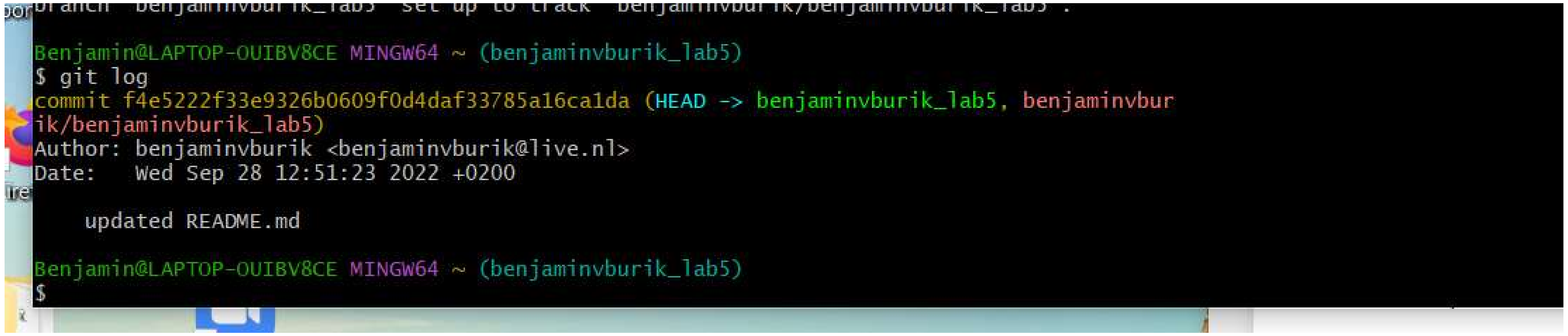
Question 5: Add and commit your work on Lab assignment 5 to the repository

- Create a folder under the `Lab5` folder. Name it as your student id.
- Put your Lab5 solutions (including all .ipynb, .html files) under the folder.
- Commit the changes



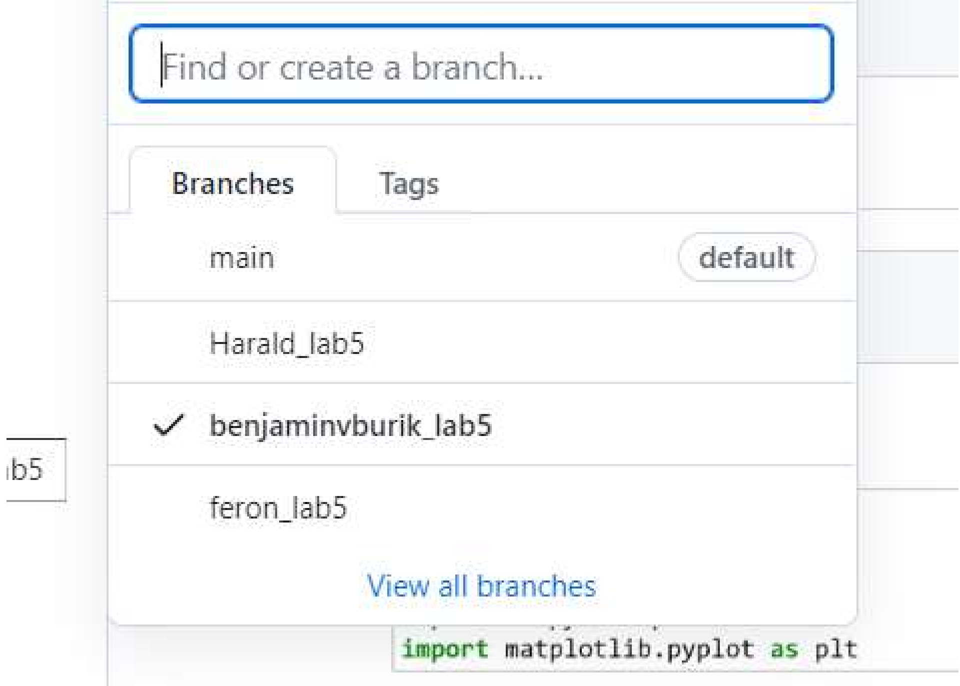
```
Benjamin@LAPTOP-OUIBV8CE MINGW64 ~ (benjaminvburik_lab5)
$ git push -u
fatal: No configured push destination.
Either specify the URL from the command-line or configure a remote repository using

    git remote add <name> <url>

and then push using the remote name

    git push <name>


Benjamin@LAPTOP-OUIBV8CE MINGW64 ~ (benjaminvburik_lab5)
$ git remote add benjaminvburik https://github.com/feron2000/TIL6010-LabAssignments.git

Benjamin@LAPTOP-OUIBV8CE MINGW64 ~ (benjaminvburik_lab5)
$ git push benjaminvburik
fatal: The current branch benjaminvburik_lab5 has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream benjaminvburik benjaminvburik_lab5

To have this happen automatically for branches without a tracking
upstream, see 'push.autoSetupRemote' in 'git help config'.


Benjamin@LAPTOP-OUIBV8CE MINGW64 ~ (benjaminvburik_lab5)
$ ^C

Benjamin@LAPTOP-OUIBV8CE MINGW64 ~ (benjaminvburik_lab5)
$ git push --set-upstream benjaminvburik benjaminvburik_lab5
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 919.70 KiB | 14.83 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'benjaminvburik_lab5' on GitHub by visiting:
remote:      https://github.com/feron2000/TIL6010-LabAssignments/pull/new/benjaminvburik_l
ab5
remote:
To https://github.com/feron2000/TIL6010-LabAssignments.git
 * [new branch]      benjaminvburik_lab5 -> benjaminvburik_lab5
branch 'benjaminvburik_lab5' set up to track 'benjaminvburik/benjaminvburik_lab5'.
```

Run git log and observe the git tree



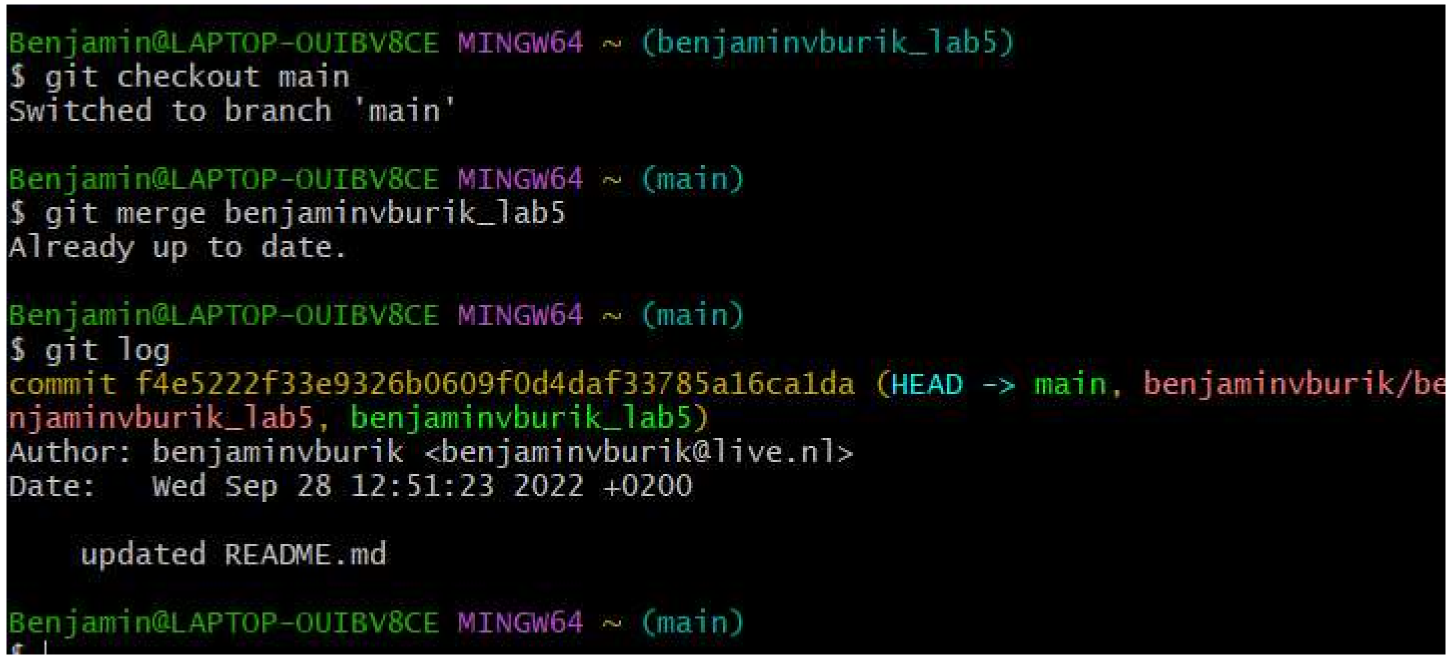
```
branch 'benjaminvburik_lab5' set up to track 'benjaminvburik/benjaminvburik_lab5'.

Benjamin@LAPTOP-OUIBV8CE MINGW64 ~ (benjaminvburik_lab5)
$ git log
commit f4e5222f33e9326b0609f0d4daf33785a16ca1da (HEAD -> benjaminvburik_lab5, benjaminvbur
ik/benjaminvburik_lab5)
Author: benjaminvburik <benjaminvburik@live.nl>
Date:   Wed Sep 28 12:51:23 2022 +0200

    updated README.md

Benjamin@LAPTOP-OUIBV8CE MINGW64 ~ (benjaminvburik_lab5)
$
```

Question 6: Push your branch to the remote repository



Question 7: Merge your branch with the main branch.

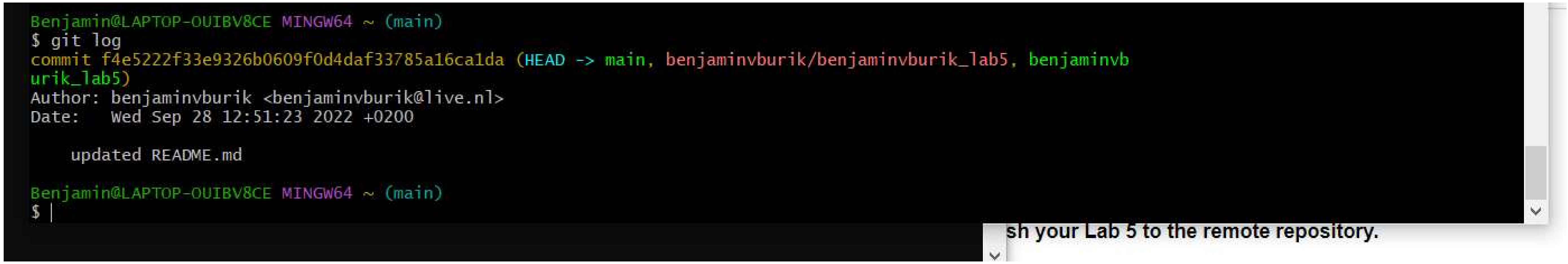
- Checkout to the `main` branch
- Merge your lab5 branch with the main branch
- Run git log to observe changes in the git tree



```
Benjamin@LAPTOP-OUIBV8CE MINGW64 ~ (benjaminvburik_lab5)
$ git checkout main
Switched to branch 'main'

Benjamin@LAPTOP-OUIBV8CE MINGW64 ~ (main)
$ git merge benjaminvburik_lab5
Already up to date.

Benjamin@LAPTOP-OUIBV8CE MINGW64 ~ (main)
$ git log
commit f4e5222f33e9326b0609f0d4daf33785a16ca1da (HEAD -> main, benjaminvburik/be
njaminvburik_lab5, benjaminvburik_lab5)
Author: benjaminvburik <benjaminvburik@live.nl>
Date:   Wed Sep 28 12:51:23 2022 +0200

    updated README.md

Benjamin@LAPTOP-OUIBV8CE MINGW64 ~ (main)
```

Question 8: Push your Lab 5 to the remote repository.

Final question: Run git log, take a screenshot, and put it below this cell



```
Benjamin@LAPTOP-OUIBV8CE MINGW64 ~ (main)
$ git log
commit f4e5222f33e9326b0609f0d4daf33785a16ca1da (HEAD -> main, benjaminvburik/benjaminvburik_lab5, benjaminvb
urik_lab5)
Author: benjaminvburik <benjaminvburik@live.nl>
Date:   Wed Sep 28 12:51:23 2022 +0200

    updated README.md

Benjamin@LAPTOP-OUIBV8CE MINGW64 ~ (main)
$ |
```

sh your Lab 5 to the remote repository.

Try to recreate similar repository for your project