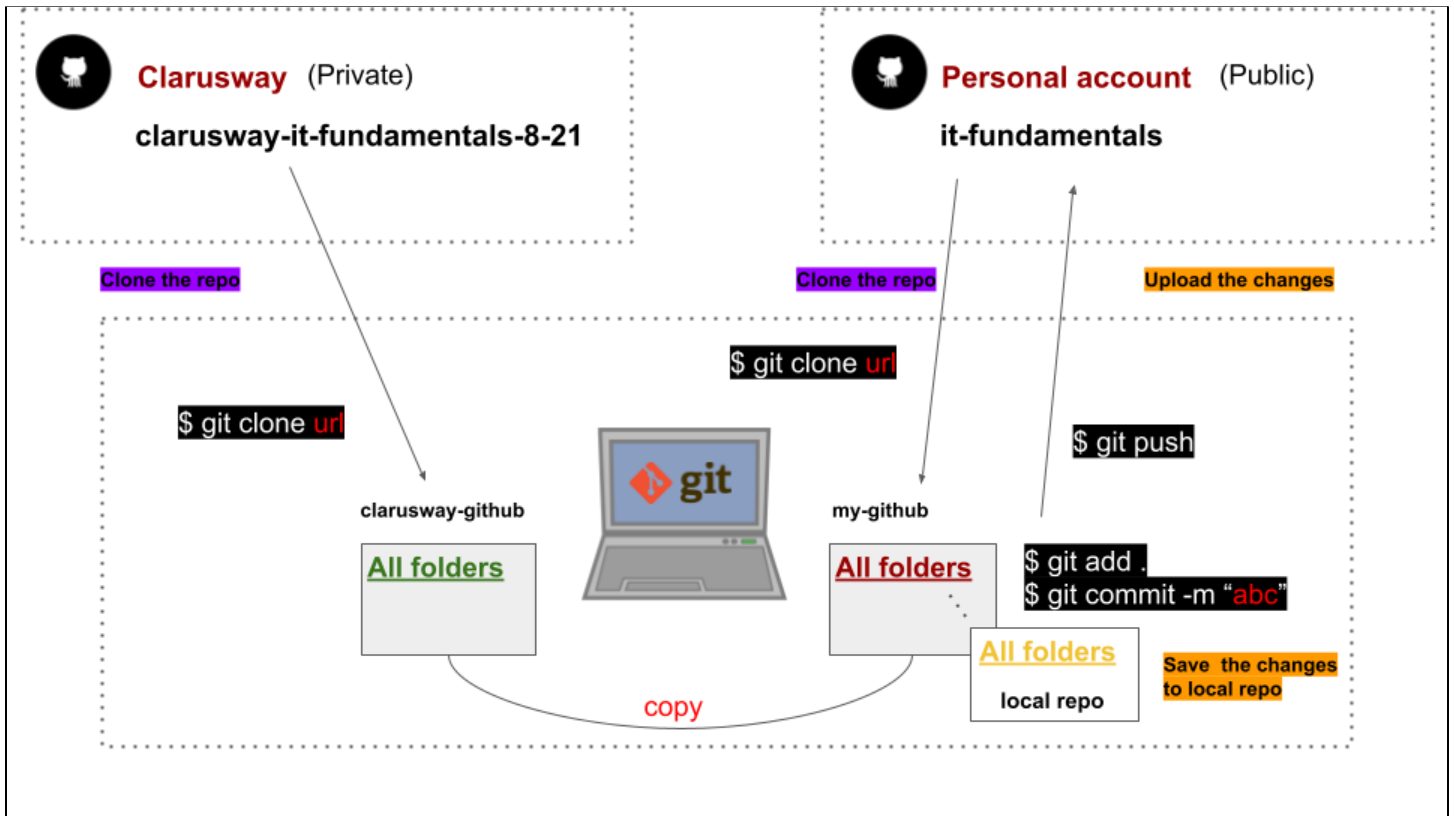


GitHub - Lab



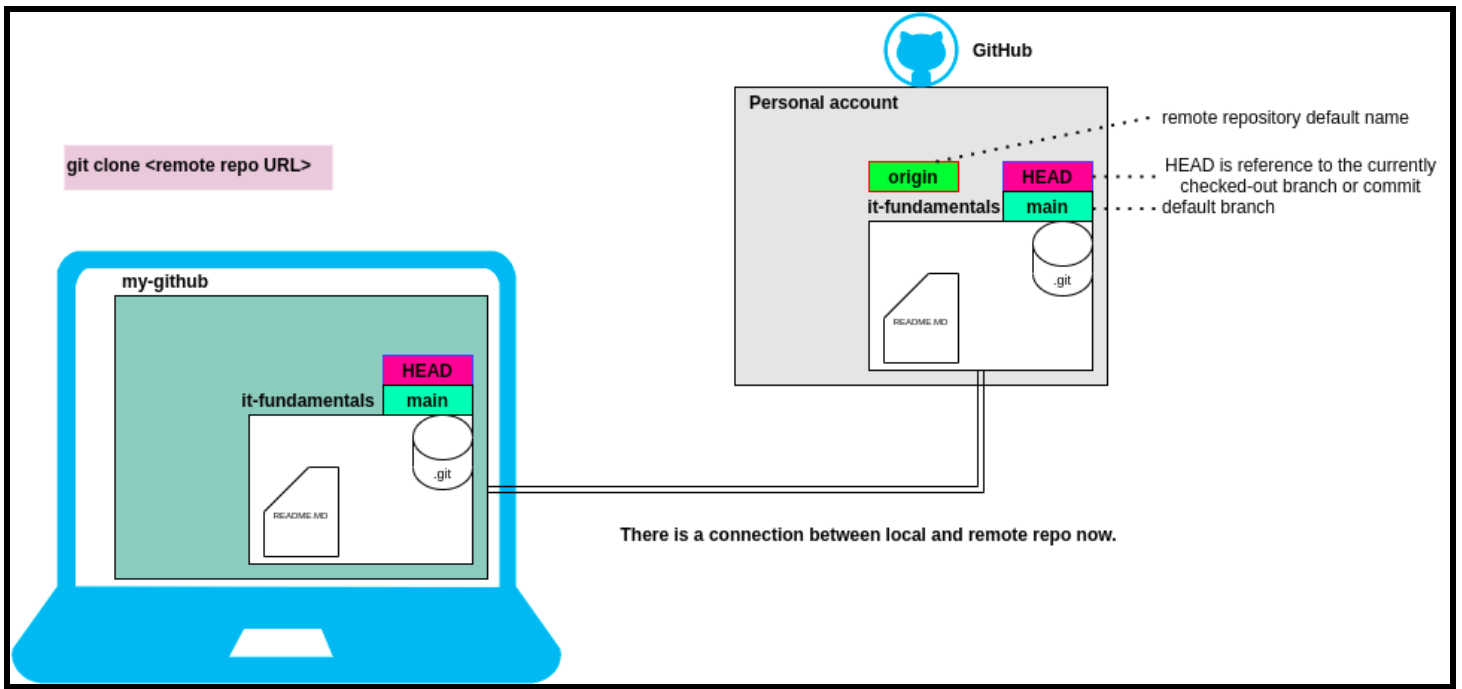
Part-1

1. Create a repository on your Github account named **it-fundamentals**.
2. Create a folder on your computer named **my-github**.

```
mkdir my-github
```

3. Clone your **it-fundamentals** remote repo in **my-github** folder.

```
cd my-github  
git clone https://github.com/xxxxxxxx/xxxxxx
```



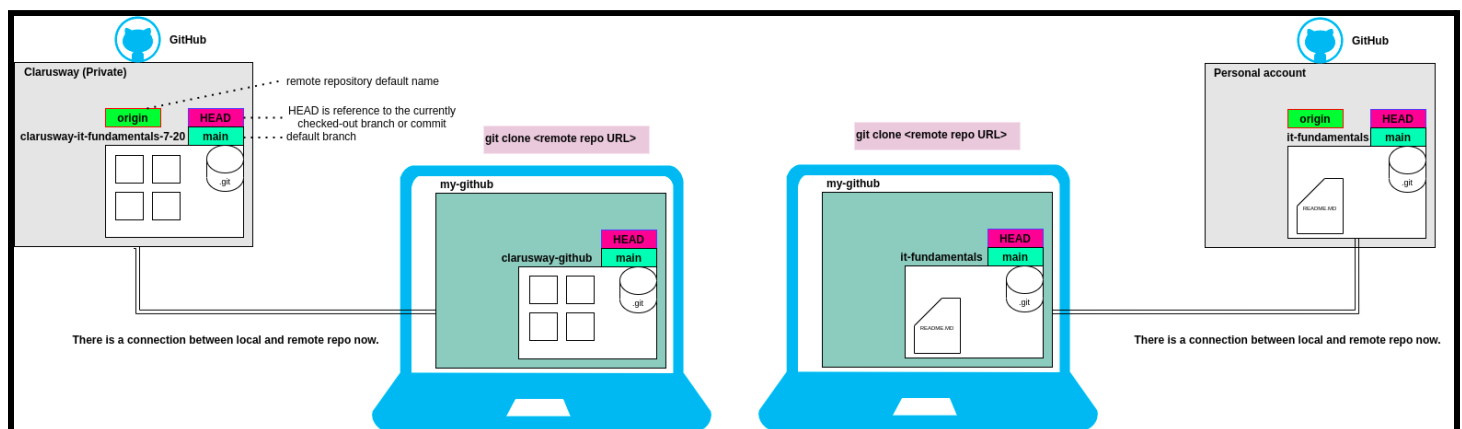
4. Create another folder on your computer named **clarusway-github**. 

```
mkdir clarusway-github
```

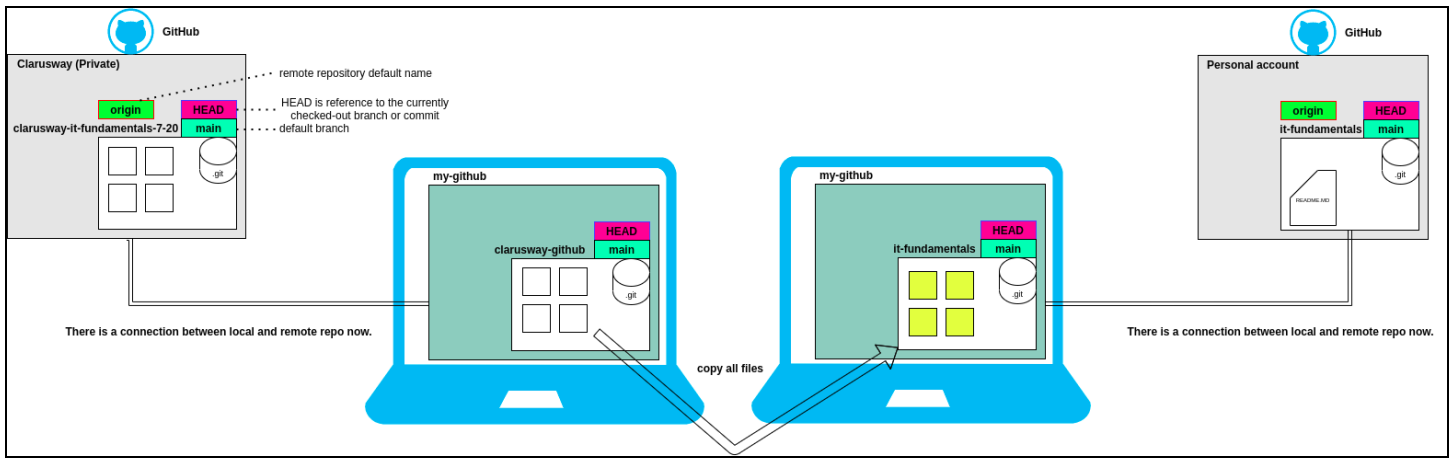
5. Clone the **clarusway-it-fundamentals-8-21** repo in **clarusway-github** folder.

```
cd clarusway-github
```

```
git clone https://github.com/clarusway/clarusway-it-fundamentals-8-21
```

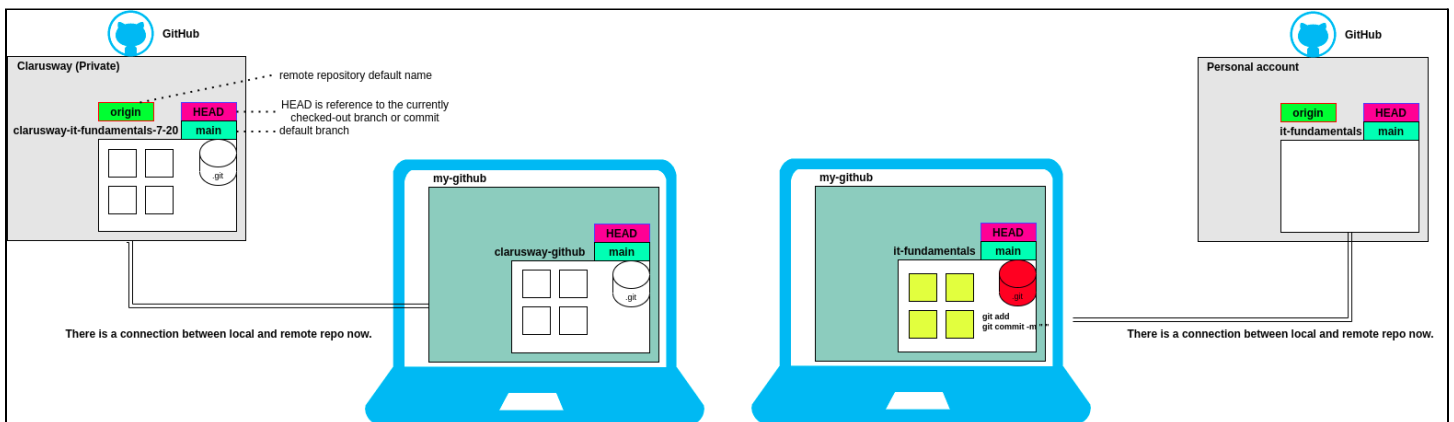


6. Copy all the files and folders in the **clarusway-github** folder to **my-github** folder.



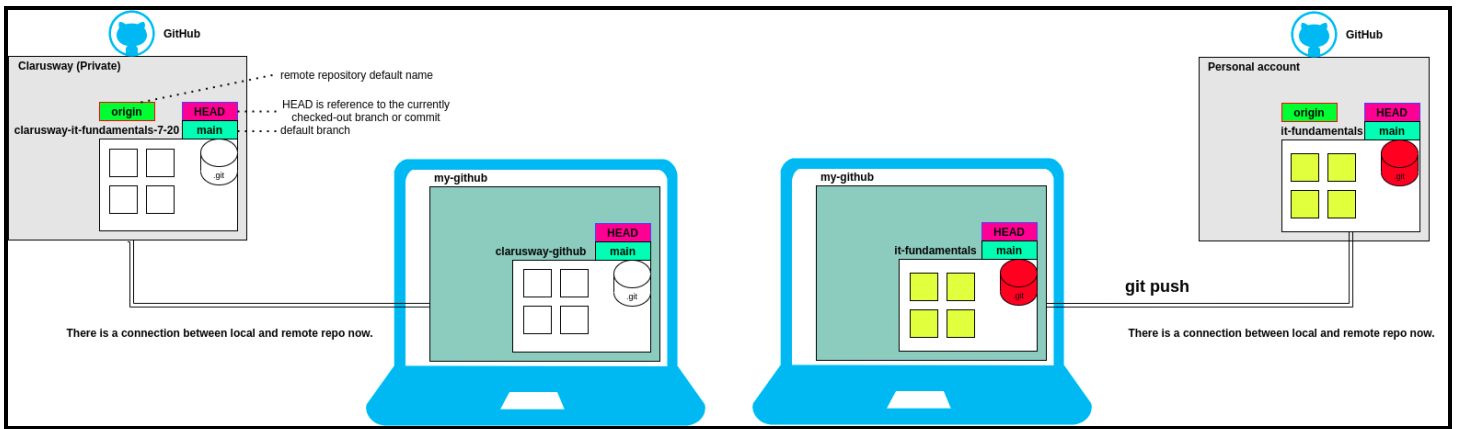
7. Add the changes to your local repo.

```
(cd my-github)
git add .
git commit -m "copied all the files"
```



8. Then send them to your remote repo.

```
git push
```



Part-2

9. Download the changes from **clarusway-it-fundamentals-8-21** (see them in the working directory)

```
git pull
```

10. Copy the **lab1.txt** file from **clarusway-github** folder to **my-github/it-fundamentals/git/lab** folder.

11. In the **my-github/it-fundamentals** directories:

Check the status of the project. Add **lab1.txt** file to **index area**. And see the status again.

```
git status
```

```
git add lab1.txt
```

```
git status
```

12. Save the changes to the local repo.

```
git commit -m "for lab"
```

13. See commit history.

```
git log --oneline
```

14. Upload the changes to your remote repository.

```
git push
```

Part-3

15. Go to “**my-github/it-fundamentals/git/lab**” directory:

- Create a new branch named **front-end**

```
git branch front-end
```

- See branches

```
git branch
```

 (show local branches)

```
git branch -r
```

 (show remote branches)

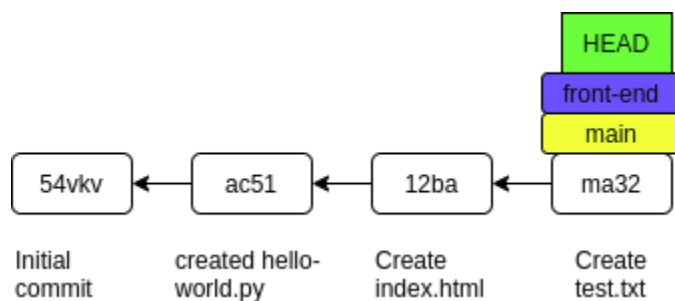
```
git branch -a
```

 (show all local and remote branches)

```
front-end
* main
remotes/origin/HEAD -> origin/main
remotes/origin/main
```

- Switch to **front-end** branch

```
git checkout front-end
```



- List the files and check the status of the working directory

```
ls
```

```
git status
```

- Make some changes in the **lab1.txt** file, and check the status

```
vim lab1.txt
```

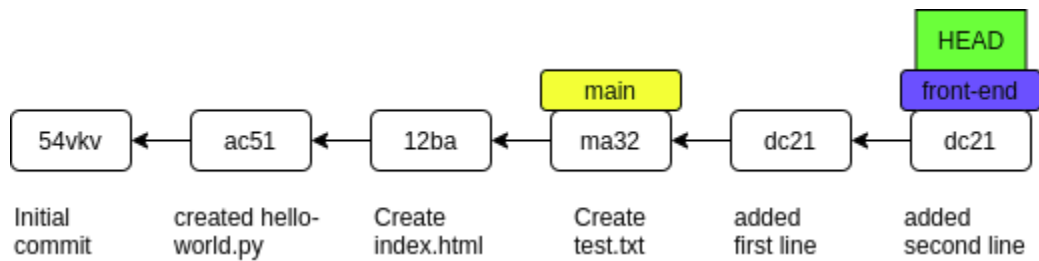
```
git status
```

- Store the changes to the repo and check the status

```
git commit -am "added first line"
git status
```

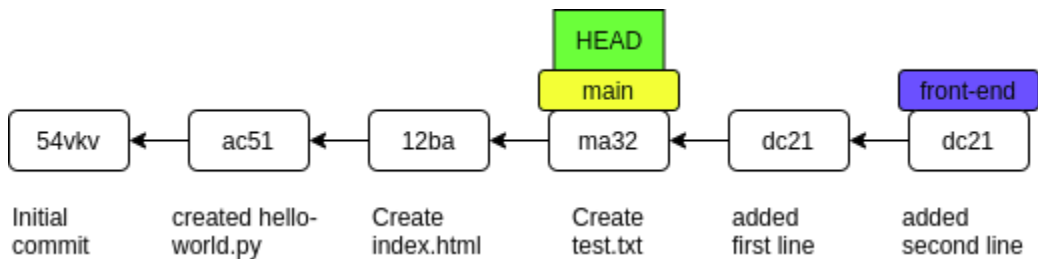
- Add another line to **lab1.txt** and store it to the local repo.

```
vim lab1.txt
git commit -am "added second line"
```



- Switch the main branch and see the content of the **test.txt**

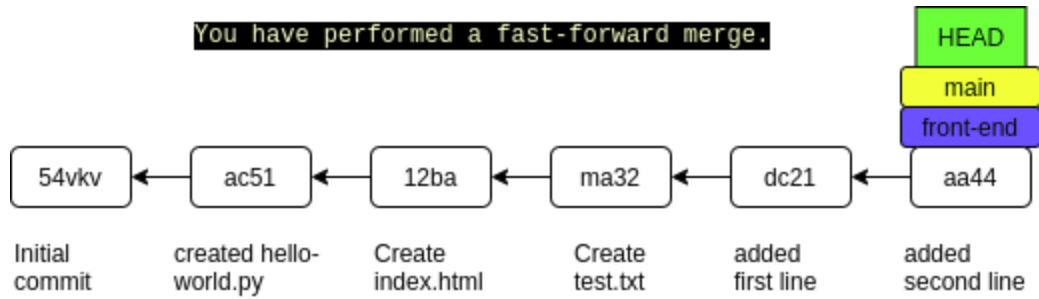
```
git checkout main
cat lab1.txt
```



- Merge front-end branch to **main** branch.

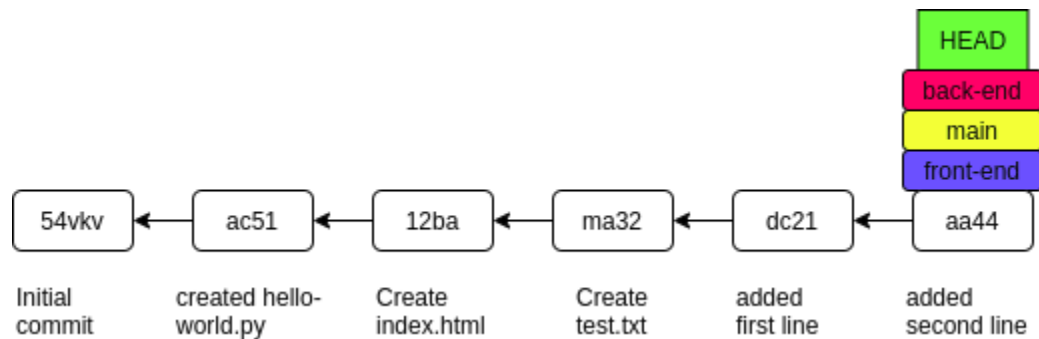
```
git merge front-end
cat lab1.txt
```

You have performed a fast-forward merge.



- Create a new branch named **back-end** and switch to it

```
git checkout -b back-end
```

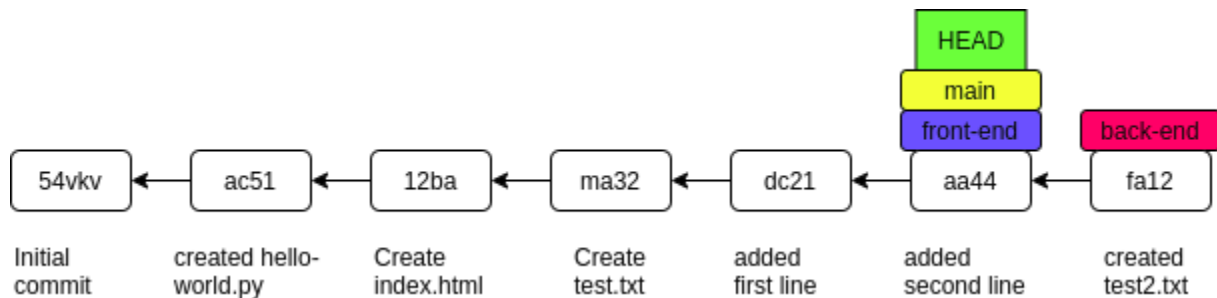


- Create a new file named **test2.txt** and store the changes to repo.

```
touch test2.txt  
git add .  
git commit -m "created text2.txt"
```

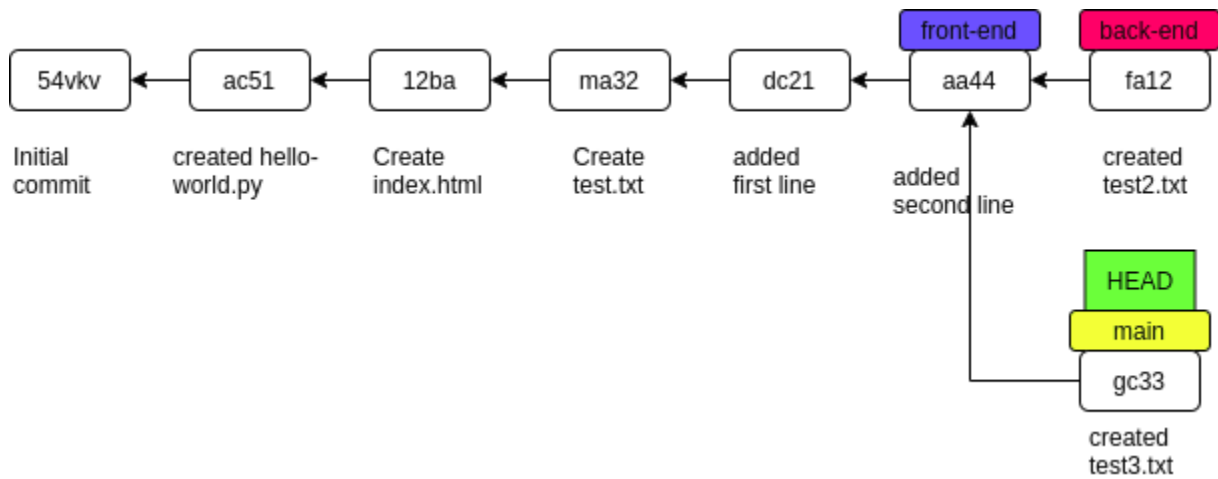
- Switch the main branch again

```
git checkout main
```



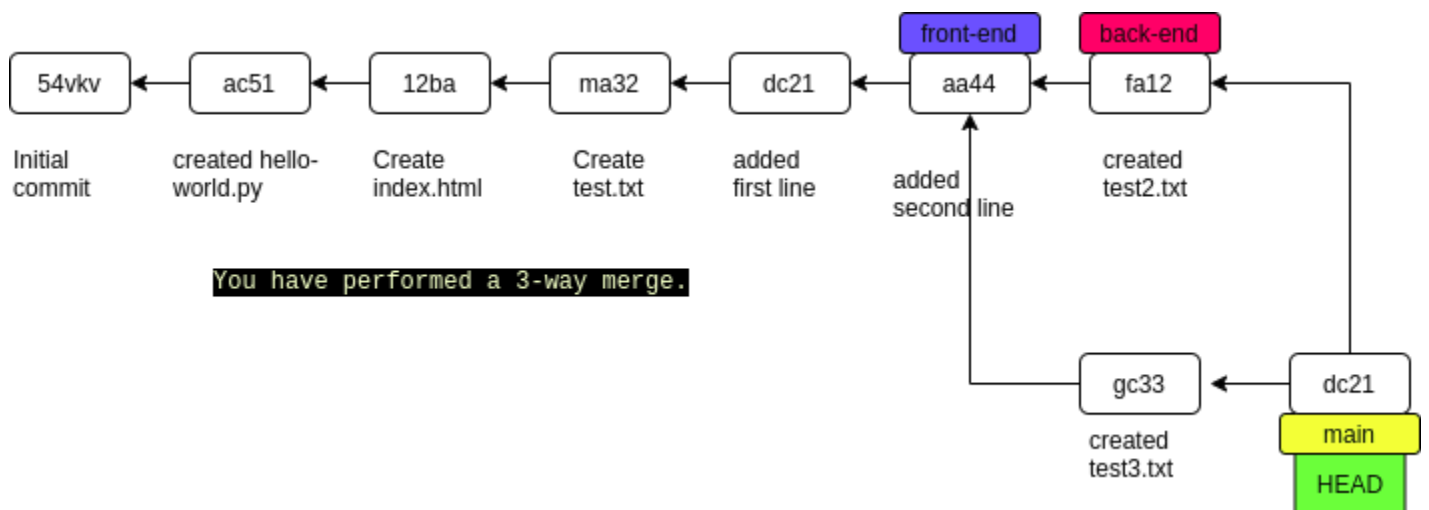
- Create a new file **test3.txt** and send the changes to local repo.

```
touch test3.txt
git add .
git commit -m "created text3.txt"
```



- Merge **main** branch with **back-end** branch

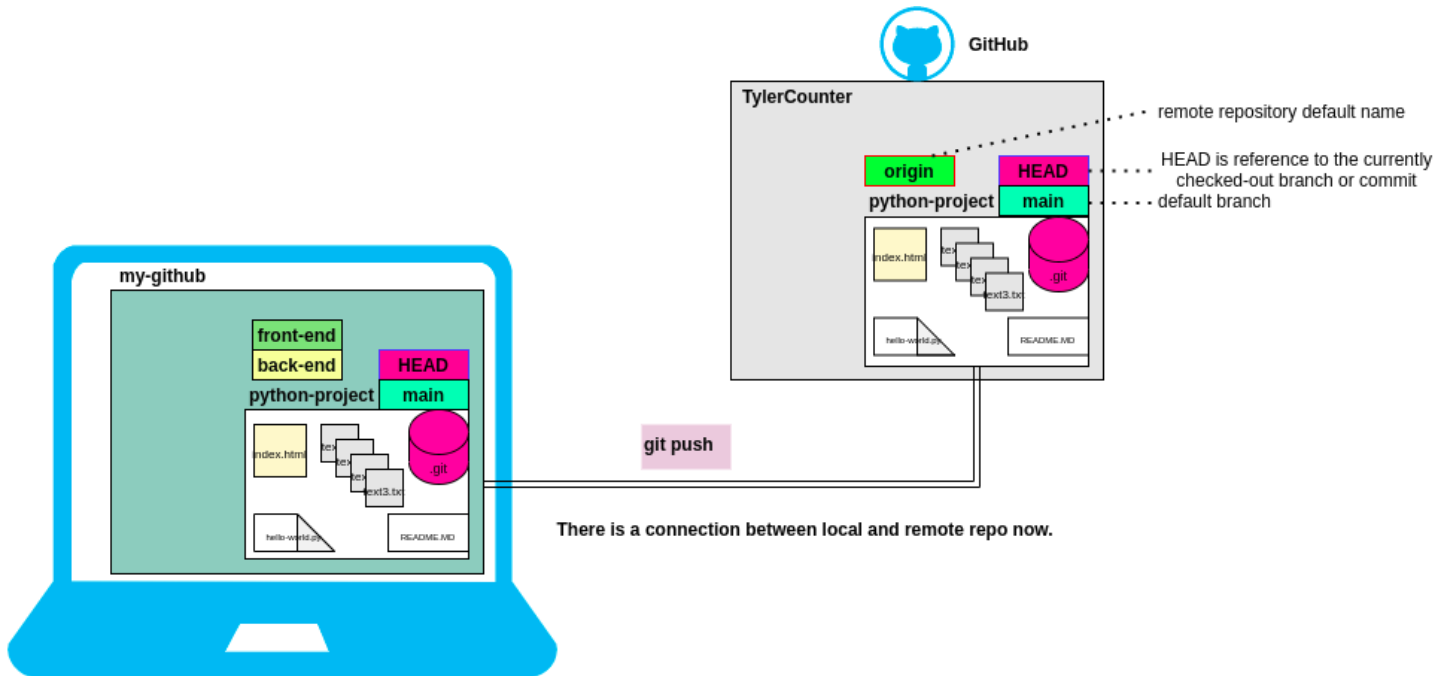
```
git merge back-end
```



Part 4:

- Send the changes to the remote repository

```
git push
```



- Go and check the remote repository

Part 5:

- Go to the terminal and delete the branches named **front-end** and **back-end**

```
git branch -d front-end  
git branch -D back-end
```

- List the all branches

```
git branch -a
```

