

# TUTORIAL 1 : SETTING UP THE REPOSITORY

---

This tutorial will take you through setting up the design repository for your project work. Design repository (or repo) is a part of any software or hardware design company which keeps the records of different versions of your design. These versions can be accessed later when needed.

## 1 CONTENTS

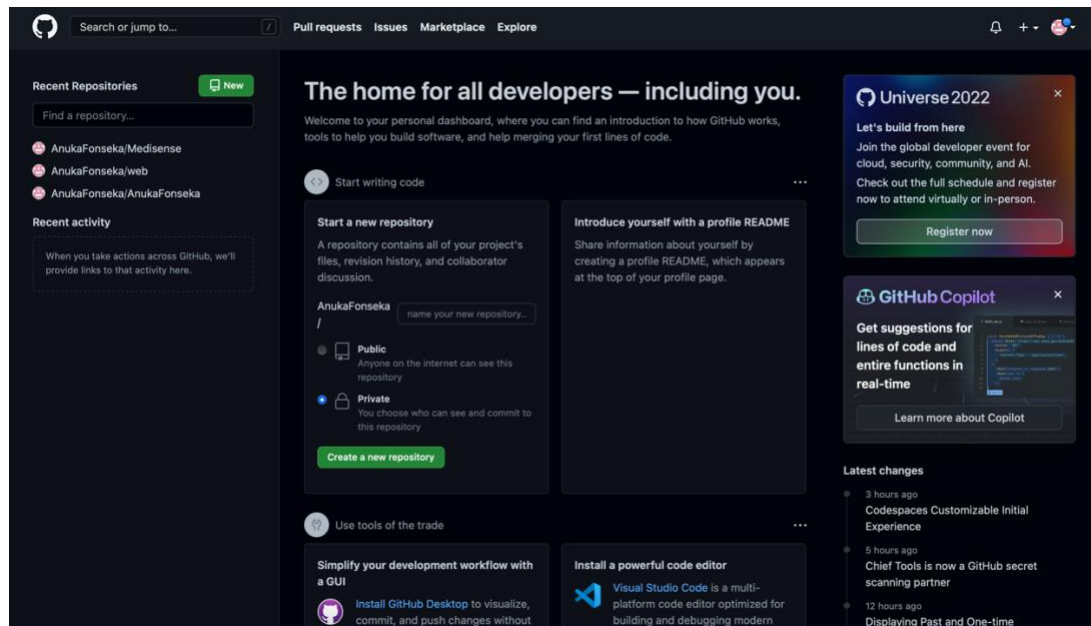
---

TUTORIAL 1 : SETTING UP THE REPOSITORY .....	1
1.1 familiarizing with Github .....	1
1.2 Github GUI .....	8
1.3 Collaboration .....	11
1.4 Tasks.....	14

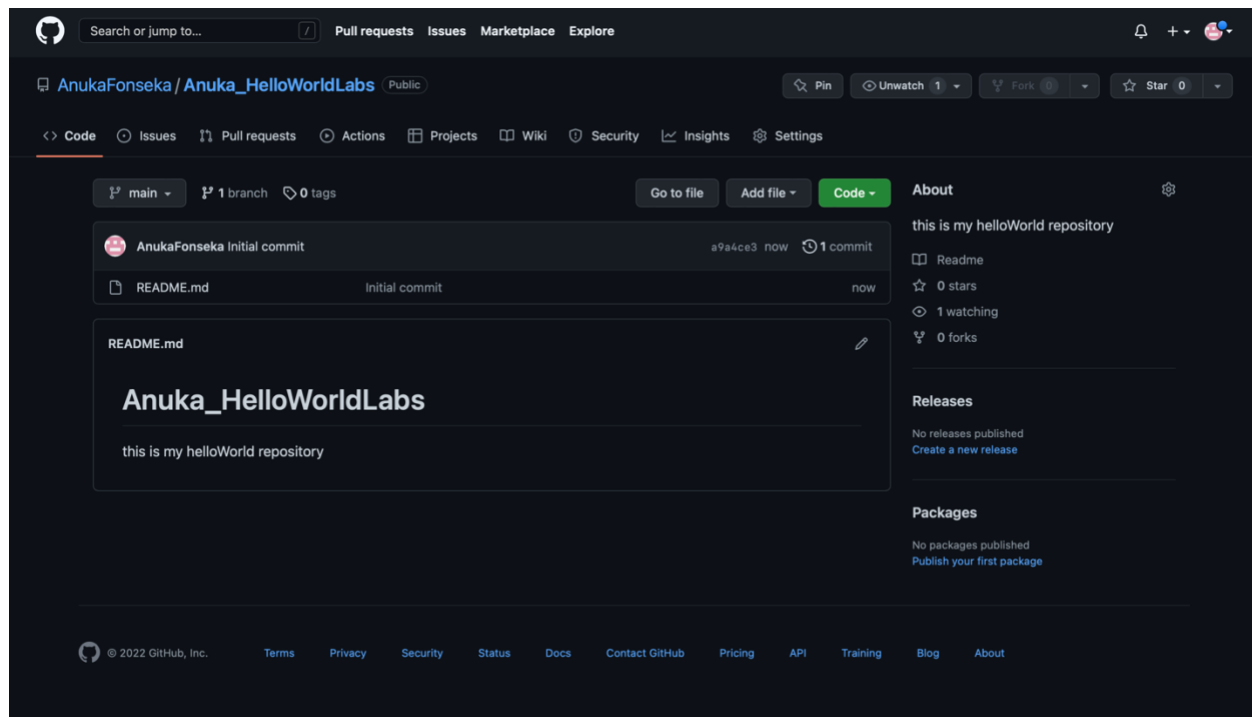
### 1.1 FAMILIARIZING WITH GITHUB

Github.com is an online hosting platform which can be used free of charge (maximum three users for the basic version). It lets you and your team work collaboratively from any where in the world. This part of the tutorial is based on <https://guides.github.com/activities/hello-world/> . Please go through the complete tutorial, if needed better explanations in some parts and for step by step guidelines.

- Step 1 : Create your github account
  - Visit [www.github.com](https://www.github.com)
  - Create your account with a good username as you may need to use it in your future work



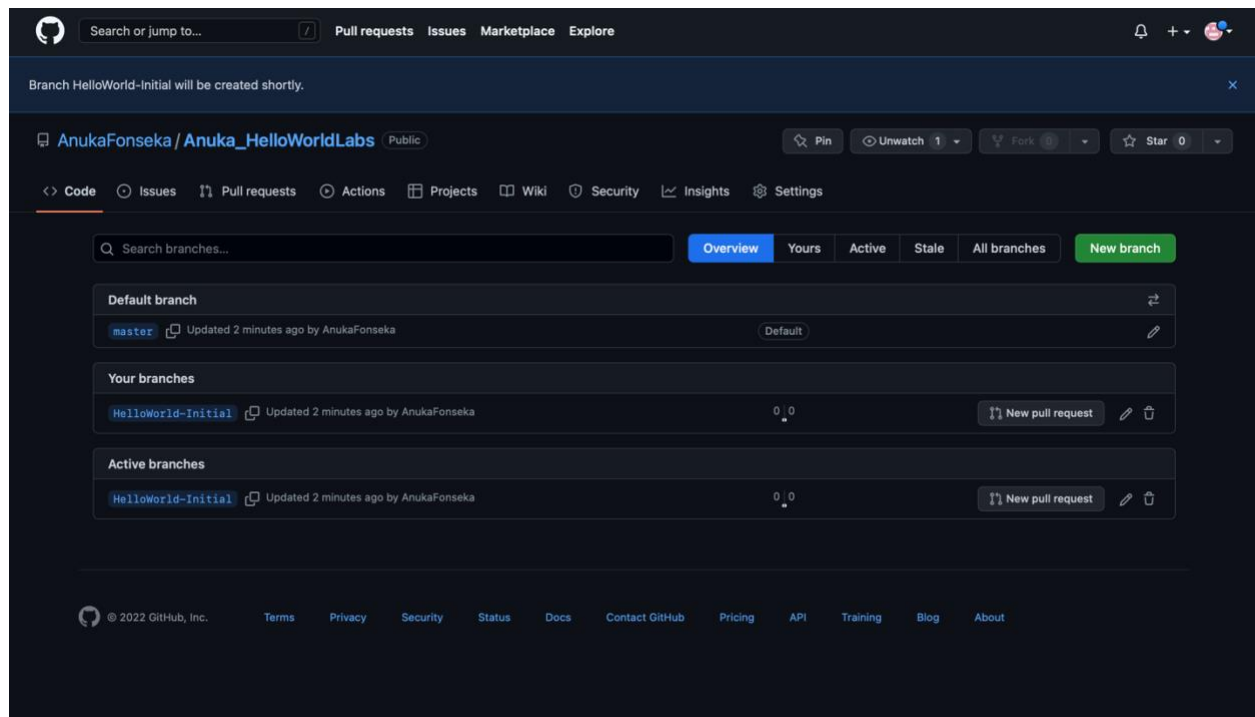
- Step 2 : Create a repository
  - A repository is used to host a single project which can contain any type of data including folders.
  - Login to your github.com page
  - By selecting '+' and New repository
  - Name of the repository "YourFirstName\_HelloWorldLabs", Use a meaningful name with no spaces
  - Add a short description. Try to describe the project in short
  - Select "Initialize this repository with a README"
  - Click create repository



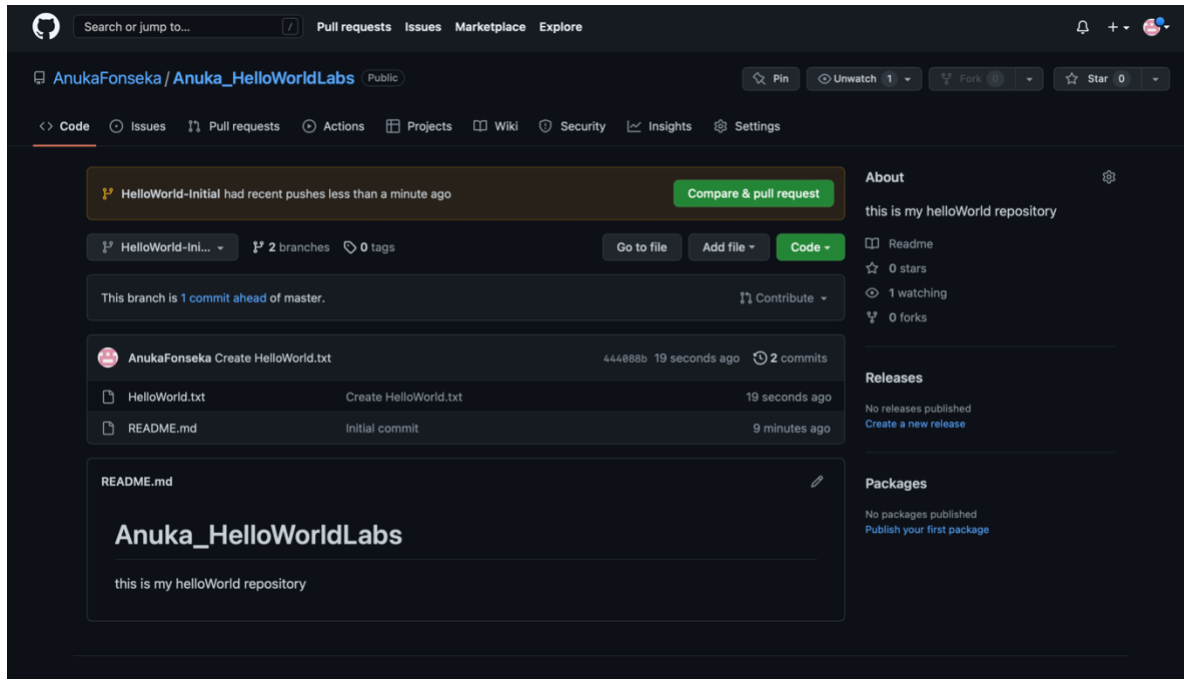
- Step 3: Branching
  - This is the way to work on different versions of a project.
  - The master branch is considered to be the working project which can be branched, edited and tested before committing back to the master branch. In this way master branch contains a working set of design files all the time. Your branch is independent from someone else's. Therefore it is possible to work simultaneously in developing different features of the same project.
  - Create a new branch named "HelloWorld-Initial"

*Figure 1: Repository home page*

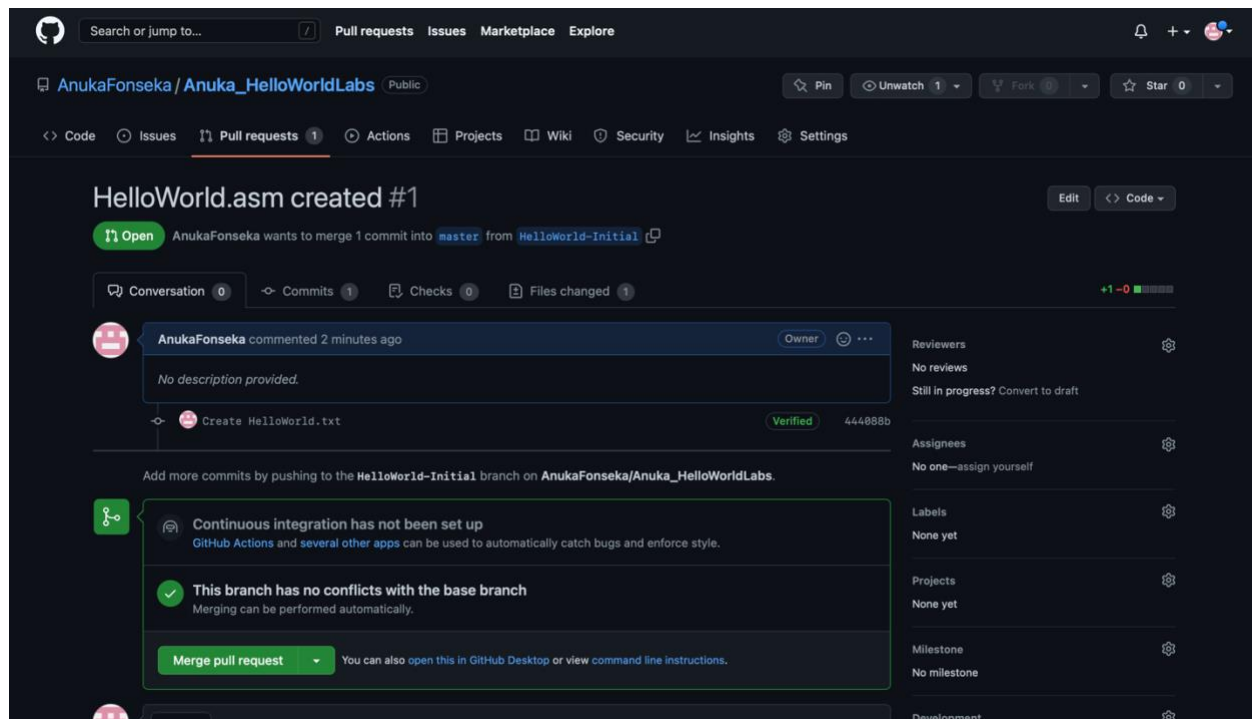
- Now we have two branches named "master " and "HelloWorld-Initial"




- Step 4: Commit changes
  - Now you can start editing the files in your branch
  - Add a new file named "HelloWorld.txt"
  - Write some text in the "HelloWorld.txt"
  - Then click "Commit changes"



- Step 5: Open a pull request
  - Using pull requests, you can ask someone to review your contributions before merge them to the master branch or some other branch.
  - It is possible open a pull request as soon as you make a commit.
  - You can use “mention” system to ask someone to give feedback
  - It is possible to open up pull requests yourself and merge them yourself.
  - Create a pull request and give it the title “HelloWorld.asm created”



- Step 6: Merge the pull request
  - Once you finish testing your branch it can be merged to the master branch
  - It shows if there are any conflicts of the edited files. If two users edit the same file from two locations, there can be conflicts.
  - Click Merge pull request and confirm the merge
  - You can delete the branch as the changes are already incorporated.



[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)


AnukaFonseka / Anuka\_HelloWorldLabs Public

[Pin](#) [Unwatch 1](#) [Fork 0](#) [Star 0](#)

[<> Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

# HelloWorld.asm created #1

Edit [Code](#)

 Merged


AnukaFonseka merged 1 commit into master from HelloWorld-Initial 12 seconds ago

Conversation 0

Commits 1

Checks 0


Files changed 1

 AnukaFonseka commented 3 minutes ago

Owner


...

No description provided.


 Create HelloWorld.txt

Verified

444888b

 AnukaFonseka merged commit eb0f797 into master 12 seconds ago

Revert


 Pull request successfully merged and closed

You're all set—the HelloWorld-Init... branch can be safely deleted.

Delete branch

Write

Preview

H B I 

Leave a comment

Reviewers

No reviews

Assignees

No one—assign yourself

Labels

None yet

Projects


None yet

Milestone

No milestone

Development

Successfully merging this pull request may close



[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)


AnukaFonseka / Anuka\_HelloWorldLabs Public

[Pin](#) [Unwatch 1](#) [Fork 0](#) [Star 0](#)

[<> Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

# HelloWorld.asm created #1

Edit [Code](#)

 Merged


AnukaFonseka merged 1 commit into master from HelloWorld-Initial 1 minute ago

Conversation 0

Commits 1

Checks 0


Files changed 1

 AnukaFonseka commented 4 minutes ago

Owner


...

No description provided.


 Create HelloWorld.txt

Verified


444888b

 AnukaFonseka merged commit eb0f797 into master 1 minute ago

Revert

 AnukaFonseka deleted the HelloWorld-Initial branch 35 seconds ago

Restore branch


 Pull request successfully merged and closed

You're all set—the HelloWorld-Init... branch can be safely deleted.

Delete branch

Write

Preview

H B I 

Leave a comment

Reviewers

No reviews

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

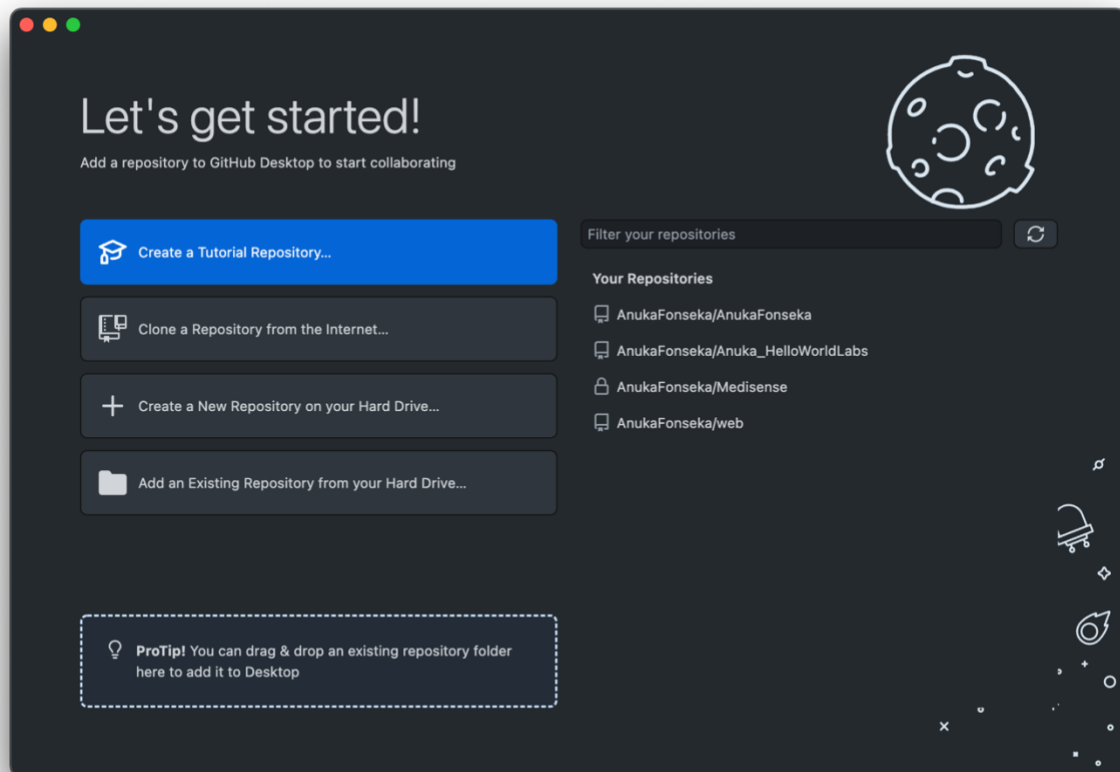
Development

Successfully merging this pull request may close

## 1.2 GITHUB GUI

You can manage your codes locally with the help of command line arguments or with the Github GUI. Following instructions are to setup the Github GUI in your computer.

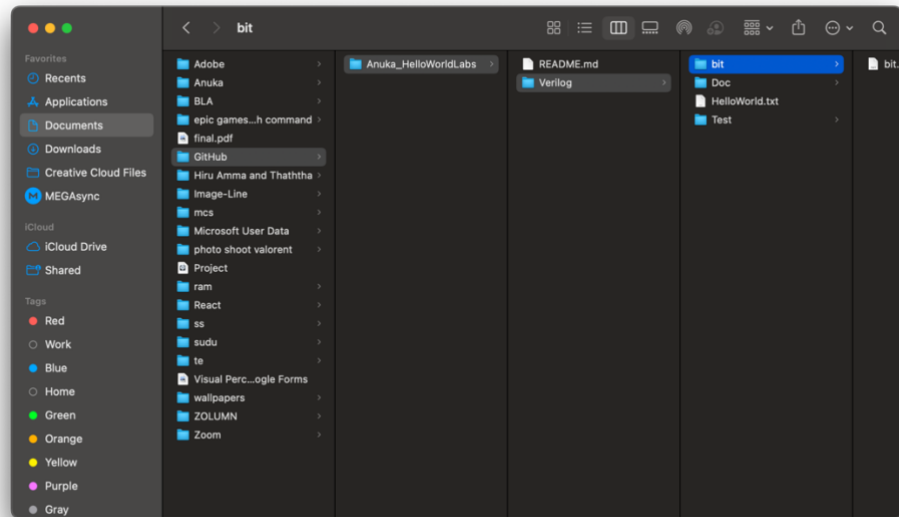
- Step 1 : Installing Github desktop
  - <https://desktop.github.com/> download and install Github desktop from here
  - Login using your Github account login details



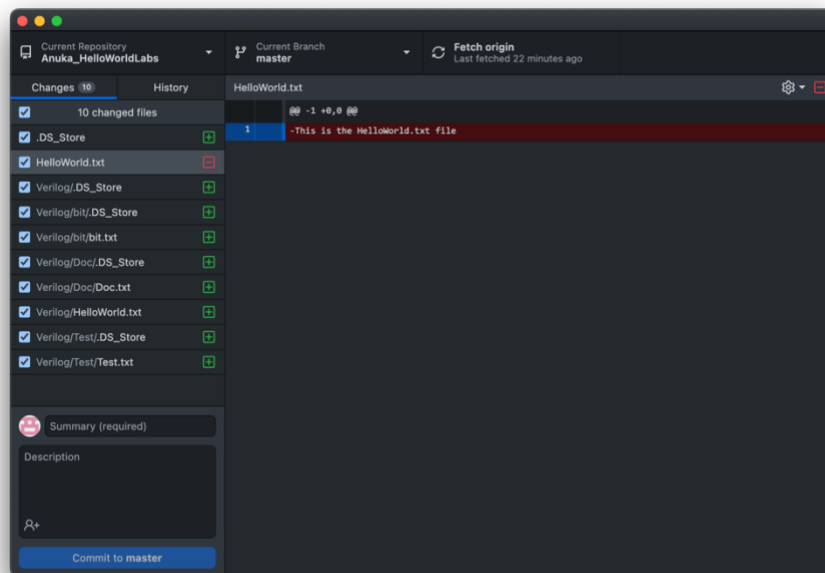
- Step 2 : Managing the repository
  - Clone the repository you created at Github.com to your computer
  - Now view the repository in file explorer
  - Create the following folder structure in your github folder
    - HelloWorld
      - verilog
        - doc
        - test
        - bit
  - Advantage of having a well-defined folder structure your project is that you don't need to look again and again where some specific files are.



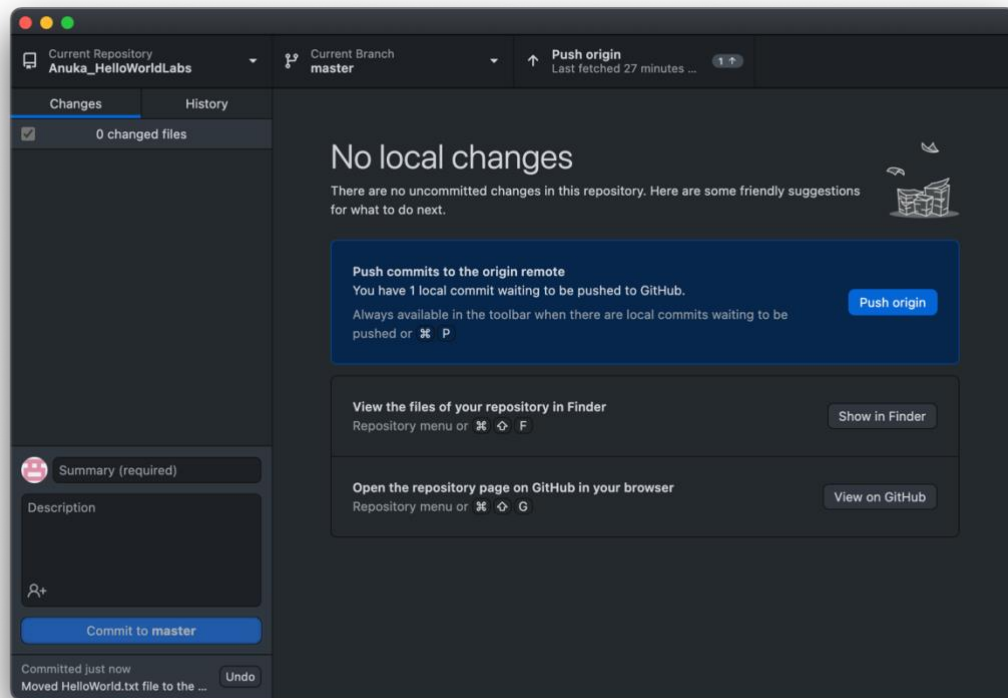
- verilog: contains the main files, in this case HelloWorld.asm
- Doc : Readme files on how to run the program
- Test : A program or software simulation files which used to test your code
- bit : bit file ready to be programmed on the FPGA
- Now drop the HelloWorld.txt file inside the c folder.
- Create some text files inside other folders as well



- Now on the GUI you can see these added files



- It shows that we have moved the HelloWorld.txt file
- Add a “Summary” – Moved HelloWorld.txt file to the C folder
- Now Commit to the Current branch



- Click push origin to upload these changes to your online repository.
- You can make a Pull request which will take you to the online repo from the Github desktop. Create your pull request as given in the first section.

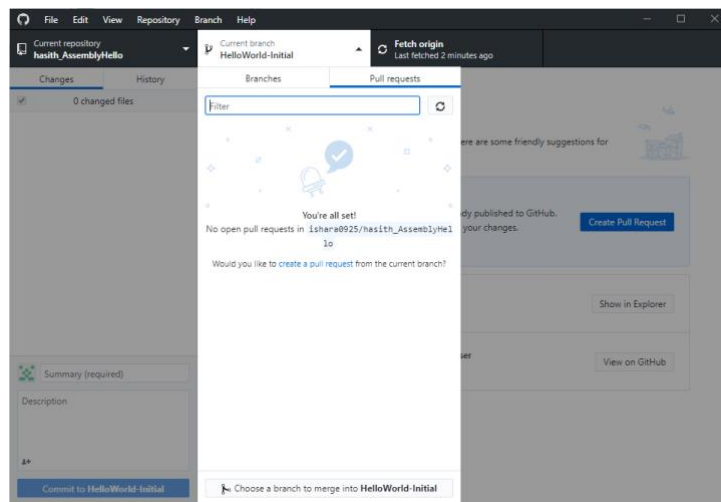


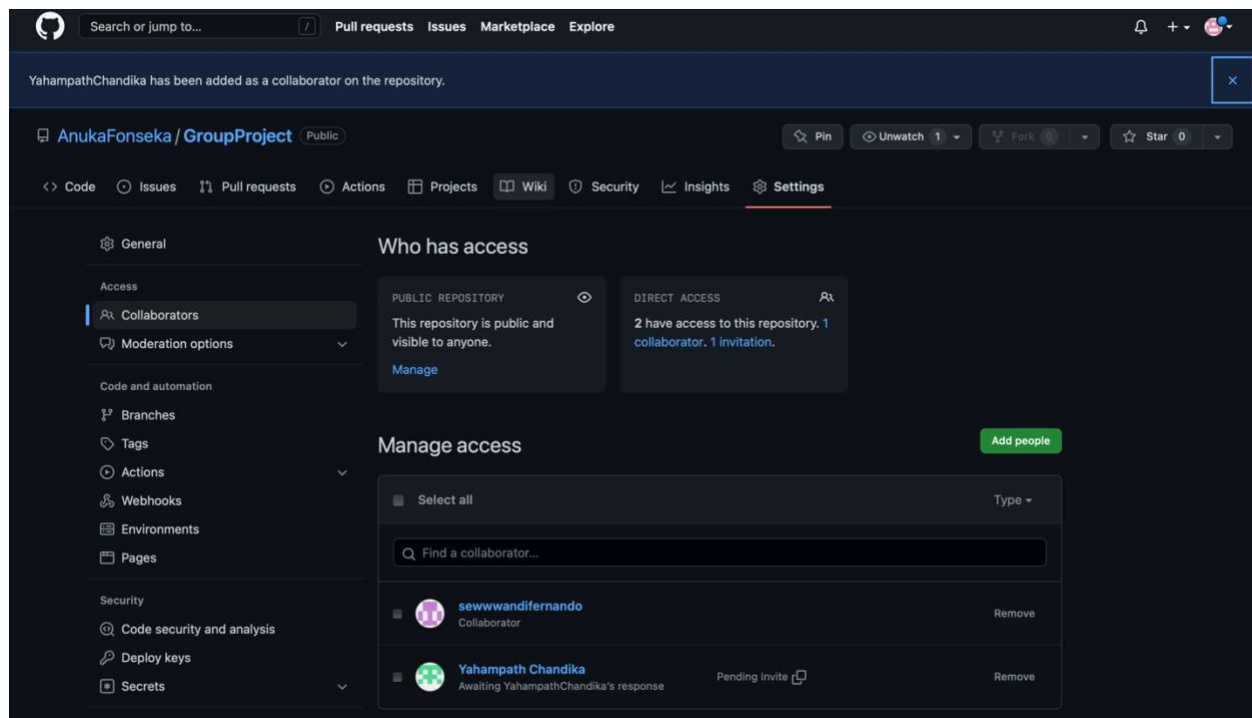
Figure 2 : Pull request

- Step 3 : Familiar yourself with Github desktop
  - <https://help.github.com/en/desktop> Please refer to the help located in the given link to overcome any issue you face.
  - Also try to use the other features available in the GUI. Get familiarize with them.

### 1.3 COLLABORATION

Main advantage of Git repo is the possibility of working simultaneously on different parts of a project. Follow the steps to create your own repo to be shared between team members.

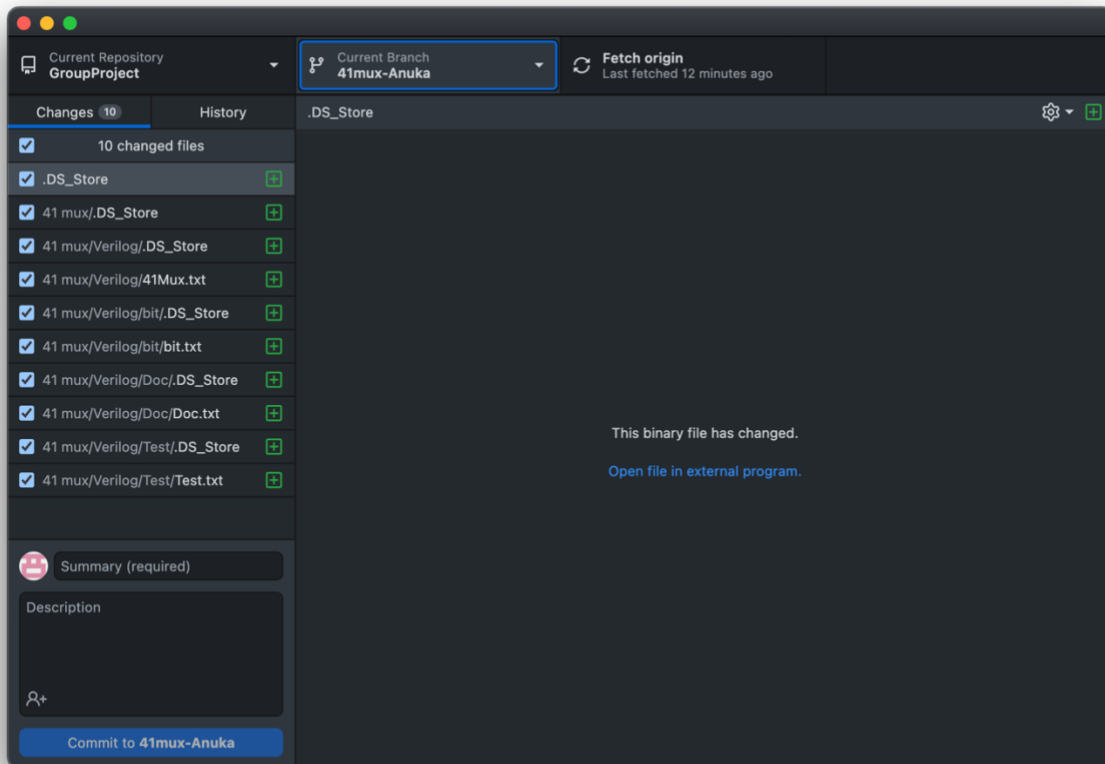
- Step 1: Creation of new repository
  - Create a public repo for your group project
  - Use a shortened name as the repository name
  - Create this repository without any files by the group leader
- Step 2: Collaboration
  - Now under the created repo go to Settings and select Manage access from the left sidebar
  - Now under Invite a collaborator, add your team members.



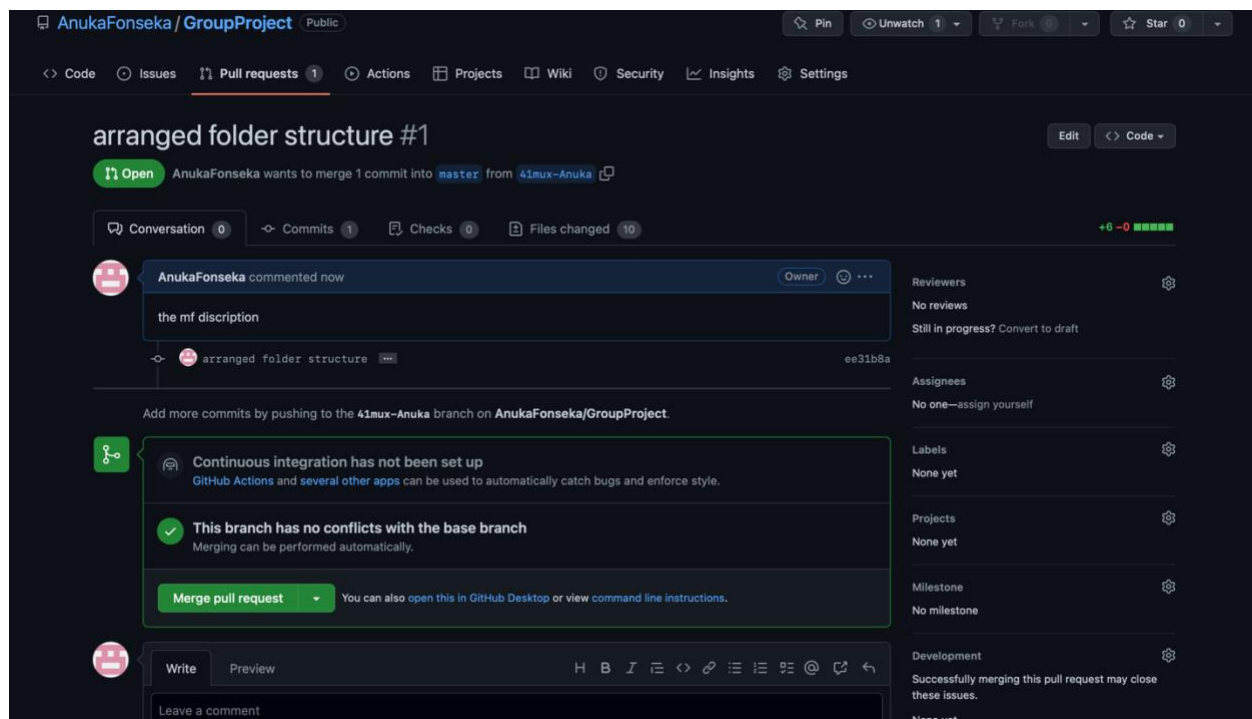
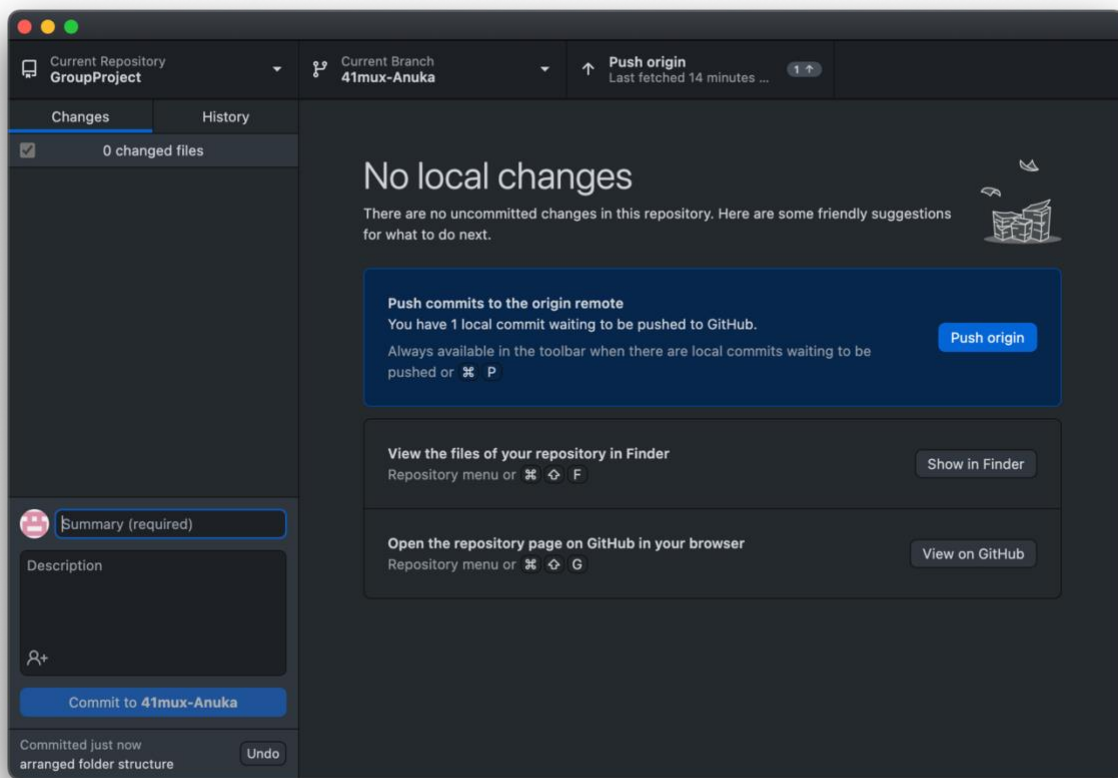
- Send the copied invitation link from the icon next to pending invite to your team members.
  - Team members can now accept the invitation to collaborate for the project
- Step 3: Project setup
  - Please make sure to create a branch from the master. NEVR MAKE CHANGES TO THE MASTER.

Figure 3: Project collaboration

- Distribute different possible modules among team members. Eg: 4:1 mux, 4-bit adder...etc.
- Create a branch with a title like “SubProjectName-‘yourfirstname’”, like that create different branch among team members
- Now create the folder structure for your chosen module. Module name is preferred to be short yet descriptive.
- Create some text files inside the folders.

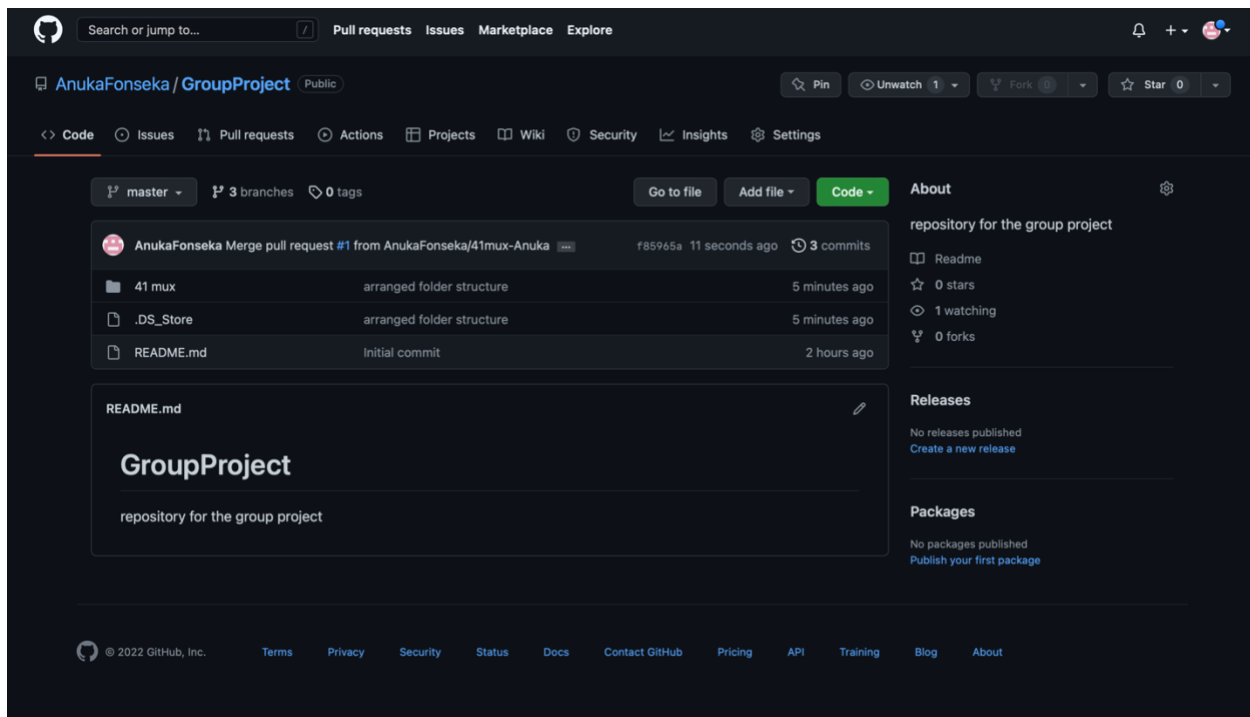


- Step 4:
  - Commit your changes and create pull requests. Make sure to give a short description of the change you have done in the commit message.



- Now the team leader can accept the pull requests if these changes are ok.

- Once accepted check the master branch and observe the differences.



## 1.4 TASKS

1. Add "ishara0925" as a collaborator to your CLEAN project repository. This repository is the one you are going to use for your project.
2. Add screen shots of the repository folder structure to your project presentation. Project presentation should be present inside the presentation folder in your project folder.
3. Add a short description of your project under the repository description.

