

EEE307 - Embedded Systems

H. M.P.C.Herath - 1262

Repository Making Tutorial : 1

(in this pdf, I publish all the screenshots step by step)

This is my original Github account created previously.

My account user name : pasinduc

Profile URL : <https://github.com/pasinduc>

1.1 FAMILIARIZING WITH GITHUB

Step 1 : Create your github account

- o Visit www.github.com
- o Create your account with a good username as you may need to use it in your future Work

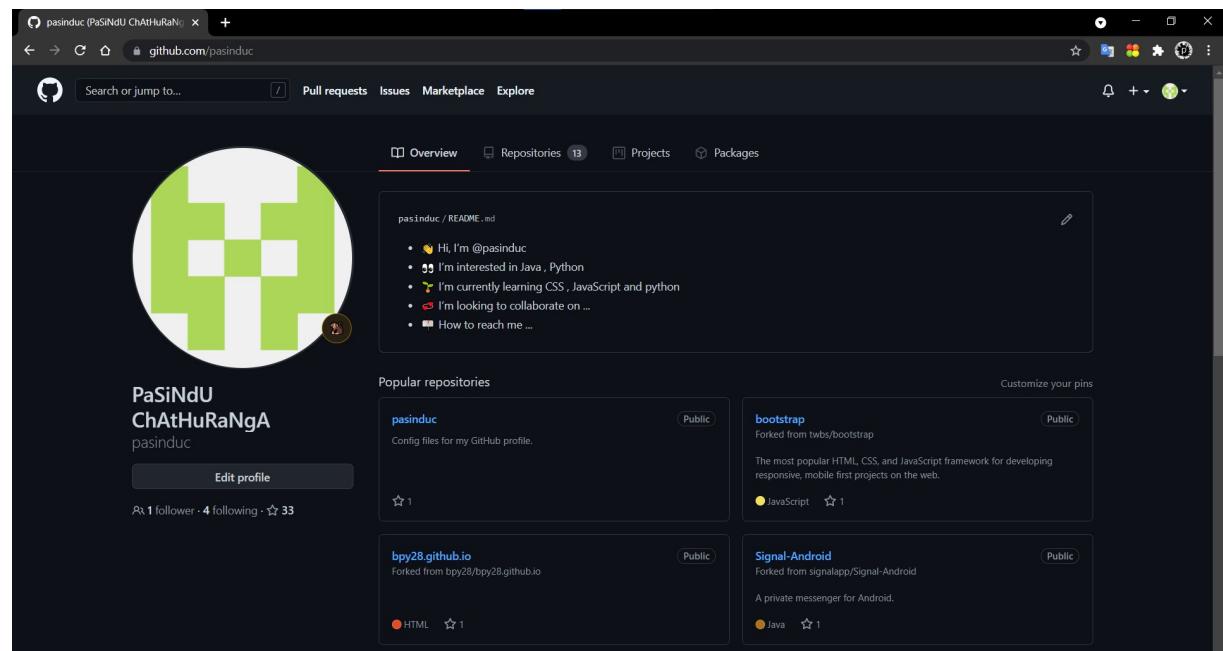
Step 2 : Create a repository

- o A repository is used to host a single project which can contain any type of data including folders.
- o Login to your github.com page
- o By selecting ‘+’ and New repository
- o Name of the repository “YourFirstName_HelloWorldLabs”, Use a meaningful name with no spaces
- o Add a short description. Try to describe the project in short
- o Select “Initialize this repository with a README”

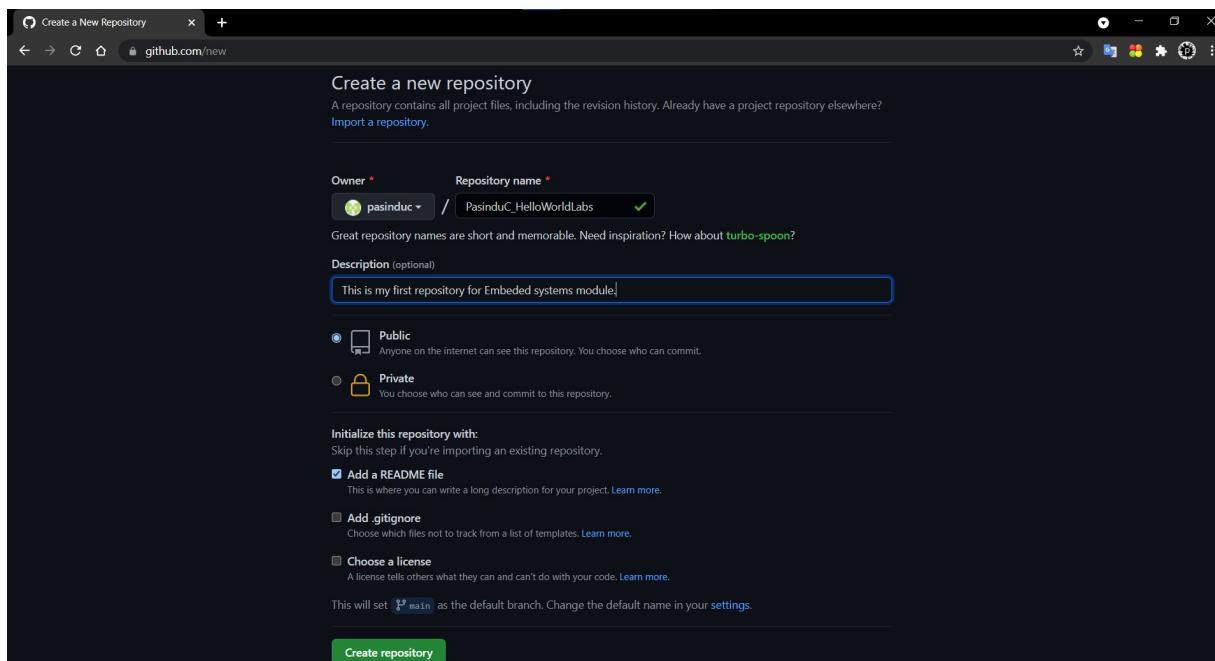
This is my original Github account created previously.

My account user name : pasinduc

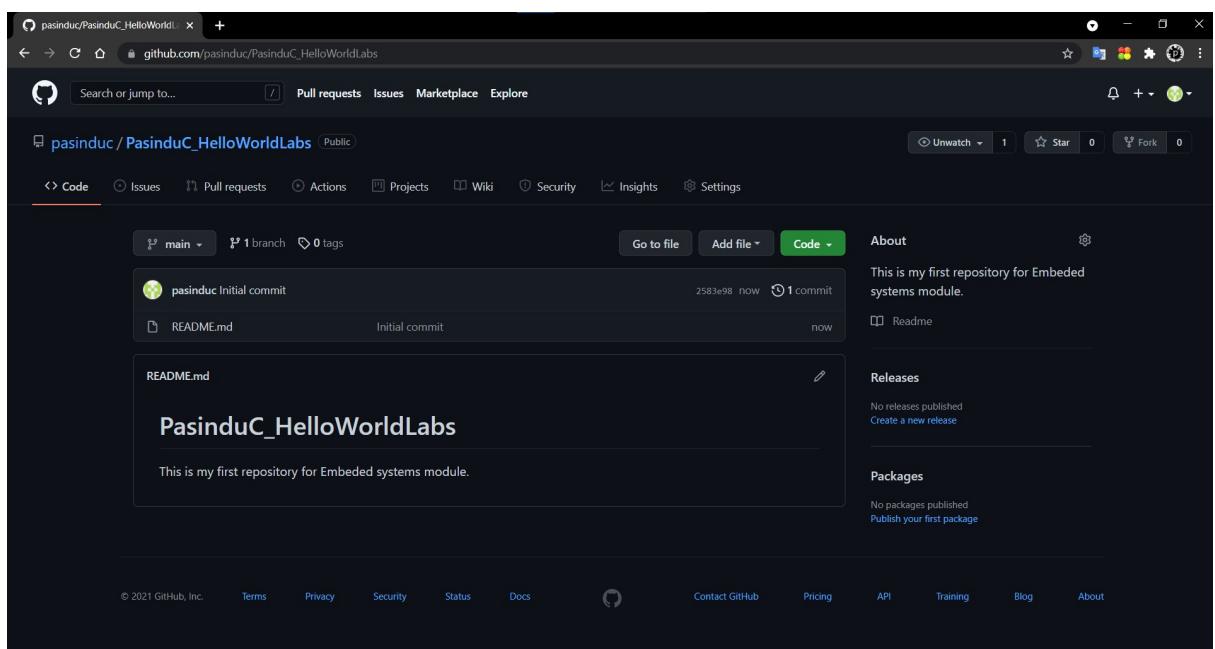
Profile URL : <https://github.com/pasinduc>



Presing + icon and select new repository



On That Repository :-

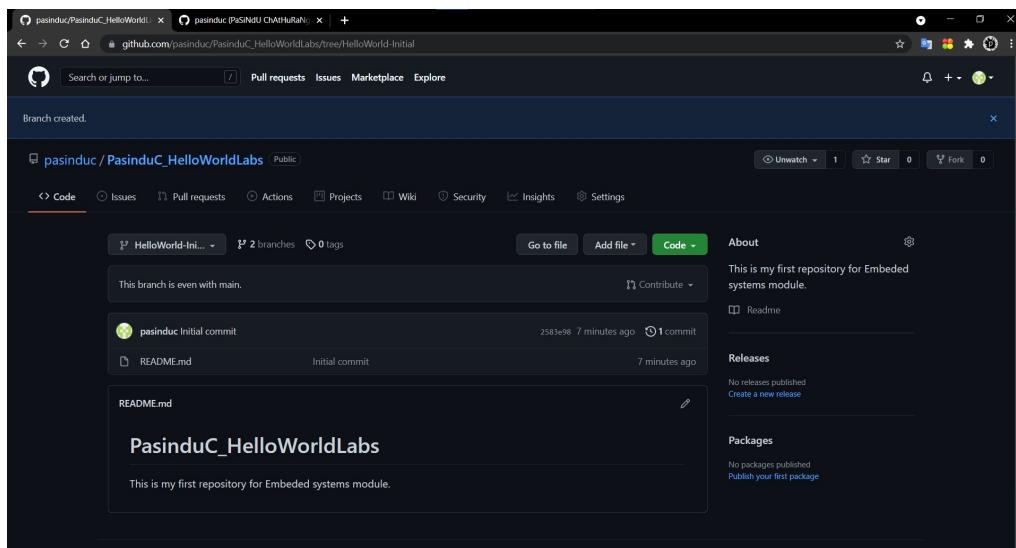


Step 3: Branching

- o This is the way to work on different versions of a project.
- o 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.
- o Create a new branch named “HelloWorld-Initial”
- o Now we have two branches named “master” and “HelloWorld-Initial”

I created another branch named “HelloWorld-Initial” and now I have 2 branches

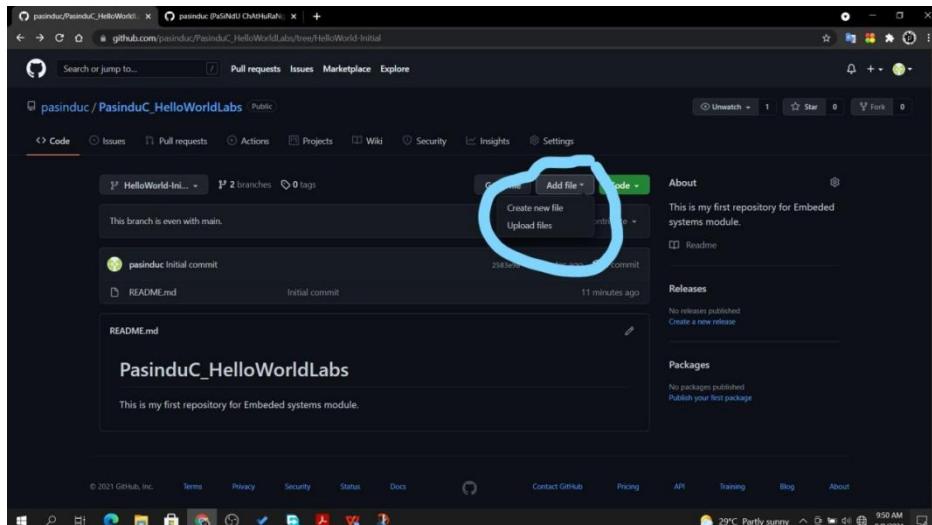
1. Main
2. HelloWorld-Initial



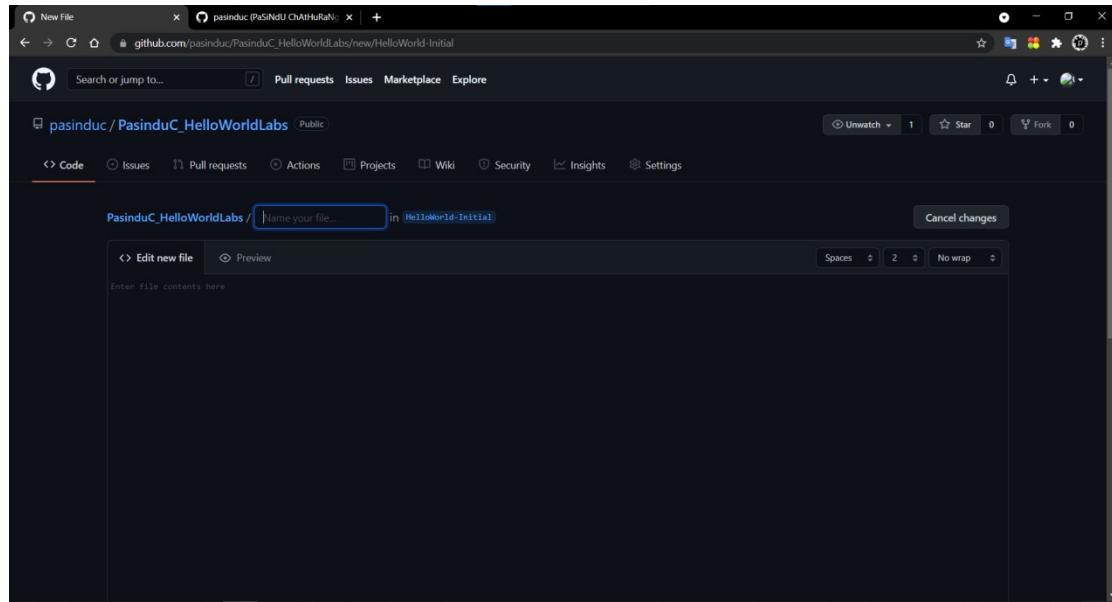
Step 4: Commit changes

- o Now you can start editing the files in your branch
- o Add a new file named “HelloWorld.txt”
- o Write some text in the “HelloWorld.txt”
- o Then click “Commit changes”

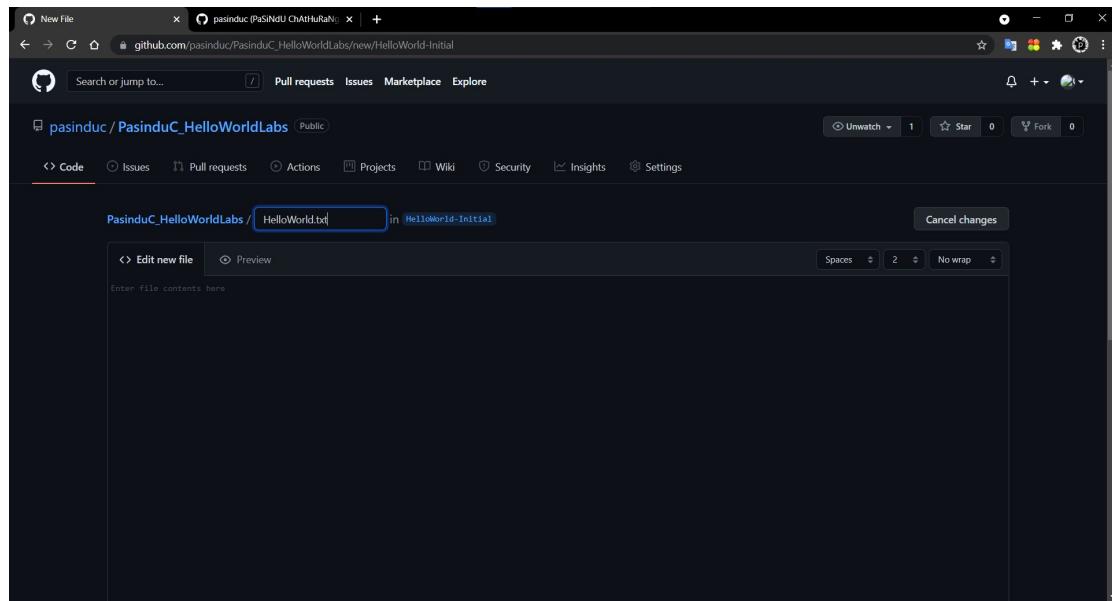
Go add file and select create new file



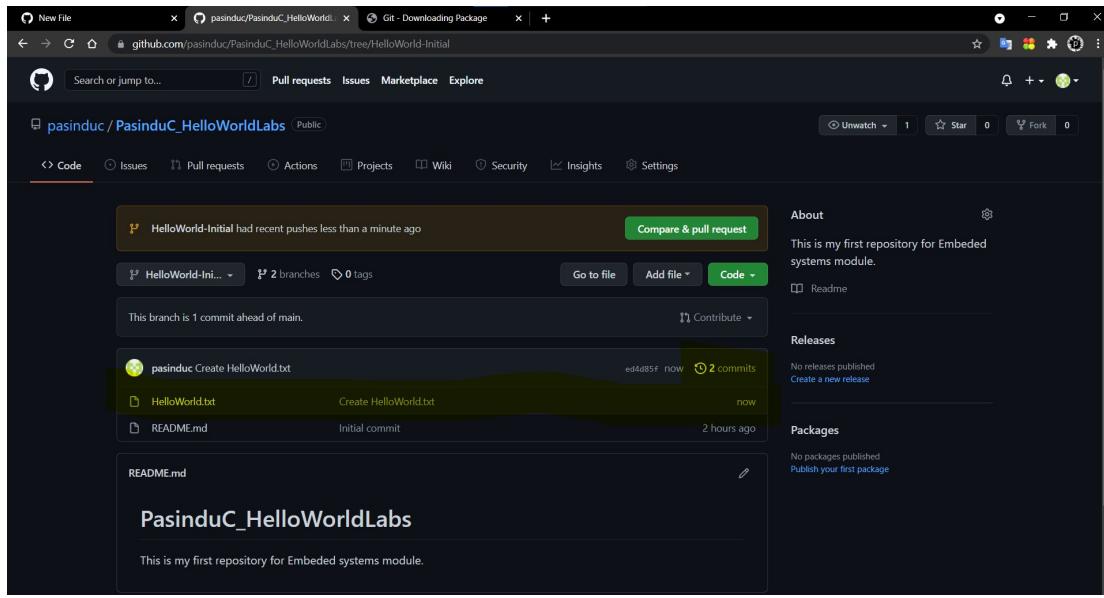
Naming that file.



Name that file with given name “ HelloWorld.txt ”

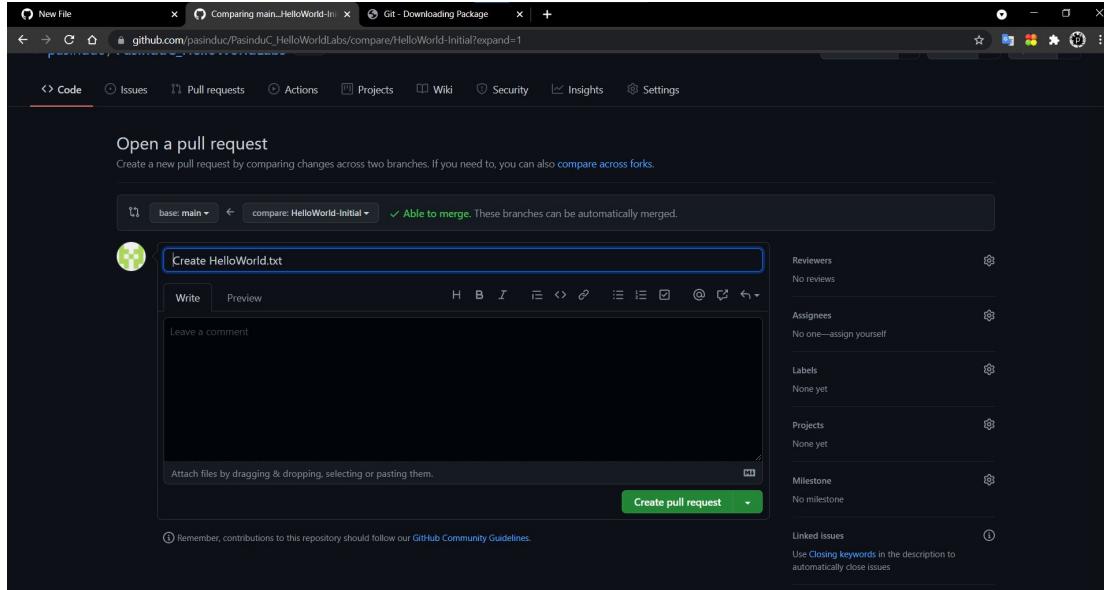


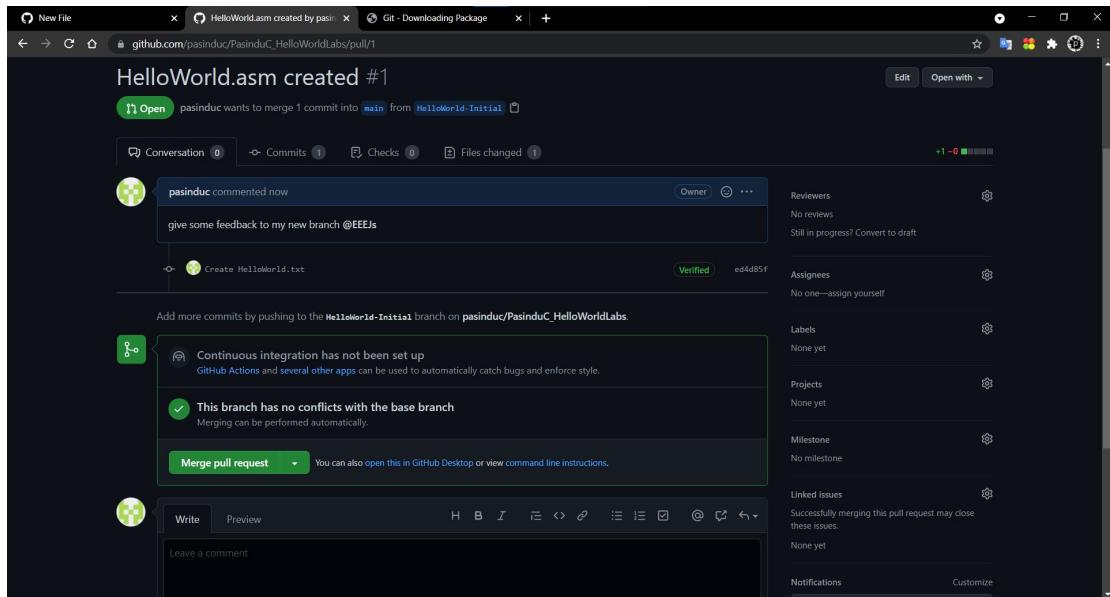
And Finished.



Step 5: Open a pull request

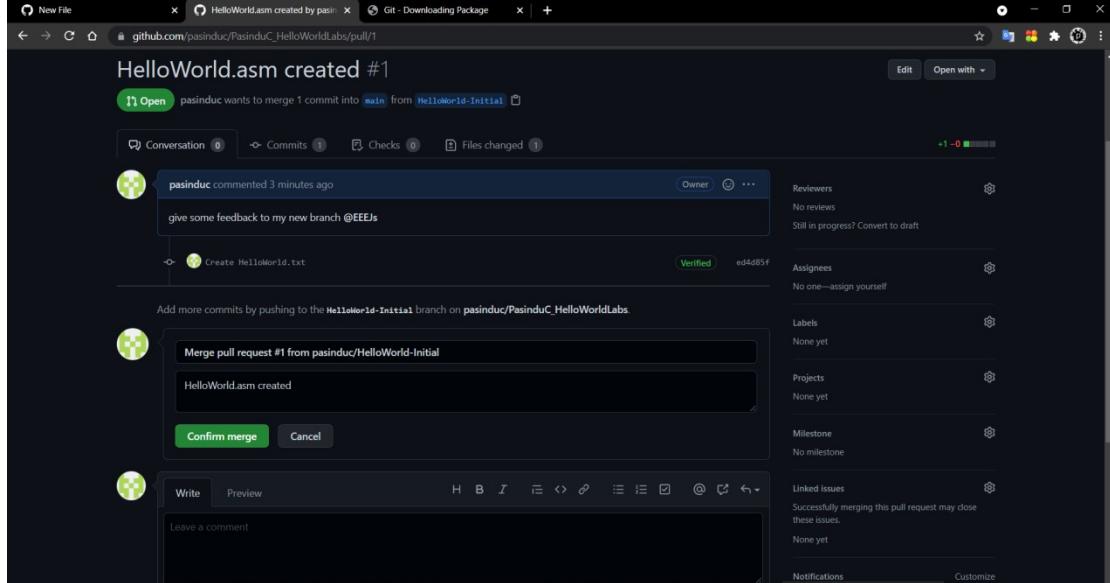
- o Using pull requests, you can ask someone to review your contributions before merge them to the master branch or some other branch.
- o It is possible open a pull request as soon as you make a commit.
- o You can use “mention” system to ask someone to give feedback
- o It is possible to open up pull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

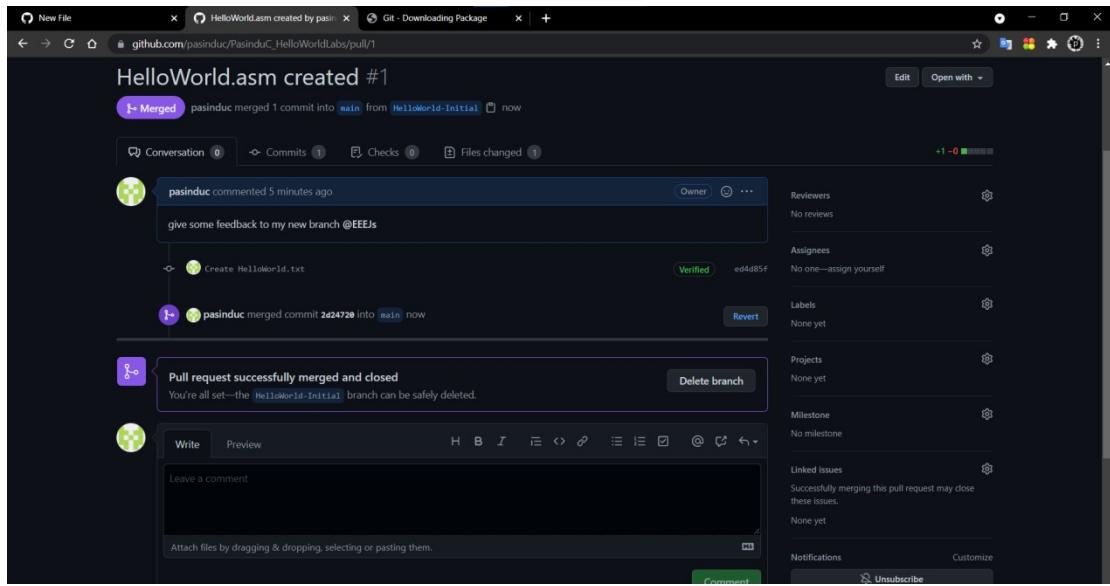




Step 6: Merge the pull request

- Once you finish testing your branch it can be merged to the master branch





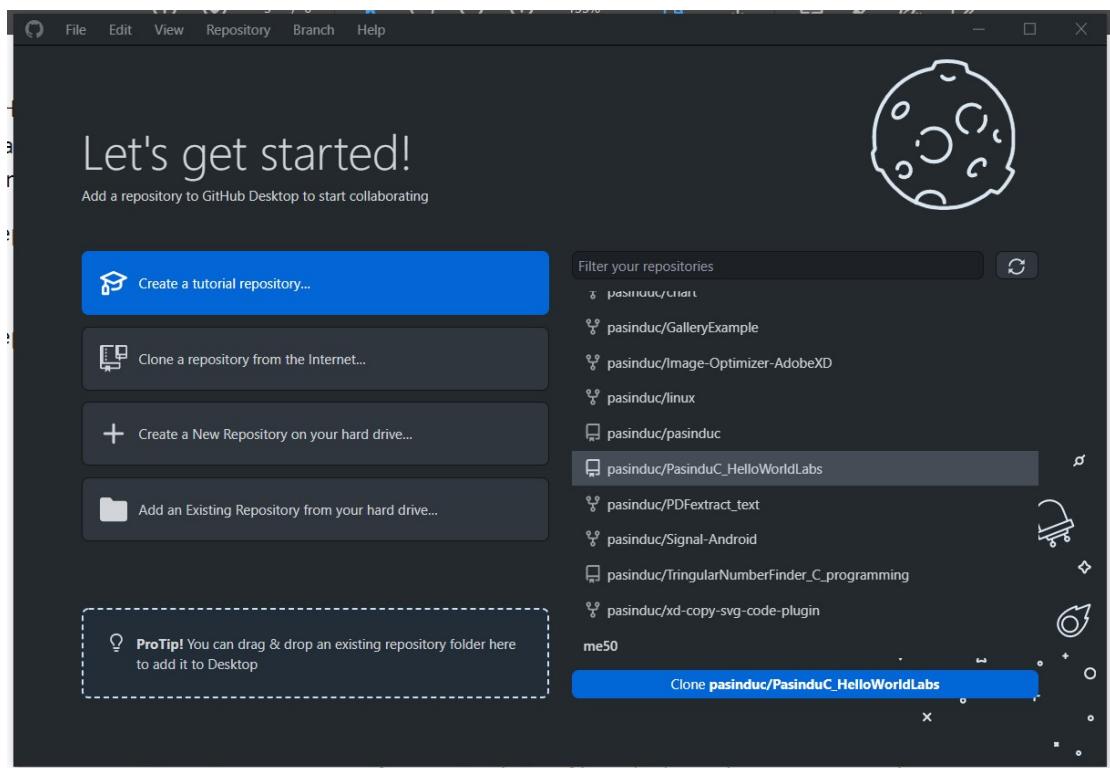
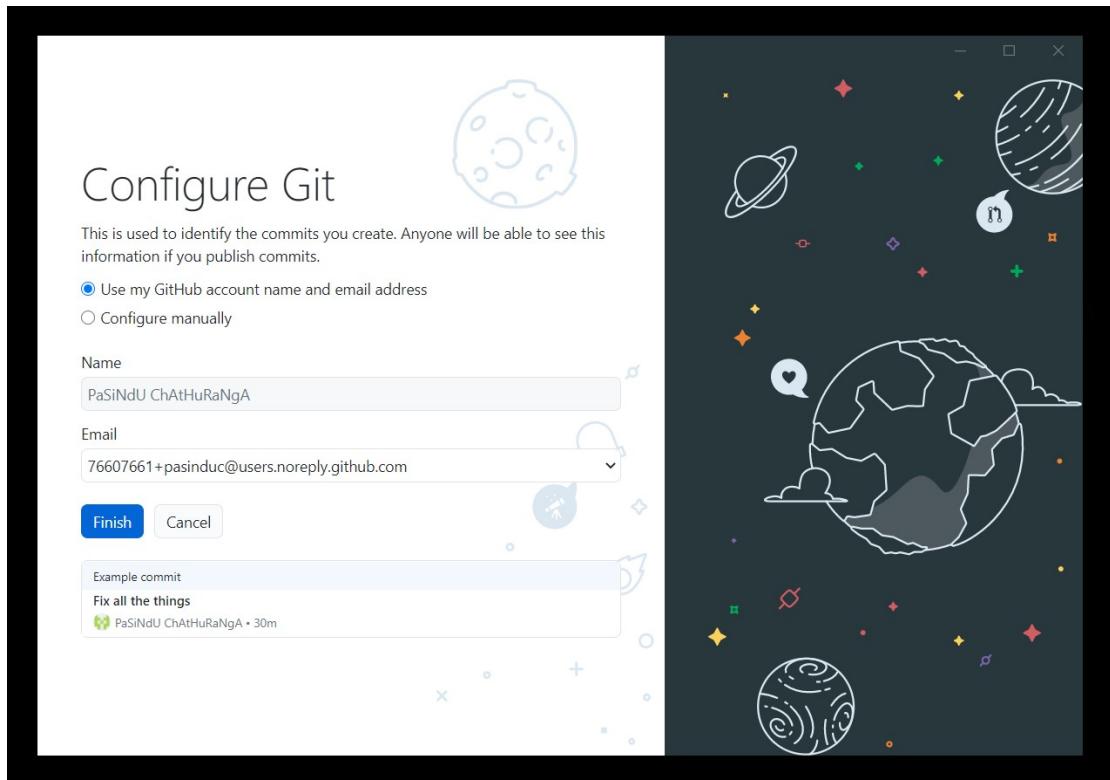
1.2 GITHUB GUI

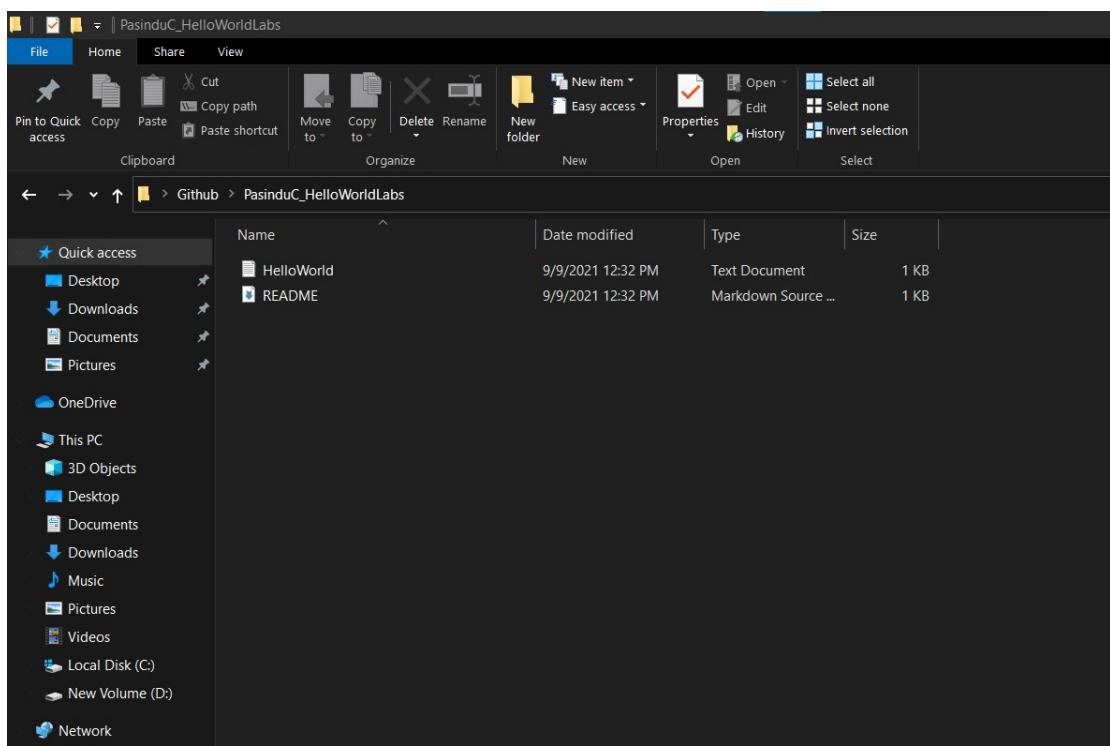
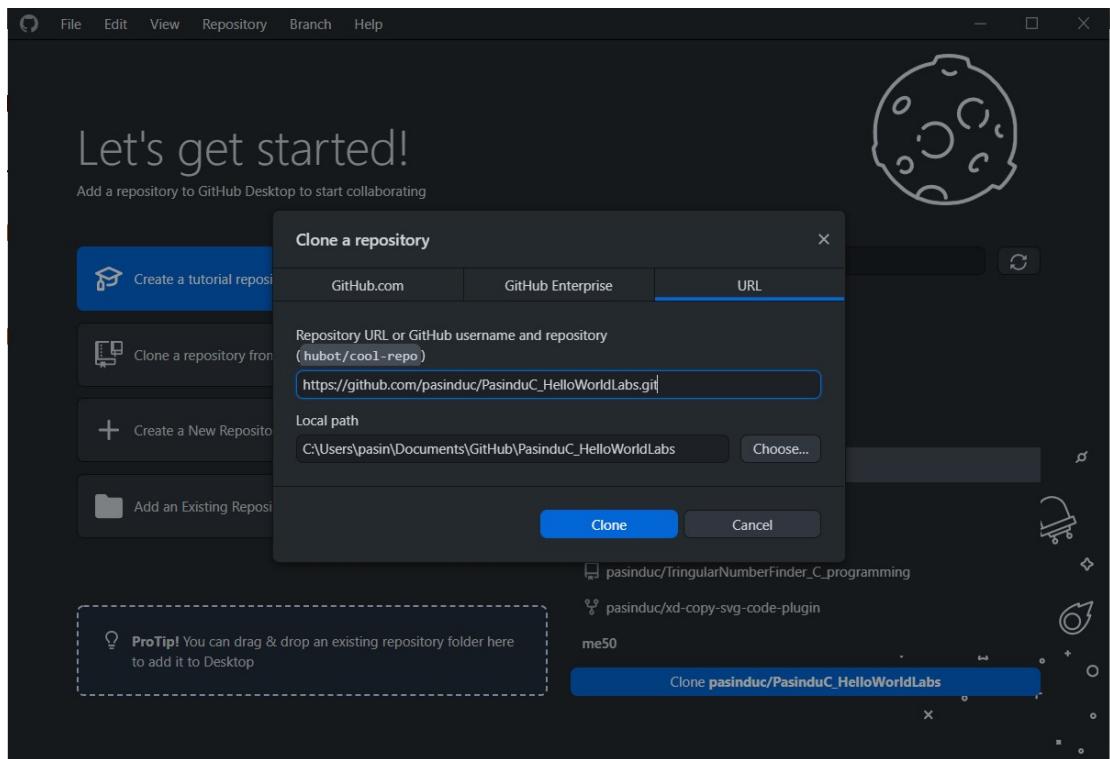
You can manage your codes locally with the help of command line arguments or with the Github GUI.

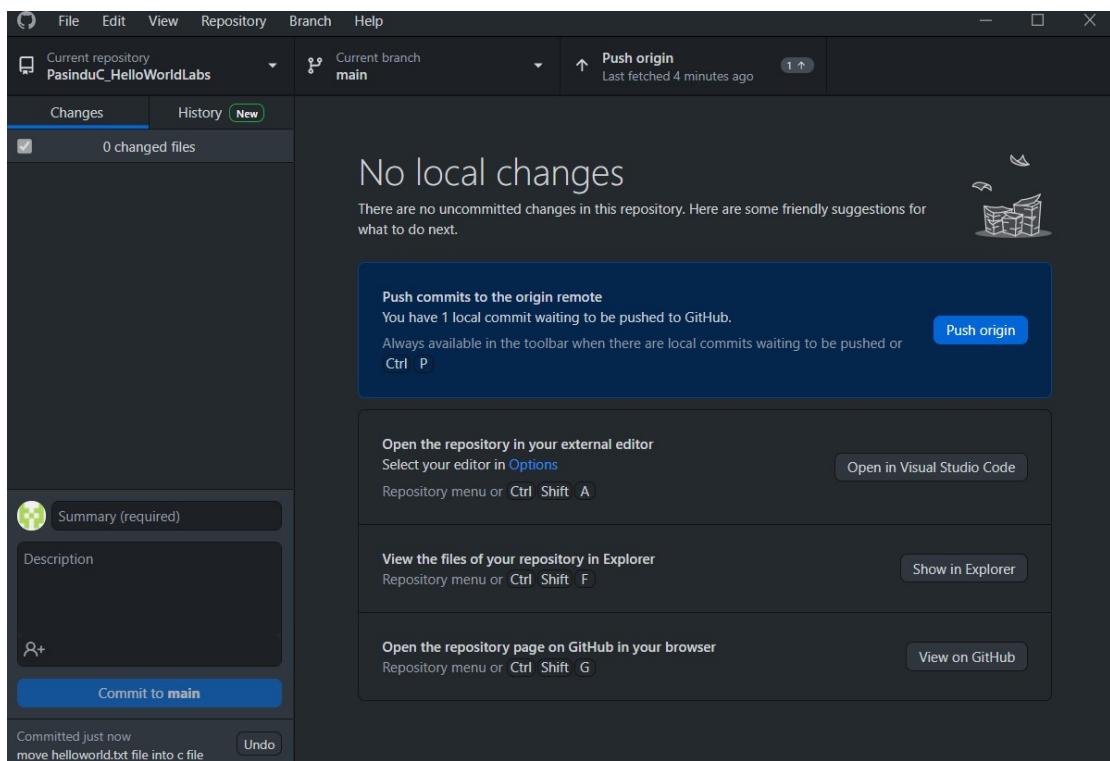
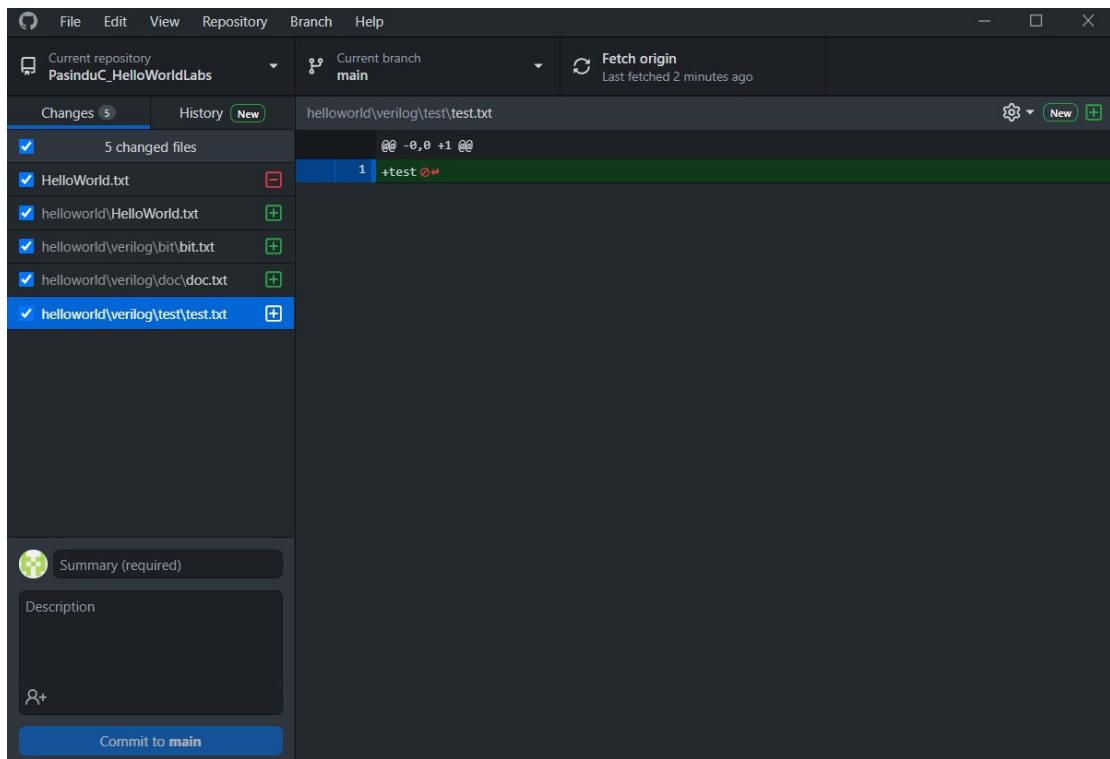
Following instructions are to setup the Github GUI in your computer.

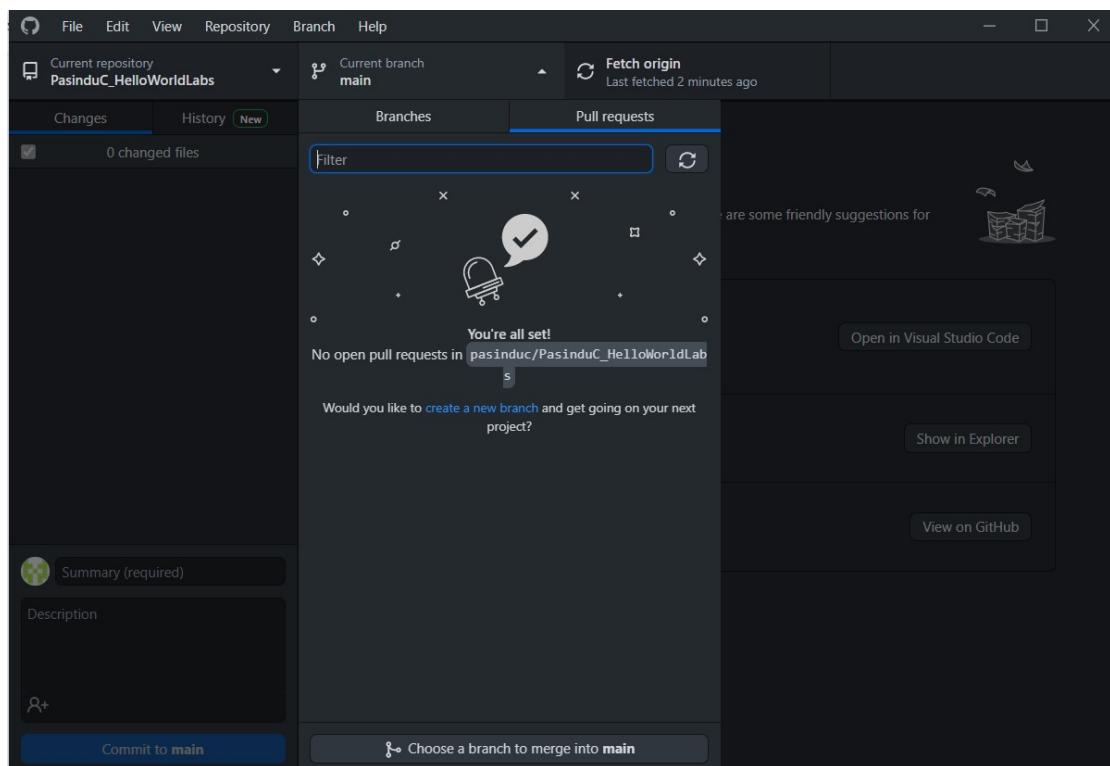
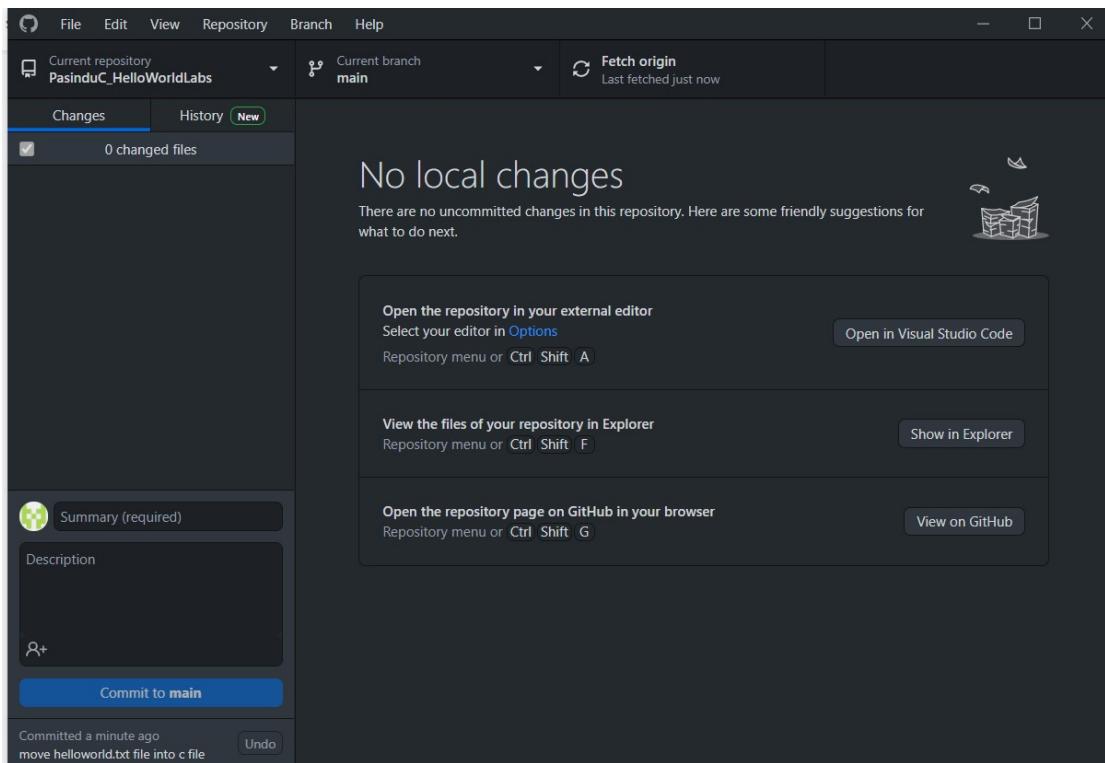
- **Step 1 :** Installing Github desktop
 - <https://desktop.github.com/> download and install Github desktop from here
 - Login using your Github account login details
- **Step 2 : Managing the repository**
 - Clone the repository your created at Github.com to your computer
 - Now view the repository 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.
 - C: 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.
 - **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.

Figure 4 : Pull request









HelloWorldLab CLI : gh repo clone pasinduc/PasinduC_HelloWorldLabs
Repository Url : https://github.com/pasinduc/PasinduC_HelloWorldLabs

1.3 COLLABORATION

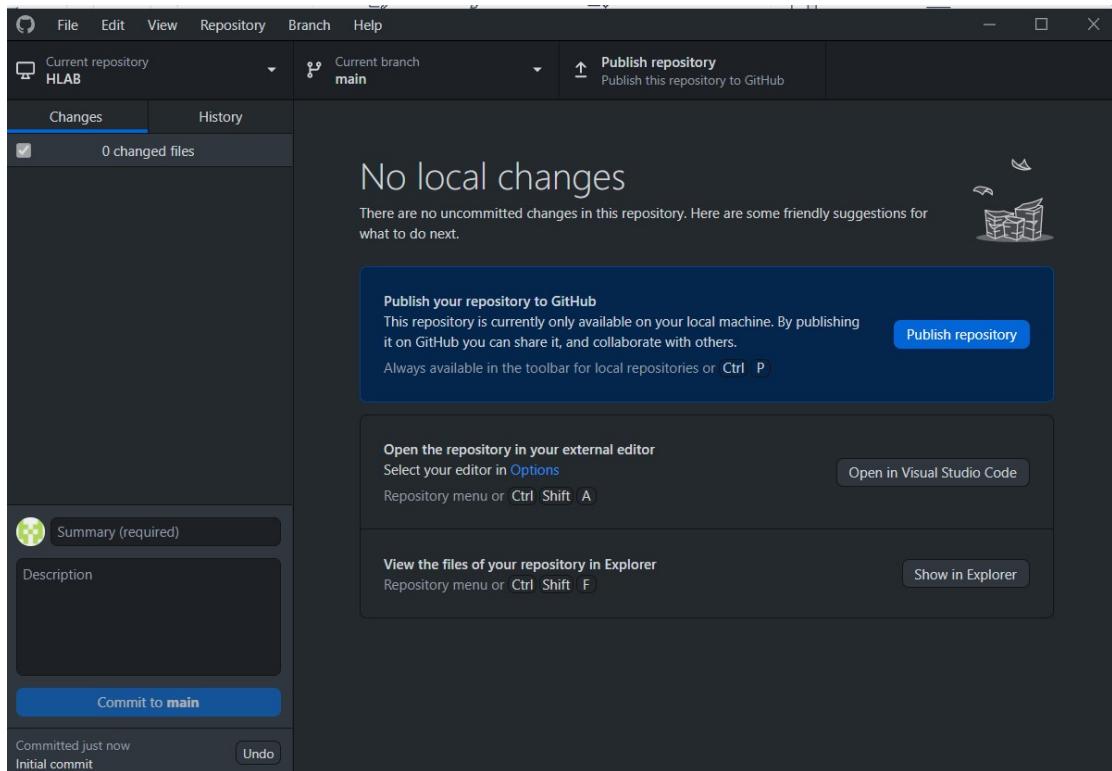
Main advantage of Git repo is the possibility of working simultaneously on different parts of a project.

Follow the steps to create your own repo to be shared between team members.

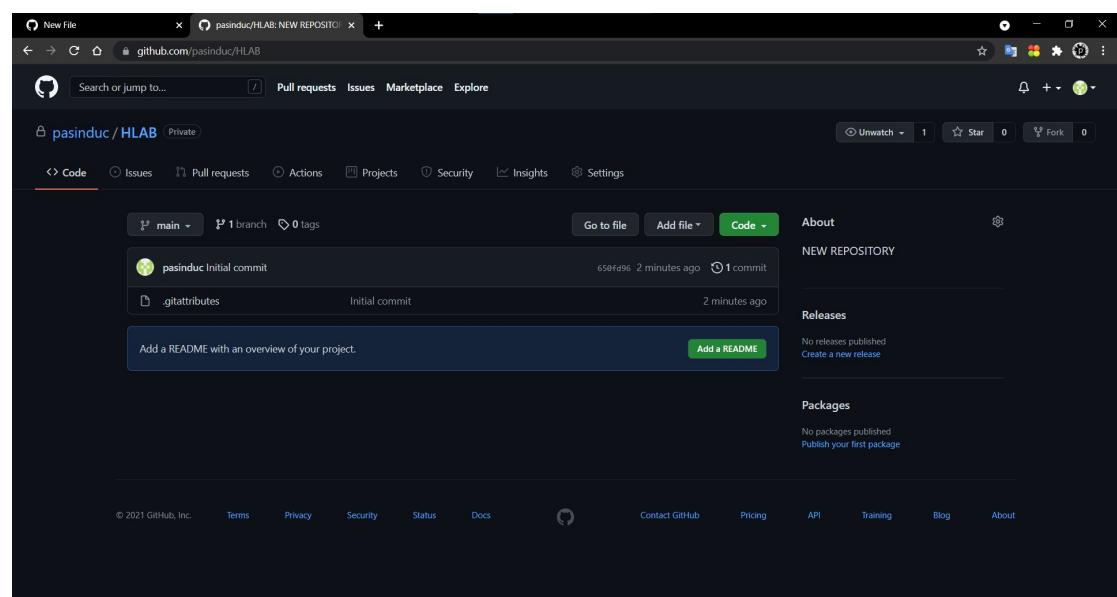
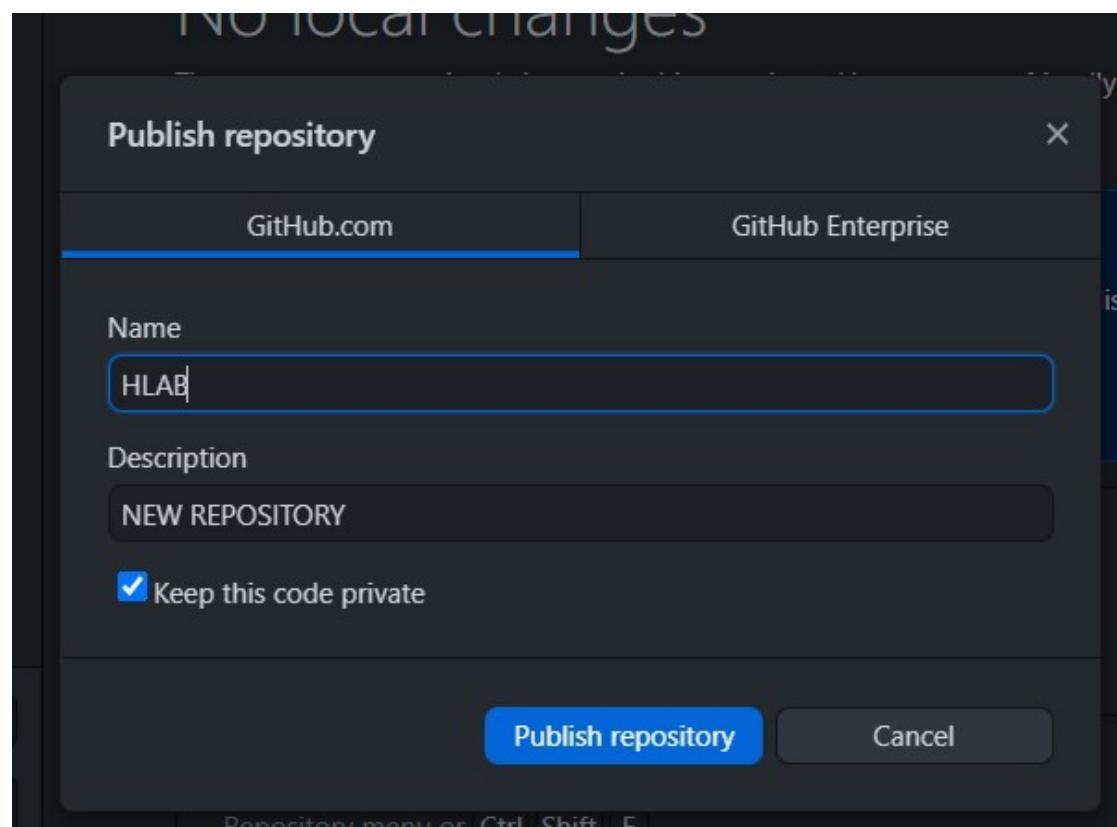
Step 1: Creation of new repository

- o Create a public repo for your group project
- o Use a shortened name as the repository name
- o Create this repository without any files by the group leader

Create new repository called HLAB

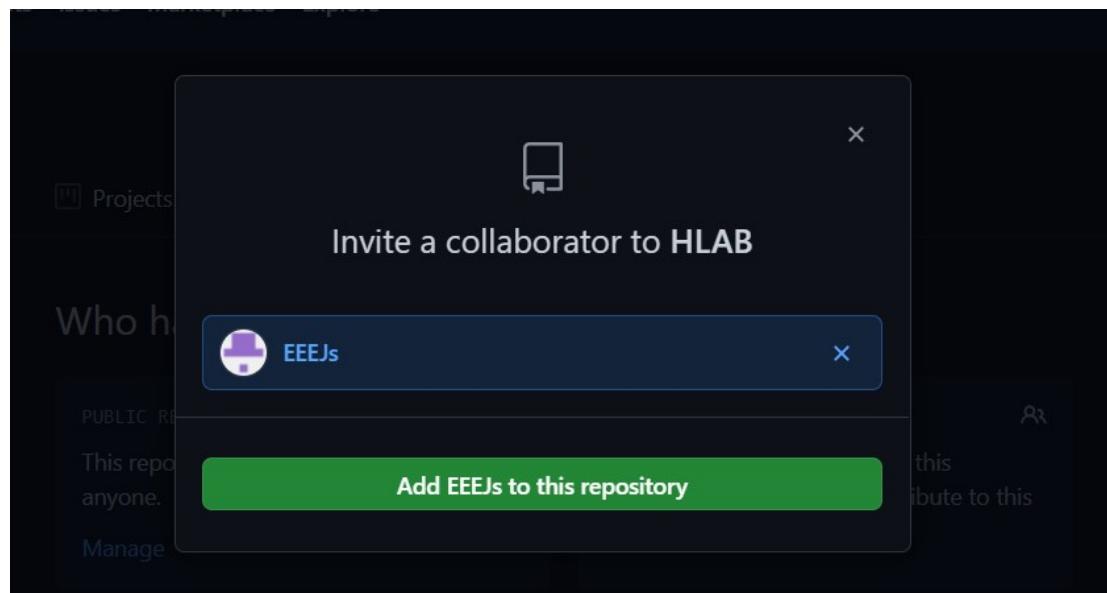
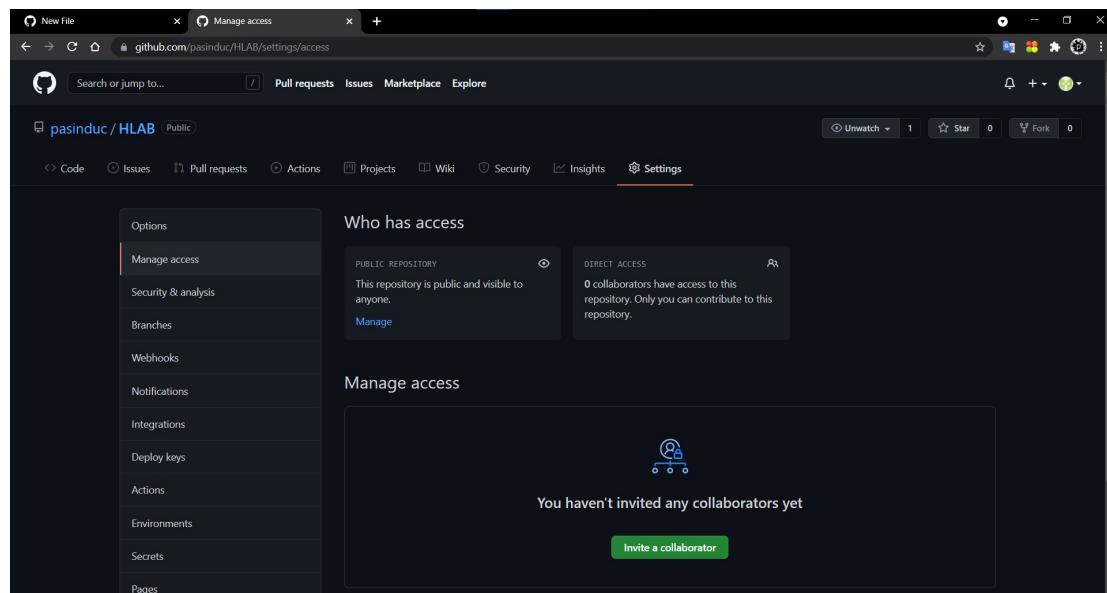


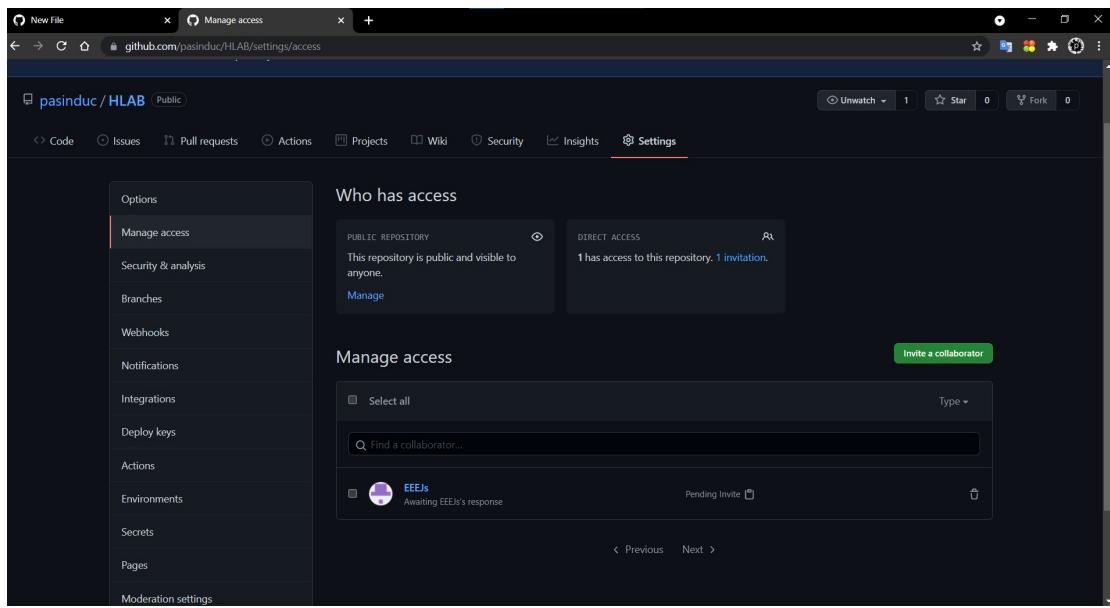
Publish the repository



Step 2: Collaboration

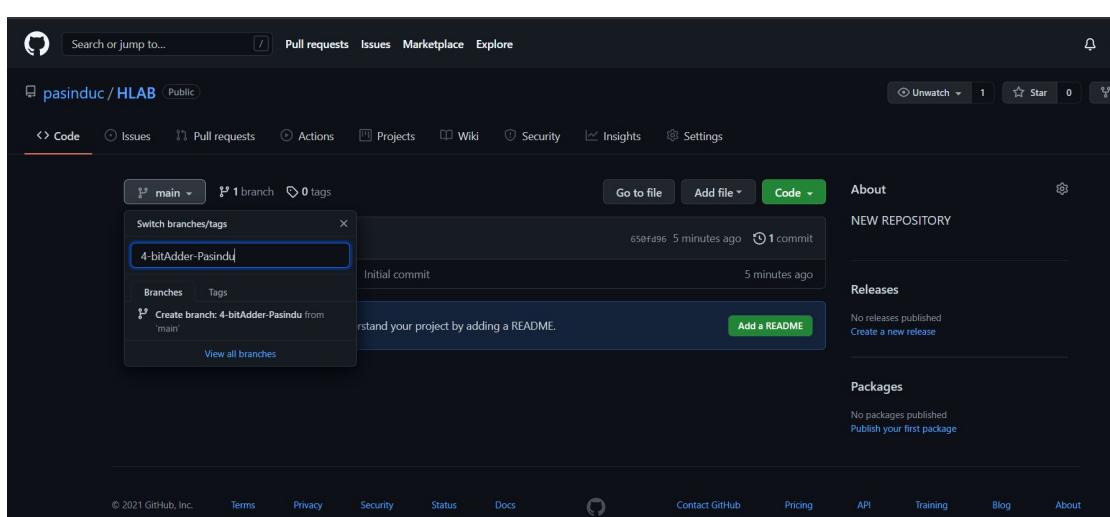
- o Now under the created repo goto Settings and select Manage access from the left sidebar
- o Now under Invite a collaborator, add your team members.
- o Send the copied invitation link from the icon next to pending invite to your team members.
- o Team members can now accept the invitation to collaborate for the project



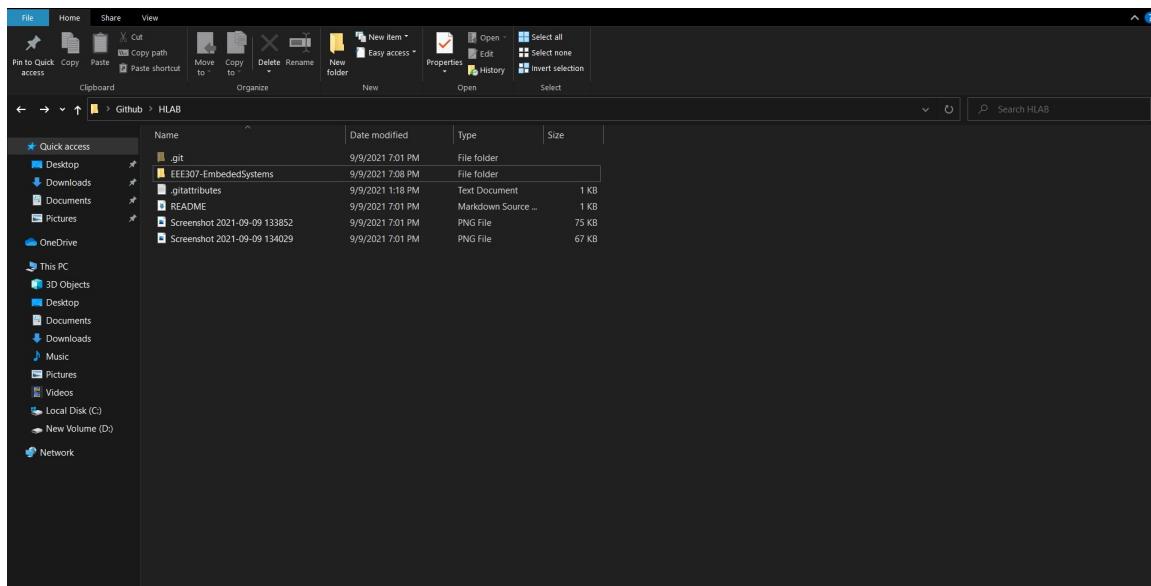


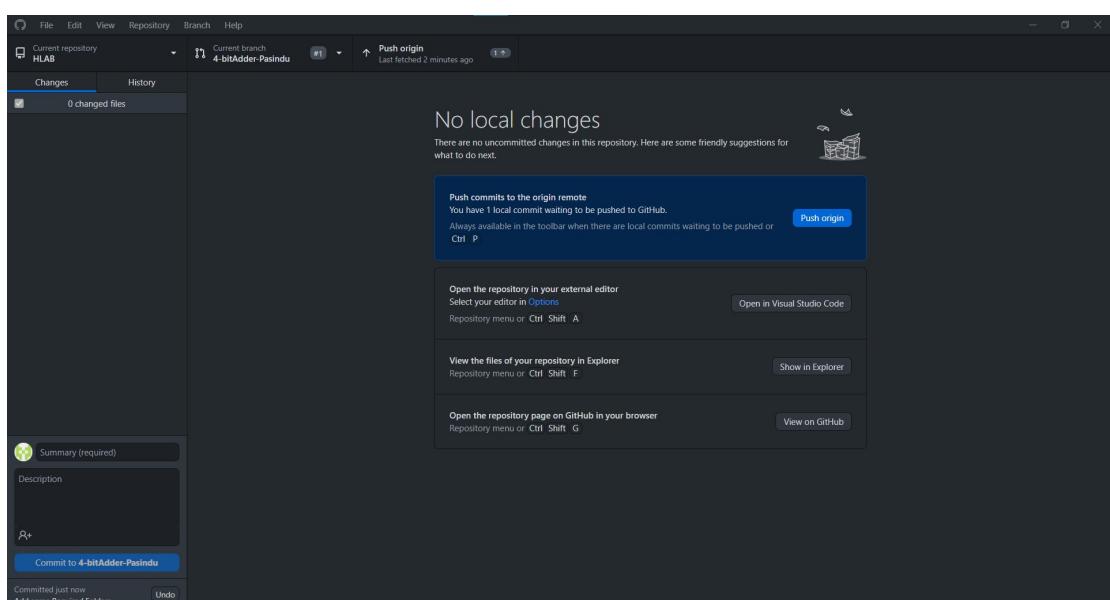
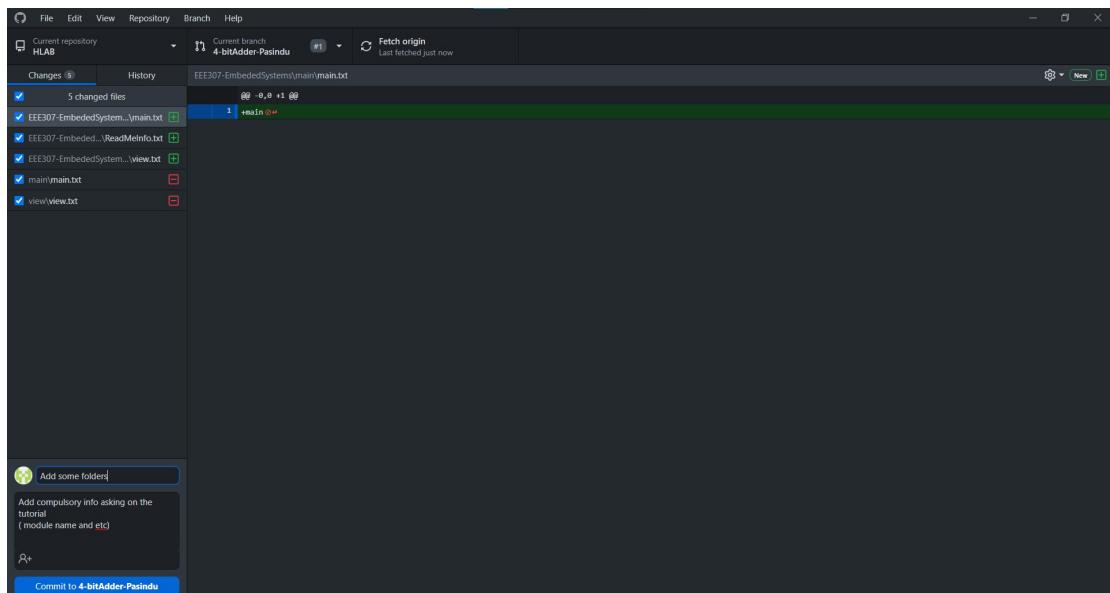
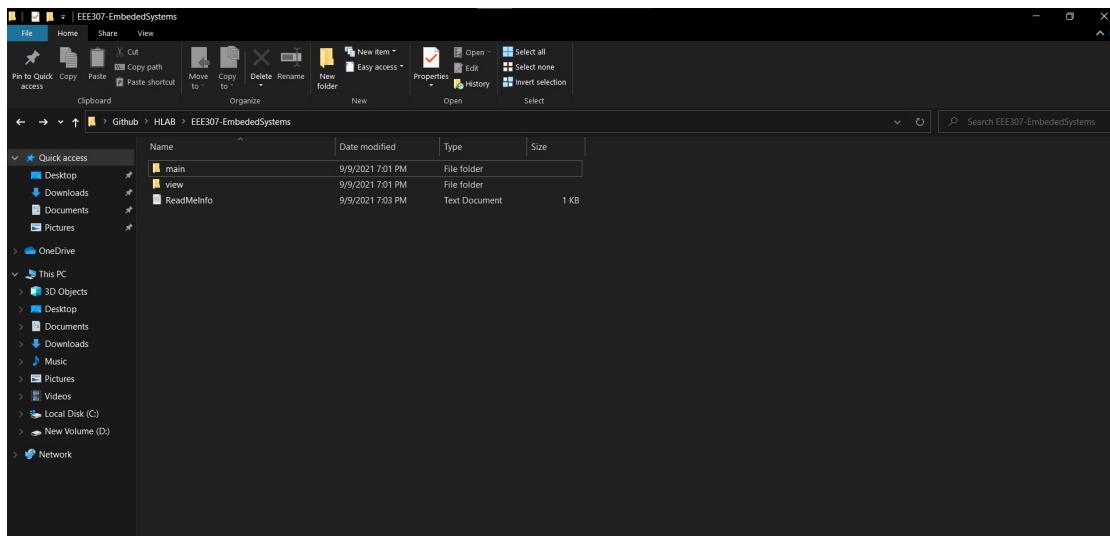
Step 3: Project setup

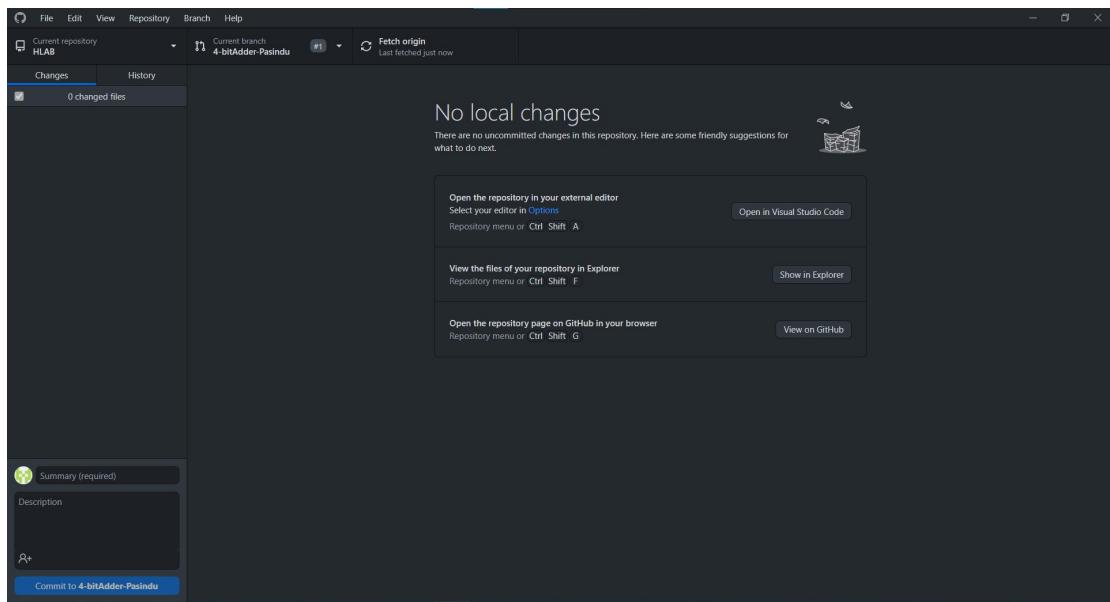
- o Please make sure to create a branch from the master. NEVER MAKE CHANGES TO THE MASTER.
- o Distribute different possible modules among team members. Eg: 4:1 mux, 4-bit adder...etc.
- o Create a branch with a title like “SubProjectName-‘yourfirstname’”, like that create different branch among team members
- o Now create the folder structure for your chosen module. Module name is preferred to be short yet descriptive.
- o Create some text files inside the folders.



A screenshot of a web browser displaying a GitHub repository page. The URL in the address bar is github.com/pasinduc/HLAB/tree/4-bitAdder-Pasindu. The page shows a "Branch created." message. The repository name is **pasinduc / HLAB** (Public). It has 2 branches and 0 tags. The main branch is even with main. A commit by **pasinduc** titled "Initial commit" was made 6 minutes ago. The commit message is "Initial commit". There is a button to "Add a README". The page also includes sections for About, Releases, and Packages.

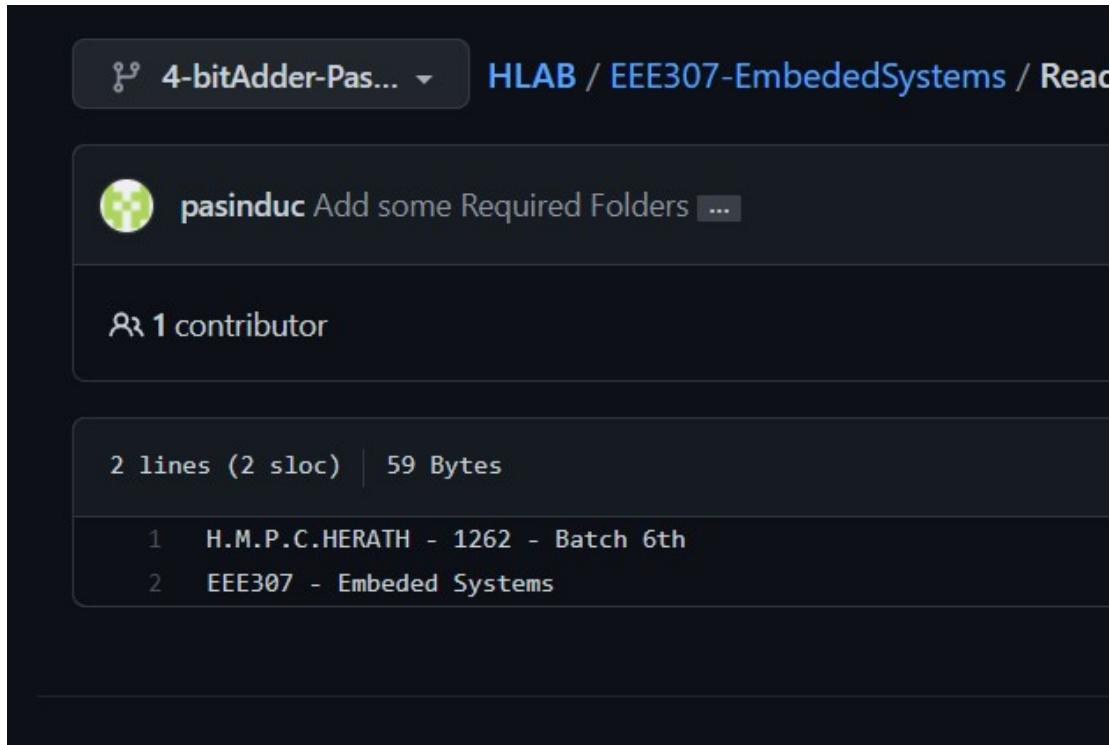






This screenshot shows the GitHub web interface for the 'HLAB' repository. The URL is 'github.com/pasinduc/HLAB/tree/4-bitAdder-Pasindu'. The page has a dark theme. At the top, there's a search bar and navigation links for Pull requests, Issues, Marketplace, and Explore. The repository name 'pasinduc / HLAB (Public)' is displayed. On the right, there are buttons for Unwatch, Star, Fork, and a copy icon. Below the header, there are tabs for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The 'Code' tab is selected. It shows a list of commits: 'Add some Required Folders' by pasinduc, 'Initial commit' by EEE307-EmbeddedSystems, 'Create README.md' by pasinduc, and two 'screen shots' by pasinduc. To the right of the code area, there are sections for About (New Repository), Releases (No releases published), and Packages (No packages published). A README.md file is shown with the text 'HLAB' and '4-ADDER BRANCH'.

This screenshot shows the GitHub web interface for the 'HLAB / EEE307-EmbeddedSystems' branch. The URL is 'github.com/pasinduc/HLAB/tree/4-bitAdder-Pasindu/HLAB%20%2F%20EEE307-EmbeddedSystems%2F'. The page has a dark theme. The repository name 'pasinduc / HLAB (Public)' is displayed. On the right, there are buttons for Unwatch, Star, Fork, and a copy icon. Below the header, there are tabs for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The 'Code' tab is selected. It shows a list of commits: 'Add some Required Folders' by pasinduc, 'Add some Required Folders' by EEE307-EmbeddedSystems, 'Add some Required Folders' by pasinduc, and 'ReadMeInfo.txt' by pasinduc. There are also sections for History and a '...' button. At the bottom, there are links for Contact GitHub, Pricing, API, Training, Blog, and About.

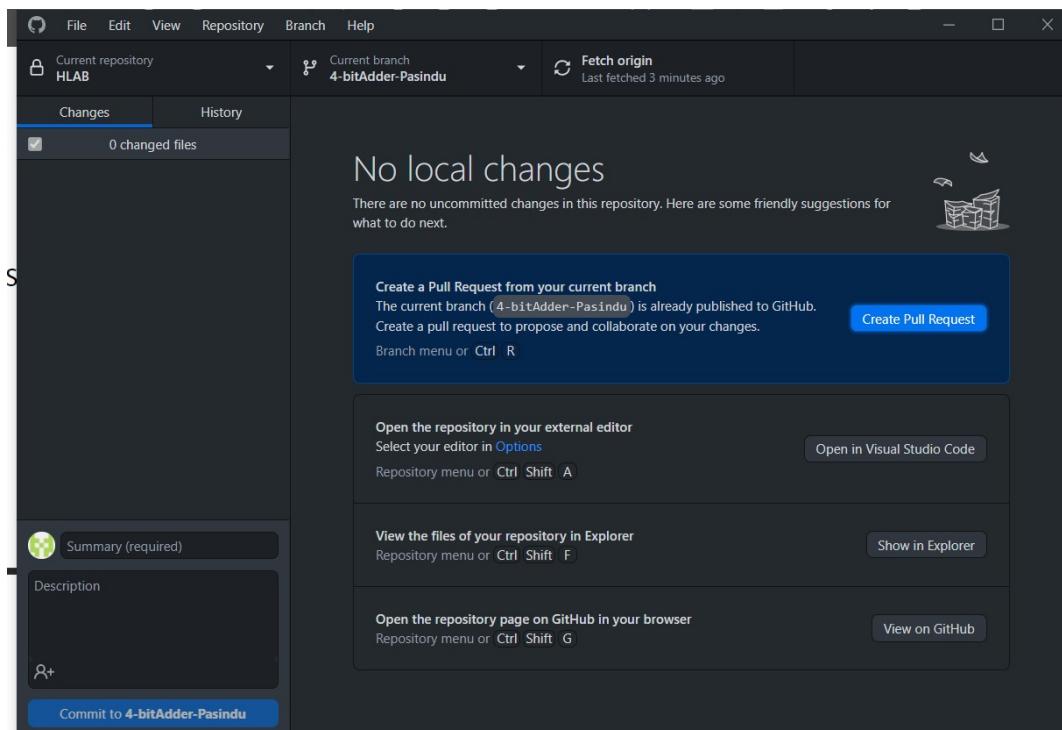


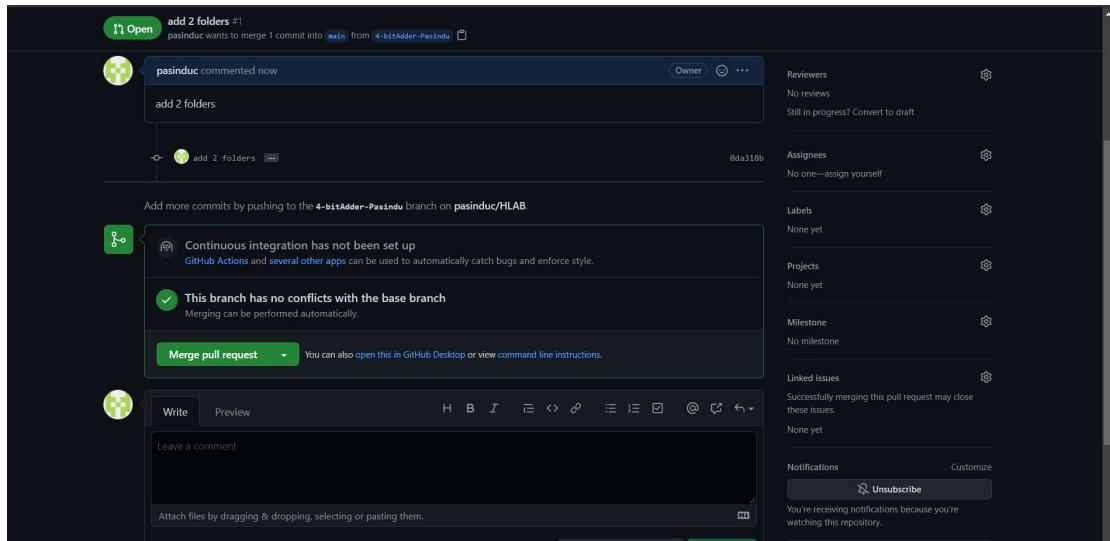
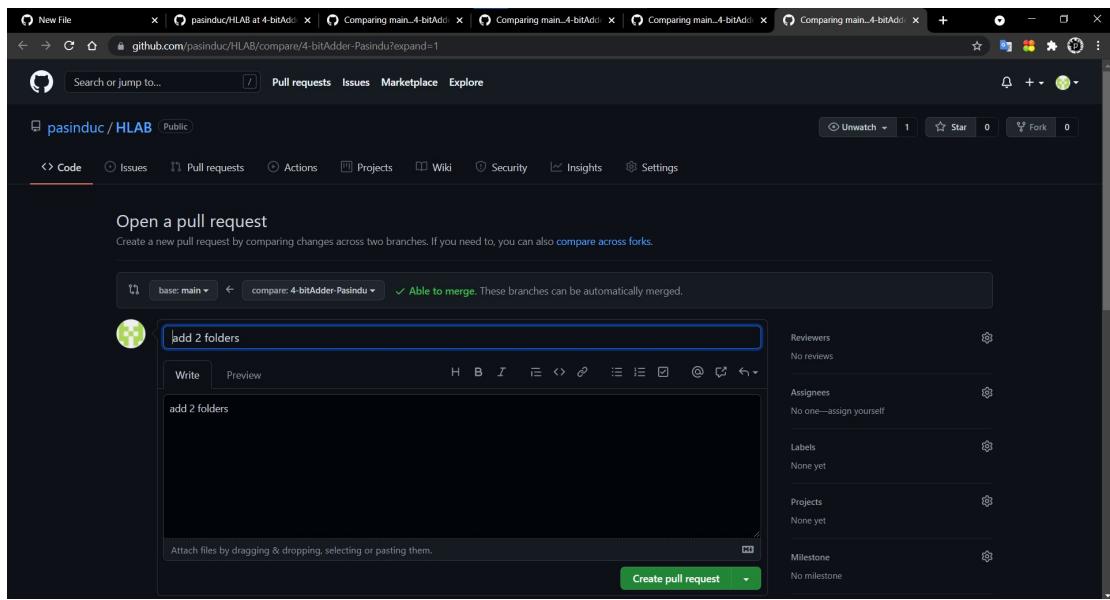
Step 4:

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

Figure 5: Project collaboration

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

The screenshot shows the "Manage access" section of a GitHub repository. It displays two sections: "PUBLIC REPOSITORY" and "DIRECT ACCESS". Under "DIRECT ACCESS", it says "2 have access to this repository. 2 invitations." Below this, there is a table listing two users:

User	Status	Action
EEEJs	Awaiting EEEJs's response	Pending Invite
ishara0925	Awaiting ishara0925's response	Pending Invite

At the bottom right of the table is a green button labeled "Invite a collaborator".

“4-bitadder-pasindu” branch

The screenshot shows the GitHub repository page for "pasinduc / HLAB". The repository is public. The main interface includes tabs for Code, Issues, Pull requests (1), Actions, Projects, Wiki, Security, Insights, and Settings. The "Code" tab is selected. The repository has 2 branches and 0 tags. A message indicates this branch is 5 commits ahead of main and 6 commits behind main. The commit history for the "4-bitadder-Pas..." branch shows the following activity:

Commit	Message	Time	Author
65d4d14	add some screenshot	44 seconds ago	6 commits
1 hour ago	EEE307-EmbeddedSystems	Add some Required Folders	
7 hours ago	.gitattributes	Initial commit	
43 seconds ago	4-bitadder-Pasindu local folder.png	add some screenshot	
6 hours ago	README.md	Create README.md	

On the right side, there are sections for About (New Repository, Readme), Releases (No releases published, Create a new release), and Packages (No packages published, Publish your first package). The repository description is "HLAB" and the branch name is "4-ADDER BRANCH".

Main branch (before add presentation file)

The screenshot shows a GitHub repository page for 'pasinduc/HLAB'. At the top, there are buttons for 'main' (selected), '2 branches', '0 tags', 'Go to file', 'Add file', and 'Code'. Below this, a list of commits is shown:

Commit	Message	Time
	pasinduc Merge branch 'main' of https://github.com/pasinduc/HLAB into main	a2d18fd 15 seconds ago
	.gitattributes Initial commit	7 hours ago
	README.md Update README.md	25 minutes ago
	Repository Local (Main) Folder View... add screenshot	19 seconds ago

On the right side of the page, there are sections for 'About', 'New', 'Releases', 'Packaging', and 'No packages published'.

The README.md file content is as follows:

```
HLAB

NEW REPOSITORY

there are 2 brances in this repository

1. Main
2. 4-bitAdder-Pasindu

all the required informations are included into "4-bitAdder-Pasindu" branch..
```

HLAB CLI : gh repo clone pasinduc/HLAB

URL FOR REPOSITORY : <https://github.com/pasinduc/HLAB>

Note :-

1st repository : HelloWorldLab CLI : gh repo clone pasinduc/PasinduC_HelloWorldLabs
Repository Url : https://github.com/pasinduc/PasinduC_HelloWorldLabs
(included 2 branches)

2nd repository: HLAB CLI : gh repo clone pasinduc/HLAB
URL FOR REPOSITORY : <https://github.com/pasinduc/HLAB>

(included 2 branches)