

Practical 4 : Introduction to Git and Github

What is Git and Github?

Git: A version control software

Github: A web server that provides a web service which runs Git on web

In this practical, we are exploring these two together.

Getting Started.

First off, go to github.com and sign up as a user of Github as shown in Figure 1. (you may ignore this step if you already have a github account)

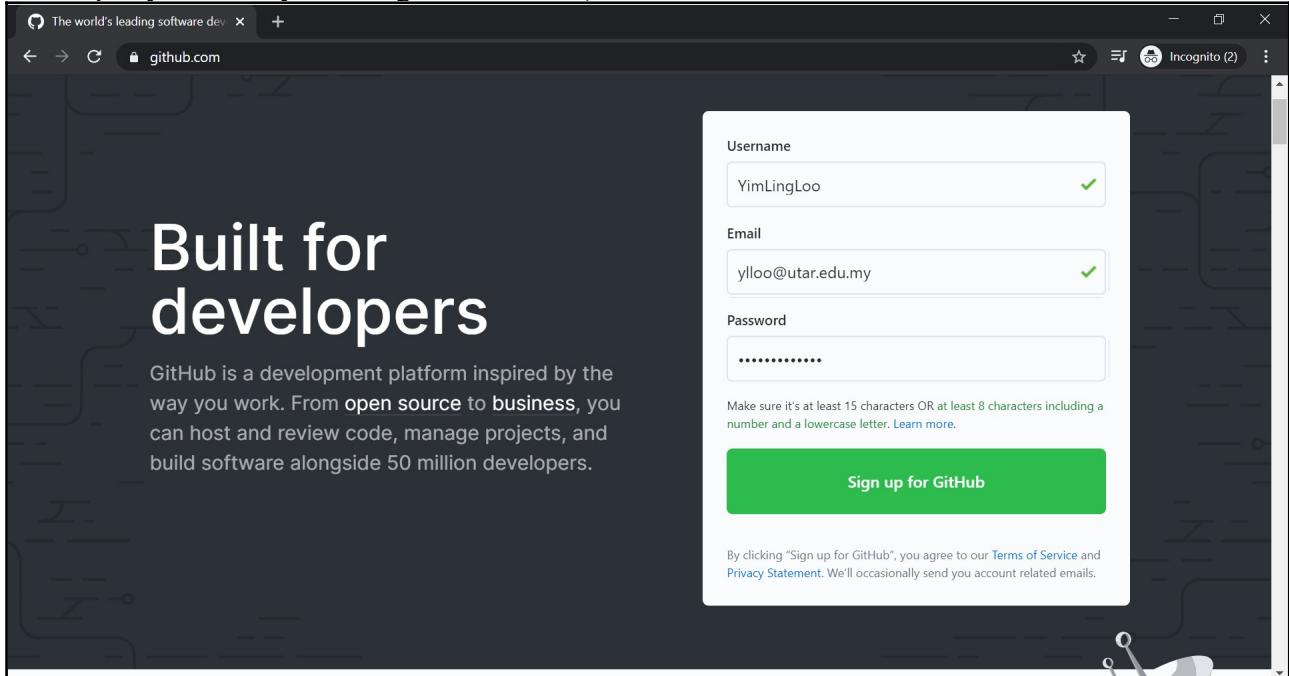


Figure 1: Create a new Github account.

After series of verification to secure the free account, proceed to create a new repository in Github as shown in Figure 2 and 3.

1. Enter the repository name (note that Github doesn't allow spaces for repo name, an automatic 'hyphen' / 'dash' will be added to white spaces)
2. Enter some descriptions for your repo
3. Put your repo 'public'
4. Initialize the repo with a README (which will contain descriptions of the repo)
5. Click on "Create Repository" button

UECS2363 SOFTWARE CONSTRUCTION AND CONFIGURATION

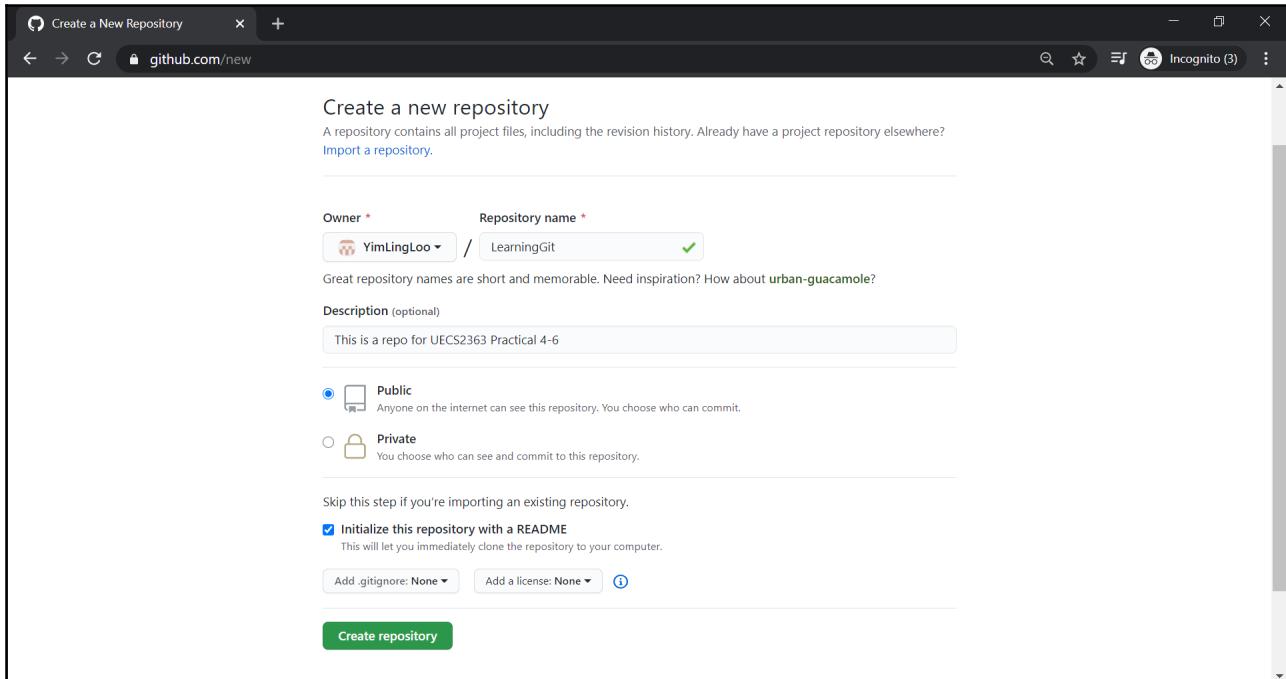


Figure 2: Create a new repo (repository) under your Github account.

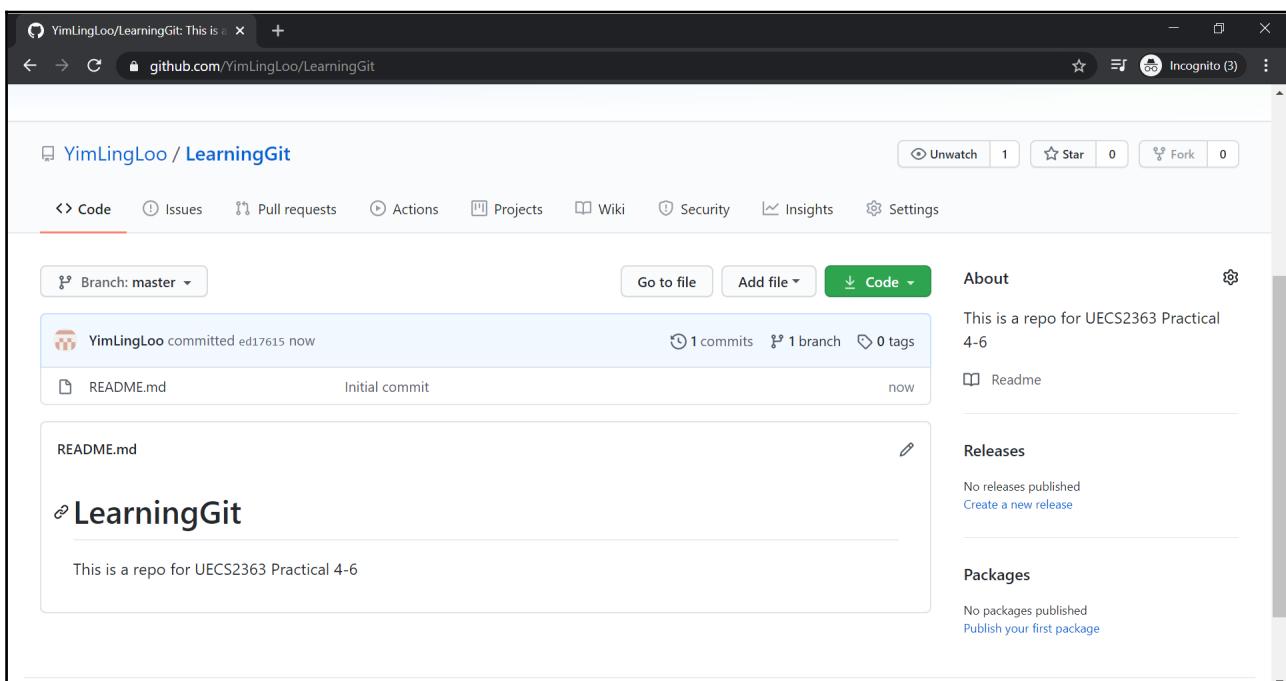


Figure 3: Preview of newly created repository.

UECS2363 SOFTWARE CONSTRUCTION AND CONFIGURATION

Create Commits.

Create a new file that you are going to work with in the repo, using the “Add file” menu as shown in Figure 4.

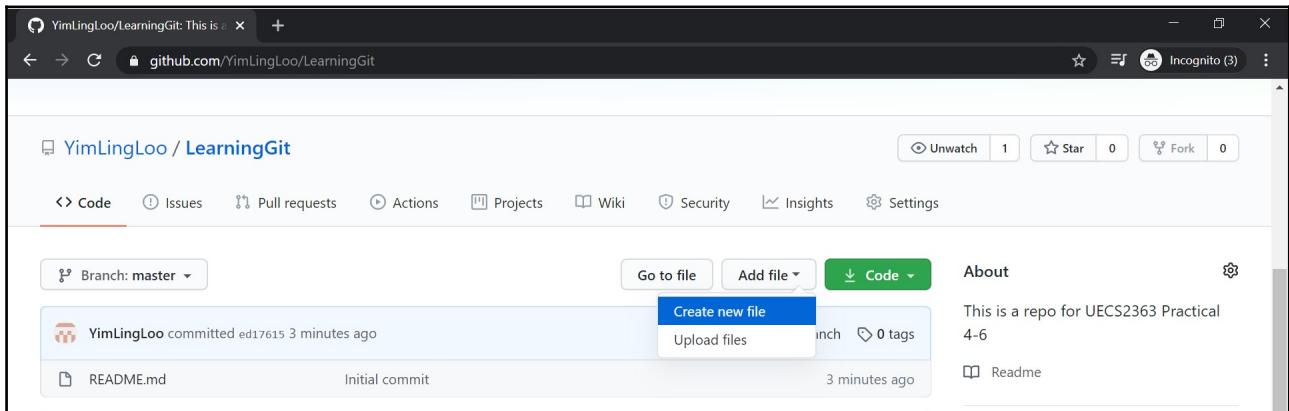


Figure 4: Create a new file in the repo.

Take note that after choosing “Create new file”, you are then redirected to a page which looks as the followings:

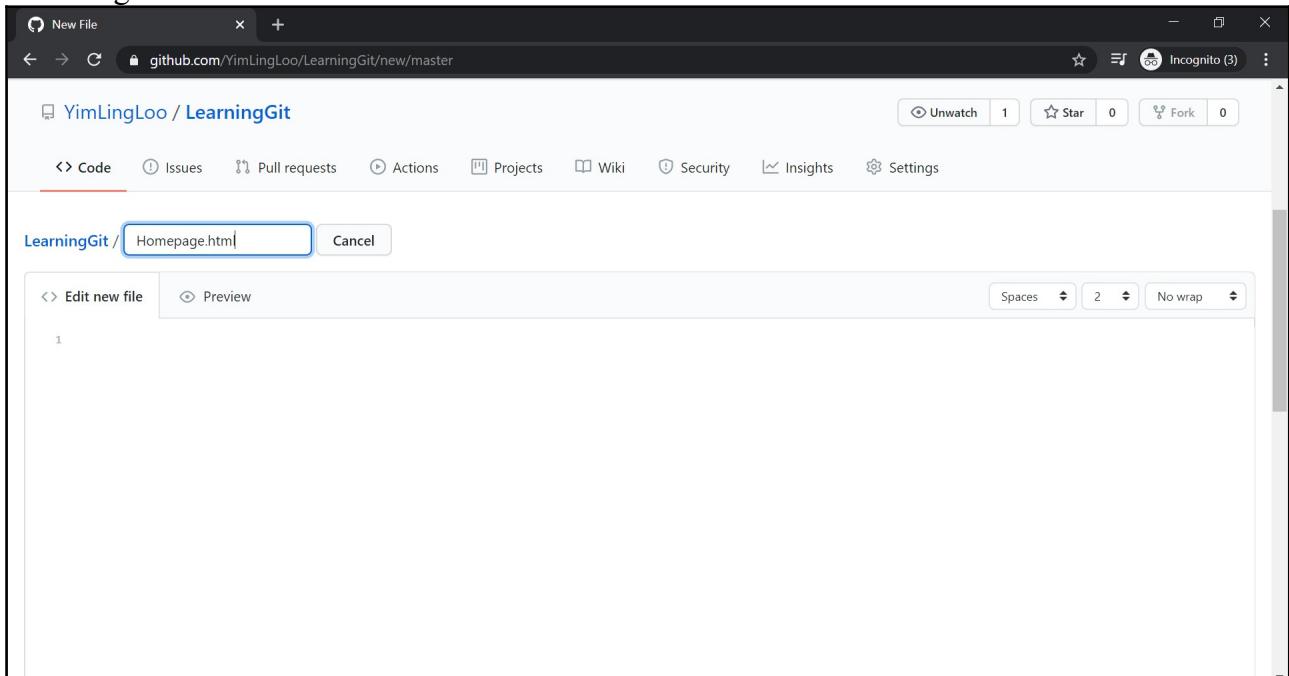


Figure 5: Upper section of “Create a new file” page.

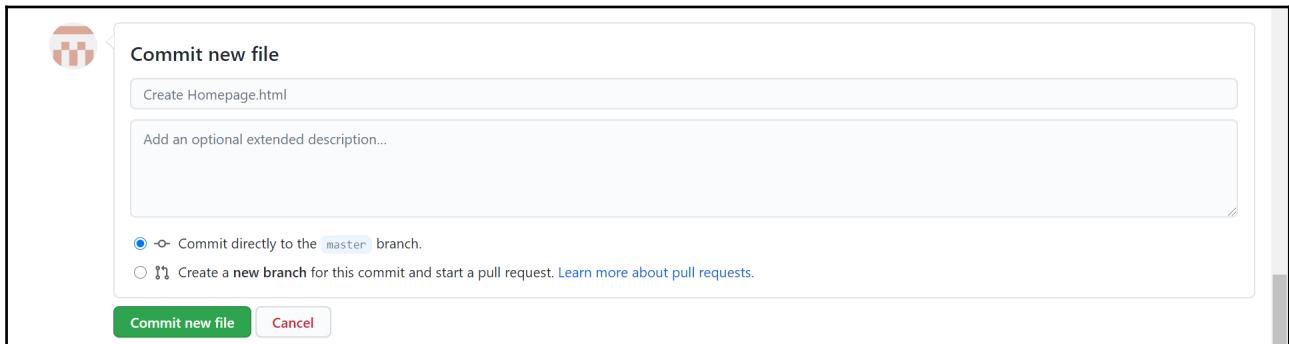
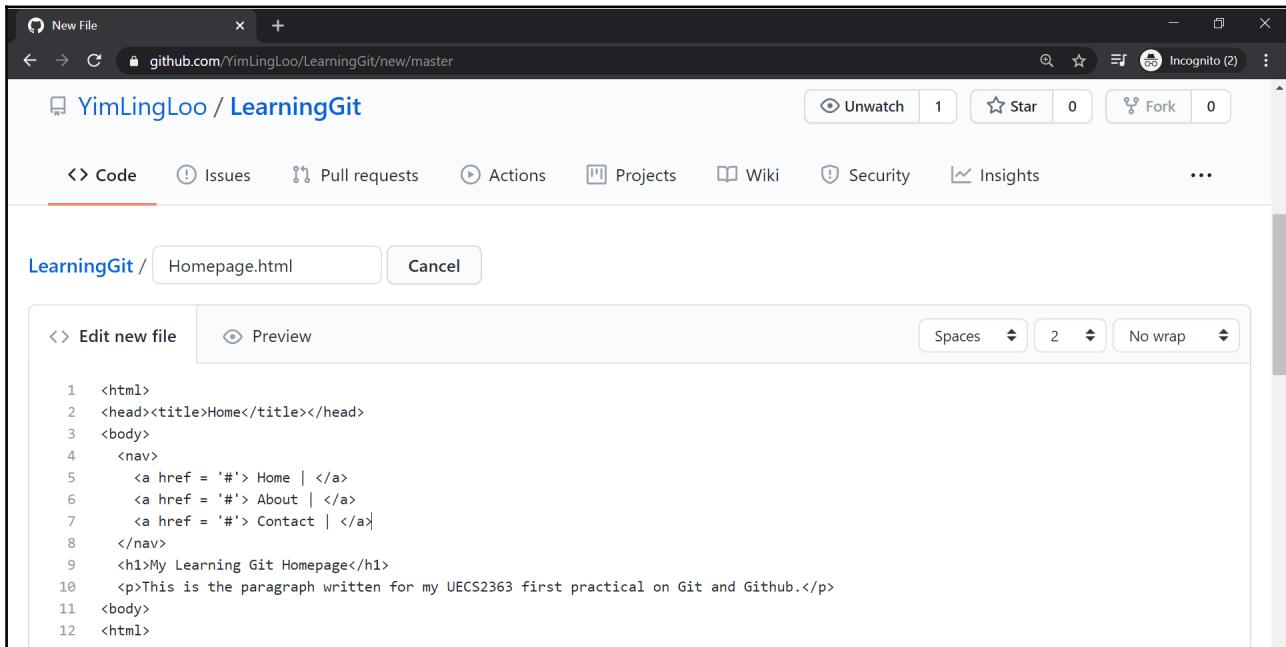


Figure 6: Lower section of “Create a new file” page.

UECS2363 SOFTWARE CONSTRUCTION AND CONFIGURATION

Start working on a new file by entering the filename together with the file type (extension). Git and Github is able to keep track a lot of different file types for you. Then, start putting something (texts/scripts/codes) into the “Edit new file” container as illustrated in Figure 7.



The screenshot shows a GitHub repository named 'YimLingLoo / LearningGit'. In the 'Code' tab, a modal window titled 'Homepage.html' is open, showing the following code:

```
1 <html>
2 <head><title>Home</title></head>
3 <body>
4 <nav>
5 <a href = '#'> Home | </a>
6 <a href = '#'> About | </a>
7 <a href = '#'> Contact | </a>
8 </nav>
9 <h1>My Learning Git Homepage</h1>
10 <p>This is the paragraph written for my UECS2363 first practical on Git and Github.</p>
11 </body>
12 </html>
```

Figure 7: Work on the new file.

After you are satisfied with the works of editing for the new file, make sure to commit the file by clicking the “Commit new file” button at the lower section of the “Create a new file” page as shown in Figure 8.

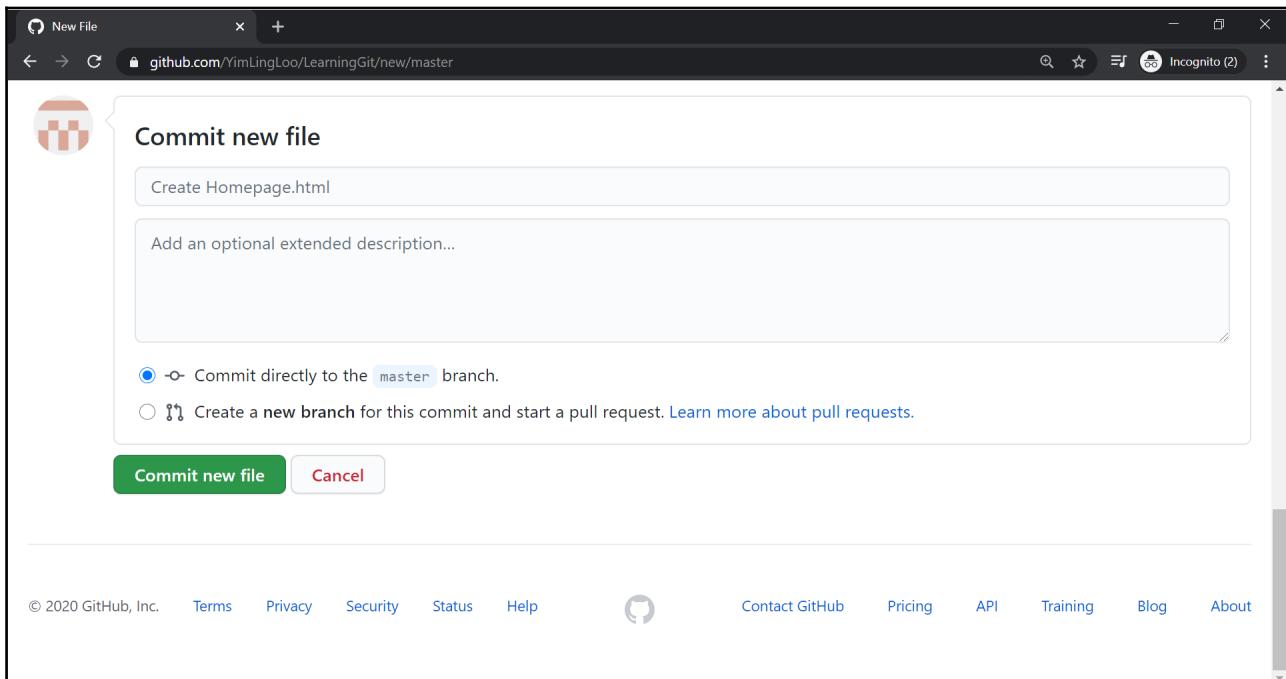


Figure 8: Commit the ‘completed’ new file.

UECS2363 SOFTWARE CONSTRUCTION AND CONFIGURATION

The screenshot shows a GitHub repository page for 'YimLingLoo / LearningGit'. The 'Code' tab is selected. A commit message from 'YimLingLoo' is visible, stating 'committed 1632fc5 now'. Below the commit, there are two files listed: 'Homepage.html' and 'README.md'. The 'Homepage.html' file was created 'now'. The 'README.md' file was committed '33 minutes ago'. On the right side of the page, there are sections for 'About', 'Releases', and 'Packages'. The 'About' section describes the repo as 'This is a repo for UECS2363 Practical 4-6'. The 'Releases' section indicates 'No releases published' and 'Create a new release'. The 'Packages' section indicates 'No packages published'.

Figure 9: The new file is committed to the repo.

When one wants to review the newly created file, simply clicking on the file, the file will be opened up as shown in Figure 10.

The screenshot shows the GitHub file editor for 'Homepage.html'. The file was created by 'YimLingLoo' and committed 1632fc5 19 minutes ago. It has 12 lines (12 sloc) and 301 Bytes. The file content is as follows:

```
1 <html>
2 <head><title>Home</title></head>
3 <body>
4 <nav>
5 <a href = '#'> Home | </a>
6 <a href = '#'> About | </a>
7 <a href = '#'> Contact | </a>
8 </nav>
9 <h1>My Learning Git Homepage</h1>
10 <p>This is the paragraph written for my UECS2363 first practical on Git and Github.</p>
11 </body>
12 </html>
```

Figure 10: The newly created file being opened.

Let's say you spotted the <body> and <html> tags at the end of the file were missing slashes ('/') and wanted to edit the file, click on the pen icon (left isde of the trash can icon) for editing.

Take note that this time, the lower section of the editing page contain a "Commit changes" instead of "Commit new file" as shown in Figure 11.

Do write the changes committed (together with description if you have extended explanation on the intended changes).

Make sure to click on "Commit changes" button to commit your changes.

UECS2363 SOFTWARE CONSTRUCTION AND CONFIGURATION

The screenshot shows a GitHub repository page for 'YimLingLoo / LearningGit'. The user is editing the file 'Homepage.html'. The code editor displays the following HTML:

```
1 <html>
2 <head><title>Home</title></head>
3 <body>
4   <nav>
5     <a href = '#'> Home | </a>
6     <a href = '#'> About | </a>
7     <a href = '#'> Contact | </a>
8   </nav>
9   <h1>My Learning Git Homepage</h1>
10  <p>This is the paragraph written for my UECS2363 first practical on Git and Github.</p>
11 </body>
12 </html>
```

Below the code editor is a 'Commit changes' dialog box. The message field contains: "Adding the '/' to line 11 and 12". The error message field highlights: "<body> and <html> tags were not closed properly at the end of html file". There are two options for committing:

- o- Commit directly to the `master` branch.
- ! Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

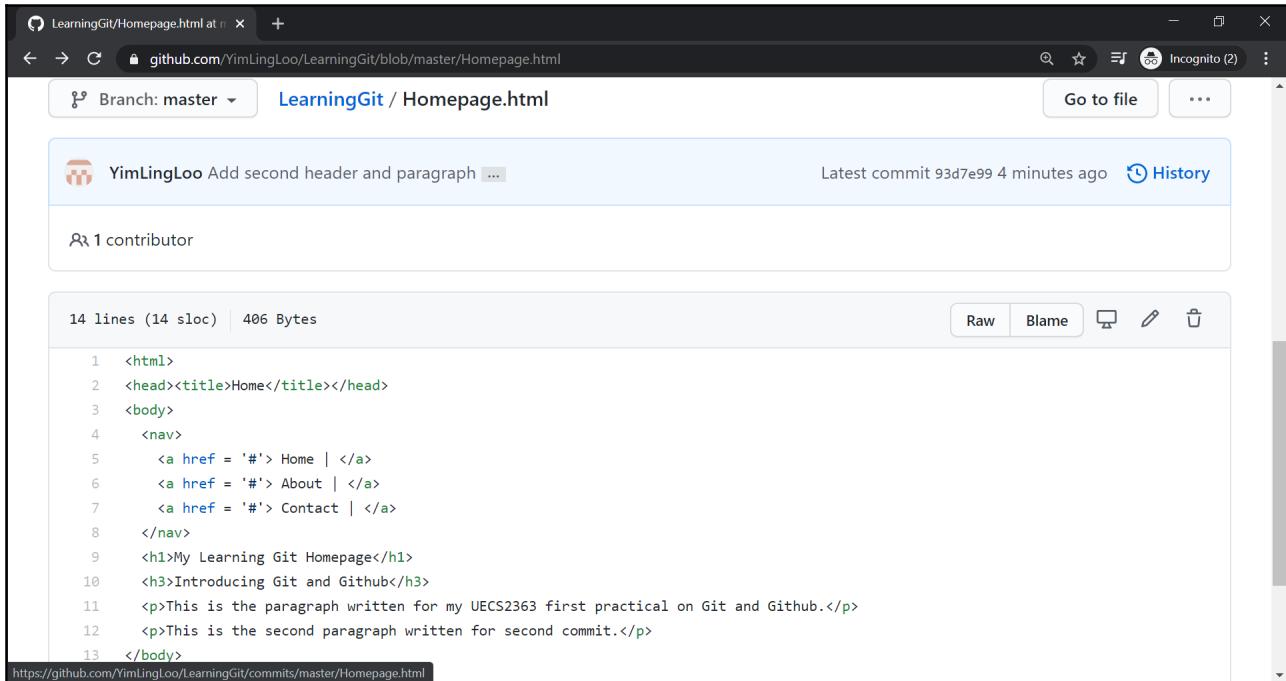
At the bottom of the dialog are 'Commit changes' and 'Cancel' buttons.

At the very bottom of the page, there is a footer with links: © 2020 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub Pricing API Training Blog About.

Figure 11: Commit changes to a previously made file.

Try to commit another change so that now the file has several changes committed instead of just once. After each commit of changes, the changes will be shown in the upper pane of the file preview as shown in Figure 12.

UECS2363 SOFTWARE CONSTRUCTION AND CONFIGURATION

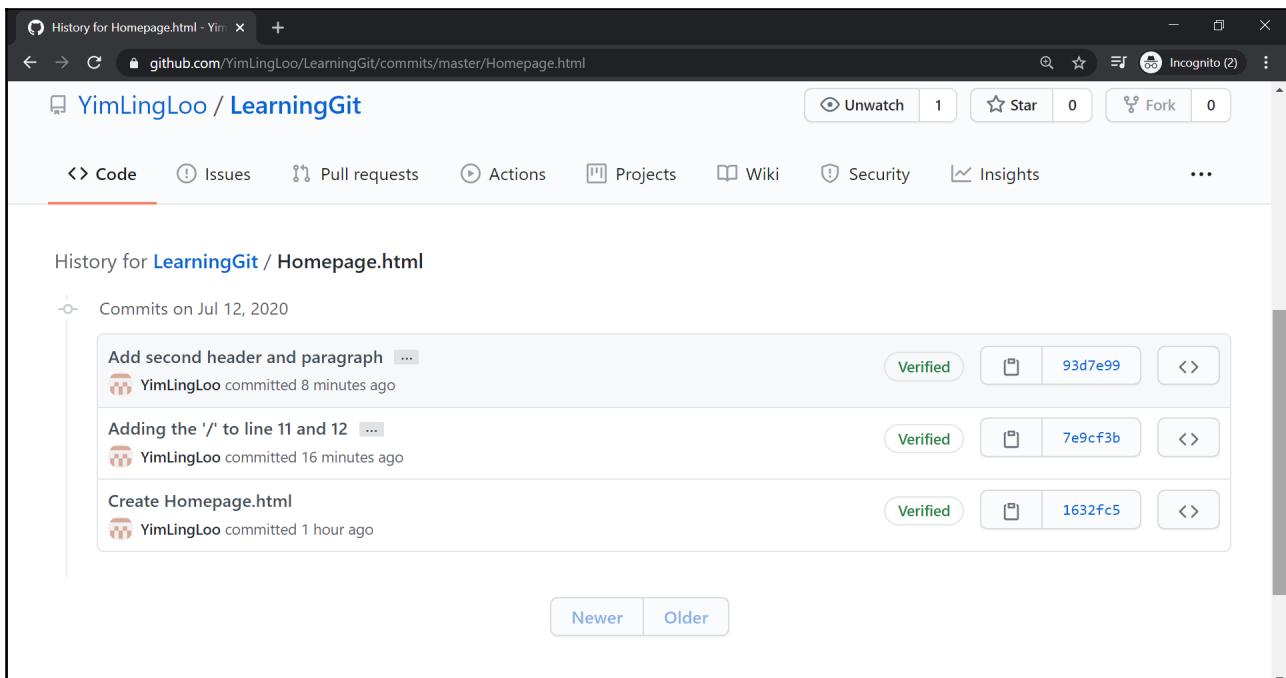


The screenshot shows a GitHub file preview for `Homepage.html`. At the top, there's a header with "Branch: master" and the file name. Below it, a commit message from "YimLingLoo" is shown: "Add second header and paragraph ...". To the right, there's a link to "History" and a note about the latest commit being 4 minutes ago. It says "1 contributor". The main area displays the HTML code with line numbers 1 through 13. The code includes a navigation bar with links to Home, About, and Contact, followed by an

tag and two tags. There are also two tags containing descriptive text. At the bottom, there are "Raw", "Blame", and "Edit" buttons, along with a copy icon. The URL at the bottom is <https://github.com/YimLingLoo/LearningGit/commits/master/Homepage.html>.

Figure 12: Changes shown on the upper pane of the file preview.

The versions can be viewed by clicking the “History” link on the upper pane of the file preview as illustrated in Figure 12.



The screenshot shows the commit history for `Homepage.html`. At the top, there's a header with "History for Homepage.html - YimLingLoo / LearningGit". Below it, there are buttons for "Code", "Issues", "Pull requests", "Actions", "Projects", "Wiki", "Security", and "Insights". The "Code" button is highlighted with a red underline. The main area is titled "History for `LearningGit / Homepage.html`". It shows three commits made on July 12, 2020:

- "Add second header and paragraph ..." by YimLingLoo committed 8 minutes ago. Status: Verified. SHA: 93d7e99.
- "Adding the '/' to line 11 and 12 ..." by YimLingLoo committed 16 minutes ago. Status: Verified. SHA: 7e9cf3b.
- "Create Homepage.html" by YimLingLoo committed 1 hour ago. Status: Verified. SHA: 1632fc5.

At the bottom, there are "Newer" and "Older" buttons.

Figure 13: History/versions of the file listed on the upper pane of the file preview.

Choose one of the changes committed and see how Github preview the commits made as illustrated in Figure 14.

UECS2363 SOFTWARE CONSTRUCTION AND CONFIGURATION

The screenshot shows a GitHub commit page for a file named "Homepage.html". The commit title is "Adding the '/' to line 11 and 12". The commit message states: "<body> and <html> tags were not closed properly at the end of html file". The commit was made by "YimLingLoo" 27 minutes ago and is verified. It has 1 parent commit (1632fc5) and a commit hash of 7e9cf3bca36a07fa11fa67e3ad8e6b47ca2688ed. The diff view shows the following changes:

```
@@ -8,5 +8,5 @@
8     8         </nav>
9     9         <h1>My Learning Git Homepage</h1>
10    10        <p>This is the paragraph written for my UECS2363 first practical on Git and Github.</p>
11    - <body>
12    - <html>
11    + </body>
12    + </html>
```

Figure 14: Highlighted changes in green and red with commit hash.

The changes were highlighted in green and the old script before changes were made was shown in red highlights. Commits are identified uniquely by Github with a “crazy” string of numerics and alphabets which are called commit “hash”.

UECS2363 SOFTWARE CONSTRUCTION AND CONFIGURATION

Create Branches.

There is a “master” branch option when we first created the new file. At this section, we will look into the “branches” in Git/Github.

In commits of changes, there are master changes that are sure to be made and kept by a developer. These are always called “master”. However, there are also instances or changes that are experimental and uncertain to be kept by a developer. These commits are “branches”.

A large software project development may consist of many aspects. One can work on “branches” of commits on top of the “master” commits, to work on different aspects of the software project.

Let’s try to create a branch with an experimental idea for the branch by clicking on the “Tree” dropdown menu and type in the branch name as shown in Figure 15.

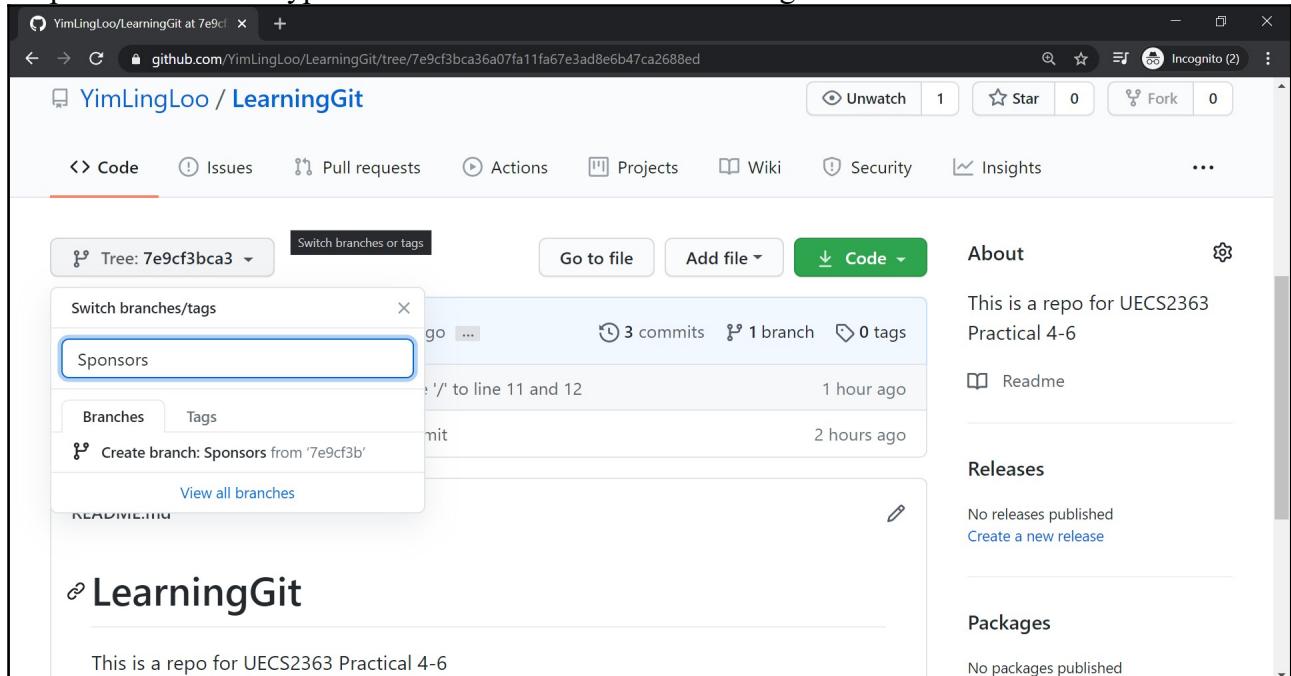


Figure 15: Create a branch.

Press “Enter” button after input of the name of branch and Github will switch to the newly created branch as shown in Figure 16.

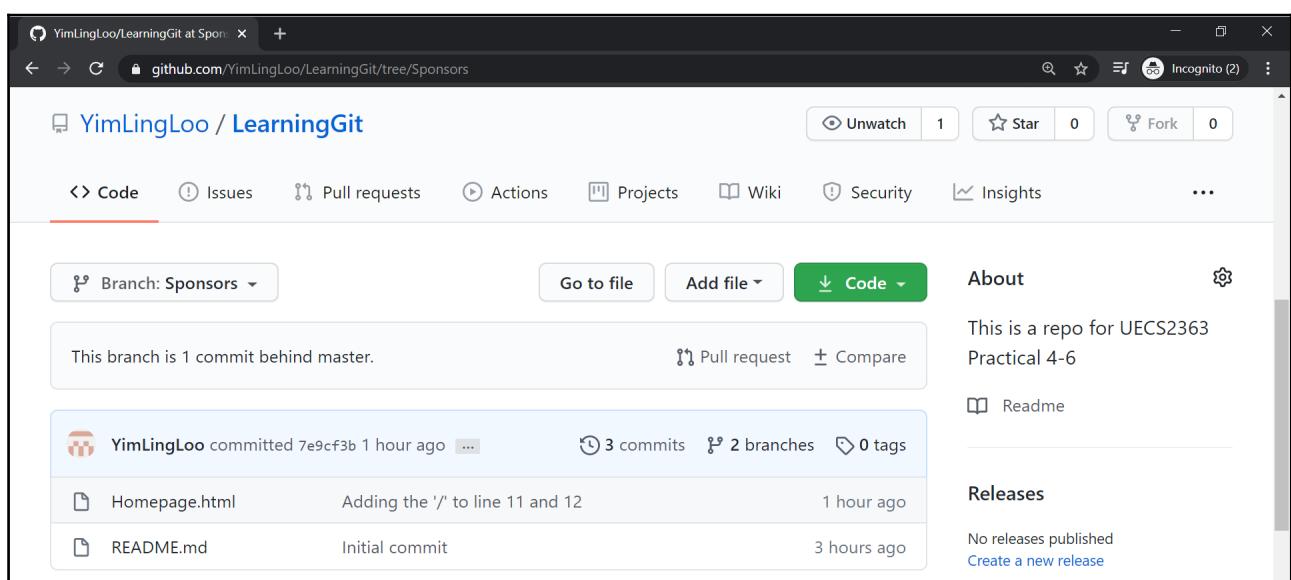


Figure 16: Switch to Sponsors branch of the ‘main tree’.

UECS2363 SOFTWARE CONSTRUCTION AND CONFIGURATION

While in branch Sponsors, select Homepage.html and choose “Edit” to edit the file as illustrated in Figure 17.

The screenshot shows a GitHub repository named 'YimLingLoo / LearningGit'. The 'Code' tab is selected. A modal window is open for editing the file 'Homepage.html'. The code editor contains the following HTML:

```
1 <html>
2 <head><title>Home</title></head>
3 <body>
4   <nav>
5     <a href = '#'> Home | </a>
6     <a href = '#'> About | </a>
7     <a href = '#'> Contact | </a>
8     <a href = '#'> Sponsors </a>
9   </nav>
10  <h1>My Learning Git Homepage</h1>
11  <p>This is the paragraph written for my UECS2363 first practical on Git and Github.</p>
12  <p>You may make a sponsor to this project by navigating to Sponsors page in the navigation pane.</p>
13 </body>
14 </html>
```

Below the code editor is a 'Commit changes' section. It contains a note 'Added sponsors' and a descriptive text: 'Optional idea for sponsors to be able to contribute their donations on the website.' There are two radio button options for committing:

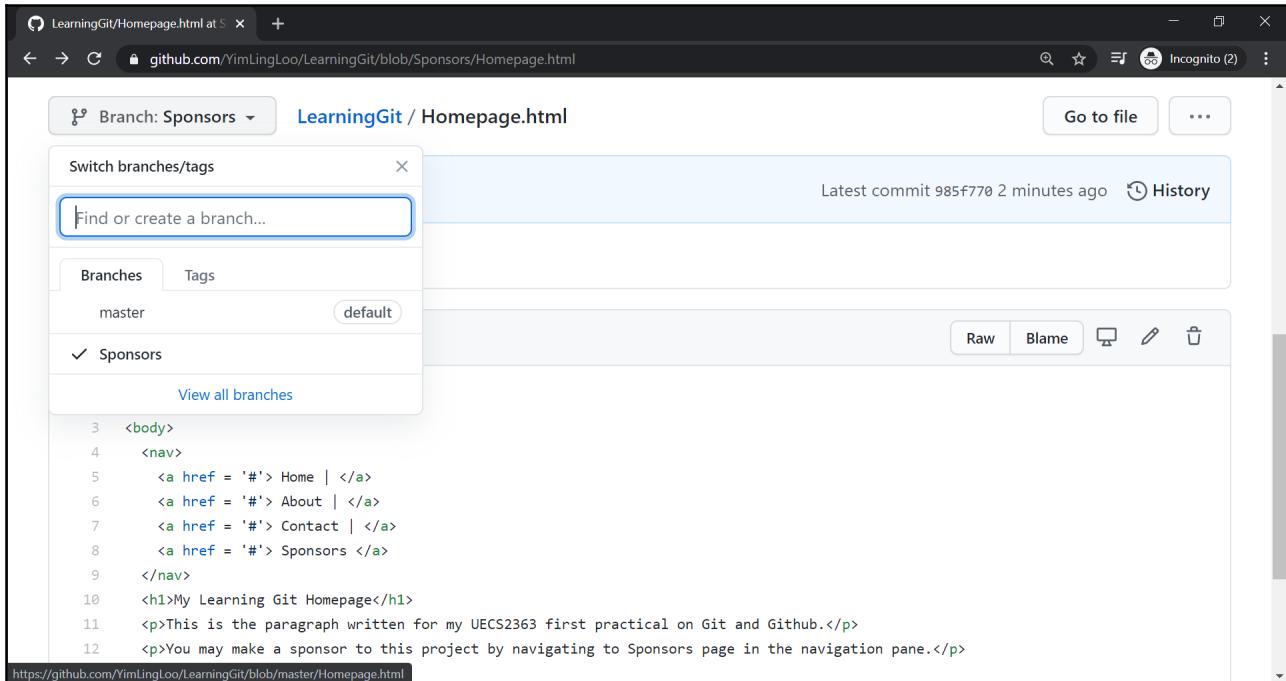
- o- Commit directly to the [Sponsors](#) branch.
- ⚡ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

At the bottom of the commit section are 'Commit changes' and 'Cancel' buttons.

Figure 17: Commit a change to Homepage.html in Sponsors branch.

Try to switch between the Sponsors branch and master to see that you have different versions of Homepage.html by the “Branch” dropdown menu as shown in Figure 18.

UECS2363 SOFTWARE CONSTRUCTION AND CONFIGURATION



The screenshot shows a GitHub repository page for 'LearningGit'. The top navigation bar has a dropdown 'Branch: Sponsors'. The main content area shows the 'Homepage.html' file with the following code:

```
3 <body>
4   <nav>
5     <a href = '#'> Home | </a>
6     <a href = '#'> About | </a>
7     <a href = '#'> Contact | </a>
8     <a href = '#'> Sponsors </a>
9   </nav>
10  <h1>My Learning Git Homepage</h1>
11  <p>This is the paragraph written for my UECS2363 first practical on Git and Github.</p>
12  <p>You may make a sponsor to this project by navigating to Sponsors page in the navigation pane.</p>
```

The URL in the address bar is <https://github.com/YimLingLoo/LearningGit/blob/Sponsors/Homepage.html>.

Figure 18: Switch between master and branches.

A developer may keep track of branches that this software project has and visualize the software project development, by choosing “Insights” on the top navigation pane, then choose “Network” which is located on the left navigation pane as illustrated in Figure 19.

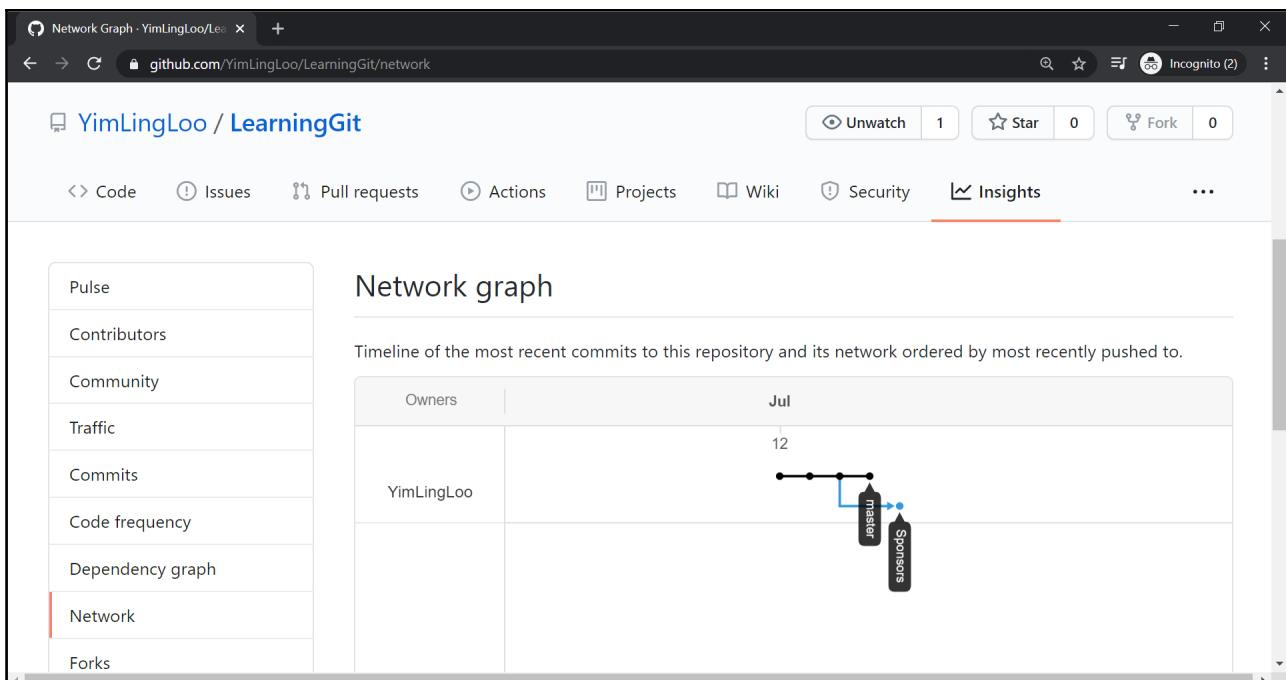


Figure 19: Visualization of master and branches.

Q: What if the developer finds that the idea of having sponsors to contribute straight to the software development using Sponsors page is indeed great?

UECS2363 SOFTWARE CONSTRUCTION AND CONFIGURATION

Pull Request and Merge.

An answer to the question of having the experimental idea from a branch to be made a main idea in the master is that the branch “request” for a “merge” to the master. A “Pull Request” can be made by clicking on the “Compare & pull request” button as shown in Figure 20.

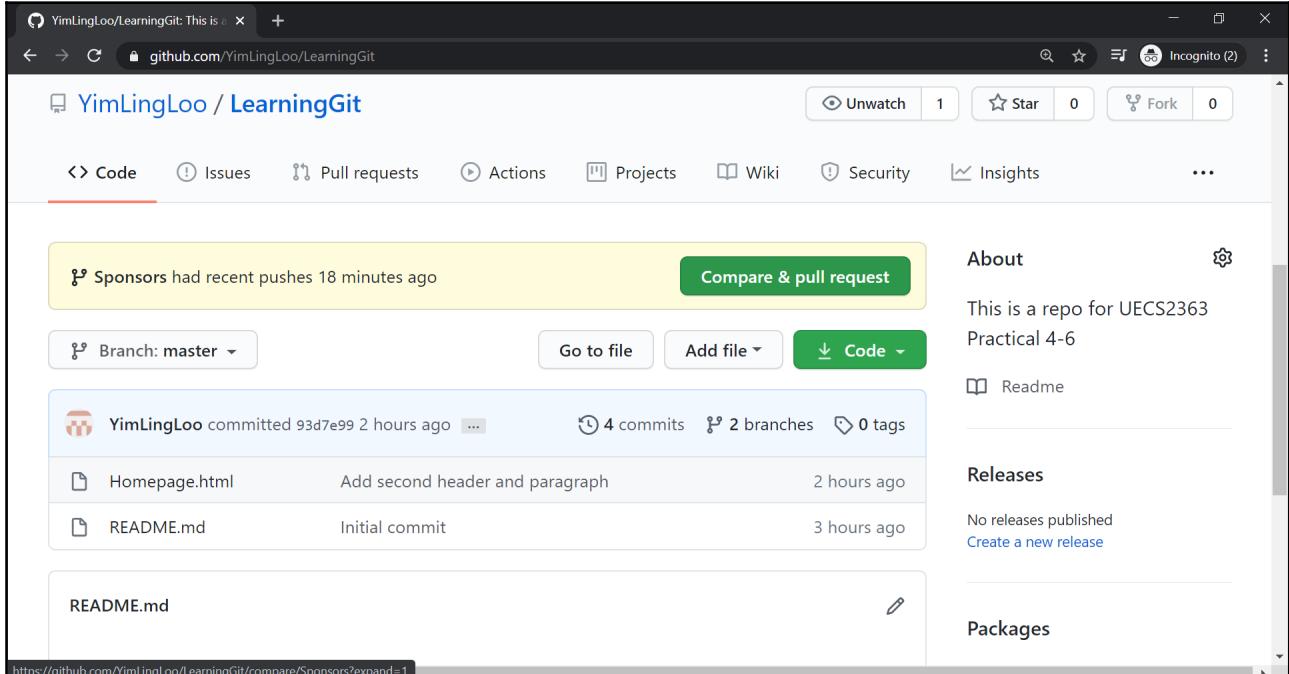


Figure 20: Create pull request to pull commits from a branch into master.

Developer will be directed to the pul request page with the result of Github analysis mentioning whether a merge can be automatically committed. Even if it cannot be automatically committed, draft the pull request by inputs of descriptions and click on “Create pull request” as shown in Figure 21.

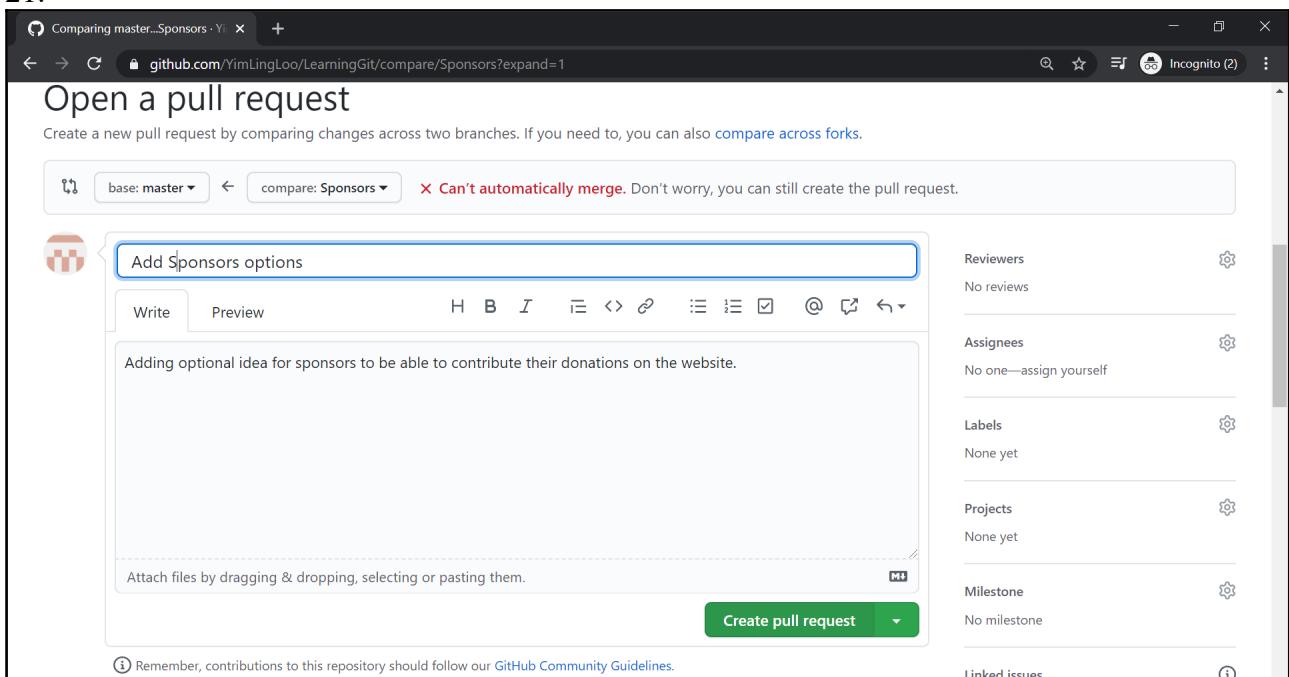


Figure 21: Drafting pull request to pull commits from a branch into master.

UECS2363 SOFTWARE CONSTRUCTION AND CONFIGURATION

Github is powerful enough to trace if there is a sense of conflict to merge between two commits. When there is a conflicting merge, one can easily look into the traces of conflicts and try to resolve by clicking on “Resolve Conflicts” button as shown in Figure 22 and following the steps as shown in Figure 23 to 26.

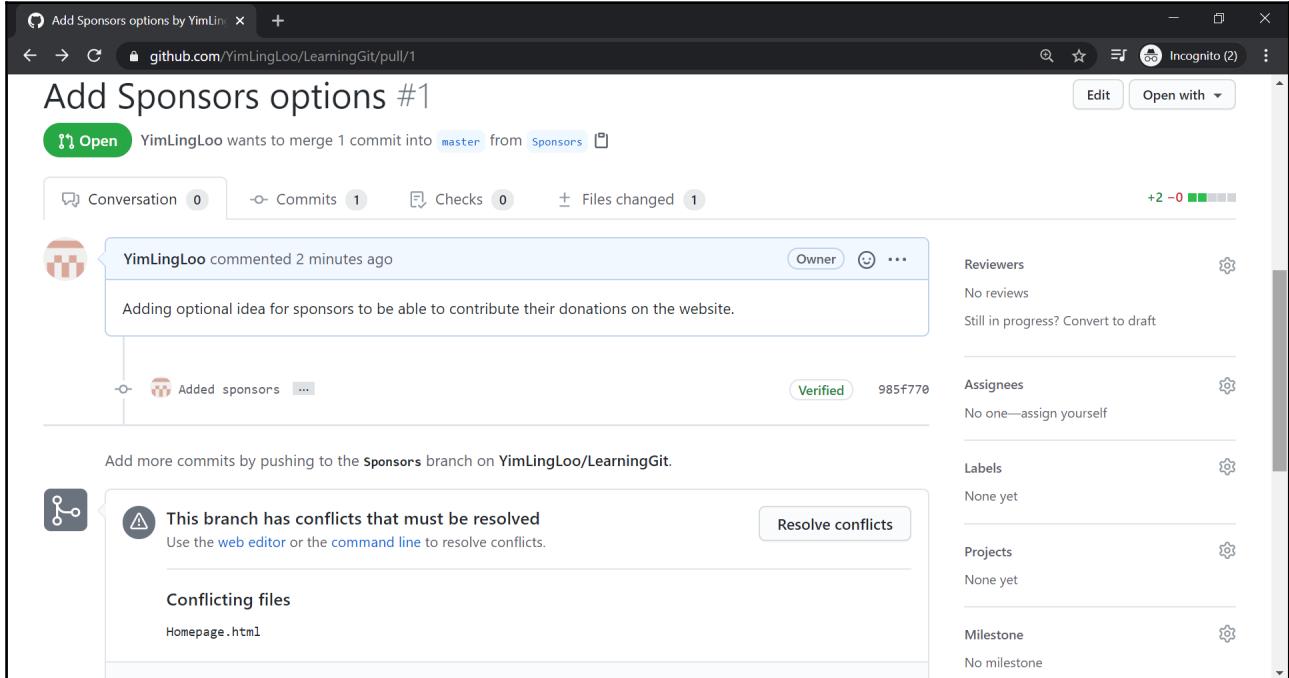


Figure 22: Occurrence of conflicting merge.

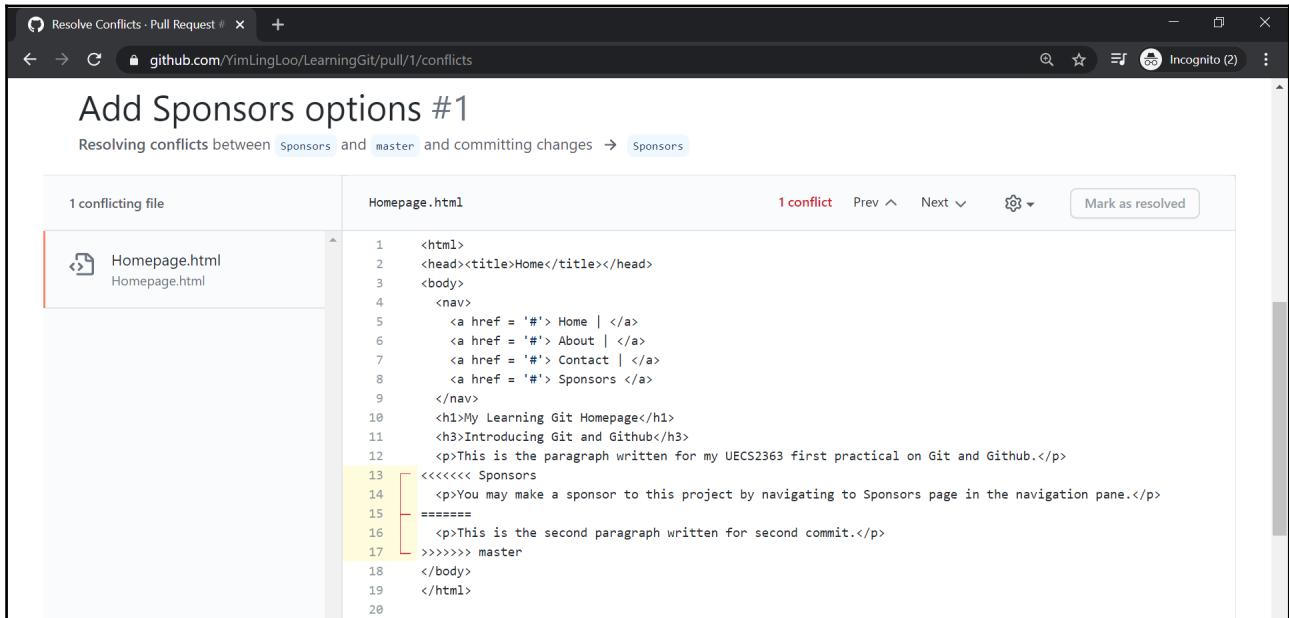


Figure 23: Labels of conflict by Github.

UECS2363 SOFTWARE CONSTRUCTION AND CONFIGURATION

```
<html>
<head><title>Home</title></head>
<body>
<nav>
<a href = '#'> Home | </a>
<a href = '#'> About | </a>
<a href = '#'> Contact | </a>
<a href = '#'> Sponsors </a>
</nav>
<h1>My Learning Git Homepage</h1>
<h3>Introducing Git and Github</h3>
<p>This is the paragraph written for my UECS2363 first practical on Git and Github.</p>
<p>This is the second paragraph written for second commit.</p>
<p>You may make a sponsor to this project by navigating to Sponsors page in the navigation pane.</p>
</body>
</html>
```

Figure 24: Edit the labeled conflicts and click “Mark as resolved”.

```
<html>
<head><title>Home</title></head>
<body>
<nav>
<a href = '#'> Home | </a>
<a href = '#'> About | </a>
<a href = '#'> Contact | </a>
<a href = '#'> Sponsors </a>
</nav>
<h1>My Learning Git Homepage</h1>
<h3>Introducing Git and Github</h3>
<p>This is the paragraph written for my UECS2363 first practical on Git and Github.</p>
<p>This is the second paragraph written for second commit.</p>
<p>You may make a sponsor to this project by navigating to Sponsors page in the navigation pane.</p>
</body>
</html>
```

Figure 25: Click on “Commit merge” to finalize the Pull Request.

UECS2363 SOFTWARE CONSTRUCTION AND CONFIGURATION

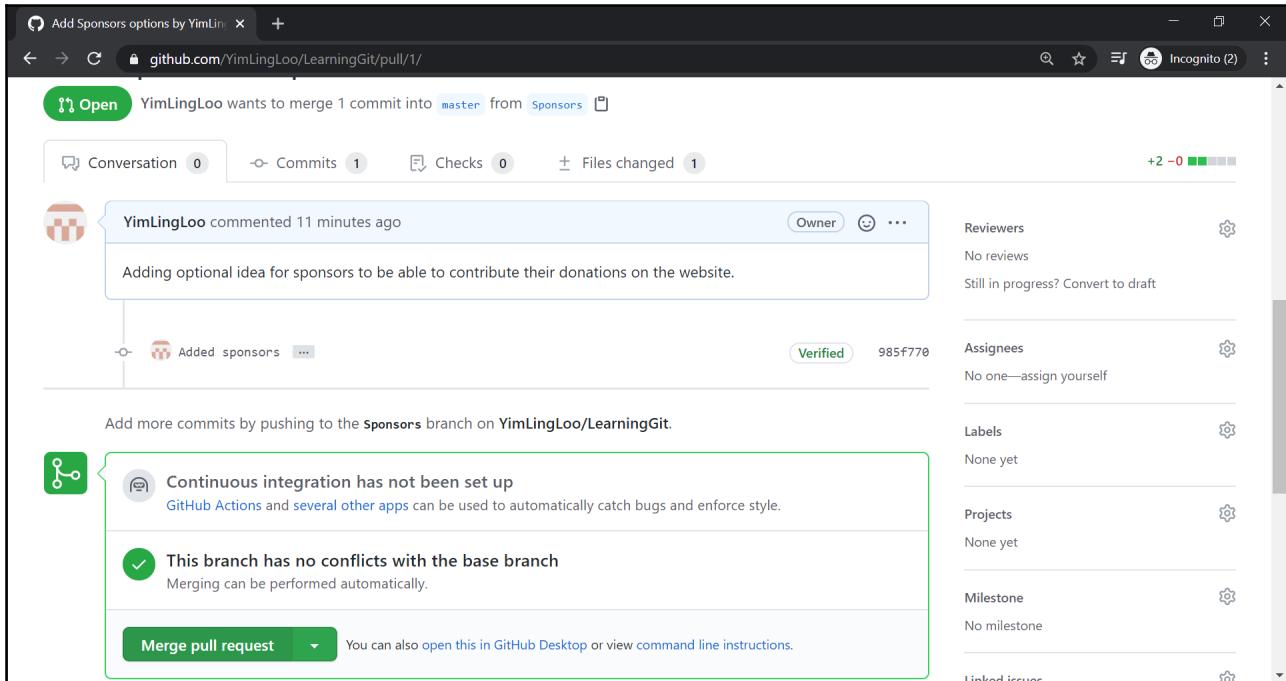


Figure 26: Click on “Merge pull request” and “Confirm merge” to finalize the Merge.

Upon finalizing a merge, the Pull Requests now will have a log of successful merge that had been performed between Sponsors branch and Master as shown in Figure 27.

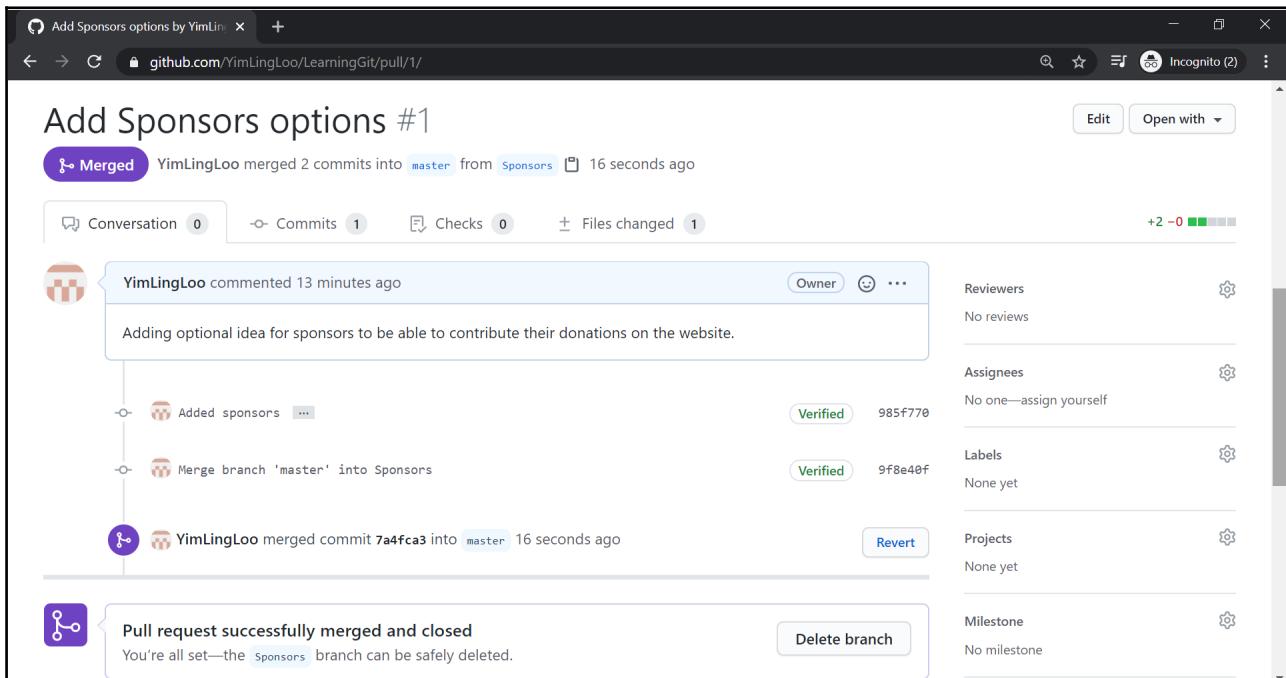


Figure 27: A log of Merge done between branch Sponsor and Master.

Q: Go back to the repo’s Insight and Network to visualize the tree of the project, what do you see?

Now, let’s say you are another developer who happened to stumble upon this software project and have an interest in it. However, you do not wish to commit new files or changes to this project in the original repo. You can either choose to “clone” the repo (which will be discussed in another practical session) OR fork the repo so that you may work on the project by having the entire repo in your account.

UECS2363 SOFTWARE CONSTRUCTION AND CONFIGURATION

Create Fork and Pull Request.

Try to search for YimLingLoo/LearningGit repo in your own account as shown in Figure 28.

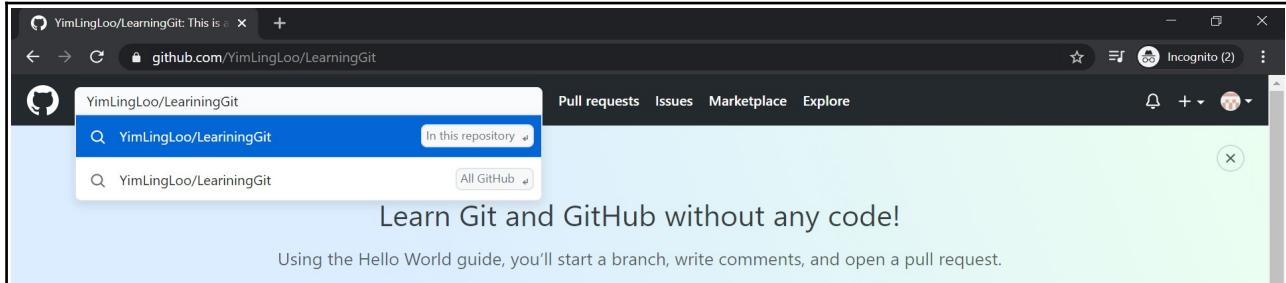


Figure 28: Search for a repository in Github.

The repo will be displayed on your account and “Fork” option is available on the top right corner of the repo preview as shown in Figure 29.

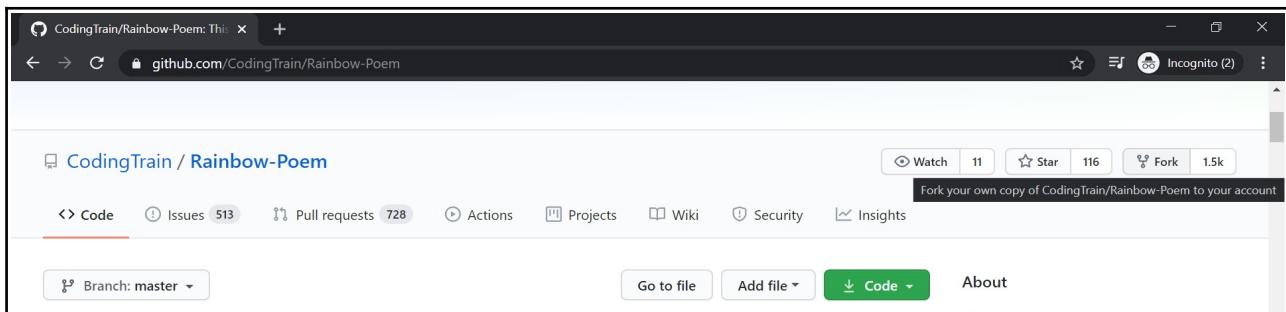


Figure 29: Selecting “Fork” option on Github.

Github takes over the forking of whole repository into your account and displays it right after forking is finished as shown in Figure 30.

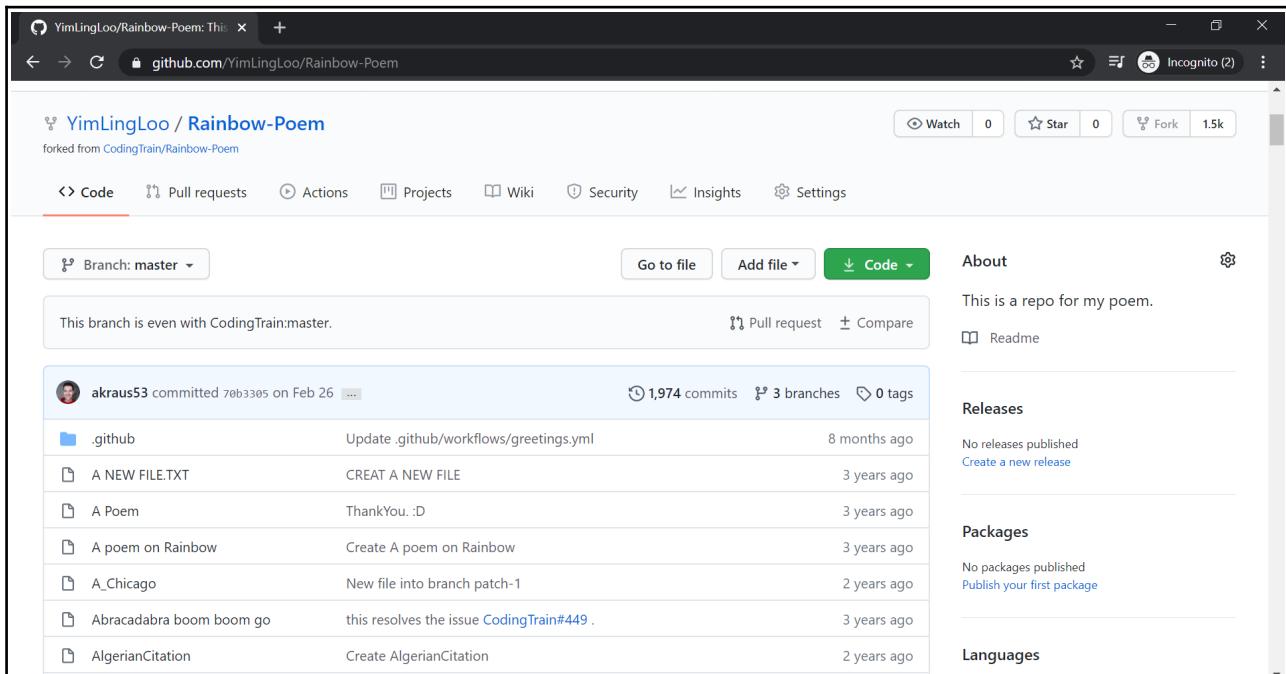


Figure 30: Repo is forked from another account into own account.

UECS2363 SOFTWARE CONSTRUCTION AND CONFIGURATION

From the newly forked repo, a developer may keep track even from where the repo is forked. Now you may repeat the steps of committing new file or a change to a file or even creating new branches on the forked repo.

Now, let's say, after committing numerous changes, you would like to contribute the changes to the original repo, a "Pull Request" need to be made. The pull request can be made by clicking "Pull request" which is available on the left of "Compare" below the "Add file" button as shown in Figure 31.

The screenshot shows a GitHub fork repository page for 'YimLingLoo / Rainbow-Poem'. The repository is forked from 'CodingTrain/Rainbow-Poem'. The main navigation bar includes 'Code', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. A green 'Code' button is highlighted. Below the navigation, a message says 'This branch is 1 commit ahead of CodingTrain:master.' with a 'Pull request' button and a 'Compare' link. The commit list shows several recent commits by 'YimLingLoo', such as 'committed 6f53b34 1 minute ago' and 'Update .github/workflows/greetings.yml'. To the right, sections for 'About', 'Releases', 'Packages', and 'Languages' are visible. The URL in the address bar is <https://github.com/YimLingLoo/Rainbow-Poem/pull/new/master>.

Figure 31: Click on "Pull request" to request a "pull" of changes to the original repo.

Then Github will redirect to the page where comparison is done and analysis of the possibility for merge as shown in Figure 32.

The screenshot shows a GitHub compare page titled 'Comparing CodingTrain:master...'. It lists the base repository as 'CodingTrain/Rainbow-Poem' and the head repository as 'YimLingLoo/Rainbow-Poem'. The compare branch is set to 'master'. A green 'Create pull request' button is prominent. The comparison summary shows '1 commit', '1 file changed', '0 commit comments', and '1 contributor'. The commit details show a single commit by 'YimLingLoo' on Jul 13, 2020, titled 'Create YimLingLooFile.txt'. The commit is verified and has a SHA of '6f53b34'. The diff view shows a single addition to 'YimLingLooFile.txt' with the content '+ This is my file after forking from another repo.'. There are 'Unified' and 'Split' options at the bottom of the diff view. The URL in the address bar is <https://github.com/CodingTrain/Rainbow-Poem/compare/master...YimLingLoo:master>.

Figure 32: Creating pull request to original repo.

UECS2363 SOFTWARE CONSTRUCTION AND CONFIGURATION

Take note that Github gives a full illustration of the pull request; zooming in Figure 32 and illustrated in Figure 33.



Figure 33: The pull request from forked repo into original repo.

Once you confirm the pull request, you just need to wait for the original author of the original repo to confirm merge. Then your contribution is accepted and changes are made in the original repo.

Summary

1. Git is a **version control software** that helps developer to keep track of the different versions of software project files created and edited. Git enables collaboration of many developers on a software project development.
2. Github is a web server with a website that runs Git software on the platform to enable developers to work on a software project and keep track of the versions.
3. **Repository** of files can be made for a software project in Github and made “public” so that other developers can **collaborate** in the development process.
4. “**Commit**” a file or a change in Git/Github means “save” a file or a change.
5. Every commits made in Github has a unique identifier which is called “hash”.
6. Developers may work on **branches** of commits if there are experimental changes/ideas or different aspects of the software to be worked on.
7. When experimental idea becomes interesting and useful to be included in the master, “**Pull Request**” and “**Merge**” is to be made.
8. Collaboration or contribution to another developer’s software project can be made by creating a “**Fork**” of the repo in own account. Then, commit preferred changes and request “**Pull Request**” to **merge** the commits into **original repo**.