

Homework 2: Homework Page on GitHub Pages

Students need to establish a directory / folder in which they can store web pages for the following:

1. A table of hyperlinks to homework assignments (this homework assignment #2)
2. A storage area for holding web resources for homework assignment #3.

This semester we will be using a private repository in GitHub Pages to store your Homework #2 (table of exercises) and #3. To use **GitHub Pages**, you must have a GitHub account. This note lists all the steps to set up a GitHub account and the additional steps to create the GitHub repository and your pages used in Homework #2 and #3.

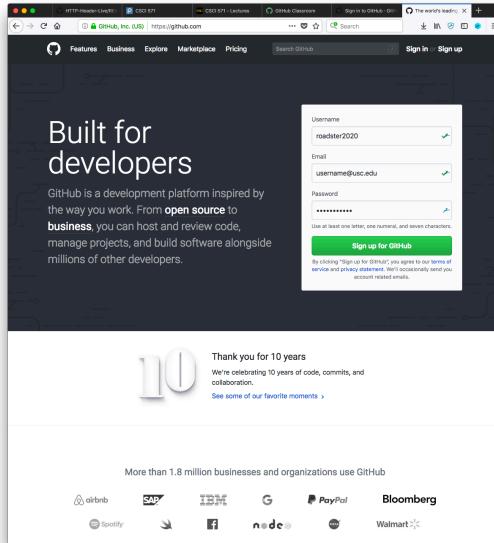
1. Sign up for GitHub

Many students already have set up a personal GitHub account. The standard repositories of a “free” GitHub account are “public” and therefore cannot be used to store “private” pages. Nicely, GitHub Pages allow students that have signed up for the GitHub Student Developer Pack to create a private repository containing your web site.

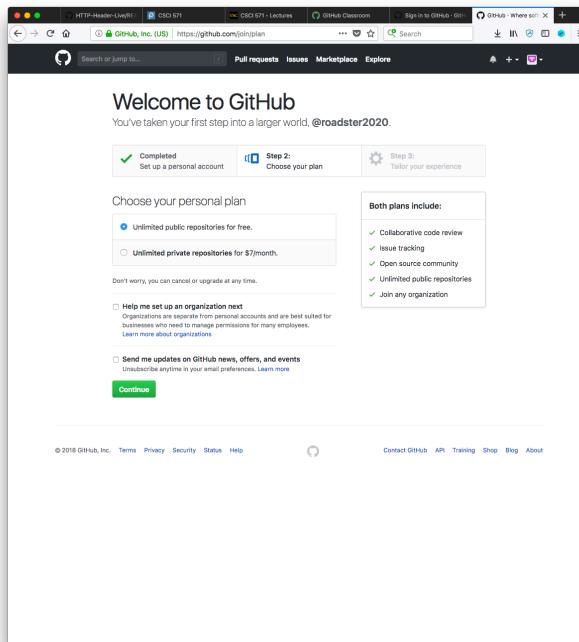
If you already have a GitHub account, you can jump to section 2, “**Sign up for GitHub Student Develop Pack**”. Otherwise, we actually recommend you sign up for a “new” GitHub account, by going to:

<https://github.com>

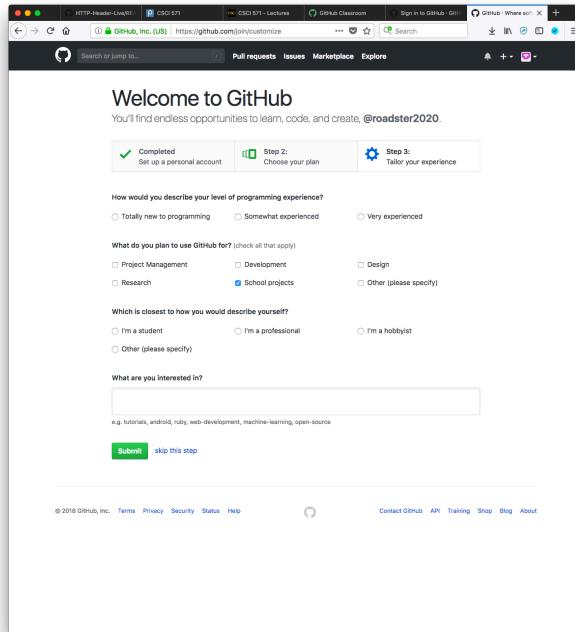
Click **Sign up** in to top right of the page, and the following dialog is displayed.



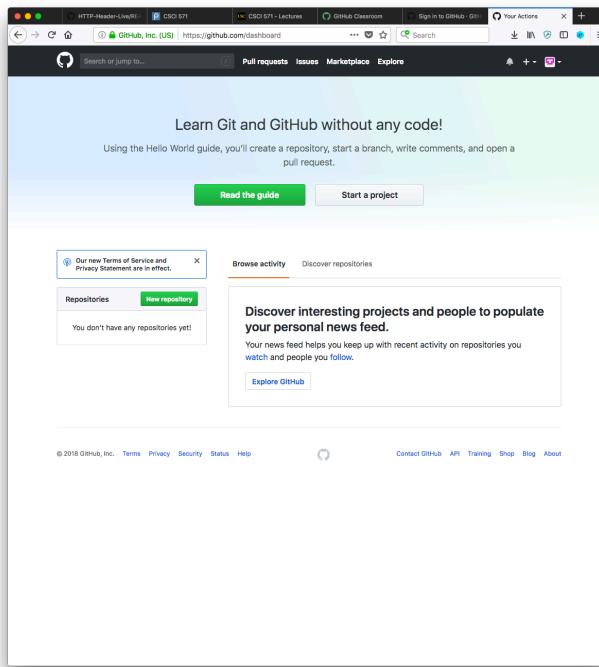
Pick a **username**, enter your username@usc.edu as the **Email** address, and create a **Password**. Then click on **Sign up for GitHub**. The **Welcome to GitHub** page is displayed.



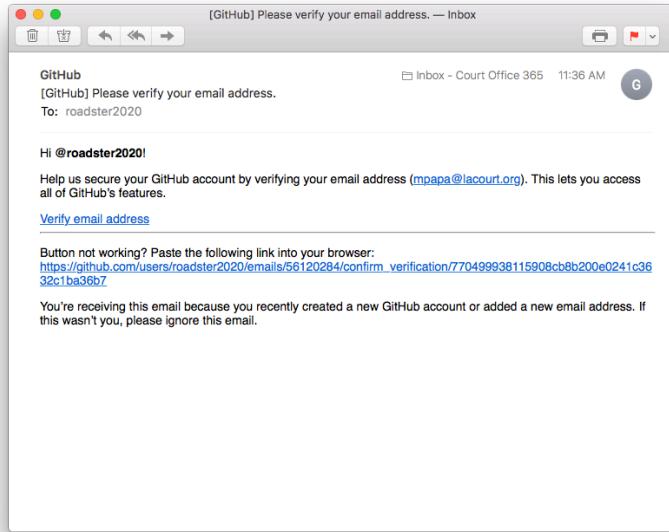
In **Step 2: Choose you plan** leave the **personal plan** selection to “**unlimited public**” and click on **Continue**.



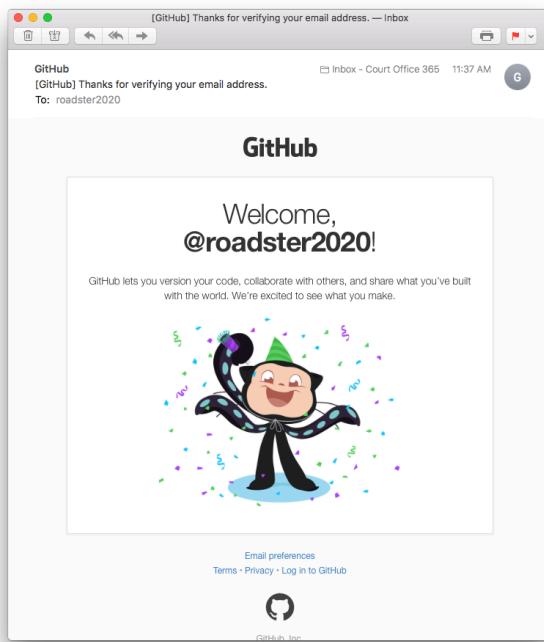
In Step 3: Tailor your experience, select School projects and click on Submit. You will be redirected to the GitHub dashboard.



Feel free to **Read the guide** and **Explore GitHub**. You should be receiving an e-mail asking you to verify your e-mail address.



Click on the **Verify email address** hyperlink. Once your e-mail address is verified, you will receive an additional **Welcome** e-mail. If you get this, your GitHub account is properly set up. Notice the email Welcome will indicate your GitHub ‘username’.

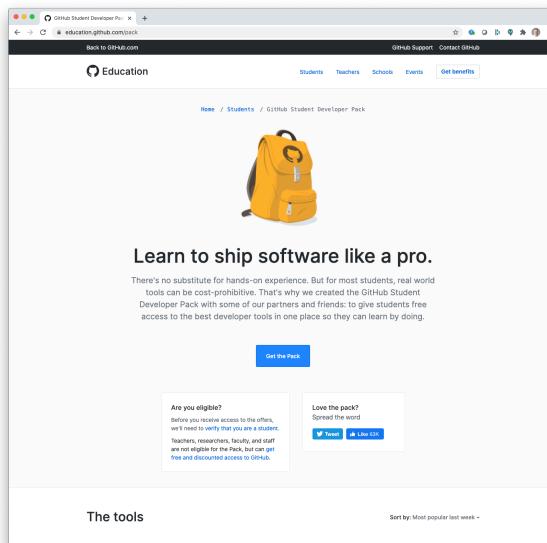


2. Sign up for GitHub Student Developer Pack

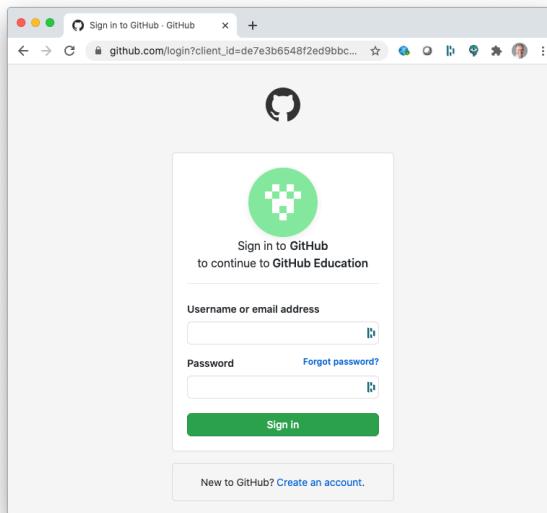
To sign up for the *GitHub Student Developer Pack*, go to this URL:

<https://education.github.com/pack>

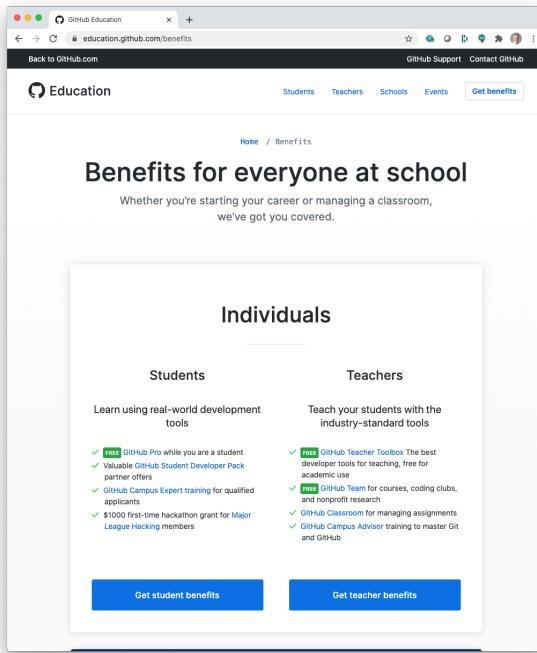
On the web page that is displayed, click on **Get the Pack**:



The **Sign in to GitHub** dialog is displayed:



Enter your GitHub username and password, Click **Sign in**. When properly authenticated, you **GitHub Education Benefits** page is displayed.



Click on **Get student benefits** and follow the instructions.

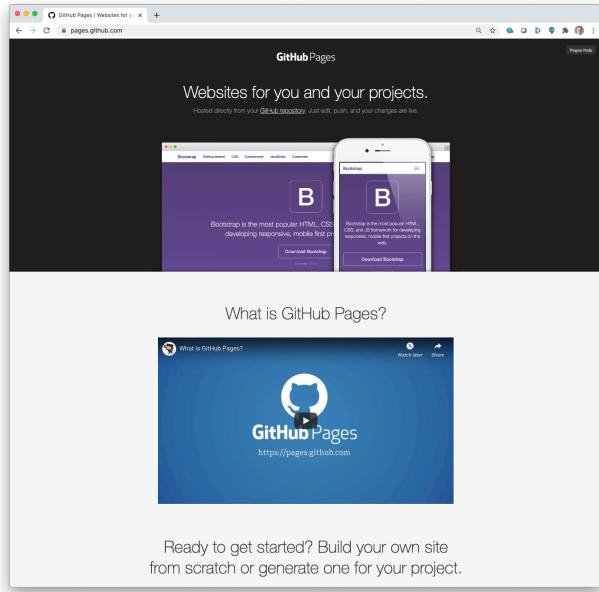
One of the benefits provided when signing up for the *GitHub Student Developer Pack* is that your **GitHub Pages** repository (see Section 3) will be “private” for free. If you do not sign up for the *GitHub Student Developer Pack*, only “public” repositories will be free.

3. Sign up for GitHub Pages

To sign up for *GitHub Pages*, go to this URL:

<https://pages.github.com/>

The **GitHub Pages** startup page is displayed.



Scroll down and select **User or organization site**, if not already selected, as shown below.

Ready to get started? Build your own site from scratch or generate one for your project.

You get one site per GitHub account and organization, and unlimited project sites. Let's get started.

[User or organization site](#) [Project site](#)

1 Create a repository

Head over to [GitHub](#) and create a new repository named `username.github.io`, where `username` is your username (or organization name) on GitHub.

If the first part of the repository doesn't exactly match your username, it won't work, so make sure to get it right.

Create a new repository

Repository name Great repository names are short and memorable. Need inspiration? [How about `Releaseweez`?](#)

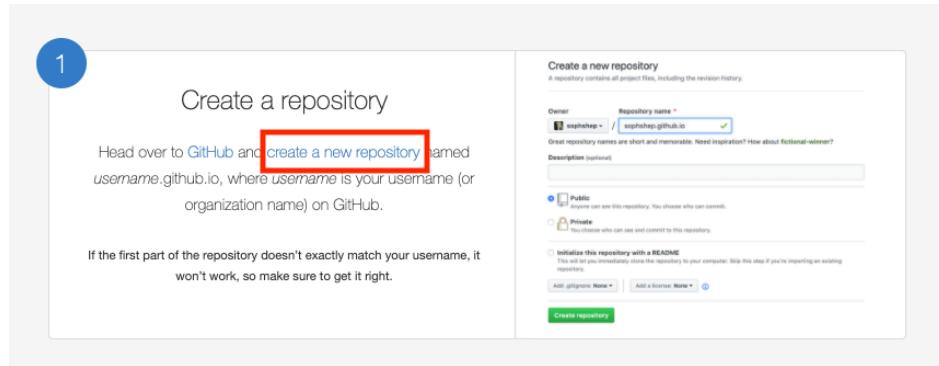
Description (optional)

Public Anyone can see this repository. You choose who can commit.
 Private Only people who can see and contribute to this repository can.

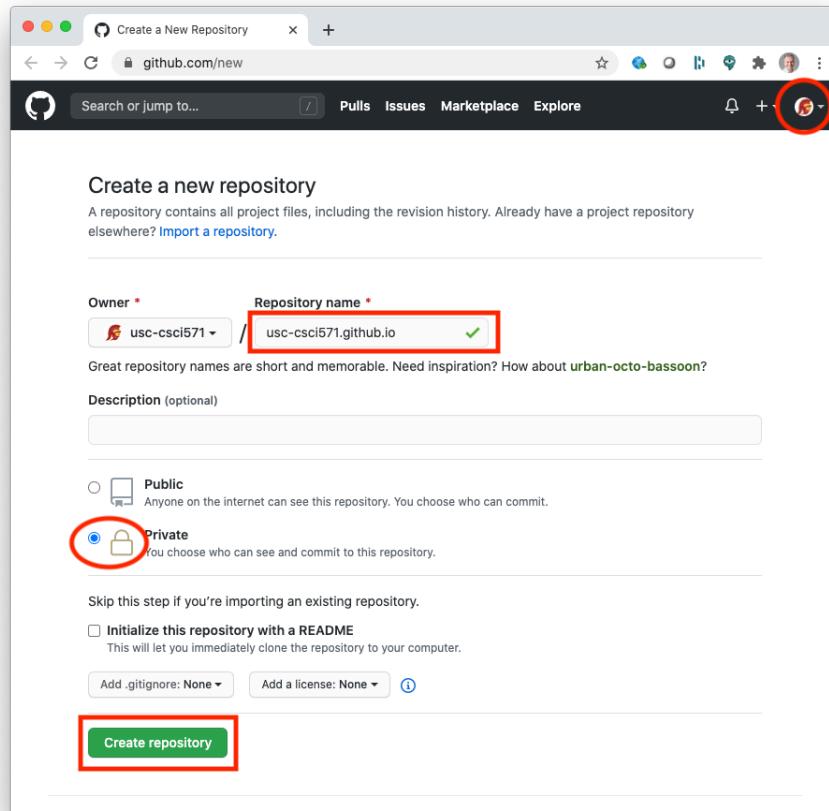
Initialize this repository with a README
 This will add a README file to your repository. Help this step if you're importing an existing project.
[Add a README file](#) [Add a license file](#)

[Create repository](#)

Next, click on the hyperlink **create a new repository**, as shown below.



The GitHub **Create a new repository** dialog is displayed. The owner will be prefilled. For Repository name enter **username.github.io**, where username is your GitHub username. You can find it when clicking your icon in the top right of the window. Select **Private** for the type of repository and click **Create repository**.



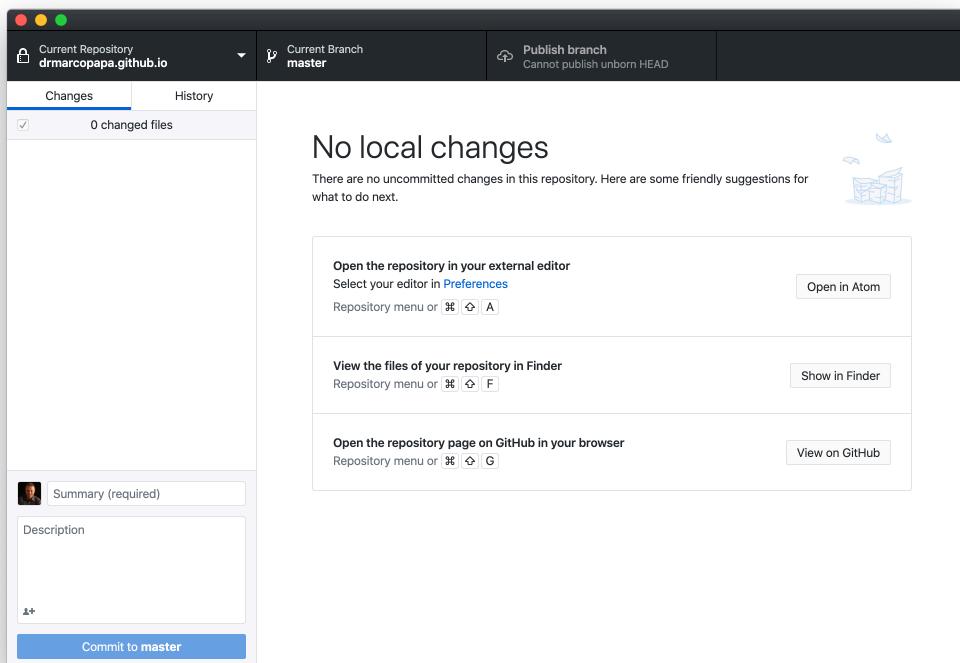
The next steps to follow are different, depending on which **git client** you use: a **terminal** window or the **GitHub Desktop** app. We include snapshots using **GitHub Desktop**.

Steps 2 though 5 are described at:

<https://pages.github.com/>

and involve the following:

2. **Clone the repository.** Click the "Set up in Desktop" button. When the GitHub desktop app opens, save the project.

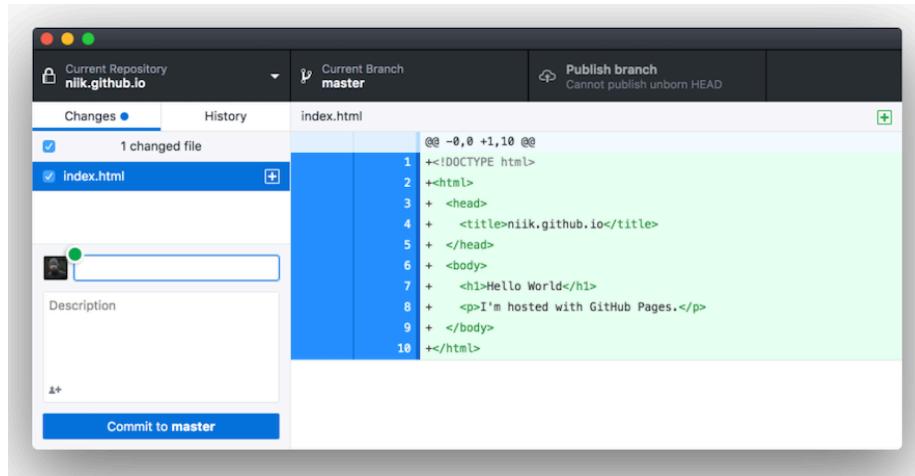


3. **Create an index file.** Grab your favorite text editor and add an index.html file to your project.

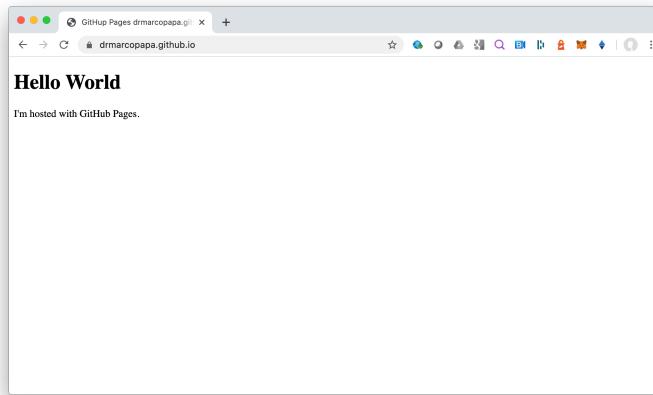
A screenshot of a text editor window titled 'index.html'. The code inside the editor is:

```
<!DOCTYPE html>
<html>
<body>
<h1>Hello World</h1>
<p>I'm hosted with GitHub Pages.</p>
</body>
</html>
```

4. **Commit and publish.** Enter the repository, commit your changes, and press the publish button.



5. **View your homepage.** Fire up a browser and go to <https://username.github.io>



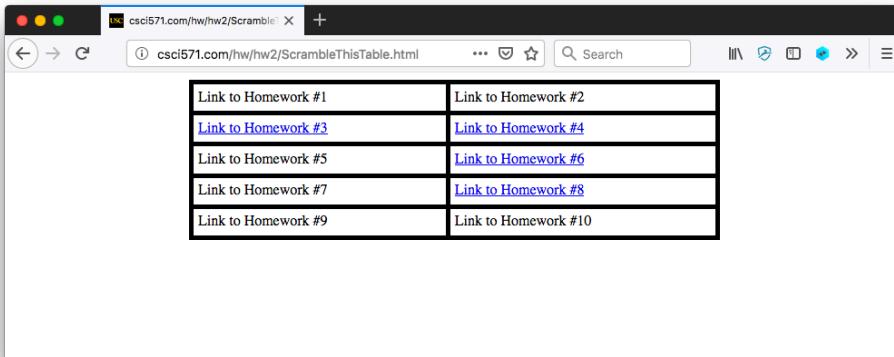
4. Create the Table of Exercises

The next step is to upload the table of exercises. We are providing you with a sample table at:

<http://csci571.com/hw/hw2/ScrambleThisTable.html>

Download this page to your local machine. You **MUST change the file name** to the name you selected during the **Class Sign Up**. The table of exercises should be named using a name that cannot be easily guessed, such as **sd45fh67.html**. Create that file, add it to your project on GitHub Pages, commit and publish, and you are done.

The sample table is shown below.



A screenshot of a web browser window titled "csci571.com/hw/hw2/Scramble". The address bar shows the same URL. The page contains a 5x2 grid table. The first column contains the following text: "Link to Homework #1", "Link to Homework #3", "Link to Homework #5", "Link to Homework #7", and "Link to Homework #9". The second column contains the following text: "Link to Homework #2", "Link to Homework #4", "Link to Homework #6", "Link to Homework #8", and "Link to Homework #10". All text is in blue, indicating it is a hyperlink.

Link to Homework #1	Link to Homework #2
Link to Homework #3	Link to Homework #4
Link to Homework #5	Link to Homework #6
Link to Homework #7	Link to Homework #8
Link to Homework #9	Link to Homework #10

Table of Exercises

Notice that the hyperlinks to the various exercises point to non-existent files and URLs. You will have to replace them with the actual web page names and URLs of your exercises, either on GitHub Pages or in the cloud.

As you complete each homework assignment, you should add the URL of the homework to the appropriate location in the table listed as "**Link to Homework #**" (see Figure 8). In this way the TA/grader can easily locate your homework assignments.

5. Install and use Git (optional)

If you do not already have **Git** installed on your laptop, instructions to download Git are available here:

<https://git-scm.com/downloads>

Documentation on Git is available here:

<https://git-scm.com/doc>

There are numerous tutorials online on how to install Git. These are my favorites. Atlassian's BitBucket "Install Git" tutorial with step-by-step instruction on how to install Git on a MacOS, Windows and Linux platforms":

<https://www.atlassian.com/git/tutorials/install-git>

Linode's "How to Install Git on Mac" tutorial, with instruction on how to install Git with

Homebrew, MacPorts, and the Package Installer:

<https://www.linode.com/docs/development/version-control/how-to-install-git-mac/>

After a successful Git install, you can just follow the create and push instructions in the Code page mentioned above. Here is an example, showing the output you will see on your screen. Notice in **bold** the CLI commands:

```
$ echo "# homerwork-1-roadster2020" >> README.md

$ git init
Initialized empty Git repository in /Users/marcopapa/Desktop/
GitHub571/.git/

$ git add README.md
warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working
directory.

$ git commit -m "first commit"
[master (root-commit) ff55c43] first commit
 1 file changed, 1 insertion(+)
 create mode 100644 README.md

$ git remote add origin https://github.com/usc-csci571/
homerwork-1-roadster2020.git

$ git push -u origin master
Username for 'https://github.com': roadster2020
Password for 'https://roadster2020@github.com': (enter
password or token when using 2FA)
Counting objects: 3, done.
Writing objects: 100% (3/3), 237 bytes | 237.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/usc-csci571/homerwork-1-roadster2020.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
```

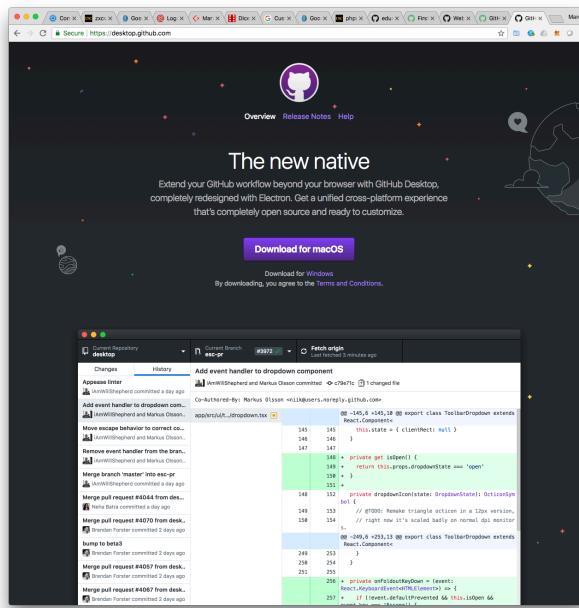
Whenever you add a new file to the local repository you need to add all the above commands, except the first 2 commands, and changing the name ‘f “first commit” to “second commit”, and so on.

Besides the CLI Git commands, a very nice desktop application is provided named **GitHub Desktop**. GitHub Desktop is available here:

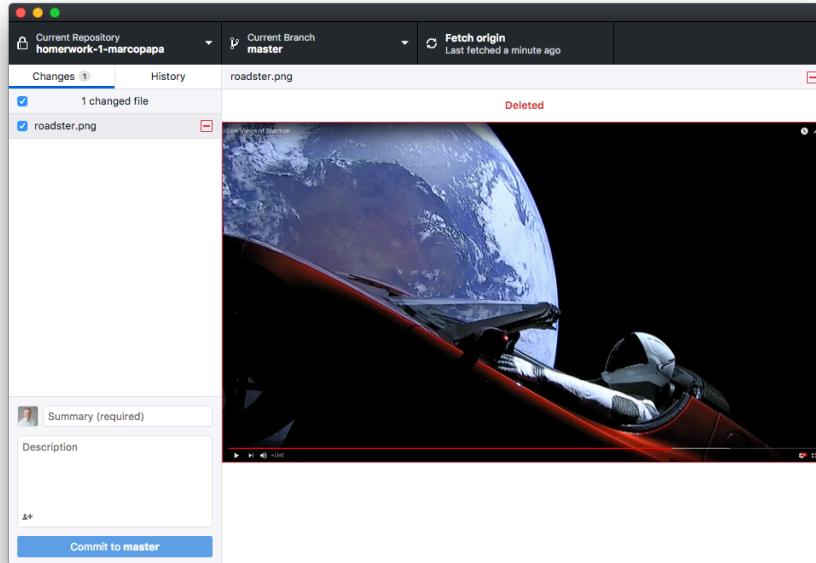
<https://desktop.github.com/>

GitHub Desktop is available for macOS and Windows. Documentation is available here:

<https://help.github.com/desktop/guides/>



After installation, the **GitHub Desktop** app interface looks like this:



The usual steps to follow in the **GitHub Desktop** app when making changes are as follows:

1. Make changes to local repository (change or add files)
2. Edit summary and Description
3. Commit to Master
4. Push origin (publish)

Happy Git.