# Class Exercise #1 – Getting Started and HTML

Instruction:

1. Please use MS Windows 10 Snipping Tool to take screen shot of the following Tasks, make sure it is legible and readable screen shot. The yellow highlighted text areas are where you need to complete.

2. Please put your full name in the header.

3. Please do not modify this document to another format, keep as MS Word with .docx.

4. Submit this document to eCampus > Submit Your Assignment Here > Class Exercise #1 Submit Here. Refer to the syllabi course calendar for due date.

=================================================================

**Part I has Task A, B, C and D: File Management and DOS Concepts**

**Part II has Task A, B and C: Software and First “Hello World” webpage**

**Part III has Task A, B and C: Create default index.html**

=================================================================

**Part I has Task A and B: File Management and DOS Concepts**

## Task A

1. Right mouse click on the Windows 10 Windows icon, bottom left, and select File Explorer.
2. Find the Document directory, create the folder structure shown below.

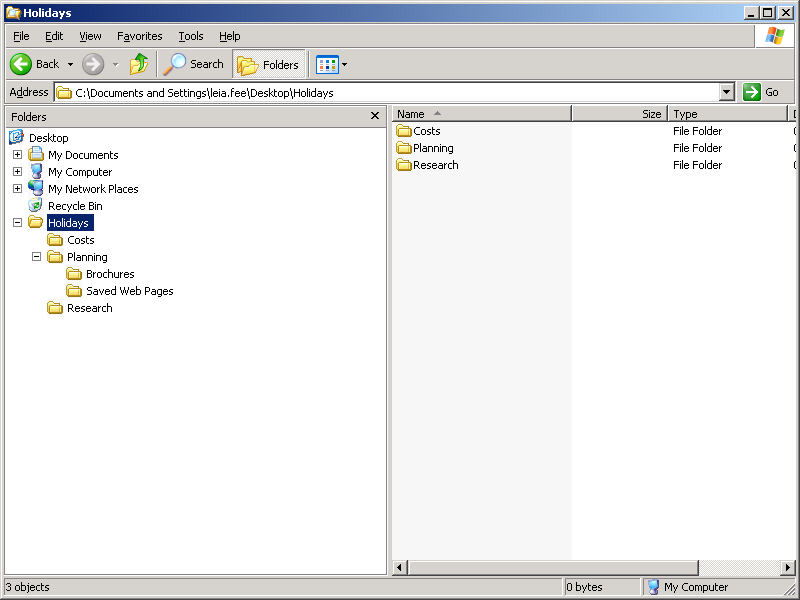


3. Use Sniping Tool to capture the screen shot of the Document directory with Schoolwork sub directory, and three sub directories, and copy/paste below:

4. The History folder needs to be further subdivided into two sub folders Courseworkand Past Exams. Create these folders in the **History** subfolder and update the diagram above to show them. (Use Sniping Tool to capture the screen shot)

## Task B

1. Under the Document subdirectory, create the folder structure shown below.



1. Use Sniping Tool to capture the screen shot, copy/paste below:

## Task C

In your Document subdirectory create the folder structure shown below.

IMED 1316

ClassProject

Documents

ClassExercises

020420

020320

013020

012920

docs

assets

img

1. Open the Folders Pane at the left of Windows Explorer and expand all folders to show the structure. Use the Snipping Tool and copy/paste below of the File structure in Task C.

Graphical user interface, application, table

Description automatically generated

**Task D**: Please review the two DOS videos and follow along with your computer.

Class Exercise #1 - Task D part 1 - DOS Commands: <https://www.youtube.com/watch?v=_CNsvn-ZfjA>

File Mangement - Part 1B - ClassExercise\_1: <https://www.youtube.com/watch?v=k1ZyW_SpgmU&feature=youtu.be>

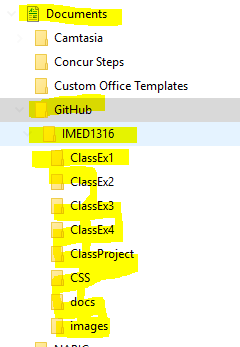
=================================================================

**Part II has Task A, B and C.** Must complete Part I before you move forward.

**Task A** – Please create one directory on your local computer with the following file structure, this location will be the default sub directory for all three software (GitHub Desktop, FTP and MS VS Code) to upload to web server.

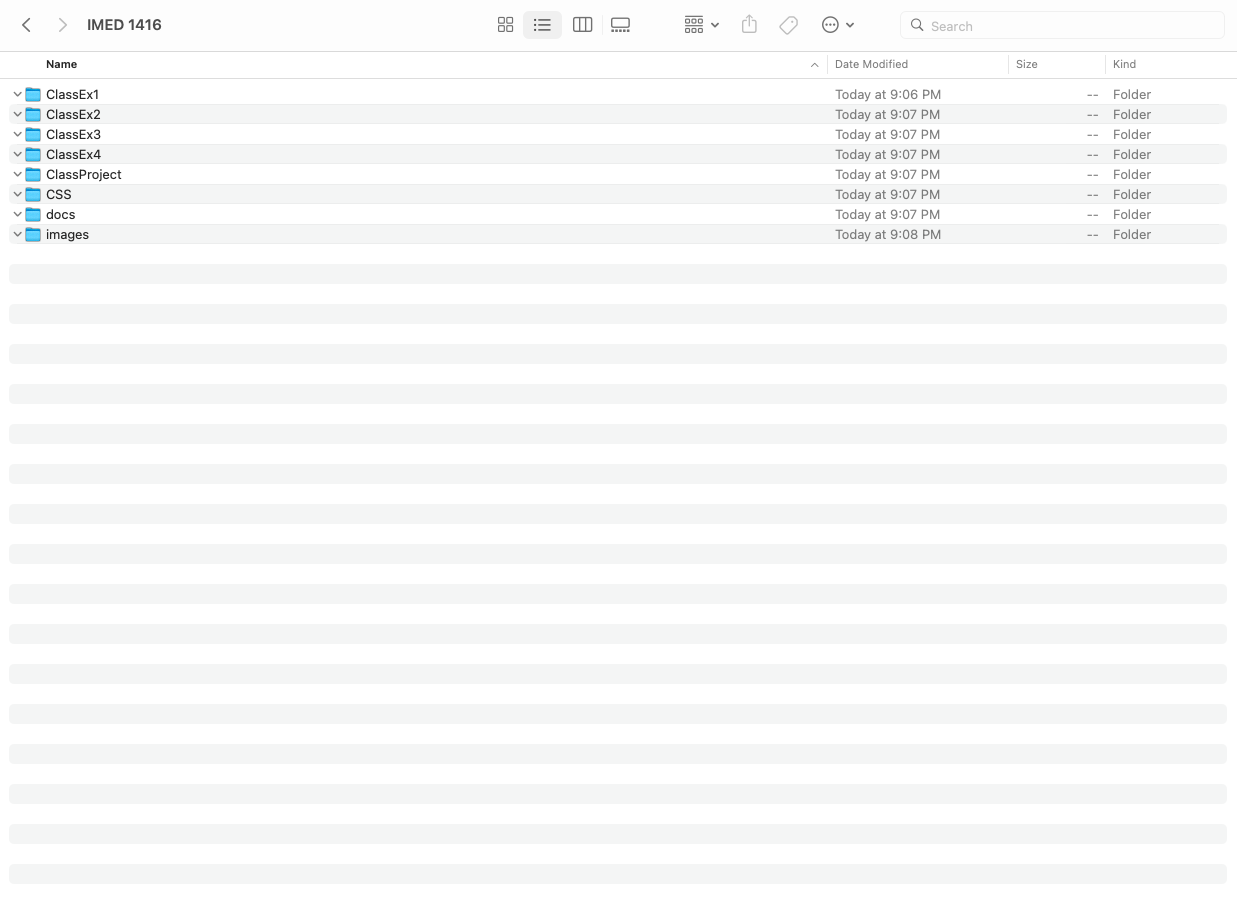
Under the Documents folder (sub directory), create the following subdirectory, depending which course you are enrolled, use one of these file names: IMED1316, IMED1416 or ARTC1317 (NOTE: no spaces or special characters to file name). Inside the sub directory, for example, if I use IMED1316 sub directory. Please create the following subdirectories: ClassEx1, ClassEx2, ClassEx3, ClassEx4, ClassProject, and images sub directories.

You will create the index.html at the level as the ClassEx1, ClassEx2, ClassEx3, ClassEx4, ClassProject, and images sub directories. My subdirectories file structure looks like this:



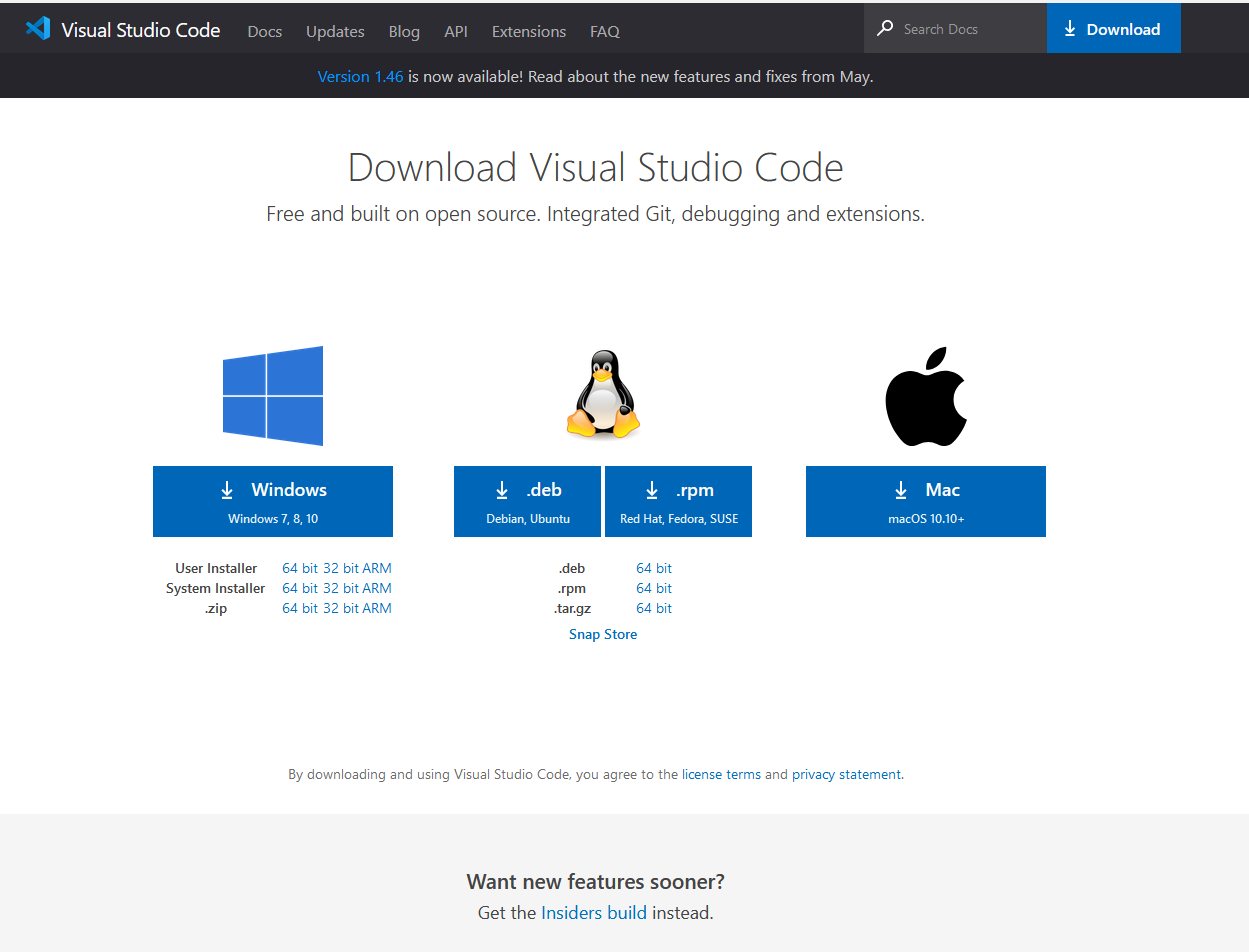
NOTE: The yellow highlighted and indentations of the sub directories, if the directories are aligned equally, means they are at the same level.

Please use the Windows Snipping Tool, with the File Explorer (right mouse click on the Start button and select File Explorer) and take screen shot of the file structure (copy/paste) of your local computer below:



**Task B** – Before you download and install the three software, please disable your antivirus on your Windows or Mac OS, or else, it going to recognize as a virus and will remove. Download and Install MS Visual Code IDE, FileZilla FTP Client, and GitHub Desktop. (These webpages might vary over time), and you need to pick the operating system and chipset that match the software.

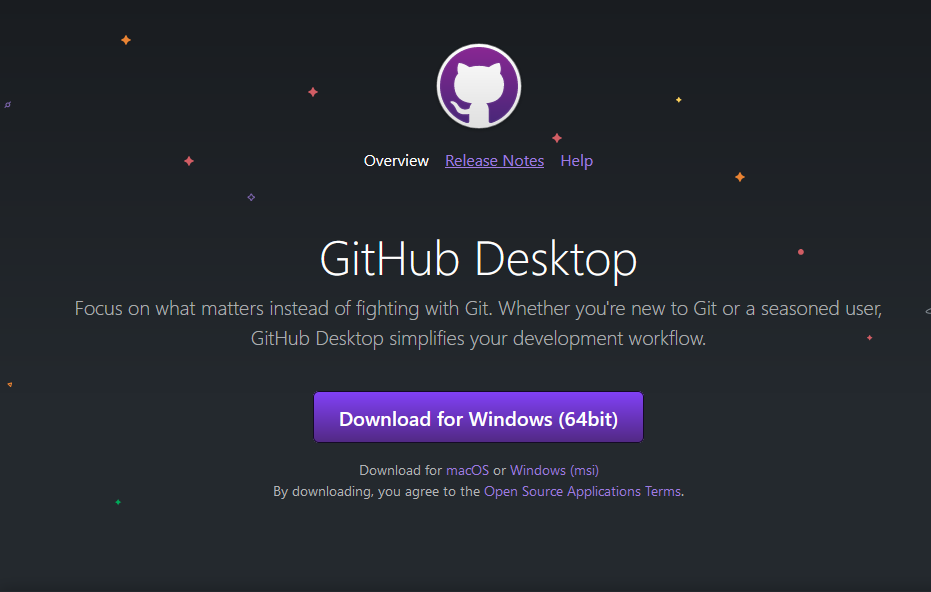
MS Visual Code IDE: <https://code.visualstudio.com/download> and install Live Server



FileZilla FTP Client: <https://filezilla-project.org/download.php?platform=win64>



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



**Task B.1**: With respect to **MS Visual Code**, you will need to download, accept the EULA, and accept default setting.

**Task B.2**: Install the **FileZilla client**, accept the EULA, and accept default settings. Look for my email with your username and password, keep in secure place, do not share with anyone.

**Task B.3**: In regard to **GitHub Desktop**, you will need a personal email and create a GitHub account. Please register, for the duration of this course, I will need to access your GitHub account to grade. After the course is over, you will need to change the password. Please complete the information below for GitHub account:

Username: nrtesfa

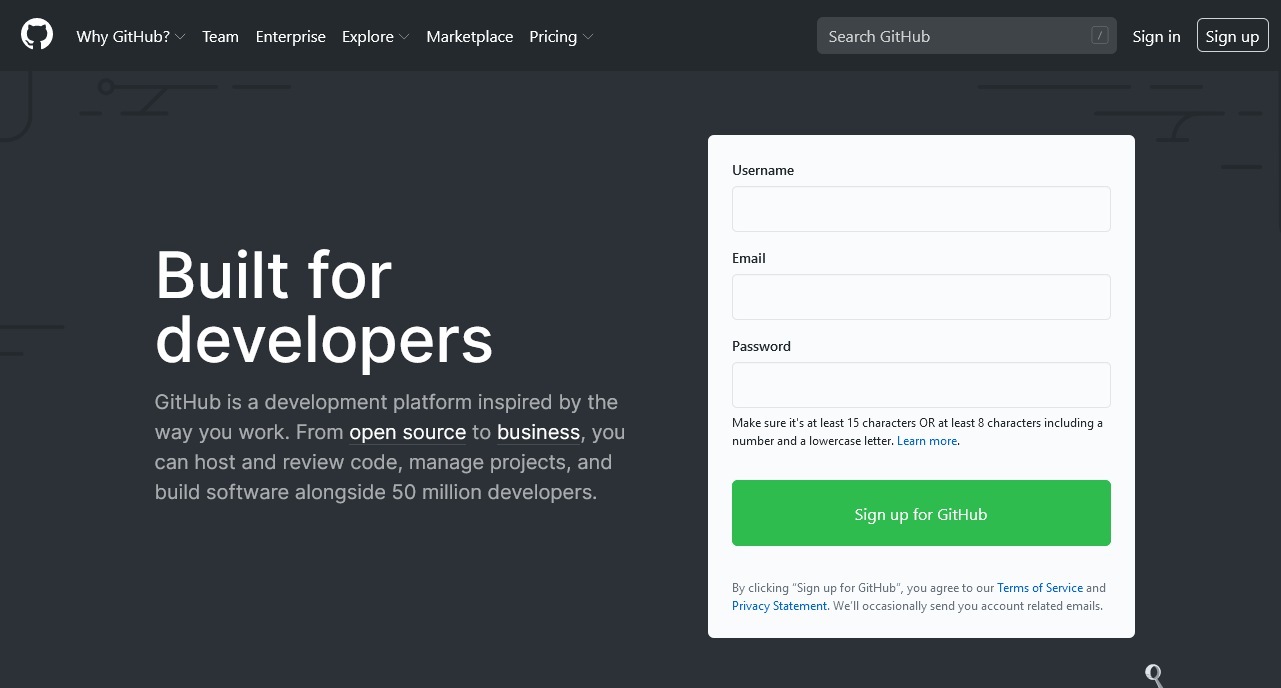
Password: Yearup10

Please following this instruction on GitHub Desktop:

GitHub is not only a great place to store and share your code with others, but they also offer free web hosting of your HTML, CSS, and JavaScript projects!

In this class exercise, I’ll show you how to get setup in order to publish and share your own static HTML website on GitHub using the **GitHub Desktop** app!

1. Create a GitHub account: go to <https://github.com/>



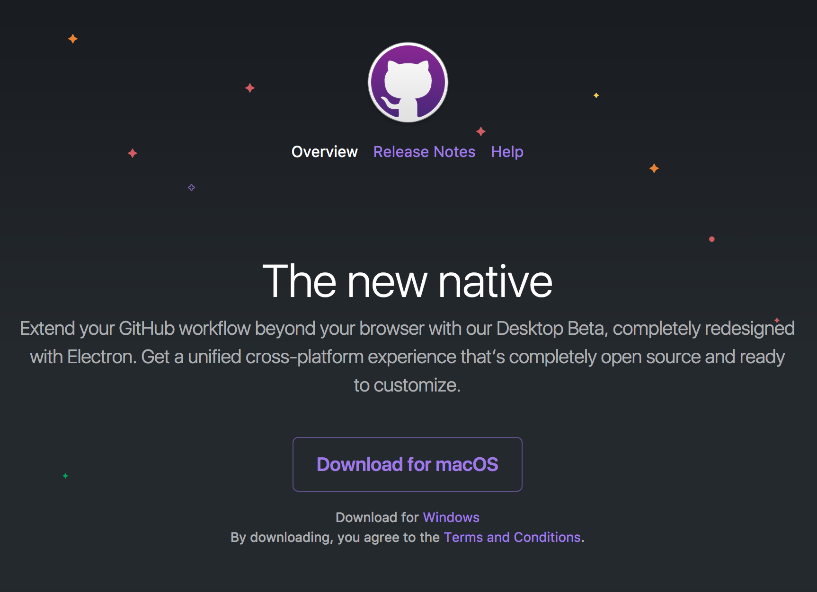
First thing’s first: get yourself a GitHub account! You won’t have access to the free hosting, or any other awesome features GitHub has to offer without one.

The GitHub homepage with signup form to the right of the screen. Along the top is the main navigation area. To the left of the screen text reads, “Built for developers” with a description of what GitHub is all about.

With a GitHub account you’ll also be able to chat with other web developers, such as yourself, through Issues, and contribute to other open source projects!

2. Download and install the GitHub Desktop app: <https://desktop.github.com/>

This is the app that we’ll use to get our code up on GitHub. It’s easy to use so don’t fret; the user interface is easy to learn!



The GitHub Desktop homepage with a “Download for macOS” link, along with a Windows link. Along the top is the main navigation area. In the middle of the screen text reads, “The new native” with a description of what GitHub Desktop is all about.

* Download GitHub Desktop for macOS   
  OR
* Download GitHub Desktop for Windows

Open the downloaded application file and continue through the installation process as required with your computer’s operating system.

I assume you are able to install this three software without issues. If you do, please post your question in the eCampus > Discussion Board > Discussion Board for Trouble Shooting/

Next, we’ll be creating a new GitHub project!

**Task C** – Create your first HelloWorld.html page in Multimedia Web Server and GitHub Web Server. Please note, the rationale for Task C is to demonstrate old technology (FTP Transfer file from your local computer to web server) and new technology using GitHub workflow (current industry). 80% of a programmer life cycle is maintaining other people code, if you work in the industry, you will need to know how to transfer files.

**Task C.1** – Use FileZilla Client to transfer from your local computer to web server.

Refer to **eCampus > Lecture and Lab Information > Class Exercise #1 – Task C – FTP Connect,** please follow from Step 1 to 5, please note, step 2. is the FileZilla Client for Windows OS, if you want the Mac, then go the FileZilla and download the Mac version.

After you complete the five steps, using Windows Snipping Tool, take screen shot of your index.html in the web browser. Make sure you have the proper URL address to view your webpage.

Please check your email for “FileZilla FTP Account Information”, recommend you copy/paste the information to avoid typos.

**URL in browser**:  http://www.mmlab2.rlc.dcccd.edu/web83XX  
**FileZilla   Host**:  sftp://www.mmlab2.rlc.dcccd.edu  
  
Username: web83XX  
password: \_\_\_\_\_\_\_\_

NOTE: I am using synonyms terminology: Username = web83XX = login = account   
  
XX is replaced with your personal account number, to view your webpage after you FTP and transfer the file (drag and drop) from local computer to FTP Client right side panel.

Open up a web browser, copy/paste this URL address:   
  
http://www.mmlab2.rlc.dcccd.edu/web83XX

NOTE: the breakdown of this URL address, there are test questions over this:

**HelloWorld.html** stands for hyper text transfer protocol  
  
 [www.mmlab2.rlc.dcccd.edu](http://www.mmlab2.rlc.dcccd.edu) part is the domain

**~~/imed1316n02~~** ~~part is the sub directory path with back slash starting~~

**/web83XX** sub directory path with back slash starting to your default **index.**html file. The web server is set to default to index.html

Both of these URL addresses are equivalent:

**http://www.mmlab2.rlc.dcccd.edu/web8391**

**and**

**http://www.mmlab2.rlc.dcccd.edu/web8391/index.html**

**The difference is the /index.html file is not required. The takeaway message is that when I ask for your domain or web address. Make sure you put the right URL address, not your local computer address. To grade, I need the address that’s on the web server.**

**file:///C:/Users/dxd8401/Documents/GitHub/IMED1316/index.html**

**This address is on my local computer not the web server, can you see the difference? Please say yes. I cannot access your local drive to grade, please know this difference between the two addresses (URL and local computer addresses).**

NOTE: In the recent email that I send you with username and password, for example, let say your user name (subdirectory) is web8391, you will need to replace the 91 in place of the XX, so the complete URL address would be:

**http://www.mmlab2.rlc.dcccd.edu/web8391**

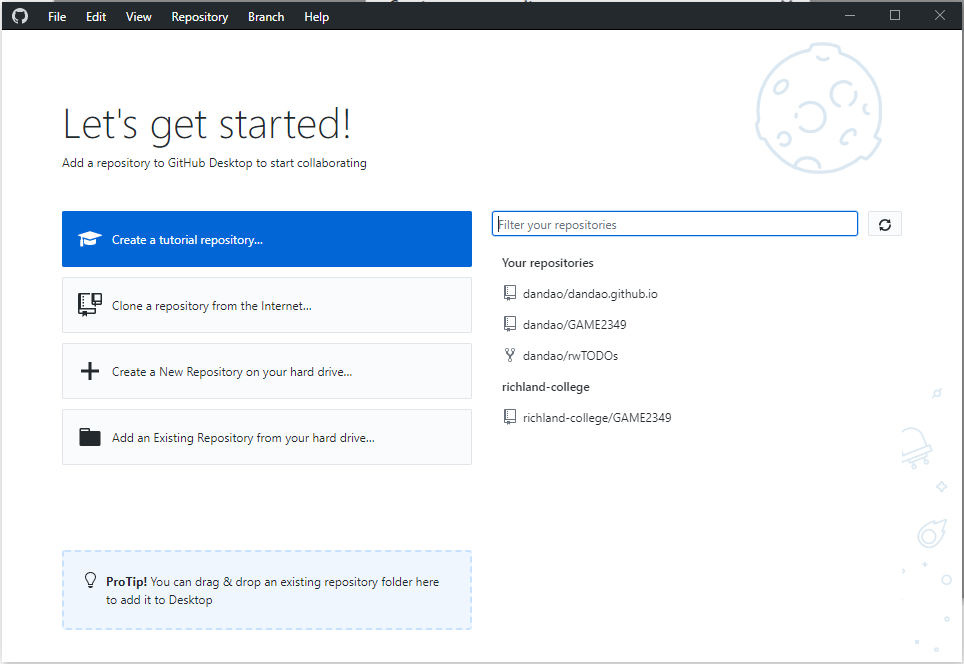
**(This is not a live webpage, just a demonstration)**

**Task C.2 – Use MS Visual Code IDE and GitHub Desktop to create HelloWorld.html file and transfer to ClassEx1 subdirectory**

**1. Please watch these videos:**

**Class Exercise #1 - Task C.2 - part 1:** <https://www.youtube.com/watch?v=eWiKHNmxqY0> **Class Exercise #1 - Task C.2 - part 2:** <https://www.youtube.com/watch?v=UNngv34gACQ> **Class Exercise #1 - Task C.2 - part 3:** <https://www.youtube.com/watch?v=cGCgTeyFrLU>

**2. click on the “Create a New Repository on your hard drive…”**



**Before we upload our code to GitHub, we first need to make GitHub Desktop aware of our project files. We do this by creating a new Repository folder, at the level where we can see the content of IMED1316 sub directory.**

**3. Open up the GitHub Desktop app and click the “Create New Repository on your hard drive…” button.**

**Please complete the following:**

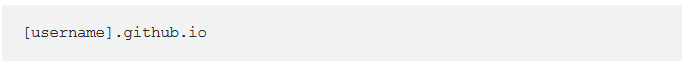
**Name: Your first name and full last name, i.e. dandao**

**Description: Class Exercise #1 – Task C.2**

**Local Path and click on the “Choose” button and navigate to your class name and number, in my case, IMED1316 contains all the subdirectories that I create earlier.**

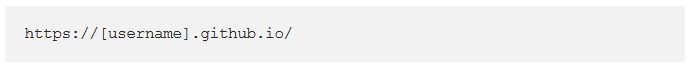
**Click on the “Create repository” button.**

**4. When the “Create a New Repository” dialog window appears, fill in the “Name” text input as:**



Replace [username] with your GitHub account username.

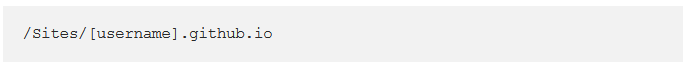
**Note:** It’s very important to name your repository in this manner; this will tell GitHub to host the files in this project automatically and display them when someone points their browser to:

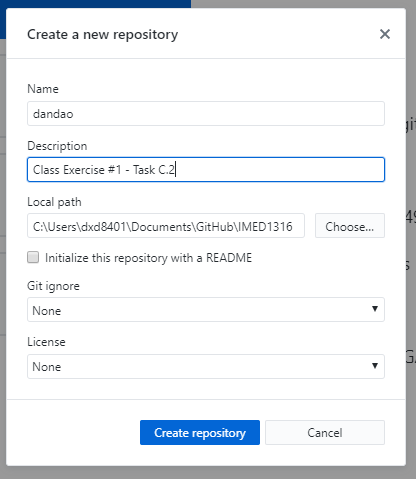


This will be the web address you will share when your site is ready to publish and go live!

The **“Local Path”** text input is where your repository will live on your computer; this is **not** where your website files currently reside on your computer’s hard drive. GitHub Desktop needs to create a new, empty folder.

We’ll need to copy your site files over later, but for now, select a folder such as /Sites in your home folder. This will create a new folder inside the /Sites folder with the name of your repository, which will look like:

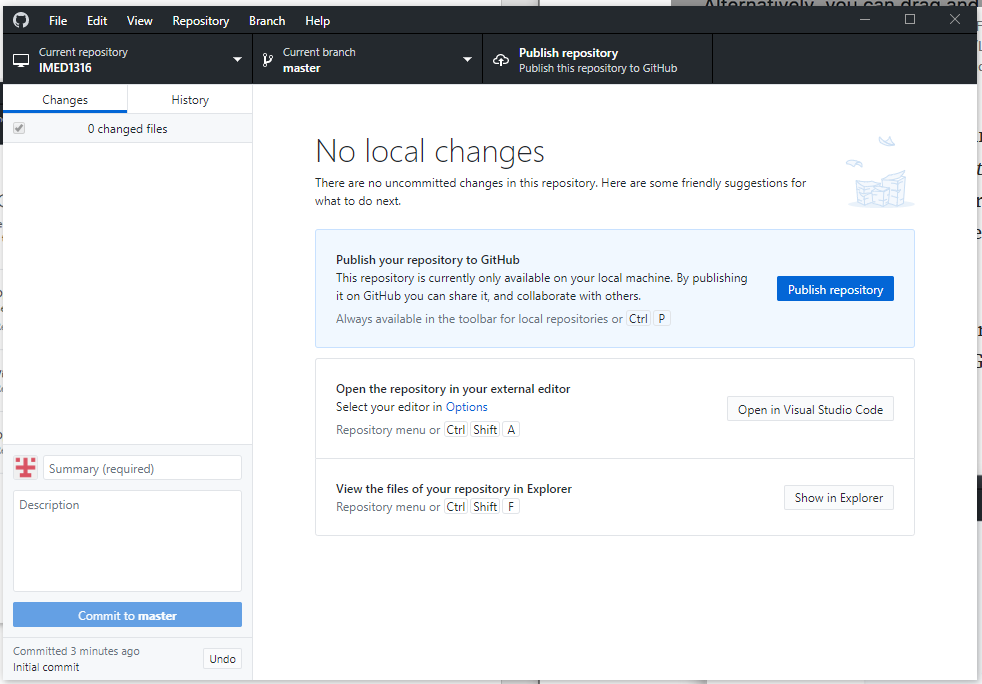




The “Create a New Repository” dialog window. Form controls include, “Name”, “Local Path”, “Initialize this repository with a README”, “Git Ignore”, and “License”. Two action controls at the bottom are labelled, “Cancel” and “Create Repository”.

**5.** Leave all the other options in the form at their default settings and click the **“Create Repository”** button. For now, we just want to create the folder which GitHub Desktop will recognize as a folder it can work with. For the sake of the guide, let’s describe this new folder as your GitHub Repository folder.

After your GitHub Repository folder has been created by clicking the **“Create Repository”** button, the GitHub Desktop app should look something like this:



Please complete in the “Summary (required)” and “Description” fields.

The GitHub Desktop app with an empty repository folder. Along the top are option controls for “Current Repository”, “Current Branch”, and “Publish repository”. The left side includes a space for a listing of files, a “Summary” text input, a “Description” text box, and a “Commit to master” button.

6. Your files to the new Repository folder

Let’s say up until now you’ve been working on your website in a folder located at /Documents/my-site; we’ll describe this as your working folder. Within this working folder there’s probably some files and folders named index.html, /css, or /javascript. What we need to do is copy everything from the working folder over to our new GitHub Repository folder.

In the previous step, we created the GitHub Repository folder inside the /Sites folder with the name, [username]. github.io. So, this means the GitHub Repository folder should be located at /Sites/[username]. github.io.

The GitHub Desktop app, now displaying the added files on the left side bar and the text of the highlighted file in the main section of the app.

With all our files in the GitHub Repository folder, we’re ready to commit our files!

To “commit” files means to take a “snapshot” of the files and folders in their current state. By doing so, we get them ready to be pushed up to a repository on GitHub.

7. Before we commit our files for publishing, we first need to tell GitHub what this commit is all about. Fill in the “Summary” text input, located near the bottom left of the user interface, with a high-level title of what this commit entails; something simple like, “My first commit! 😀” works just fine in this instance.

Optionally, you can add more details about the commit in the “Description” text box. Usually you’d want to include things like what changed; maybe you fixed a bug or added some new content. It’s great to have these notes available for other developers or even future you to make sure you know why you made those changes oh so long ago!

Once the “Summary” and optional “Description” fields are filled out to your liking, click the “Commit to master” button!

8. Publish your website

After clicking the “Commit to master” button in the previous step, you may be wondering:

What just happened? My files are gone, and the app is empty now!

**Don’t fret!** In the bottom left corner of the GitHub Desktop interface there’s a little indicator that your commit was successful.

9. Use Windows Snipping Tool, take screen shot of web browsers of two files: **HelloWorld.html** and **index.html files** from your [username]. github.io

Graphical user interface, application, Word

Description automatically generated

Graphical user interface, application, Word

Description automatically generated

For instance, mine domain is <https://dandao.github.io/>

What is your GitHub domain? <https://nrtesfa.github.io/nayabtesfa/>

**Submit this document to eCampus > Submit Your Assignment Here > Class Exercise #1 Submit Here. Refer to the syllabi course calendar for due date.**

**NOTE:** Please follow the instruction from Task C three videos, you will need to link from the index.html to HelloWorld.html, and from HelloWorld.html bac to index.html. All codes are demonstrated in the Task C.2 videos.