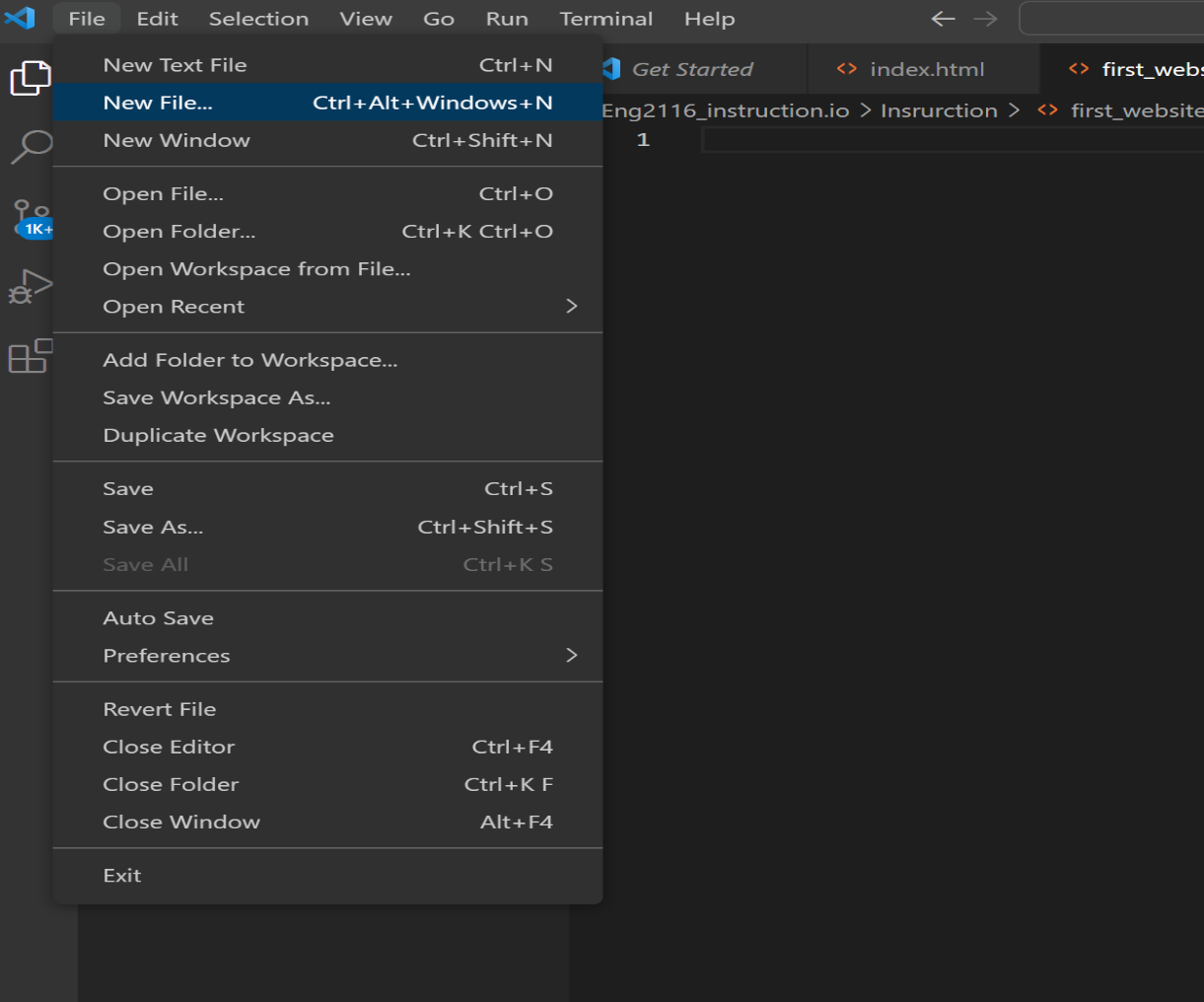
**Set up Coding Workplace and Clone Repository for Front-end Developers**

I will create manual instructions for a website application developer to set up the workplace for coding. This instruction guides users to prepare a workplace folder, manage code files, and pull up their code to GitHub without issues.

The problems this instruction would solve for the users are quickly determining the instructions and knowing all material needed for the website developer and designer.

Warning! Don’t start with writing code on a code editor to develop your website. Click files, create new files by double-clicking on New Files, and start coding is not recommended.

It’s wrong for a website developer or front-end developer to start coding without a workplace folder. Because they struggle to manage their files, and sometimes the local host is not responding to executing code, we need a cloned repository to the workplace folder.





This one is the wrong way to start.

1

2



Figure 1: Specify the wrong way to start writing code on visual video code.

**Material Needed:**

* Notepad, Visual Video Code (Visual Video Code is recommended for website developers.)
* GitHub account
* Browser (Fixer fox and Chrome are recommended)
* Desktop GitHub

GitHub account and GitHub Desktop are the fundamental tools for a front-end developer. GitHub is an open resource; it helps to store code files and executes code from the GitHub website.



**Step 1) Create a GitHub account.**



****

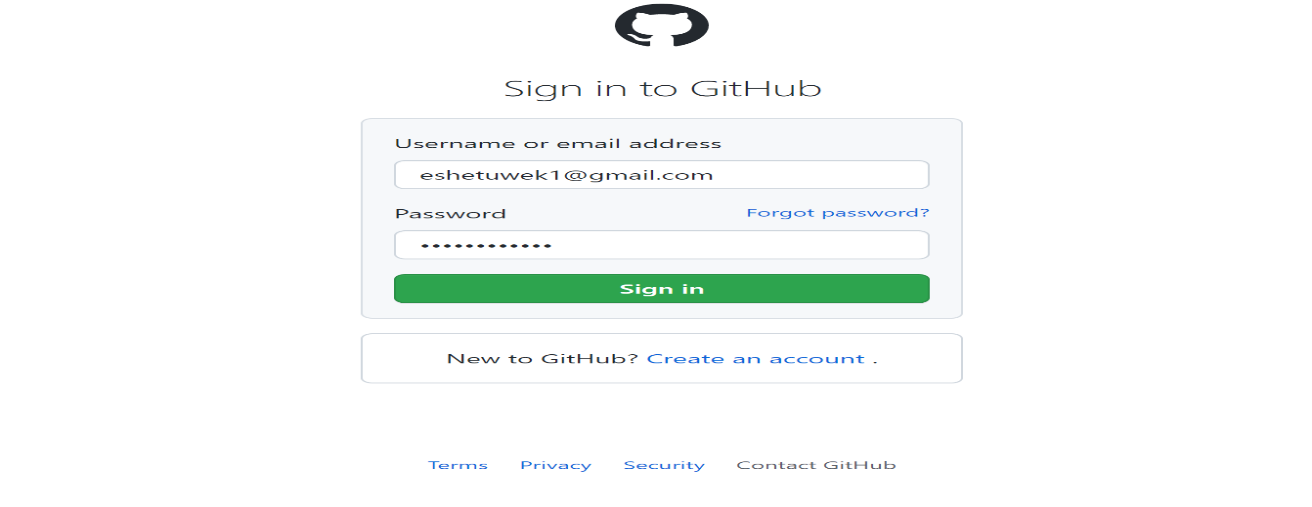
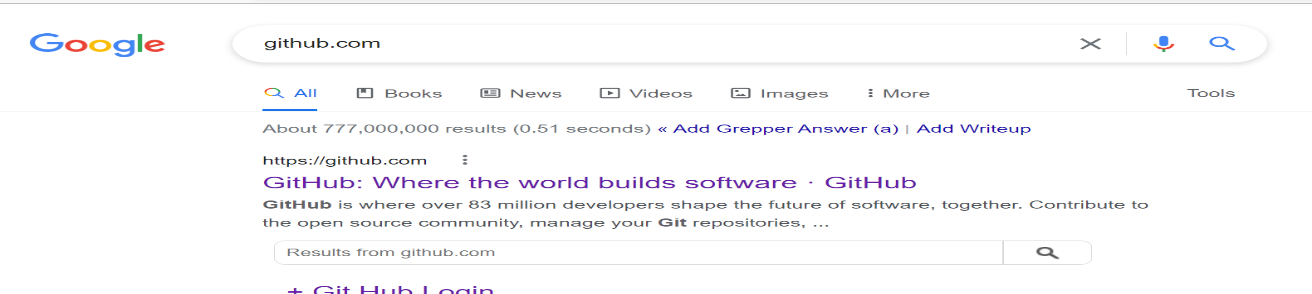
1. Go to browsers and write github.com.

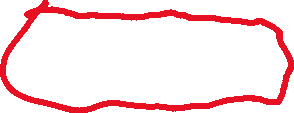


1. Choose the second one
2. If you have one before, log in to your account by writing your email and password, then press the sign button.



1. If not, create a new one.





3

4

2

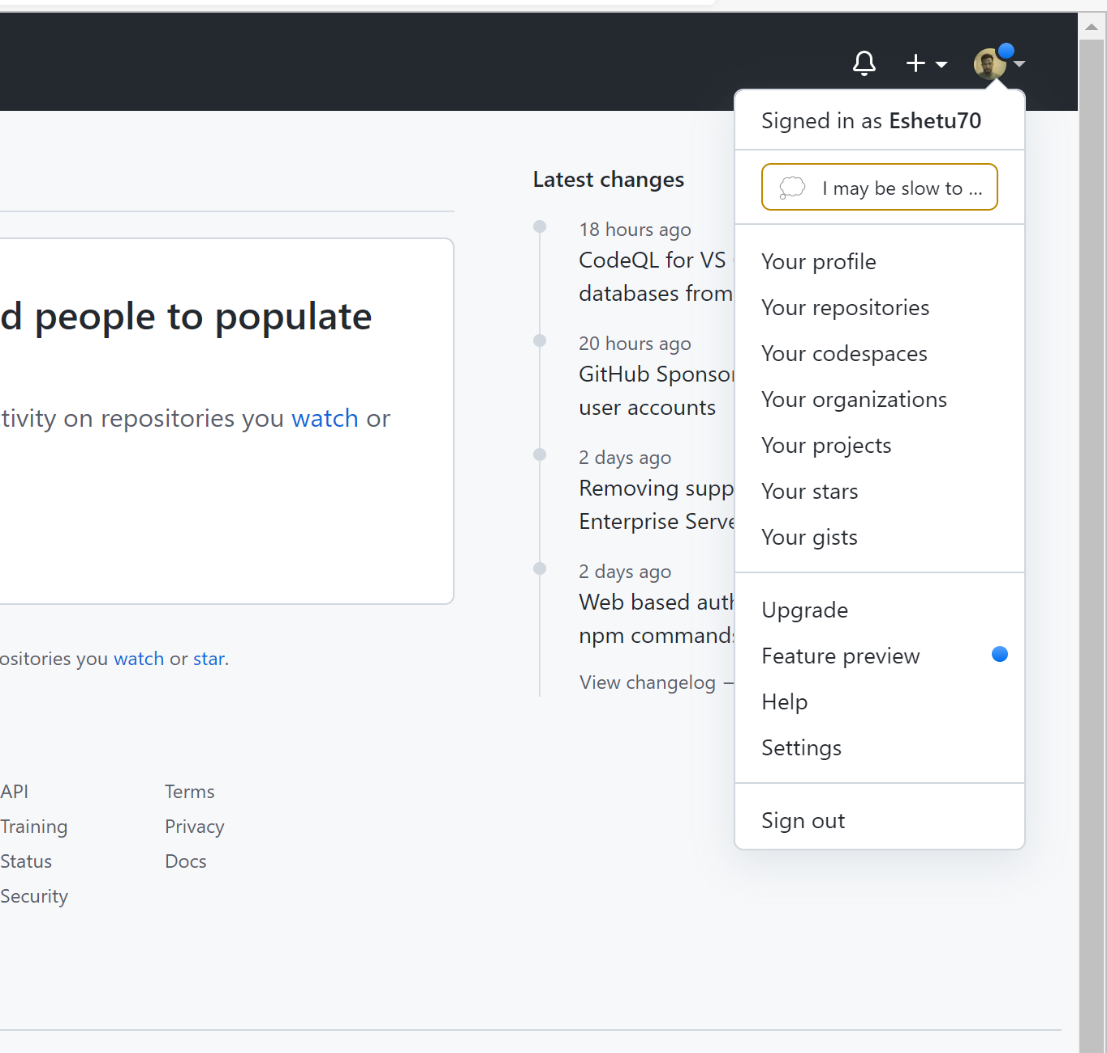
1



Figure 2: Create a GitHub account and sign up

**Step 2) Create a new repository.**

* After login into your account, go to your profile, click on it, then select repositories.
* Follow steps one and two.



Step 1



Step2



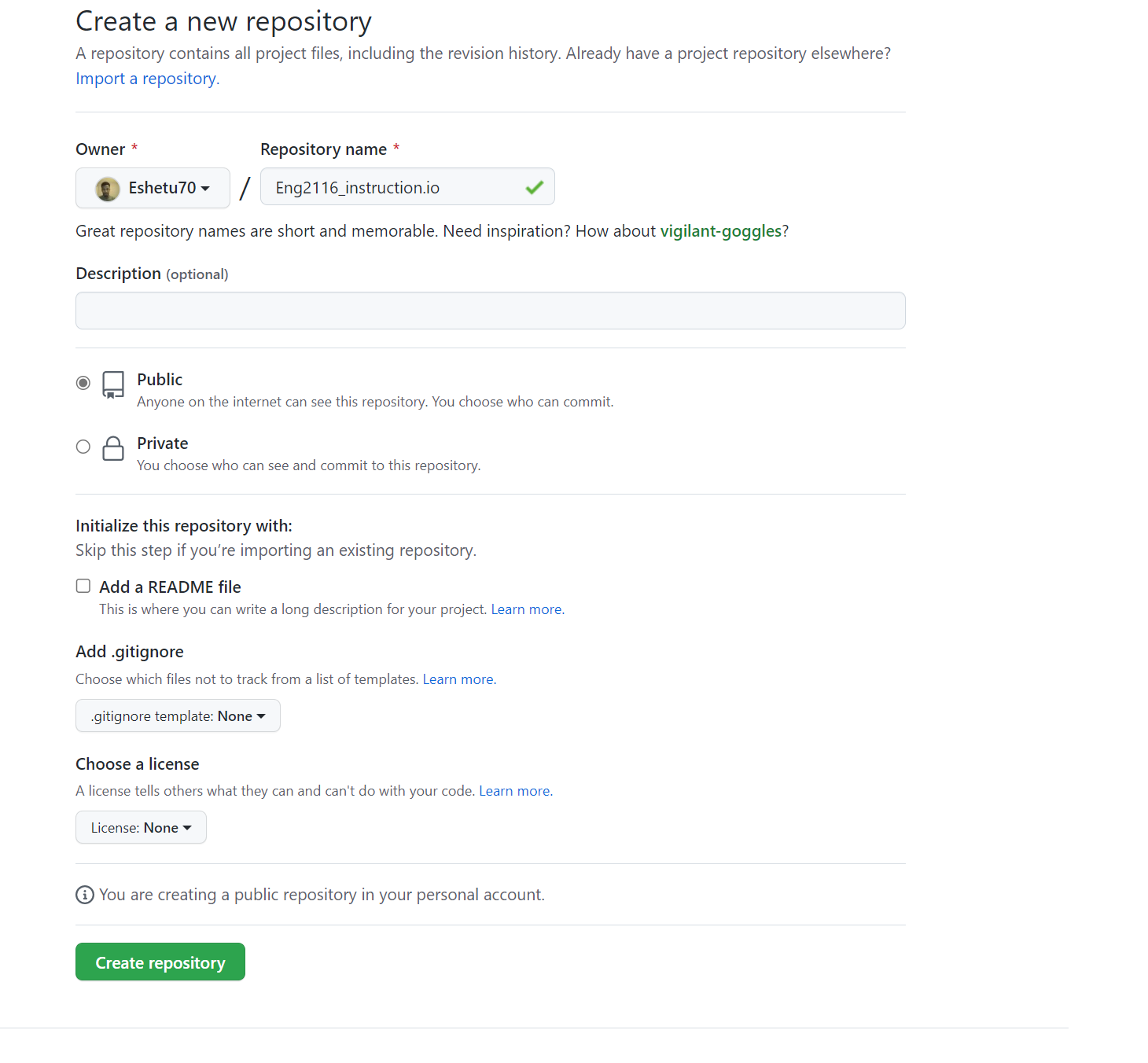
Figure 3: Select repositories from the profile list

**Step 2. a) Write your repository name, including(.io)**

Warning: when creating a new repository, include repository name dot io like (instruction.io).

Gihub.io is the GitHub page that lets you deploy that project into elegant websites, documents, or anything you want to share with the world.

1. Write your repository name, including dot io at the end.
2. Click on the Create repository button.





2

1

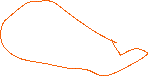


Figure 3. a: Create a new repository including (.io)

Then you have a new page like the picture shown below.

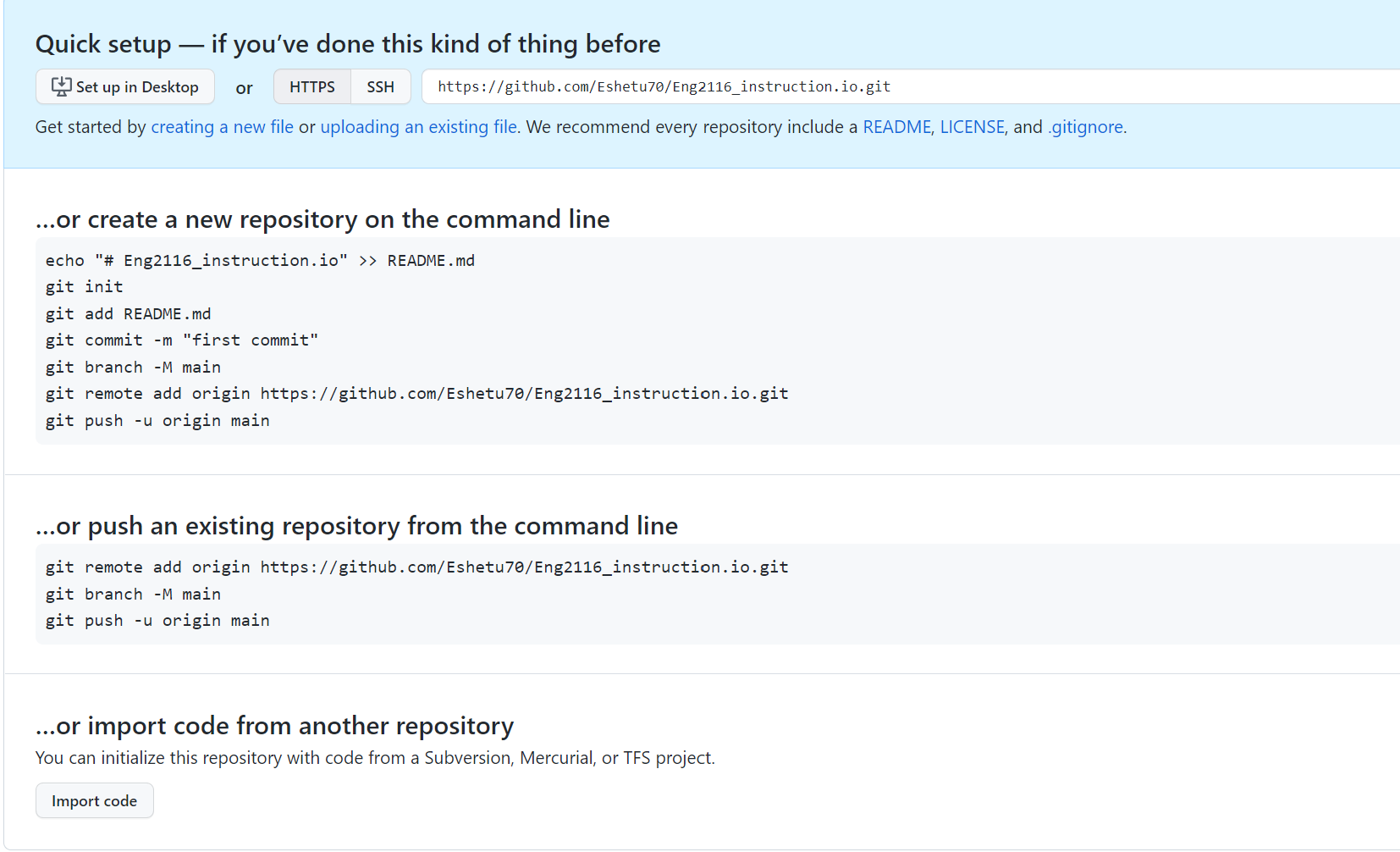


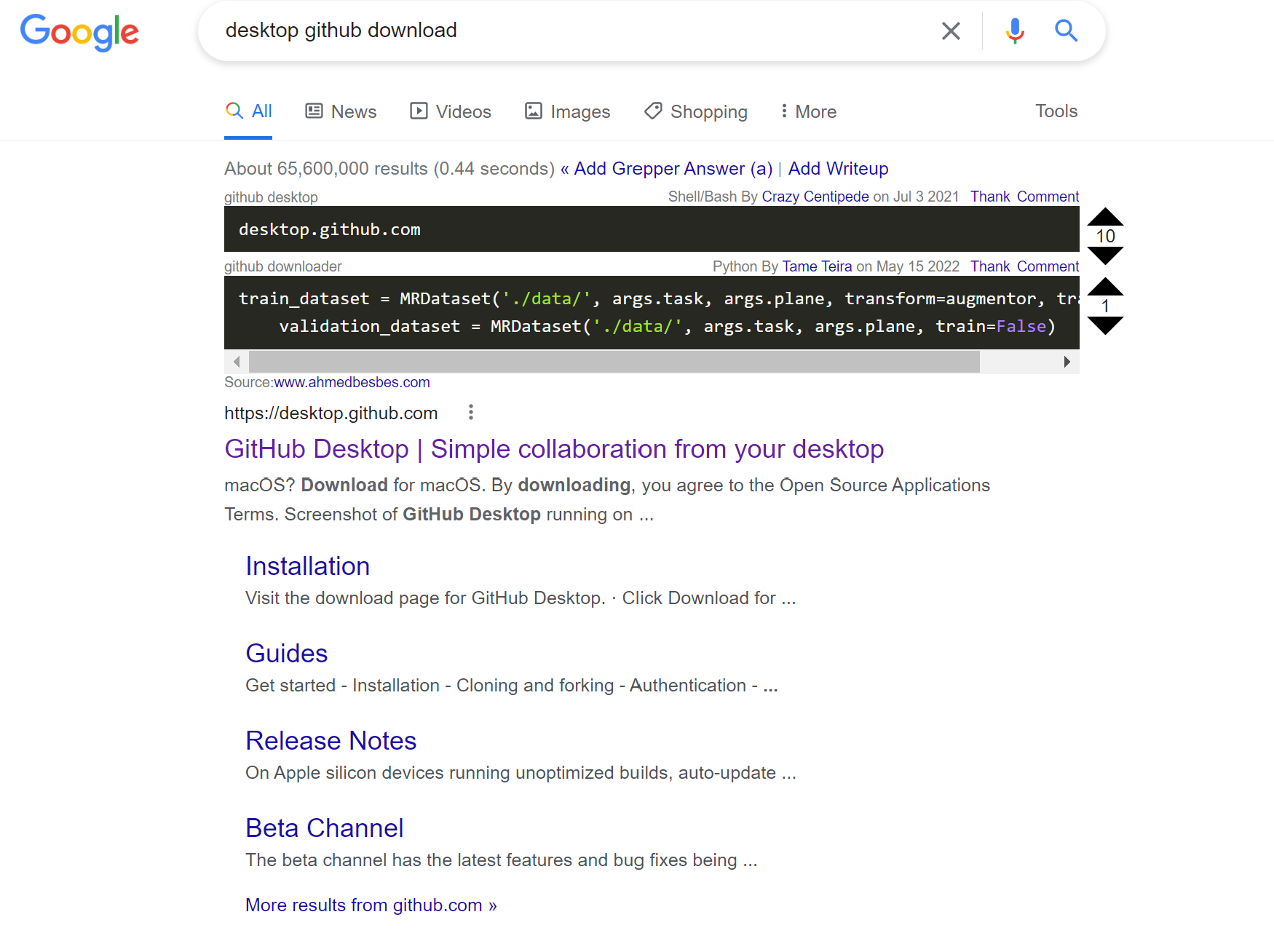
Figure 3. b: Created a new repository command line and pushed an existing repository command line

**Step 3)** **Download the GitHub Desktop if you don’t have it.**

GitHub Desktop is an application that enables a web developer to interact with GitHub using a GUI instead of the command line or a web browser.

* Flow steps one and two, then choose number two.







2

1

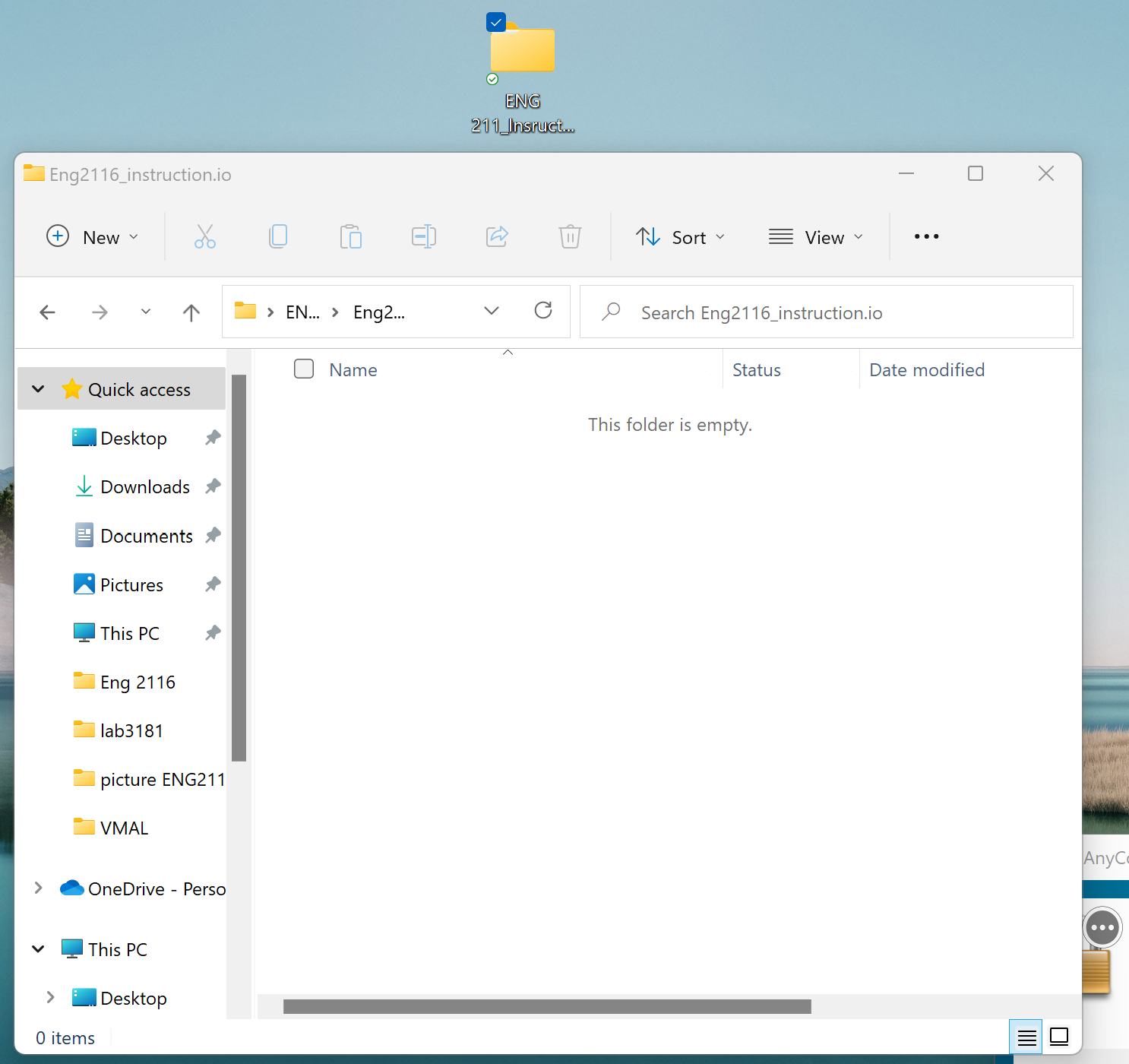


Figure 4: Download the Desktop GitHub

**Step 4) Create an empty new folder on your personal computer.**

Warning. Your folder must be empty because the git can clone only empty folders

* The folder must be empty



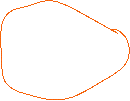
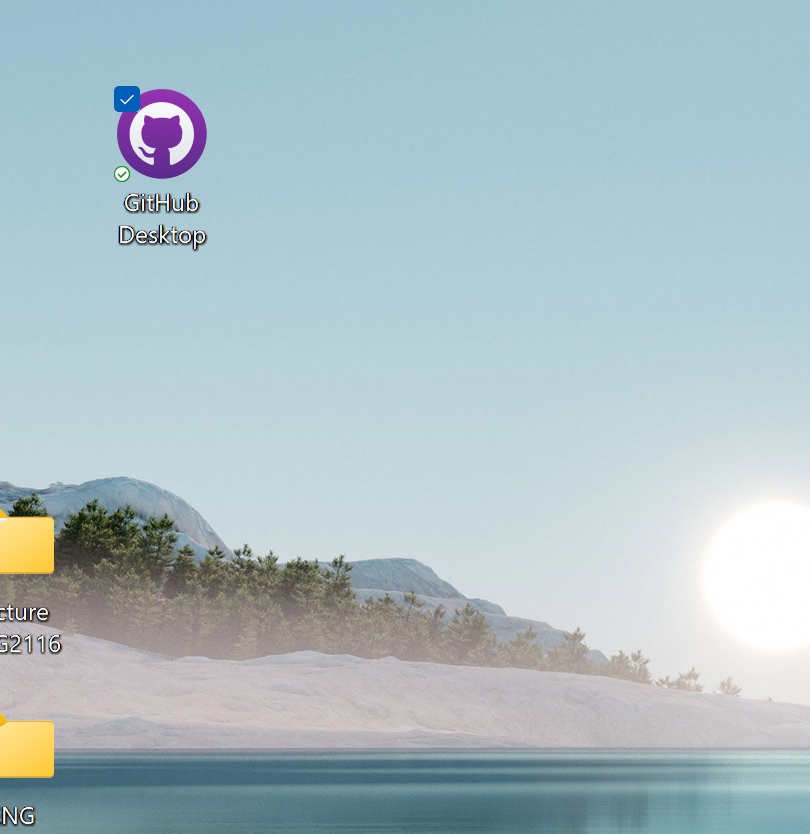
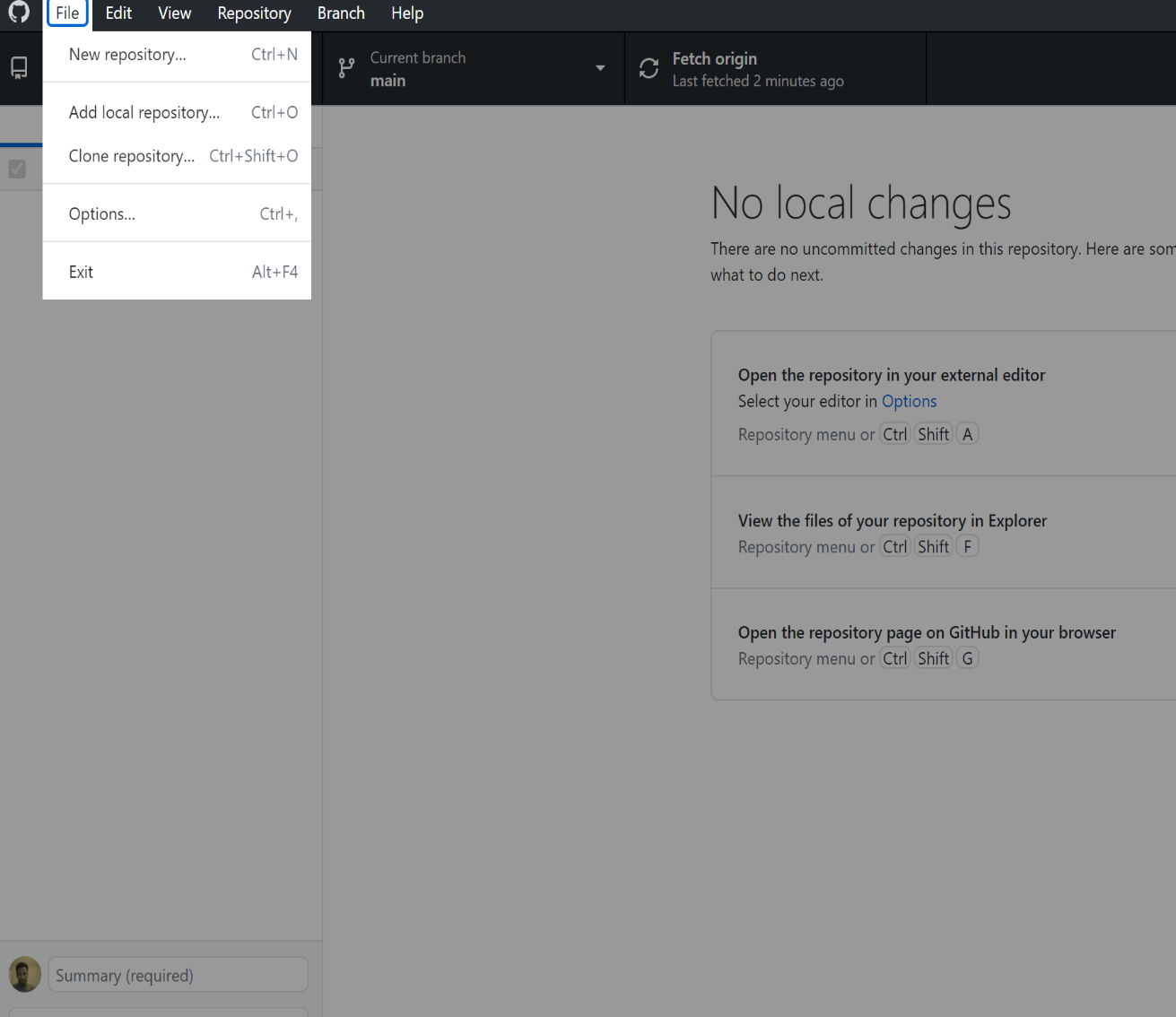


Figure 5: A new created empty folder in the personal computer

**Step 5) Clone the GitHub repository.**

1. Open Desktop GitHub from your computer, downloaded in (**Step 3)**.
2. Click on the file
3. Then click on clone repository



Step C



Step B

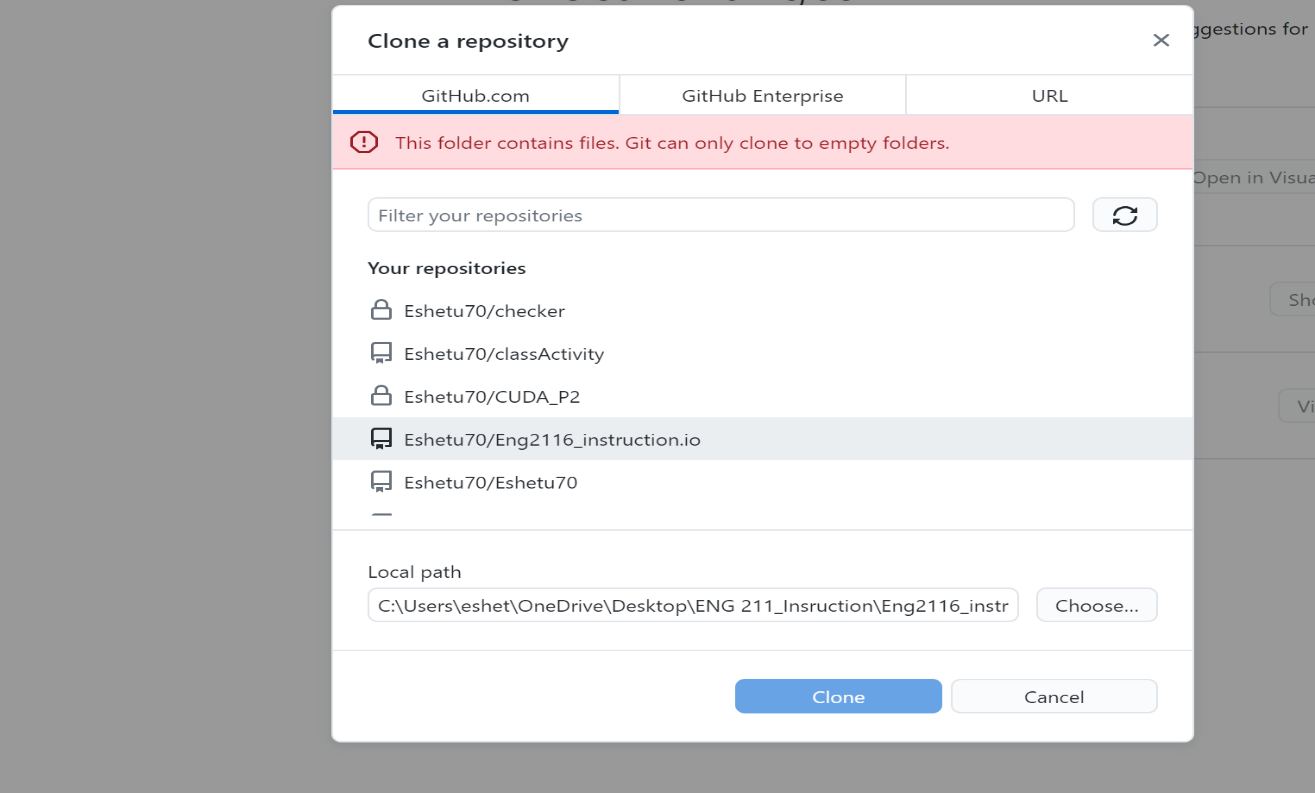


Step A

Figure 6: Open Desktop GitHub and choose clone repository

**Step 5. a) Choose the correct folder and path (see Figure 6. a).**

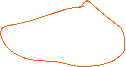
1. Select the repository folder you created in (**Step 2. a)**
2. Then select the empty folder you created in (**Step 4)** for your local path
3. Click the clone button





B

C



A



Figure 6. a: Choose correct files for repository and local workplace folder and confirm clone.

Now you have the GitHub repository folder inside the workplace folder that created **Step 4.** which includes git tools (see Figure6. b).

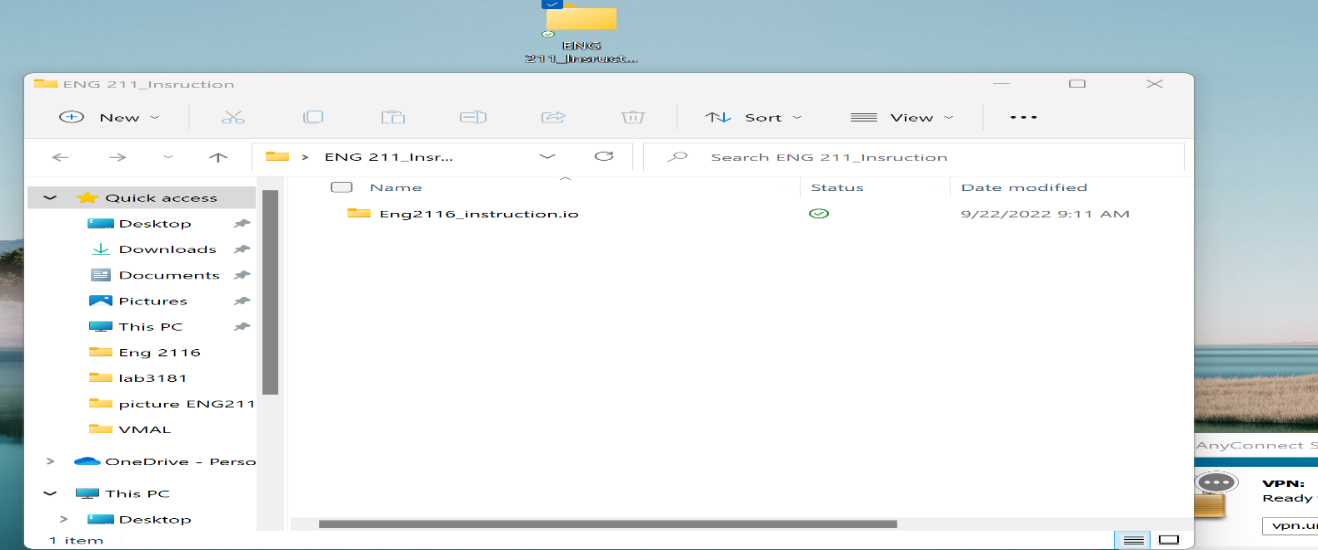


Figure 6. b: The cloned folder (repository) pops up in the workplace folder



**Step 6)** **Open the code editor and select the workplace folder.**



Open the visual video code from your computer; if you don’t have it, download it, it is a free resource for code editor

1. Right-click on the Open Folder icon



1. Then select the folders you created in **Step 4**

## 



B

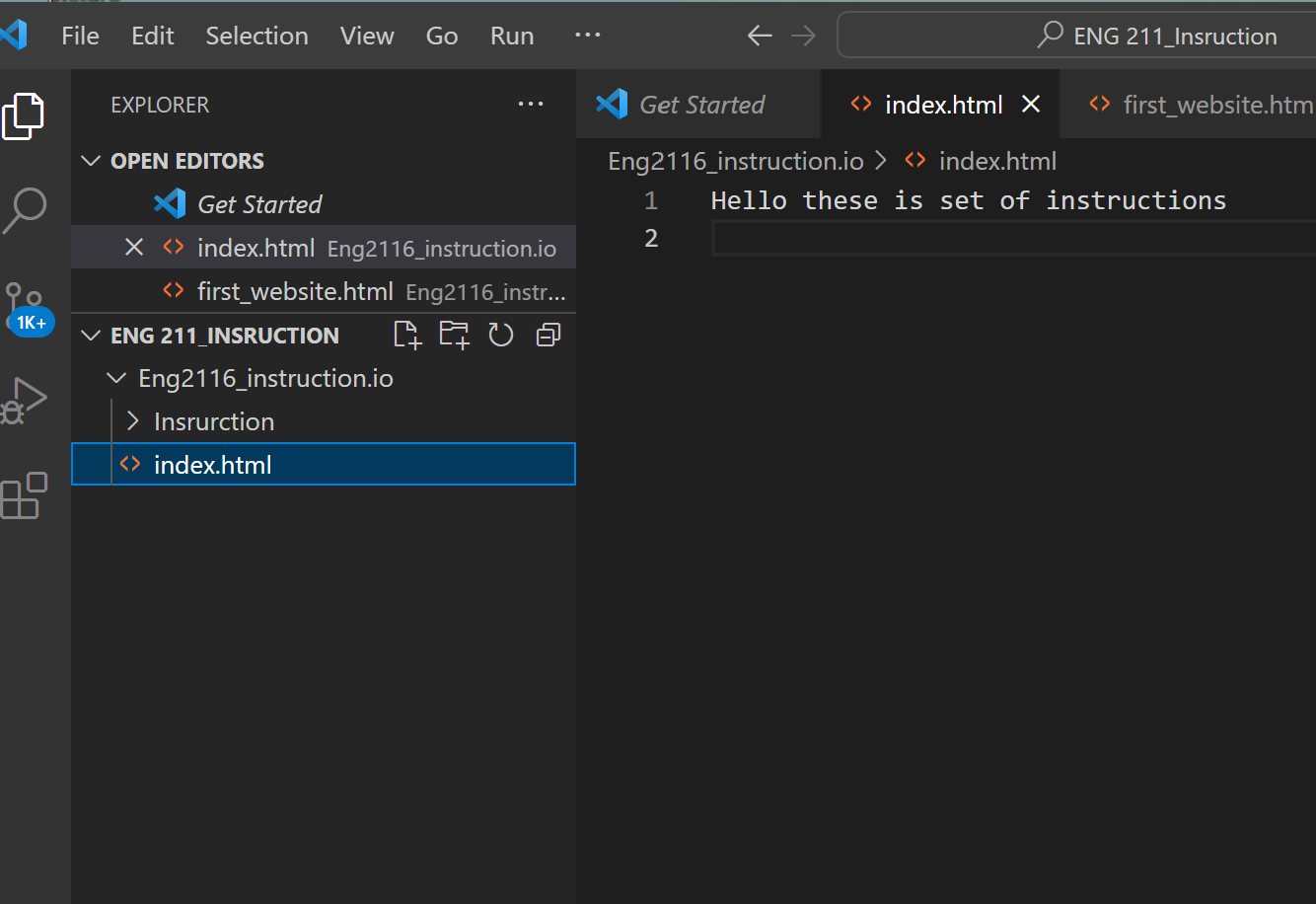
A



Figure 7: Indicates workplace folder and opened by visual video code

**Step 7)** **Create a new folder and index files Inside the repository folder**

* Create a new folder inside your repository folder that includes dot io (the Github.io folder cloned in **Step 5. a)**.
* Create an index.html file
* Then open index.html and write something you want.



Write something for your index files



Index files

This should be the new folder you created

Used for creating folders

Used for creating a files

GitHub.io folder (repository folder)

Figure 8: Visual video code with details description for creating files and folders

**Tips:** Create a Script, Styles, and Component folder inside the folder you created in **Step (7)**

Having those folders is effective in delivering clear designs for a website consistently.

* The style folder contains CSS files.
* The scripts folder contains JavaScript files.
* The component folder contains header.html and footer .html

Once created the component folder then, you don’t write the header and footer for all pages; you can copy and paste the header and footer to all pages using JavaScript.

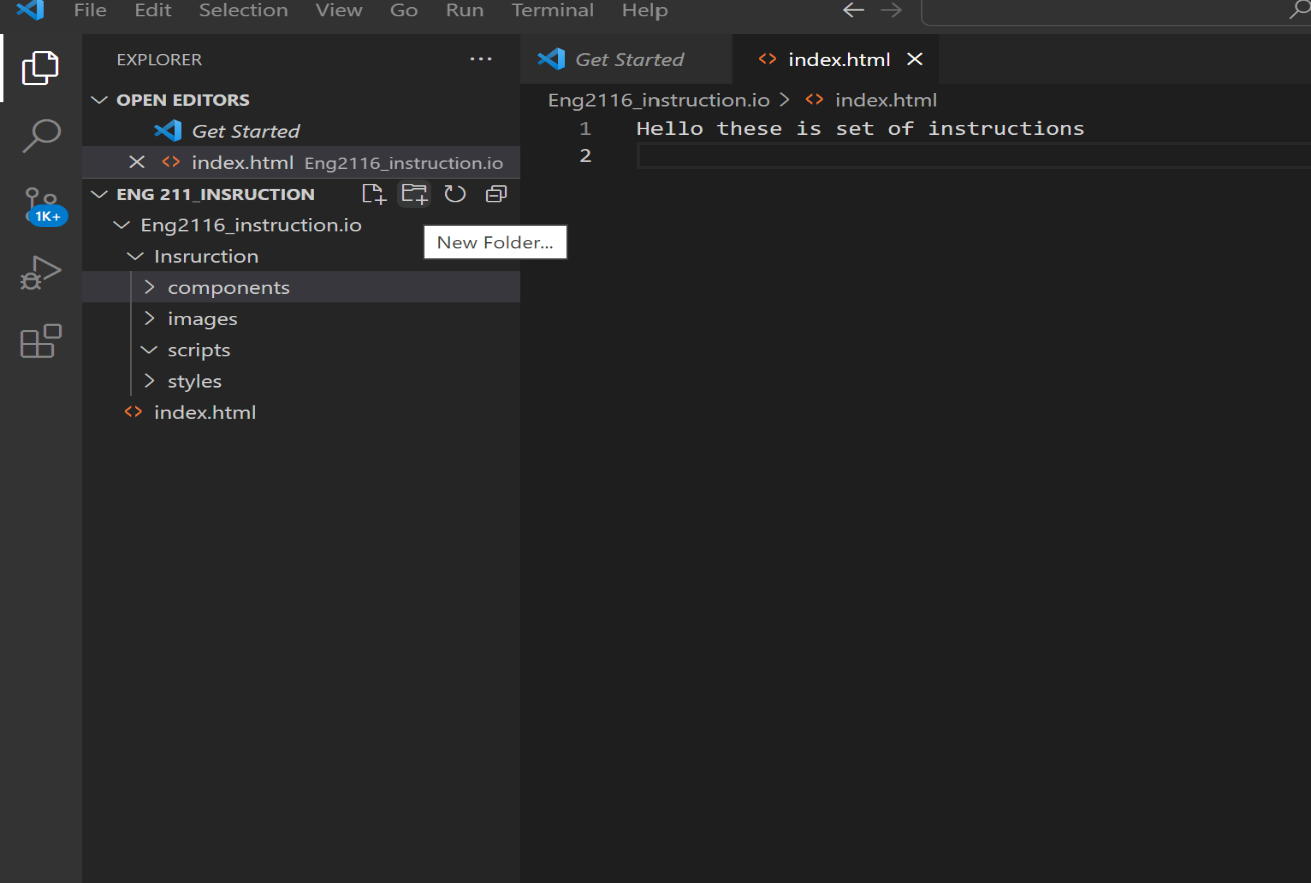


Figure 8. a: Created folders for scripts, styles, and components in the side workplace folder.

**Step 8) You are ready to start writing code for a web application.**

* Create your files name with all small letters and no space
* Start writing your code and save files (press Ctrl + S)

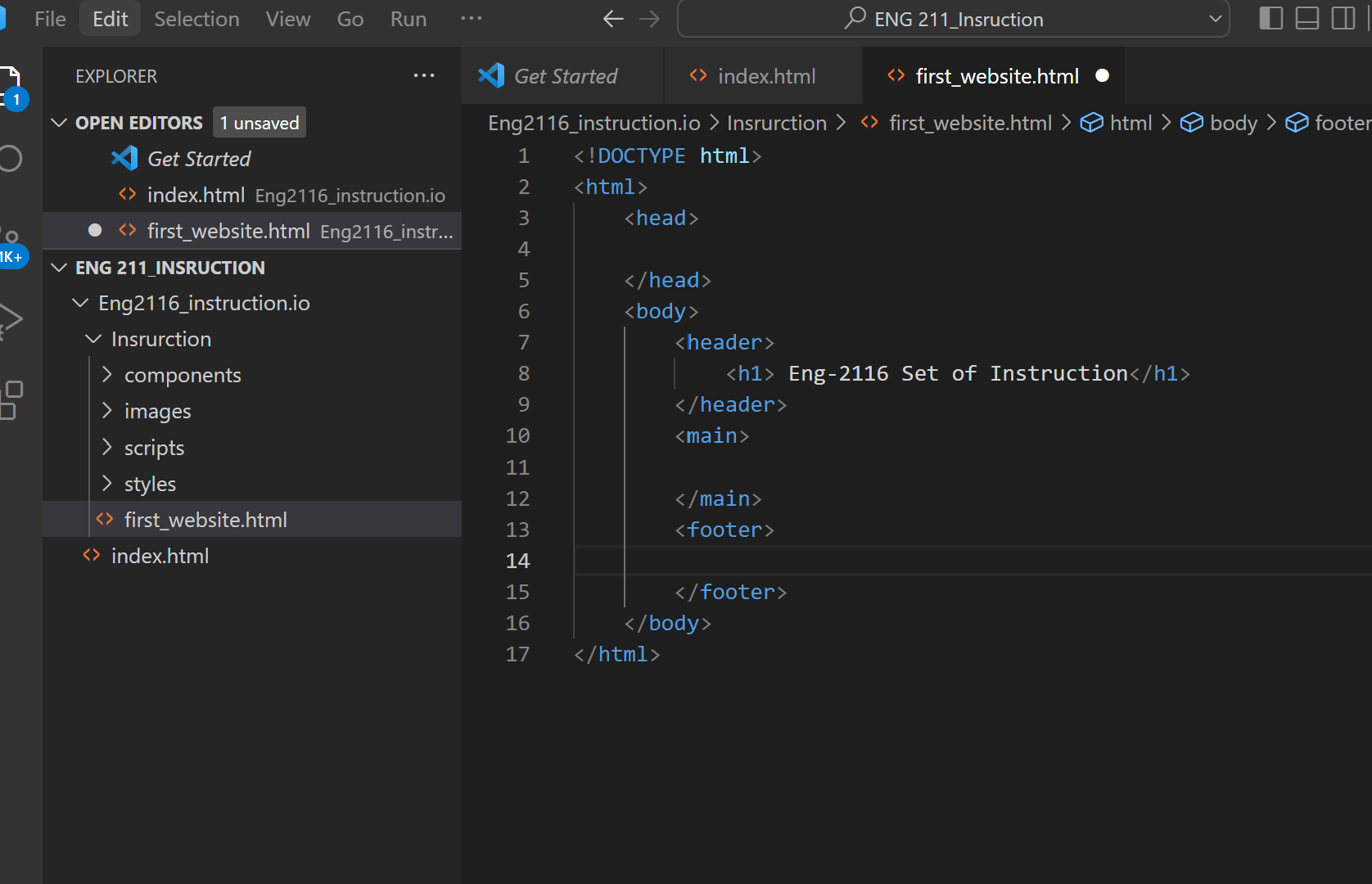
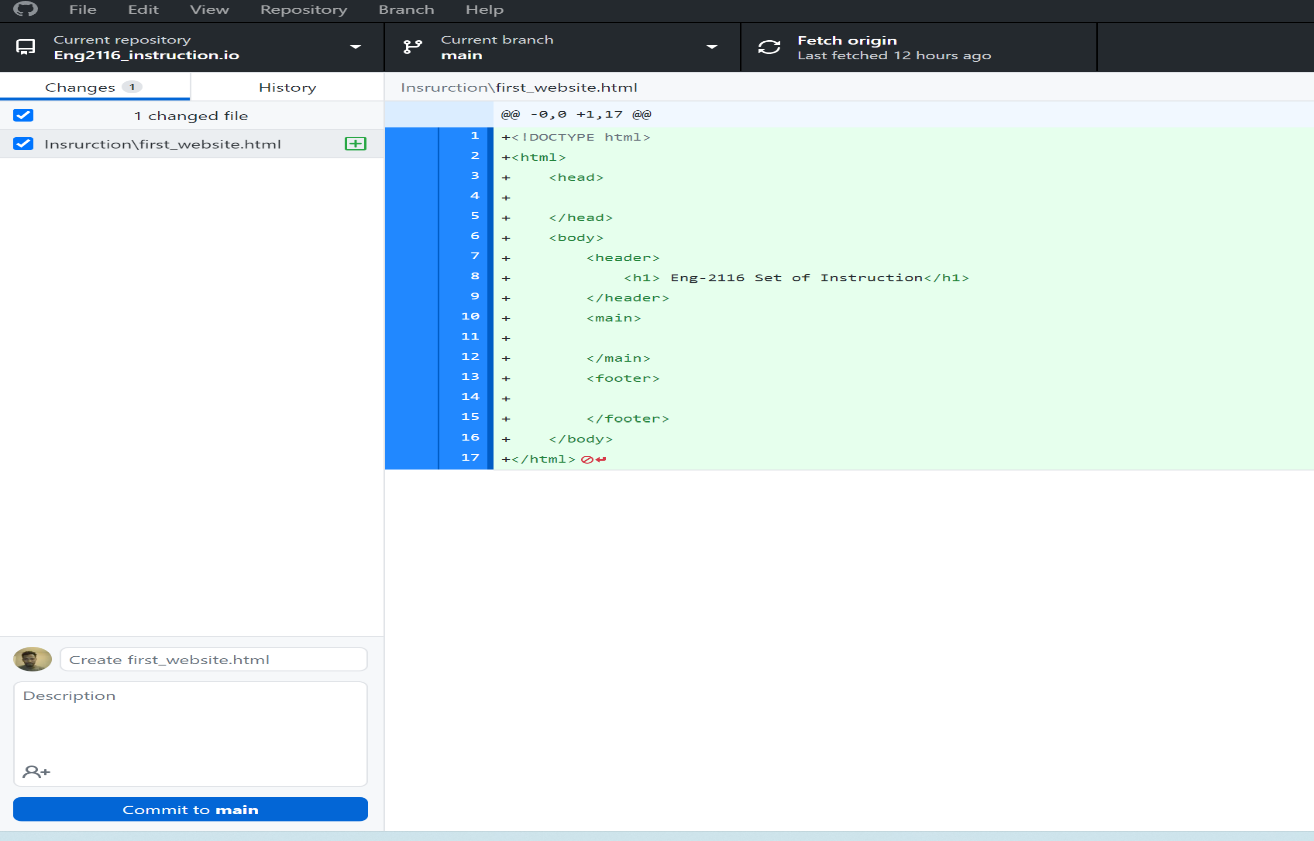




Figure 9: Create Html files and start writing a code

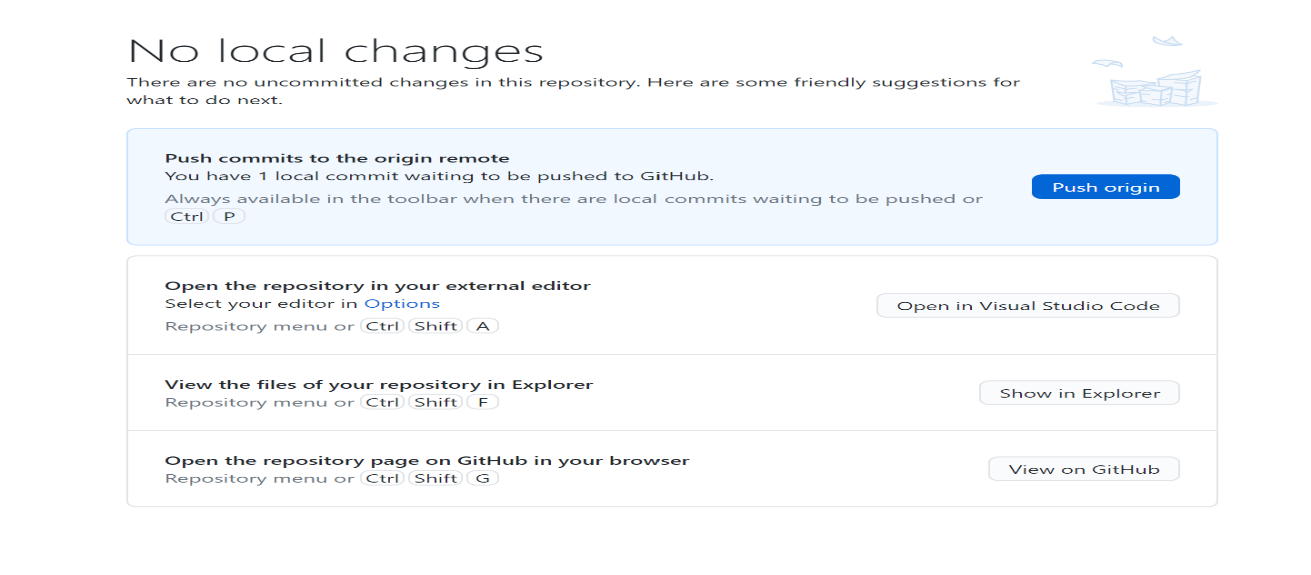
**Step 9) Open Desktop GitHub and commit and push Html files to the GitHub account.**

1. Click on the commit to the main button
2. Click on the Push origin button



A





B



Figure 10: Commit and push files to the GitHub account

You can easily upload your code files on the GitHub account and execute code from the GitHub website.