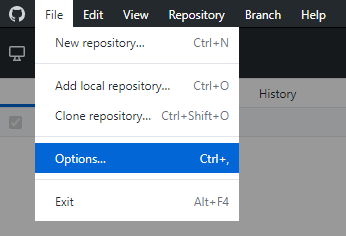
**Clone GitHub Remote Repository using SSH (and use GitHub Desktop to manage the repository)**

**Prerequisites**

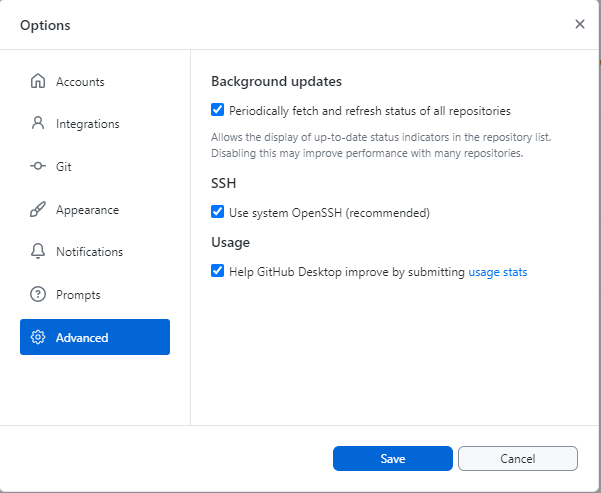
**Prerequisite 1**: Install [Git](https://git-scm.com/)

**Prerequisite 2**: Install [GitHub Desktop](https://desktop.github.com/) (Git GUI)

Step 1: Open GitHub Desktop and click on **File->Options** menu



Step 2: Click on Advanced tab and ensure that the **Use system OpenSSH (recommended) option** is checked and click **Save.**



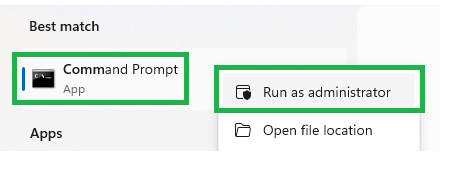
**Prerequisite 3**: Generate Private/Public SSH keys

**NOTE**:

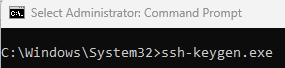
The instructions are for generating the SSH Keys on Windows.

If you are using another operating system, ensure that the instructions are adjusted accordingly.

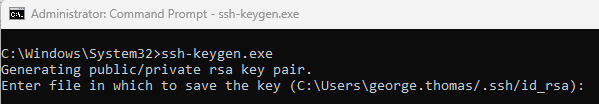
Step 1: Open the Windows Command Prompt in Administrative mode



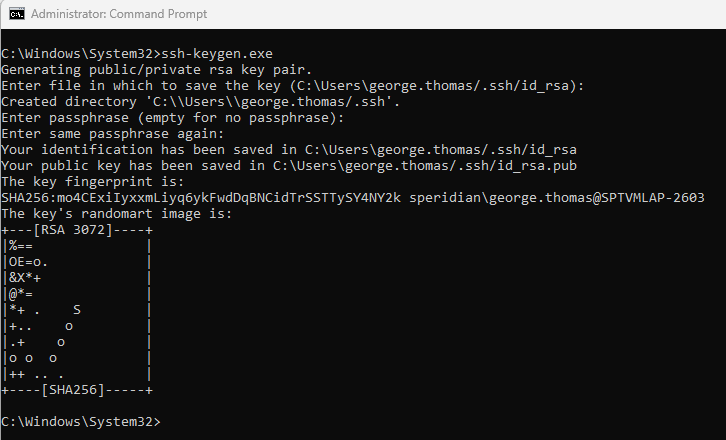
Step 2: Type **ssh-keygen.exe** and press enter.



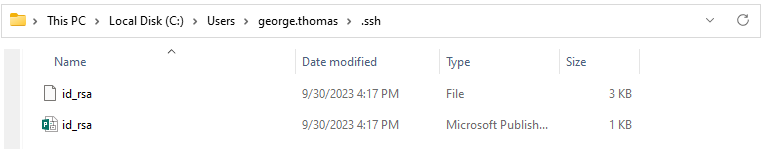
Step 3: Don’t make any change to the default file location. Just press enter.



Step 4: You will be prompted to enter a passphrase. It is not necessary to provide a passphrase. Press enter to reconfirm the passphrase. Again, press enter.



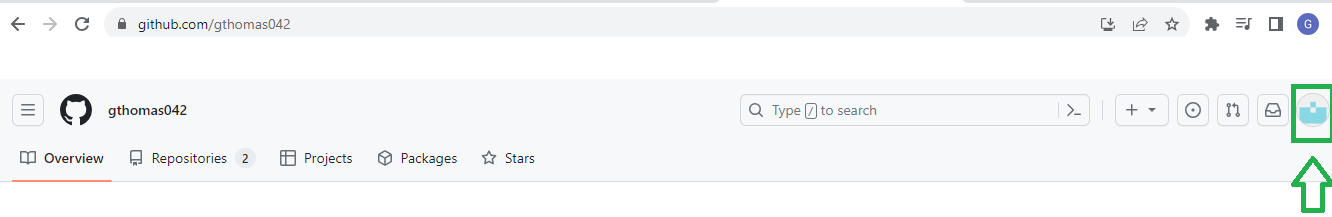
Step 5: Confirm if the SSH keys have been generated – There should be two files, a private key and a public key.



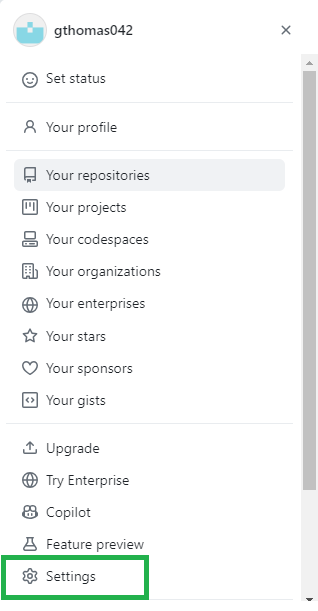
Step 6: Open the public key file (having extension **.pub**) in your text editor (Used to register the SSH against your GitHub account in the later part of the tutorial)

Step 7: Log into your GitHub Account

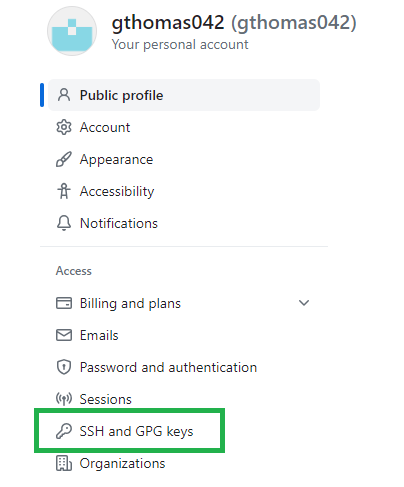
Step 7.1: Click on profile icon



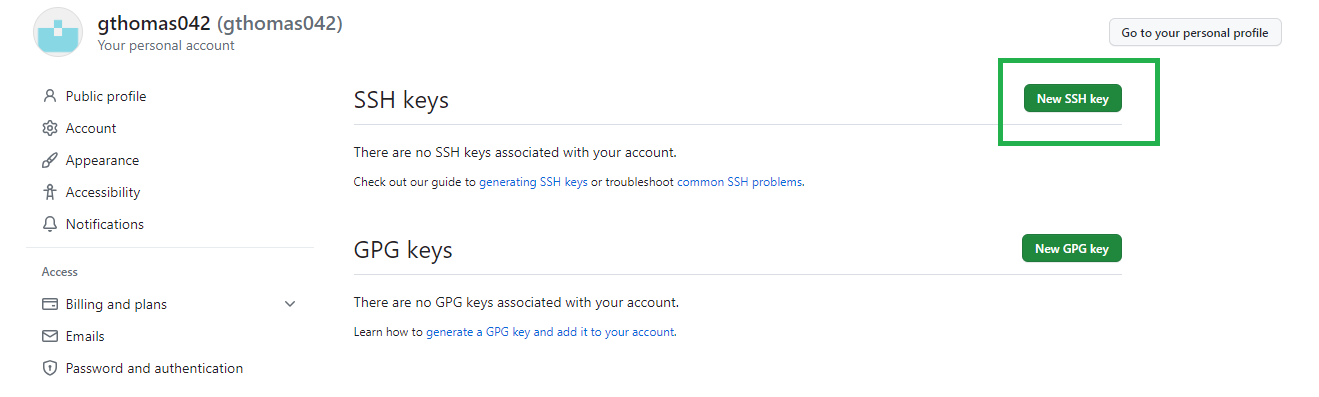
Step 7.2: Click on **Settings**



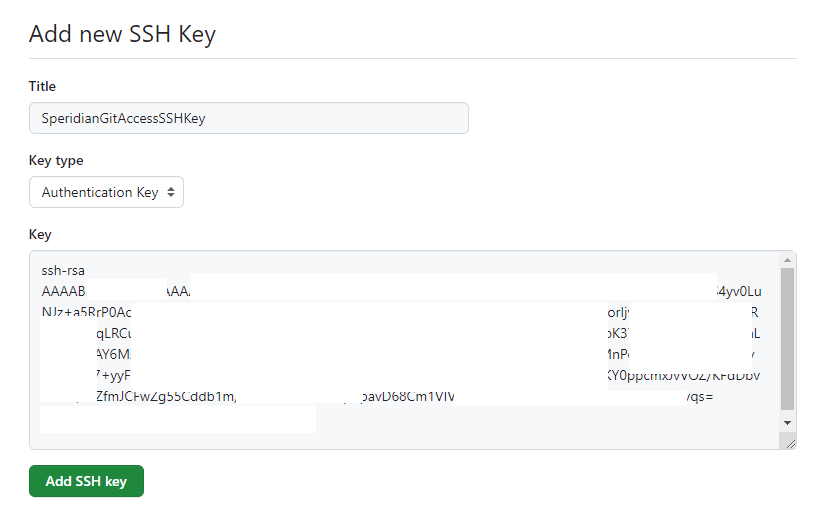
Step 7.3: Click on **SSH and GPG keys**menu



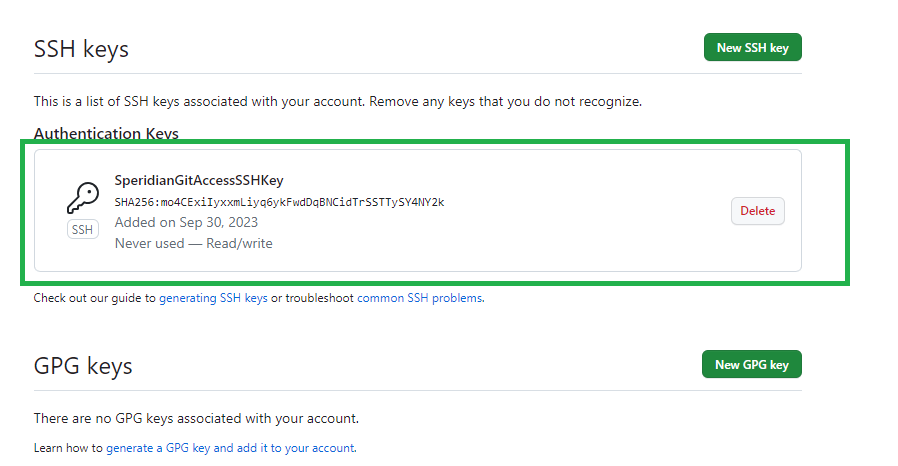
Step 7.4: Click on the “**New SSH key**” button.



Step 7.5: Give a suitable **Title** and copy the public key contents (refer Step 6) and paste it as the **Key** value. Let the **Key type** remain as **Authentication Key** and click **Add SSH key**

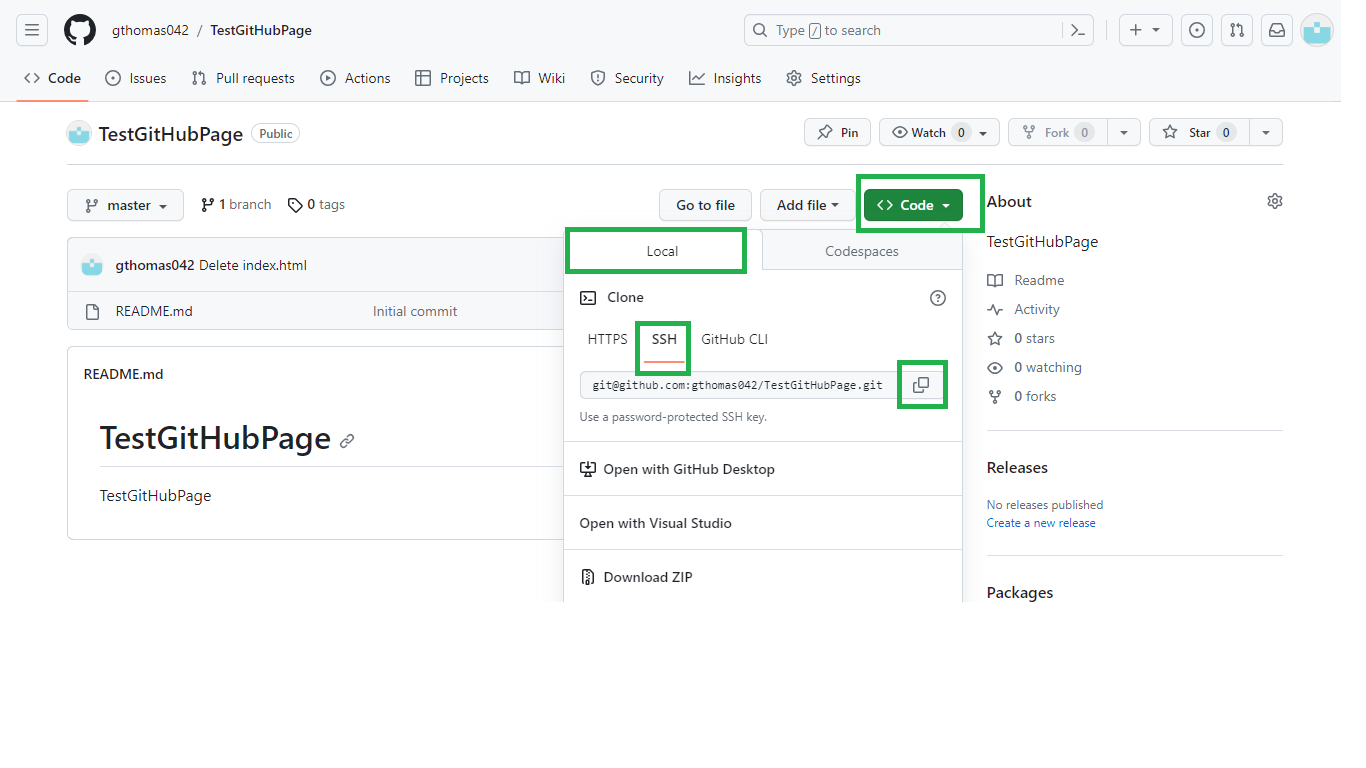


Step 7.6: Verify if the key comes under the **SSH keys/Authentication Keys**.



Step 7.7: Head over to the repository to be cloned and click the **Code** button.

Under **Local** tab, choose the **SSH** taband click the  icon to copy the SSH repository link.



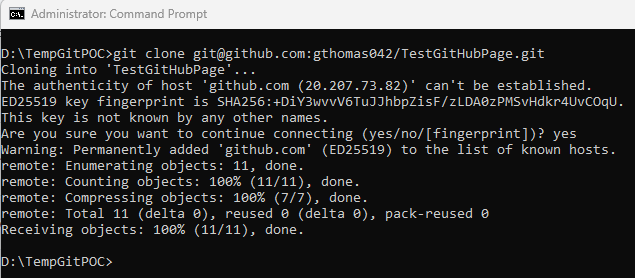
Step 7.8: Open Command Prompt and navigate to the local folder where the repo should be cloned.

**Rationale**: GitHub Desktop does not support the initial cloning process since it presents some prompts during this process so we resort to command line as in steps 7.8.1 and 7.8.2 to do the initial clone and henceforth take it forward with GitHub Desktop.

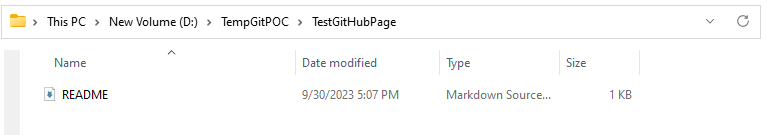
Step 7.8.1: Enter the command **git clone <REPOSITORY\_SSH\_LINK>**

**NOTE**: The local folder should have full permissions.

Step 7.8.2: For the prompt “Are you sure you want to continue connecting …..” type **yes** and press **enter.**

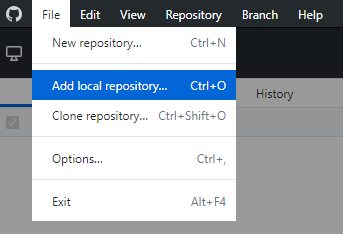


Step 7.9: Navigate to local folder to verify the files have been downloaded.



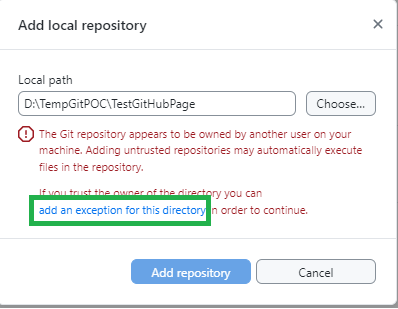
Step 8: Open GitHub Desktop and do the following steps

Step 8.1: Click on **File -> Add local repository…**

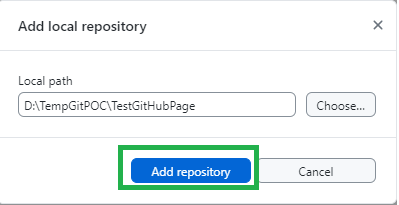


Step 8.2: Navigate to local repository folder. You will get the warning “The Git repository……..”.

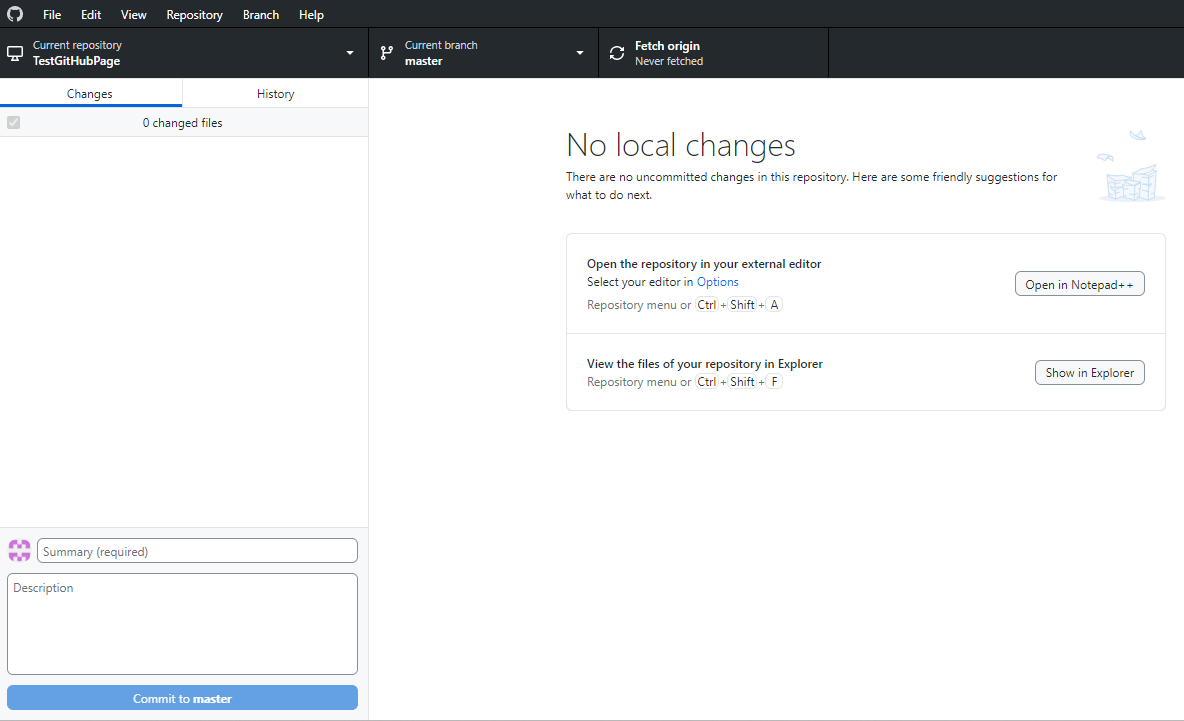
Click the “**add an exception for this directory**” link. The warning message will disappear.



Step 8.3: Click the **Add repository** button. Please be patient.



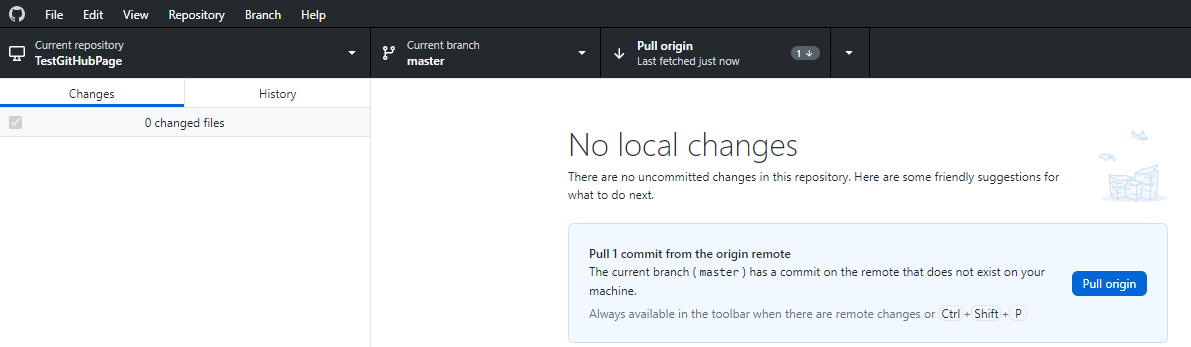
Step 8.4: GitHub Desktop recognizes any changes to the folder from now onwards.



Let’s test the changes

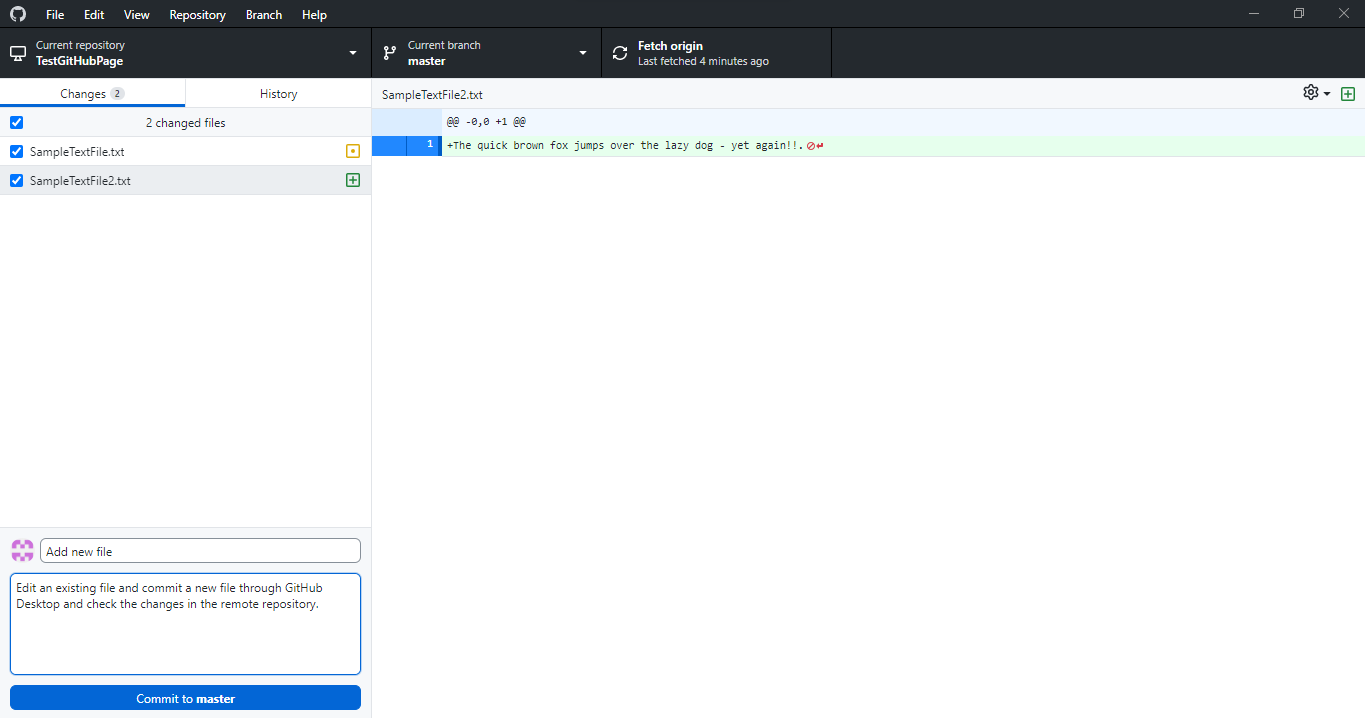
Test 1: Add file to GitHub remote repository and take a pull from GitHub Desktop.

Status: PASSED

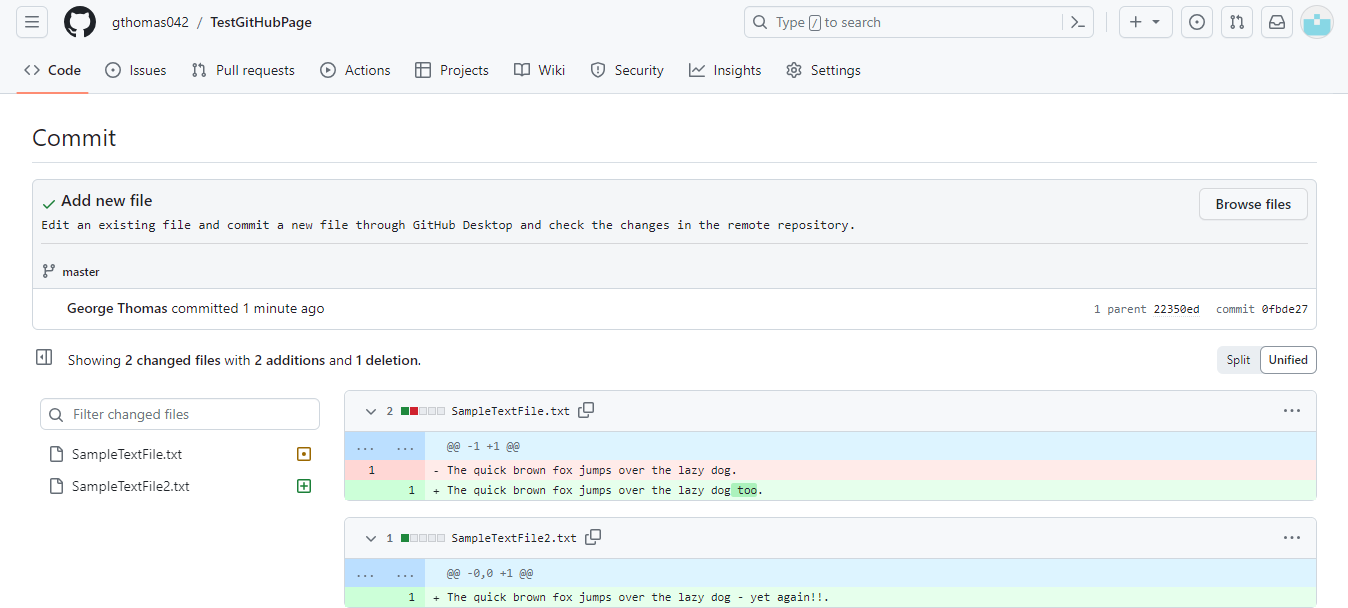


Test 2: Edit an existing file and commit a new file through GitHub Desktop and check the changes in the remote repository.

Status: PASSED

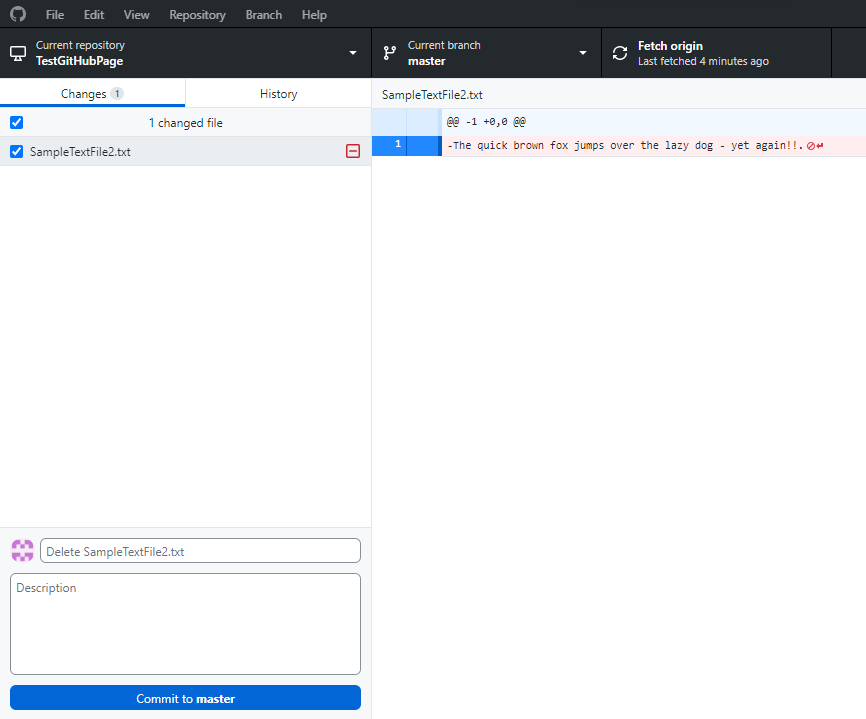


On successful commit, it reflects in the remote repository.



Test 3: Delete a file in the local repository folder, commit the change and check if it reflects in the remote repository.

Status: PASSED



On successful commit, it reflects in the remote repository.

