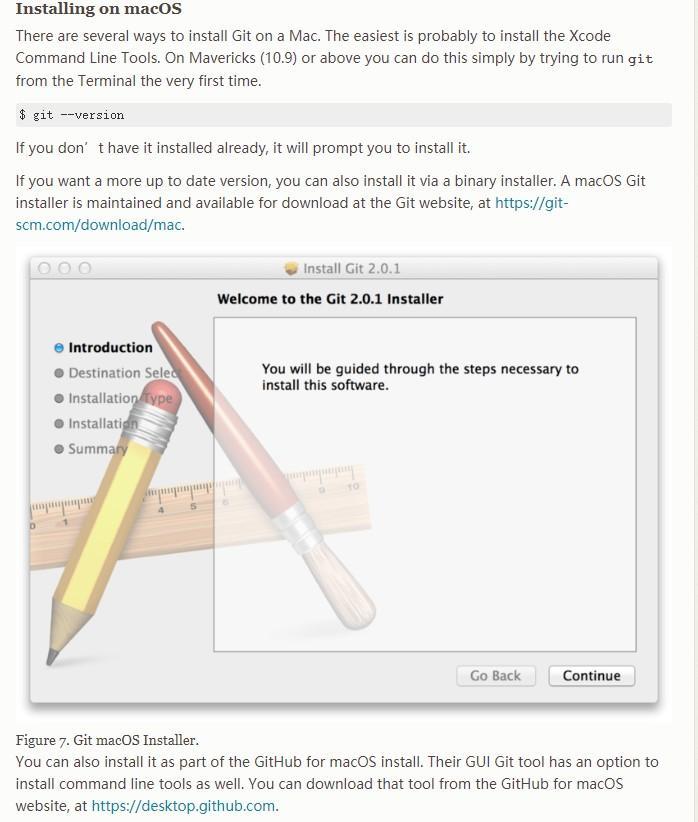
Lab # 2

# Git Download

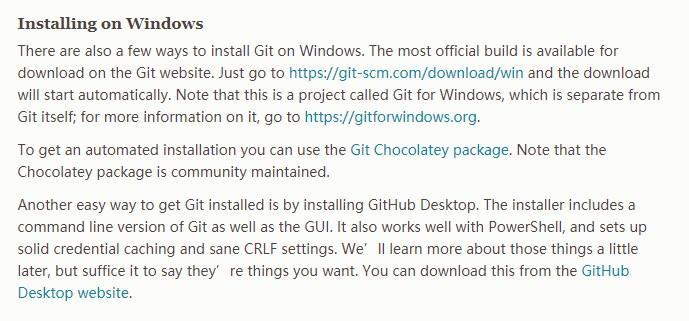
<https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>

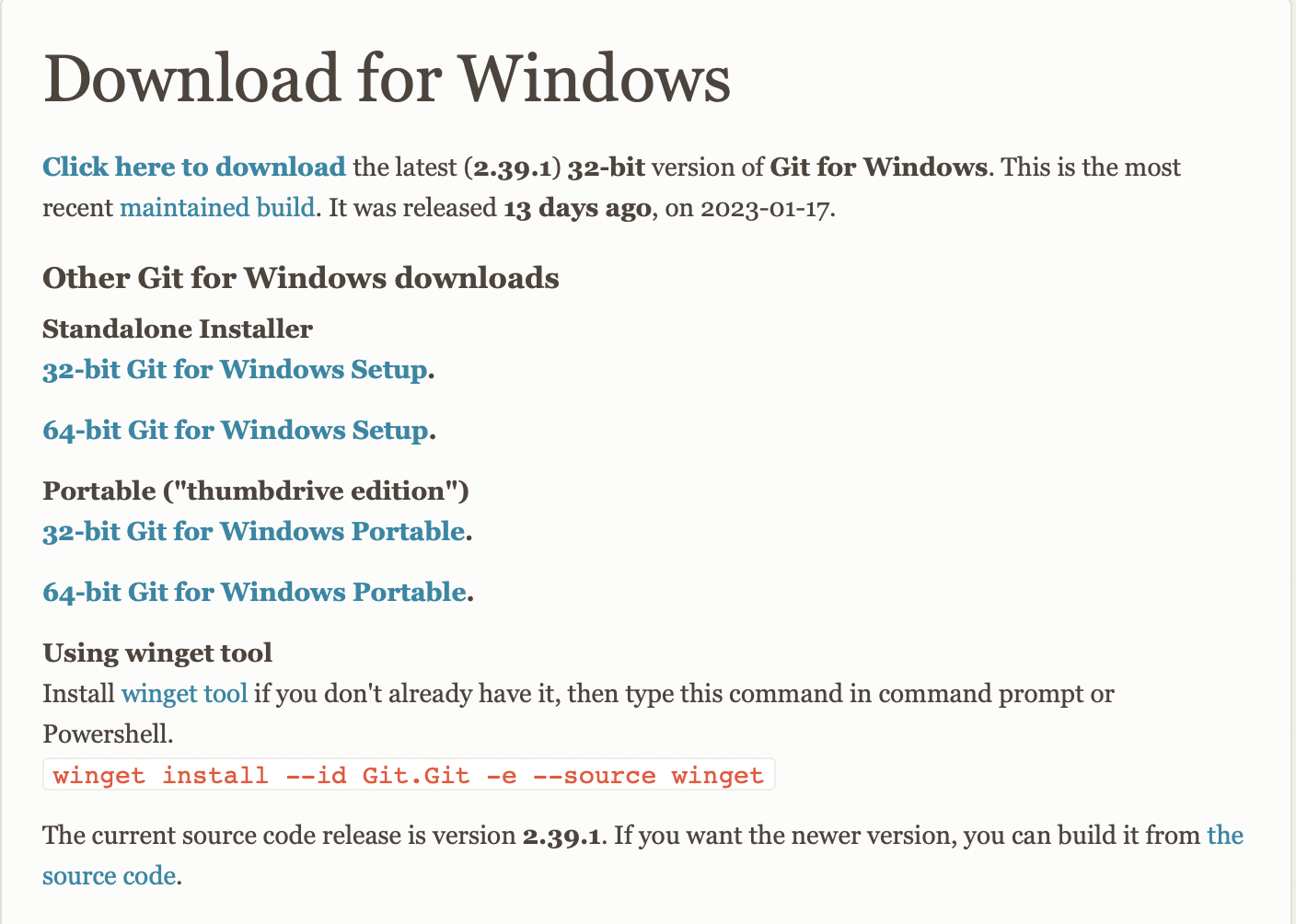


For Mac Users:



For Windows users:





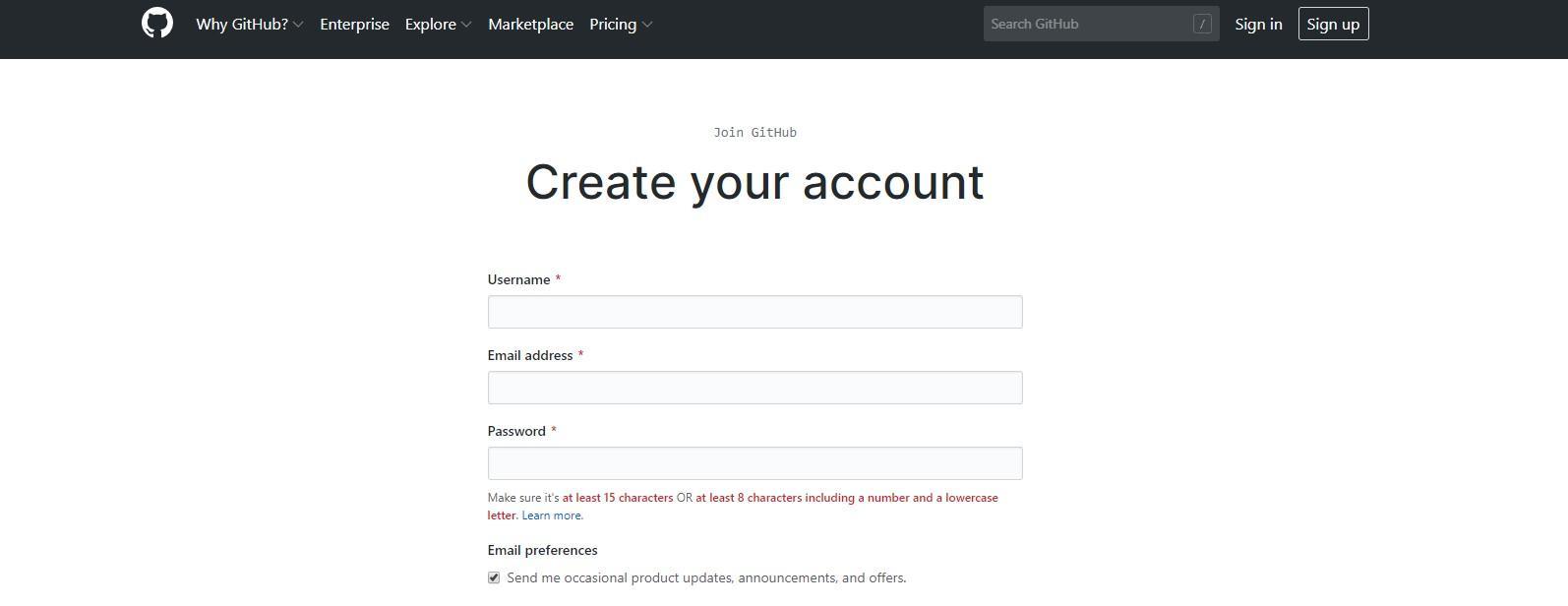
**Git Installation**

**Just follow the default settings**

# GitHub

Once you’ve down with Git, create a Free GitHub account at this link: <https://github.com/join>

Then, you will be navigated to the following webpage:



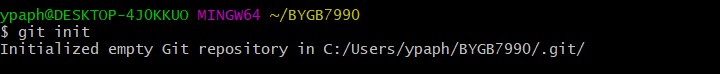
# Git Tutorial

## Step 1: Create a local git repository

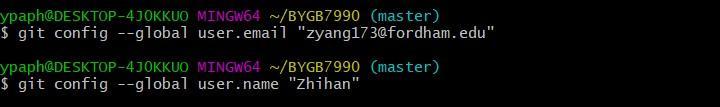
When creating a new project on your local machine using git, you’ll first create a new repository.

To begin, check the default working directory using the pwd command. Move to where you want to place the project on your local machine using the cd command.

To initialize a git repository in the root of the folder, run the git init command.



## Step 2: Registration



## Step 3: Add a new file to the repository

Go ahead and add a new file to the project using any text editor you like or just running a touch command.

However, you could always copy and paste folders or files directly to your git working directory.

To add files in the repo use command **git add .**

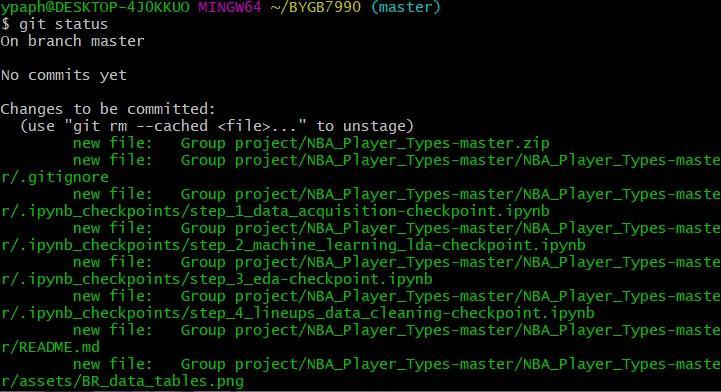
It will add all the files to git repo

Graphical user interface, application

Description automatically generated

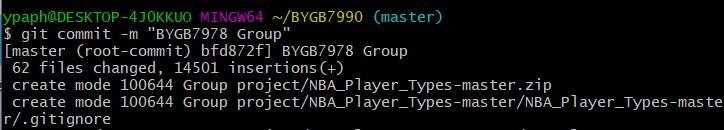
Once you've added or modified files in a folder containing a git repo, git will notice that changes have been made inside the repo. Git won't officially keep track of the file (that is, put it in a commit - we'll talk more about commits next) unless you use the add command.

Then, you can use git status command to check the status.



## Step 4: Create a commit

Use the git commit -m “logging” command to commit your previous changes. Notice here “logging” can be changed to whatever you want to input.



## Step 5: Create a new branch

Branches allow you to make a new feature but are worried about making changes to the main project to the main project while developing the feature. This is where git branches come in.

Branches allow developers to move back and forth between ‘states’ of a project. For instance, if you want to add a new page to website you can simply create a new branch without affecting the main part of the project. Once you’re done with the page, you can then merge changes from your branch into the master branch.

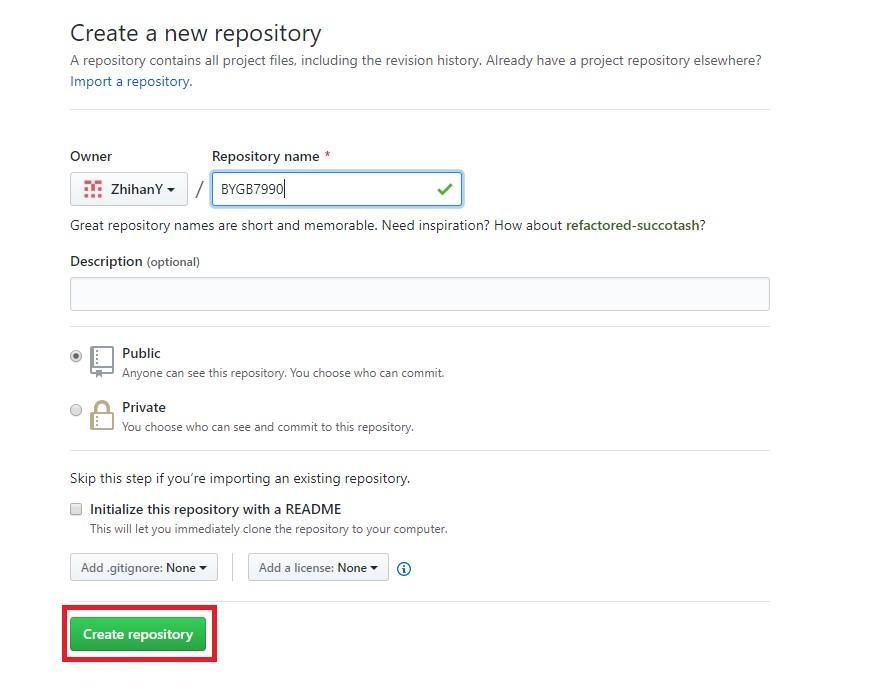
To create a new branch, run git checkout -b <branch name>, and git creates a new branch and automatically switch to that new branch. You can also use git branch <name> to create a new branch.

## Step 6: Create a new repository on GitHub & Push a branch to GitHub

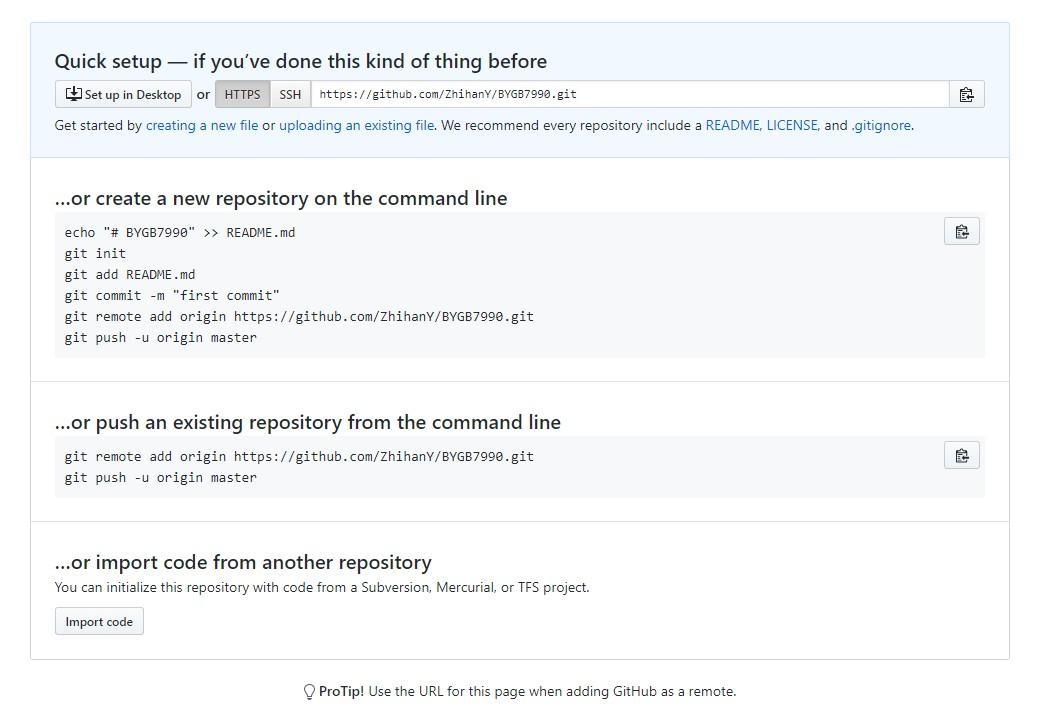
If you only want to keep track of your code locally, you don't need to use GitHub. But if you want to work with a team, you can use GitHub to collaboratively modify the project's code.

To create a new repo on GitHub, log in and go to the GitHub home page. You should see a green ‘New’ button:

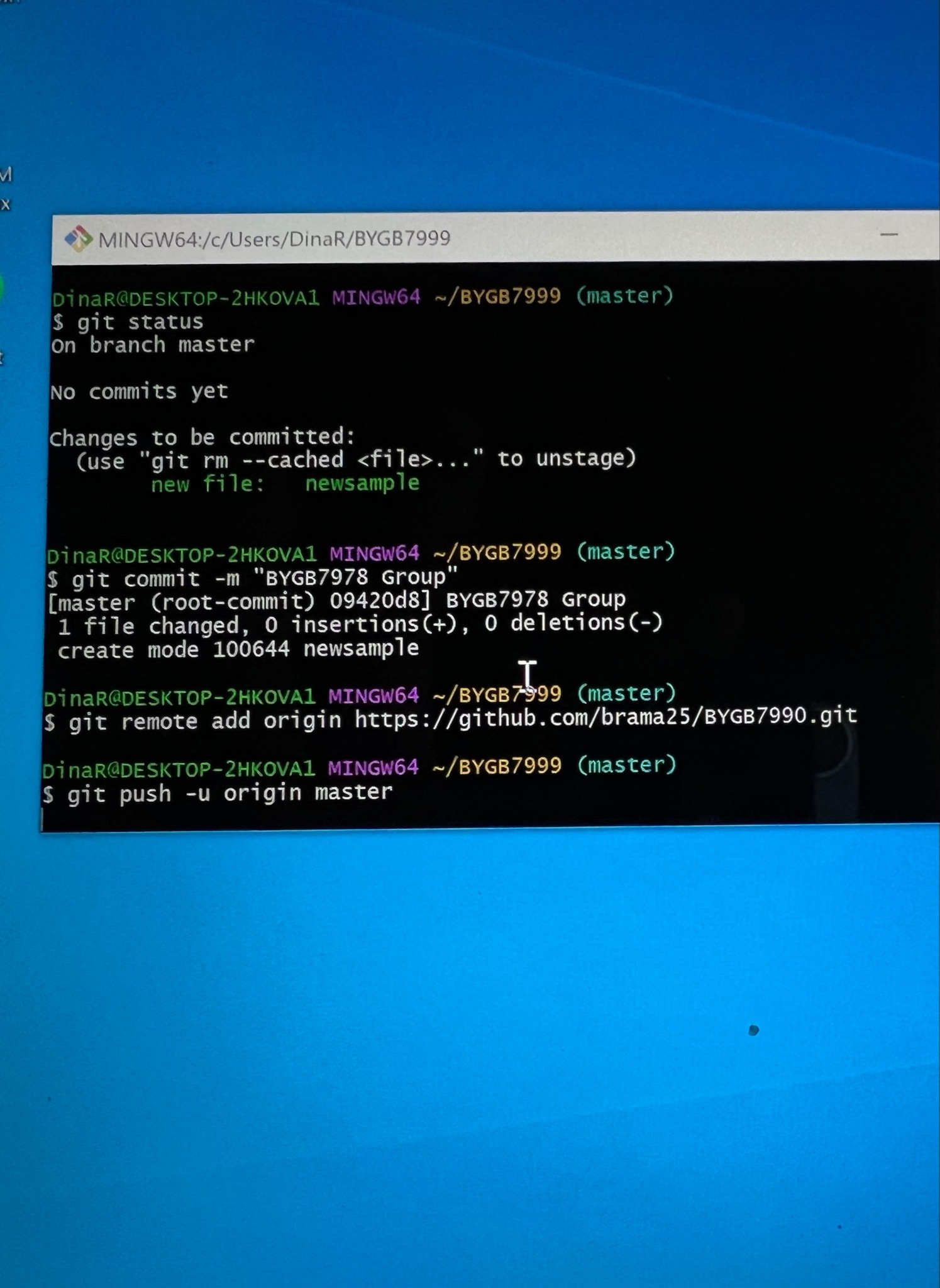




After Setting up your repository, you will see a page like this:

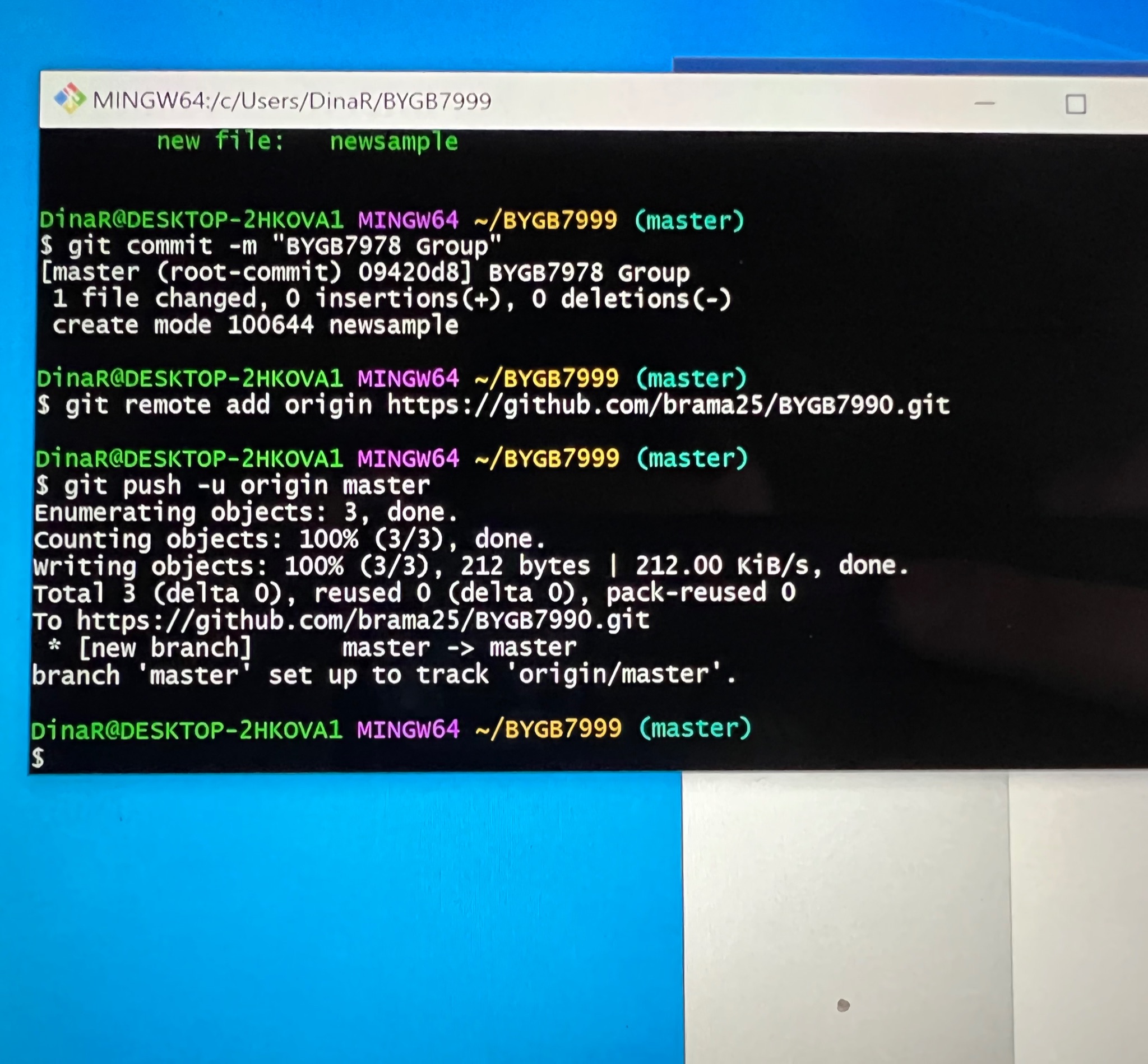


Then, type the commands shown in the middle section of the previous picture (Yours are different from mine):

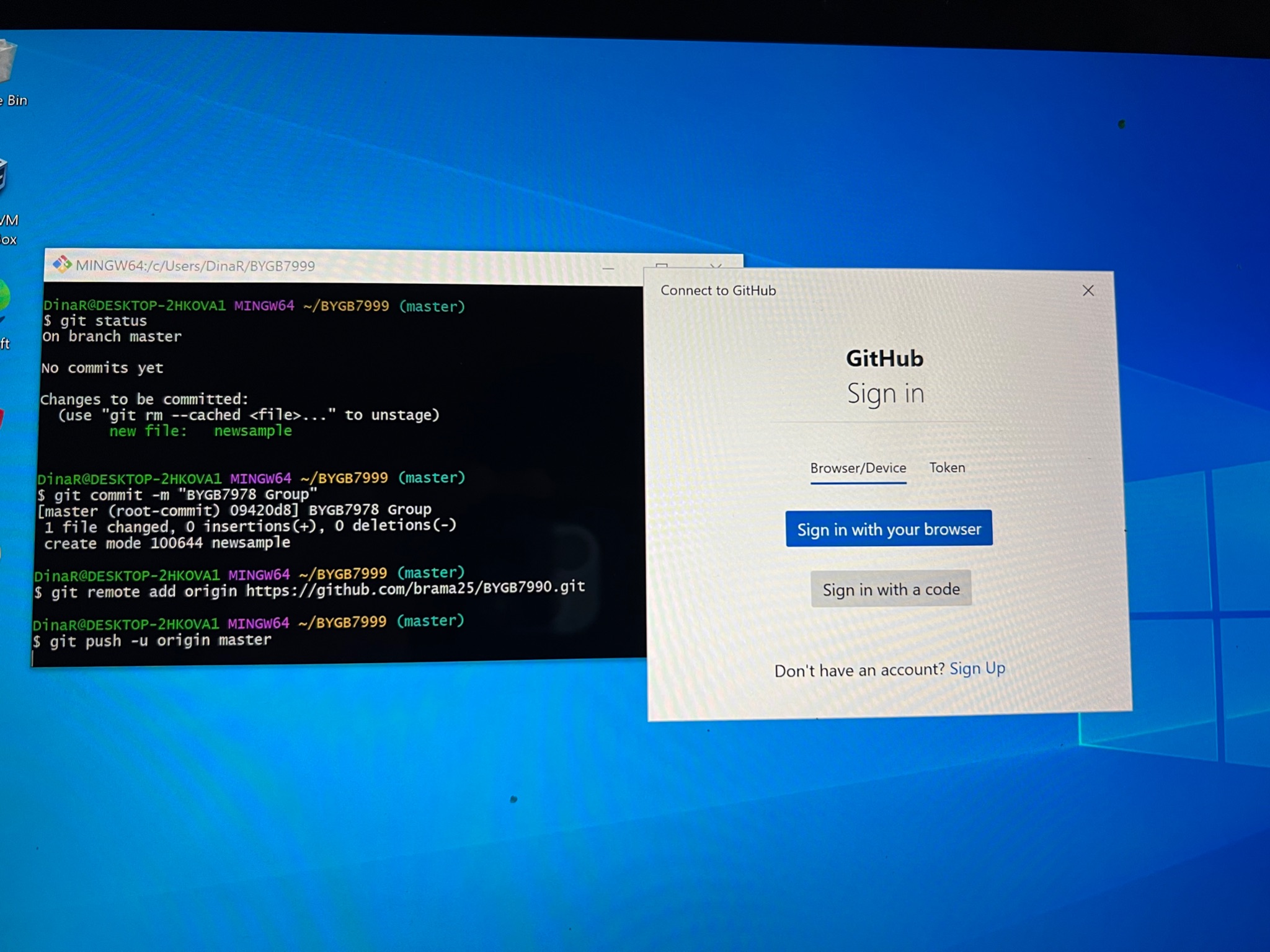


The reason why there is a -u command in the second row is that your remote repository is now empty. Next time, when you upload documents, you won’t need to use that so the command will be:

$ git push origin master

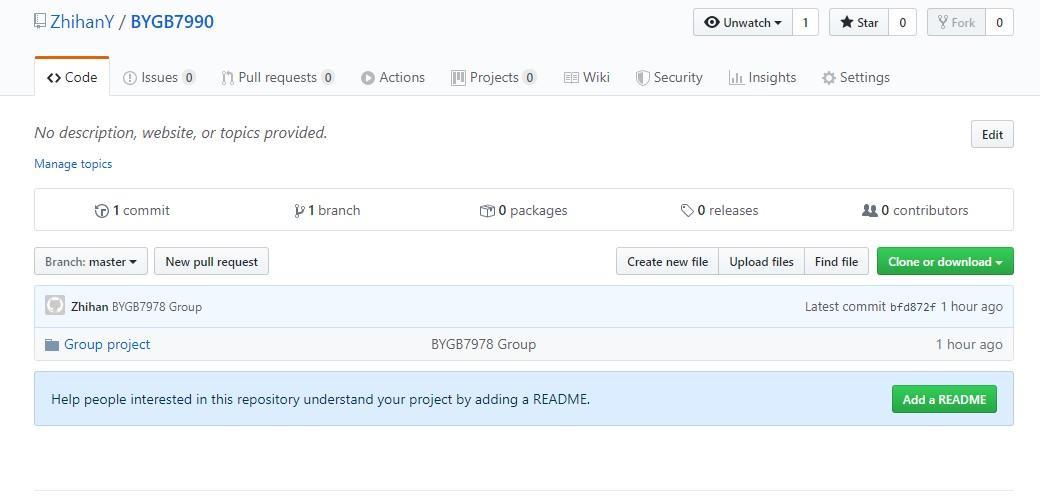


It will pop out a GitHub login window, enter your registered email address and password to sign in.



After implementing the code before, you will see the following sign similar to the picture above. Remember to click the sign in with your browser. You will then go to an authentication page meaning that the sign in is complete and you will go back to Github. (Remember to refresh if nothing has changed)

Ultimately, you will see something like this:



## Step 7: Write a README file

Open source community is growing rapidly. Developers release new open source projects on GitHub every day. As a result, it’s becoming more and more difficult to get your own project and stand out from the community. However, you can do a few things to increase your chances of grabbing other developer’s attention. One effective and simple technique is putting up a nice-looking and helpful README file.

Good README should include enough details to help a new user get started, e.g. how to compile, how to install, and how to starting integrating.

Lastly, after learning about the writing fundamentals of README, let’s dive into the styling of README which also called formatting.

Formatting is an essential part of README. You can learn about how to format your README from [here a](https://github.com/github/markup/blob/master/README.md)nd [here.](https://help.github.com/en/github/writing-on-github/basic-writing-and-formatting-syntax) README Templates:

<https://gist.github.com/fvcproductions/1bfc2d4aecb01a834b46><https://github.com/dbader/readme-template>

## Step 8: Create a pull request

A pull request is a way to alert a repo’s owner that you want to make some changes to their code. It allows them to review and make sure it looks good before putting your changes to the master branch.

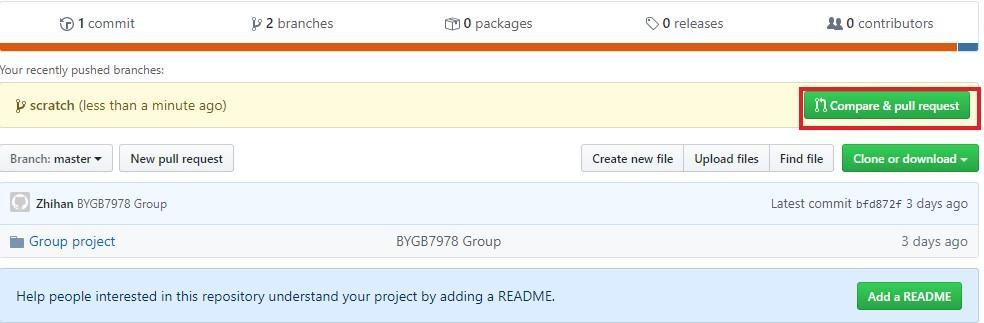
Let’s make a new local branch and then create a readme file:

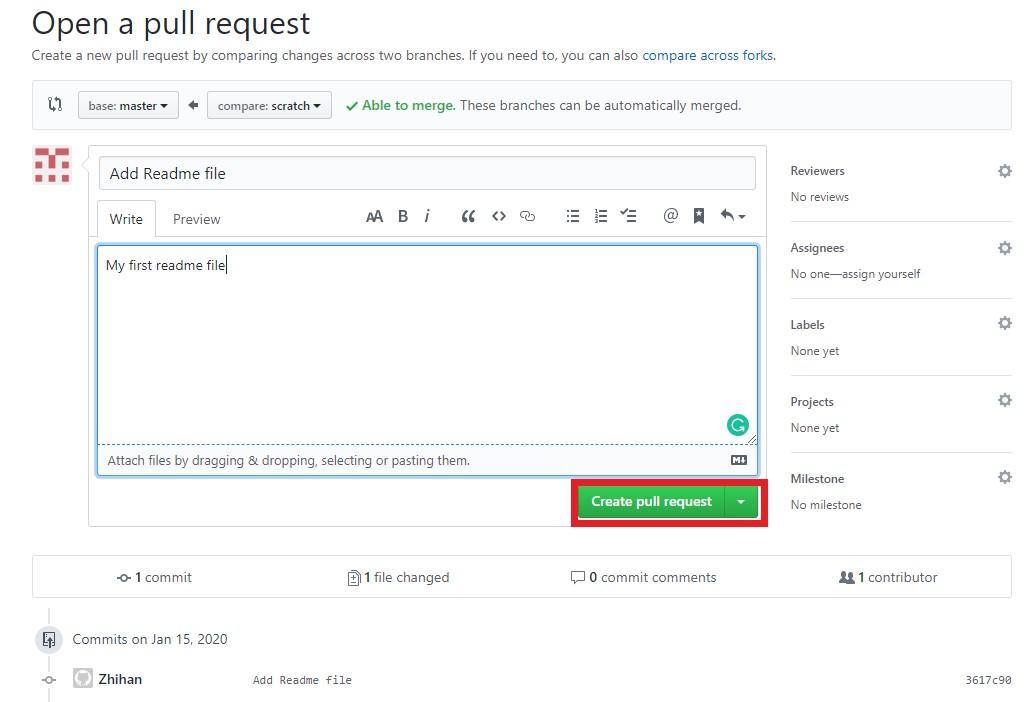


Push the new branch to GitHub:



Then you will see a Compare & pull request button where you can create your pull request:





A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generatedText

Description automatically generated