GIT AND GITHUB

GitHub account creation

To create your account, you need to go to **GitHub**'s website and fill out the signup form.

Git installation

Now we need to install Git's tools on our computer. You can refer the following URL to downloading git on your operating system **https://git-scm.com/downloads**

Git for Mac Installer

Install Git with Homebrew

If you have **installed Homebrew** to manage packages on OS X, you can follow these instructions to install Git:

1. Open your terminal and install Git using Homebrew:
   ```
   $ brew install git
   ```
2. Verify the installation was successful by typing :
   ```
   $ git --version
   ```

Install Git on Windows

Git for Windows stand-alone installer

1. Download the latest **Git for Windows installer**.
2. When you've successfully started the installer, you should see the **Git Setup** wizard screen. Follow the **Next** and **Finish** prompts to complete the installation. The default options are pretty sensible for most users.
3. Open a Command Prompt (or Git Bash if during installation you elected not to use Git from the Windows Command Prompt).

Install Git on Linux

Debian / Ubuntu (apt-get)

Git packages are available via apt:

1. From your shell, install Git using apt-get:
```
$ sudo apt-get update
$ sudo apt-get install git
```
2. Verify the installation was successful by typing:
```
$ git --version
```

The final step after setup step is to enter your name and email

Configure your Git username and email using the following commands, replacing Thanmaisai name with your own and enter your email/gmail used to create the GitHub account. These details will be associated with any commits that you create:

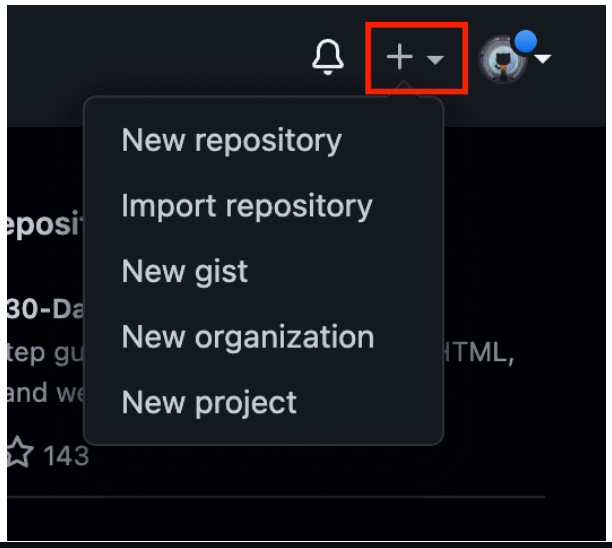
```
$ git config --global user.name "Thamaisai"
$ git config --global user.email "Thanmaisai@gmail.com"
```

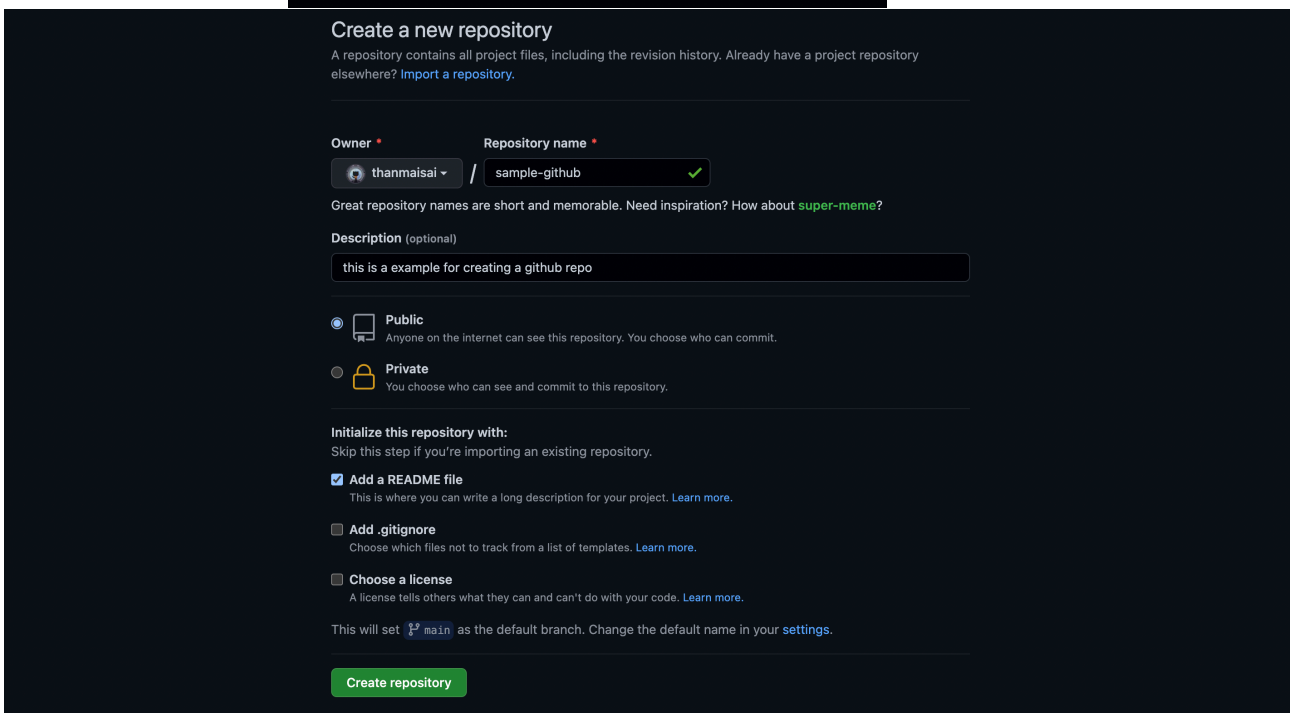
Working over git and GitHub can be in described in two ways such as:

1. Create the repository, clone it to your PC, and work on it.
2. Work on your project locally then create the repository on GitHub and push it to remote.

Create the repository, clone it to your PC, and work on it.

1. Create a new repository by clicking the “new repository” button on the GitHub web page.

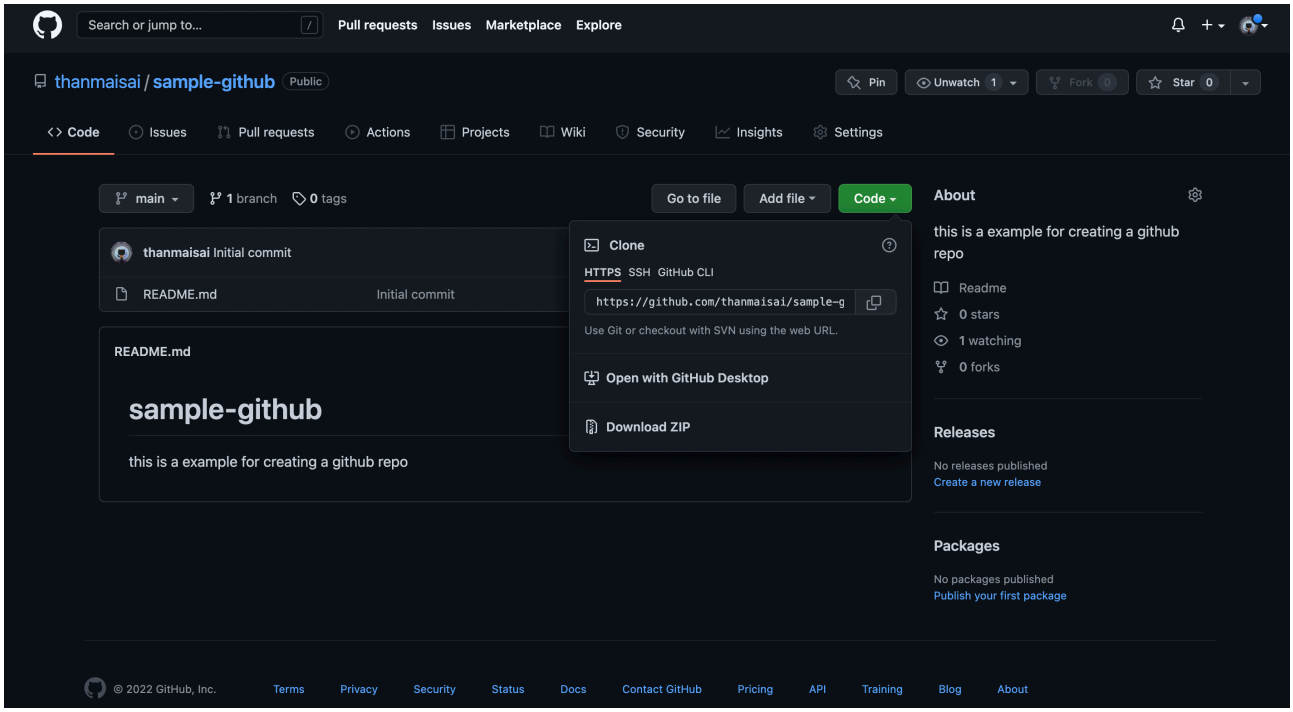




2. Pick a name for your first repository, add a small description, check the Add a README file, and click on the “Create repository” button.

Well we have created your first repository on GitHub.

3. Your first mission is to get a copy of the repository on your computer. To do that, you need to “clone” the repository on your computer. To clone a repository means that you're taking a repository that’s on the server and cloning it to your computer – just like downloading it. On the repository page, you need to get the “HTTPS” address.



4. Once you have the address of the repository, you need to use your terminal. Use the following command on your terminal. When you’re ready you can enter this:
   git clone [HTTPS ADDRESS]

This command will make a local copy of the repository hosted at the given address.

The git clone command on the terminal will give an output similar to the image below.

```
[thanmaisai@THANMAIs-MacBook-Air ~ % git clone https://github.com/thanmaisai/sample-github.git
Cloning into 'sample-github'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
```

5. Now, your repository is on your computer. You need to move in it with the following command.
   cd [NAME OF REPOSITORY]

```
thanmaisai@THANMAIs-MacBook-Air ~ % cd sample-github
thanmaisai@THANMAIs-MacBook-Air sample-github %
```

As you can see in the above picture, my repository name is “My-GitHub-Project” and this command made me go to that specific directory.

6. Now, in the current directory we can create files, work on them, and save them locally. To save them in a remote place “like GitHub” we have do a process called a “commit”.
   There are 4 steps in a commit: ‘status’ , ‘add’ , ‘commit’ and ‘push’.

A. The first thing you need to do is to check the files you have modified. To do this, you can type the following command to make a list of changes appear.
   git status

```
thanmaisai@THANMAIs-MacBook-Air sample-github % git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

B. With the help of the change list, you can add all files you want to upload with the following command.
   git add [FILENAME] [FILENAME] […]

In our case, we’ll add a simple python file(python_sample.py)

```
thanmaisai@THANMAIs-MacBook-Air sample-github % touch python_sample.py
thanmaisai@THANMAIs-MacBook-Air sample-github % ls
README.md               python_sample.py
thanmaisai@THANMAIs-MacBook-Air sample-github % git add python_sample.py
```

C. Now that we have added the files of our choice, we need to write a message to explain what we have done. This message may be useful later if we want to check the change history. Here is an example of what we can put in our case.

```
git commit -m "Added Python_sample file that contain
HELLO WORLD program."
```

```
[thanmaisai@THANMAIs-MacBook-Air sample-github % git commit
[main 4608142] "Added Python_sample file that contain HELLO WORLD program."
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 python_sample.py
```

D. Now we can put our work on GitHub. To do that we have to 'push' our files to Remote. Remote is a duplicate instance of our repository that lives somewhere else on a remote server. To do this, we must know the remote's name (Mostly remote is named origin). To figure out that name, type the following command.
git remote

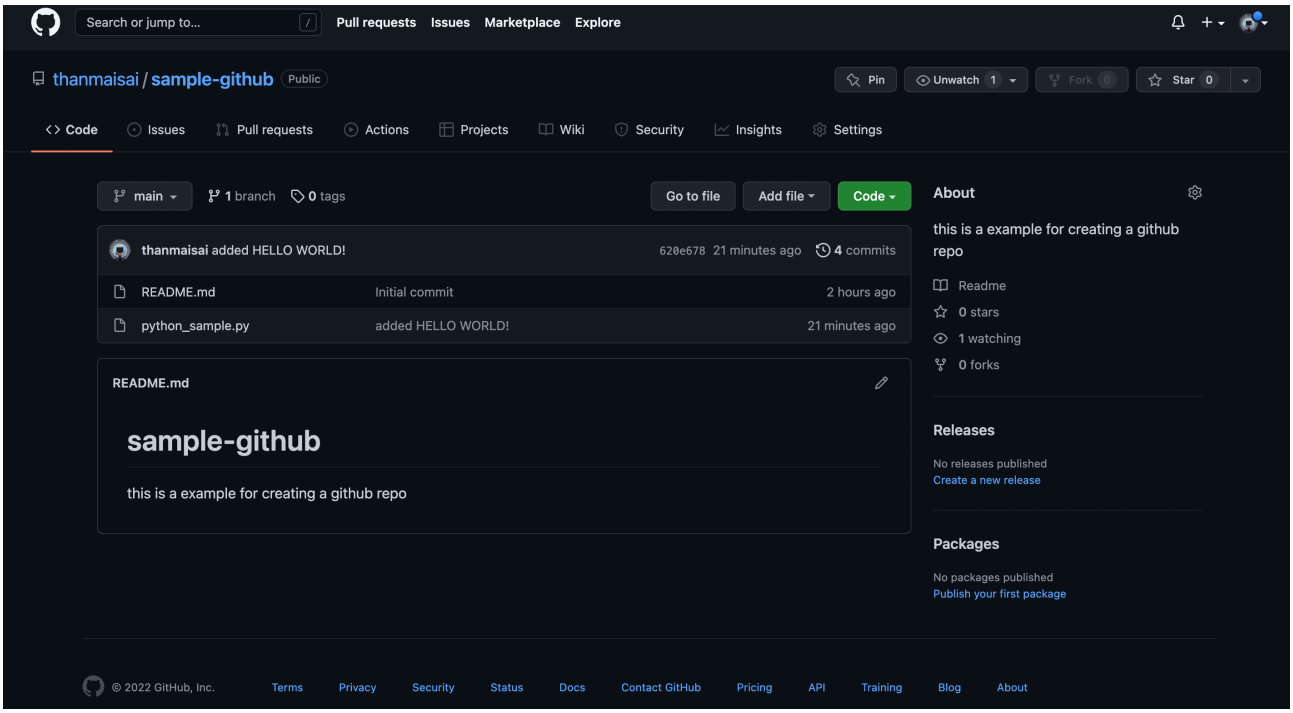
```
[thanmaisai@THANMAIs-MacBook-Air sample-github % git remote
origin
```

E. As you can see in the above image, it says that our remote's name is origin. Now we can safely 'push' our work by the following command.
git push origin master (or) git push -f origin main

Then add your user name and the password of your Github account in the terminal.

NOTE: If you get a error instead of using your GitHub password set an SSH key from shorturl.at/eowzU

F. Now, if we go to our repository on the GitHub web page, we can see the python_sample.py file that we've pushed to remote GitHub.



Work on your project locally then create the repository on GitHub and push it to remote.

In a lot of cases you might have actually already made something on your computer that you want to suddenly turn into a repository on GitHub.
Let's put some files in a git repo which you haven't put in your git previously

1. By default, any directory on our computer is not a Git repository – but we can turn it into a Git repository by executing the following command in the terminal.
   git init

```
[thanmaisai@THANMAIs-MacBook-Air botchat % git init
Reinitialized existing Git repository in /Users/thanmaisai/botchat/.git/
```

2. So there are few files in that directory that we need to "add" to our Repo.
   git add [FILENAME] [FILENAME] […]

NOTE: To "add" all of the files in our Repository we can use the following command:
git add .

3. After converting our directory to a Git repository, the first thing we need to do is to check the files we have by using the following command.
   git status

```
thanmaisai@THANMAIs-MacBook-Air botchat % git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   chat.py
        new file:   intents.json
        new file:   model.py
        new file:   nltk_utils.py
        new file:   train.py

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .DS_Store
        .idea/
        README.md
        __pycache__/
        chat with bot-exp.png
        data.pth
        venv/
```

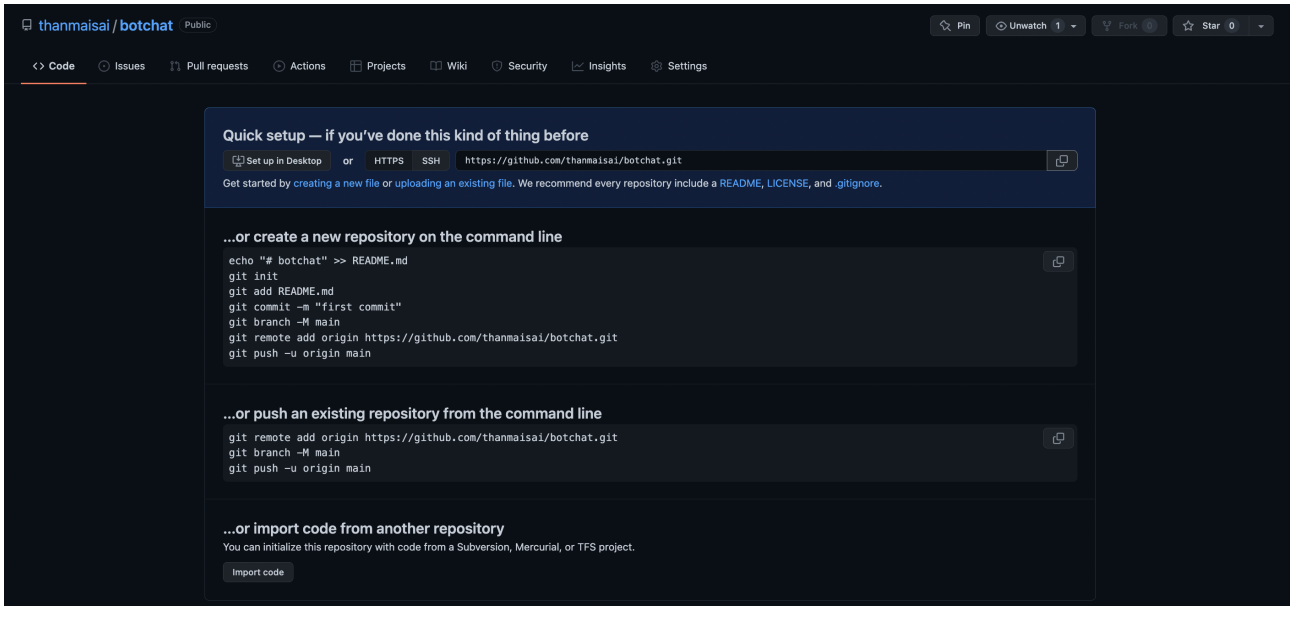
4. After the staging area (the add process) is complete, we can check whether the files are successfully added or not by executing the `git status`

5. Then we have to “commit” with a description in it.
   git commit -m "creating a bot to have a fun conversation"

```
thanmaisai@THANMAIs-MacBook-Air botchat % git commit -m "creating a bot to have a fun conversation"
[master (root-commit) 8d023ec] creating a bot to have a fun conversation
 5 files changed, 340 insertions(+)
 create mode 100644 chat.py
 create mode 100644 intents.json
 create mode 100644 model.py
 create mode 100644 nltk_utils.py
 create mode 100644 train.py
```

6. So to add that remote, we have to go to GitHub first. Create a new repository and name it whatever you want to store it in GitHub. Then click the “Create repository” button.

NOTE: don’t initialize the repository with a README file when creating a new repository on the GitHub.



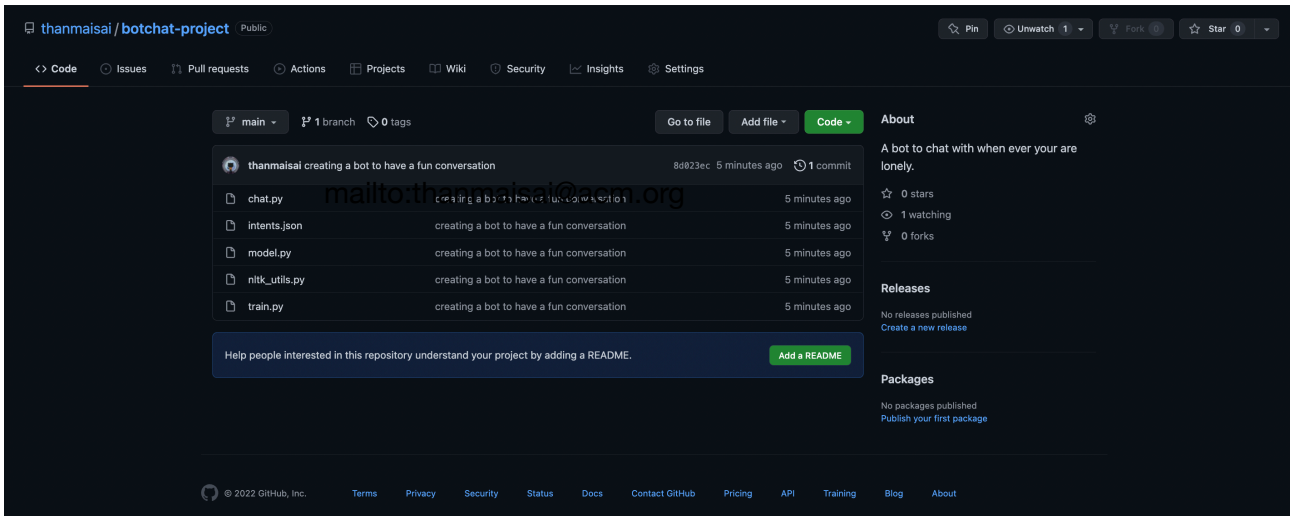
7. Now we wee go to the main branch and the create the remote repo
   git branch -M main

8. Copy the HTTPS address. Now we'll create the remote for our repository.
   git remote add origin [HTTPS ADDRESS]

9. Now lets push to git by using the following command
   git push -u origin main

```
thanmaisai@THANMAIs-MacBook-Air botchat % git push -u origin main
Enter passphrase for key '/Users/thanmaisai/.ssh/id_rsa':
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 3.49 KiB | 3.49 MiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:thanmaisai/botchat-project.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

10. Now go back to your git hub and you can find that your files has been added the repo which you have created earlier.



If any queries your can reach us at thanmaisai@acm.org