

Cloning of Repository from Github

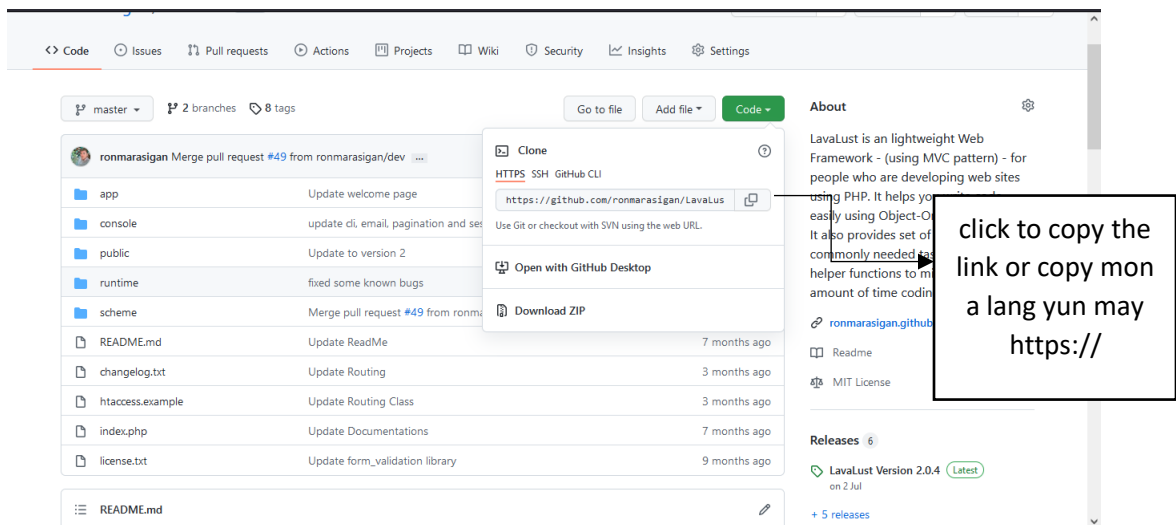
Requirements:

1. Git was already installed.
2. You must have a Github account.
3. Ok na sa dalawang ito. Nai-run na sa terminal.

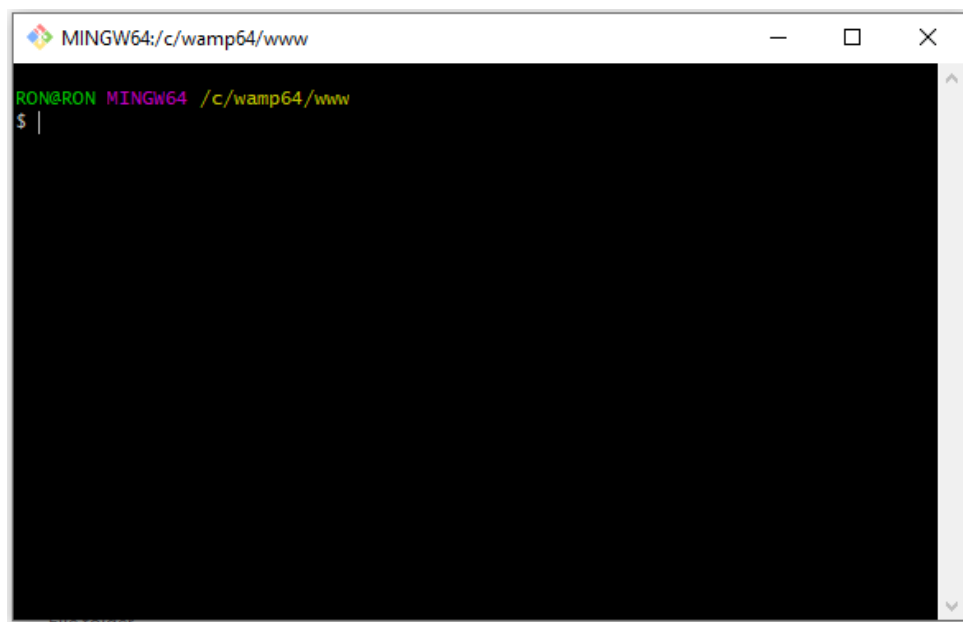
- `git config --global user.email "registered_email_sa_github"`
- `git config --global user.name "username_ninyo_sa_github or kahit name mo nalang"`

1. Open the repository that you want to clone.

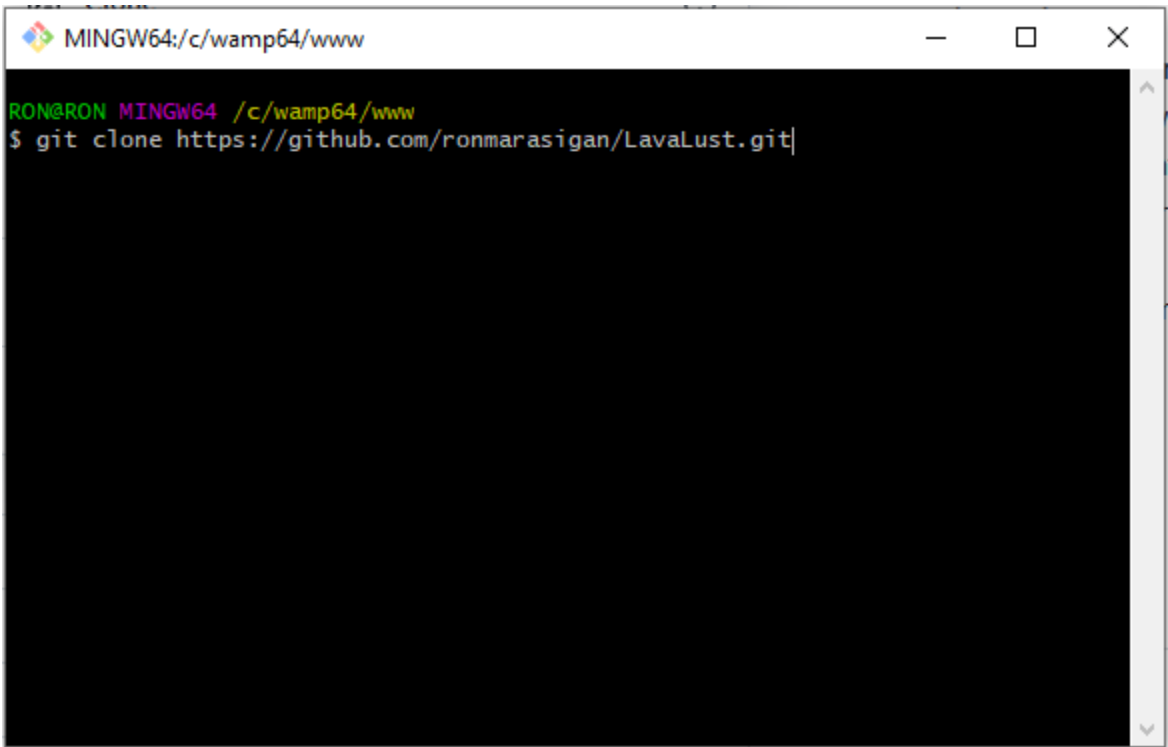
Lavalust Repository Link: <https://github.com/ronmarasigan/LavaLust>



2. Open git terminal inside the server folder. (wampserver -> www or xampp -> htdocs)



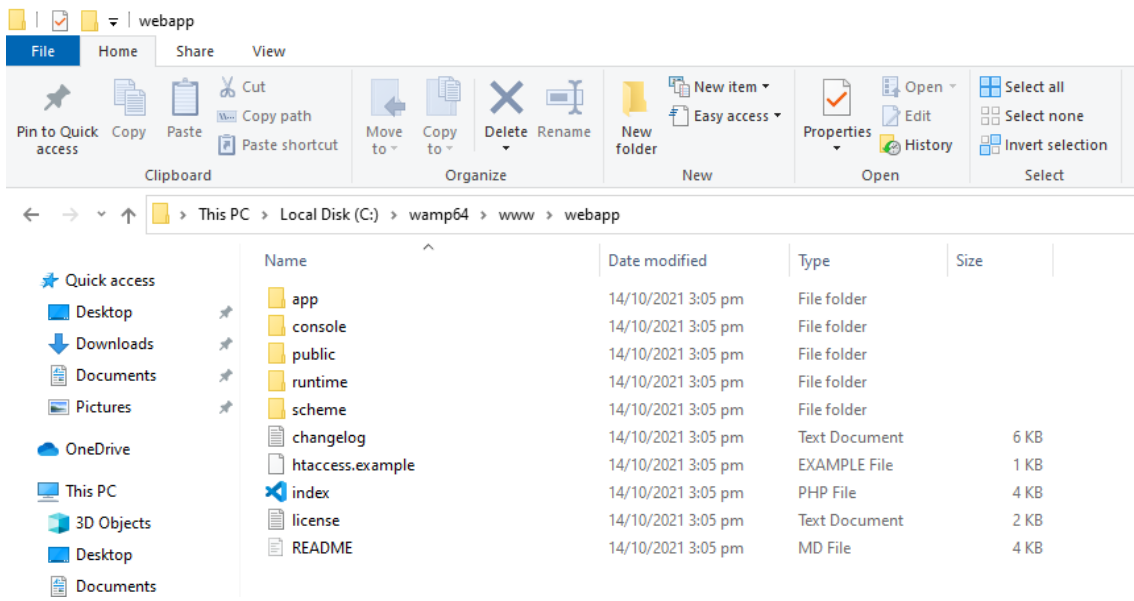
3. Clone the repository. Type `git clone <repository link>`. The one you copy a while ago. Hit enter.



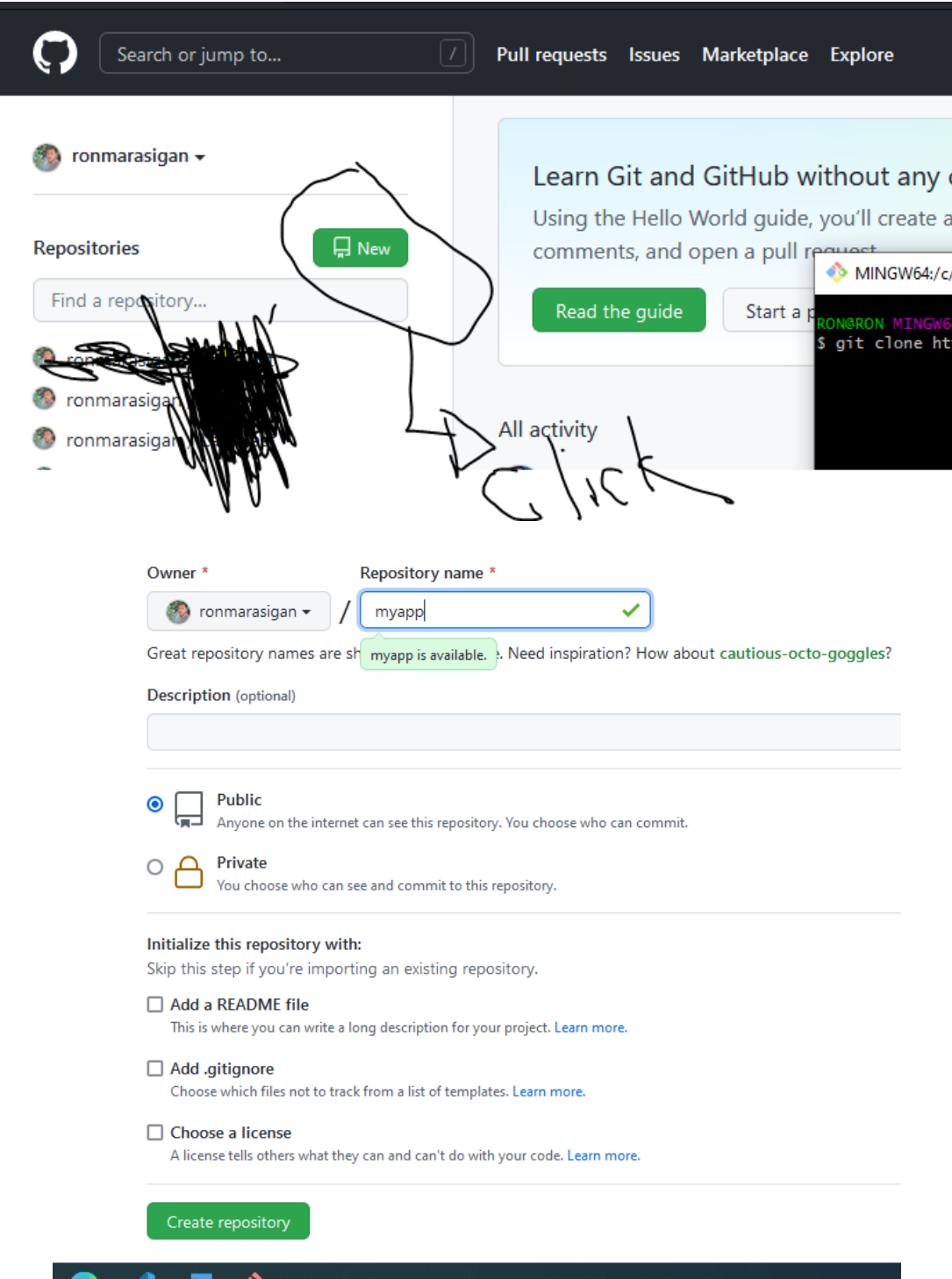
If you want to create a different folder instead of the default one, after `LavaLust.git` add space the type the folder you want the repo to be created.

Example: `git clone https://github.com/ronmarasigan/LavaLust.git webapp` (where `webapp` is the folder)

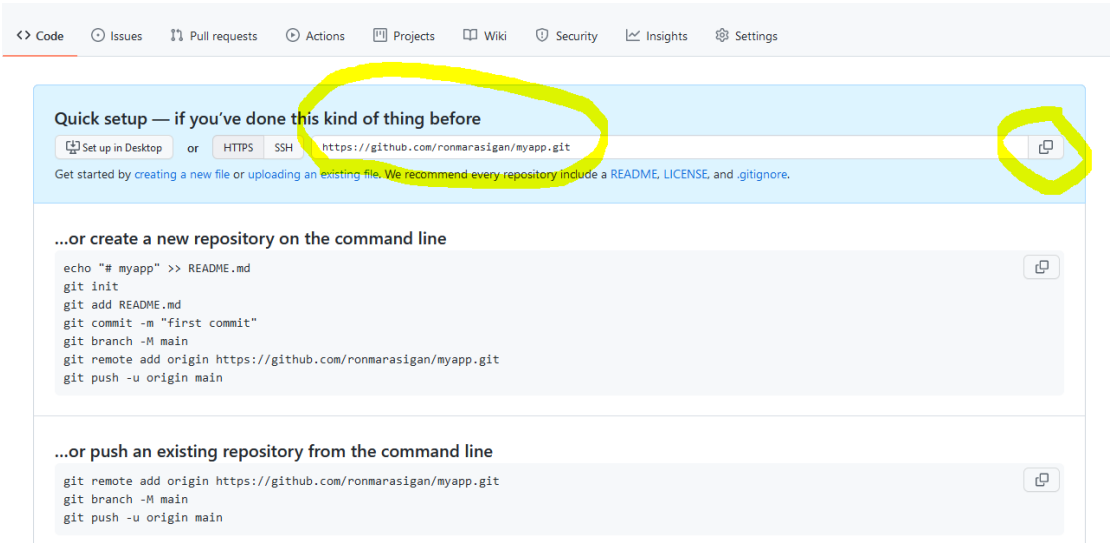
4. Wait for it to be cloned. After cloning, you will see the newly created folder inside the server directory (wampserver -> www or xampp -> htdocs). Open that folder and you will see something like this.



Inside this folder right-click then choose Git Bash again. Now we will change it's remote url. But before that kindly proceed to your github account and create a repository.



Input a repository name then hit Create repository. An empty repo will be created like the one below.



Copy the repository link (like the one with the yellow circle or click on the copy icon. We will be needing this in the next step.

5. Change remote url using the following command. You will need the link copied a while ago.

```
git remote set-url <remote-name> <remote-url>
```

Example:

```
git remote set-url origin https://github.com/ronmarasigan/myapp.git
```

6. To add and commit updates to your local repo use the following commands:

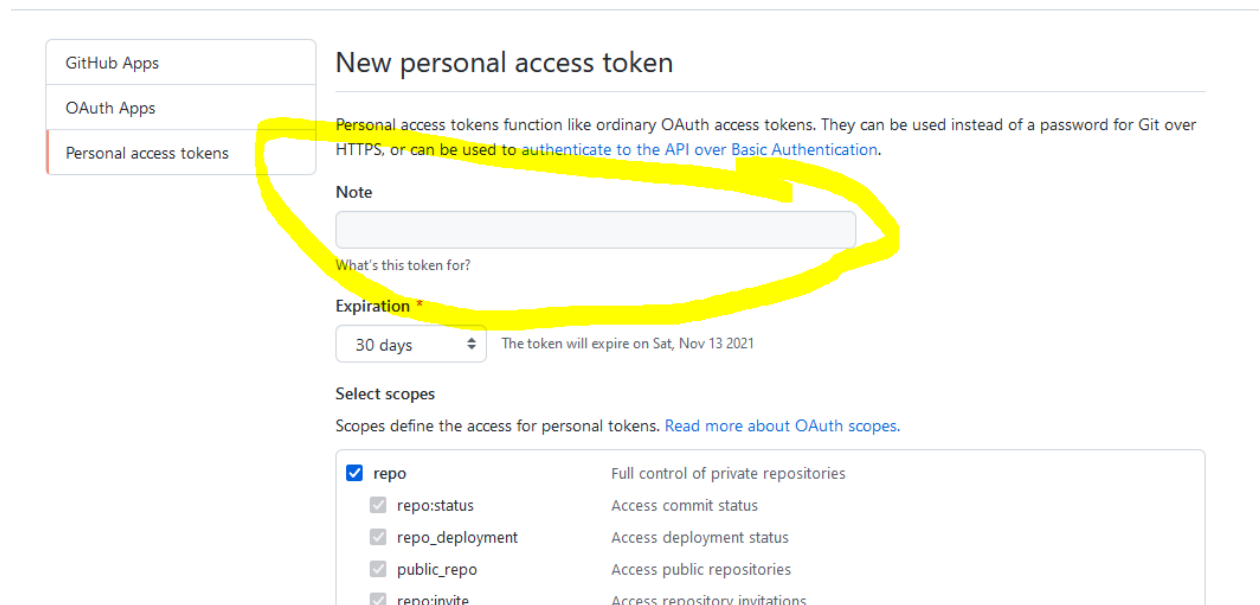
```
git status (to check if there in an unstage update)
```

```
git add . (to add all updates)
```

```
git commit -m "message" (to commit the change)
```

7. Create a personal access token before pushing the update to your newly created repo.

Follow this link to create PAT (Personal Access Token) <https://github.com/settings/tokens> or click Setting -> Developer Settings -> Personal access token. Click Generate new token. Save the token in a safe place. (Yung may yellow, kahit ano ilagay nyo or purpose)



The screenshot shows the GitHub 'New personal access token' page. On the left, there is a sidebar with three options: 'GitHub Apps', 'OAuth Apps', and 'Personal access tokens', with the last one selected. The main content area has a title 'New personal access token' and a description: 'Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.' Below this is a 'Note' field, which is highlighted by a yellow circle. Underneath the note is a text input field labeled 'What's this token for?'. Below that is an 'Expiration' section with a dropdown menu set to '30 days' and a note that the token will expire on 'Sat, Nov 13 2021'. At the bottom, there is a 'Select scopes' section with a list of scopes and their descriptions: 'repo' (Full control of private repositories), 'repo:status' (Access commit status), 'repo_deployment' (Access deployment status), 'public_repo' (Access public repositories), and 'repo:invite' (Access repository invitations). All these scopes are checked.

8. Push the update to your repository

```
git push
```

A window will popup, personal access token will be asked. Input your created PAT. That's it. You will see Your repository with the same content as the one in your local server. You can use step 6 and 8 if there are updates from your local that you want to be pushed to your github repository.