



GITHUB

Pinoy Guide

NA MAY KONTING INDIANO HAHAHA



Notes and Arranged By:



What is Github?

Ang web-based platform na gagamitin natin para sa collaboration, hosting, and version control ay tinatawag na "GitHub". It enables developers to collaborate with one another and host their code repositories in one place. Kalma par we'll walk you through using GitHub and its main features in this document, pero yung basics lang and common hehe.



Creating a GitHub Account:

Una sa lahat, kelangan nyo mag-sign up sa GitHub. Punta lang kayo sa website nila na <https://github.com/> tapos hanapin nyo yung "Sign up" button. Follow nyo lang yung instructions para makagawa ng account at ma-verify yung email nyo.

Creating a Repository:

Pagkatapos nyong mag-create ng account, kelangan nyo naman mag-create ng repository. Repository yung storage space para sa code nyo. Para mag-create ng repository, click nyo yung "New repository" button sa GitHub dashboard nyo. Fill up nyo na lang yung necessary details, tulad ng repository name, description, at kung public or private yung repository nyo.



Adding Files to Your Repository:

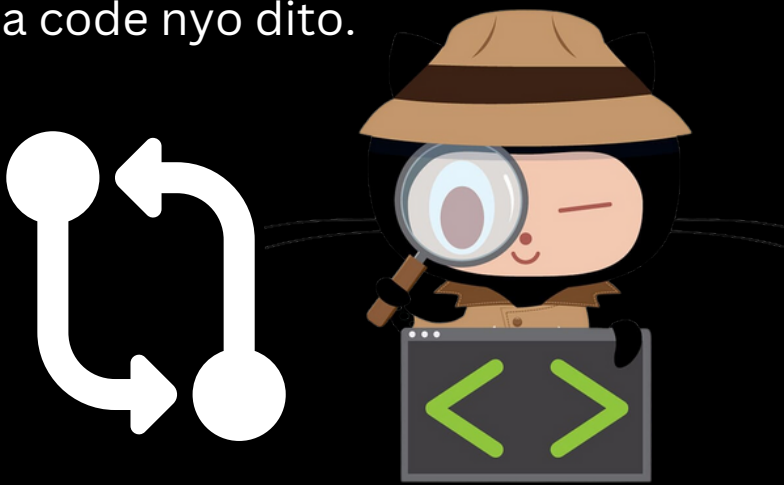
Pagkatapos mag-create ng repository, pwede na kayong mag-add ng files dito. Para mag-add ng files, click nyo yung "Add file" button sa repository page nyo. Pwede kayong mag-upload ng files or mag-create ng bago using the editor ng GitHub. Pwede din naman sa Git Bash, Terminal ng VScode or Tortoise Git. Pero mamaya ko na yan ituturo.

Managing Collaborators:

Pwede din kayong mag-collaborate with other developers sa inyong projects. Para mag-add ng collaborators, pumunta kayo sa repository settings nyo tapos click nyo yung "Collaborators". Pwede nyo na dito hanapin yung gusto nyong i-add na GitHub users.

Creating Branches:

Ang branches ay para mag-work kayo sa different versions ng code nyo simultaneously. Para mag-create ng bagong branch, pumunta kayo sa repository page nyo tapos click nyo yung "Branch" dropdown. Mag-create kayo ng bagong branch at pwede na kayong mag-work sa code nyo dito.



Making Changes and Committing:

Pagkatapos nyong mag-create ng branch, pwede na kayong gumawa ng changes sa code nyo. Pagkatapos nyong gumawa ng changes, kelangan nyo i-commit yun sa branch. Para mag-commit ng changes, pumunta kayo sa "Changes" tab tapos click nyo yung "Commit". Add nyo yung message na nag-de-describe ng changes nyo tapos click nyo yung "Commit changes". Pwede nyo rin naman gawin ito sa Git Bash, VS Code Terminal o kahit sa Git Tortoise

Creating a Pull Request:

Pagkatapos i-commit yung changes nyo sa branch, kelangan nyo naman mag-create ng pull request para ma-merge yung changes nyo sa main branch. Para mag-create ng pull request, pumunta kayo sa "Pull requests" tab tapos click nyo yung "New pull request". Piliin nyo yung branches na gusto nyong i-merge tapos add nyo yung description ng changes nyo. I-review ng mga collaborators nyo yung pull request nyo, at pwede nilang i-approve or reject yung changes.



Git Commands

1. **git clone:** Ito yung command na ginagamit para gumawa ng kopya ng remote repository (nakatago sa GitHub) sa inyong local machine. Ginagawa nitong bagong directory na may pangalan ng repository at kopya lahat ng files at folders mula sa repository papunta sa inyong local machine.
2. **git add:** Ginagamit ito para i-stage yung mga changes na ginawa nyo sa inyong local repository. Ibig sabihin, ito yung unang step para ma-commit yung mga changes nyo.
3. **git commit:** Ito naman yung command na ginagamit para mag-commit ng mga changes na naka-stage na. Dito nyo ida-dagdag yung commit message na naglalarawan ng mga changes na ginawa nyo.
4. **git push:** Kapag naka-commit na kayo ng mga changes, ito yung command na ginagamit para i-upload yung mga changes nyo sa remote repository (sa GitHub).
5. **git pull:** Kapag may bago kayong updates sa remote repository (sa GitHub) na hindi nyo pa nadownload sa local repository nyo, ito yung command na ginagamit para makuha yung latest changes sa remote repository.



6. **git branch**: Ito yung command na ginagamit para mag-create, mag-delete, o mag-check ng mga branches sa inyong repository.
7. **git merge**: Kapag naka-create na kayo ng bagong branch at naka-commit na kayo ng mga changes, ito yung command na ginagamit para i-merge yung changes nyo sa main branch ng repository.
8. **git status**: Ito yung command na ginagamit para i-check yung status ng inyong repository. Makikita nyo dito kung may mga changes na hindi pa naka-stage or naka-commit, o kung may mga updates sa remote repository na hindi pa nyo na-download.
9. **git log**: Kapag gustong malaman yung history ng mga commits sa inyong repository, ito yung command na ginagamit para mag-check ng mga previous commits at commit messages.

<https://bit.ly/YE-GitCommands> 

Nood ka muna sa mga bespren nating Indiano
Para mas malinaw at may guide ka <3



**Ito note oh sundan mo
nalang para mas mabilis**

...or create a new repository on the command line

```
echo "# SidebysideLeaflet" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/dpakprajul/SidebysideLeaflet.git
git push -u origin main
```



Pang sabi sa error “**Please tell me who you are eme eme**”

Run mo tong dalawa magkasunod:

Lagay mo email na gamit mo sa GitHub

```
git config --global user.email "pedro@gmail.com"
```

Lagay mo username na gamit mo sa GitHub

```
git config --global user.name "akosiPedro"
```

Pang sabi sa error “**FATAL REPOSITORY NOT FOUND**”

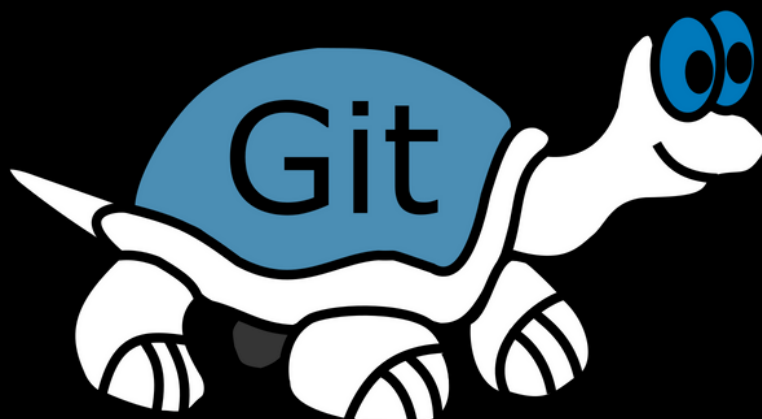
```
1-  
git init  
  
2-  
git add .  
  
3-  
git commit -m "first commit"  
  
4-  
git remote set-url origin https://github.com/Moustafa-desha/Ecommerce_Laravel8.git  
  
5-  
git push -u origin master
```



REMOVING GIT FILES

```
codie@codes-codie:~/Projects/Rails/foodlog$ ls -a
.  app  config  db      Gemfile.lock  .gitattributes  .idea  log      Rakefile  .ruby-version  test  vendor
.. bin  config.ru  Gemfile  .git      .gitignore     lib      public  README.md  storage        tmp
codie@codes-codie:~/Projects/Rails/foodlog$ git rm -rf --cache .idea
rm '.idea/.gitignore'
rm '.idea/foodlog.iml'
rm '.idea/modules.xml'
rm '.idea/vcs.xml'
codie@codes-codie:~/Projects/Rails/foodlog$ git commit -m "Idea removed"
[main f24da48] Idea removed
4 files changed, 513 deletions(-)
delete mode 100644 .idea/.gitignore
delete mode 100644 .idea/foodlog.iml
delete mode 100644 .idea/modules.xml
delete mode 100644 .idea/vcs.xml
codie@codes-codie:~/Projects/Rails/foodlog$ git push
```

Di mo need kabisaduhin lahat ng commands, familiarize kalang hanggang masanay ka. Pero kung nakakaurat yang mga commands, I introduce, "**TORTOISE GIT**" ito yung "kwento mo sa pagong" legit



TORTOISE GIT

Ito simple guide sa pag gamit ng Tortoise Git:

1. Unang-una, i-install mo muna ang TortoiseGit sa computer mo.
2. Pagkatapos nun, gumawa ka ng bagoong folder para sa project mo sa file explorer.
3. Right-click mo yung folder na yun, then piliin mo yung "Git Clone..." para ma-download mo yung repository mula sa remote Git server.
4. Ilagay mo yung URL ng remote Git repository, then click mo yung "OK" para ma-start yung cloning process.
5. Kapag natapos na ang cloning process, puwede ka nang mag-work sa project mo by editing files sa folder.
6. Para magdagdag ng changes, right-click mo yung file, then pumili ka ng "Git Commit -> "main" para ma-commit yung changes sa local repository.
7. Sa commit dialog, maglagay ka ng commit message na naglalarawan ng mga changes mo, then click mo yung "OK" para ma-save yung changes sa local repository.
8. Para ma-push yung changes sa remote Git server, right-click mo yung folder, then piliin mo yung "Git Push..." para i-upload yung mga changes mo sa remote repository.
9. Ipasok mo yung Git credentials mo para ma-authenticate yung push operation, then click mo yung "OK" para matapos ang push process.

<https://bit.ly/YE-TortoiseGitPlaylist> 

**Nood ka muna ulit sa mga bespren nating Indiano
Para mas malinaw at may guide ka <3**



Notice of Disclaimer:

The images presented in this document are not owned by the author and are being used for educational purposes only. The author hereby acknowledges that the author has no ownership or copyright claim over any of the images used in this document. The use of these images is not intended to infringe on any copyrights or plagiarize the work of others. The author takes no responsibility for any legal or ethical issues that may arise from the use of these images. This notice of disclaimer serves as a legal statement that the author does not claim ownership of any of the images used in this document and that the use is solely for educational purposes.

