GITHUB AND CPANEL GIT REPOSITORIES USER GUIDE

Prepared by:

Name : Nur Anis Athirah binti Yusri
Position : Intern, IT
Date :

Reviewed by:

Name : Muhammad Faiz Humaidi bin Mohamed Husain
Position : Junior Executive, IT
Date :

Approved by:

Name : Tengku Juliani Syahida binti Tengku Jaafar
Position : Head of Department, IT
Date :

Table of Content

List of Figure

1.0 Introduction

1.1 Git and GitHub Introduction

Git is a popular version control system. It was created by Linus Torvalds in 2005, and has been maintained by Junio Hamano since then. It enables multiple users to work simultaneously on a single project. It comes with features below:

- ✓ Tracking code changes
- ✓ Tracking who made changes
- ✓ Coding collaboration

Please visit https://www.w3schools.com/git for more details on Git and GitHub.

1.1.1 What does Git do?

- ✓ Manage projects with Repositories
- ✓ Clone a project to work on a local copy
- ✓ Control and track changes with Staging and Committing
- ✓ Branch and Merge to allow for work on different parts and versions of a project
- ✓ Pull the latest version of the project to a local copy
- ✓ Push local updates to the main project

1.1.2 Working with Git

- ✓ Initialize Git on a folder, making it a Repository
- ✓ Git now creates a hidden folder to keep track of changes in that folder
- ✓ When a file is changed, added or deleted, it is considered modified
- ✓ You select the modified files you want to Stage
- ✓ The Staged files are Committed, which prompts Git to store a permanent snapshot of the files
- ✓ Git allows you to see the full history of every commit.
- ✓ You can revert back to any previous commit.
- ✓ Git does not store a separate copy of every file in every commit, but keeps track of changes made in each commit!

1.1.3 Git Platform

- ✓ GitHub - https://github.com/
- ✓ Bitbucket - https://bitbucket.org/
- ✓ GitLab - https://gitlab.com/

1.1.4 Why Git?

- ✓ Over 70% of developers use Git!
- ✓ Developers can work together from anywhere in the world.
- ✓ Developers can see the full history of the project.
- ✓ Developers can revert to earlier versions of a project.

1.1.5 What is GitHub?

- ✓ Git is not the same as GitHub.
- ✓ GitHub makes tools that use Git.
- ✓ GitHub is the largest host of source code in the world, and has been owned by Microsoft since 2018.

1.2 Requirement

Before using GitHub, Git –fast-version-control need to be installed on your device. Git can be installed through https://git-scm.com/downloads. They come with Git GUI and Git Bash terminal.

1.3 GitHub Access

1.3.1 Through links

Sign up for GitHub account (if you are first time user) and login into your GitHub account through the link provided below:

https://github.com

2.0 New Repositories

2.1 Create Repositories

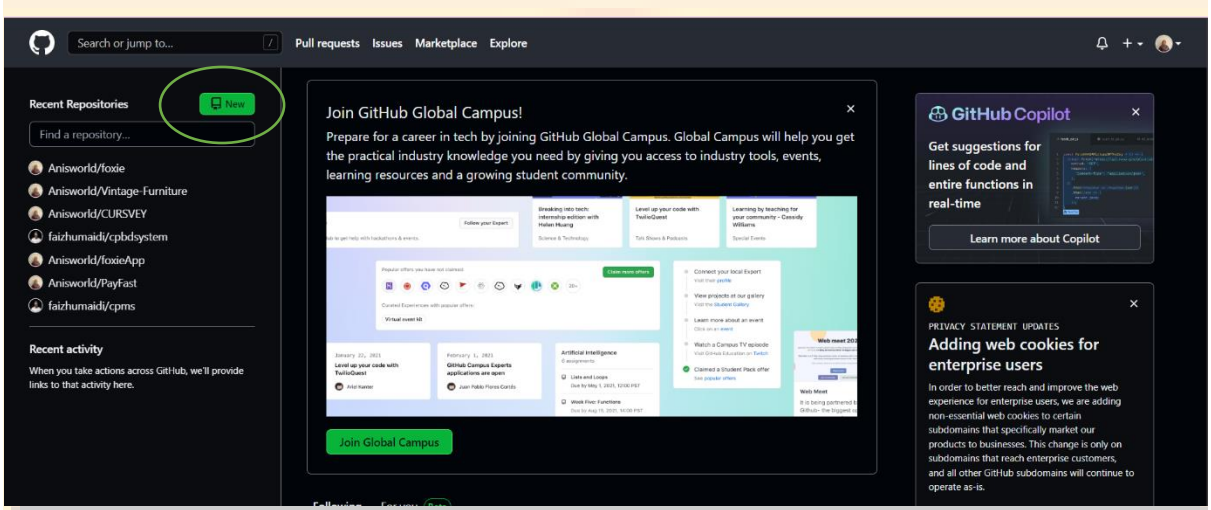

Figure 2.1 GitHub Home Page

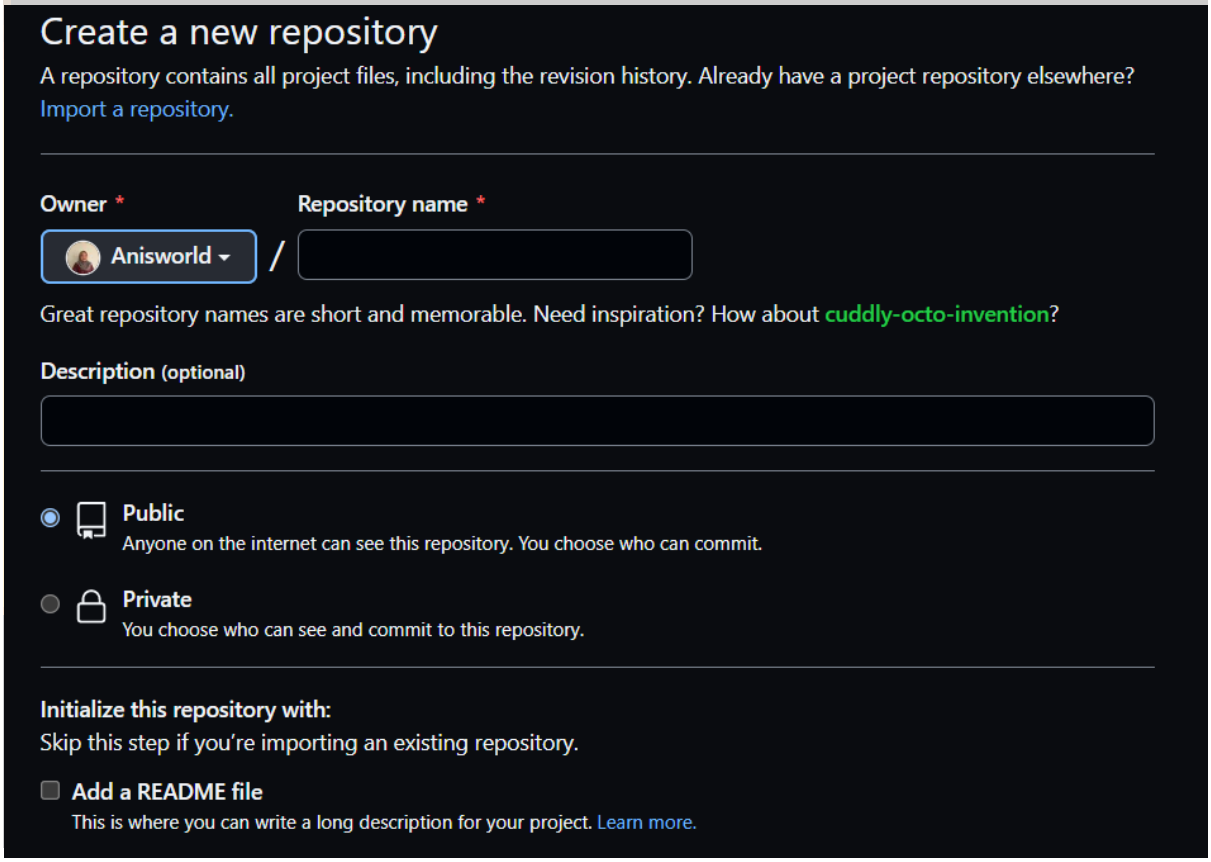

Figure 2.2 Form Create Repository

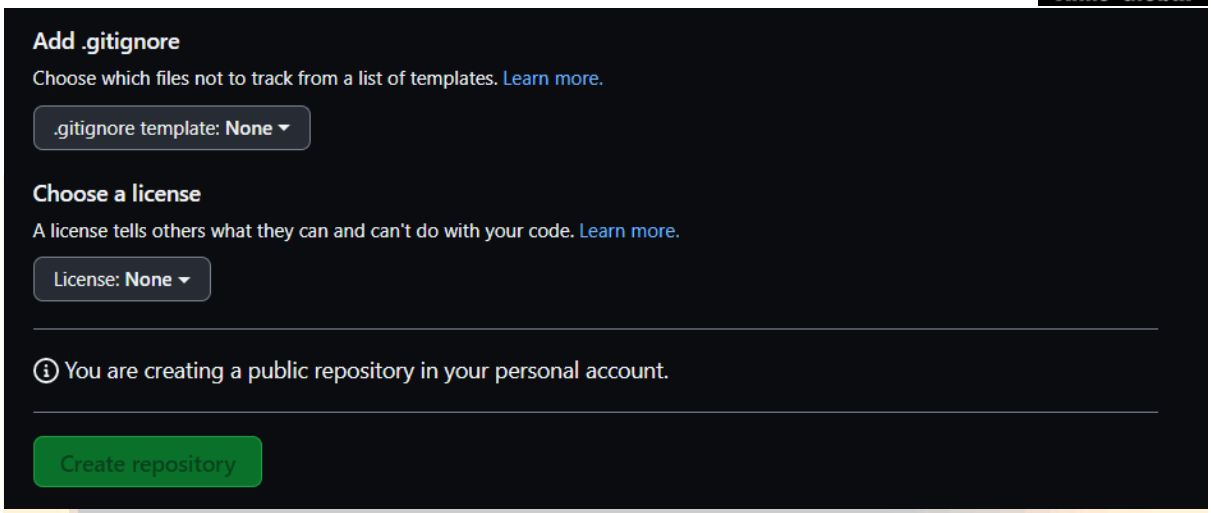

Figure 2.3 Form Create Repository (Cont'd)

GitHub home page will be shown as in Figure 2.1. Click 'New' button on left side of screen as marked in the green circle.

Fill the form as shown in the Figure 2.2 and Figure 2.3. Enter your desired repository name (system shortform is recommended). Choose private to give access only to certain people or choose public to give access to all people on the internet. However, private repositories are recommended since IT department are developing in-house application only. Click Create Repositories after complete.

2.2 Create README.md file

Open file project and add new text document. Rename the file “README” and change the file extension to “.md”. Write a simple description in the file. That description will be displayed in GitHub repositories. Figure below shows the example of README.md file.

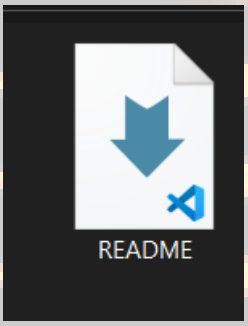

Figure 2.4 README.md file

2.3 Add system/Project into GitHub

Go into project directory in localhost (C:AppServ/www/your_project), right click and click git bash here as shows in the Figure 2.5. Git Bash Terminal will appear as shown in the Figure 2.6.

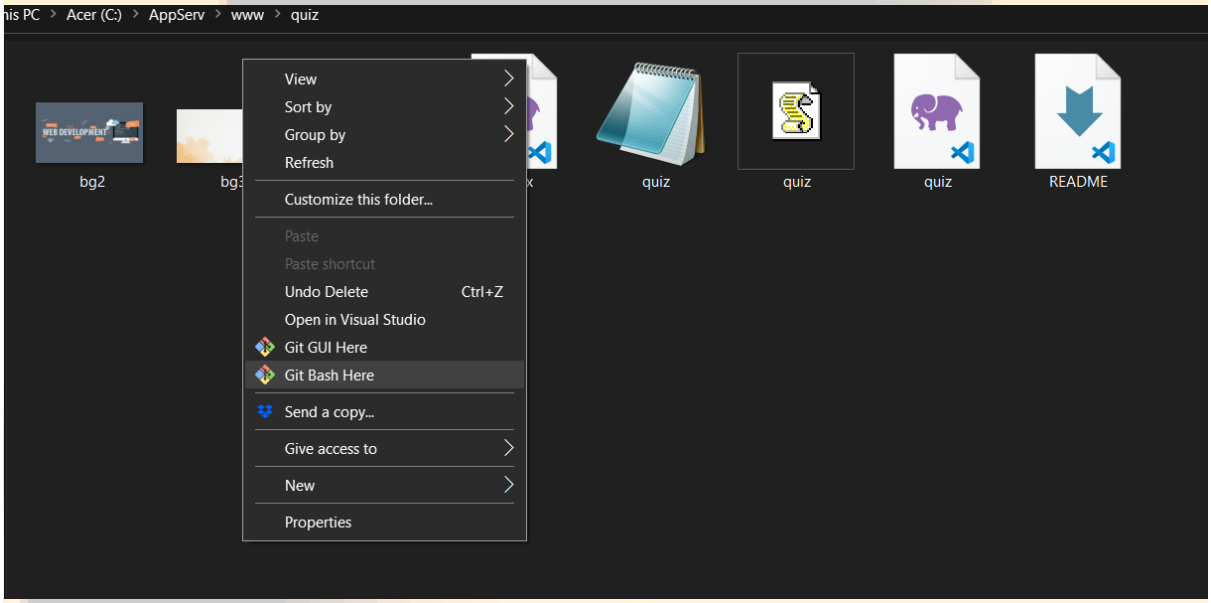

Figure 2.5 Project directory

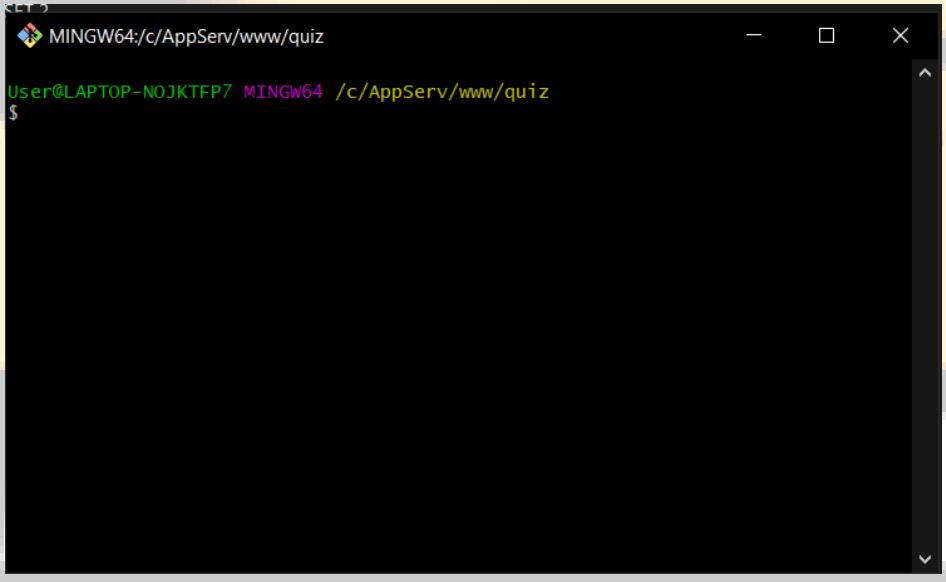

Figure 2.6 Git bash terminal

2.3.1 Initialize Git

Insert command below in the git bash terminal as shown in the Figure 2.7.

```
git init
```

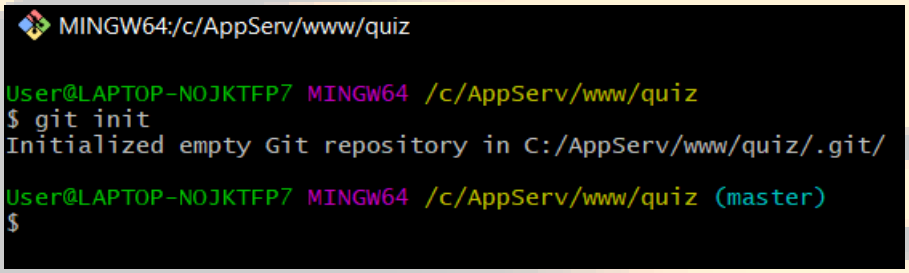

Figure 2.7 Git init

2.3.2 Git Remote

Copy https link from your GitHub Repository as shown in Figure 2.8. Insert command below in git bash terminal as shown in Figure 2.9.

```
git remote add origin https://github.com/your_username/your_repo.git
```

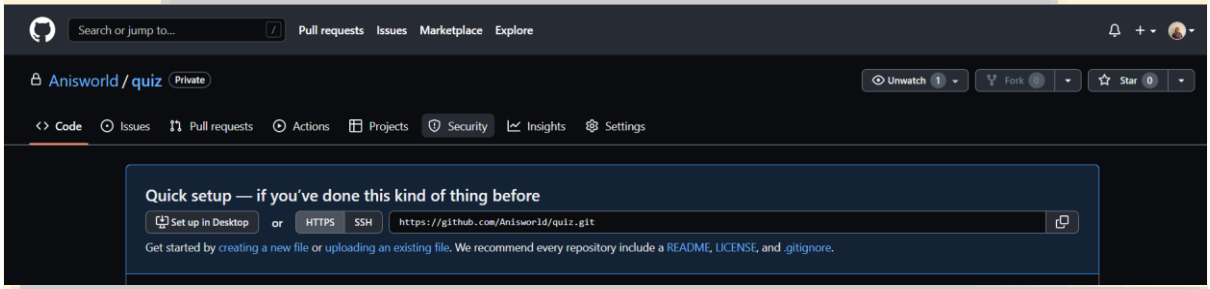

Figure 2.8 GitHub Repository https link

```
User@LAPTOP-NOJKTFP7 MINGW64 /c/AppServ/www/quiz (main)
$ git remote add origin https://github.com/Anisworld/QUIZ.git~

User@LAPTOP-NOJKTFP7 MINGW64 /c/AppServ/www/quiz (main)
$
```

Figure 2.9 Add remote repository

2.3.3 Git Add

There are two ways to add files into staging which are add all edited files and add single files one by one. Refer step below to add files into staging.

- All Files

  Insert command below in Git Bash Terminal to add all files.

```
git add .
```

- Single Files

  Insert command below in Git Bash Terminal to add single files. Let's choose dashboard.php as example.

```
git add dashboard.php
```

```
User@LAPTOP-NOJKTFP7 MINGW64 /c/AppServ/www/quiz (master)
$ git add .

User@LAPTOP-NOJKTFP7 MINGW64 /c/AppServ/www/quiz (master)
$
```

Figure 2.10 Git add

2.3.4 Git Commit

Insert command below to set commit message for your git push command.

```
git commit -m “Your message here”
```

Example output for this command is shown in Figure 2.11.

```
User@LAPTOP-NOJKTFP7 MINGW64 /c/AppServ/www/quiz (master)
$ git commit -m "new project"
[master (root-commit) ca32bd5] new project
 8 files changed, 821 insertions(+)
 create mode 100644 F1213_F1171_PBT1 DFP50193 WEB-SET 2.docx
 create mode 100644 README.md
 create mode 100644 bg2.jpg
 create mode 100644 bg3.jpg
 create mode 100644 index.php
 create mode 100644 quiz.css
 create mode 100644 quiz.js
 create mode 100644 quiz.php

User@LAPTOP-NOJKTFP7 MINGW64 /c/AppServ/www/quiz (master)
$
```

Figure 2.11 Git Commit

2.3.5 Git Push

Enter command below to push all added and committed files into your GitHub repository. Output of this command is shown in Figure 2.12.

```
git push -u origin main
```

```
User@LAPTOP-NOJKTFP7 MINGW64 /c/AppServ/www/quiz (main)
$ git push -u origin main
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 8 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (10/10), 6.25 MiB | 1.24 MiB/s, done.
Total 10 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Anisworld/quiz.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

User@LAPTOP-NOJKTFP7 MINGW64 /c/AppServ/www/quiz (main)
$
```

Figure 2.12 Git Push

Now, refresh your GitHub Repository in browser (e.g., Google Chrome) and make sure all files that added just now are showing. See Figure 2.13.

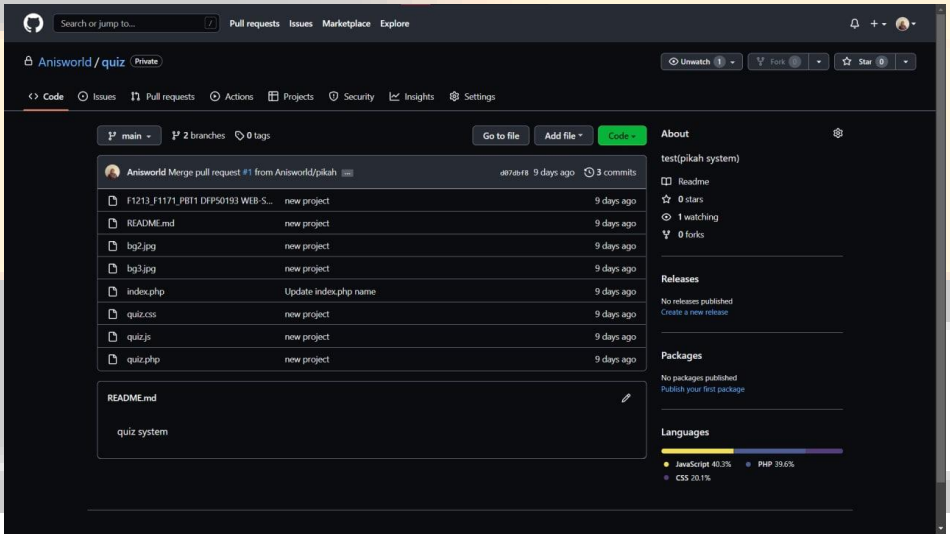

Figure 2.13 GitHub Repository after git push

2.3.6 Git Branch

Insert command below to rename your current working branch to main. 'main' is common branch name for master. You also can create or rename another working branch. Please visit Git Documentation for details.

```
git branch -M main
```

Output of the command is shown in Figure 2.14.

```
User@LAPTOP-NOJKTFP7 MINGW64 /c/AppServ/www/quiz (master)
$ git branch -M main

User@LAPTOP-NOJKTFP7 MINGW64 /c/AppServ/www/quiz (main)
$
```

Figure 2.14 Git Branch

3.0 Collaborator

Private repository can be worked by individual or in a team. If the repository or project are worked in a team, collaborators need to be added into the repository.

3.1 Add Collaborator

Go to setting in the GitHub Repository, choose 'Collaborators' in left side menu, and click 'Add people' button as shown in the Figure 3.1. A popup will appear and collaborators can be added using the popup. Simply type username or email of their GitHub account to add them as collaborators.

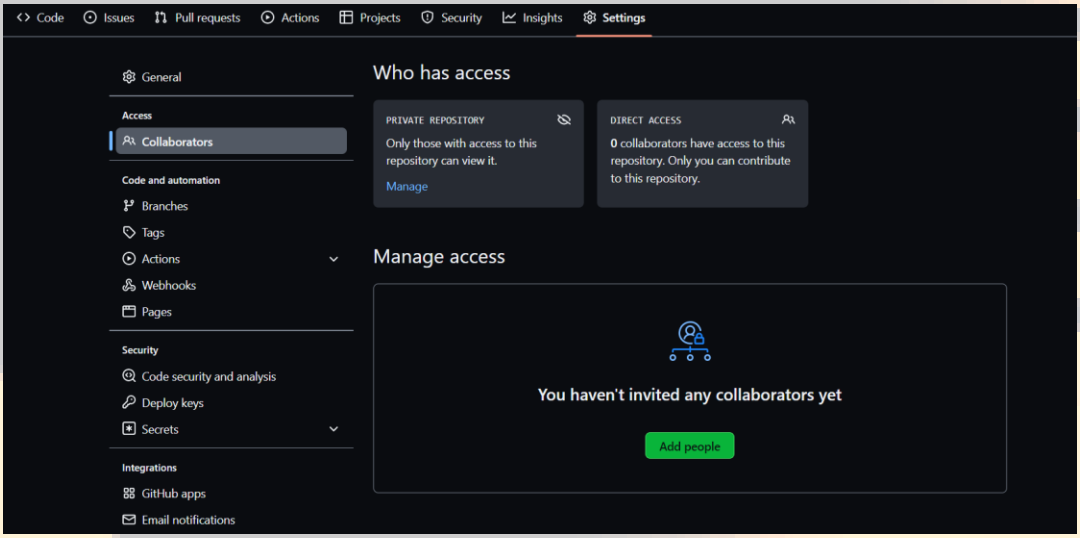

Figure 3.1 Collaborators

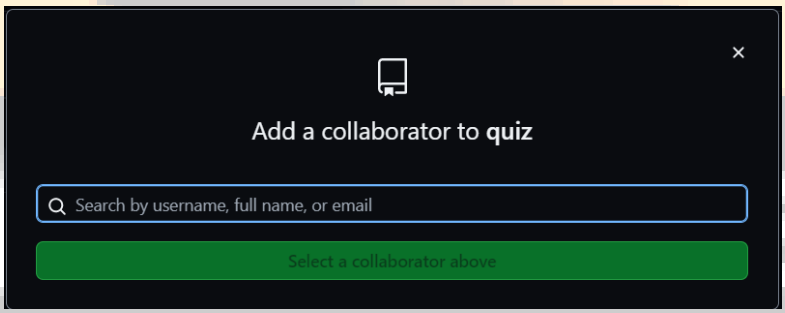

Figure 3.2 Popup Add a Collaborator

After adding collaborators, they need to accept collaborators invitation in 7 days. Please inform your collaborators to accept invitation in https://github.com. Simply click profile picture in top-right screen to view menu, and choose 'Your Repositories' to check for any invitation. Click 'Accept' button and confirm access by entering your GitHub account password.

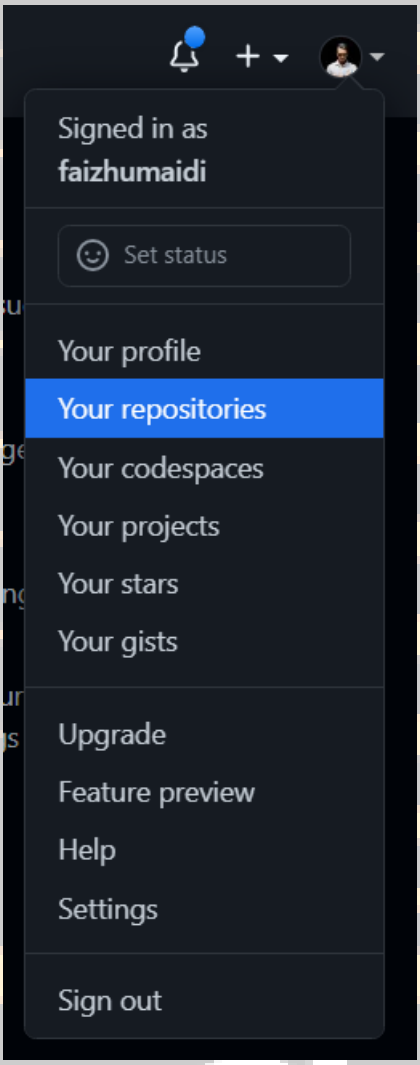

Figure 3.3 Top-right Profile Menu

4.0 Clone Repositories and Pull Updates

Clone is a full copy of a repository, including all logging and versions of files. People can clone any repository on GitHub which means download a copy of the project along with git features. Public repository can be clone by anyone (including for those who do not have GitHub account) using https link. While public repository can only be cloned by people who have access to the repository (GitHub account is required).

Using git clone enables developers to work on project development in their local device. They can pull latest updates or push their modified/added files into repository on GitHub.

4.1 Git Clone

Insert command below to clone repository from GitHub. Remember that collaborator access is required for private repository. Replace <github_repo_https_link> with your desired repository link where you can get it on https://github.com .

```
git clone <github_repo_https_link>
```

```
[user@localhost] $ git clone https://github.com/w3schools-test/w3schools-test.github.io.git
Cloning into 'w3schools-test.github.io'...
remote: Enumerating objects: 33, done.
remote: Counting objects: 100% (33/33), done.
remote: Compressing objects: 100% (15/15), done.
remote: Total 33 (delta 18), reused 33 (delta 18), pack-reused 0
Receiving objects: 100% (33/33), 94.79 KiB | 3.16 MiB/s, done.
Resolving deltas: 100% (18/18), done.
```

Figure 4.1 Example output of git clone

Figure 4.1 shows an example output of git clone command. Then you can open the project using your preferred Integrated Development Environment (IDE) such as

Visual Studio Code and JetBrains PhpStorm. These IDE provide built-in Git GUI for developers to making pull, push, commit, view modified files, and many more.

4.2 Git Pull

Git pull enables developer to get the most recent changes remote repository to local copy (localhost). Pull is a combination of 2 different commands which are fetch and merge. Visit w3schools.com/git/git_pull_from_remote for details.

Insert command below to get most recent changes from GitHub. Example output is shown in Figure 4.2.

```
git pull origin
```

```
[user@localhost] $ git pull origin
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 794 bytes | 1024 bytes/s, done.
From https://github.com/w3schools-test/hello-world
   a7cdd4b..ab6b4ed  master       -> origin/master
Updating a7cdd4b..ab6b4ed
Fast-forward
 README.md | 2 ++
 1 file changed, 2 insertions(+)
```

Figure 4.2 Example output of git pull

Instead of using command line, you also can use IDE with built-in Git GUI such as Visual Studio Code and JetBrains PhpStorm. Visit your preferred IDE official site for documentation.

5.0 Branch

In Git, a branch is a new/separate version of the main repository. Branches allow you to work on different parts of a project without impacting the main branch. When the work is complete, a branch can be merged with the main project. You can even switch between branches and work on different projects without them interfering with each other.

If you are working on updates of existing project, you are recommended to updates on separated branch first before merging with main branch so that we can avoid impacting the project that has been hosted.

Git Branch is recommended especially you are working on a project with teams. In other words, you can create branch for each developer in your team to avoid interference between your part and your teammate's part. You can merge the branch with the main in the end of development.

5.1 Create Branch in GitHub

Go to https://github.com and click button new branch as shown in the Figure 5.1

Fill branch name and click 'Create Branch' as shown in Figure 5.2.

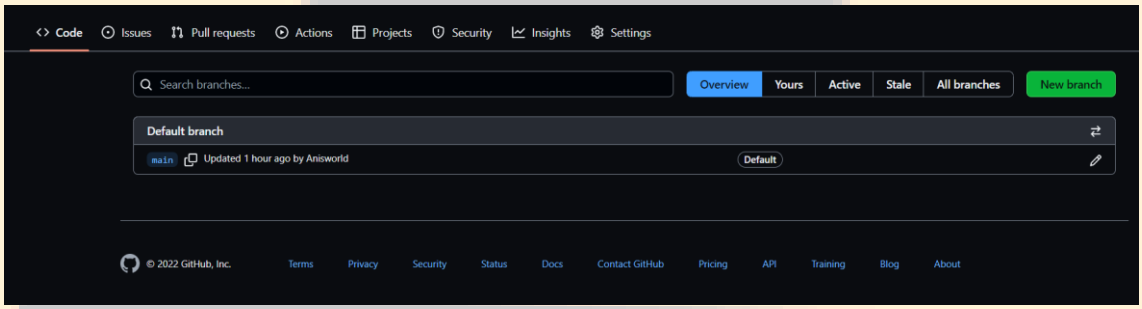

Figure 5.1 Create Branch in GitHub

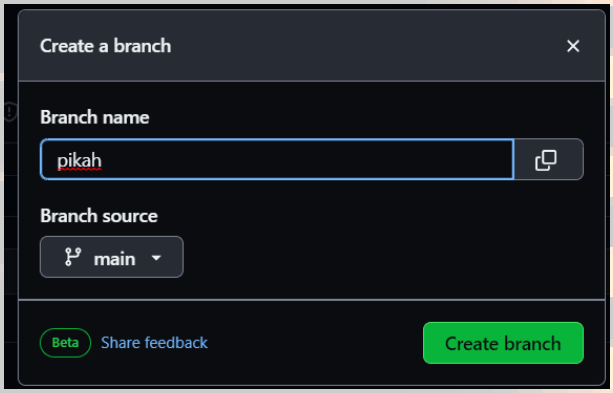

Figure 5.2 Create branch popup form

View branch list on GitHub or using terminal by inserting command below:

```
git branch
```

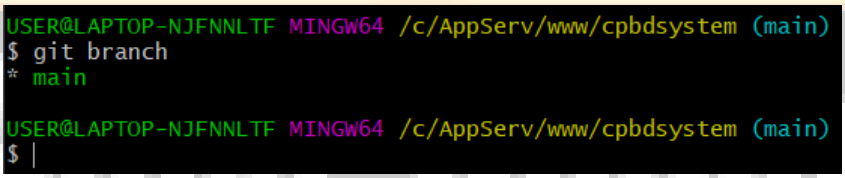

Figure 5.3 Example output of git branch

5.2 Switch Branch

Git checkout command enables developer to switch between working branch. There are multiple ways to use git checkout especially using IDE with built-in Git GUI.

5.2.1 JetBrains PhpStorm

There two options to git checkout in the PhpStorm. First, by using git checkout command and second is by clicking branch option on the bottom-right screen.

- Terminal

  Insert command below and you will change branch immediately. See Figure 5.4 for example.

```
git checkout your_branch_name
```

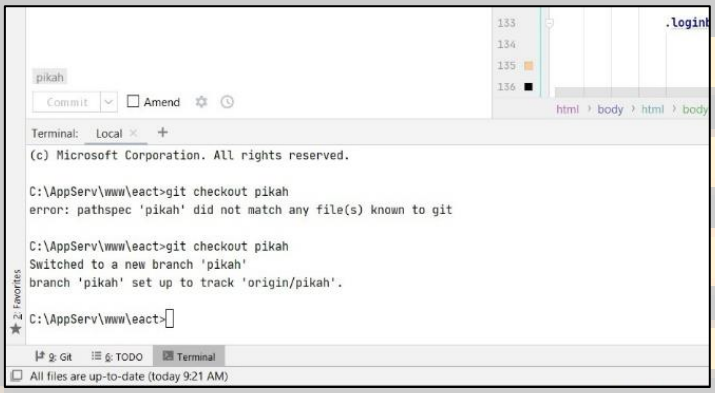

Figure 5.4 Switch Branch using terminal

- Branch Option Interface

  Click branch option on bottom-right screen. Options for branch will appear and choose any branch under 'Local Branches' label that you want to switch. See Figure 5.5 and Figure 5.6.

Figure 5.5 Branch option button in PhpStorm

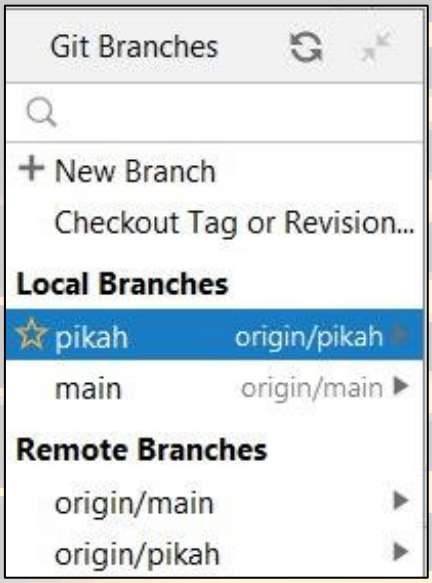

Figure 5.6 Branch options in PhpStorm

5.2.2 GitHub Desktop

Open GitHub Desktop and click at the current branch and will appear as shows in figure below, then can click branch want to use as shows in the Figure 5.7.

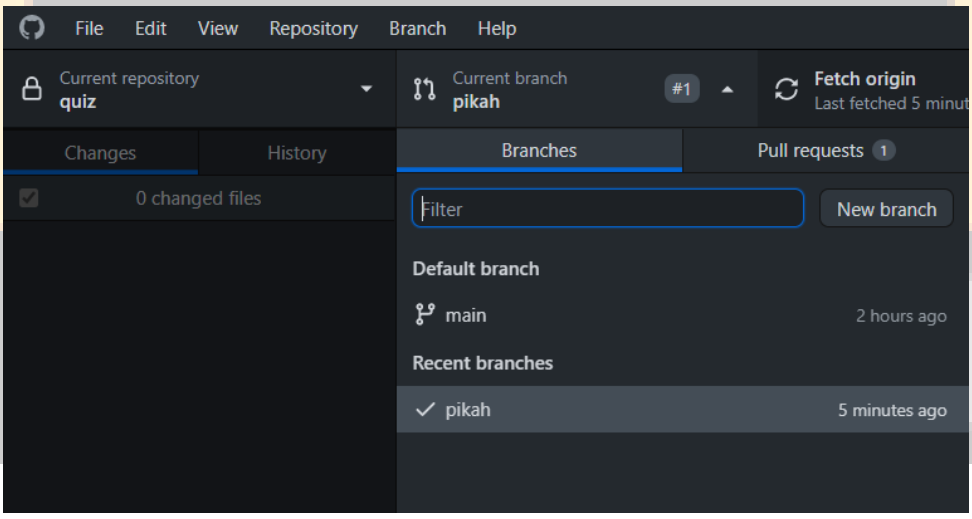

Figure 5.7 Git Checkout in GitHub Desktop

5.3 Git Merge

5.3.1 Git Merge for PhpStorm

Git merge also can be done in different ways whether using terminal or using any Git GUI.

- Terminal

Let's say you are currently in main branch. Insert command below to merge defined branch to your current branch.

```
git merge pikah
```

All changes from branch 'pikah' will be merge into branch main after inserting the command.

- Built-in GUI

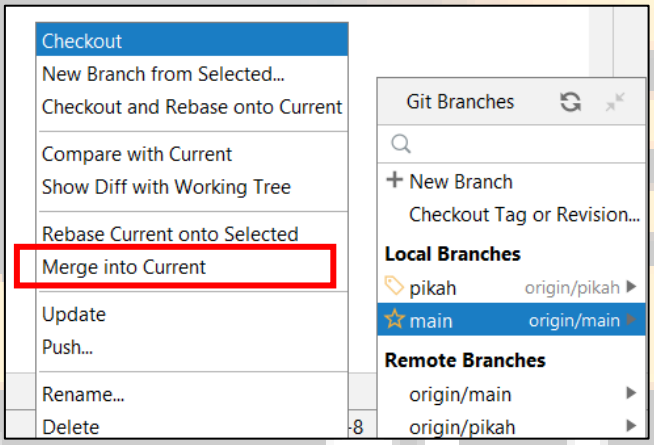

Figure 5.8 Git Merge for PhpStorm

Click branch options, and find 'Merge into Current' options from desired branch to merge.

5.3.2 Git Merge for GitHub Desktop

Click ‘Current Branch’ tab and click ‘Choose a branch to merge into currentbranch_name’ as shown in Figure 5.9. After that, click your desired branch to merge into the current branch, then click ‘Merge Branch’ into Branch as shown in the Figure 5.10. Lastly, click ‘Push Origin’ to push your local changes to the remote repository as shown in Figure 5.11.

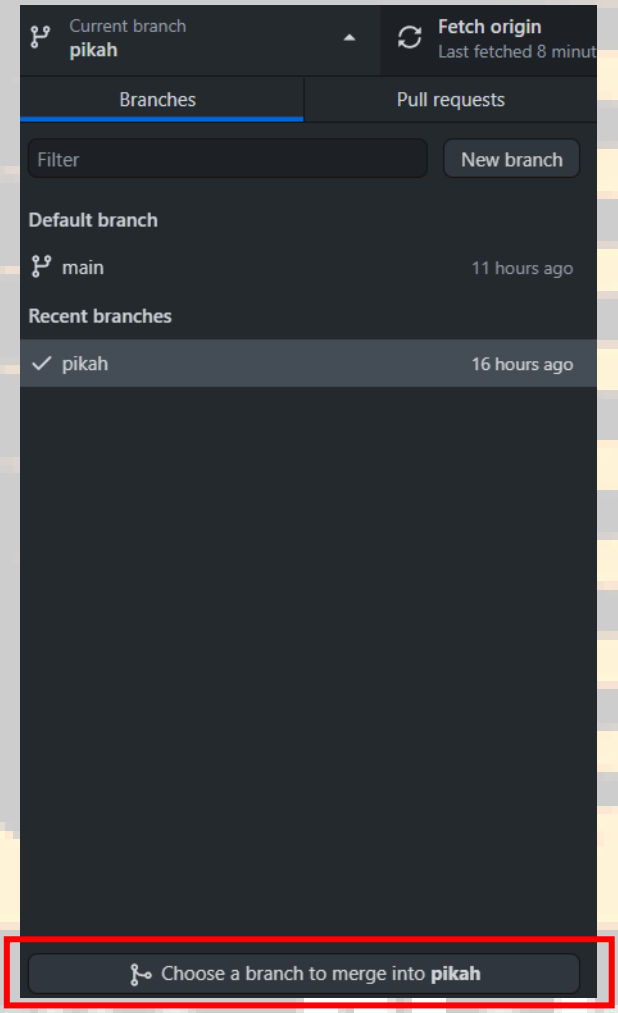

Figure 5.9 Git Merge for GitHub Desktop

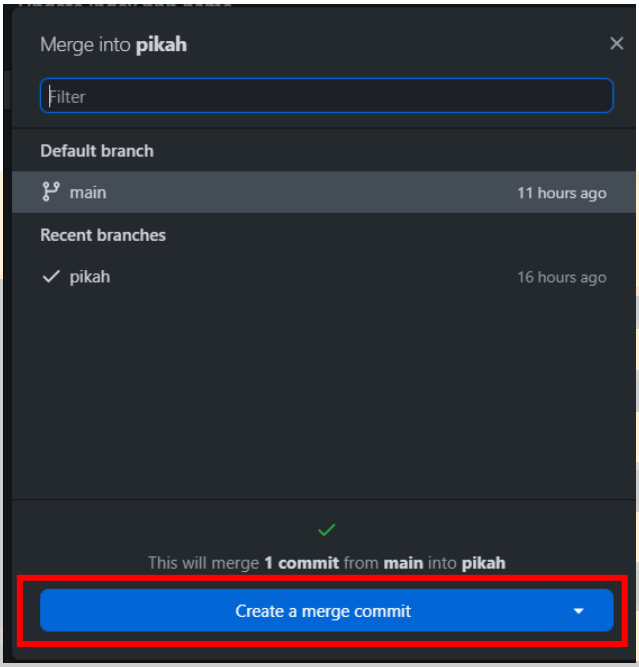

Figure 5.10 Popup Merge in GitHub Desktop

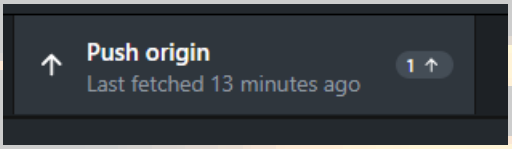

Figure 5.11 Push Origin after Git Merge

5.4 Delete branch

Delete branch also can be done in terminal or git GUI of any IDE.

5.4.1 Delete Branch for PhpStorm

- Terminal

  Insert command below to delete branch where branch_name is your branch name that will be deleted.

```
git branch -d branch_name
```

- Built-in GUI

  Use branch option as shown in Figure 5.12 to delete branch. However, please avoid delete main branch as it is the main source for the project.

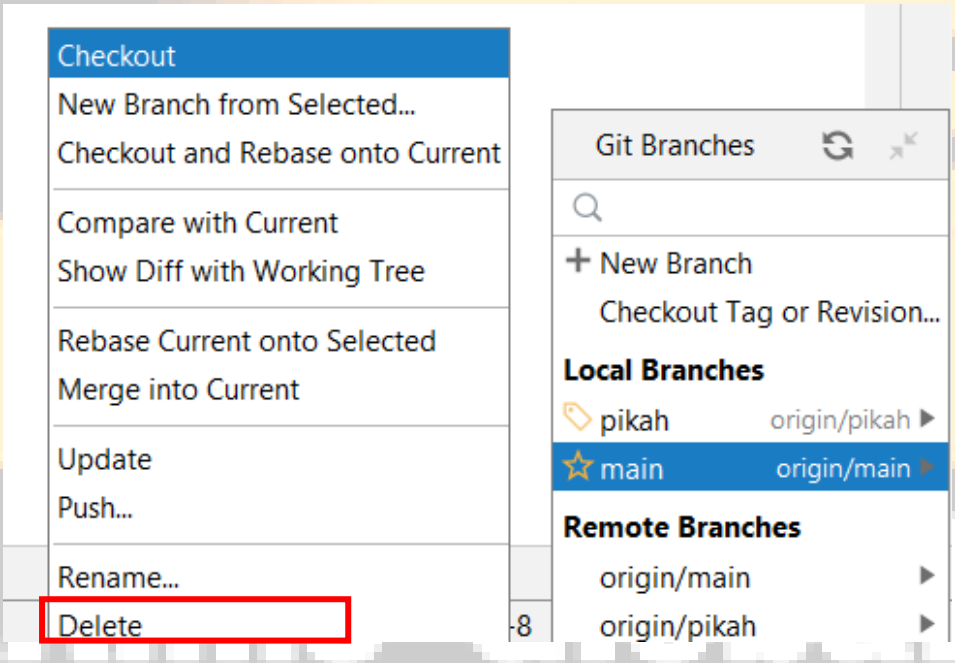

Figure 5.12 Delete Branch for PhpStorm

5.4.2 Delete Branch for GitHub Desktop

Switch to branch that will be deleted. Go to 'Branch' at the top screen, and click delete in the dropdown option as shown in the Figure 5.13.

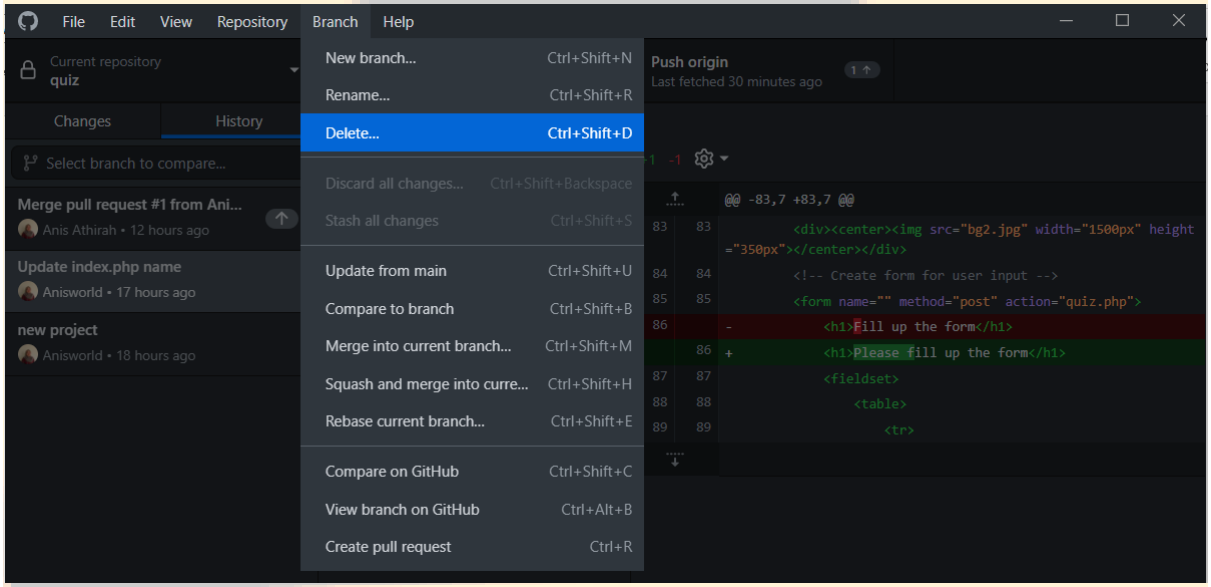

Figure 5.13 Delete Branch for GitHub Desktop

6.0 Connecting cPanel with GitHub

6.1 cPanel Git™ Version Control Introduction

The Git™ Version Control feature allows you to easily host Git repositories on your cPanel account. You can use Git to maintain any set of files (web-app files).

SSH key are required to access private repositories from GitHub. However, cPanel Git Version Control system only offering one SSH key to access GitHub private repository. In other words, only one GitHub account can be access by one cPanel account.

In this case, we recommend using the GitHub account of the Programming Team Leader to connect and gain access from Git Version Control System to GitHub repository. Team Leader GitHub account must have access to all projects (private repositories) which are from another GitHub account. If and only if the project will be deployed in the cPanel Git Version Control System. See Figure 6.1.

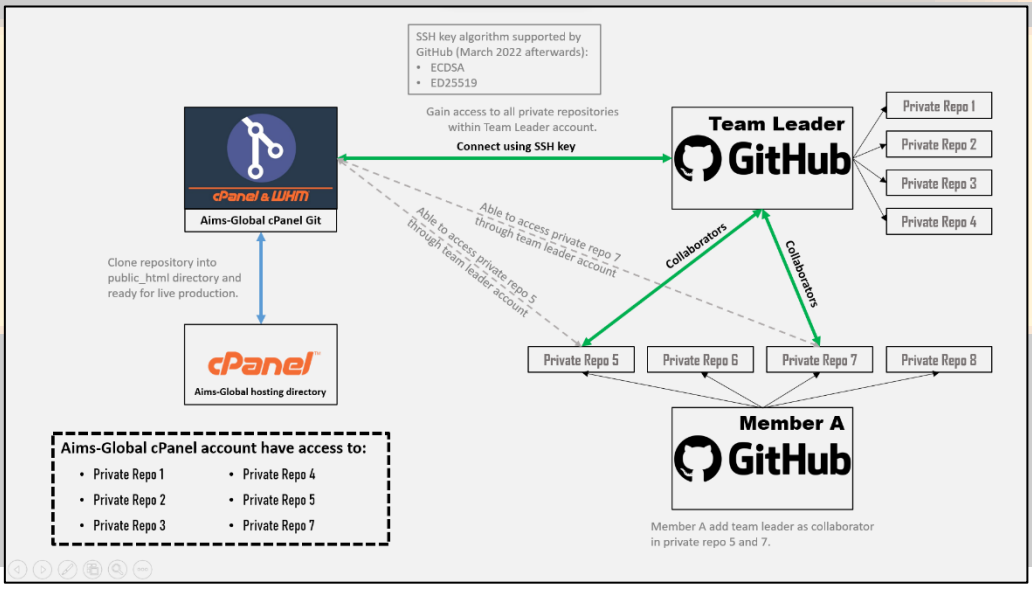

Figure 6.1 cPanel Git and GitHub connection flow

6.2 cPanel Access

Only IT department staffs are authorized to access cPanel Server. Please ask for assistance from authorities (IT Staffs).

6.3 Generate SSH key

6.3.1 Generate key and add to SSH agent

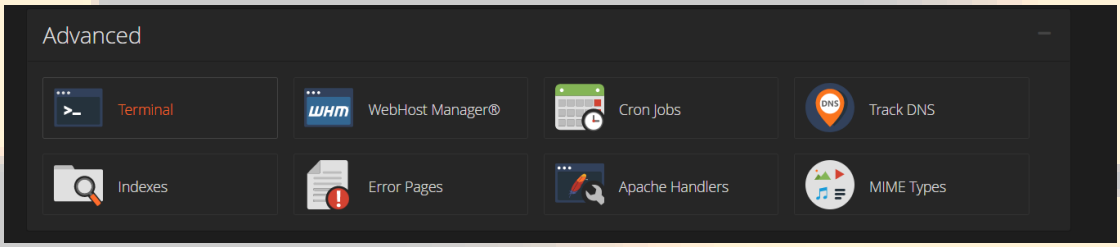

Figure 6.2 cPanel Terminal

Please skip this step if current cPanel Git is linked with one GitHub account. All system access will be terminated if SSH key was overwritten.

Go to terminal in cPanel as shown in figure above and insert command below.

```
ssh-keygen -t ed25519 -C "your_email@example.com"
```

Then it will ask you to enter location to save the SSH key file. Click enter to save in the suggested location. Make sure you are not overwriting SSH key files. Next, it will ask for a passphrase for the generated key. Do not enter anything for the passphrases and click enter. Click enter again for confirmation.

```
> Enter passphrase (empty for no passphrase): [Type a passphrase]
> Enter same passphrase again: [Type passphrase again]
```

The generated key needs to be added into ssh agent. Simply enter command below to add the key into ssh agent.

```
eval "$(ssh-agent -s)"
```

Command above will start SSH agent in background. Example output for command above is shown in Figure 6.3

```
# start the ssh-agent in the background
$ eval "$(ssh-agent -s)"
> Agent pid 59566
```

Figure 6.3 Example output of starting SSH agent command

Add your generated key into SSH agent by inserting command below. id_ed25519 is default SSH key file name. It is common file name for SSH key. However, please check your SSH key file in directory (/home/aimsglob/.ssh) for confirmation.

```
ssh-add ~/.ssh/id_ed25519
```

The output will be “Identity has been added: /home/aimsglob/.ssh/id_ed25519”. Generating SSH key and adding into SSH agent has been completed.

6.3.2 Authorize SSH key

Open SSH access under category Security in cPanel interface as shown in Figure 6.4. Find your generated key (id_ed25519 – usually with status 'not authorized) and click manage. Click authorized, go back to SSH Access list (Figure 6.5) and you will see that your key is now authorized as shown in Figure 6.7. [1]

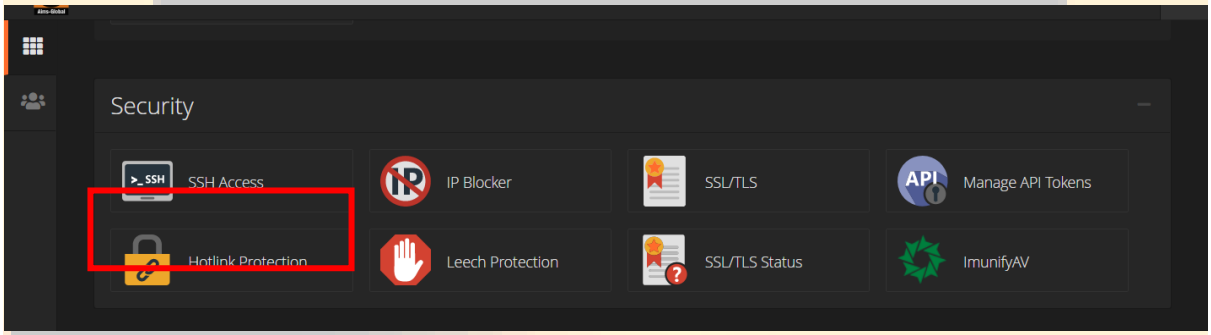

Figure 6.4 SSH Access Icon

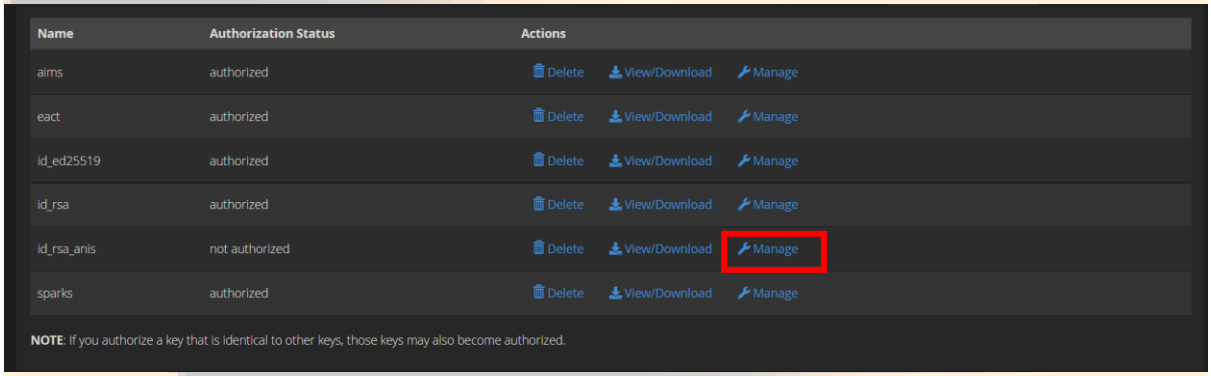

Figure 6.5 SSH Access (Manage)

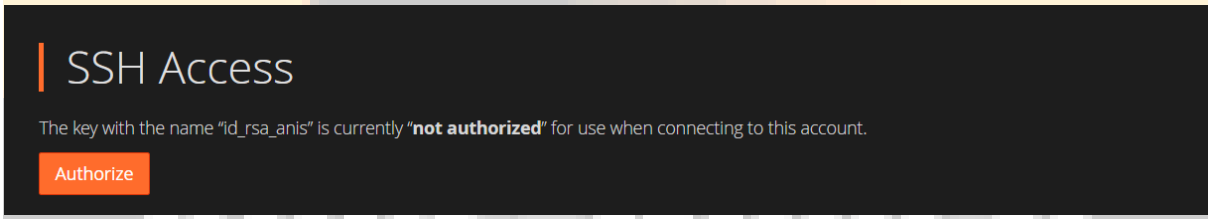

Figure 6.6 Authorize Key

[1] Figure 6.5, Figure 6.6, Figure 6.7, and Figure 6.8 are shown for reference only. id_rsa_anis key is an rsa key type and not accepted by GitHub for access.

6.3.3 Register SSH key with GitHub account

In order to gain GitHub account access for cPanel Git, SSH key need to be register in GitHub account (in our case is Team Leader account). Go to SSH Access list and find your generated key. Click 'View/Download' and copy all the content displayed. See Figure 6.7 and Figure 6.8 for reference.[2]

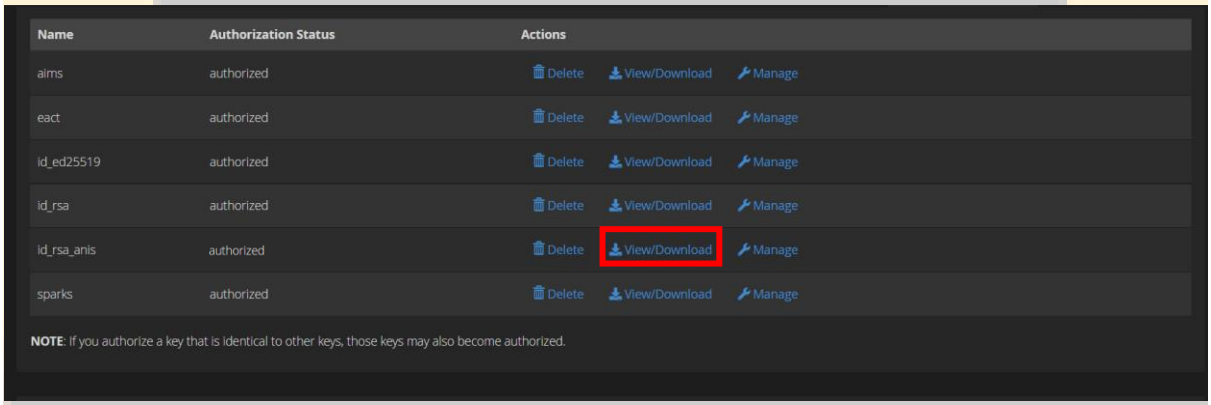

Figure 6.7 SSH Access (View/Download)

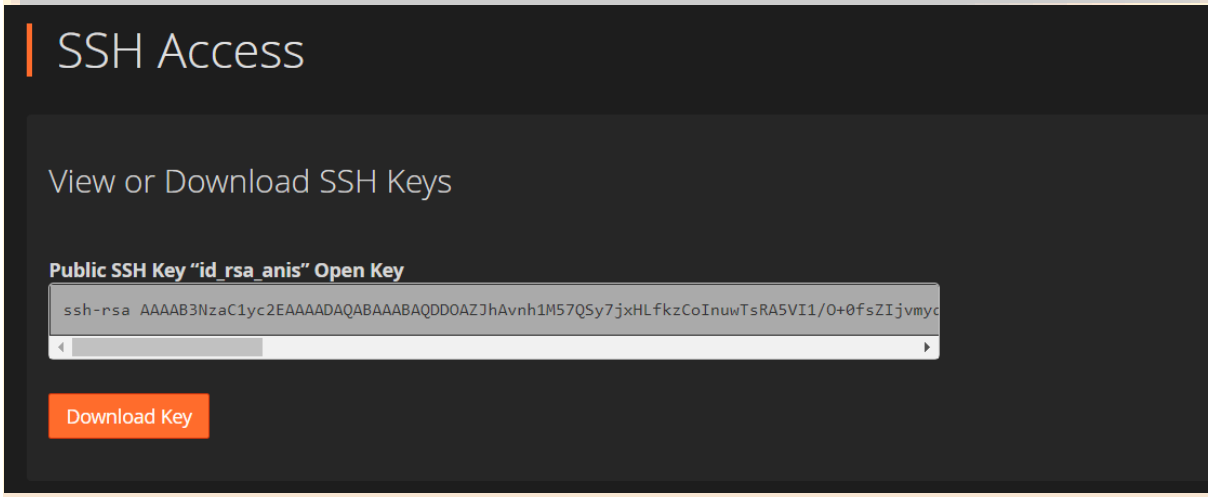

Figure 6.8 SSH View/Download

[2] Figure 6.7, Figure 6.8, Figure 6.9, and Figure 6.10 are shown for reference only. id_rsa_anis key is an rsa key type and it is not accepted by GitHub for access.

Go to github.com and click ‘Setting’ in dropdown from top right screen as shown in the Figure 6.9. Go to ‘SSH and GPG keys’ in left menu sidebar, click ‘New SSH key’ button as shown in the Figure 6.10.

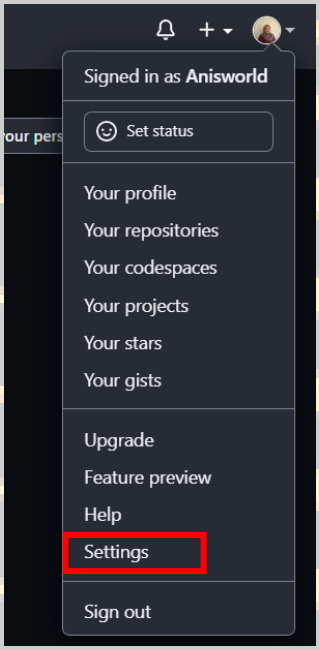

Figure 6.9 GitHub Setting Account

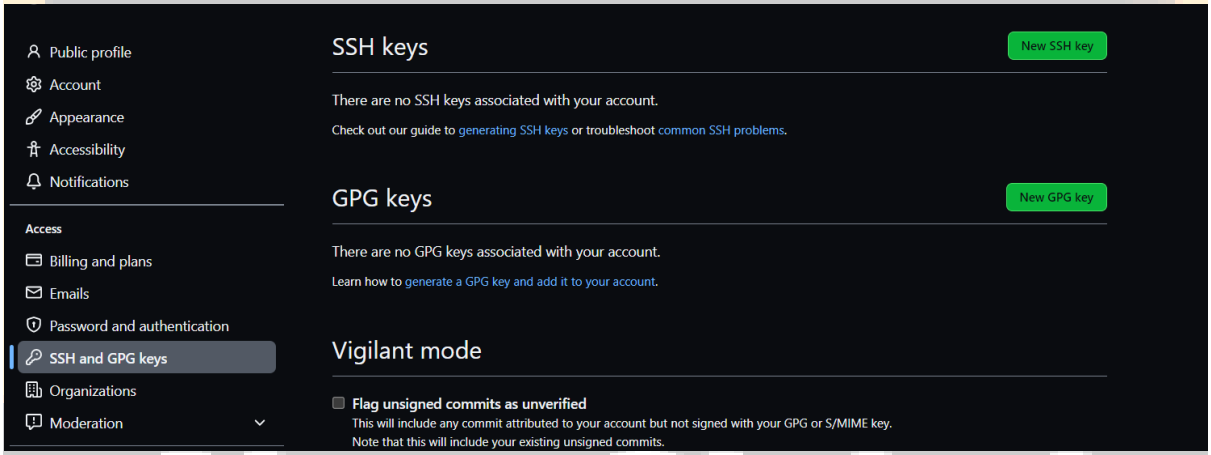

Figure 6.10 Github SSH Key

Now, SSH key form will be showed. Insert your preferred name in title input, choose ‘Authentication Key’ as key type, and paste your copied key from cPanel SSH Access in key input. Make sure there is no line-breaking (line with no character) at the end of the key to avoid error. Click ‘Add SSH Key’ after complete. See Figure 6.11 for reference.

Figure 6.11 SSH/ Add New Key in Github

6.4 Git Repository

6.4.1 Add team leader as collaborator

Add team leader as collaborators in GitHub Repository so that cPanel Git can gain access to the private repository as shown in the Figure 6.12.

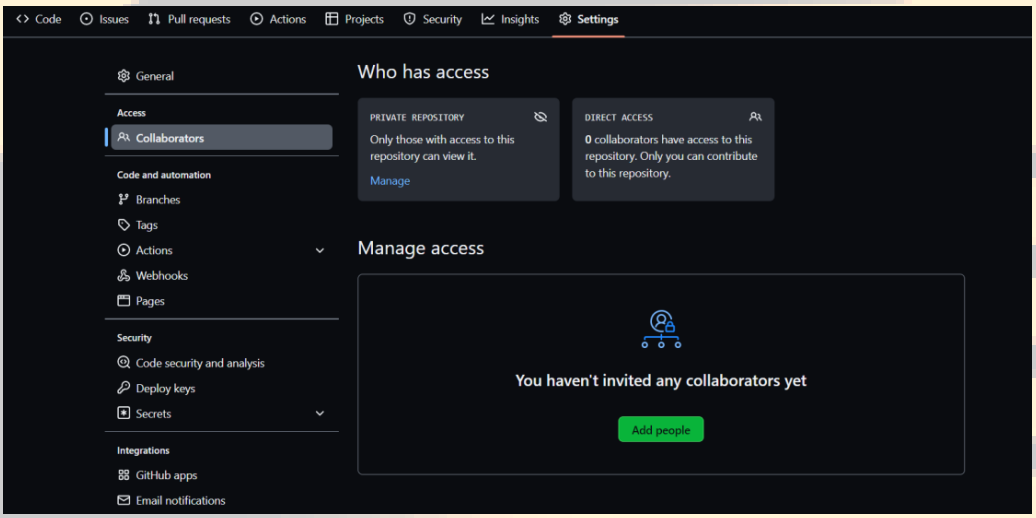

Figure 6.12 Add Collaborators

6.4.2 Create and clone repository in cPanel Git Version Control

Go to the Git™ Version Control in the cPanel as shown in the Figure 6.13. Click 'Create' as shown in Figure 6.14 and a form will appear.

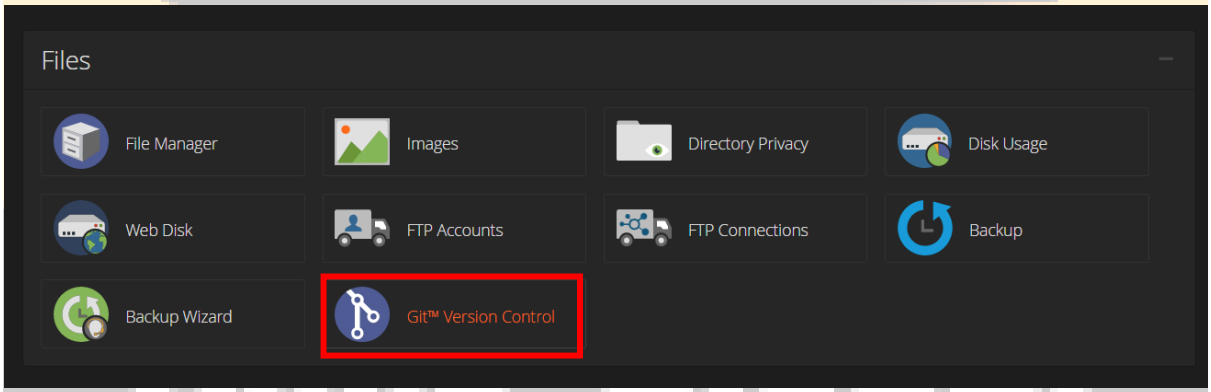

Figure 6.13 Git Version Control

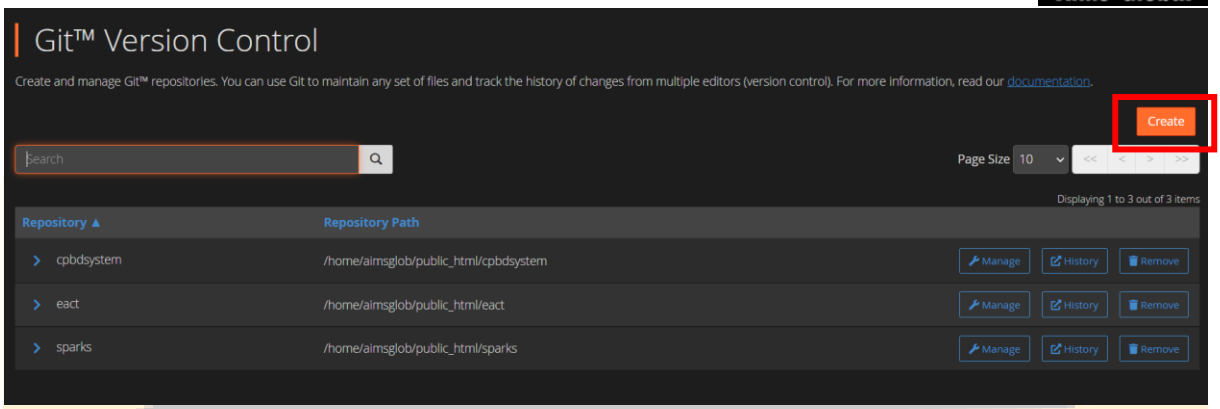

Figure 6.14 Git Repositories Create

Follow steps below to complete Git Version Control Form. See Figure 6.15 and Figure 6.16 for references.

- Go to GitHub Repository and copy SSH link for that repository (not https link).
- Paste the SSH link into Clone URL input.
- Make sure repository path is “public_html/<system_name>”. System name will be repository cloned folder. Make sure the folder is not existed or an empty folder (directory).
- Repository name can be anything. However, default name is recommended.
- Click ‘Create’ button after complete and a success notification will appear. You will be redirected to cPanel Git Repository List.

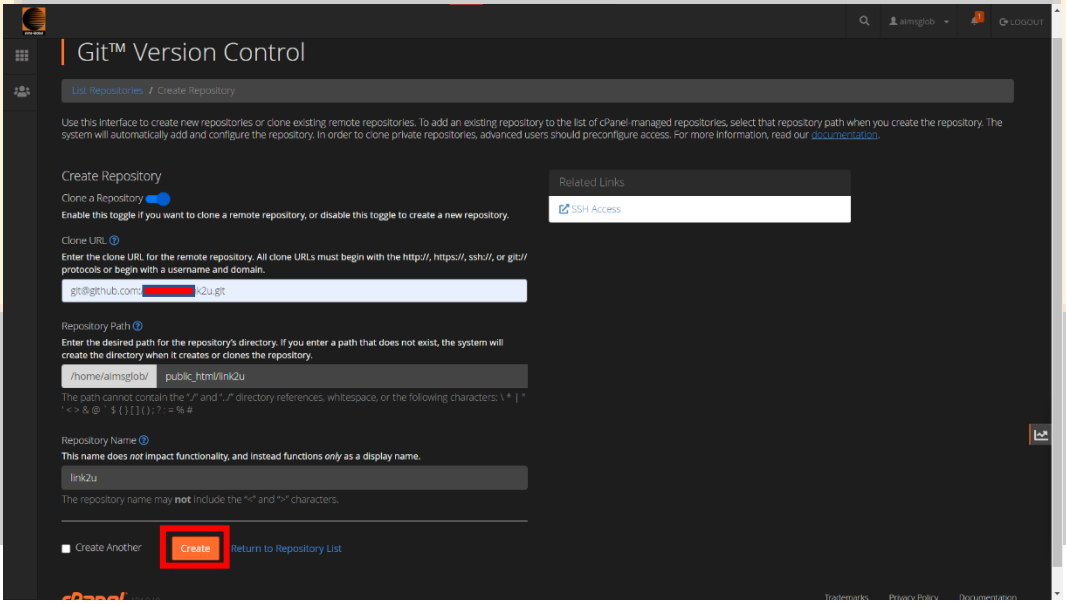

Figure 6.15 Form Create Git Repositories

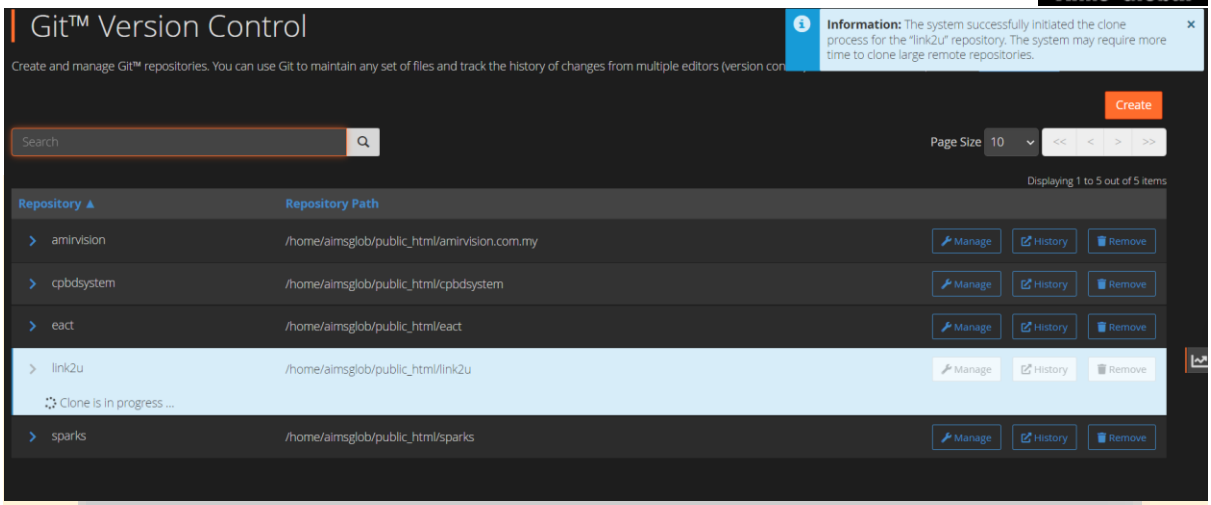

Figure 6.16 Successful Create Git Repositories

6.5 Change Permission

After cloning process is completed, you can see the system folder in File Manager with 0700 permission. This permission will cause the system cannot be load from web browser and will produce 'File not Found' text. The folder permission needs to be change to 0755 to be accessed from web browser. The steps are as follows:

1. Open File Manager from cPanel Tools Interface. See Figure 6.17.
2. Go to the cloned system directory. In this case is 'public_html/link2u'.
3. Click system folder and click permission button at the top of page as shown in Figure 6.18. You also can use right-click to open permission settings.
4. Change permission to 0755. Please refer to Figure 6.19

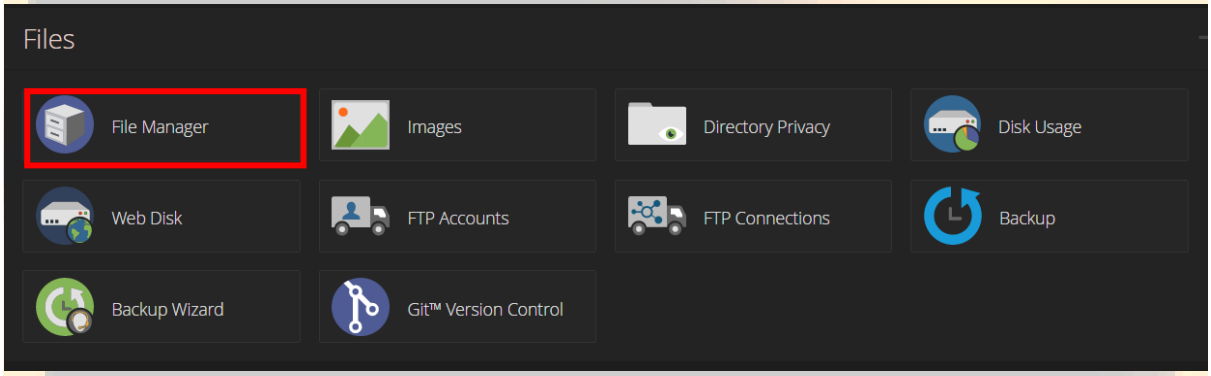

Figure 6.17 File Manager

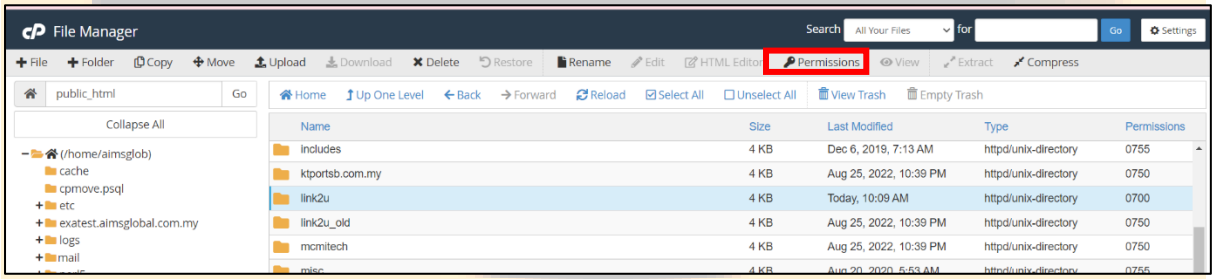

Figure 6.18 Permission button

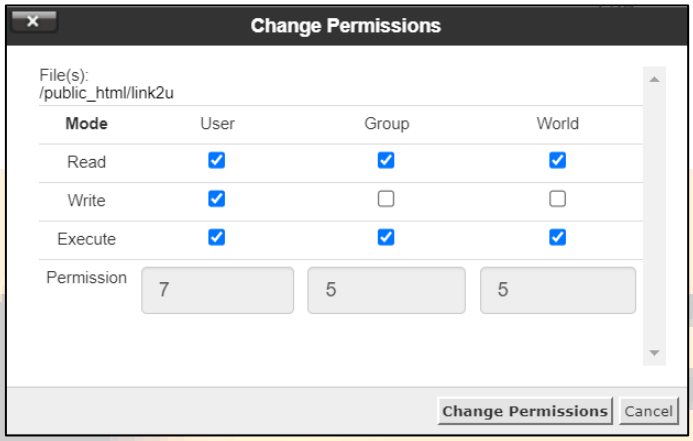

Figure 6.19 Change Permission Popup

6.6 Pull changes from GitHub

Pull changes from GitHub will update all modified files, add new files, or delete any files according to commit from GitHub repository. After completing any changes from local devices, commit and push the changes into GitHub, then you can use cPanel Git Version control to pull changes from GitHub. The steps are as follows:

1. Go to the Git™ Version Control as shown in the Figure 6.20
2. go to Pull and deploy section
3. click button Update from Remote as shown in the Figure 6.21.

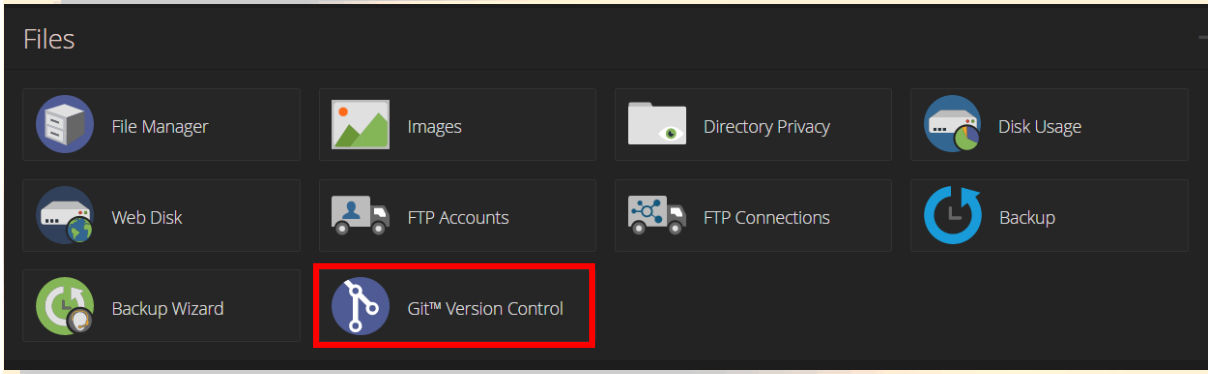

Figure 6.20 Git Version Control Icon

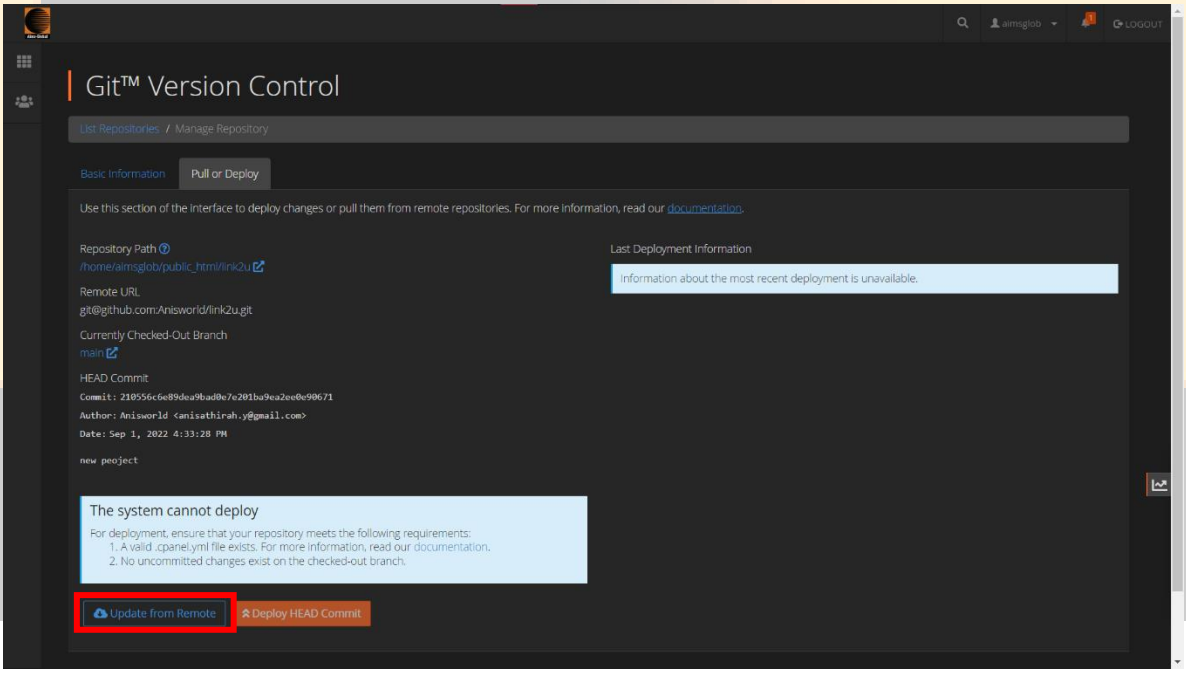

Figure 6.21 cPanel Git Pull or Deploy