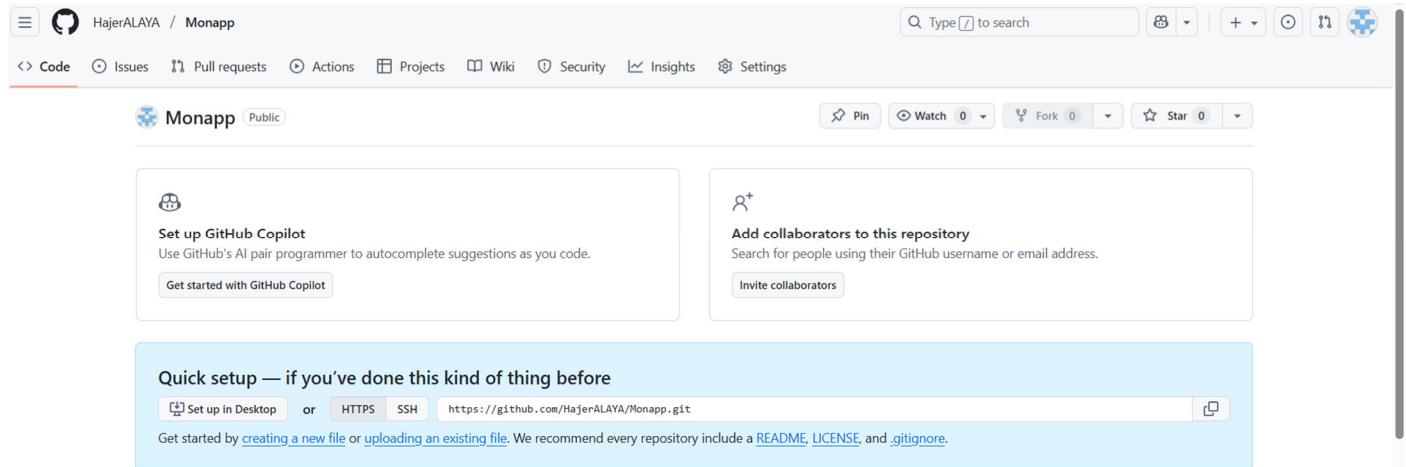


# Guide de l'utilisation de Git et Github pour une application Web PHP

## 1. Vérifier que vous avez installé Git

Vérifier que vous avez un compte sous Github ou créer le

1. Create account at [github.com](https://github.com)
2. Create a new repository on GitHub Name "Monapp"



The screenshot shows a GitHub repository named 'Monapp' owned by 'HajerALAYA'. The repository is public. Key elements include:

- Code**, **Issues**, **Pull requests**, **Actions**, **Projects**, **Wiki**, **Security**, **Insights**, **Settings** navigation tabs.
- A main repository card for 'Monapp' with a 'Public' badge.
- A 'Set up GitHub Copilot' section with a 'Get started with GitHub Copilot' button.
- An 'Add collaborators to this repository' section with a 'Invite collaborators' button.
- A 'Quick setup — if you've done this kind of thing before' box containing:
  - 'Set up in Desktop' or 'HTTPS' or 'SSH' cloning options.
  - A URL: <https://github.com/HajerALAYA/Monapp.git>.
  - Instructions: 'Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

Exemple : <https://github.com/HajerALAYA/Monapp.git>

## 2. Initialisé Git Repository

### Navigate to Your Project

bash

```
# Il faut que vous avez déjà executer le tp1 et vous avez deploier l'app monapp sous xampp  
# Sous git bash taper la commande
```

```
cd C:\xampp\htdocs\monapp
```

### Initialize Git

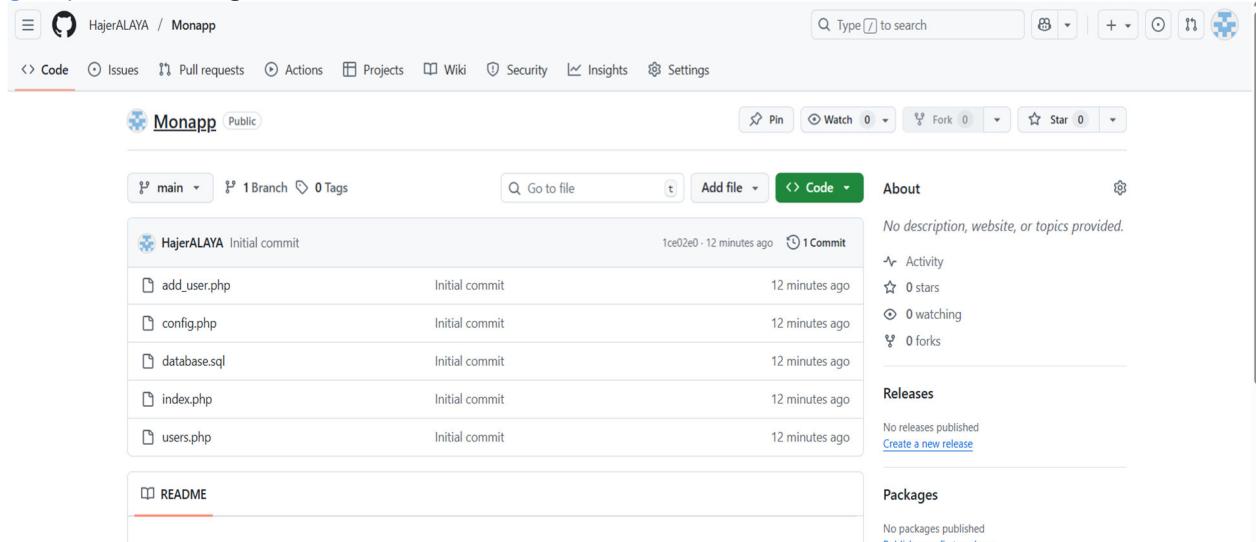
```
# Sous git bash taper les commandes  
bash  
  
git init  
git add .  
git commit -m "Initial commit"
```

### 3. Connect to GitHub Repository

#### Link Local Repository to GitHub

bash

```
git remote add origin https://github.com/HajerALAYA/Monapp.git  
git branch -M main  
git push -u origin main
```



### 4. Daily Workflow

#### Making Changes

1. **Pull latest changes** (if working in team):

bash

```
git pull origin main
```

2. **Make your changes** to PHP/HTML/CSS/JS files in XAMPP

Changer par exemple le fichier add\_user.php

3. **Check status** of changes:

bash

```
git status  
$ git status  
On branch main  
Your branch is up to date with 'origin/main'.  
  
Changes not staged for commit:  
  (use "git add <file>..." to update what will be committed)  
    (use "git restore <file>..." to discard changes in working directory)  
      modified:   add_user.php  
  
no changes added to commit (use "git add" and/or "git commit -a")  
(base)
```

#### 4. Add specific files or all changes:

bash

```
# Add specific file  
git add filename.php  
ou  
# Add all changes  
git add .  
ou  
# Add all PHP files only  
git add *.php
```

```
DELL@HAJER-ALAYA MINGW64 /c/xampp/htdocs/monapp (main)  
$ git add add_user.php  
(base)
```

Exemple :

#### 5. Commit changes:

bash

```
git commit -m "fichier add_user.php modifier"
```

```
DELL@HAJER-ALAYA MINGW64 /c/xampp/htdocs/monapp (main)  
$ git commit -m "fichier add_user.php modifier"  
[main eca9cd5] fichier add_user.php modifier  
 1 file changed, 1 insertion(+), 1 deletion(-)  
(base)
```

#### 6. Push to GitHub:

bash

```
git push origin main
```

The screenshot shows a GitHub repository named 'Monapp'. The repository has one branch ('main') and no tags. The commit history shows an initial commit by 'HajerALAYA' at 26 minutes ago, followed by a commit at 2 minutes ago that modified 'add\_user.php'. The repository has 0 stars, 0 forks, and 0 releases. The README file is present but empty.

File	Description	Time
add_user.php	fichier add_user.php modifier	2 minutes ago
config.php	Initial commit	26 minutes ago
database.sql	Initial commit	26 minutes ago
index.php	Initial commit	26 minutes ago
users.php	Initial commit	26 minutes ago

## 5. Important Git Commands for Development

### View History

```
bash

git log
git log --oneline
```

### Check Differences

```
bash

# See what changed but not staged
git diff

# See staged changes
git diff --staged
```

### Undo Changes

```
bash

# Discard changes in working directory
git checkout -- filename.php

# Unstage a file
git reset HEAD filename.php

# Reset to previous commit
git reset --hard HEAD^
```

## 6. .gitignore for XAMPP Projects

Le **.gitignore** est essentiel pour :

-  **Protéger** les données sensibles
- **Garder propre** le repository Git
-  **Adapter** aux spécificités XAMPP
-  **Optimiser** les performances de Git

> Ce PC > Disque local (C:) > xampp > htdocs > monapp

Nom	Modifié le	Type	Taille
add_user.php	26/10/2025 7:59 PM	Fichier PHP	2 Ko
config.php	27/09/2025 8:52 PM	Fichier PHP	1 Ko
database	27/09/2025 10:47 PM	SQL Text File	1 Ko
index.php	27/09/2025 8:52 PM	Fichier PHP	1 Ko
users.php	27/09/2025 8:52 PM	Fichier PHP	1 Ko
.gitignore	26/10/2025 8:21 PM	txtfile	1 Ko

Create `.gitignore` file in your project root use blocnote:

```
text

# XAMPP specific
/xampp/
/htdocs/other-projects/

# Log files
*.log

# Environment files
.env
config.php

# Temporary files
tmp/
temp/

# Database files (if stored in project)
*.sql
*.mdb

# OS files
.DS_Store
Thumbs.db

# IDE files
.vscode/
.idea/
```

```
*.swp  
*.swo
```

## 7. Branching Workflow (Recommended)

### Create Feature Branch

**Feature Branch** est une branche temporaire utilisée à des fins de développement ou de test.

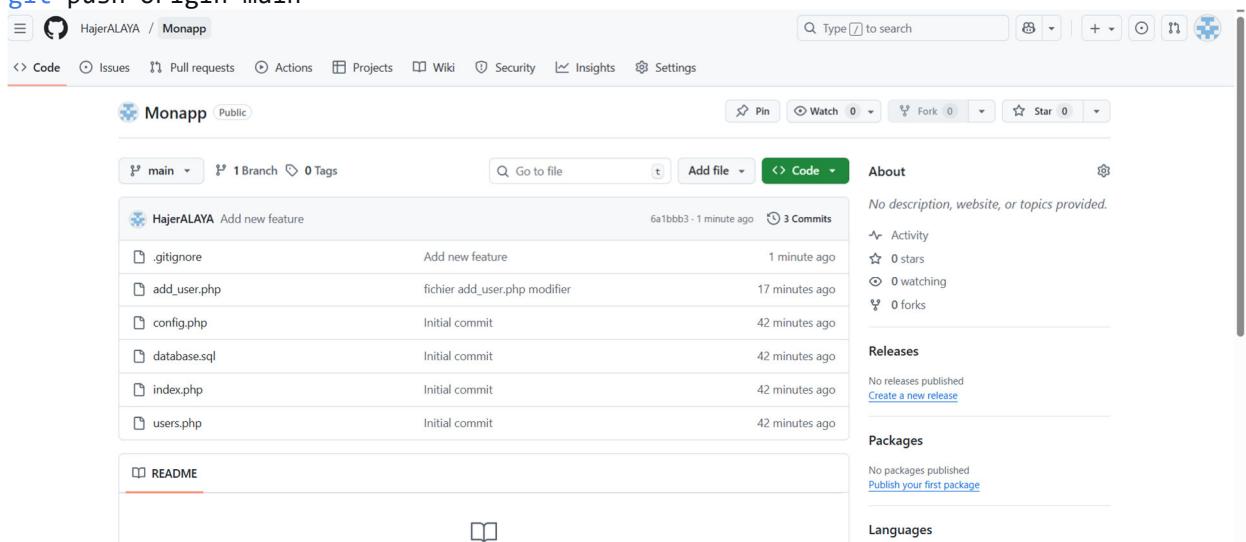
```
bash  
  
git checkout -b feature/new-feature
```

### Work on Feature

```
bash  
  
# Make changes, then:  
git add .  
git commit -m "Add new feature"
```

### Merge Back to Main

```
bash  
  
git checkout main  
git pull origin main  
git merge feature/new-feature  
git push origin main
```



The screenshot shows a GitHub repository interface for a project named 'Monapp'. The repository has 1 branch and 0 tags. The 'main' branch is selected. The commit history shows three commits from 'HajerALAYA' made one minute ago, all titled 'Add new feature'. The commits affect files like '.gitignore', 'add\_user.php', 'config.php', 'database.sql', 'index.php', and 'users.php'. The repository has 0 stars, 0 forks, and 0 releases. It also has 0 packages published and 0 languages listed.

### Delete Feature Branch

```
bash
```

```
git branch -d feature/new-feature
```

## 8. Database Considerations

### For Database Changes

Dans le cas où il y a des changements au cours de développement au niveau de la base de données :

1. **Export database** from phpMyAdmin regularly
2. **Include SQL files** in version control if needed
3. **Or keep separate** and document schema changes in commits

### Example workflow:

Ajouter par exemple une nouvelle table dans mysql puis prépare le fichier database\_backup.sql et tester les commandes

bash

```
# Export database
mysqldump -u root -p your_database > database_backup.sql

# Add to git if needed
git add database_backup.sql
git commit -m "Update database schema"
```

## 9. Team Collaboration

### Clone Existing Repository

bash

```
cd C:\xampp\htdocs\
git clone https://github.com/HajerALAYA/Monapp.git
```