**GITHUB**

**What is GitHub?**

GitHub is a web-based platform that uses Git, a version control system, to manage and store code. It offers hosting for software development and version control using Git. GitHub provides all of the distributed version control and source code management (SCM) functionality of Git as well as adding its own features.

**What is Git Bash?**

Git Bash is an application for Microsoft Windows environments that provides an emulation layer for a Git command-line experience. It is a package that installs Bash, some common bash utilities, and Git on a Windows operating system. Bash is a popular text-based shell and command language.

**Why We Use GitHub**

1. **Version Control**: Allows multiple developers to work on a project simultaneously without overwriting each other’s work.

2. **Collaboration**: Provides tools for team collaboration, including code review, issue tracking, and project management.

3. **Backup**: Acts as a backup for your codebase.

4. **Open Source Contributions**: Facilitates contributing to open source projects.

5. **Documentation**: Allows you to host documentation for your projects.

6. **CI/CD Integration:** Integrates with continuous integration and continuous deployment tools.

7. **Community**: Offers a community for developers to share, discuss, and improve code.

**Basic GitHub Commands and Workflow**

**Setting Up Git**

1. Install Git:

- For Windows: Download and install from [Git for Windows](https://gitforwindows.org/).

- For macOS: Install via Homebrew: `brew install git`.

- For Linux: Install via package manager: `sudo apt-get install git` (Debian-based), `sudo yum install git` (Red Hat-based).

2. Configure Git:

bash

git config --global user.name "Your Name"

git config --global user.email "your.email@example.com"

**Creating a Repository**

1. Initialize a New Repository:

bash

git init

2. Clone an Existing Repository:

bash

git clone https://github.com/username/repository.git

# Basic Commands

1. Add Files to Staging Area:

bash

git add <filename>

git add . # Adds all files in the current directory

2. Commit Changes:

bash

git commit -m "Your commit message"

3. Push Changes to Remote Repository:

bash

git push origin main # Pushes to the main branch

4. Pull Changes from Remote Repository:

bash

git pull origin main

5. Check Status of Repository:

bash

git status

6. View Commit History:

bash

git log

Adding a Folder to GitHub

1. Create a Local Repository (if not already created):

bash

mkdir my\_project

cd my\_project

git init

2. Add Folder and Files:

bash

mkdir new\_folder

touch new\_folder/file.txt

git add new\_folder

git commit -m "Add new folder with file"

3. Push Changes to GitHub:

bash

git remote add origin https://github.com/username/repository.git

git push -u origin main

Updating a Repository

1. Make Changes Locally: Edit or add files as needed.

2. Add and Commit Changes:

bash

git add .

git commit -m "Updated files"

3. Push Changes to Remote Repository:

bash

git push origin main

Branching and Merging

1. Create a New Branch:

bash

git branch new-feature

git checkout new-feature

2. Merge Branch into Main Branch:

bash

git checkout main

git merge new-feature