Day 3 Notes

GIT, GITHUB & PYTHON VIRTUAL ENVIRONMENT

- **Difference between Git, Github, and Gitlab**

Git

- A **version control system**
- Used to track changes in code
- Works **locally on your computer**
- Helps manage different versions of files

Example: tracking changes in app.py

GitHub

- A **cloud-based hosting platform** for Git repositories
- Used to store code **online**
- Supports collaboration, pull requests, issues

Example: github.com/username/project

GitLab

- Similar to GitHub
- Provides **Git repository hosting**
- Has built-in **CI/CD tools**
- Often used in companies

2. Basic Git codes

```
Git clone https    #Creates a local copy of a remote repository from GitHub/GitLab.

Git pull    #Fetches latest changes and merges them into the current branch.

Git fetch --all    #Downloads updates from all remotes without changing local code.
Git branch --all       #Shows all branches (local + remote).

Git branch       #Shows only local branches.

Git checkout branchname       #Switches to an existing branch.

Git checkout -b newbranch name       #Creates a new branch and switches to it.
```

Example files:

App.py

Main.py

App1.py

Main5.py

```
Git add .  #Adds all changed files (new/modified/deleted) in the current directory to staging.

Git add app.py    #Adds only app.py(foldername) to staging.(foldername)

Git commit -m message    #Saves staged changes as a commit with a message.

Git push     #Uploads committed changes to the remote repository.
```

3. What is a Virtual Environment?

- An **isolated Python environment**
- Keeps project dependencies separate
- Prevents version conflicts

4. How to create a virtual environment in Python?

```
Python -m venv foldername     #creating a virtual environment (isolate python)

Sriya/script/activate    #to activate the environment

Pip install -r requirements.txt    #to install requirements for the virtual environment

Python app.py        #Runs the Python application.

Deactivate        #Exits the virtual environment.
```

Git + Virtual Environment Workflow (Real World)

1. Clone project from GitHub
2. Create virtual environment
3. Activate environment
4. Install requirements
5. Run the project
6. Make changes
7. Add → Commit → Push