Introduction to Software Development Week 10: Version Control with Git & GitHub

1. Learning Objectives

- Explain the purpose of a Version Control System (VCS).
- Initialize a Git repository and perform the basic workflow: `add`, `commit`.
- Understand the difference between a local repository (Git) and a remote repository (GitHub).
- Push a local repository to GitHub.

2. Core Concepts

- **What is Version Control?**
  - A system that records changes to a file or set of files over time so that you can recall specific versions later.
  - It's like an unlimited "undo" button for your entire project.
  - Essential for collaboration, allowing multiple developers to work on the same project without overwriting each other's work.
- **Git: The Local Repository**
  - Git is a distributed version control system. Every developer has a full copy of the project's history on their local machine.
  - **Repository (`.git` folder):** The database tracking all changes.
  - **Working Directory:** The actual files you are working on.
  - **Staging Area (Index):** A "drafting" area where you prepare what will be included in the next "snapshot" (commit).
- **The Basic Git Workflow:**

  1. **Modify** files in your working directory.
  2. **`git add <filename>`:** Stage the changes (move them to the staging area).
  3. **`git commit -m "Descriptive message"`:** Take a permanent snapshot of the staged files and save it to the repository's history.
- **GitHub: The Remote Repository**
  - A web-based platform for hosting Git repositories.
  - Enables collaboration, code sharing, and project management.
  - **`git push`:** Uploads your committed changes from your local repository to a remote repository like GitHub.
  - **`git pull`:** Downloads changes from a remote repository to your local one.
  - **`git clone`:** Creates a local copy of a remote repository.

3. Key Commands Summary

- `git init`: Initialize a new, empty repository.
- `git status`: Show the status of the working directory and staging area.
- `git add <file>`: Add a file to the staging area.
- `git commit -m "message"`: Commit staged changes with a message.
- `git log`: View the commit history.
- `git push`: Send commits to a remote repository.

4. Summary

Git and GitHub are fundamental tools for modern software development. Mastering this workflow is non-negotiable for collaboration and for managing any project, big or small.