

Lesson 2: Mastering Git & GitHub with a Fun Example and Demo Code for Your Expense Tracker Project

Objective: Become a Git & GitHub pro, capable of leading a team project with confidence and ease.

The Fun Example: Building a Treehouse (Your Expense Tracker Project)

Imagine you and your friends are building a treehouse (the expense tracker project). You need a way to organize who does what and keep track of all the changes without getting in each other's way. Git is like your toolbox, and GitHub is the tree where you're building the house. Each tool (Git command) helps you build more efficiently.

1. Setting Up Your Workspace:

- **git init:** Creates a new project. It's like laying the foundation of your treehouse. In your project folder, run:

```
git init
```

- **git clone:** Copies an existing treehouse to your location. Use this to get a copy of the expense tracker from GitHub:

```
git clone https://github.com/username/expense-tracker.git
```

2. Adding Pieces to the Treehouse (Staging Changes):

- **git status:** Checks what's been changed. It's like looking at what parts of the treehouse have been worked on:

```
git status
```

- **git add:** Adds your changes to the blueprint (staging area). Let's say you added a new feature to track expenses by category:

```
git add categories.py
```

3. Saving Your Work (Committing Changes):

- **git commit:** Saves your changes. It's like taking a photo of your treehouse progress:

```
git commit -m "Add expense category feature"
```

4. Sharing Your Work (Pushing Changes):

- **git push:** Updates the treehouse on the GitHub tree with your changes:

```
git push origin main
```

5. Getting Updates (Pulling Changes):

- **git pull:** Gets the latest updates from the tree. Before you start working, always run:

```
git pull
```

6. Creating New Designs (Branching):

- **git branch:** Creates a new branch. It's like sketching a new idea for a treehouse extension:

```
git branch feature/new-hammock
```

- **git checkout:** Switches to your new branch. It's like moving to work on that part of the treehouse:

```
git checkout feature/new-hammock
```

7. Combining Designs (Merging):

- **git merge:** Combines changes from one branch into another. When your hammock design is ready, merge it into the main design:

```
git checkout main  
git merge feature/new-hammock
```

8. Resolving Design Conflicts (Merge Conflicts):

- When two friends work on the same part of the treehouse, you need to decide which design to keep. Edit the files to resolve the conflicts, then save the changes with `git add` and `git commit`.

9. Keeping Track of Changes (Git Log):

- **git log:** Shows the history of changes. It's like a guestbook for everyone who's worked on the treehouse:

```
git log
```

10. Saving Work for Later (Stashing):

- **git stash:** Puts your changes in a safe place while you work on something else. It's like putting your tools in a box while you help a friend:

```
git stash  
git stash apply
```

11. Leading the Team Project:

- As a team leader, you're responsible for organizing the work. Use GitHub issues to assign tasks, pull requests to review code, and project boards to track progress.
- Communicate with your team. Have regular "treehouse meetings" (stand-ups) to discuss what everyone is working on.
- Encourage good commit messages and frequent commits. It's like updating the treehouse blueprint and taking lots of photos to show how it's evolving.