**Git -Hub**

**1. Repository (Repo)**

* A storage place where your project files and their version history are kept.
* Can be **local** (on your computer) or **remote** (on GitHub).

Example: A folder named MyProject that Git tracks.

**2. Commit**

* A snapshot of your project at a certain point in time.
* Like “saving a checkpoint” in a game.
* You usually add a **message** describing what changed.

📌 Command:

***git commit -m "Added login feature"***

**You add a new login.java file → commit = “save this change”.**

## 3. ****Branch****

* A separate line of development.
* Allows you to work on a feature without affecting the main project.
* Default branch on GitHub = **main** (older Git used **master**).

📌 Example:

* main branch = stable project
* feature-login branch = experimental login system
* Later, you can **merge** it back into main.

## 4. ****Push****

* Sends your local commits (your work) to the **remote repository** (GitHub).

📌 Command:

**git push origin main**

📌 Example:

* You made commits on your laptop → push uploads them to GitHub so your teammates can see.

## 5. ****Pull****

* Downloads the latest changes from the **remote repository** into your **local repo**.
* Basically = fetch (download) + merge (combine).

📌 Command:

**git pull origin main**

📌 Example:

* Your teammate added a dashboard.java file → you run git pull → now you also have that file.

## 6. ****Clone****

* Makes a copy of a remote repository (from GitHub) onto your computer.

📌 Command:

**git clone https://github.com/user/repo.git**

📌 Example:

* You see a project on GitHub → git clone to download and start working on it.

## 7. ****Stage / Index (git add)****

* Before committing, you must **stage** files using git add.
* This tells Git: “Include these files in the next snapshot (commit).”

📌 Example:

**git add file1.java file2.java**

## 8. ****Merge****

* Combines changes from one branch into another.

📌 Example:

* You finished feature-login branch → merge it into main.

## 9. ****Remote****

* A connection to a repository stored somewhere else (like GitHub).
* Usually named **origin**.

📌 Example:

**git remote add origin https://github.com/user/repo.git**

**Quick Analogy (Git = Working on a School Project):**

* **Repository** → The project notebook
* **Commit** → Taking a photo of the notebook after writing notes
* **Branch** → A separate notebook page for experiments
* **Push** → Sending your notebook photo to the teacher (GitHub)
* **Pull** → Downloading your teammate’s updated notes
* **Merge** → Combining your notes with your teammate’s