Solution for Lab#01: Introduction to Version Control with Git and GitHub

**1. Difference Between Git and GitHub**

| **Feature** | **Git** | **GitHub** |
| --- | --- | --- |
| **Definition** | A distributed version control system that manages code changes locally. | A cloud-based hosting service that allows developers to store, collaborate, and manage Git repositories. |
| **Functionality** | Tracks file changes, branches, and commits locally. | Provides remote repository hosting, collaboration tools, and issue tracking. |
| **Usage** | Used via the command line for version control. | Web-based interface with additional tools for team collaboration. |
| **Installation Required?** | Yes, must be installed on a local system. | No, accessed via a web browser or GitHub Desktop. |
| **Example Command** | git commit -m "message" | git push origin main |

**2. Commands Used to Check Repository Status and Commit History**

| **Task** | **Command** |
| --- | --- |
| Check repository status | git status |
| View commit history | git log |

**3. Command to Push Local Changes to Remote Repository**

git push origin main

**4. Command to Create and Switch to a New Branch**

git branch feature-branch # Create a new branch

git checkout feature-branch # Switch to the new branch

OR (shortcut in newer Git versions):

git checkout -b feature-branch

**5. Clone a Repository from GitHub and Screenshot**

**Command to Clone a Repository:**

git clone <repository-url>

Example:

git clone https://github.com/user/my-git-project.git

* After running this command, take a **screenshot** of the terminal output showing successful cloning.

**Conclusion**

In this lab, we successfully installed Git, configured user settings, initialized a repository, committed changes, and pushed them to GitHub. We also practiced working with branches, cloning repositories, and checking repository status. Mastering Git and GitHub helps in efficient code management and team collaboration in software development.