

LAB 1

Source Code Management with Git

Objective:

The main objective of this lab was to get practical experience with **Git**, a widely-used source code management tool. By the end of the session, we were expected to:

- Grasp the fundamentals of version control
 - Create and manage Git repositories
 - Use essential commands like **add**, **commit**, **push**, **pull**, and **clone**
 - Work with branches and handle merge conflicts
 - Collaborate effectively through **GitHub**
-

Tools Used

During this lab, we worked with the following tools:

- **Git** — for version control and managing project history
 - **GitHub** — to host our repositories and collaborate online
 - **Command Line / Terminal** — to execute Git commands
 - **VS Code** — as our text editor and coding environment
-

Key Learnings

Before diving into the actual commands, we briefly went over the importance of source code management. It's essentially a way to track code changes, maintain backups, and collaborate efficiently on projects.

Some core concepts we covered included:

- **Repository**: A directory or storage space where Git tracks project files
- **Commit**: A record capturing the state of the project at a specific moment
- **Branching**: Creating a separate line of development for new features or fixes
- **Merging**: Integrating changes from different branches into one

Steps Followed

1. Initialize a Git Repository

```
mkdir Agile-Lab // file name
cd Agile-Lab
git init
```

2. Add and Commit a File

```
echo "# My First Git Project" > README.md
git add README.md
git commit -m "Initial commit"
```

3. Push to GitHub

```
git remote add origin https://github.com/Prabesh61/Agile-Lab.git
git push -u origin master
```

4. Clone a Repository

```
git clone https://github.com/Prabesh61/NewProject.git
```

5. Create and Merge Branches

```
git checkout -b feature-login
# Made some changes...
git add .
git commit -m "Added login feature"
git checkout master
git merge feature-login
```

Conclusion

This lab provided valuable hands-on experience with Git and showed how version control simplifies project management. It made clear how essential tools like Git and GitHub are for organizing code and collaborating efficiently in both solo and team projects.