Lab 1

Title: Introduction to Source Code Management using Git

Objective:

To understand the fundamentals of source code management using Git.

- Initialize a Git repository
- Track changes using git add, git commit
- View version history
- Use branching and merging
- Push code to a remote repository (e.g., GitHub)

Theory:

Source Code Management (SCM) is the practice of tracking and managing changes to code. It allows multiple developers to collaborate on code, track changes, roll back to previous versions, and maintain a history of a project.

Git is a distributed SCM tool widely used in software development. Key Git concepts:

- **Repository**: A Git project directory.
- **Commit**: A snapshot of changes.
- **Branch**: A parallel version of the repository, useful for working on features.
- **Merge**: Integrating changes from one branch into another.
- **Remote Repository**: A version of your project hosted on services like GitHub, GitLab, or Bitbucket.

Code

Step 1: Initialize a Git repository

```
git init
```

Step 2: Check the status of files

```
git status
```

Step 3: Add files to staging area

`git add <filename>` - To add a specific file

`git add .` - To add all files

Step 4: Commit the changes

```
git commit -m "Initial commit"
```

Step 5: Create a new branch and switch to the branch

```
git checkout -b feature-login
```

Step 6: Merge changes back to main

`git checkout main` `git merge feature-login`

Step 7: Push to remote GitHub repo

```
git remote add origin https://github.com/llaxmi/agile.git
```

```
git push -u origin main
```

Conclusion

In this lab, we successfully learned how to manage source code using Git. We covered key commands like init, add, commit, branch, checkout, and push. Understanding Git is essential for collaborative software development, and this lab provided a practical foundation for future work in teams and version-controlled environments.