**Placement Empowerment Program**

**Cloud Computing and DevOps Centre**

**Set up Git Branching:**

**Create a new branch in your Git repo for testing**

**Add a new feature and merge it.**

**Name: AAKASH RAMADURAI B  
Department: MECHANICAL ENGINEERING**



**INTRODUCTION**

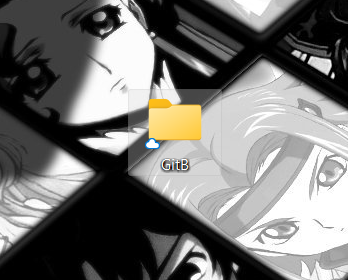
Creating a separate branch in your Git repository for testing enables you to develop new features, fix bugs, or experiment without impacting the main branch (commonly called main or master). This keeps your core codebase stable as you work on and improve new functionality. After implementing and verifying your changes, you can reintegrate the branch into the main one using either a merge or rebase approach. This approach helps maintain a structured and organized version history while promoting seamless collaboration among developers. Additionally, addressing any merge conflicts ensures the final combined code functions properly. Effective use of branching and merging in Git boosts workflow efficiency, minimizes errors, and supports continuous integration and deployment (CI/CD) processes.

**Git Commands used:**

1. Git init
2. Git add .
3. Git commit -m “ “
4. Git checkout -b
5. Git merge
6. Git branch -d

**Step-by-step process:**

**Step 1:** Create a folder in your deskstop.

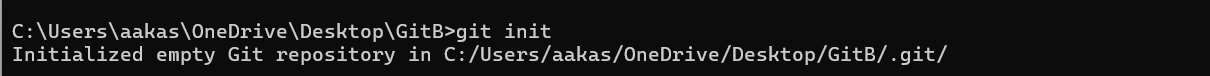


**Step 2:** Open the cmd prompt and set the path to the folder that you created.

**Step 3**: Initialize the git cmds for the process.

**Git commands:**

**Cmd 1:** git init



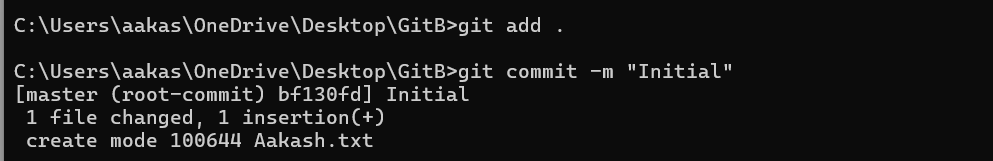
**Note:**

**Before cmd 2 you need to create a text file.**

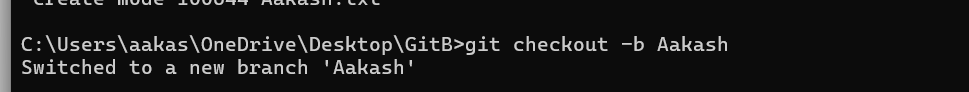


**Cmd 2:** git add .

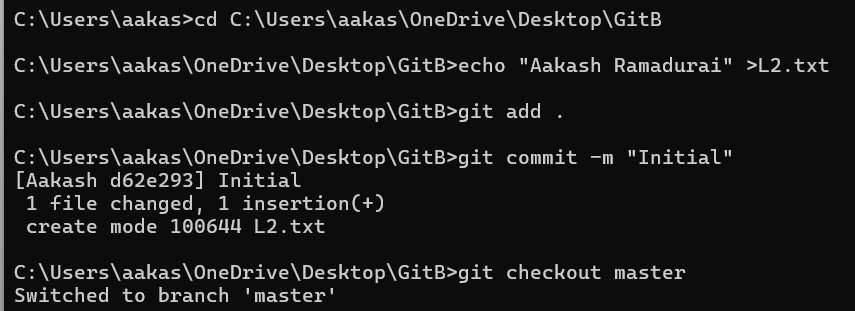


**Cmd 3:** git commit -m

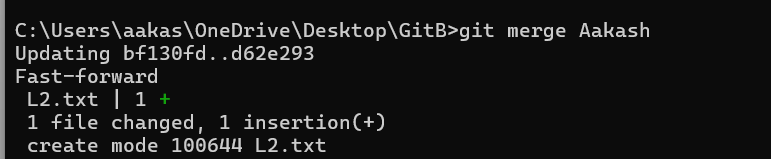
**Cmd 4:** git checkout – for switching to a new branch.



**Step 4**: Repeat the above “Cmds and steps for creating another text file.

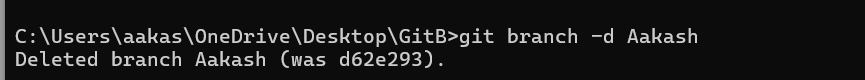


**Step 5:** Merge changes from Aakash to L2.

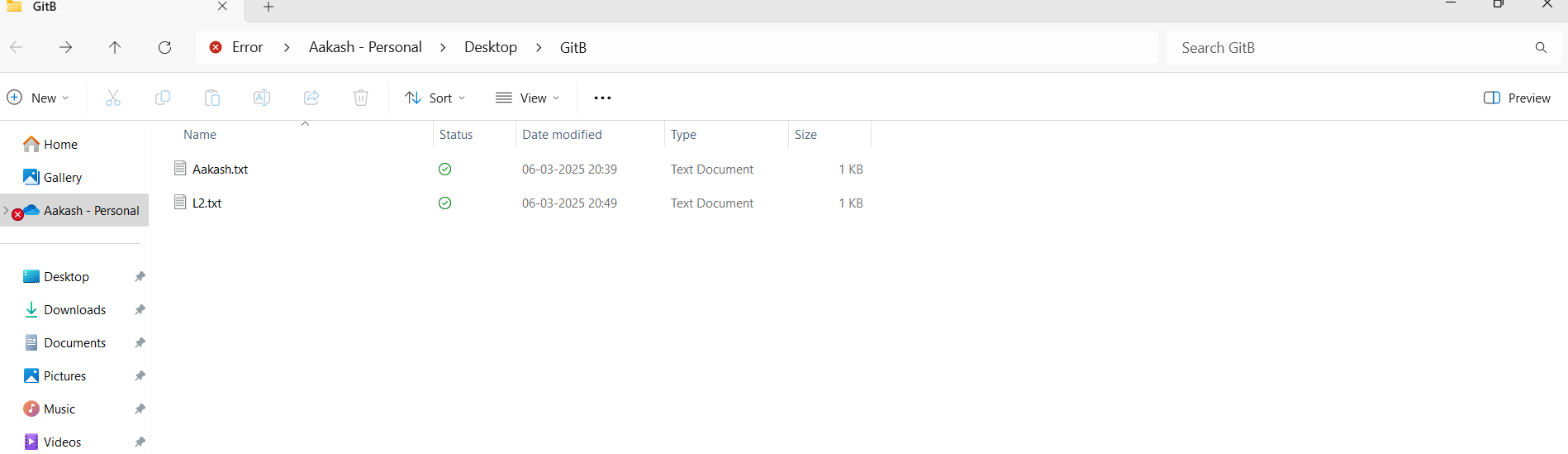


**Step 6:** Once you are done with merging, you can delete the

kalai branch.



**Step 7: C**heck the changes that has been made.



**OUTCOME**

The result of generating a new branch, implementing a feature, and merging it in Git is a well-structured and efficient development process. It enables secure testing and experimentation without interfering with the primary codebase. After merging, the new feature is incorporated into the main branch, ensuring a systematic and controlled integration of updates. This method also enhances teamwork among developers, minimizes conflicts, and preserves a transparent version history, simplifying the tracking of changes.

THANK YOU!