**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: Kalairajan S Department: IT**



**INTRODUCTION**

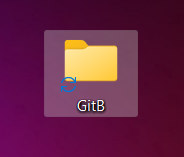
Creating a new branch in your Git repository for testing allows you to work on new features, bug fixes, or experiments without affecting the main branch (often named main or master). This ensures that your primary codebase remains stable while you develop and refine new functionality. Once you’ve made and tested your changes, you can merge the branch back into the main branch using a merge or rebase strategy. This process helps in maintaining a clean and organized version history while enabling smooth collaboration among developers. Additionally, resolving merge conflicts, if any, ensures that the final integrated code functions correctly. Proper use of branching and merging in Git enhances workflow efficiency, reduces errors, and facilitates continuous integration and deployment (CI/CD) practices.

**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.

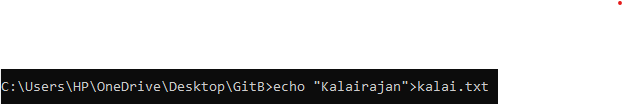
**Git commands:**

**Cmd 1:** git init



**Note:**

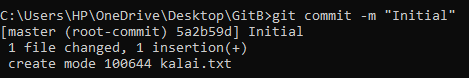
**Before cmd 2 you need to create a text file.**



**Cmd 2:** git add .



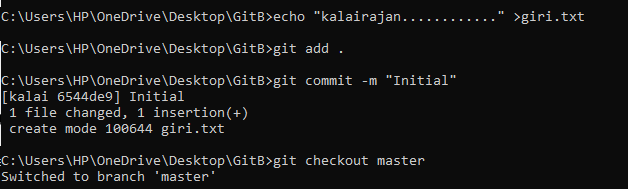
**Cmd 3:** git commit -m



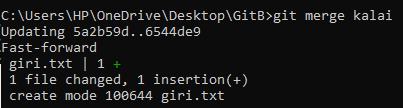
**Cmd 4:** git checkout – for switching to a new branch.



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



**Step 5:** Merge changes from kalai to giri.

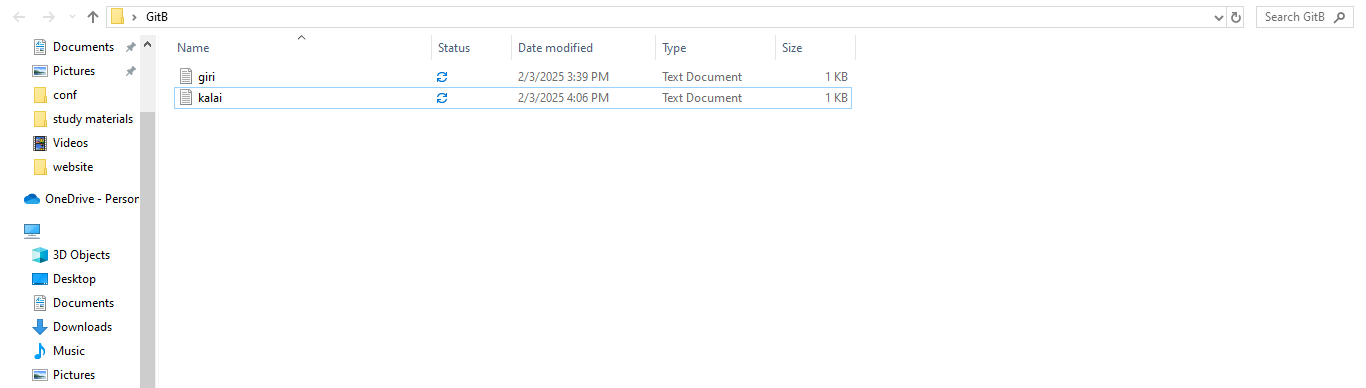


**Step 6:** Once you are done with merging, you can delete the

kalai branch.



**Step 7:**check the changes that has been made.



**OUTCOME**

The outcome of creating a new branch, adding a feature, and merging it in Git is a structured and efficient workflow that enhances software development. It allows for **safe testing and experimentation** without disrupting the main codebase. Once merged, the new feature becomes part of the main branch, ensuring an **organized and controlled integration** of changes. This approach also improves **collaboration among developers**, reduces conflicts, and maintains a **clear version history**, making it easier to track modifications.