A close up of a logo

Description automatically generated

**Disclaimer: The content is curated from online/offline resources and used for educational purpose only**

**Code Repository**

**LAB MANUAL**

**Managing Branches for Feature Development**

**Objective:**

* To understand how to manage branches in Git for organizing and isolating feature development.
* To learn how to create, switch, and delete branches.
* To practice developing features in separate branches and merging them into the main branch.

**Equipment Required:**

* A computer with internet access
* Git installed (git --version to verify)
* GitHub account (or any Git hosting platform)
* Text editor or IDE (e.g., VS Code)

**Prerequisites:**

* Basic knowledge of version control concepts
* Familiarity with Git commands: git init, git status, git add, git commit, git push
* A GitHub repository (can be newly created or cloned)

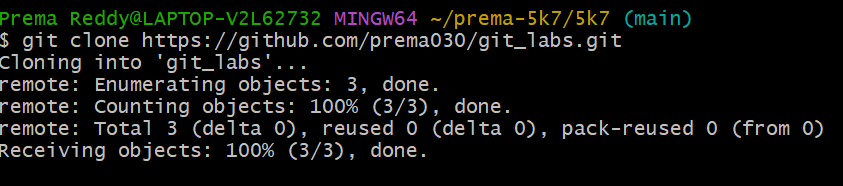
**Problem Statement:**

You are part of a development team working on a web application. Each new feature must be developed in isolation to avoid disrupting the main codebase. Your task is to:

* Create a new branch for a feature,
* Develop and commit code in that branch,
* Merge the branch back into the main branch once the feature is complete.

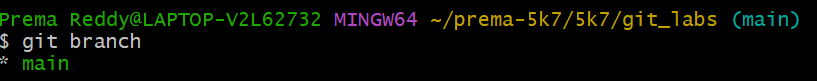
**Procedure:**

**Step 1: Clone or Initialize a Git Repository**

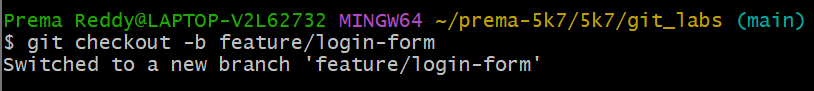
****

****

**Step 2: Check the Current Branch**

****

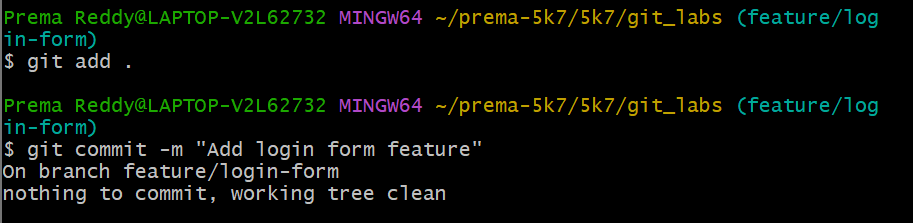
**Step 3: Create a New Branch for the Feature**

****

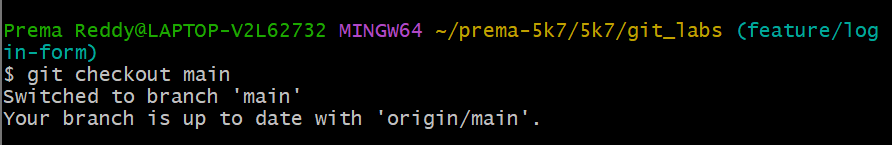
**Step 4: Make Code Changes in the New Branch**

* Open a file or create a new one (e.g., login.html) ****
* Add code related to the feature (e.g., login form UI)

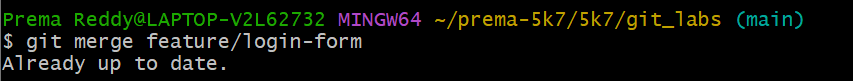
**Step 5: Stage and Commit Your Changes**

****

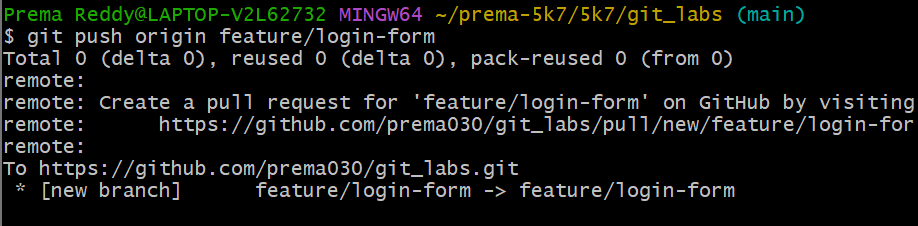
**Step 6: Switch Back to Main Branch**

****

**Step 7: Merge Feature Branch into Main**

****

**Step 8: Push Changes to Remote (if connected)**

****