A close up of a logo

Description automatically generated**ad**

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

**Code Repository**

**LAB MANUAL**

**Staging and Committing Changes**

**Objective:**

* Modify existing files in a Git-tracked project.
* Stage files using Git.
* Commit changes with clear and meaningful commit messages.
* Understand the difference between working directory, staging area, and commit history.

**Equipment Required:**

* Computer with internet access
* Git installed on the system
* GitHub or GitLab account (or any remote repository provider)
* Git Bash (Windows) or Terminal (macOS/Linux)
* Code/text editor (e.g., VS Code, Notepad++)

**Prerequisites:**

1. Understand Git concepts: working directory, staging area, and repository.
2. Have Git installed and configured.
3. Be familiar with basic CLI commands (cd, ls, etc.).
4. Have an existing cloned or initialized Git repository.

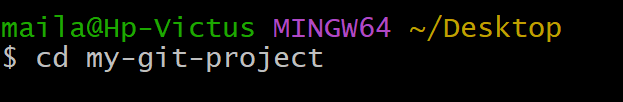
**Problem Statement:**

As a developer, you're tasked with updating documentation in a project. You will:

* Modify a text file in an existing Git repository.
* Review the changes using Git commands.
* Stage the file(s).
* Commit the change with a meaningful message.
* Confirm the change using Git log.

**Procedure:**

**Step 1: Navigate to Your Git Repository**

****

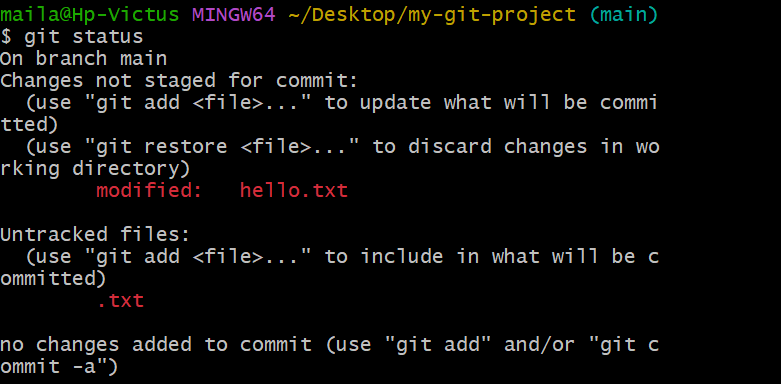
**Step 2: Modify an Existing File**

**Open any file (e.g., README.md or hello.txt) in your editor.**

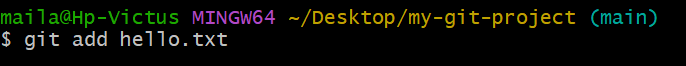
****

**Add the line and Save the file.**

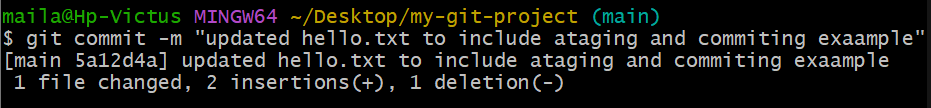
**Step 3: Check the Current Status**

****

**Step 4: Stage the File**

****

**Step 5: Commit with a Meaningful Message**

****