**Lesson-End Project**

**Implementing Basics of Version Control System with Git**

**Project agenda**: To create a remote GitHub repository and use Git commands to perform various version control operations for managing the remote repository

**Description:** You work as a developer in an IT firm. Your company is undertaking a project that consists of three modules, and you have been asked to work on one of these modules. You have also been instructed to upload all the project files to the GitHub repository. To avoid impacting the main codebase, create a new branch and conduct your work there. Once your module is complete, merge it into the main branch.

**Tools required:** Git and GitHub

**Prerequisites:** You must have Git installed in the lab to proceed.

**Expected deliverables**: A step-by-step guide to creating, managing, and deleting branches and files in a Git repository, including scripts and commands to streamline version control for collaborative development

Steps to be followed:

1. Create a new repository
2. Clone the GitHub repository
3. Create a new branch and verify it
4. Rename the existing branch and list all the branches
5. Create a new branch and switch to the new branch
6. Create a file, commit the changes, and check the status of the new branch
7. Delete the branch and verify it
8. Switch back to the main branch and merge

**Step 1: Create a new repository**

1. Open the browser and navigate to [**https://github.com/**](https://github.com/) to log in to your GitHub account using your credentials

**A screenshot of a login form

Description automatically generated**

**Note**: If you do not have a GitHub account, visit the official website at **https://github.com/signup** and create a new account

1. Click on the **+** icon in the upper-right corner of the page and select the **New repository** option from the drop-down menu

A screenshot of a computer

Description automatically generated

1. Enter the **Repository name** as **Control-System-Git**

A screenshot of a computer

Description automatically generated

1. Choose **Public** for the repository type, select **Add a README file**, and click on **Create repository**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Once you create the repository, the above screen will appear.

**Step 2: Clone the GitHub repository**

* 1. In the **Control-System-Git** repository main page, click on **Code**

A screenshot of a computer

Description automatically generated

* 1. Click on the copy icon to copy the **HTTPS URL** as shown below:

A screenshot of a computer

Description automatically generated

* 1. Open the terminal tab and execute the following command to clone the repository:

**git clone <URL>**

**Note**: Replace the URL with the copied URL

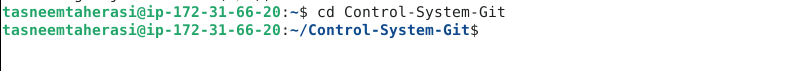
A screenshot of a computer

Description automatically generated

**Step 3: Create a new branch and verify it**

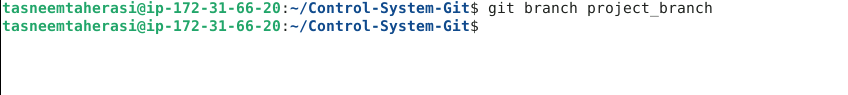
* 1. Navigate to the **Control-System-Git** project directoryby executing the following command:

**cd Control-System-Git**

****

* 1. Execute the following command to create a new branch named **project\_branch** in your repository:

**git branch project\_branch**

****

* 1. Execute the following command to verify the creation of the new branch:

**git branch**

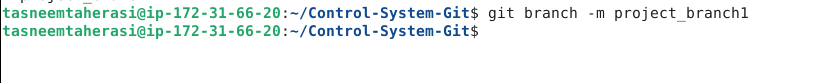
**A close-up of a computer

Description automatically generated**

**Step 4: Rename the existing branch and list all the branches**

1. Use the following command to rename the new branch:

**git branch -m project\_branch1**

****

1. Execute the following command to list the branches to verify the new name of the branch:

**git branch**

**A close-up of blue text

Description automatically generated**

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

* 1. Execute the following command to create a new branch:

**git branch project\_branch2**

**A close-up of a sign

Description automatically generated**

* 1. Execute the following command to verify the creation of the new branch:

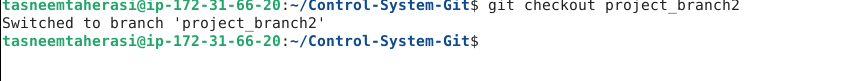
**git branch**

**A close-up of a computer screen

Description automatically generated**

* 1. Use the following command to switch to the newly created **project\_branch2**:

**git checkout project\_branch2**

****

**Step 6: Create a file, commit the changes, and check the status of the new branch**

* 1. Use the following command to create a file:

**vi index.html**

****

* 1. Add the following code to the **index.html** file:

**<html>**

**<body>**

**<p> This is a lesson end project. </p>**

**</body>**

**</html>**

**Note**: Press the **Esc** button and enter **: wq** to save and exit the file

**A screen shot of a computer

Description automatically generated**

* 1. Execute the following command to add the file to the **project\_branch2**:

**git add index.html**

**A close-up of a computer screen

Description automatically generated**

* 1. Use the following command to commit all the modified files to the branch **project\_branch2**:

**git commit -a -m "file modified"**

**A screenshot of a computer

Description automatically generated**

* 1. Check the status of the new branch using the following command:

**git status**

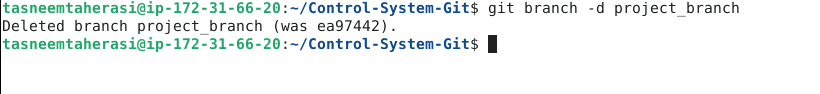
**A close-up of a computer screen

Description automatically generated**

**Step 7: Delete the branch and verify it**

* 1. Execute the following command to delete the newly created branch:

**git branch -d project\_branch**

****

* 1. Verify the deletion of the branch using the following command:

**git branch**

**A screen shot of a computer

Description automatically generated**

**Step 8: Switch back to the main branch and merge**

* 1. Execute the following command to switch back to the main branch:

**git checkout project\_branch1**

A screen shot of a computer

Description automatically generated

* 1. Execute the following command to merge the test branch with the main branch:

**git merge project\_branch2**

A screen shot of a computer

Description automatically generated

By following these steps, you have successfully created a remote GitHub repository and used Git commands to perform various version control operations for managing the remote repository.