**Lesson 02 Demo 06**

**Merging Branches in Git**

**Objective:** To demonstrate how to merge branches in Git to integrate changes from one branch into another while maintaining a cohesive codebase and version history

**Tools required:** Git and GitHub

**Prerequisites:** You must have Git installed to proceed with this demo.

Steps to be followed:

1. Create a new GitHub repository
2. Clone the GitHub repository
3. List all the branches in your repository
4. Create and switch to the new branch
5. Create a file and commit the changes
6. Check the status of the new branch
7. Switch back to the main branch
8. Merge the branches

**Step 1: Create a new GitHub repository**

1. Click on the **New** button to create a new repository

A screenshot of a computer

Description automatically generated

1. Enter a repository name and click on the **Create repository** button

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Step 2: Clone the GitHub repository**

1. Open the created repository in GitHub and click on the **Code** button

**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 on your lab and use the following command to clone the repository:

**git clone <URL>**

****

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

**Step 3: List all the branches in your repository**

1. Navigate to the cloned repository using the following command:

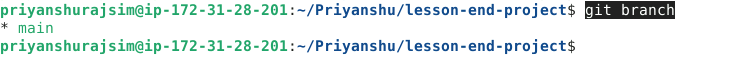
**cd lesson-end-project**

**A close-up of blue text

Description automatically generated**

1. Run the following command to display all repository branches:

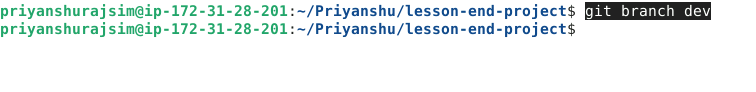
**git branch**

****

**Step 4: Create and switch to the new branch**

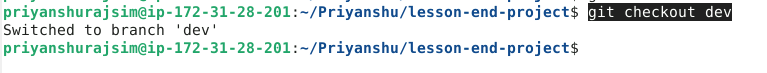
1. Run the following command to create a new branch in your repository:

**git branch dev**

****

1. Use the following command to switch to the newly created branch:

**git checkout dev**

****

**Step 5: Create a file and commit the changes**

* 1. Execute the given command to create a file:

**vi index.html**

****

* 1. Add the given code snippet into the **index.html** file:

**<html>**

**<body>**

**<p> This is a Test HTML file. </p>**

**</body>**

**</html>**

**A close-up of a computer screen

Description automatically generated**

**Note**: Press **i** to edit the files. Press the **Esc** button to exit insert mode and enter **:wq** to save the file

* 1. Use the following command to add the file to the **dev** branch:

**git add .**

**A white background with blue text

Description automatically generated**

* 1. Use the following command to commit the changes:

**git commit -m "Added Index.html"**

**A white background with blue text

Description automatically generated**

**Step 6: Check the status of the new branch**

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

**git status**

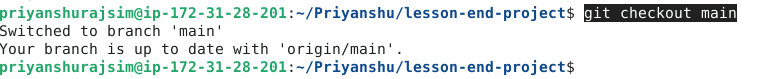
**A close up of a name

Description automatically generated**

**Step 7: Switch back to the main branch**

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

**git checkout main**

****

**Step 8: Merge the branches**

* 1. Use the following command to merge the **dev** branch to the main branch:

**git merge dev**

A white background with blue text

Description automatically generated

* 1. Push the changes to the remote repository using the following command:

**git push origin main**

**A screen shot of a computer

Description automatically generated**

By following these steps, you have successfully demonstrated merging branches in Git to integrate changes from one branch into another, while ensuring a cohesive codebase and version history.