**Assessment 3 :write a step to create a new branch and merge with master branch**

Sure! Here are the steps to create a new branch and then merge it with the master branch in Git:

Step-by-Step Guide

1. \*\*Navigate to Your Project Directory\*\*

Open your terminal or command prompt and navigate to your project's directory.

**cd /path/to/your/project**

1. \*\*Ensure You Are on the Master Branch\*\*

Switch to the master branch to ensure you are starting from the correct base.

**git checkout master**

1. \*\*Pull the Latest Changes\*\*

Ensure your master branch is up-to-date with the remote repository.

**git pull origin master**

1. \*\*Create a New Branch\*\*

Create a new branch from the master branch.

**git checkout -b new-feature-branch**

1. \*\*Make Your Changes\*\*

Make the necessary changes in your project files.

1. \*\*Stage and Commit Your Changes\*\*

Stage your changes and commit them to the new branch.

**git add .**

**git commit -m "Add new feature"**

1. \*\*Push the New Branch to Remote\*\* Push the new branch to the remote repository.

**git push origin new-feature-branch**

1. \*\*Switch Back to the Master Branch\*\* Switch back to the master branch.

**git checkout master**

1. \*\*Pull the Latest Changes on the Master Branch\*\* Ensure your master branch is up-to-date.  **git pull origin master**
2. \*\*Merge the New Branch into Master\*\* Merge your new branch into the master branch.

**git merge new-feature-branch**

1. \*\*Resolve Any Merge Conflicts (if any)\*\*

If there are merge conflicts, Git will indicate which files are conflicting. Open these files, resolve the conflicts, and then stage the resolved files.  **git add <file-with-conflict>**

1. \*\*Commit the Merge (if necessary)\*\* If conflicts were resolved, commit the merge.  **git commit -m "Resolve merge conflicts"**
2. \*\*Push the Updated Master Branch to Remote\*\*

Push the merged changes to the remote master branch.

**git push origin master**

Summary of Commands

**cd /path/to/your/project git checkout master git pull origin master git checkout -b new-feature-branch**

# Make your changes  **git add .**

**git commit -m "Add new feature" git push origin new-feature-branch git checkout master git pull origin master git merge new-feature-branch**

# Resolve conflicts if any  **git push origin master**

By following these steps, you will have successfully created a new branch, made changes, and merged it back into the master branch.