Lesson 04 Lesson-End Project

Creating a New Branch and Merging It into Git

Project Agenda: To establish a new branch in Git and subsequently merge it

Description: You are employed at an IT company. Your company is undertaking a project that comprises three modules, and you have been assigned one of these modules. You have been instructed to upload all the project files to the GitHub repository. To avoid impacting the main code base, you can create a new branch and carry out your work there. Once your module is complete, you can merge it into the master branch.

Tools required: Git and GitHub

Prerequisites: You must have Git installed in the lab to proceed. If you do not have it, please refer to Lesson 4, Demo 1, for instructions on how to install and set up Git.

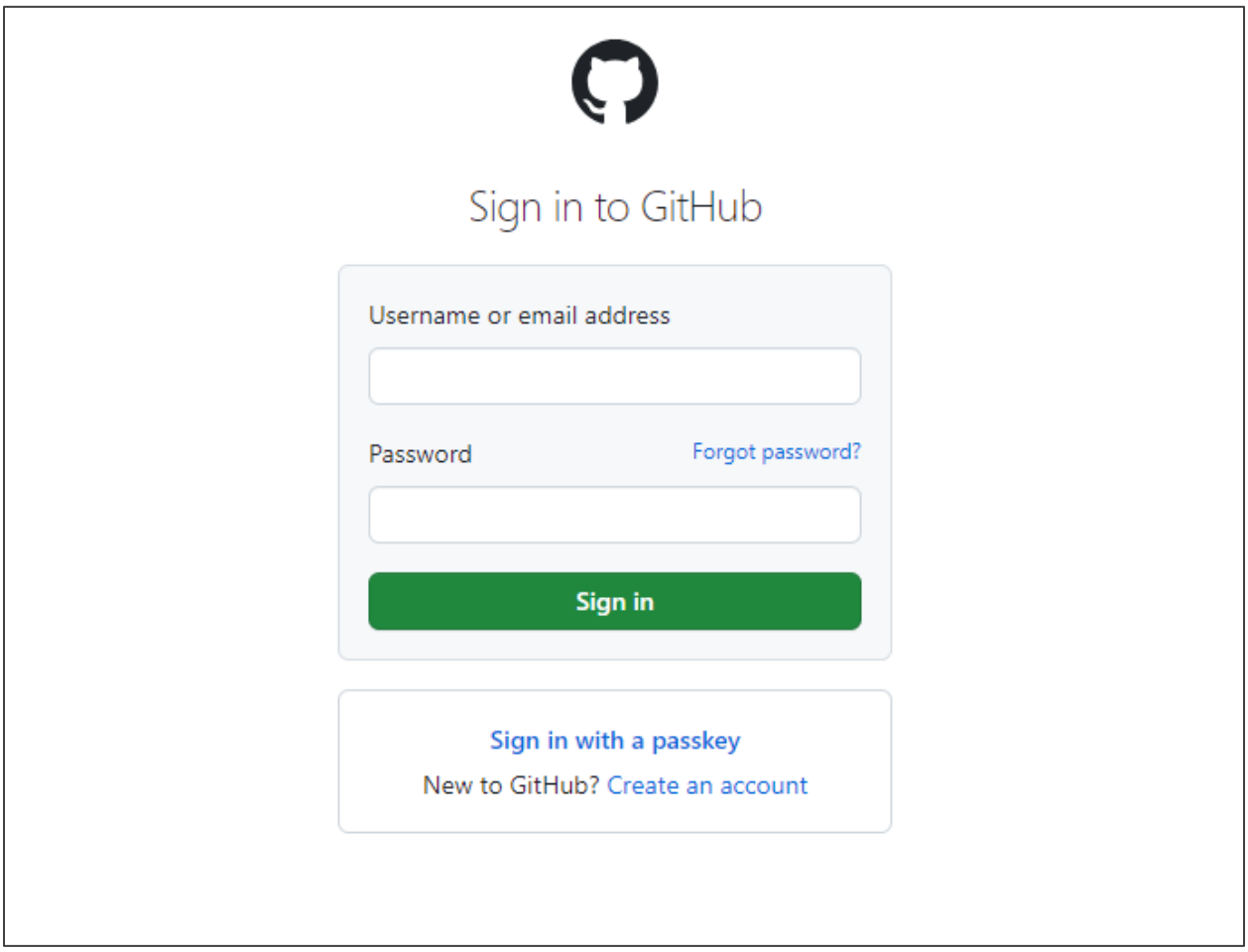
Expected Deliverables: Create a GitHub repository, clone the GitHub repository, create a new branch, switch between the branches, and merge the branch

Steps to be followed:

1. Create a new GitHub repository
2. Clone the repository
3. Create a new branch and verify its creation
4. Rename an existing branch and list all the branches in your repository
5. Create a new branch and switch to the new branch
6. Create a file and commit the changes
7. Check the status of the new branch
8. Delete the branch and verify its deletion
9. Switch back to the main branch
10. Merge the branches
11. Push the changes to GitHub
12. Pull the changes from GitHub

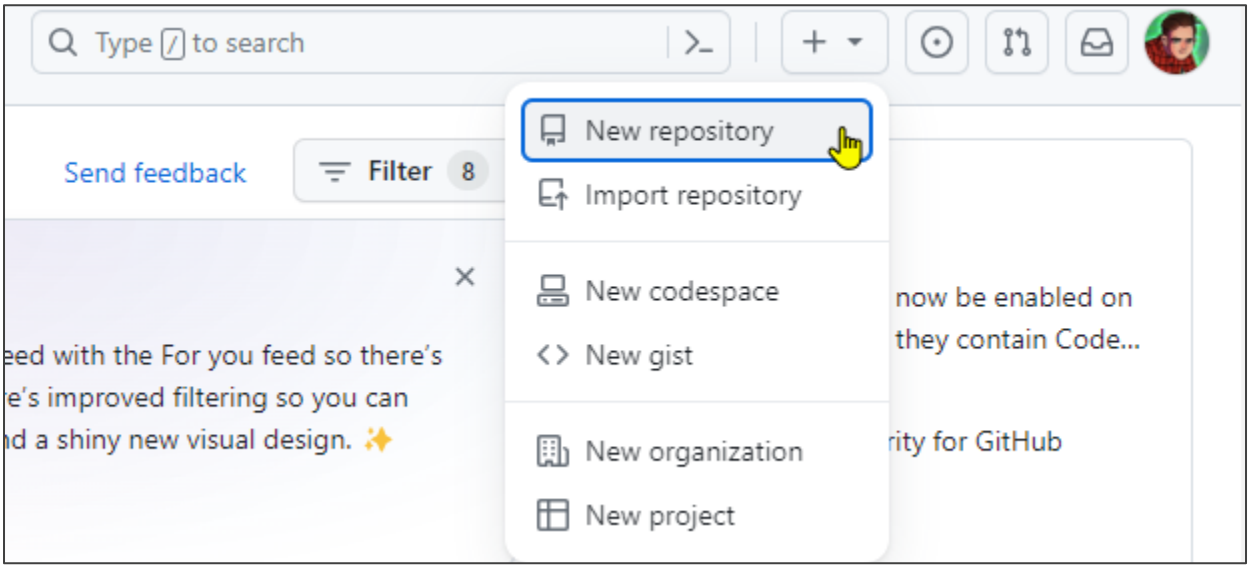
Step 1: Create a new GitHub repository

1.1 Open the browser in your lab, go to **github.com**, and log in to your account

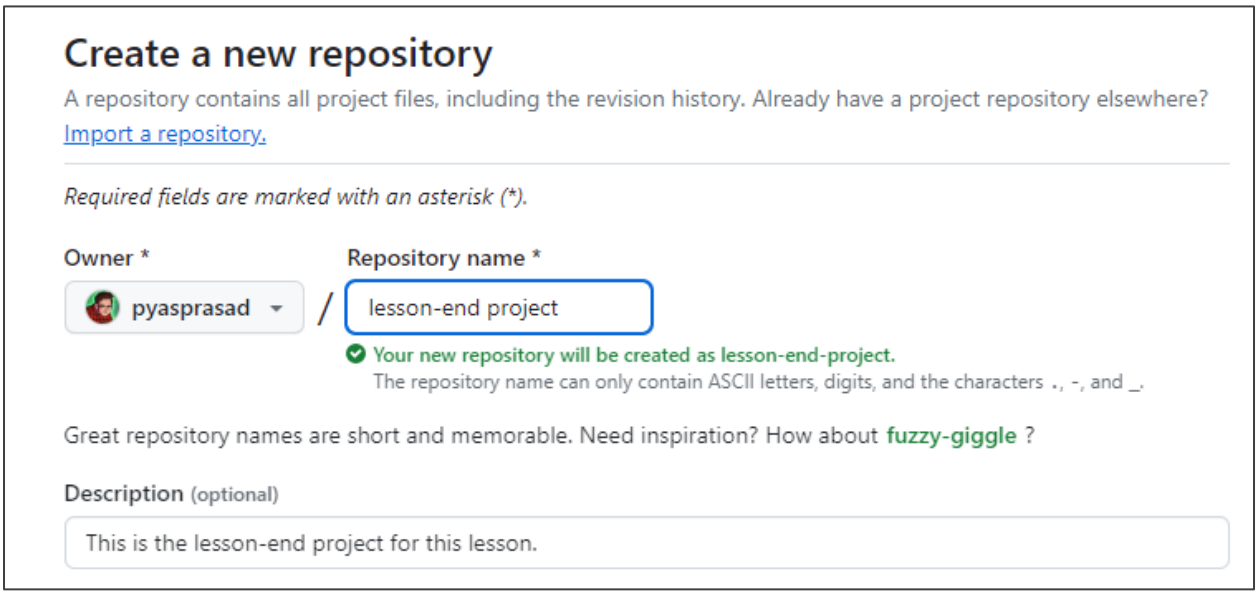


Note: If you do not have a GitHub account, visit the official website at **https://github.com/signup** and create a new account.

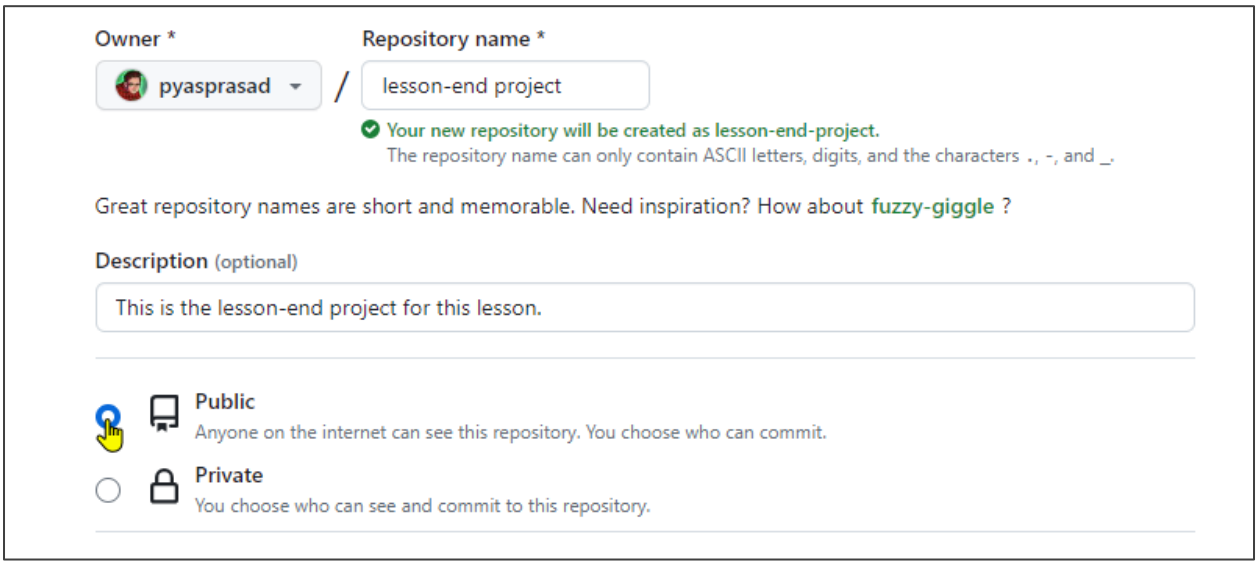
1.2 Click on the **+** icon from the upper-right corner of the page and select **New repository** from the drop-down menu



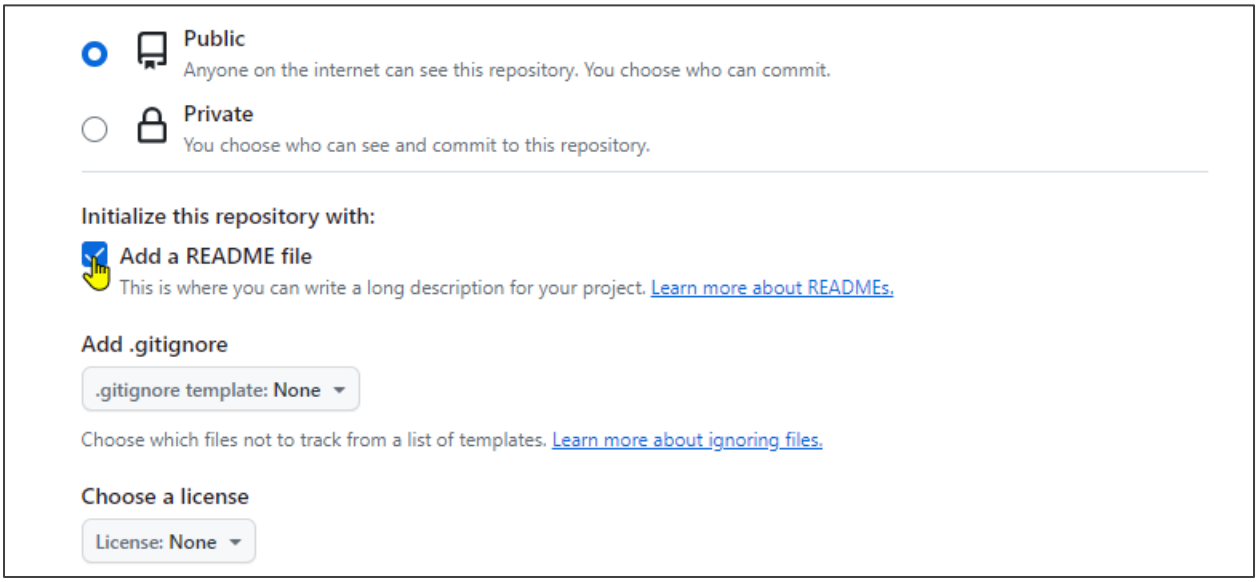
1.3 Enter an arbitrary name for the repository; however, the **Description** is optional



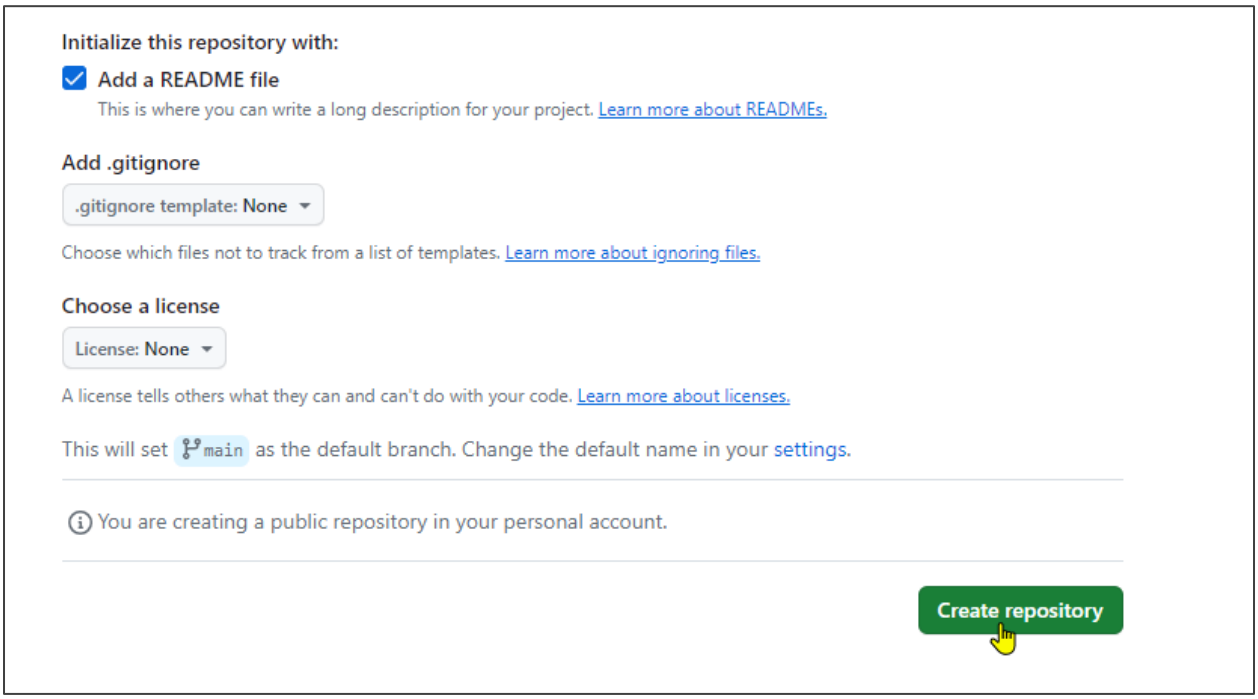
1.4 Choose **Public** for the repository type

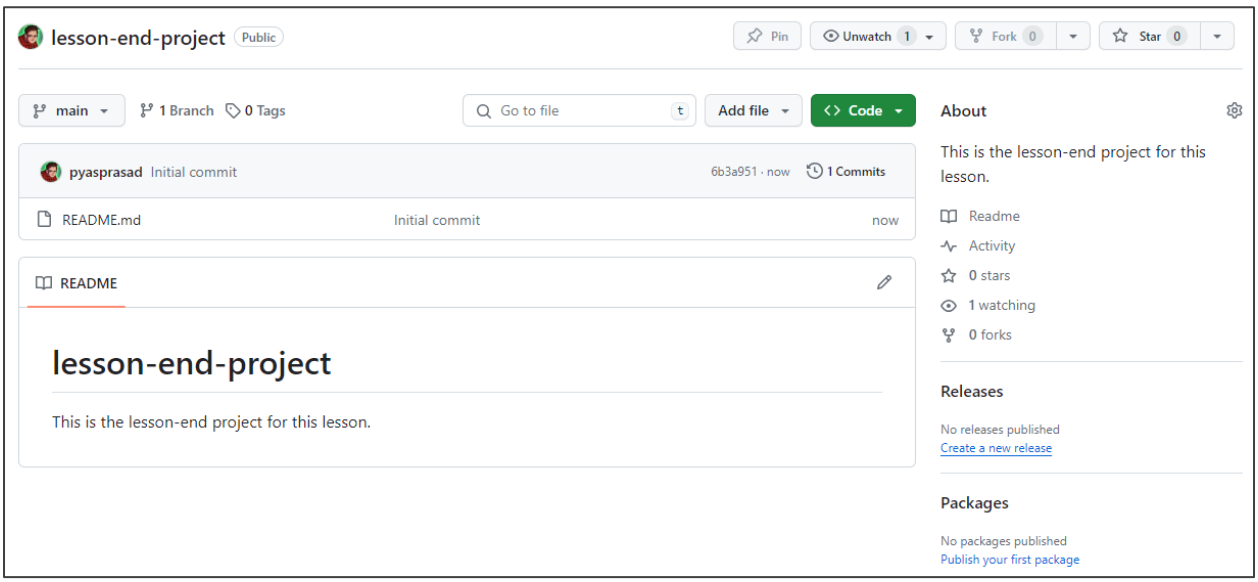


1.5 Select **Add a README file** to initialize the repository with a README file



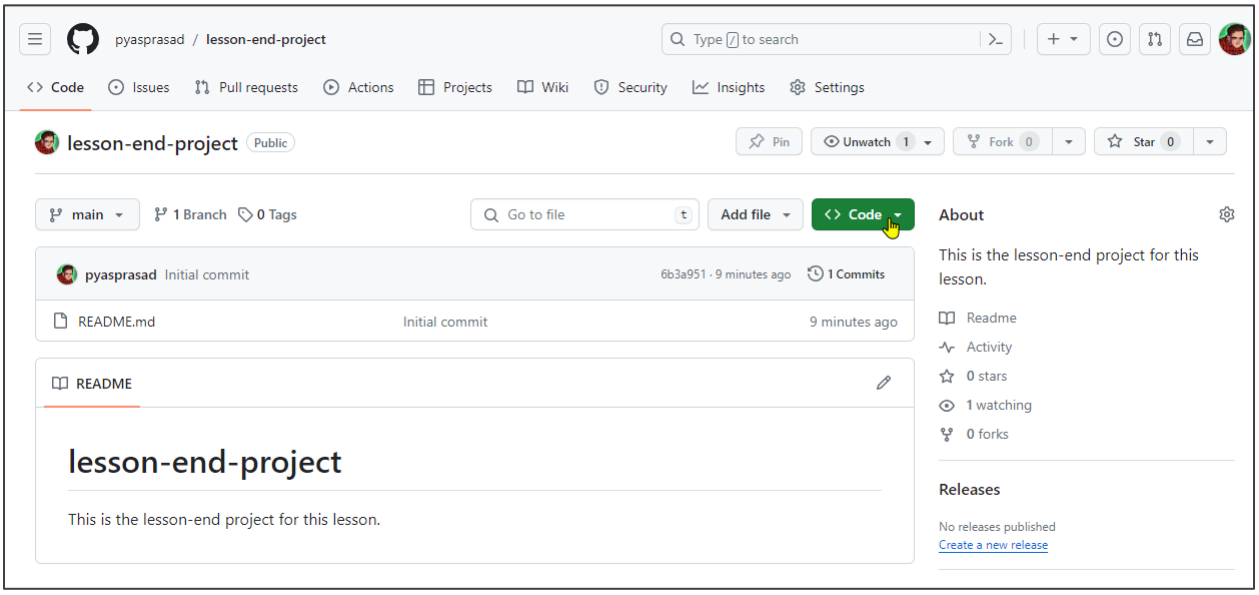
1.6 Click on the **Create repository** button



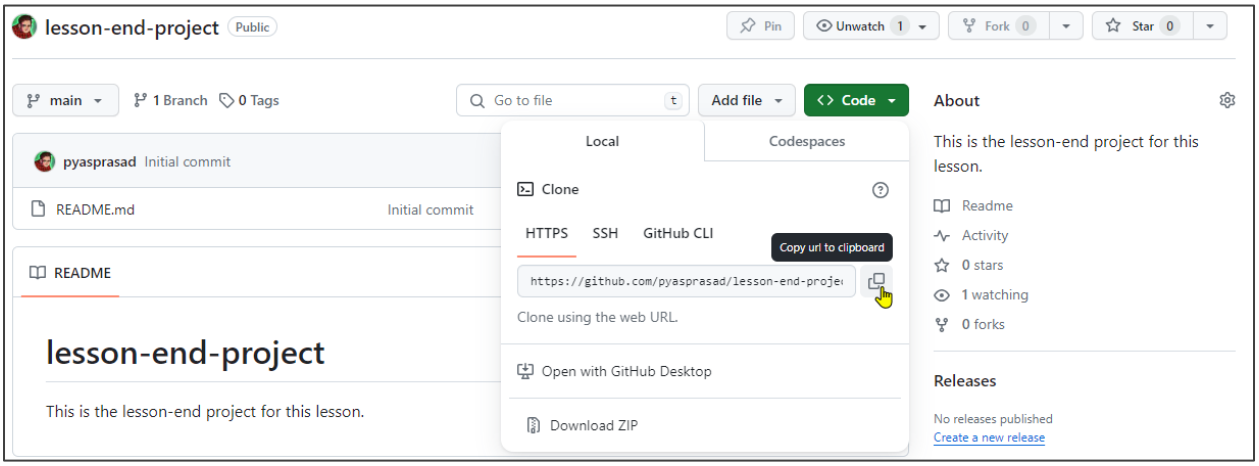


Step 2: Clone the GitHub repository

2.1 Open the **lesson-end-project** repo and click on the **Code** button



2.2 Click on the icon as shown in the following image to copy the **URL** provided under **HTTPS**



2.3 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 repository

```
labsuser@ip-172-31-40-118:~$ git clone https://github.com/pyasprasad/lesson-end-project.git
Cloning into 'lesson-end-project'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
labsuser@ip-172-31-40-118:~$
```

Step 3: Create a new branch and verify its creation

3.1 Navigate to the lesson-end-project using the following command:
cd lesson-end-project

```
labsuser@ip-172-31-40-118:~$ cd lesson-end-project
labsuser@ip-172-31-40-118:~/lesson-end-project$
```

3.2 Use the following command to create a new branch in your repository:
git branch project_branch

```
labsuser@ip-172-31-40-118:~/lesson-end-project$ git branch project_branch
labsuser@ip-172-31-40-118:~/lesson-end-project$
```

3.3 Verify the creation of the new branch using the following command:
git branch

```
labsuser@ip-172-31-40-118:~/lesson-end-project$ git branch
* main
  project_branch
labsuser@ip-172-31-40-118:~/lesson-end-project$
```

Step 4: Rename an existing branch and list all the branches in your repository

4.1 Use the following command to rename the branch:
git branch -m project_branch1

```
labsuser@ip-172-31-40-118:~/lesson-end-project$ git branch -m project_branch1
labsuser@ip-172-31-40-118:~/lesson-end-project$
```

4.2 Use the following command to list the branches to verify the new name of the branch:
git branch

```
labsuser@ip-172-31-40-118:~/lesson-end-project$ git branch
  project_branch
* project_branch1
labsuser@ip-172-31-40-118:~/lesson-end-project$
```

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

5.1 Execute the following command to create a new branch:
git branch project_branch2

```
labsuser@ip-172-31-40-118:~/lesson-end-project$ git branch project_branch2
labsuser@ip-172-31-40-118:~/lesson-end-project$
```

5.2 Verify the creation of the new branch using the following command:
git branch

```
labsuser@ip-172-31-40-118:~/lesson-end-project$ git branch
  project_branch
* project_branch1
  project_branch2
labsuser@ip-172-31-40-118:~/lesson-end-project$
```

5.3 Use the following command to switch to the newly created **project_branch2**:
git branch -m project_branch1

```
labsuser@ip-172-31-40-118:~/lesson-end-project$ git checkout project_branch2
Switched to branch 'project_branch2'
labsuser@ip-172-31-40-118:~/lesson-end-project$
```

Step 6: Create a file and commit the changes

6.1 Use the following command to create a file:
vi index.html

```
labsuser@ip-172-31-40-118:~/lesson-end-project$ vi index.html
labsuser@ip-172-31-40-118:~/lesson-end-project$
```

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

```
<html>
  <body>
    <p> This is a lesson end project. </p>
  </body>
</html>
```

```
<html>
   <body>
      <p> This is a lesson end project. </p>
   </body>
</html>
```

6.3 Use the following command to add the file to the **project_branch2**:
git add index.html

```
labsuser@ip-172-31-40-118:~/lesson-end-project$ git add index.html
labsuser@ip-172-31-40-118:~/lesson-end-project$
```

6.4 Use the following command to commit the changes:
git commit -a -m "file modified"

```
labsuser@ip-172-31-40-118:~/lesson-end-project$ git commit -a -m "file modified"
[project_branch2 04f6540] file modified
 1 file changed, 6 insertions(+)
 create mode 100644 index.html
labsuser@ip-172-31-40-118:~/lesson-end-project$
```

Step 7: Check the status of the new branch

7.1 Check the status of the new branch using the following command:
git status

```
labsuser@ip-172-31-40-118:~/lesson-end-project$ git status
On branch project_branch2
nothing to commit, working tree clean
labsuser@ip-172-31-40-118:~/lesson-end-project$
```

Step 8: Delete the branch and verify its deletion

8.1 Use the following command to delete the newly created branch:
git branch -d project_branch

```
labsuser@ip-172-31-40-118:~/lesson-end-project$ git branch -d project_branch
Deleted branch project_branch (was 6b3a951).
labsuser@ip-172-31-40-118:~/lesson-end-project$
```

8.2 Verify the deletion of the branch using the following command:
git branch

```
labsuser@ip-172-31-40-118:~/lesson-end-project$ git branch
  project_branch1
* project_branch2
labsuser@ip-172-31-40-118:~/lesson-end-project$
```

Step 9: Switch back to the main branch

9.1 Use the following command to switch back to the main branch:
git checkout project_branch1

```
labsuser@ip-172-31-40-118:~/lesson-end-project$ git checkout project_branch1
Switched to branch 'project_branch1'
Your branch is up to date with 'origin/main'.
labsuser@ip-172-31-40-118:~/lesson-end-project$
```

Step 10: Merge the branches

10.1 Use the following command to merge the test branch to the main branch:
git merge project_branch2

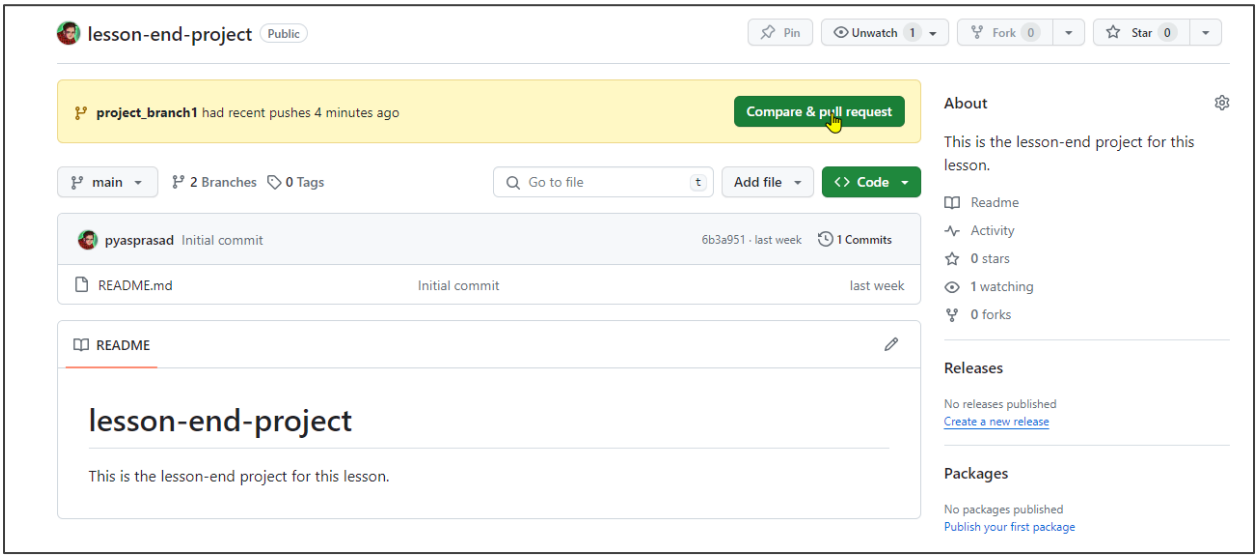
```
labsuser@ip-172-31-40-118:~/lesson-end-project$ git merge project_branch2
Updating 6b3a951..04f6540
Fast-forward
 index.html | 6 ++++++
 1 file changed, 6 insertions(+)
 create mode 100644 index.html
labsuser@ip-172-31-40-118:~/lesson-end-project$
```

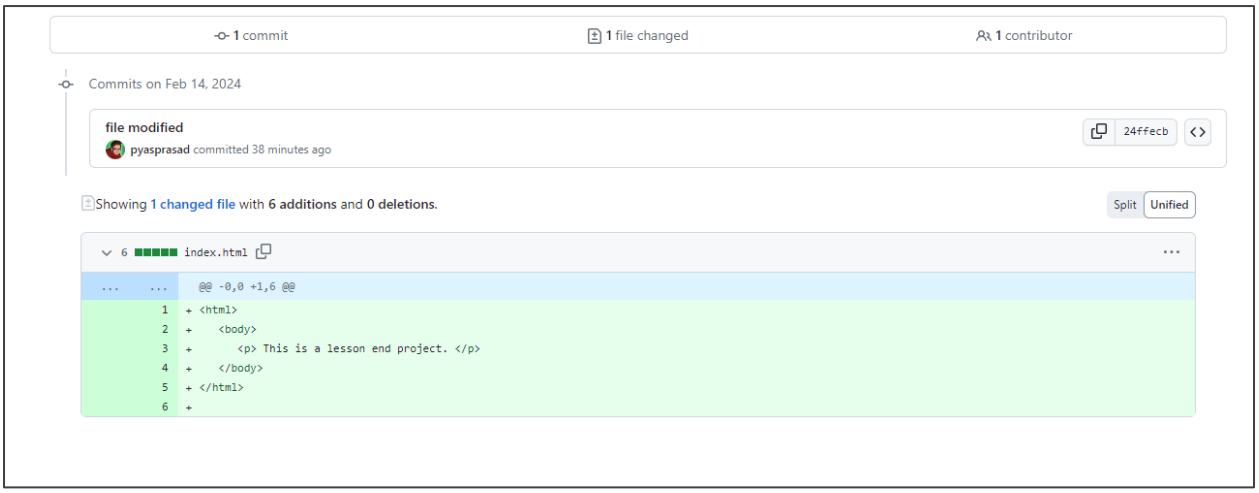
Step 11: Push the changes to GitHub

11.1 Use the following command to push the changes from the local to the remote repository:
git push -u origin HEAD

```
labsuser@ip-172-31-44-226:~/lesson-end-project$ git push -u origin HEAD
Username for 'https://github.com': pyasprasad
Password for 'https://pyasprasad@github.com':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 2 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 350 bytes | 350.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'project_branch1' on GitHub by visiting:
remote:      https://github.com/pyasprasad/lesson-end-project/pull/new/project_branch1
remote:
To https://github.com/pyasprasad/lesson-end-project
 * [new branch]      HEAD -> project_branch1
Branch 'project_branch1' set up to track remote branch 'project_branch1' from 'origin'.
labsuser@ip-172-31-44-226:~/lesson-end-project$
```

11.2 Click on the **Compare & pull request** button on the lesson-end-project repo page to compare and create a pull request





Step 12: Pull the changes from GitHub

12.1 Use the following command to pull changes from the GitHub

git pull

```
labsuser@ip-172-31-44-226:~/lesson-end-project$ git pull
Already up to date.
labsuser@ip-172-31-44-226:~/lesson-end-project$
```

By following these steps, you have successfully created a new repository on GitHub, cloned the repository to your local machine, and performed various operations such as creating, renaming, and switching between branches. You have also made changes to files, committed those changes, and merged branches.