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

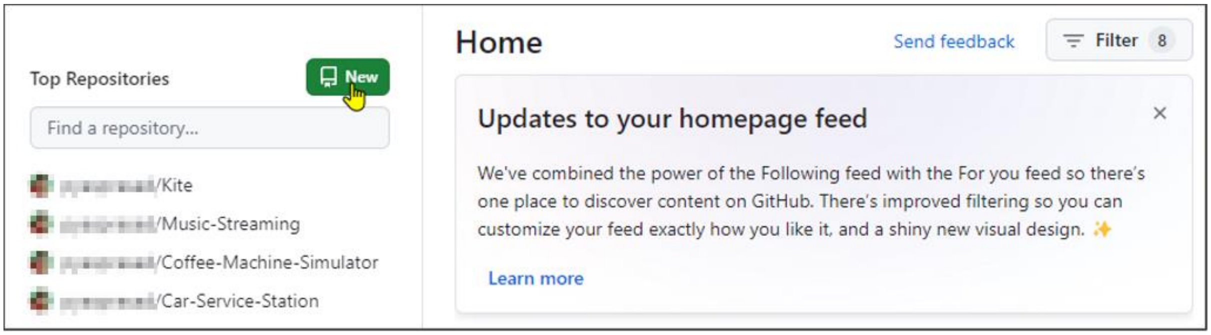
Prerequisites: You must have Git installed to proceed with this demo.

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

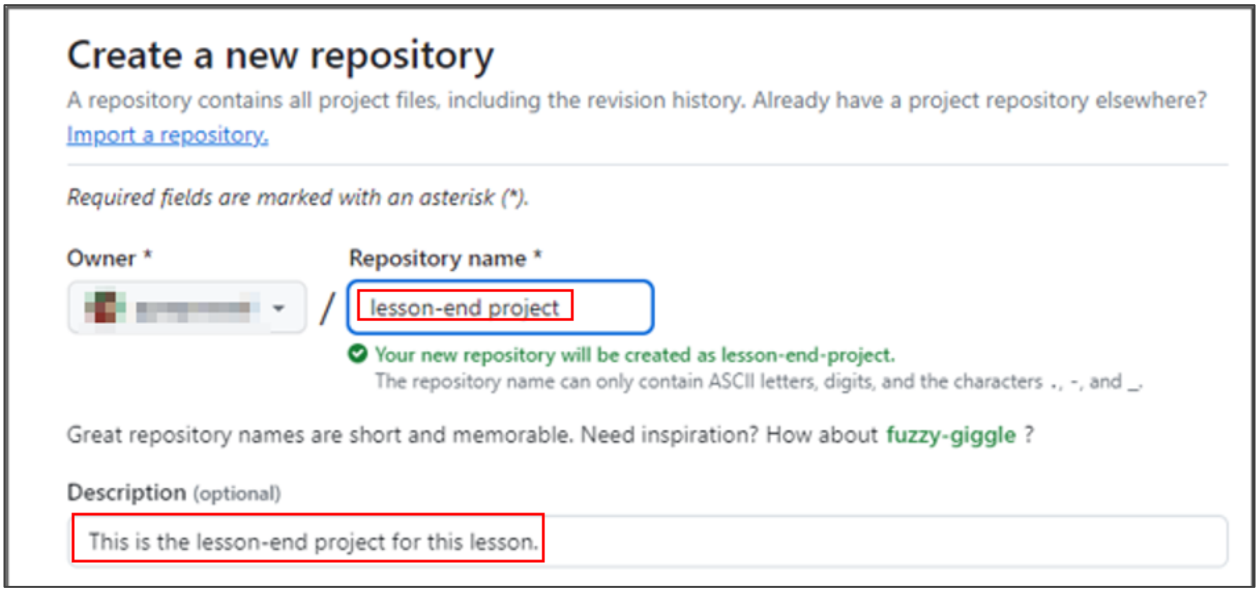
Step 1: Create a new GitHub repository

1.1 Click on the **New** button to create a new repository


Top Repositories
New
Find a repository...
[redacted]/Kite
[redacted]/Music-Streaming
[redacted]/Coffee-Machine-Simulator
[redacted]/Car-Service-Station
Home
Send feedback
Filter 8
Updates to your homepage feed
×
Learn more

We've combined the power of the Following feed with the For you feed so there's one place to discover content on GitHub. There's improved filtering so you can customize your feed exactly how you like it, and a shiny new visual design. ✨

1.2 Enter a repository name and click on the **Create repository** button



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Required fields are marked with an asterisk (*).

Owner * [redacted] / Repository name * lesson-end project

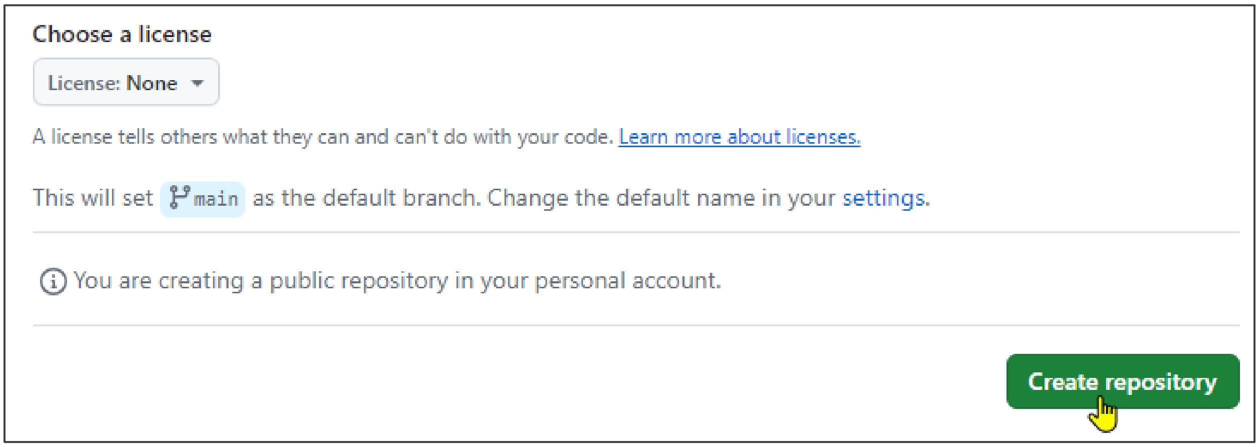
Your new repository will be created as lesson-end-project.

The repository name can only contain ASCII letters, digits, and the characters ., -, and _.

Great repository names are short and memorable. Need inspiration? How about **fuzzy-giggle** ?

Description (optional)

This is the lesson-end project for this lesson.



Choose a license

License: None

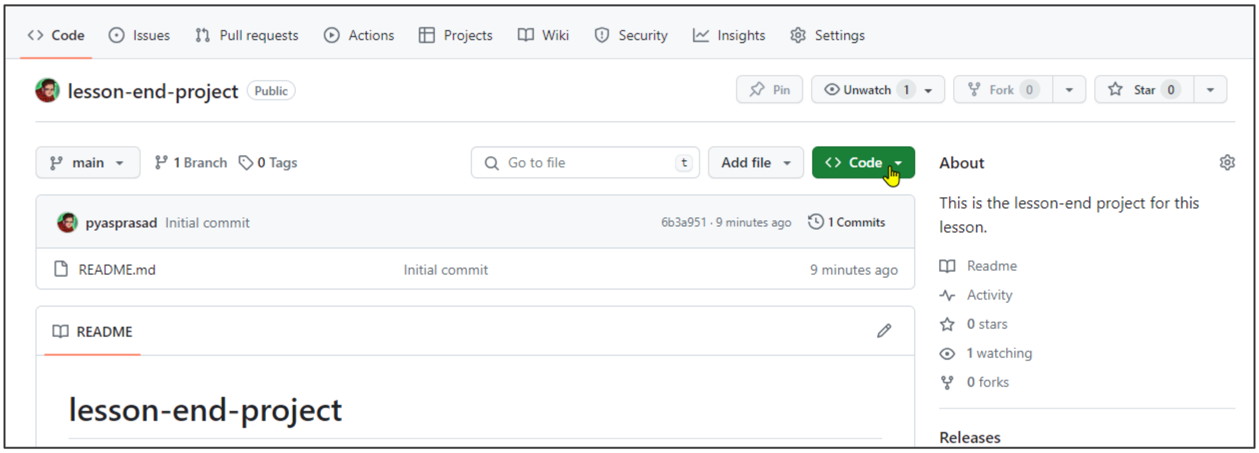
A license tells others what they can and can't do with your code. Learn more about licenses.

This will set `main` as the default branch. Change the default name in your settings.

You are creating a public repository in your personal account.

Step 2: Clone the GitHub repository

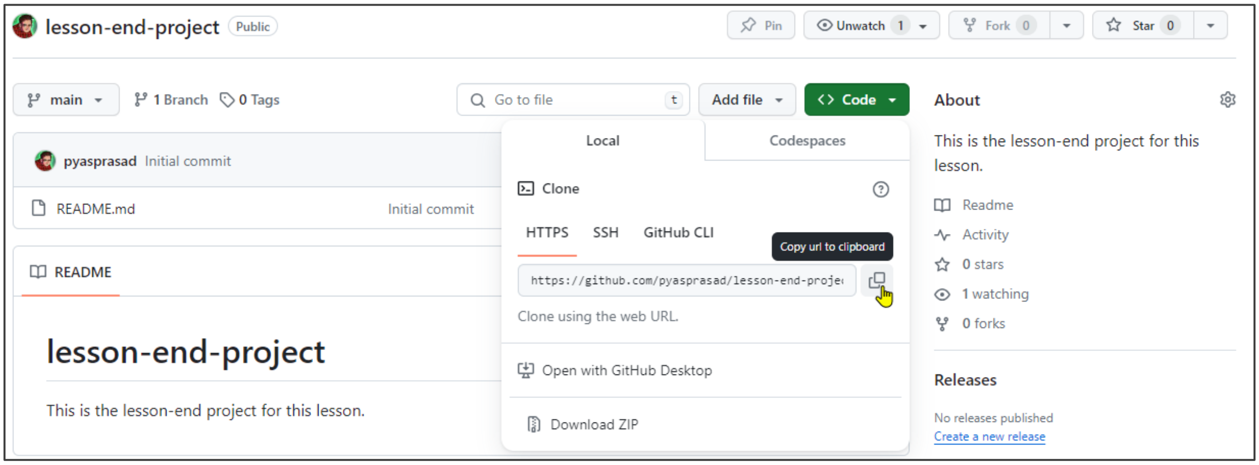
2.1 Open the created repository in GitHub and click on the **Code** button



lesson-end-project

This is the lesson-end project for this lesson.

2.2 Click on the copy icon to copy the **HTTPS URL,** as shown below:



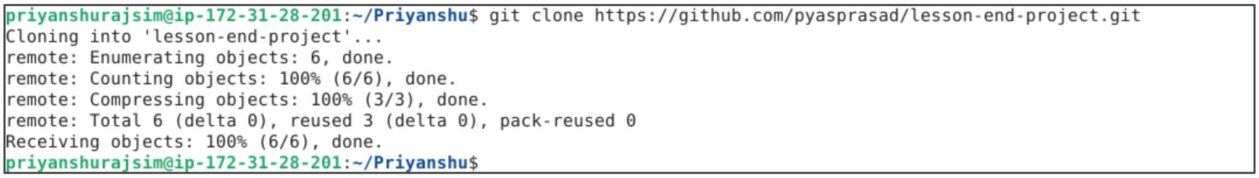
https://github.com/pyasprasad/lesson-end-proje

Clone using the web URL.

lesson-end-project

This is the lesson-end project for this lesson.

2.3 Open the terminal tab on your lab and use the following command to clone the repository:
git clone <URL>

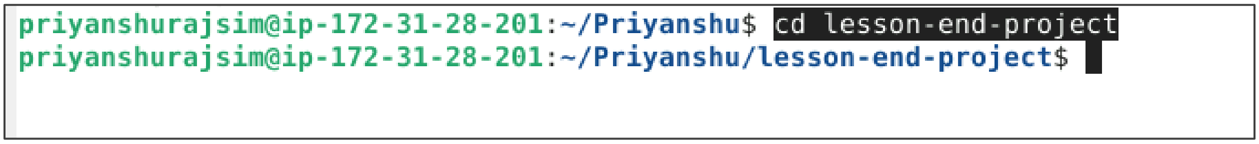


```
priyanshurajsim@ip-172-31-28-201:~/Priyanshu$ git clone https://github.com/pyasprasad/lesson-end-project.git
Cloning into 'lesson-end-project'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 6 (delta 0), reused 3 (delta 0), pack-reused 0
Receiving objects: 100% (6/6), done.
priyanshurajsim@ip-172-31-28-201:~/Priyanshu$
```

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

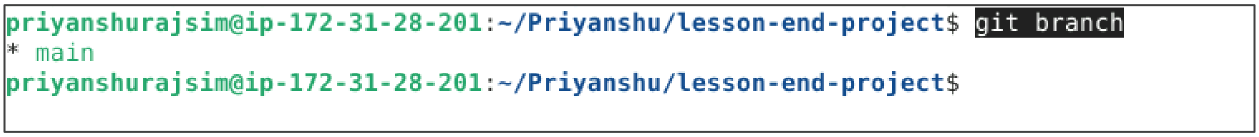
Step 3: List all the branches in your repository

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



```
priyanshurajsim@ip-172-31-28-201:~/Priyanshu$ cd lesson-end-project
priyanshurajsim@ip-172-31-28-201:~/Priyanshu/lesson-end-project$
```

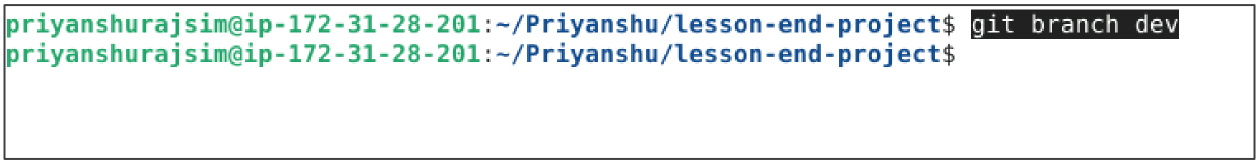
3.2 Run the following command to display all repository branches:
git branch



```
priyanshurajsim@ip-172-31-28-201:~/Priyanshu/lesson-end-project$ git branch
* main
priyanshurajsim@ip-172-31-28-201:~/Priyanshu/lesson-end-project$
```

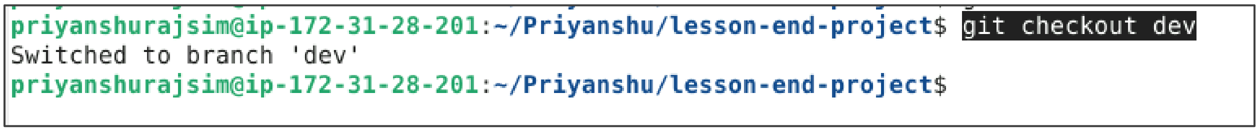
Step 4: Create and switch to the new branch

4.1 Run the following command to create a new branch in your repository:
git branch dev



```
priyanshurajsim@ip-172-31-28-201:~/Priyanshu/lesson-end-project$ git branch dev
priyanshurajsim@ip-172-31-28-201:~/Priyanshu/lesson-end-project$
```

4.2 Use the following command to switch to the newly created branch:
git checkout dev



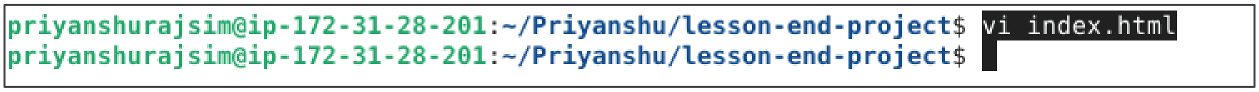
```
priyanshurajsim@ip-172-31-28-201:~/Priyanshu/lesson-end-project$ git checkout dev
Switched to branch 'dev'
priyanshurajsim@ip-172-31-28-201:~/Priyanshu/lesson-end-project$
```

.

Step 5: Create a file and commit the changes

5.1 Execute the given command to create a file:

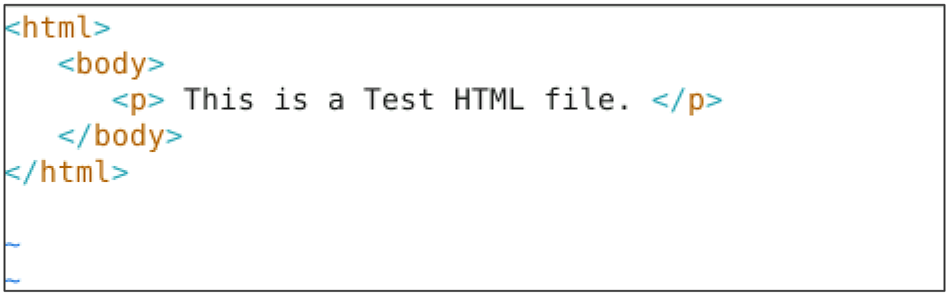
vi index.html



```
priyanshurajsim@ip-172-31-28-201:~/Priyanshu/lesson-end-project$ vi index.html
priyanshurajsim@ip-172-31-28-201:~/Priyanshu/lesson-end-project$ 
```

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

```
<html>
  <body>
    <p> This is a Test HTML file. </p>
  </body>
</html>
```



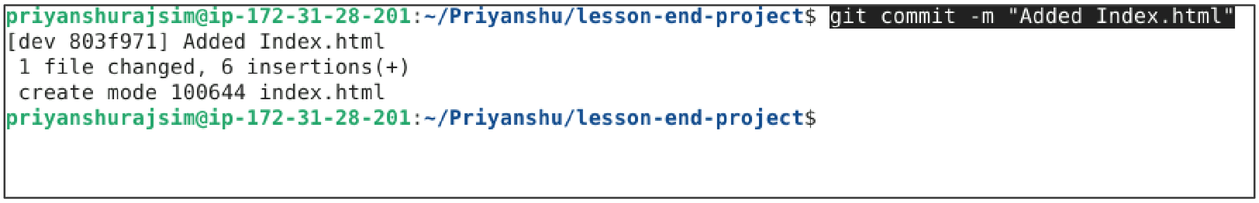
```
<html>
   <body>
      <p> This is a Test HTML file. </p>
   </body>
</html>

~
~
```

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

5.3 Use the following command to add the file to the **dev** branch:

git add .

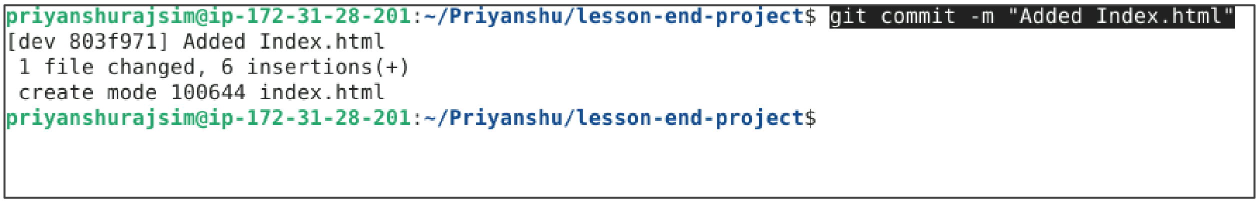


```
priyanshurajsim@ip-172-31-28-201:~/Priyanshu/lesson-end-project$ git commit -m "Added Index.html"
[dev 803f971] Added Index.html
 1 file changed, 6 insertions(+)
 create mode 100644 index.html
priyanshurajsim@ip-172-31-28-201:~/Priyanshu/lesson-end-project$
```

.

5.4 Use the following command to commit the changes:

git commit -m "Added Index.html"

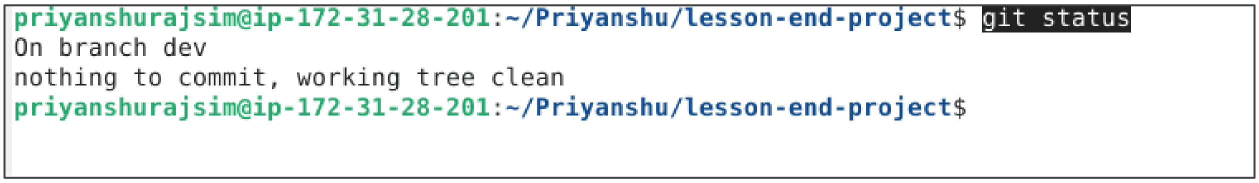


```
priyanshurajsim@ip-172-31-28-201:~/Priyanshu/lesson-end-project$ git commit -m "Added Index.html"
[dev 803f971] Added Index.html
 1 file changed, 6 insertions(+)
 create mode 100644 index.html
priyanshurajsim@ip-172-31-28-201:~/Priyanshu/lesson-end-project$
```

Step 6: Check the status of the new branch

6.1 Check the status of the new branch using the following command:

git status

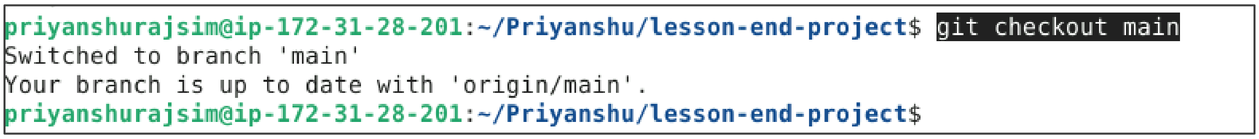


```
priyanshurajsim@ip-172-31-28-201:~/Priyanshu/lesson-end-project$ git status
On branch dev
nothing to commit, working tree clean
priyanshurajsim@ip-172-31-28-201:~/Priyanshu/lesson-end-project$
```

Step 7: Switch back to the main branch

7.1 Use the following command to switch back to the main branch:

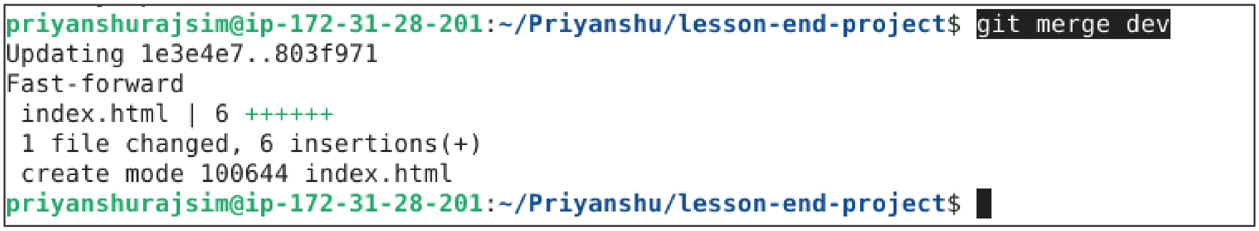
git checkout main



```
priyanshurajsim@ip-172-31-28-201:~/Priyanshu/lesson-end-project$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
priyanshurajsim@ip-172-31-28-201:~/Priyanshu/lesson-end-project$
```

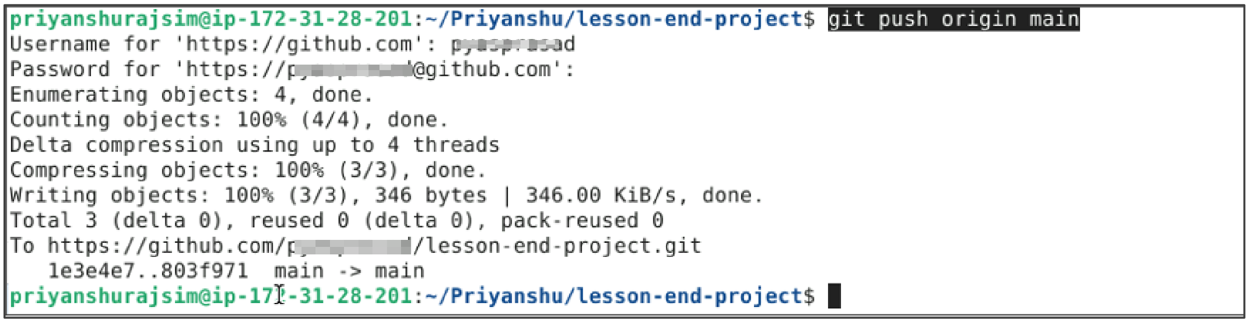
Step 8: Merge the branches

8.1 Use the following command to merge the **dev** branch to the main branch:
git merge dev



```
priyanshurajsim@ip-172-31-28-201:~/Priyanshu/lesson-end-project$ git merge dev
Updating 1e3e4e7..803f971
Fast-forward
 index.html | 6 ++++++
 1 file changed, 6 insertions(+)
 create mode 100644 index.html
priyanshurajsim@ip-172-31-28-201:~/Priyanshu/lesson-end-project$ 
```

8.2 Push the changes to the remote repository using the following command:
git push origin main



```
priyanshurajsim@ip-172-31-28-201:~/Priyanshu/lesson-end-project$ git push origin main
Username for 'https://github.com': [redacted]
Password for 'https://[redacted]@github.com':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 346 bytes | 346.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/[redacted]/lesson-end-project.git
   1e3e4e7..803f971  main -> main
priyanshurajsim@ip-172-31-28-201:~/Priyanshu/lesson-end-project$ 
```

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.