Lesson 04 Demo 04

Creating a Branch in Git

Objective: To create branch in GitHub to work on the features affecting the main code
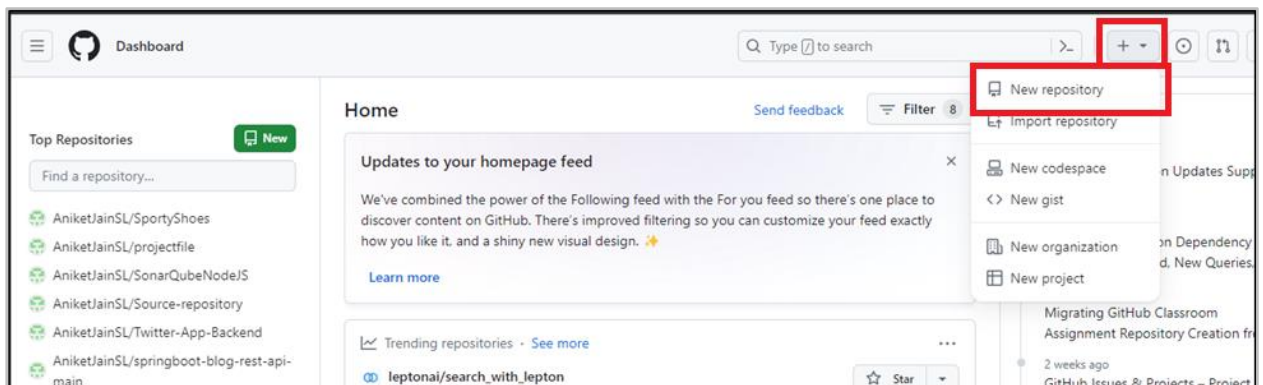
Tools required: Git and GitHub

Prerequisites: None

Steps to be followed:

1. Create a new GitHub repository
2. Clone the repository
3. List all the branches in your repository
4. Create a new branch and verify its creation
5. Rename an existing branch
6. Delete the branch and verify its deletion

Step 1: Create a new GitHub repository

1.1 In your lab's browser, visit **github.com** and log in. Click the **+** icon in the upper-right corner, select **New repository**, and create a new repository.

1.2 Enter an arbitrary name for the repository file as **'projectfile'**

Owner *

/ projectfile ✓

Repository name *

Great repository names are short and memorable. Need inspiration? How about **studious-adventure**?

Description (optional)

⦿ 📕 **Public**
Anyone on the internet can see this repository. You choose who can commit.

○ 🔒 **Private**
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☑ **Add a README file**
This is where you can write a long description for your project. Learn more.

☐ **Add .gitignore**
Choose which files not to track from a list of templates. Learn more.

Click on **Create Repository** to create a new repository

Initialize this repository with:
Skip this step if you're importing an existing repository.

☐ **Add a README file**
This is where you can write a long description for your project. Learn more.

☐ **Add .gitignore**
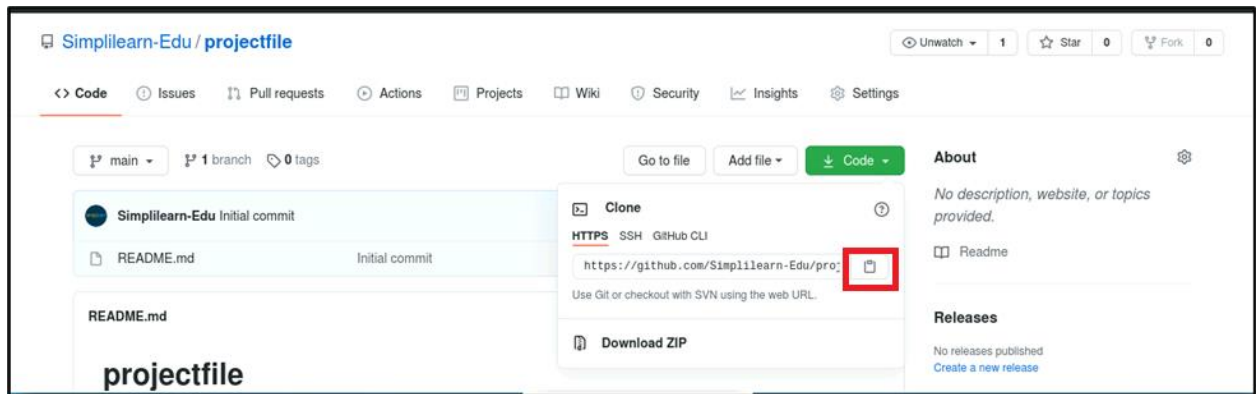Choose which files not to track from a list of templates. Learn more.

☐ **Choose a license**
A license tells others what they can and can't do with your code. Learn more.

Create repository

Step 2: Clone the repository

2.1 Open the **projectfile** repository, click on the **Code** button, and click on the copy icon as shown in the following screenshot to copy the **URL** provided under **HTTPS**

2.2 Open the terminal on your lab and use the following command to clone the repository:
git clone URL

> **Note:** Replace URL with the copied URL from the repository

Step 3: List all the branches in your repository

3.1 Open the terminal in your lab, and navigate to the **projectfile** folder by entering the following command:
cd projectfile

3.2 Use the following command to list all the branches in your repository:
git branch

```
████████████@ip-172-31-71-23:~$ cd projectfile
████████████@ip-172-31-71-23:~/projectfile$ git branch
* main
████████████@ip-172-31-71-23:~/projectfile$ ▌
```

Step 4: Create a new branch and verify its creation

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

```
████████████@ip-172-31-71-23:~/projectfile$  git branch new_branch
```

4.2 To verify the branch created, run the following command:
git branch

```
████████████@ip-172-31-71-23:~/projectfile$  git branch new_branch
████████████@ip-172-31-71-23:~/projectfile$  git branch
* main
  new_branch
████████████@ip-172-31-71-23:~/projectfile$ ▌
```

Step 5: Rename an existing branch

5.1 Use the following command to rename the new_branch:
git branch -m new_branch1

```
████████████@ip-172-31-71-23:~/projectfile$ git branch -m new_branch1
```

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

```
@ip-172-31-71-23:~/projectfile$ git branch -m new_branch1
@ip-172-31-71-23:~/projectfile$  git branch
  new_branch
* new_branch1
@ip-172-31-71-23:~/projectfile$
```

Step 6: Delete the branch and verify its deletion

6.1 Use the following command to delete the newly created branch:
   git branch -d new_branch

```
@ip-172-31-71-23:~/projectfile$ git branch -d new_branch
Deleted branch new_branch (was d77a2b5).
@ip-172-31-71-23:~/projectfile$
```

6.2 To verify whether the branch is deleted or not, run the following command:
   git branch

```
@ip-172-31-71-23:~/projectfile$ git branch
* new_branch1
@ip-172-31-71-23:~/projectfile$
```

By following these steps, you have successfully created a new branch in your GitHub repository, providing a segregated space to work on specific changes without affecting the main codebase.