Lesson-End Project

Implementing Basics of Version Control System with Git

Project agenda: To create a remote GitHub repository and use Git commands to perform various version control operations for managing the remote repository

Description: You work as a developer in an IT firm. Your company is undertaking a project that consists of three modules, and you have been asked to work on one of these modules. You have also been instructed to upload all the project files to the GitHub repository. To avoid impacting the main codebase, create a new branch and conduct your work there. Once your module is complete, merge it into the main branch.

Tools required: Git and GitHub

Prerequisites: You must have Git installed in the lab to proceed.

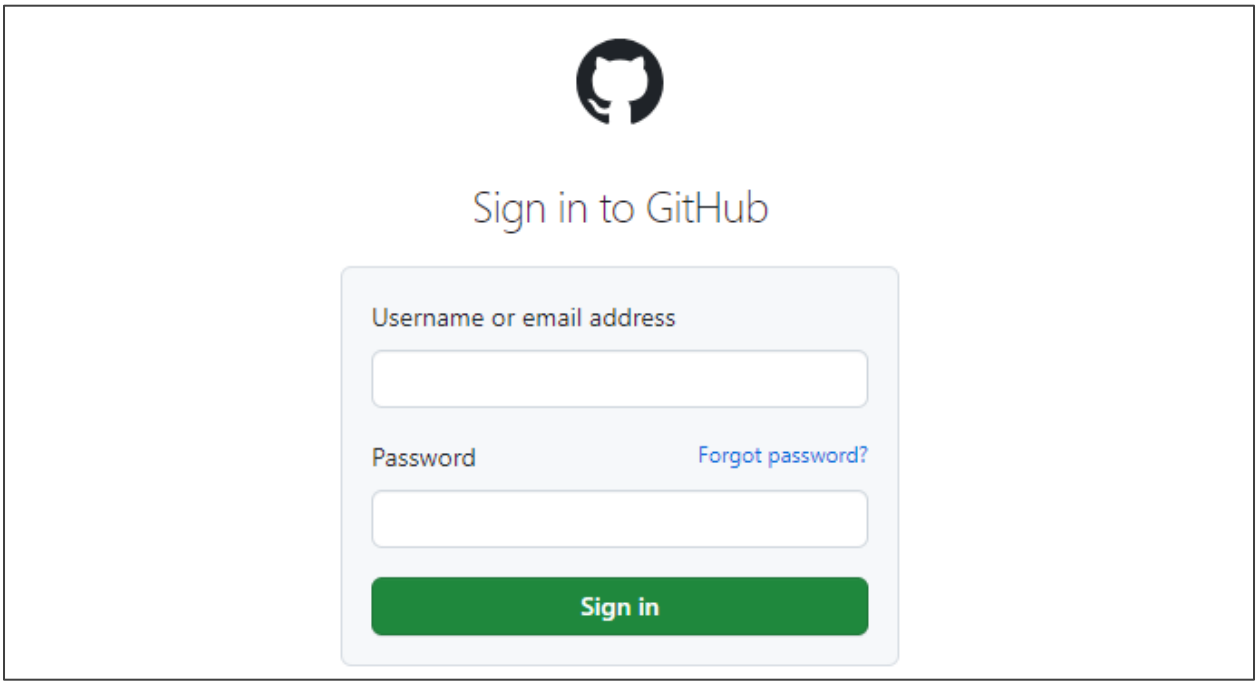
Expected deliverables: A step-by-step guide to creating, managing, and deleting branches and files in a Git repository, including scripts and commands to streamline version control for collaborative development

Steps to be followed:

1. Create a new repository
2. Clone the GitHub repository
3. Create a new branch and verify it
4. Rename the existing branch and list all the branches
5. Create a new branch and switch to the new branch
6. Create a file, commit the changes, and check the status of the new branch
7. Delete the branch and verify it
8. Switch back to the main branch and merge

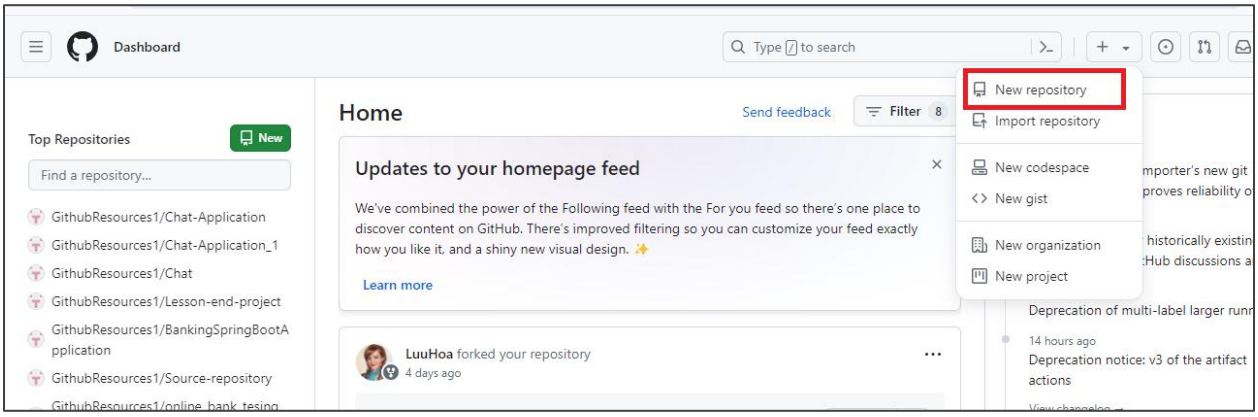
Step 1: Create a new repository

1.1 Open the browser and navigate to **https://github.com/** to log in to your GitHub account using your credentials

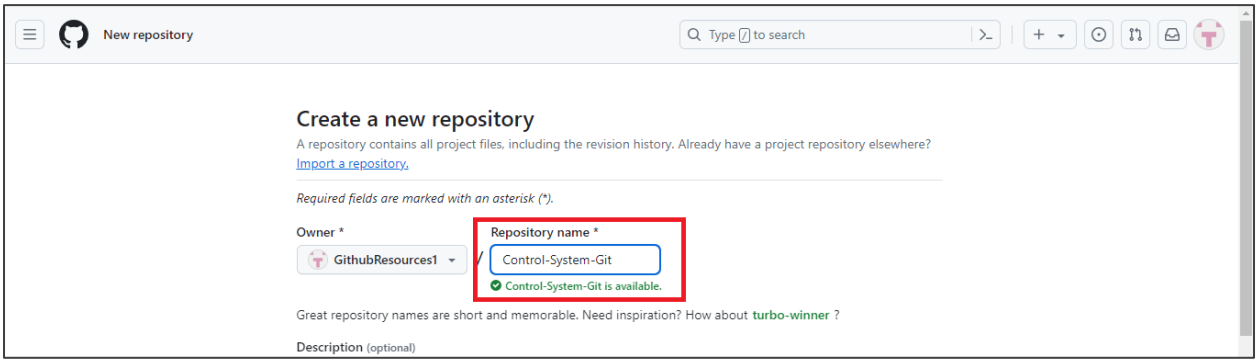


> **Note**: If you do not have a GitHub account, visit the official website at **https://github.com/signup** and create a new account

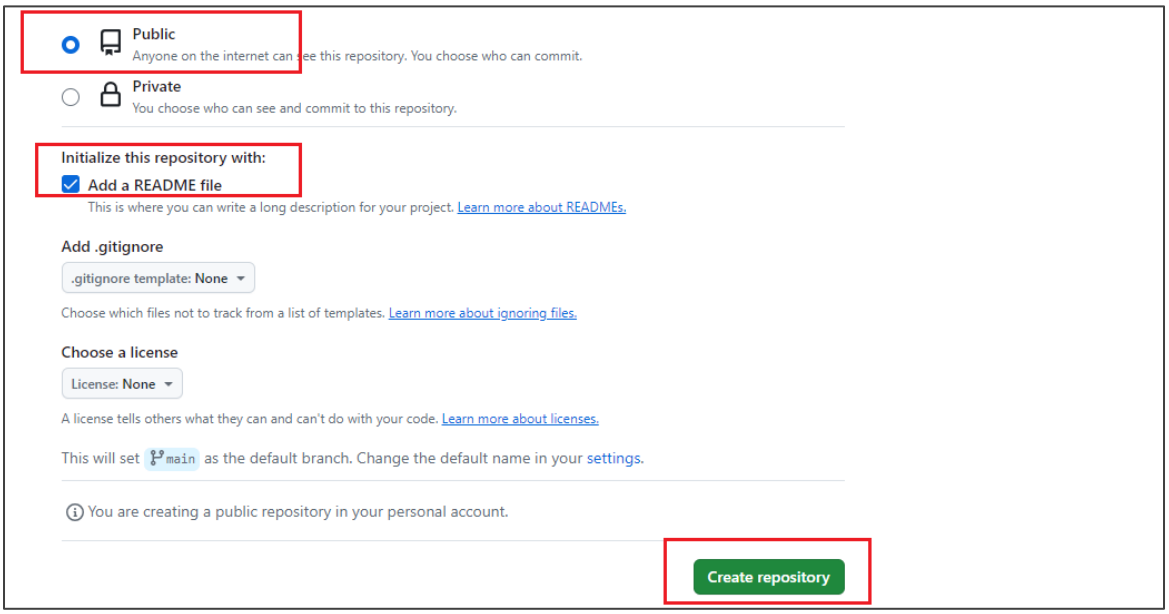
1.2 Click on the **+** icon in the upper-right corner of the page and select the **New repository** option from the drop-down menu

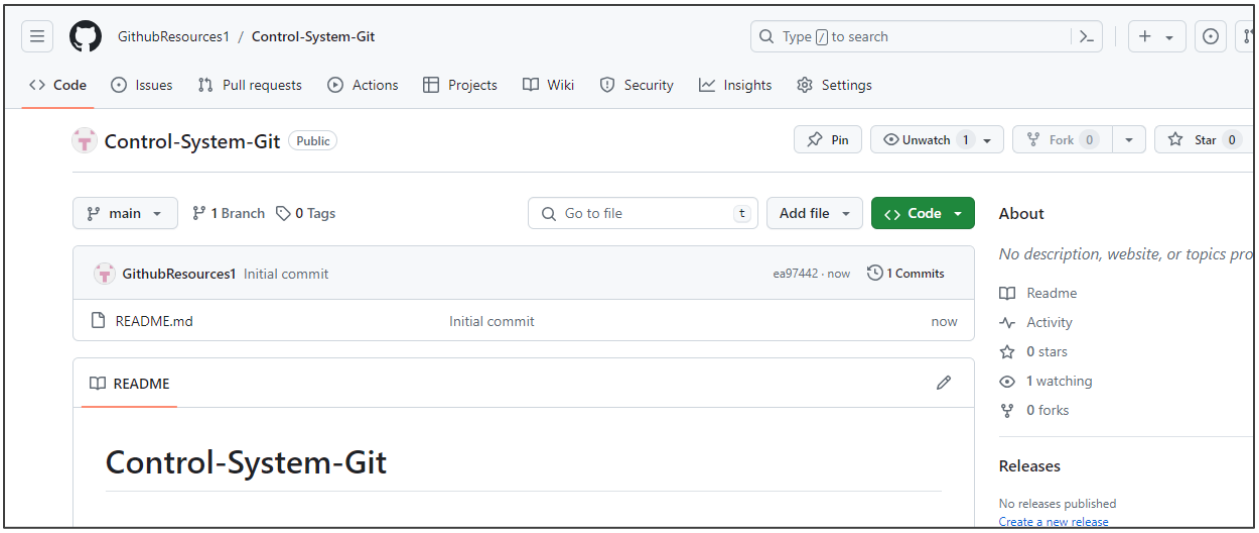


1.3 Enter the **Repository name** as **Control-System-Git**



1.4 Choose **Public** for the repository type, select **Add a README file**, and click on **Create repository**

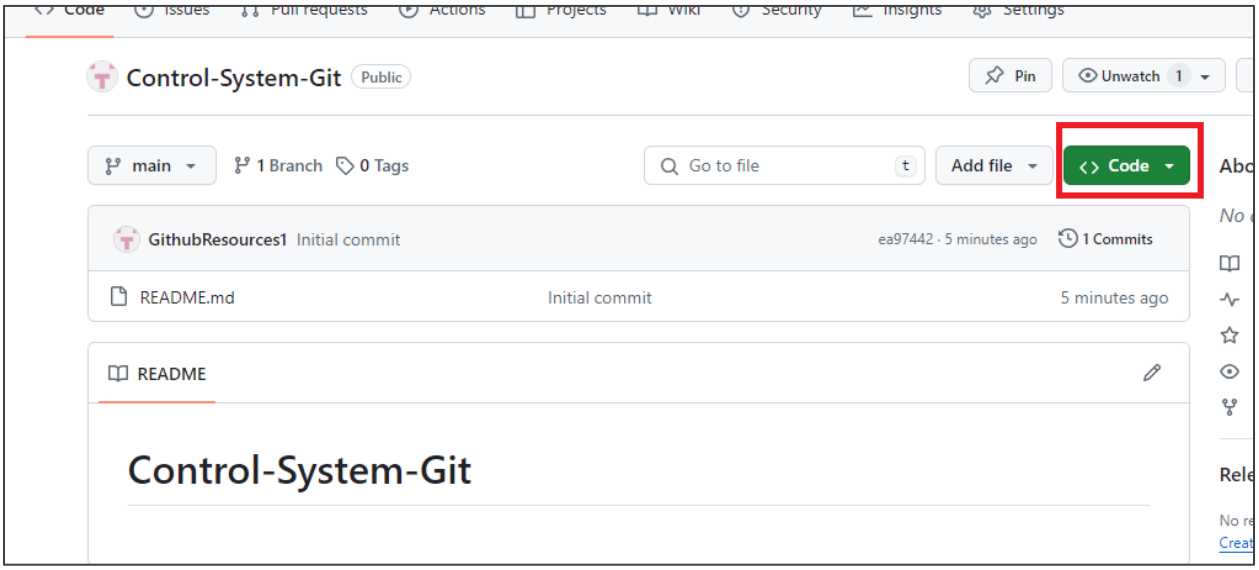




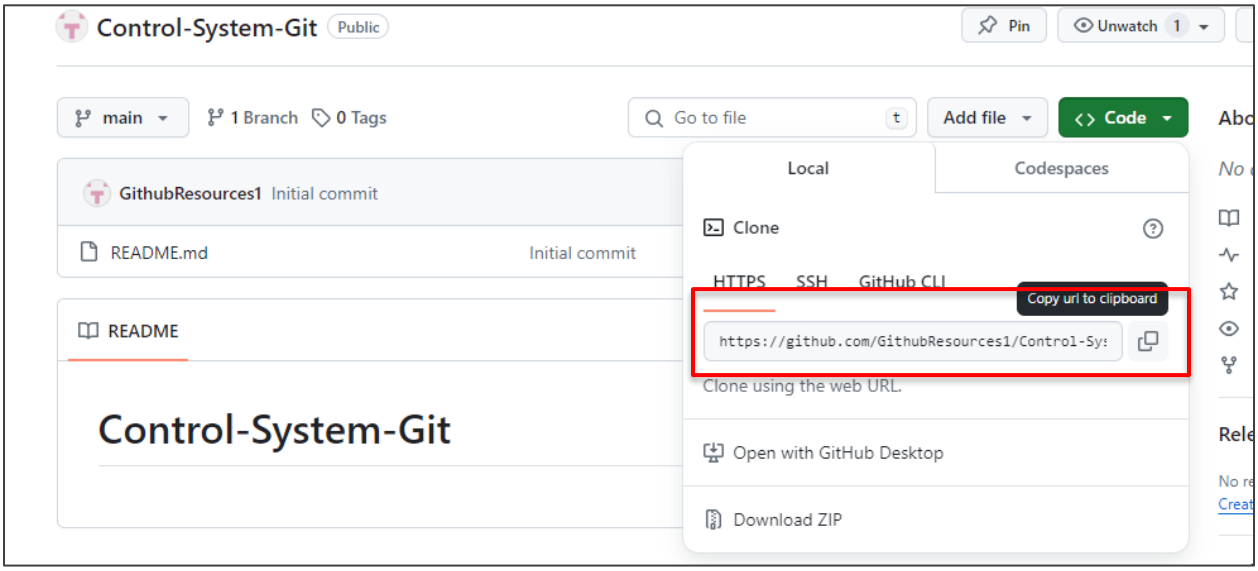
Once you create the repository, the above screen will appear.

Step 2: Clone the GitHub repository

2.1 In the **Control-System-Git** repository main page, click on **Code**



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



2.3 Open the terminal tab and execute the following command to clone the repository:
git clone <URL>

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

```
tasneemtaherasi@ip-172-31-66-20:~$ git clone https://github.com/GithubResources1/Control-System-Git.git
Cloning into 'Control-System-Git'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
tasneemtaherasi@ip-172-31-66-20:~$
```

Step 3: Create a new branch and verify it

3.1 Navigate to the **Control-System-Git** project directory by executing the following command:
cd Control-System-Git

```
tasneemtaherasi@ip-172-31-66-20:~$ cd Control-System-Git
tasneemtaherasi@ip-172-31-66-20:~/Control-System-Git$
```

3.2 Execute the following command to create a new branch named **project_branch** in your repository:
git branch project_branch

```
tasneemtaherasi@ip-172-31-66-20:~/Control-System-Git$ git branch project_branch
tasneemtaherasi@ip-172-31-66-20:~/Control-System-Git$
```

3.3 Execute the following command to verify the creation of the new branch:
git branch

```
tasneemtaherasi@ip-172-31-66-20:~/Control-System-Git$ git branch
* main
  project_branch
tasneemtaherasi@ip-172-31-66-20:~/Control-System-Git$
```

Step 4: Rename the existing branch and list all the branches

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

```
tasneemtaherasi@ip-172-31-66-20:~/Control-System-Git$ git branch -m project_branch1
tasneemtaherasi@ip-172-31-66-20:~/Control-System-Git$
```

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

```
tasneemtaherasi@ip-172-31-66-20:~/Control-System-Git$ git branch
  project_branch
* project_branch1
tasneemtaherasi@ip-172-31-66-20:~/Control-System-Git$
```

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

```
tasneemtaherasi@ip-172-31-66-20:~/Control-System-Git$ git branch project_branch2
tasneemtaherasi@ip-172-31-66-20:~/Control-System-Git$
```

5.2 Execute the following command to verify the creation of the new branch:
git branch

```
tasneemtaherasi@ip-172-31-66-20:~/Control-System-Git$ git branch
  project_branch
* project_branch1
  project_branch2
tasneemtaherasi@ip-172-31-66-20:~/Control-System-Git$
```

5.3 Use the following command to switch to the newly created **project_branch2**:
git checkout project_branch2

```
tasneemtaherasi@ip-172-31-66-20:~/Control-System-Git$ git checkout project_branch2
Switched to branch 'project_branch2'
tasneemtaherasi@ip-172-31-66-20:~/Control-System-Git$
```

Step 6: Create a file, commit the changes, and check the status of the new branch

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

```
tasneemtaherasi@ip-172-31-66-20:~/Control-System-Git$ vi index.html
tasneemtaherasi@ip-172-31-66-20:~/Control-System-Git$
```

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

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

> **Note**: Press the **Esc** button and enter **: wq** to save and exit the file

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

6.3 Execute the following command to add the file to the **project_branch2**:

git add index.html

```
tasneemtaherasi@ip-172-31-66-20:~/Control-System-Git$ git add index.html
tasneemtaherasi@ip-172-31-66-20:~/Control-System-Git$
```

6.4 Use the following command to commit all the modified files to the branch **project_branch2**:

git commit -a -m "file modified"

```
tasneemtaherasi@ip-172-31-66-20:~/Control-System-Git$ git commit -a -m "file modified"
[project_branch2 a9c5920] file modified
 1 file changed, 6 insertions(+)
 create mode 100644 index.html
tasneemtaherasi@ip-172-31-66-20:~/Control-System-Git$
```

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

```
tasneemtaherasi@ip-172-31-66-20:~/Control-System-Git$ git status
On branch project_branch2
nothing to commit, working tree clean
tasneemtaherasi@ip-172-31-66-20:~/Control-System-Git$
```

Step 7: Delete the branch and verify it

7.1 Execute the following command to delete the newly created branch:
git branch -d project_branch

```
tasneemtaherasi@ip-172-31-66-20:~/Control-System-Git$ git branch -d project_branch
Deleted branch project_branch (was ea97442).
tasneemtaherasi@ip-172-31-66-20:~/Control-System-Git$
```

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

```
tasneemtaherasi@ip-172-31-66-20:~/Control-System-Git$ git branch
  project_branch1
* project_branch2
tasneemtaherasi@ip-172-31-66-20:~/Control-System-Git$
```

Step 8: Switch back to the main branch and merge

8.1 Execute the following command to switch back to the main branch:
git checkout project_branch1

```
tasneemtaherasi@ip-172-31-66-20:~/Control-System-Git$ git checkout project_branch1
Switched to branch 'project_branch1'
Your branch is up to date with 'origin/main'.
tasneemtaherasi@ip-172-31-66-20:~/Control-System-Git$
```

8.2 Execute the following command to merge the test branch with the main branch:
git merge project_branch2

```
tasneemtaherasi@ip-172-31-66-20:~/Control-System-Git$ git merge project_branch2
Updating ea97442..a9c5920
Fast-forward
 index.html | 6 ++++++
 1 file changed, 6 insertions(+)
 create mode 100644 index.html
tasneemtaherasi@ip-172-31-66-20:~/Control-System-Git$
```

By following these steps, you have successfully created a remote GitHub repository and used Git commands to perform various version control operations for managing the remote repository.