Lesson 03 Demo 02

Executing Basic Git Commands

Objective: To demonstrate the execution of basic Git commands for synchronizing code or files with GitHub, covering repository initialization, adding files, and committing changes

Tools Required: Git

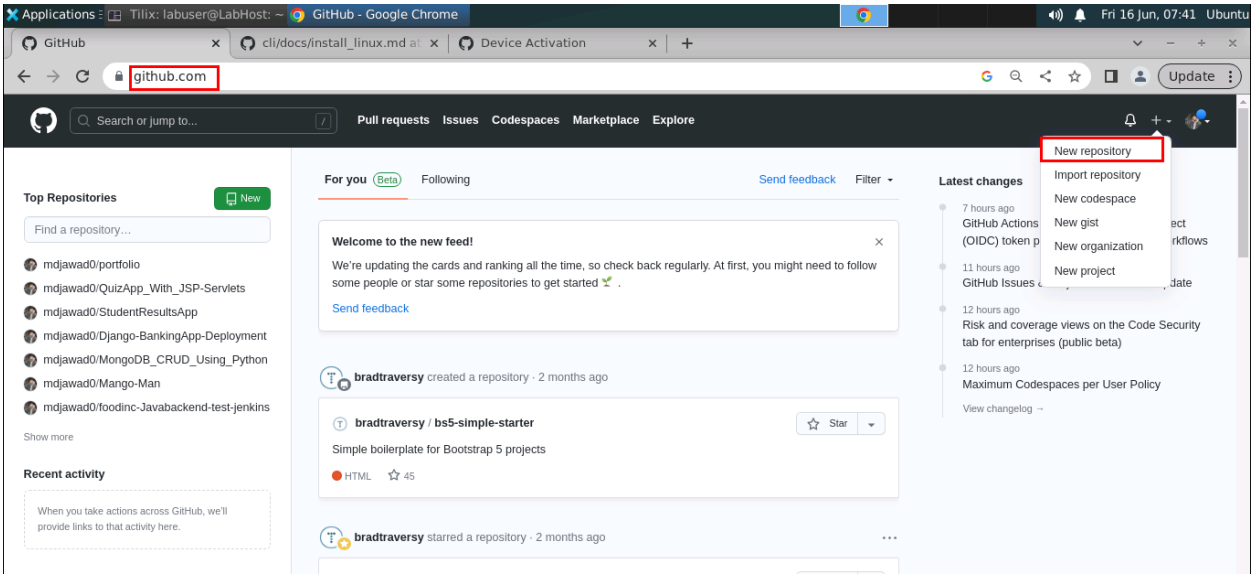
Prerequisites: Lesson 02 Demo 02

Steps to be followed:

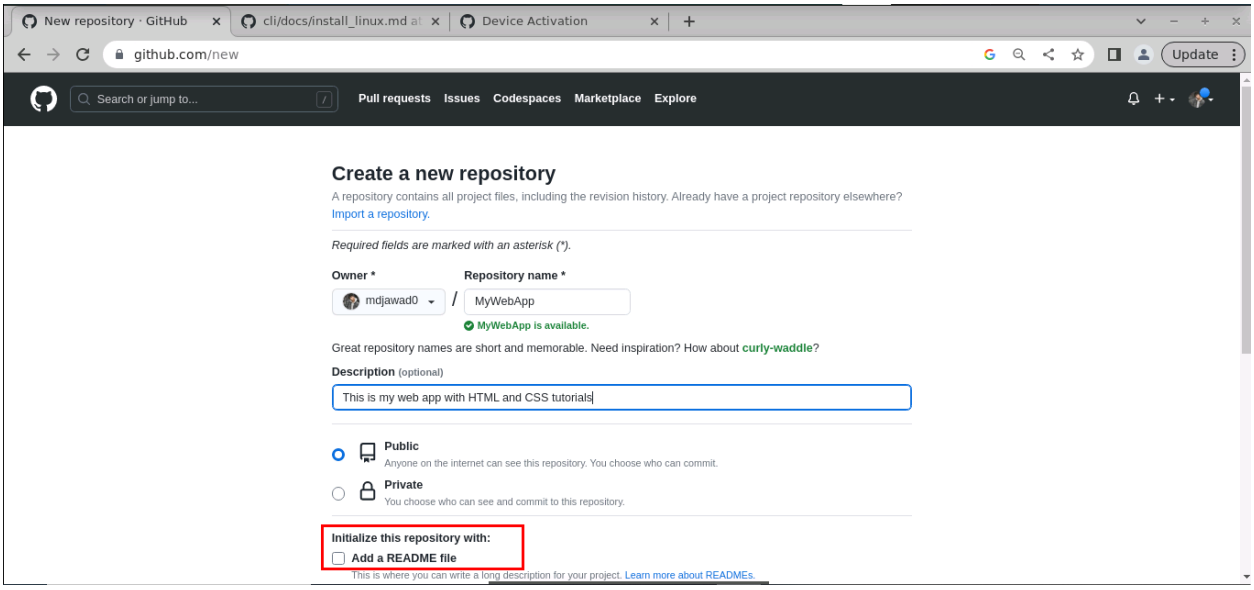
1. Execute Git commands

Step 1: Execute Git commands

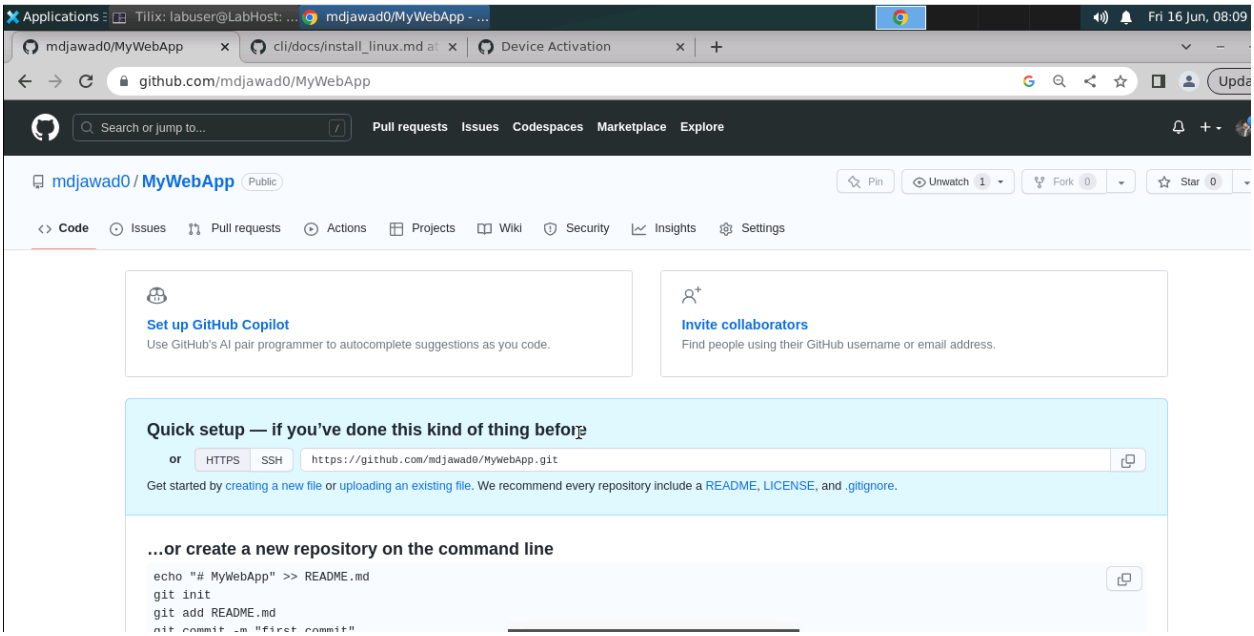
1.1 Go to **github.com** and create a new repository. You can use the name **MyWebApp** or any name of your choice.



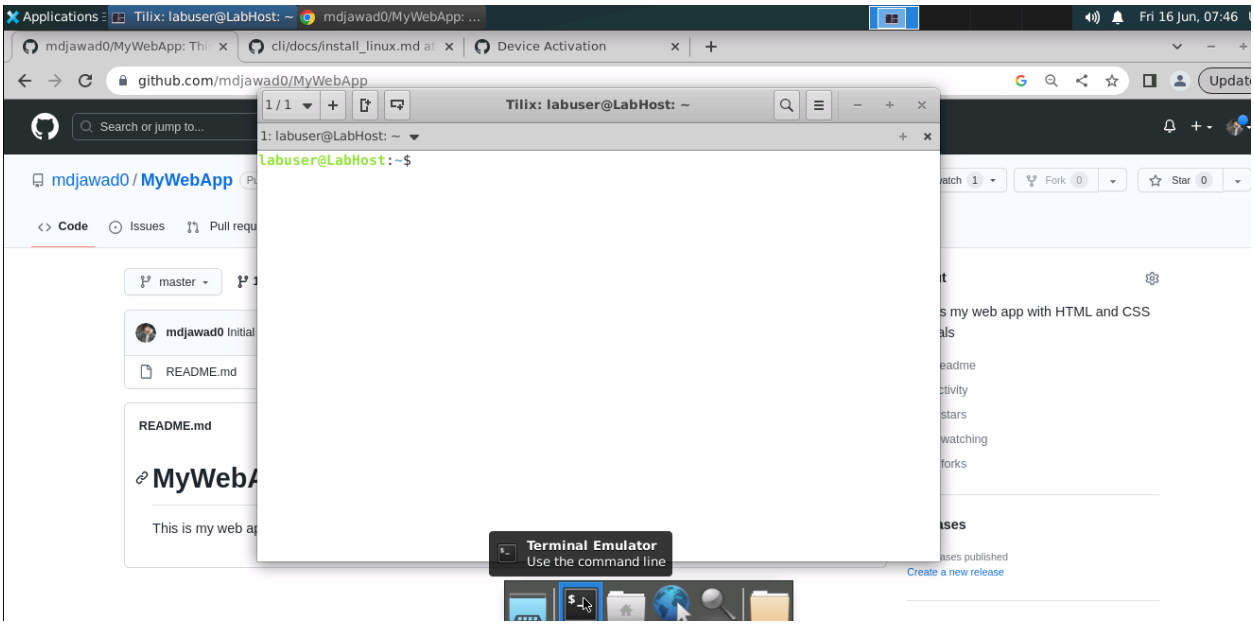
1.2 Describe the repository, such as **This is my web app with HTML and CSS tutorials.** You can choose to make the repository **public** or **private**.



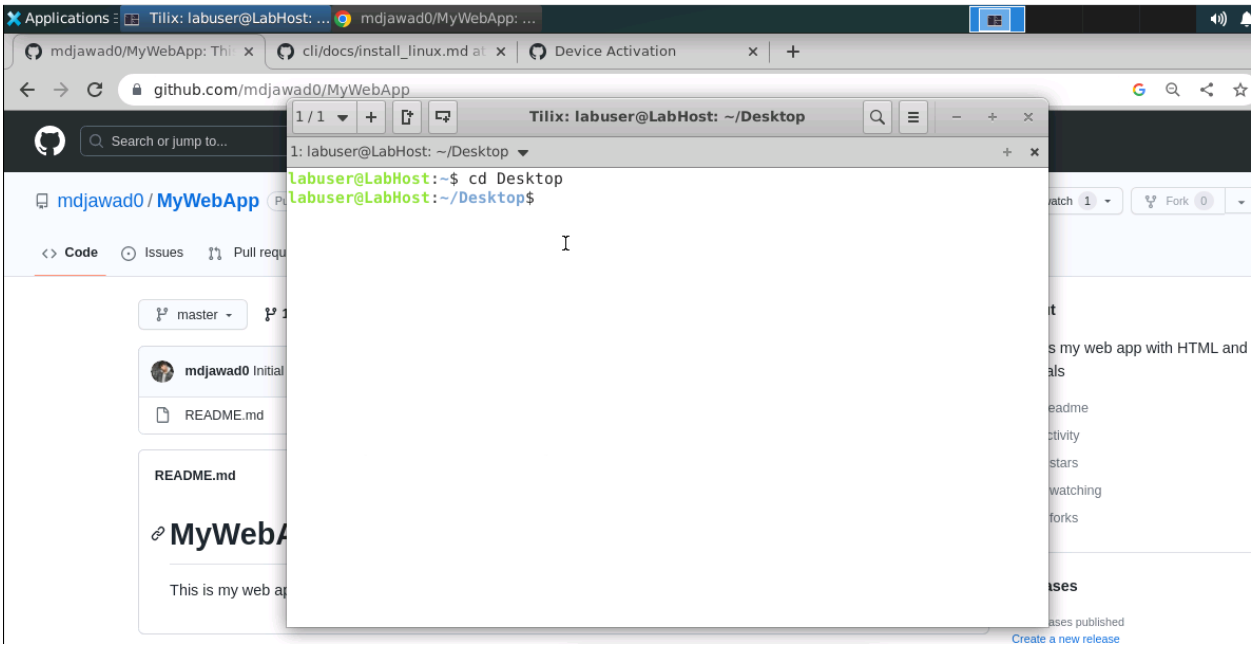
Note: Keep the **Add a README file** checkbox unchecked



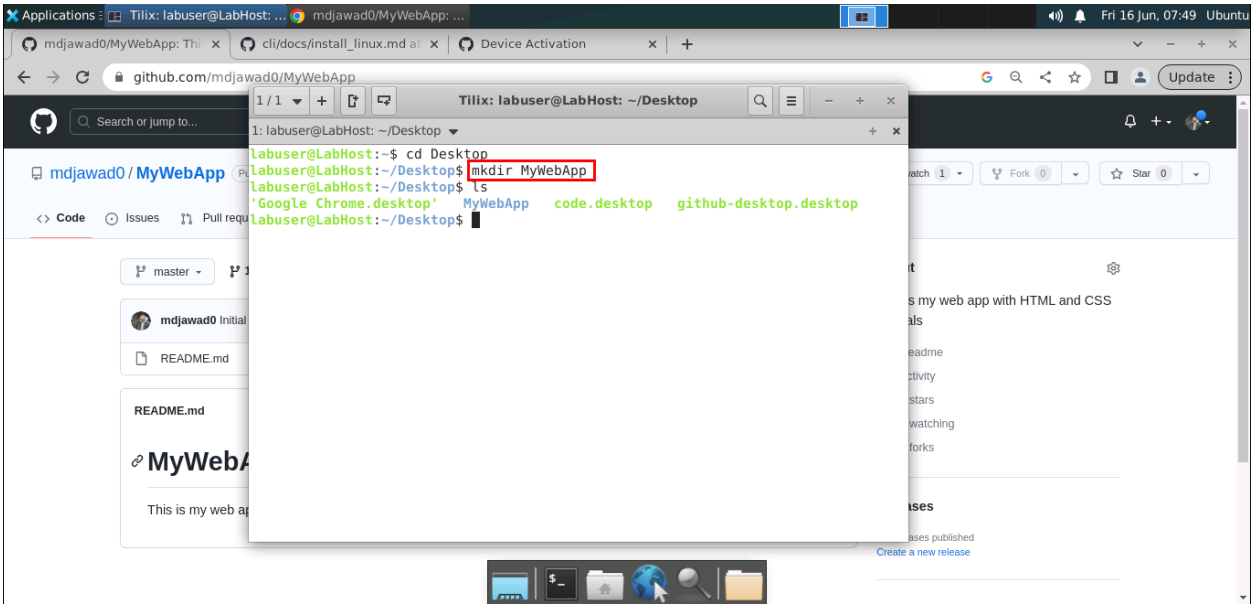
1.3 Open the terminal in the lab



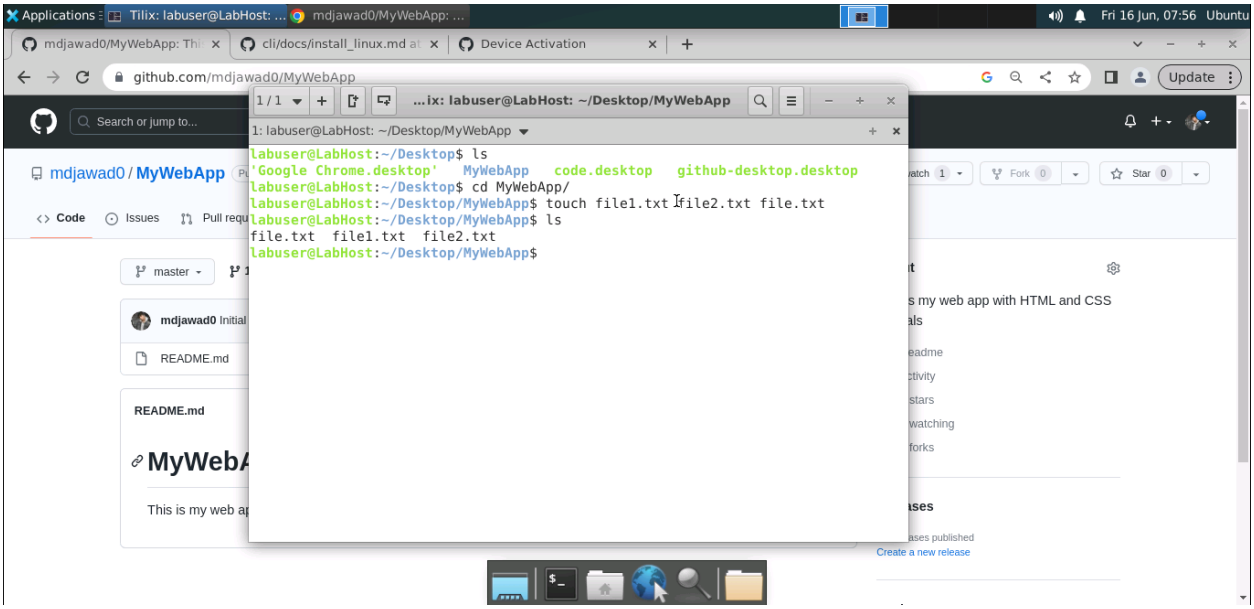
1.4 Choose a directory where you want to create the **MyWebApp** project folder. For example, you can navigate to your desktop using the cd command: **cd Desktop**.



1.5 Create a new directory named **MyWebApp** by running the command:
mkdir MyWebApp



1.6 Navigate to the **MyWebApp** and create some dummy text files by running the following command:
touch file1.txt file2.txt file.txt



Use the **ls** command to view the files.

```
palakkharbandas@ip-172-31-17-240:~/Desktop$ ls
MyWebApp  code.desktop  eclipse.desktop  file.txt  file1.txt  file2.txt
palakkharbandas@ip-172-31-17-240:~/Desktop$
```

1.7 Now, initialize Git in the project folder by running the following command:
git init

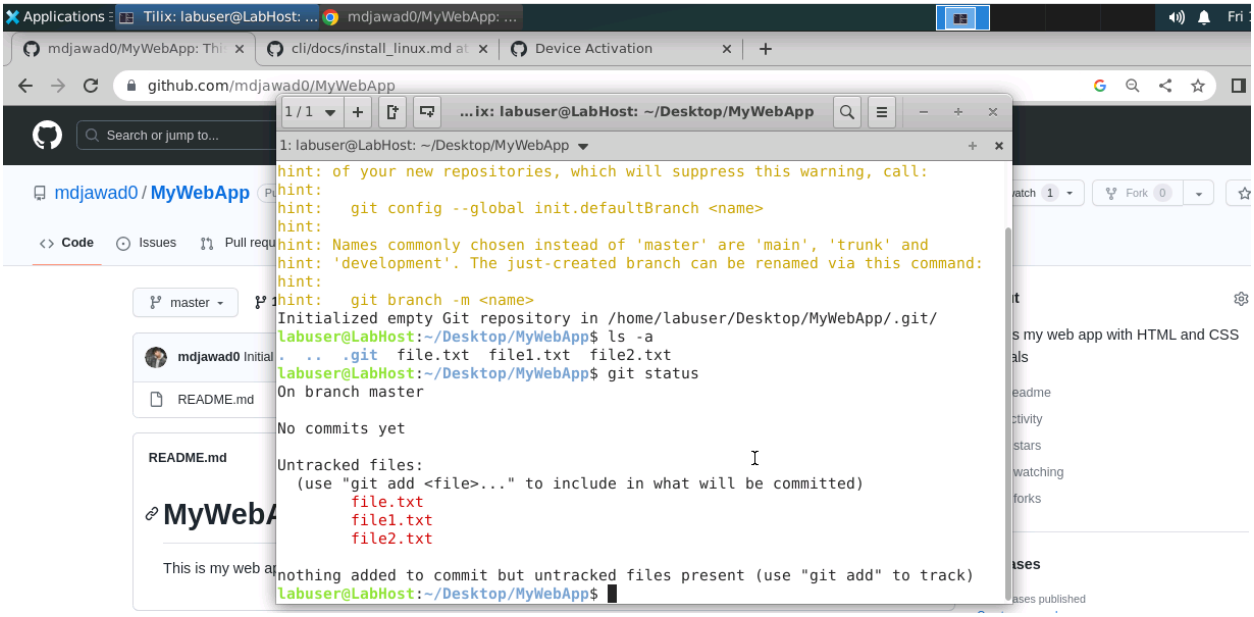
```
palakkharbandas@ip-172-31-17-240:~/Desktop$ git init
Initialized empty Git repository in /home/palakkharbandas/Desktop/.git/
palakkharbandas@ip-172-31-17-240:~/Desktop$
```

This will create a hidden **.git** directory in your project folder to track changes and manage version control.

1.8 Use the **ls -a** command to view the hidden files in your project folder, including the newly created **.git** directory

```
palakkharbandas@ip-172-31-17-240:~/Desktop$ git init
Initialized empty Git repository in /home/palakkharbandas/Desktop/.git/
palakkharbandas@ip-172-31-17-240:~/Desktop$ ls -a
.  ..  .git  MyWebApp  code.desktop  eclipse.desktop  file.txt  file1.txt  file2.txt
palakkharbandas@ip-172-31-17-240:~/Desktop$
```

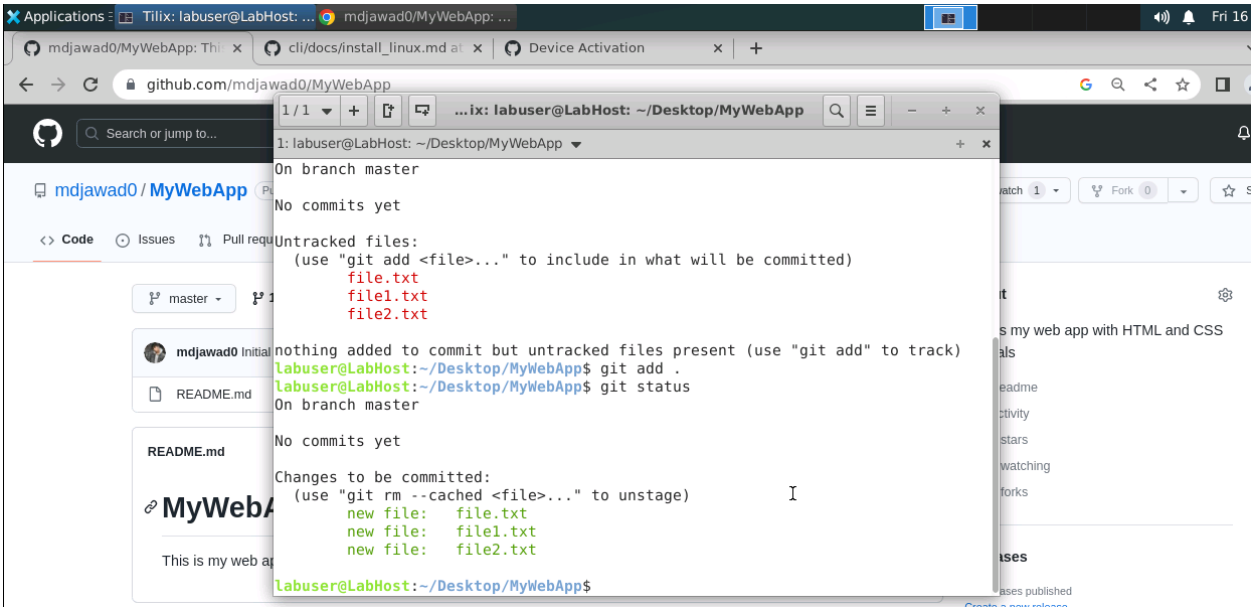
1.9 Execute the command **git status** to check the status of your Git repository



This command will show any untracked files in your project that can be added to the repository.

1.10 Add the files to the repository by running the following command:

git add .



The **.** specifies that all files in the current directory should be added. After adding the files, use **git status** again to verify that the files are staged and ready to be committed.

1.11 Commit the changes to the repository using the following commands:

git config --global user.email “user email”
git config --global user.name “user name”
git commit -m "MyWebApp Tutorial Code"

```
palakkharbandas@ip-172-31-17-240:~/Desktop/palak$ git config --global user.email "palak.kharbanda@simplilearn.net"
palakkharbandas@ip-172-31-17-240:~/Desktop/palak$ git config --global user.name "PalakSimplilearn"
palakkharbandas@ip-172-31-17-240:~/Desktop/palak$ git commit -m "MyWebApp Tutorial Code"
[master (root-commit) 05bbeda] MyWebApp Tutorial Code
 3 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file.txt
 create mode 100644 file1.txt
 create mode 100644 file2.txt
palakkharbandas@ip-172-31-17-240:~/Desktop/palak$
```

This will create a new commit with the specified message. After committing the changes, you can push them to the repository on GitHub.

By following these steps, you have successfully executed Git commands to initialize a repository, add files, and commit changes on GitHub. These steps demonstrate the basic workflow of using Git for version control and collaborating with others on your project.