

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

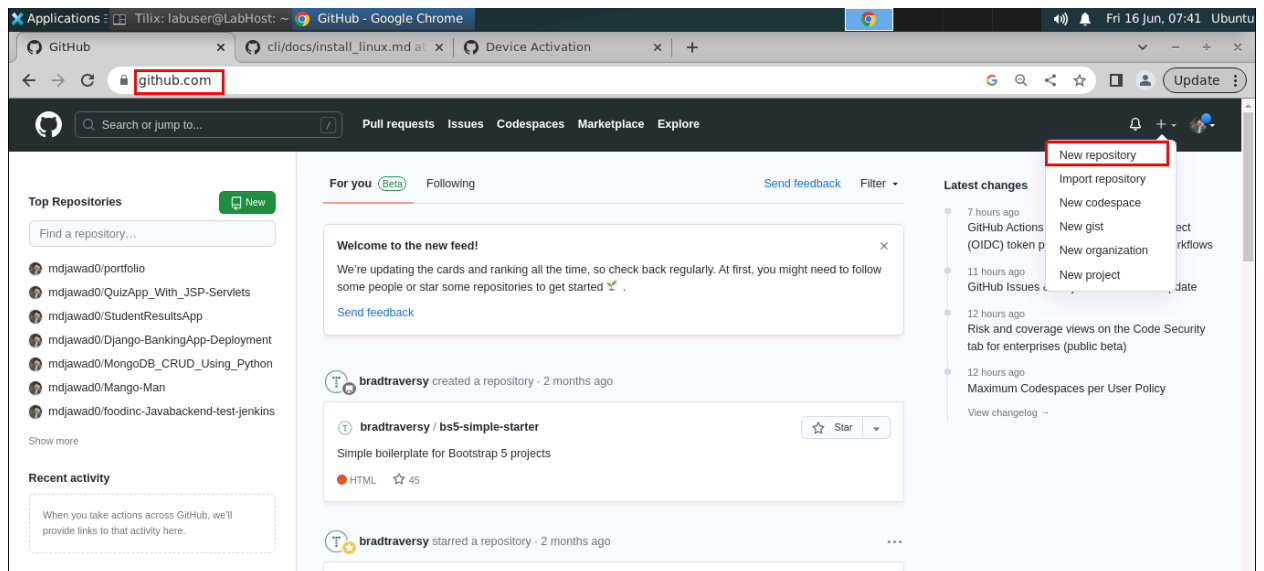
Prerequisites: Lesson 02 Demo 02

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**.

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 * **Repository name ***

/ MyWebApp

MyWebApp is available.

Great repository names are short and memorable. Need inspiration? How about curly-waddle?

Description (optional)

This is my web app with HTML and CSS tutorials

☒ **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:

☐ **Add a README file**

This is where you can write a long description for your project. [Learn more about READMEs](#)

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

mdjawad0 / MyWebApp Public

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

[Set up GitHub Copilot](#)
Use GitHub's AI pair programmer to autocomplete suggestions as you code.

[Invite collaborators](#)
Find people using their GitHub username or email address.

Quick setup — if you've done this kind of thing before

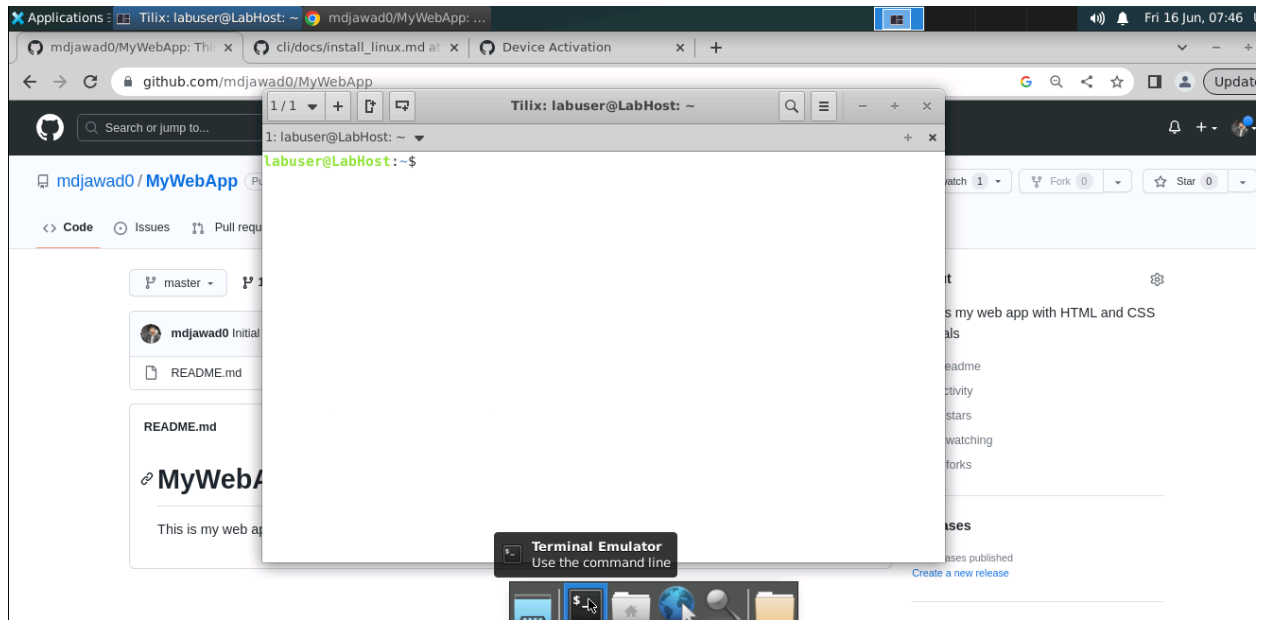
or **HTTPS** **SSH** `https://github.com/mdjawad0/MyWebApp.git`

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

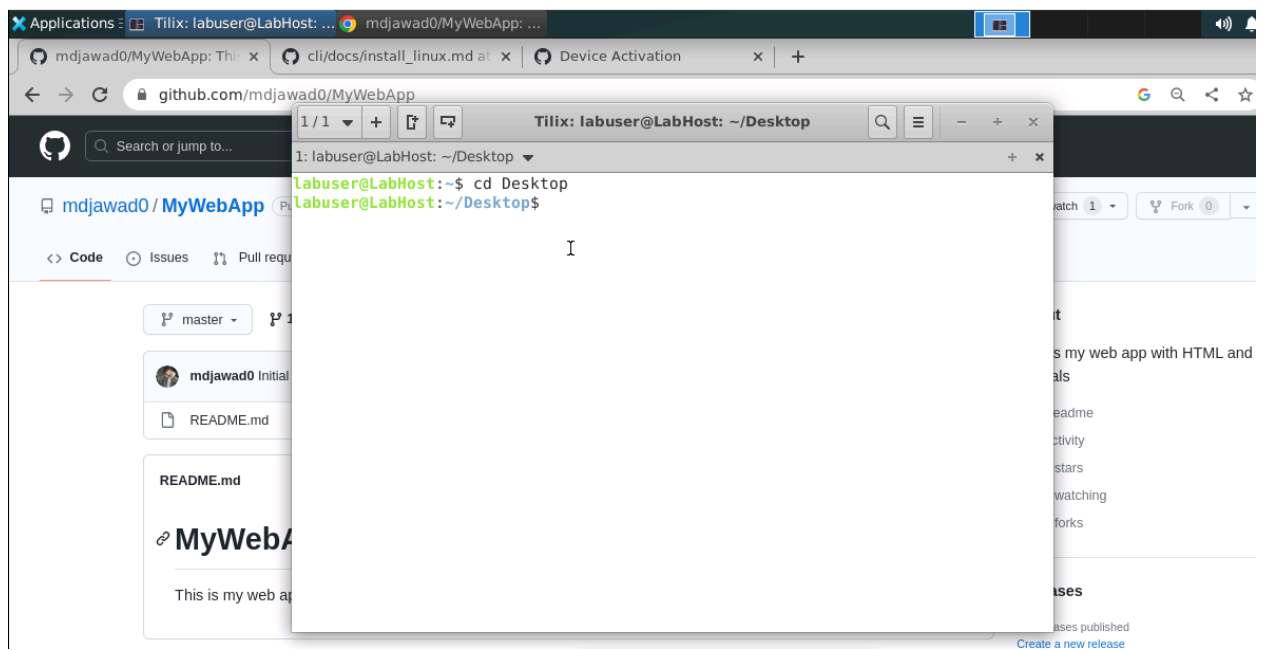
...or create a new repository on the command line

```
echo "# MyWebApp" >> README.md
git init
git add README.md
git commit -m "first commit"
```

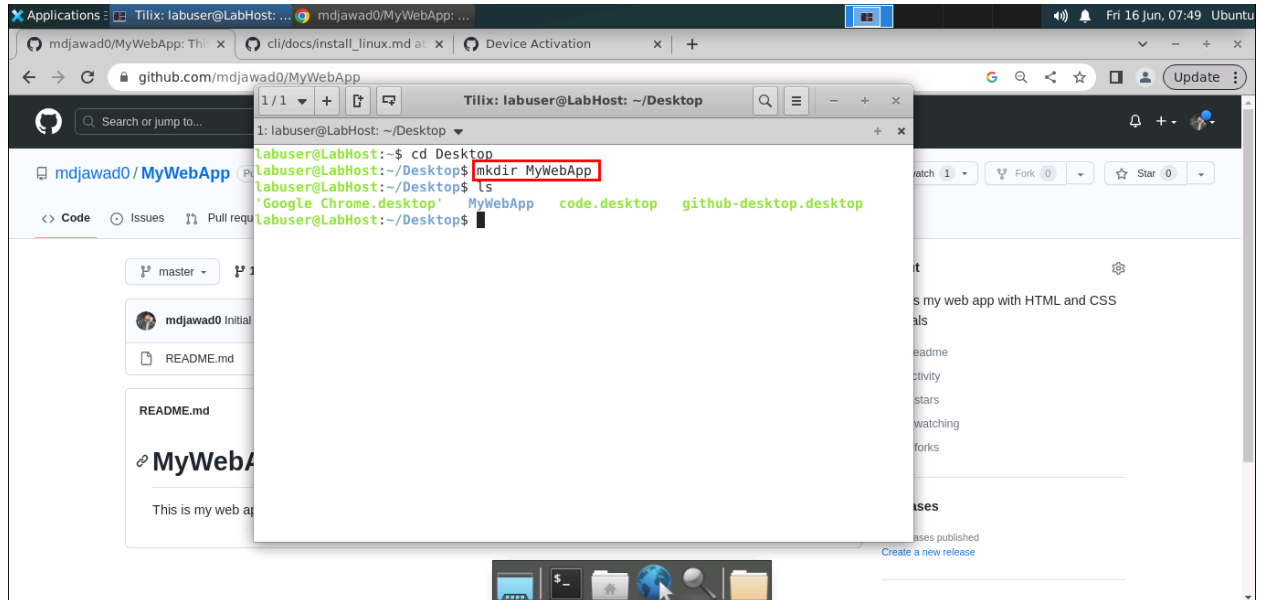
1.3 Open the terminal in the lab



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**.



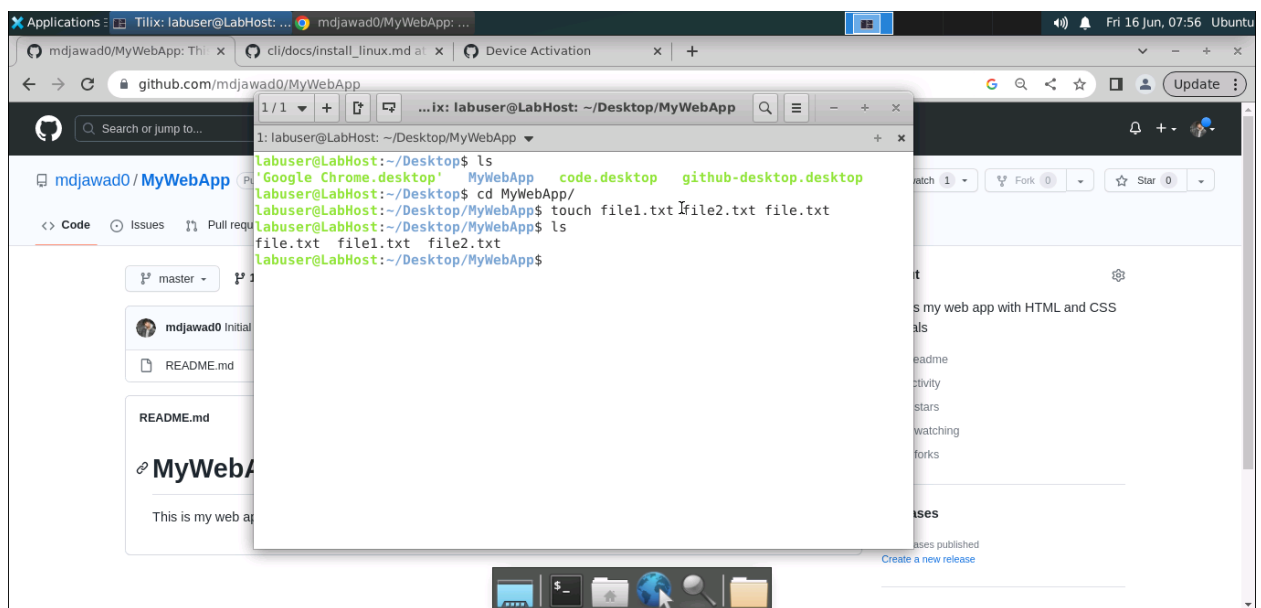
1.5 Create a new directory named **MyWebApp** by running the command:
mkdir MyWebApp



The screenshot shows a terminal window titled 'Tilix: labuser@LabHost: ~/Desktop'. The user has navigated to the Desktop directory and run the command `mkdir MyWebApp`, which is highlighted with a red box. The output of the command is `mkdir: created directory 'MyWebApp'`. The background shows a web browser displaying the GitHub repository for 'mdjawad0/MyWebApp'.

```
labuser@LabHost:~/Desktop$ cd Desktop
labuser@LabHost:~/Desktop$ mkdir MyWebApp
labuser@LabHost:~/Desktop$ ls
'Google Chrome.desktop'  MyWebApp  code.desktop  github-desktop.desktop
labuser@LabHost:~/Desktop$
```

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



The screenshot shows the terminal window with the title 'Tilix: labuser@LabHost: ~/Desktop/MyWebApp'. The user has navigated to the MyWebApp directory and run the command `touch file1.txt file2.txt file.txt`. The output shows the files have been created. The background shows the same GitHub repository page.

```
labuser@LabHost:~/Desktop$ ls
'Google Chrome.desktop'  MyWebApp  code.desktop  github-desktop.desktop
labuser@LabHost:~/Desktop$ cd MyWebApp/
labuser@LabHost:~/Desktop/MyWebApp$ touch file1.txt file2.txt file.txt
labuser@LabHost:~/Desktop/MyWebApp$ ls
file.txt  file1.txt  file2.txt
labuser@LabHost:~/Desktop/MyWebApp$
```

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

```

1: labuser@LabHost: ~/Desktop/MyWebApp
labuser@LabHost:~/Desktop/MyWebApp$ ls -a
.  ..  .git  file.txt  file1.txt  file2.txt
labuser@LabHost:~/Desktop/MyWebApp$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        file.txt
        file1.txt
        file2.txt

nothing added to commit but untracked files present (use "git add" to track)
labuser@LabHost:~/Desktop/MyWebApp$

```

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 .**

```

1: labuser@LabHost: ~/Desktop/MyWebApp
labuser@LabHost:~/Desktop/MyWebApp$ git add .
labuser@LabHost:~/Desktop/MyWebApp$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   file.txt
        new file:   file1.txt
        new file:   file2.txt

labuser@LabHost:~/Desktop/MyWebApp$

```

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) 05bbda] 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.