R U E T

Rajshahi University of Engineering & Technology
Department of Computer Science & Engineering

Lab Report - 1

Course Code: CSE 3206

Course Title: Software Engineering Sessional

Submitted By-	Submitted To-
Name : Sajidur Rahman Tarafder	**Farjana Parvin**
Department : CSE	**Lecturer**
Roll No. : 2003154	**Department of CSE, RUET**
Section : C	
Session:2020-21	

Lab Task: Setting up Git and Uploading a Repository to GitHub

Objective:

This lab aims to provide a clear understanding of how to install and configure Git, create a local repository, and upload it to a GitHub repository. It covers the foundational steps of setting up Git, creating and managing repositories, and utilizing Git commands to track, commit and push changes effectively to GitHub.

Requirements:

1. Git
2. VsCode
3. GitHub Account

Procedure:
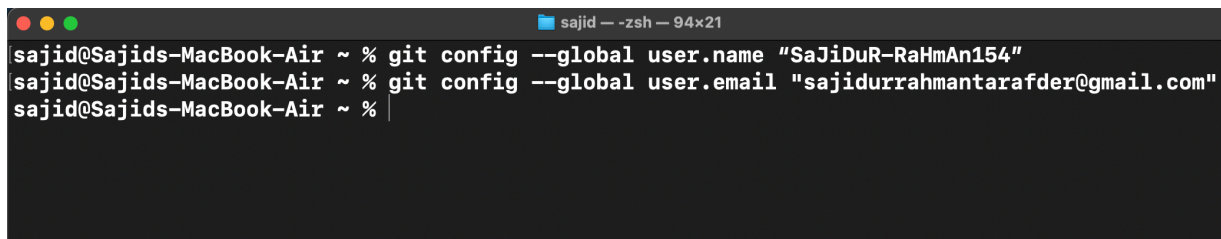
Step-1: Installation of Git

1. Download Git
2. Install Git

Step-2: **Configuring Git**

Configuring Git username:

git config --global user.name "SaJiDuR-RaHmAn154"

Configuring email address:

git config --global user.email "sajidurrahmantarafder@gmail.com"


```
sajid@Sajids-MacBook-Air ~ % git config --global user.name "SaJiDuR-RaHmAn154"
sajid@Sajids-MacBook-Air ~ % git config --global user.email "sajidurrahmantarafder@gmail.com"
sajid@Sajids-MacBook-Air ~ %
```
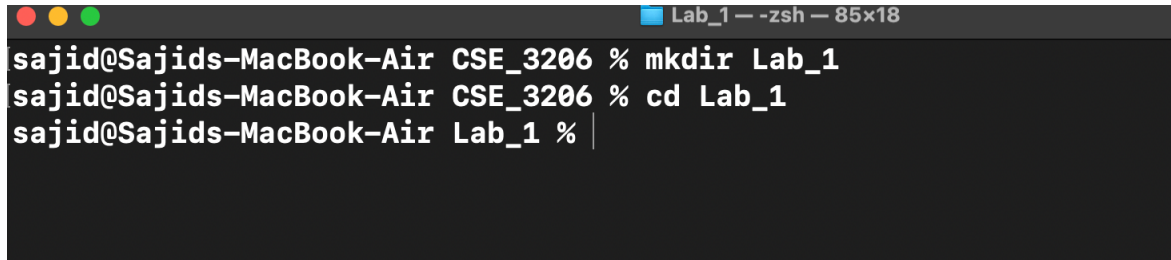
Step-3: Creating a Local Repository

Navigate to the folder to use or create a new one.

Terminal Commands and Output:

mkdir Lab_1
cd Lab_1

```
sajid@Sajids-MacBook-Air CSE_3206 % mkdir Lab_1
sajid@Sajids-MacBook-Air CSE_3206 % cd Lab_1
sajid@Sajids-MacBook-Air Lab_1 %
```

Step-4: Initializing a Git repository

This step creates a (.git) folder in the Lab_1 directory, which tracks the changes in your project.

Terminal Command and Output:

git init

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS    POSTMAN CONSOLE

sajid@Sajids-MacBook-Air Lab_1 % git init
Initialized empty Git repository in /Users/sajid/Desktop/CSE_3206/Lab_1/.git/
sajid@Sajids-MacBook-Air Lab_1 %
```

Step-5: Create some files in Lab_1 Directory

Creating some demo files in the Lab_1 directory.

Terminal Commands and Output:

touch index.html
touch style.css
touch app.js

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   POSTMAN CONSOLE                                          zsh - Lab_1  □ 📖 ⋯ ∧ ✕
sajid@Sajids-MacBook-Air Lab_1 % touch index.html
sajid@Sajids-MacBook-Air Lab_1 % touch style.css
sajid@Sajids-MacBook-Air Lab_1 % touch app.js
sajid@Sajids-MacBook-Air Lab_1 %
```

```
                                               CSE_3206
EXPLORER                    ...   index.html U ✕   style.css U      app.js  U
∨ CSE_3206                        Lab_1 >  index.html >  html >  body >  h1
  □ Lab_1                     1   <!DOCTYPE html>
    app.js            U       2   <html lang="en">
    index.html        U       3   <head>
    style.css         U       4       <meta charset="UTF-8">
                              5       <meta name="viewport" content="width=device-width, initial-scale=1.0">
                              6       <title>Document</title>
                              7   </head>
                              8   <body>
                              9       <h1>Hello World</h1>
                             10   </body>
                             11   </html>
```

Step-6: Adding Files to the Repository

Create or copy files into the directory. Add files to the staging area.

Terminal Command and Output:

git add .

The . adds all files in the directory to the staging area.

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   POSTMAN CONSOLE                                          zsh - Lab_1  □ 📖 ⋯ ∧ ✕
sajid@Sajids-MacBook-Air Lab_1 % git add .
sajid@Sajids-MacBook-Air Lab_1 % git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   app.js
        new file:   index.html
        new file:   style.css
```

Step-6: Commit the Changes

Commit the changes to the repository with a message.

Terminal Command and Output:

 git commit -m "Initial Commit"

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   POSTMAN CONSOLE                                    zsh - Lab_1        ...  ^  X
• sajid@Sajids-MacBook-Air Lab_1 % git commit -m "Initial Commit"
 [main (root-commit) 1a8a551] Initial Commit
  3 files changed, 15 insertions(+)
  create mode 100644 app.js
  create mode 100644 index.html
  create mode 100644 style.css
```

Step-7: Creating a GitHub Repository

Click the New Repository button.

```
Find a repository...                                        Type ▾   Language ▾   Sort ▾   🖵 New
```

Provide a name for the repository (CSE_3206_Lab_1) and optionally add a description. Leave other settings as default and click Create Repository.

Copy the repository's remote URL:
(https://github.com/SaJiDuR-RaHmAn154/CSE_3206_Lab_1.git)

Step-8: Link Local Repository to GitHub

Adding the remote repository to my local repository.

Terminal Command and Output:

git remote add origin https://github.com/SaJiDuR-RaHmAn154/CSE_3206_Lab_1.git
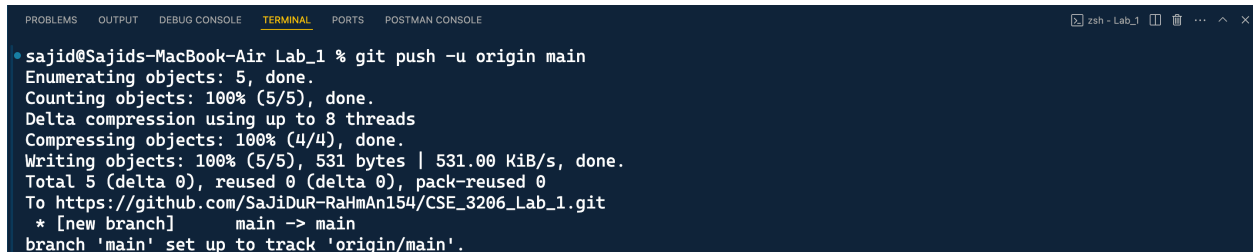
Verify the remote repository:

git remote -v

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   POSTMAN CONSOLE                                    zsh - Lab_1        ...  ^  X
• sajid@Sajids-MacBook-Air Lab_1 % git remote add origin https://github.com/SaJiDuR-RaHmAn154/CSE_3206_Lab_1.git
• sajid@Sajids-MacBook-Air Lab_1 % git remote -v
 origin  https://github.com/SaJiDuR-RaHmAn154/CSE_3206_Lab_1.git (fetch)
 origin  https://github.com/SaJiDuR-RaHmAn154/CSE_3206_Lab_1.git (push)
```

Step-9: Push Changes to GitHub

Push the committed changes to GitHub repository.
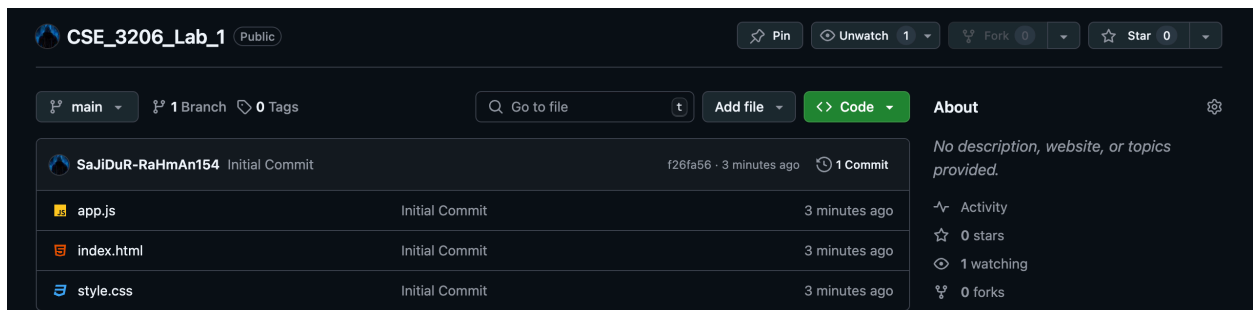**Terminal Command and Output:**

git push -u origin main

Step-10: Verify on GitHub

Go to your GitHub repository page and refresh the browser.
Confirm that your files are visible in the repository.

Results:

The files from the local directory were effectively tracked, committed, and uploaded to the GitHub repository. All Git commands executed smoothly without any issues, and the repository is now successfully live on GitHub.

Discussion:

This lab provided a step-by-step guide to the basic workflow of Git and GitHub. It covered essential commands such as initializing a repository,

tracking changes, making commits, and pushing updates to a remote repository. These concepts are fundamental for version control and collaborative software development.

Conclusion:

This lab successfully illustrated the setup process for Git and its integration with GitHub. Mastering these techniques is vital for efficient code management and teamwork in distributed environments. Future sessions can delve deeper into advanced Git operations, including branching, merging, and resolving conflicts.