Lab Manual: Getting Started with Git and GitHub

# Step 1: Create a GitHub Account

1. Visit https://github.com  
2. Click on 'Sign up' and follow the instructions.  
3. Choose a username, provide your email, and set a password.  
4. Verify your email address and finish the account setup.

# 

# Step 2: Install Git

1. Go to https://git-scm.com/downloads  
2. Download the Git installer for your operating system (Windows, macOS, or Linux).  
3. Run the installer and use default settings.  
4. After installation, open terminal/command prompt and type:  
 git --version  
 This should display the installed Git version.

# Step 3: Configure Git

1. Set your Git username and email using the following commands:  
 git config --global user.name "Your Name"  
 git config --global user.email "your\_email@example.com"

# Step 4: Create a Repository on GitHub

1. Login to your GitHub account.  
2. Click the '+' icon in the top-right corner and select 'New repository'.  
3. Give your repository a name (e.g., lab-git-demo).  
4. Set it as Public or Private and click 'Create repository'.

# 

# Step 5: Clone Repository Locally

1. Copy the repository's URL from GitHub (e.g., https://github.com/user/lab-git-demo.git).  
2. Open terminal or command prompt and type:  
 git clone <repository-URL>  
3. Navigate into the cloned folder:  
 cd lab-git-demo

# 

# Step 6: Add Files to Repository

1. Create a new file (e.g., hello.txt) inside the folder.

echo. >hello.txt

echo $(date) Hi. >> hello.txt

type hello.txt

2. Add it to staging:  
 git add hello.txt  
 or add all files:  
 git add .

# Step 7: Commit Changes

1. Save your changes in a commit with a message:  
 git commit -m "Added hello.txt"

# Step 8: Push Changes to GitHub

1. Push your local commits to GitHub:  
 git push origin main  
2. Login using your GitHub credentials or Personal Access Token (PAT) if prompted.

# Step 9: Verify Changes on GitHub

1. Visit your GitHub repository page.  
2. You should see the newly added file and commit message there.

**Congratulations! You’ve successfully completed your Git and GitHub setup lab.**

## Lab Task Make and Track Edits

* Modify a file, commit the change, and push again.
* Steps:
* Add more lines to hello.txt
* Run:
* git add hello.txt
* git commit -m "Updated about\_me.txt"
* git push origin main
* Lab Task Git Log Exploration
* View and understand the commit history.
* Steps:
* git log --oneline

## ****Create and Switch to a New Branch****

**Objective**: Learn how to work in isolated development branches.

**Steps**:

git checkout -b feature/login-page

* Add a new file or modify existing one
* Commit changes

git add .

git commit -m "Added login page feature"

### ****Lab Task : Merge a Branch into Main****

**Objective**: Merge changes from the feature branch into the main branch.

**Steps**:

git checkout main

git merge feature/login-page

### ****Lab Task : Resolve a Merge Conflict****

**Objective**: Experience and resolve a simulated merge conflict.

**Instructions**:

* Modify the same line in a file on two different branches
* Merge branches and resolve conflict manually
* Commit the resolved version

**Step 1: Create two branches from main**

* git checkout main
* git checkout -b branch1
* echo "Line A" > conflict.txt
* git add .
* git commit -m "Branch1: Added Line A"
* git push origin branch1

**Step 2: Switch to second branch and edit the same line**

* git checkout main
* git checkout -b branch2
* echo "Line B" > conflict.txt
* git add .
* git commit -m "Branch2: Added Line B"
* git push origin branch2

**Step 3: Attempt to merge → will trigger conflict**

* git checkout branch1
* git merge branch2

## Lab Task:

* Create a repository with your Final Year Projects Name add members by assigining them their branches.
* Add your FYP Documentation into it.
* Clone that repo on your local machine
* See the files and update some text in it.
* Merge changes to main branch.
* Commit changes and see the updations on github

### ****Create a Repository for FYP****

**Owner (Team Leader)**

1. Log in to GitHub →Click New Repository
2. Name it something like: FYP-SmartEnergyMonitor
3. (Optional) Add a README file
4. Create repo → copy the HTTPS link

### ****Step 2: Add Members as Collaborators****

1. On the repo page → Go to **Settings → Collaborators**
2. Add your team members by GitHub username
3. Each member will receive an invite to join

### ****Step 3: Assign Branches to Members****

Each member creates and works on their own branch:

git checkout -b ali-ui-module

git push -u origin ali-ui-module

Similarly:

git checkout -b sara-docs

git push -u origin sara-docs

(Use descriptive branch names: login-auth, data-modeling, etc.)

### S****tep 4: Add FYP Documentation****

Upload the full documentation (Word/PDF) to the repo or one of the branches (e.g., sara-docs):

* Use “Upload files” button on GitHub  
  **OR**
* Clone the repo and add via terminal (next steps)

### ****Step 5: Clone Repo Locally****

On **any team member’s** machine:

git clone https://github.com/yourusername/FYP-SmartEnergyMonitor.git

cd FYP-SmartEnergyMonitor

### ****Step 6: View and Modify Files****

Open the cloned folder:

* Use any editor (e.g., VS Code)
* Modify the documentation (e.g., edit a .docx or .txt)

### ****Step 7: Commit & Push Changes****

git add .

git commit -m "Updated introduction in FYP documentation"

git push origin your-branch-name

### ****Step 8: Merge to Main Branch****

Once team members are done, the team leader (or assigned reviewer):

git checkout main

git pull origin main

git merge ali-ui-module

git merge sara-docs

git push origin main

### ****Step 9: Verify Changes on GitHub****

1. Visit the GitHub repo in browser
2. Check the main branch → Documentation should be updated
3. Go to **Commits** tab to verify history