Git & GitHub Usage Guide

Smart Energy Consumption Analysis & Forecasting Project

---

1. Introduction

This document is a **beginner-friendly Git and GitHub guide** for interns working on the **Smart Energy Consumption Analysis and Prediction using Machine Learning** project.

By following this guide, interns will learn how to: - Use Git for version control - Work with GitHub repositories correctly - Follow proper commit and folder-management practices - Submit work for mentor review

📌**Important:** This is an **individual project**. Do not copy code or repositories from other interns.

---

2. Why Git Is Mandatory for This Project

Git helps you: - Track code and notebook changes - Safely experiment without losing work - Maintain clean project history - Submit progress transparently for evaluation

🗄️Projects **without proper Git usage will not be evaluated**.

---

3. Tools Required

Ensure the following tools are installed:

- **Git** (latest stable version)
- **GitHub Desktop** (recommended for beginners)
- **Visual Studio Code (VS Code)**
- **Python (Anaconda preferred)**

🔗Verify Git installation:

```
git --version
```

---

4. GitHub Repository Setup

Official Project Repository

All interns must work inside the **central mentor repository**:

```
https://github.com/springboardmentor1361k/Smart-Energy-Consumption-Analysis-and-Prediction-using-Machine-Learning-with-Device-Level-Insights-.git
```

📌**Do NOT create separate repositories.**

Each intern must create and work on an **individual branch** inside this repository.

---

5. Clone Repository Using GitHub Desktop

1. Open **GitHub Desktop**
2. File → Clone Repository
3. Select your repository
4. Choose local path:

```
C:\Users\<Username>\Documents\SmartEnergyML
```

🔍Do NOT use OneDrive folders

---

6. Standard Project Folder Structure

Create the following folders:

```
SmartEnergyML
│
├── data
│   ├── raw
│   └── processed
├── notebooks
├── models
├── backend
├── frontend
├── dashboard
├── deployment
├── README.md
└── requirements.txt
```

📌Use VS Code terminal:

```
mkdir data data\raw data\processed notebooks models backend frontend dashboard deployment
```

---

7. Git Rule: Empty Folders Are Not Tracked

Git does **not track empty folders**.

Solution: Add `.gitkeep`

Create `.gitkeep` inside every folder:

```
New-Item data\.gitkeep
New-Item notebooks\.gitkeep
New-Item models\.gitkeep
```

---

8. Daily Git Workflow (MANDATORY)

🔍Make sure you are on **your personal branch** before starting work.

Step 1: Check Current Branch

```
git branch
```

Step 2: Pull Latest Main Changes (Daily)

```
git checkout main
git pull origin main
git checkout <your-branch>
git merge main
```

Step 3: Check Status

```
git status
```

Step 4: Stage Changes

```
git add .
```

Step 5: Commit Changes

Use **clear and meaningful commit messages**:

🔗Good examples: - `EDA completed for device-level consumption` - `Linear regression baseline model implemented`

🔩Bad examples: - `update` - `final` - `changes`

```
git commit -m "Your message here"
```

Step 6: Push to Your Branch

```
git push origin <your-branch>
```

📌Push **at least once per working day**.

---

9. Notebook Rules for Interns

When working with Jupyter notebooks: - Name notebooks clearly - Follow module-wise naming

Examples: - `01_data_understanding.ipynb` - `02_data_preprocessing.ipynb` - `03_linear_regression_model.ipynb` - `04_lstm_model.ipynb`

📌Do NOT upload unnecessary output cells or random experiments.

---

10. Branch Policy (MANDATORY)

This project follows a **branch-based individual workflow**.

🔹 Branch Rules for Interns

- Each intern must create **one personal branch**
- Branch name format:

```
<firstname>-<lastname>
```

Examples: - `ananya-m` - `akshay-kurane` - `dhayanidhi-s`

🗄️**Never commit directly to** `main` **branch**

Creating Your Branch

```
git checkout -b ananya-m
```

Push branch to GitHub:

```
git push origin ananya-m
```

Working on Your Branch

Always confirm branch before work:

```
git branch
```

All commits must be pushed to **your own branch only**.

---

11. README Update Policy

Update README.md whenever: - A new module is completed - A model is added - Dashboard or API is implemented

Example update:

```
## Current Status
Baseline Linear Regression model completed
```

---

12. Common Mistakes (Avoid These)

🔩Uploading ZIP files 🔩Copy-pasting others' repositories 🔩Large datasets without discussion 🔩No commit history 🔩Single final commit only

---

13. Evaluation via GitHub

Mentor will evaluate: - Commit frequency - Code clarity - Folder discipline - Progress consistency - Documentation quality

📌**Your GitHub repository is your project proof.**

---

14. Git Checklist for Interns

🔧 Repository created 🔧Correct folder structure 🔧Daily commits 🔧Clear commit messages 🔧 README updated regularly 🔧Code pushed to GitHub

---

🕐**Follow this Git guide strictly to avoid evaluation issues and project rejection.**

Happy Coding!