> Sunbeam

GIT & GITHUB – COMPLETE SETUP & WORKFLOW DOCUMENTATION

This document provides **complete, professional, end-to-end Git & GitHub documentation**, including installation, account setup, security setup, repository workflow, branching, contribution flow, and merging strategies.

---

1. Install Git

1.1 Download Git

For Windows / macOS / Linux:

1. Open browser → search **"Download Git"**
2. Open official site → **https://git-scm.com**
3. Click **Download** for your OS:
   - **Windows** → Git for Windows (.exe)
   - **macOS** → .dmg installer (or install via Homebrew)
   - **Linux** → use package manager:
     Ubuntu / Debian: `sudo apt install git`
     Fedora: `sudo dnf install git`
     Arch: `sudo pacman -S git`

1.2 Install Git (Windows/macOS)

Run the installer. Keep all default settings unless specific needs arise:

Important Installer Screens:

- **Editor Selection**: VS Code / Notepad++ / Vim
- **Adjust PATH** → choose:
  "Git from the command line and also from 3rd-party software"
- **Line ending style** →
  "Checkout Windows-style, commit Unix-style"

Click **Next** → **Next** → **Install** → **Finish**.

1.3 Verify Git Installation

Open CMD / Terminal / PowerShell:

```
git --version
```

If version appears → Git is installed correctly.

---

2. Create GitHub Account

1. Open **https://github.com**
2. Click **Sign Up**
3. Enter:
    - Email
    - Password
    - Username
4. Verify email using the OTP sent to inbox.
5. Complete basic profile setup.

Your GitHub account is ready.

---

3. Enable Two-Factor Authentication (2FA) with Authenticator App

GitHub strongly recommends 2FA for account security.

3.1 Open GitHub Security Settings

1. Log into GitHub
2. Click profile → **Settings**
3. Go to **Password and authentication** (or **Security** → **2FA**)

3.2 Start 2FA Setup

1. Click **Enable two-factor authentication**
2. Select **Authentication app** (recommended method)
3. GitHub will display:
    - QR code
    - Secret key

3.3 Configure Chrome Authenticator Extension

1. Open Chrome Web Store
2. Search → **Authenticator** or **GitHub OTP Authenticator**
3. Click **Add to Chrome** → **Add extension**
4. Open the extension

5. Scan GitHub's QR code OR enter secret key manually

The extension will generate 6-digit OTPs.

3.4 Complete 2FA Setup

1. Enter the 6-digit code from the Authenticator extension
2. Save GitHub's **recovery codes** safely

2FA is now enabled and secure.

---

4. Create a New GitHub Repository

4.1 Create Repository

1. Log into GitHub
2. Click **+** → **New repository**
3. Fill details:
    - Repository name
    - Description
    - Public or Private
    - Optionally: **Add README**
4. Click **Create repository**

4.2 Repository URL

After creation, GitHub shows your repo URL:

```
https://github.com/your-username/repo-name.git
```

Use this to clone your repo locally.

---

5. Add Collaborator to Repository

1. Open the repository
2. Go to **Settings** (repository level)
3. Click **Collaborators** / **Manage access**
4. Click **Add people**
5. Enter collaborator GitHub username or email
6. Collaborator receives invite → must accept

Collaborator now has access to push changes.

---

6. Clone Repository & Initial Git Setup

6.1 Configure Git Identity (Only Once)

```
git config --global user.name "Your Name"
git config --global user.email "youremail@example.com"
```

6.2 Clone Repository

```
git clone https://github.com/your-username/repo-name.git
cd repo-name
```

Repository is now on your machine.

---

7. Branching Workflow

7.1 Create a New Branch

```
git checkout -b feature-login
```

Creates AND switches to branch **feature-login**.

7.2 List All Branches

```
git branch
```

7.3 Switch Branches

```
git checkout main
git checkout feature-login
```

---

8. Make Changes: Add → Commit → Push

8.1 Check Status

```
git status
```

8.2 Stage Files

Add a specific file:

```
git add file.txt
```

Add all changes:

```
git add .
```

8.3 Commit Changes

```
git commit -m "Add login feature"
```

8.4 Push Branch to GitHub

```
git push origin feature-login
```

---

9. Pull Request & Merge Process

Standard collaboration workflow.

9.1 Open Pull Request

1. Go to repository on GitHub
2. Click **Compare & pull request** OR
3. Go to **Pull requests** → **New Pull Request**
4. Select branches:
    - Base: **main**
    - Compare: **feature-login**
5. Add title + description
6. Click **Create pull request**

9.2 Review & Merge

Owner/collaborator reviews PR.

If approved:

- Click **Merge pull request**
- Confirm merge

Branch is now merged into **main**.

9.3 Cleanup Branch (Optional)

On GitHub: Click **Delete branch**.

Locally:

```
git branch -d feature-login
git pull origin main
```

---

10. Keep Local Main Updated

Always pull the latest changes:

```
git checkout main
git pull origin main
```

---

11. Complete Git Workflow Summary

Local Development Flow

```
git checkout -b feature-1
# make changes
git add .
git commit -m "message"
git push origin feature-1
```

GitHub Collaboration Flow

```
Create Pull Request
Review Code
```

```
Merge to main
Delete branch
```