GIT & GITHUB – NEW PROJECT WORKFLOW (BEGINNER FRIENDLY)

🟢 PART 1: STEPS YOU WILL FOLLOW EVERY TIME

🔹 STEP 0: Before You Start (ONE TIME ONLY)

✓ Git installed
✓ GitHub account ready

You've already done this ✅

🔹 STEP 1: Create a Project Folder (Local)

Create a normal folder anywhere on your PC.

Example:

DSA

Put your files inside it:

sum.c

reverse.c

🔹 STEP 2: Open Terminal in That Folder

👉 Right-click inside the folder
👉 **Open Git Bash here** (or PowerShell)

🔹 STEP 3: Initialize Git

```
git init
```

📌 **What this does:**
Turns your normal folder into a **Git-tracked project**
(Creates a hidden .git folder)

🔹 STEP 4: Check File Status

```
git status
```

📌 **What this does:**
Shows:

- untracked files
- modified files
- commit status

🔹 STEP 5: Add Files to Staging

```
git add .
```

📌 **Meaning of .**
Adds **ALL files** in the folder to staging.

🔹 STEP 6: Commit Your Files

```
git commit -m "Initial commit"
```

📌 **What this does:**
Saves a **snapshot** of your project locally.

🔹 STEP 7: Create GitHub Repository

On **github.com**:

1. Click **+ → New repository**
2. Repo name (same as folder recommended)
3. ❌ Don't add README
4. Click **Create repository**

🔹 STEP 8: Connect Local Folder to GitHub

```
git remote add origin https://github.com/USERNAME/REPO_NAME.git
```

📌 **What this does:**
Links your local project to GitHub.

Check connection:

```
git remote -v
```

🔹 STEP 9: Push Code to GitHub (FIRST TIME)

```
git branch -M main
```

```
git push -u origin main
```

📌 **What this does:**
Uploads your project to GitHub
-u remembers the branch for future pushes.

🎉 Your project is now live on GitHub!

🔄 DAILY WORK (MOST IMPORTANT PART)

Whenever you change or add files:

```
git pull
```

```
git add .
```

```
git commit -m "describe your change"
```

```
git push
```

📌 **Why this order?**

- git pull → avoids errors
- git add → stages changes
- git commit → saves version
- git push → uploads to GitHub

🧹 OPTIONAL BUT PROFESSIONAL: .gitignore

Create .gitignore file:

```
notepad .gitignore
```

Add:

```
*.exe

*.out
```

📌 **What this does:**
Prevents useless compiled files from going to GitHub.