Here is your **Final PowerShell Version** of the **Complete Git + GitHub SSH Workflow** — written as a .ps1 script. This version is:

✅ **Runnable line-by-line** in **PowerShell** (either inside or outside VS Code)
✅ Includes **comments (#)** explaining each step
✅ Ideal for **real work** and **interview demonstration**

---

1 📜 Final Git + GitHub Workflow — PowerShell Script Version

powershell

CopyEdit

```
# ----------------------------------------
# 🚀 STEP 1: Project Setup
# ----------------------------------------
mkdir MyProject
cd MyProject
code .                    # Open VS Code in this directory

# ----------------------------------------
# 📁 STEP 2: Initialize Git Repo
# ----------------------------------------
git init                  # Start version control

# ----------------------------------------
# 📝 STEP 3: Create Files
```

```
# ----------------------------------------

New-Item README.md -ItemType File

Add-Content README.md "# My Project"


New-Item index.js -ItemType File

Add-Content index.js "console.log('Hello');"


# ----------------------------------------

# ✅ STEP 4: Stage & Commit Changes

# ----------------------------------------

git add .

git commit -m "Initial commit"


# ----------------------------------------

# 🌳 STEP 5: Rename Default Branch to 'main'

# ----------------------------------------

git branch -M main


# ----------------------------------------

# 🔐 STEP 6: Add GitHub Remote via SSH

# Replace with your actual SSH repo URL

# ----------------------------------------

git remote add origin git@github.com:your-username/your-repo.git
```

```
git remote -v                  # Confirm connection


# ----------------------------------------
# 🚀 STEP 7: Push to GitHub
# ----------------------------------------
git push -u origin main          # Push main branch for the first time


# ----------------------------------------
# 🌿 STEP 8: Create & Push Feature Branch
# ----------------------------------------
git switch -c feature1

New-Item feature1.js -ItemType File
Add-Content feature1.js "console.log('Feature 1');"

git add .
git commit -m "Add feature1"
git push -u origin feature1


# ----------------------------------------
# 🔁 STEP 9: Merge Feature Branch to Main
# ----------------------------------------
git switch main
```

```
git pull origin main            # Ensure local main is up-to-date
git merge feature1
git push

# ----------------------------------------
# 🏷️ STEP 10: Add a Tag
# ----------------------------------------
git tag v1.0 -m "Initial release"
git push origin v1.0

# ----------------------------------------
# 🔍 STEP 11: Inspect Repository
# ----------------------------------------
git status
git branch
git log --oneline --graph
git remote -v

# ----------------------------------------
# ❌ STEP 12: Undo Mistakes (MS Paint Undo)
# ----------------------------------------

# Undo file changes (not yet staged)
```

```
git restore filename.js

# Unstage a file (but keep changes)
git reset HEAD filename.js

# Undo last commit (keep code)
git reset --soft HEAD~1

# Undo last commit and unstage
git reset --mixed HEAD~1

# Danger: Undo last commit and delete code
# git reset --hard HEAD~1

# Undo pushed commit (DANGER)
# git reset --hard HEAD~1
# git push origin main --force

# Abort an in-progress merge
git merge --abort

# Stash uncommitted work
git stash
```

```
git stash pop


# Delete untracked files (be careful!)
# git clean -f


# ----------------------------------------
# 🔐 STEP 13: SSH Key Setup (If Needed)
# ----------------------------------------
# ssh-keygen -t rsa -b 4096 -C "you@example.com"
# cat ~/.ssh/id_rsa.pub
# Paste into GitHub → Settings → SSH → New Key


# ----------------------------------------
# 🎓 STEP 14: Interview Smart Extras
# ----------------------------------------


# Create + switch new branch
git checkout -b dev


# View commits clearly
git log --oneline --graph --all


# Revert a specific commit safely
```

```
git revert <commit-hash>


# Delete a branch (local)

git branch -d feature1


# Delete a branch (remote)

git push origin --delete feature1


# Rebase for cleaner history (advanced)

# git rebase main


# Push rebased branch (if already pushed earlier)

# git push --force-with-lease
```

---

2 📌 Key Notes (Embedded in the Script)

- ✅ Designed to run in PowerShell (standalone or VS Code terminal)
- 💡 Includes safety warnings where destructive actions are possible
- 🧠 Use comments to revise before interviews as well
