

1: Create a project folder

Open Git Bash and make a new folder:

```
mkdir harshith-project  
cd harshith-project
```

Now you're inside your new local project folder.

⚙️ Step 2: Initialize Git

```
git init
```

✅ This creates a hidden folder `.git` — that's where Git tracks everything you do (like a project's time machine 🕒).

Output:

```
Initialized empty Git repository in  
C:/Users/Harshith/Desktop/harshith-project/.git/
```

Now this folder becomes your **local repository**.

👤 Step 3: Configure your identity

Set your name and email — these appear in your commits:

```
git config --global user.name "HarshithK-froxcy"  
git config --global user.email "froxcy31@gmail.com"
```

Check if saved:

```
git config --list
```

This ensures your commits show *who* made the changes.

Step 4: Create a file

```
echo "This is my first Git project" > notes.txt
```

Check if it's created:

```
ls
```

Step 5: Check Git Status

```
git status
```

It shows:

```
Untracked files:
  notes.txt
```

Meaning — Git sees this file, but it's not being tracked yet.

Step 6: Add file to staging area

```
git add notes.txt
```

Now Git is *ready* to save that file in the next commit.

You can check again:

```
git status
```

Now it shows:

```
Changes to be committed:
```

```
new file: notes.txt
```

Step 7: Commit changes

```
git commit -m "Initial commit - added notes.txt"
```

✓ This permanently saves your current project version inside the Git history.

Step 8: View commit history

```
git log
```

You'll see something like:

```
commit 2a9f4c9c7d...
Author: HarshithK-froxcy <froxcy31@gmail.com>
Date:   Sun Oct 27 21:00:00 2025
```

```
Initial commit - added notes.txt
```

Each commit has a **unique ID** (you can go back to any of them later).

Step 9: Create a remote repo on GitHub

- Go to **GitHub** → **New Repository**
- Name it for example: `harshith-project`
- Don't initialize with README (since you already have local files)

Copy the HTTPS link (example):

```
https://github.com/HarshithK-froxcy/harshith-project.git
```

-

Step 10: Connect local to remote

Now link your local repo with GitHub:

```
git remote add origin  
https://github.com/HarshithK-froxcy/harshith-project.git
```


You can verify:

```
git remote -v
```

Step 11: Push code to GitHub

```
git push origin master
```

- `origin` = name of the remote (GitHub)
- `master` = your main branch

Now your project is on GitHub 

Check your repo online — you'll see `notes.txt`.

Step 12: Make some changes

Edit the file:

```
echo "Added one more line" >> notes.txt
```

Then check:

```
git status
```

It shows file modified.


Add + Commit again:

```
git add .  
git commit -m "Updated notes.txt with new line"  
git push origin master
```

Step 13: Clone a repository (for new system or teammate)

If you or your friend want to copy the repo:

```
git clone  
https://github.com/HarshithK-froxcy/harshith-project.git
```

 It downloads everything — commits, files, history, etc.

Step 14: Reset (Undo changes)

Let's say you made a mistake in the last commit — you can undo it.

Option 1 — Just unstage a file:

```
git reset notes.txt
```


It removes it from staging but doesn't delete the file.

Option 2 — Undo the last commit but keep changes:

```
git reset --soft HEAD~1
```

Option 3 — Undo the last commit completely (delete changes):

```
git reset --hard HEAD~1
```

 This helps when you want to roll back to a previous version.

Summary of Commands with Meaning

Command	Description
<code>git init</code>	Start a new local Git repository
<code>git config</code>	Set user info (name & email)
<code>git status</code>	Show current file states
<code>git add</code>	Stage files for commit
<code>git commit</code>	Save changes in local repo
<code>git log</code>	View commit history
<code>git remote add</code>	Link local repo to GitHub
<code>git push</code>	Upload commits to GitHub
<code>git clone</code>	Copy a repo from GitHub
<code>git reset</code>	Undo commits or unstages changes

Final Flow Recap:

```
mkdir project → cd project
git init
git config ...
echo "text" > file.txt
git add file.txt
git commit -m "message"
```

```
git remote add origin <GitHub link>
git push origin master
(edit file)
git add .
git commit -m "update"
git push origin master
git log
git reset --hard HEAD~1
```