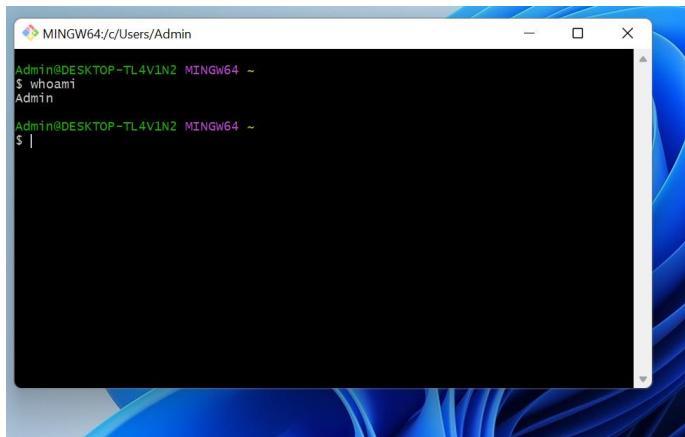
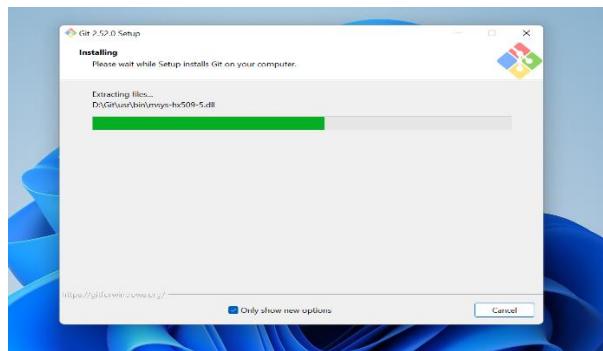
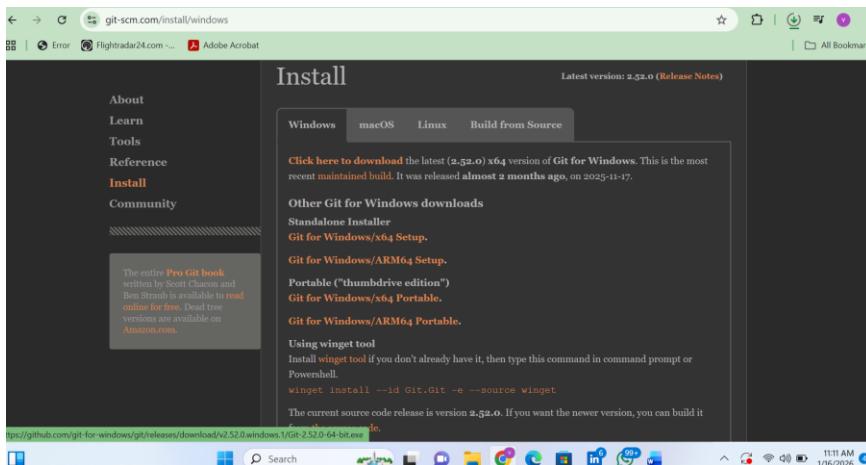


TASK 2: Git & GitHub Fundamentals

1. Install Git & GitHub Desktop

Install Git

- Download from: <https://git-scm.com>



2. Initialize a Local Git Repository

- Create a project directory

This is the command to create directory

“mkdir git-task2”

This command use for go into the created directory

“cd git-task2”

```
Design Layout References Mailings Review View Help 🔍 Tell me what you want to do Comments
vishnu@DESKTOP-TL4V1N2: ~$ git task2
[sudo] password for vishnu:
http://archive.ubuntu.com/ubuntu/nob... InRelease
http://archive.ubuntu.com/ubuntu/nob... InRelease [126 kB]
http://security.ubuntu.com/ubuntu/nob... InRelease [126 kB]
http://archive.ubuntu.com/ubuntu/nob... InRelease [126 kB]
http://archive.ubuntu.com/ubuntu/nob... InRelease [126 kB]
http://archive.ubuntu.com/ubuntu/nob... InRelease [1694 kB]
http://archive.ubuntu.com/ubuntu/nob... Translation-en [313 kB]
http://archive.ubuntu.com/ubuntu/nob... InRelease [175 kB]
http://archive.ubuntu.com/ubuntu/nob... InRelease [175 kB]
http://archive.ubuntu.com/ubuntu/nob... InRelease [1532 kB]
http://archive.ubuntu.com/ubuntu/nob... InRelease [377 kB]
http://archive.ubuntu.com/ubuntu/nob... InRelease [212 kB]
http://archive.ubuntu.com/ubuntu/nob... InRelease [212 kB]
http://archive.ubuntu.com/ubuntu/nob... InRelease [948 kB]
http://archive.ubuntu.com/ubuntu/nob... InRelease [7308 kB]
http://archive.ubuntu.com/ubuntu/nob... InRelease [10.5 kB]
http://archive.ubuntu.com/ubuntu/nob... InRelease [606 kB]
http://archive.ubuntu.com/ubuntu/nob... InRelease [212 kB]
http://security.ubuntu.com/ubuntu/nob... InRelease [21.5 kB]
http://security.ubuntu.com/ubuntu/nob... InRelease [71.5 kB]
http://security.ubuntu.com/ubuntu/nob... InRelease [212 kB]
http://security.ubuntu.com/ubuntu/nob... InRelease [1208 kB]
Building package lists... Done
Reading state information... Done
22 packages can be upgraded. Run 'apt list --upgradable' to see them.
vishnu@DESKTOP-TL4V1N2: ~$ git config --list
vishnu@DESKTOP-TL4V1N2: ~$ mkdir git-task2
vishnu@DESKTOP-TL4V1N2: ~$ cd git-task2/
vishnu@DESKTOP-TL4V1N2: ~$ git task2
vishnu@DESKTOP-TL4V1N2: ~$ git task2$
```

```
git clone https://github.com/vishnu123456789/task2.git  
Reading state information... done  
122 packages can be upgraded. Run 'apt list --upgradable' to see them.  
vishnu@DESKTOP-TL4V1N2:~$ cd task2  
vishnu@DESKTOP-TL4V1N2:~/task2$ git config --list  
vishnu@DESKTOP-TL4V1N2:~/task2$ git init  
vishnu@DESKTOP-TL4V1N2:~/task2$ git init  
hint: Using 'master' as the name for the initial branch. This default branch name  
hint: is subject to change. To configure the initial branch name to use in all  
hint: of your new repositories, which will suppress this warning, call:  
hint:   git config --global init.defaultBranch <name>  
hint:  
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and  
hint: 'development'. The just-created branch can be renamed via this command:  
hint:  
hint:   git branch -m <name>  
Initialized empty Git repository in /home/vishnu/git-task2/.git/  
vishnu@DESKTOP-TL4V1N2:~/task2$ ls -a  
.git  
vishnu@DESKTOP-TL4V1N2:~/task2$
```

```
vishnu@DESKTOP-TL4V1N2:~/git-demo$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/vishnu/git-demo/.git/
vishnu@DESKTOP-TL4V1N2:~/git-demo$ ls -a
.  .. .git
vishnu@DESKTOP-TL4V1N2:~/git-demo$ touch file1.txt
vishnu@DESKTOP-TL4V1N2:~/git-demo$ 
vishnu@DESKTOP-TL4V1N2:~/git-demo$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    file1.txt

nothing added to commit but untracked files present (use "git add" to track)
vishnu@DESKTOP-TL4V1N2:~/git-demo$
```

3. Create Files & Track Changes (status, add, commit)

- Create a file: touch file1.txt
- Check status: git status
 - ‘File will appear as Untracked’
- Add file to staging area: git add file1.txt
- Commit changes: git commit -m "Initial commit - added file1.txt"
- Modify file: echo "Hello Git" >> file1.txt
- Track modified file: git status
 - git add .
 - git commit -m "Updated file1.txt with content"

4. Create GitHub Repository & Link with Local Repo

Step 1: Create repository on GitHub

Go to GitHub → New Repository

Repository name: git-demo

Do NOT add README

Click Create repository

Step 2: Connect using HTTPS

git remote add origin <https://github.com/your-username/git-demo.git>

```
vishnu@DESKTOP-TLAVIN2:~/git-demo$ git remote -v
origin https://github.com/your-username/git-demo.git (fetch)
origin https://github.com/your-username/git-demo.git (push)
vishnu@DESKTOP-TLAVIN2:~/git-demo$ git remote remove origin
vishnu@DESKTOP-TLAVIN2:~/git-demo$ git remote remove origin
error: No such remote: 'origin'
vishnu@DESKTOP-TLAVIN2:~/git-demo$ git remote rm origin
error: No such remote: 'origin'
vishnu@DESKTOP-TLAVIN2:~/git-demo$ git remote -v
vishnu@DESKTOP-TLAVIN2:~/git-demo$ git remote add origin https://github.com/Vishnukumar777/git-demo.git
vishnu@DESKTOP-TLAVIN2:~/git-demo$ git remote -v
origin https://github.com/Vishnukumar777/git-demo.git (fetch)
origin https://github.com/Vishnukumar777/git-demo.git (push)
vishnu@DESKTOP-TLAVIN2:~/git-demo$ git branch -M main
```

Add key to GitHub (Correct Way)

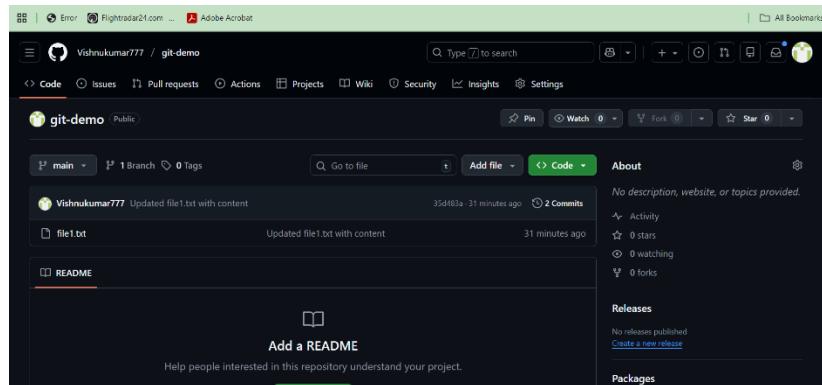
- Go to GitHub → Settings
 - Click SSH and GPG keys
 - Click New SSH key
 - Title: Laptop SSH Key
 - Paste the FULL output from step 2
 - Click Add SSH key

```
vishnu@vishnu-OptiPlex-5090:~$ ssh-keygen -t rsa -b 2048 -C "vishnuvps777@gmail.com"
Enter same passphrase again:
Your identification has been saved in /home/vishnu/.ssh/id_ed25519
Your public key has been saved in /home/vishnu/.ssh/id_ed25519.pub
The key's randomart image is
+--[Ed25519 256]--+
|          .+o.+
|         .+o.+
|        .+o.+
|       .+o.+
|      .+o.+
|     .+o.+
|    .+o.+
|   .+o.+
|  .+o.+
| .+o.+
+----+[SHA256]----+
[vishnu@vishnu-OptiPlex-5090:~]$ cat ~/.ssh/id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZnM0eXAQDwzqf7yJLm57Hf73jN5LGMxKzI0sI vishnuvps777@gmail.com
[vishnu@vishnu-OptiPlex-5090:~]$ ssh -T git@github.com
The authenticity of host "github.com (20.97.73.82)" cannot be established.
ED25519 key fingerprint is SHA256:+DIYavw5WU7UJhpQzlsfJzD4ZpMSHdk+4UvCoQ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Please enter your GitHub password: 
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
[vishnu@vishnu-OptiPlex-5090:~]$ git remote add origin git@github.com:vishnuvps777/git-demo.git
[vishnu@vishnu-OptiPlex-5090:~]$ git push origin main
Enumerating objects: 6, done
Counting objects: 100% (6/6), done.
Delta compression using up to 2 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 471 bytes | 78.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:vishnuvps777/git-demo.git
 ! [new branch] main > main
[vishnu@vishnu-OptiPlex-5090:~]$ git remote -v
origin git@github.com:vishnuvps777/git-demo.git (fetch)
origin git@github.com:vishnuvps777/git-demo.git (push)
[vishnu@vishnu-OptiPlex-5090:~]$
```

5. Push local commits to remote GitHub repository.

This are the commands to push local to remote github repo

- git branch -M main
 - git push -u origin main



This is the outcome when the file go to github repo

6. Create a New Branch, Make Changes & Merge

- Create and switch to new branch: git checkout -b feature-branch

```
vishnu@DESKTOP-TL4V1N2:~/git-demo$ git checkout -b feature-branch
Switched to a new branch 'feature-branch'
vishnu@DESKTOP-TL4V1N2:~/git-demo$ echo "Feature branch change" >> file1.txt
vishnu@DESKTOP-TL4V1N2:~/git-demo$ git add .
vishnu@DESKTOP-TL4V1N2:~/git-demo$ git commit -m "Added feature branch changes"
[feature-branch e68a927] Added feature branch changes
1 file changed, 1 insertion(+)
vishnu@DESKTOP-TL4V1N2:~/git-demo$
```

- Make changes: echo "Feature branch change" >> file1.txt

```
git add .
```

```
git commit -m "Added feature branch changes"
```

- Switch back to main: git checkout main

```
vishnu@DESKTOP-TL4V1N2:~/git-demo$ git checkout main
Switched to branch 'main'
vishnu@DESKTOP-TL4V1N2:~/git-demo$
```

- Merge branch: git merge feature-branch

```
vishnu@DESKTOP-TL4V1N2:~/git-demo$ git merge feature-branch
Switched to branch 'main'
Updating 35d483a..e68a927
Fast-forward
 file1.txt | 1 +
 1 file changed, 1 insertion(+)
vishnu@DESKTOP-TL4V1N2:~/git-demo$
```

7. Practice rollback using git log, git reset, and git checkout.

- View commit history: git log --oneline

```
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Writing objects: 100% (3/3), 290 bytes | 290.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:Vishnukumar777/git-demo.git
 35d483a..e68a927 main -> main
vishnu@DESKTOP-TL4V1N2:~/git-demo$ git log --oneline --all --decorate
e68a927 (HEAD -> main, origin/main, feature-branch) Added feature branch changes
35d483a Updated file1.txt with content
3286601 Initial commit - added file1.txt
vishnu@DESKTOP-TL4V1N2:~/git-demo$ git log --oneline
e68a927 (HEAD -> main, origin/main, feature-branch) Added feature branch changes
35d483a Updated file1.txt with content
3286601 Initial commit - added file1.txt
vishnu@DESKTOP-TL4V1N2:~/git-demo$ git log --oneline
e68a927 (HEAD -> main, origin/main, feature-branch) Added feature branch changes
35d483a Updated file1.txt with content
3286601 Initial commit - added file1.txt
vishnu@DESKTOP-TL4V1N2:~/git-demo$ git log --oneline --all --decorate
e68a927 (HEAD -> main, origin/main, feature-branch) Added feature branch changes
35d483a Updated file1.txt with content
3286601 Initial commit - added file1.txt
vishnu@DESKTOP-TL4V1N2:~/git-demo$ git push origin main
git reset --soft <commit-id>
```

