

Lab Assignment 1: Introduction to Version Controlling, Git Workflows, and Actions

INTRODUCTION, SETUP, AND TOOLS

Introduction

The purpose of this lab is to make us familiarize with the basic concepts of version control systems (VCS), understand its importance in software development and testing, and get hands-on experience with Git. We get to learn for this lab and were able to Set up and configure Git on our systems and perform basic Git operations such as initializing a repository, adding files, committing changes, and viewing the commit history on both local and remote repositories on GitHub. Lastly we understand the purpose and benefits of using version control systems.

Setup

- **Git Installation:** we need to install Git on the local machine before starting the lab from [Git Downloads \[1\]](#).
- **GitHub Account:** We need to have GitHub at Account.
- **Code Editor:** Visual Studio Code (VS Code) is preferred for code editing.

Tools

- **Git:** A distributed version control system that will keep track of changes in our code.
- **GitHub:** A platform that will host Git repositories and allow remote collaboration.
- **Pylint (via GitHub Actions):** A static code analysis tool that checks for errors in Python code.

METHODOLOGY AND EXECUTION WITH RESULT AND ANALYSIS

Git Basics

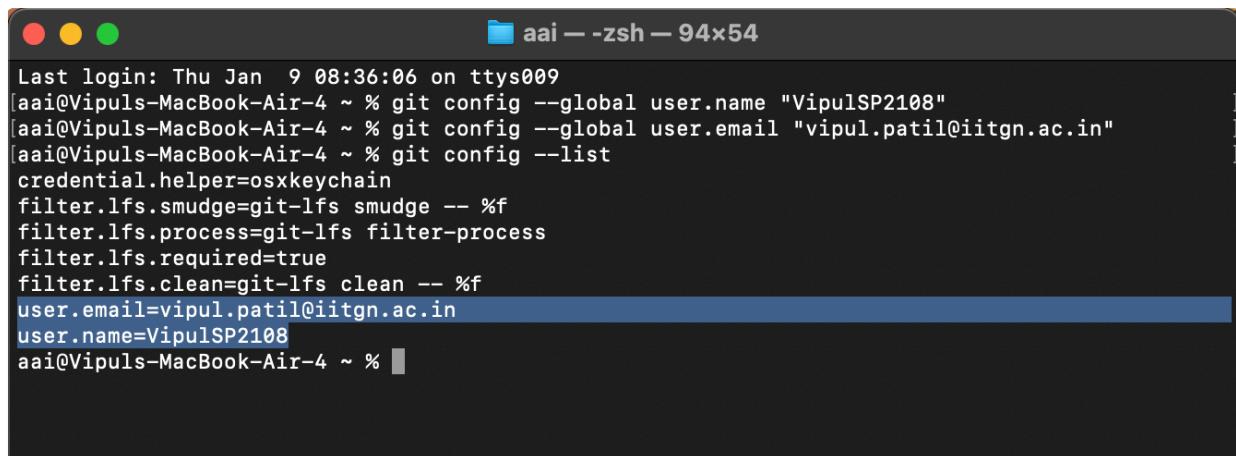
Setting up Git:

1. Configure Git with name and email with

```
git config --global user.name "VipulSP2108"  
git config --global user.email "vipul.patil@iitgn.ac.in"
```

2. Next is to verify configuration with

```
git config --list
```

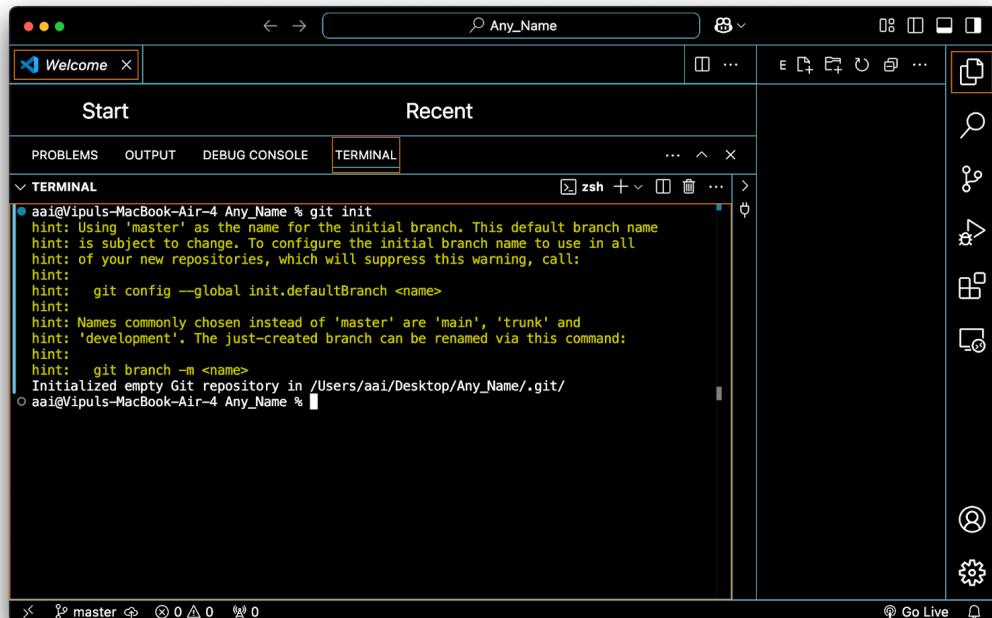


```
Last login: Thu Jan  9 08:36:06 on ttys009
[aai@Vipuls-MacBook-Air-4 ~ % git config --global user.name "VipulSP2108"
[aai@Vipuls-MacBook-Air-4 ~ % git config --global user.email "vipul.patil@iitgn.ac.in"
[aai@Vipuls-MacBook-Air-4 ~ % git config --list
credential.helper=osxkeychain
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
filter.lfs.clean=git-lfs clean -- %f
user.email=vipul.patil@iitgn.ac.in
user.name=VipulSP2108
aai@Vipuls-MacBook-Air-4 ~ %
```

Initializing a Local Repository:

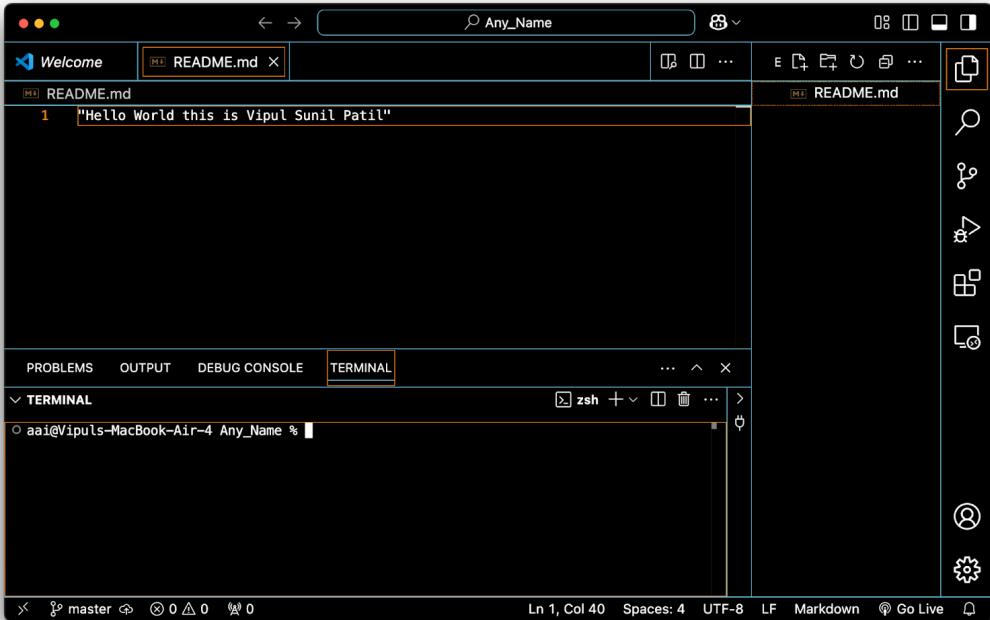
1. Created a new folder (with name "Any_Name")
2. open it to VS code (open VS code and in open find the folder you created)
3. and Initialize a Git repository with

```
git init
```



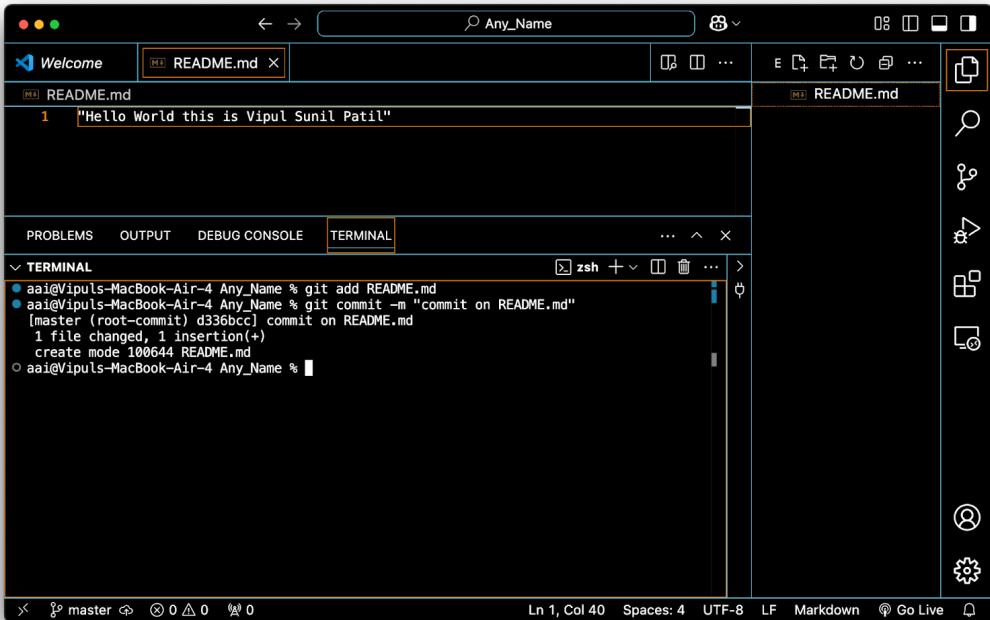
Adding and Committing Files:

1. Creating a new file README.md by clicking create file option in explorer of VS code and adding some content.

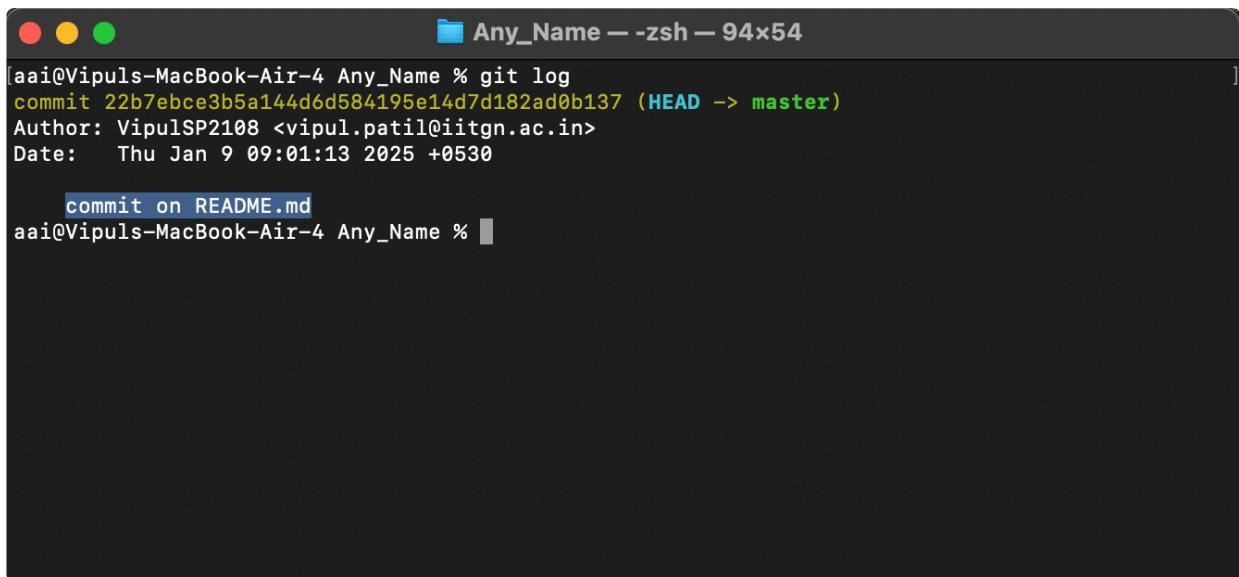


2. Adding the file to the staging area and Commiting the file to local repo with some message.

```
git add README.md  
git commit -m "commit on README.md"
```



Viewing the Commit History:



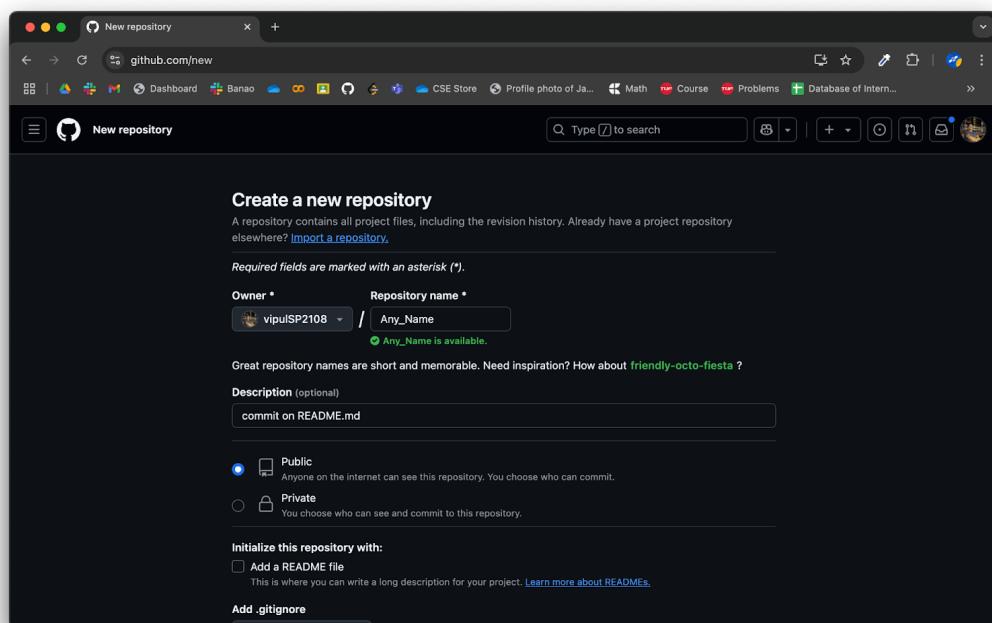
```
[aai@Vipuls-MacBook-Air-4 Any_Name % git log
commit 22b7ebce3b5a144d6d584195e14d7d182ad0b137 (HEAD -> master)
Author: VipulSP2108 <vipul.patil@iitgn.ac.in>
Date:   Thu Jan 9 09:01:13 2025 +0530

    commit on README.md
aai@Vipuls-MacBook-Air-4 Any_Name % ]
```

Working with Remote Repositories

Connecting to GitHub:

1. Creating a new repository on GitHub named "Any_Name".



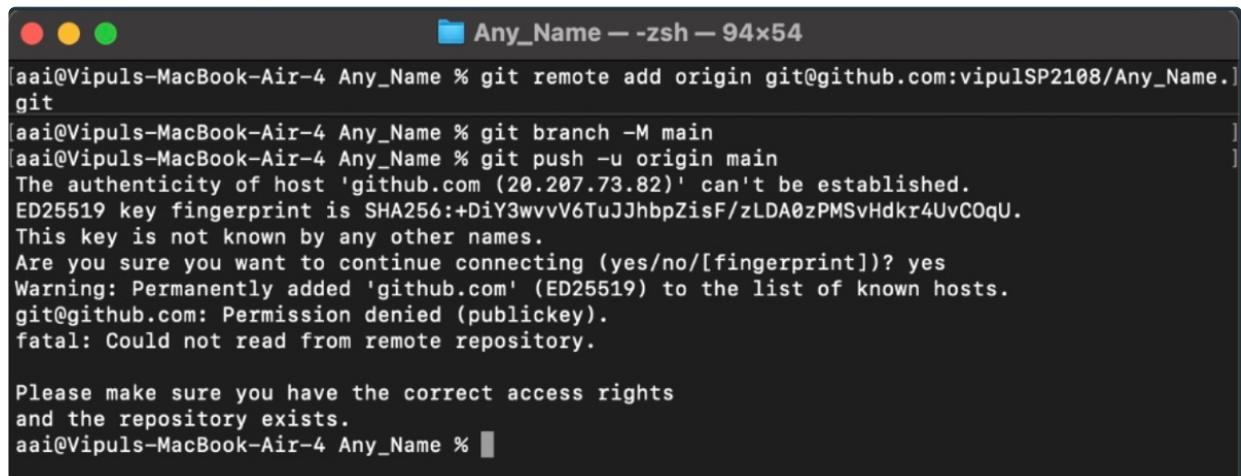
2. Link the local repository to GitHub using the same command that GitHub showed after creating a repository (**use your_githubID/your_repoName instead of vipulSP2108/Any_Name**)

```
git remote add origin git@github.com:vipulSP2108/Any_Name.git
```

Pushing Changes to GitHub:

1. Pushing the committed changes to GitHub using.

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

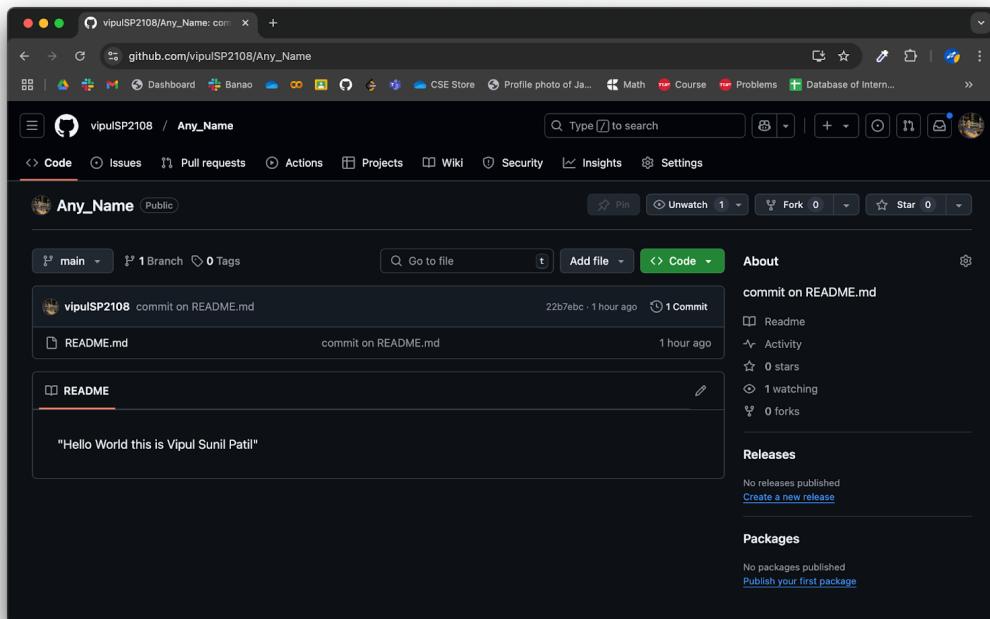


```
[aai@Vipuls-MacBook-Air-4 Any_Name % git remote add origin git@github.com:vipulSP2108/Any_Name.  
git  
[aai@Vipuls-MacBook-Air-4 Any_Name % git branch -M main  
[aai@Vipuls-MacBook-Air-4 Any_Name % git push -u origin main  
The authenticity of host 'github.com (20.207.73.82)' can't be established.  
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbhpZisF/zLDA0zPMSvHdkr4UvC0qU.  
This key is not known by any other names.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.  
git@github.com: Permission denied (publickey).  
fatal: Could not read from remote repository.  
  
Please make sure you have the correct access rights  
and the repository exists.  
aai@Vipuls-MacBook-Air-4 Any_Name % ]
```

Next I was getting a error message that indicates that Git is unable to authenticate you with GitHub when trying to push to the repository.

(Host Authenticity Warning and Permission Denied (publickey))

2. this is Reflection of Readme.md file added on Github



Cloning a Repository:

1. to create a copy this repository to your local machine (Cloning the Repository) use

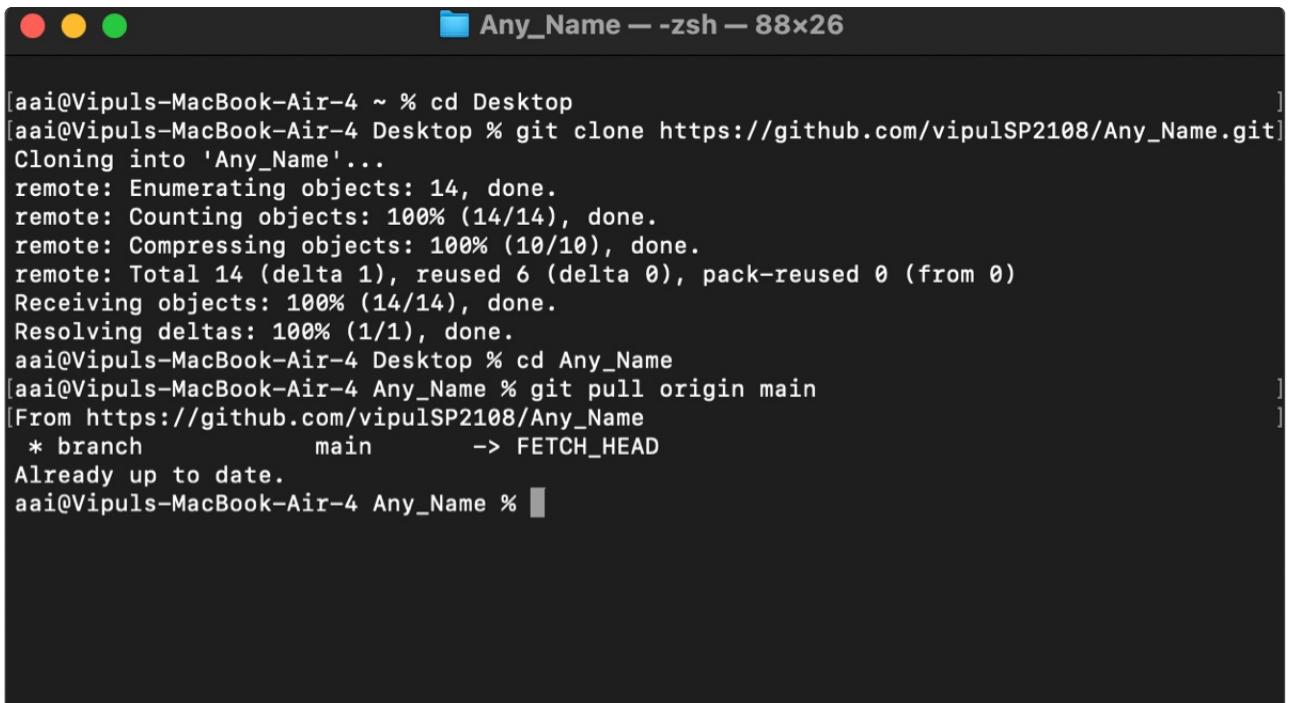
```
git clone https://github.com/vipulSP2108/Any_Name.git
```

Pulling changes:

- for integrating changes from the remote repository to your local repository we use after going into that **specific** folder.

```
git pull origin main
```

This is Reflection of Cloning and Pulling changes from Remote Repository to local Repository



A screenshot of a terminal window titled "Any_Name — zsh — 88x26". The window shows the following command history:

```
[aa@Vipuls-MacBook-Air-4 ~ % cd Desktop
[aa@Vipuls-MacBook-Air-4 Desktop % git clone https://github.com/vipulSP2108/Any_Name.git
Cloning into 'Any_Name'...
remote: Enumerating objects: 14, done.
remote: Counting objects: 100% (14/14), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 14 (delta 1), reused 6 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (14/14), done.
Resolving deltas: 100% (1/1), done.
aa@Vipuls-MacBook-Air-4 Desktop % cd Any_Name
[aa@Vipuls-MacBook-Air-4 Any_Name % git pull origin main
[From https://github.com/vipulSP2108/Any_Name
 * branch      main      -> FETCH_HEAD
Already up to date.
aa@Vipuls-MacBook-Air-4 Any_Name % ]
```

Setuping a pylint workflow (via GitHub Actions), and committing our own code (>=30 lines of Python3) and resolving it until all errors are solved and a green tick () appears.

Added main.py file and committed it on Github.

```

1 def factorial(n):
2     if n == 0 or n == 1:
3         return 1
4     else:
5         return n * factorial(n - 1)
6
7 num = int(input("Enter a number: "))
8 print(f"The factorial of {num} is {factorial(num)}")

```

TERMINAL

```

aai@ipuls-MacBook-Air-4 Any_Name % git add main.py
aai@ipuls-MacBook-Air-4 Any_Name % git commit -m "commit on main.py --code added"
[main 9cab552] commit on main.py --code added
1 file changed, 8 insertions(+)
create mode 100644 main.py
aai@ipuls-MacBook-Air-4 Any_Name % git log
zsh: command not found: gitlog
aai@ipuls-MacBook-Air-4 Any_Name % git log
commit 9cab5529da6cd446960e0352e672ba8d672fb8 (HEAD -> main)
Author: VipulSP2108 <vipul.patil@iitgn.ac.in>
Date: Thu Jan 9 10:31:16 2025 +0530

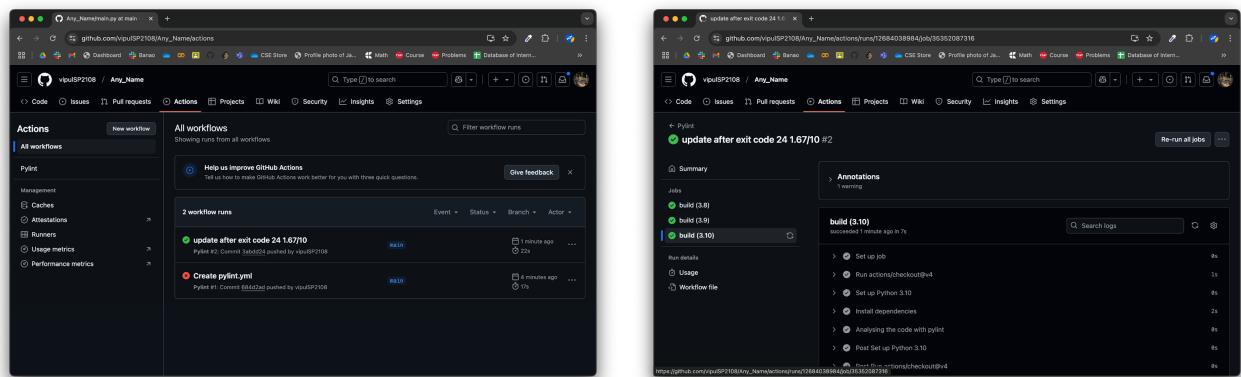
    commit on main.py --code added

commit 22b7ebce3b5a144d6d584195e14d7d182ad0b137 (origin/main)
Author: VipulSP2108 <vipul.patil@iitgn.ac.in>

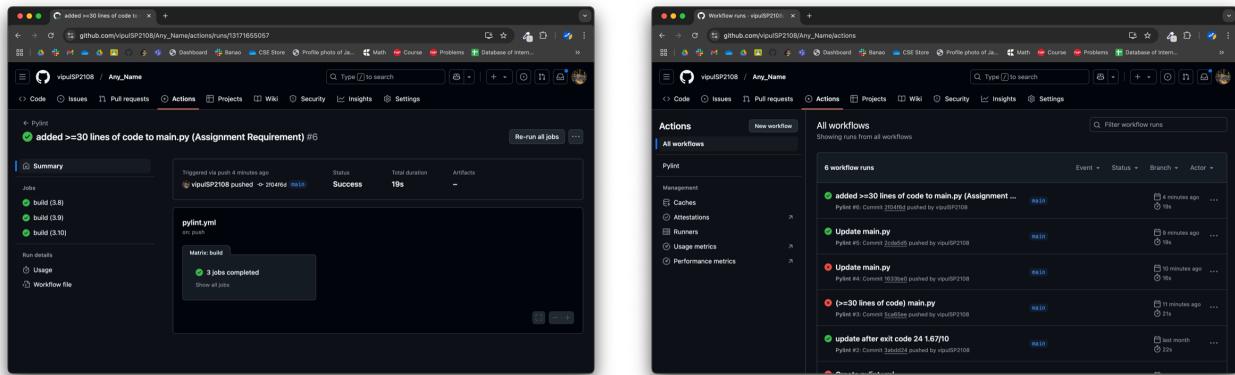
```

To Add Pylint workflow and workflow runs:

1. Open your repository on GitHub and Navigate to the **Actions** tab in the top navbar.
2. Search for **Pylint** in the Actions search bar and Click on **Configure**.
3. GitHub will automatically generate the necessary Pylint configuration with the required code or dependencies.
4. Just click **Commit changes** to add the workflow.
5. This will create a `.github/workflows` folder in your repository.
6. Go back to the **Actions** tab and click on **Workflow runs**.
7. resolve your code untill all errors are solved and a green tick () appears.



I wasn't aware that we needed to have a code of 30 or more lines for checking in Pylink, so I updated it. following screenshots are after runing pylint on new changes.



My Github Repo link: https://github.com/vipulSP2108/Any_Name

↑ Lab Assignment 1: Introduction to Version Controlling, Git Workflows, and Actions

Next I was getting a error message that indicates that Git is unable to authenticate you with GitHub when trying to push to the repository.

(Host Authenticity Warning and Permission Denied (publickey))

I tried this commad from [stackoverflow](#) [2]

```
ssh-keygen -t ed25519 -C "vipul.patil@iitgn.ac.in"
```

1. It will ask for a file name and passphrase.
2. This will create both `file_name` and `file_name.pub` files. (In my case, `file_name` is `git_sshKeyGenerate`).
3. Either:
 - o Use this command to see the key, or (**use your_fileName.pub insterd of git_sshKeyGenrate.pub**)

```
cd git_sshKeyGenrate.pub
cat git_sshKeyGenrate.pub
```

- o Open the folder in VS Code, and in `file_name.pub`, you will find the key.
4. After generating the SSH key
 1. Either copy the key and

1. Go to GitHub or either:
 - Click on your profile photo in the top-right corner, and select **Settings**.
 - Then search for **SSH and GPG keys** in the left-hand menu,
 - Next, click **New SSH key** under the **SSH keys** section (at the top).

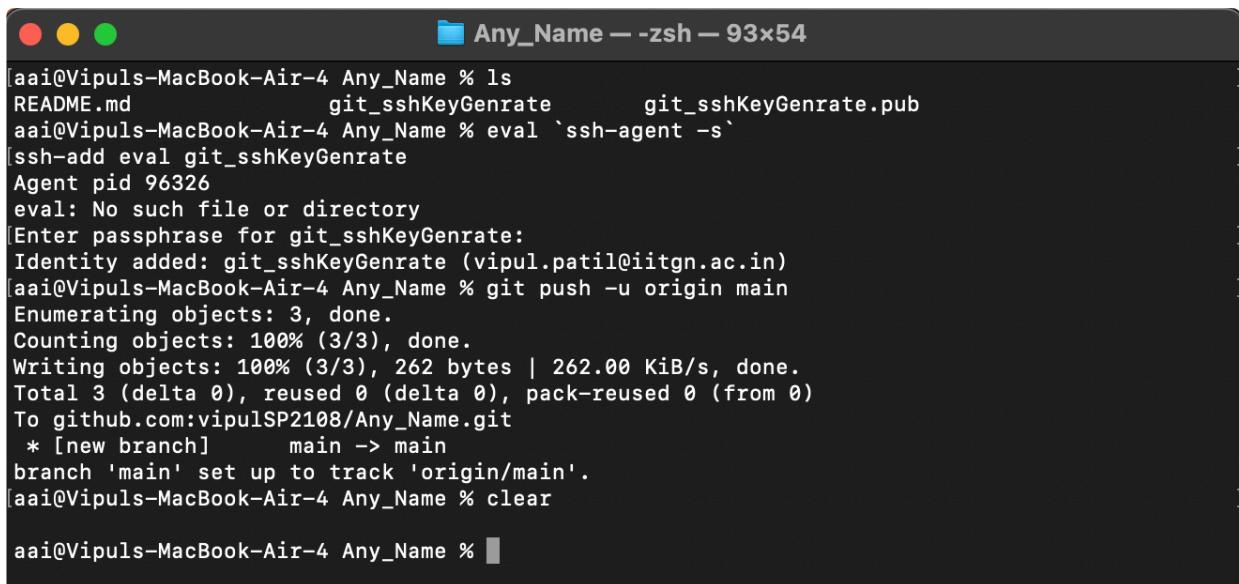
OR

1. Directly visit <https://github.com/settings/ssh/new>.
2. And paste your full SSH key in the provided field. Formated like:
`<start> <your-key> <your_emailid>`

OR

1. in your main folder use this command (as you are in `file_name.pub` first use `cd..`)

```
eval `ssh-agent -s` ssh-add git_sshKeyGenrate
```



```
[aaipuls-MacBook-Air-4 Any_Name % ls
 README.md          git_sshKeyGenrate      git_sshKeyGenrate.pub
 aaipuls-MacBook-Air-4 Any_Name % eval `ssh-agent -s`
 [ssh-add eval git_sshKeyGenrate
 Agent pid 96326
 eval: No such file or directory
 [Enter passphrase for git_sshKeyGenrate:
 Identity added: git_sshKeyGenrate (vipul.patil@iitgn.ac.in)
 aaipuls-MacBook-Air-4 Any_Name % git push -u origin main
 Enumerating objects: 3, done.
 Counting objects: 100% (3/3), done.
 Writing objects: 100% (3/3), 262 bytes | 262.00 KiB/s, done.
 Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
 To github.com:vipulSP2108/Any_Name.git
 * [new branch]    main -> main
 branch 'main' set up to track 'origin/main'.
 aaipuls-MacBook-Air-4 Any_Name % clear
 aaipuls-MacBook-Air-4 Any_Name % ]
```

DISCUSSION AND CONCLUSION

Discussion

From this lab we get to understand the importance of version control in software development and why it is used so widely also we Perform essential Git operations such as initializing repositories, adding files, committing changes, and pushing to remote repositories. Next we learnt how to configure Git and GitHub on their local systems and lastly we Set up and use GitHub Actions for automatic code linting with Pylint and ensuring quality checks of code.

This lab gave a comprehensive understanding of Git and GitHub workflows. By following the steps to initialize a repository, add files, commit changes, and push them to a remote repository and gained practical knowledge of version control. The integration of GitHub Actions to check code quality through Pylint emphasized the importance of continuous integration and code quality in collaborative projects.

The setup process for SSH keys provided an understanding of securing Git operations through SSH, which is vital when working with private repositories. This also highlighted the importance of correct authentication when working with remote repositories.

Conclusion

- Understanding the importance of version control in software development.
- Configure Git and GitHub on our local systems.
- Perform essential Git operations such as initializing repositories, adding files, committing changes, and pushing to remote repositories.
- Set up and use GitHub Actions for automatic code linting with Pylint, ensuring quality checks on code.

APPENDIX

I would like to express my sincere gratitude to my course instructor, Prof. [**Shouwick Mondal**](#), for his invaluable guidance and support throughout this lab. I also appreciate the assistance from All TAs, whose help with me with troubleshooting.

Additionally, I am grateful for the resources provided, including Which I have sited down which helped me resolve issues efficiently. Finally, I'd like to thank my peers for contributing to a collaborative and supportive learning environment.

CITATION

[1] Get started with GitHub documentation - GitHub Docs. (n.d.). GitHub Docs. <https://docs.github.com/en/get-started>

[2] GitHub Error Message - Permission denied (publickey). (n.d.). Stack Overflow. <https://stackoverflow.com/questions/12940626/github-error-message-permission-denied-publickey>

[3] Class Slides - <https://drive.google.com/file/d/16tTzLdotE0VJyIM cwdUcuMOAn8-Yo0at/view?usp=sharing>

[4] Lab Manual - https://drive.google.com/file/d/1BpmEwbM1t-utv15Sh0abBXdI6IN88gXR/view?usp=classroom_web&authuser=0

[5] Video Demonstration- https://iitgnacin-my.sharepoint.com/personal/shouwick_mondal_iitgn_ac_in/_layouts/15/stream.aspx?id=%2Fpersonal%2Fshouwick%5Fmondal%5Fiitgn%5Fac%5Fin%2FDocuments%2FSTT%5FCSE%5FLec%5F1%5Fhands%5Fon%2Ewebm&nav=eyJwbGF5YmFja09wdGlvbMiOnt9LCJyZWZlcnJhbEluZm8iOnsicmVmZXJyYWxBcHAIoIJTdHJIYw1XZWJBcHAIcJyZWZlcnJhbE1vZGUiOjtaXMiLCJyZWZlcnJhbFZpZXciOjwb3N0cm9sbC1jb3B5bGluaylsInJlZmVycmFsUGxheWJhY2tTZXNzaW9uSWQiOjJhOTI1NzlkMC1kNGlwLTRiMDUtYTdkNS02ODk3ZTU2OTUxNzlifX0&ga=1&referrer=StreamWebApp%2EWeb&referrerScenario=AddressBarCopied%2Eview%2E4df511d4%2De250%2D434d%2Db0a7%2Dbac8a687b6de