



EXPERIMENT 3

Managing Source Code on
GitHub using Git



Prepared by

NEEL VAGHASIYA

Semester - 7



BTech in Information Technology
Batch year: 2021-2025

 neelvaghasiya003@gmail.com



TABLE OF CONTENTS



01

Introduction to
GitHub.



02

Creating
Remote Repository
&
Managing Codebase .



03

Collaborating and
Contributing on
GitHub .



01 INTRODUCTION TO GITHUB

01 INTRODUCTION TO GITHUB



What is GitHub?

- Web-based platform that is used for version control and collaborative software development.
- Hosts Git repositories.

Features:

- Repository hosting
- Version Control
- Easy collaboration
- CI/CD using GitHub Actions

01 INTRODUCTION TO GITHUB



Git vs GitHub

 Git	 GitHub
1 Locally installed software	1 Web-based platform
2 Tool to manage different versions of codebase	2 Space to host git repositories
3 Does not require an internet connection	3 Requires an internet connection
4 Provides version control features (Branching, Merging, Committing)	4 Extends Git's functionalities and adds collaboration tools (Pull requests, Issue tracking, Project management)

01 INTRODUCTION TO GITHUB



Examples of some popular Git Hosting Platforms / Services



GitHub



GitLab



Bitbucket



SourceForge



Launchpad



AWS CodeCommit



Azure DevOps



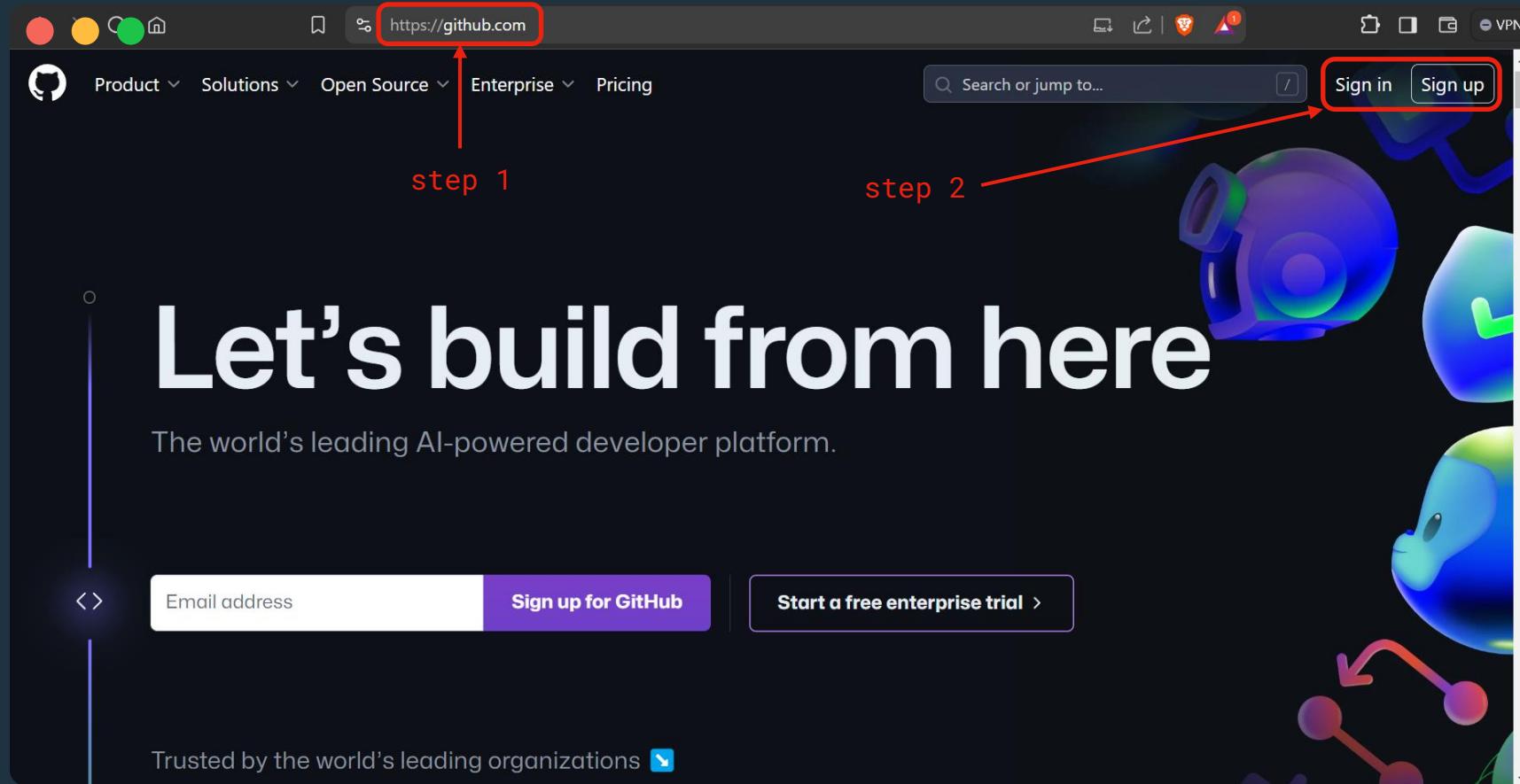
Cloud Source Repositories



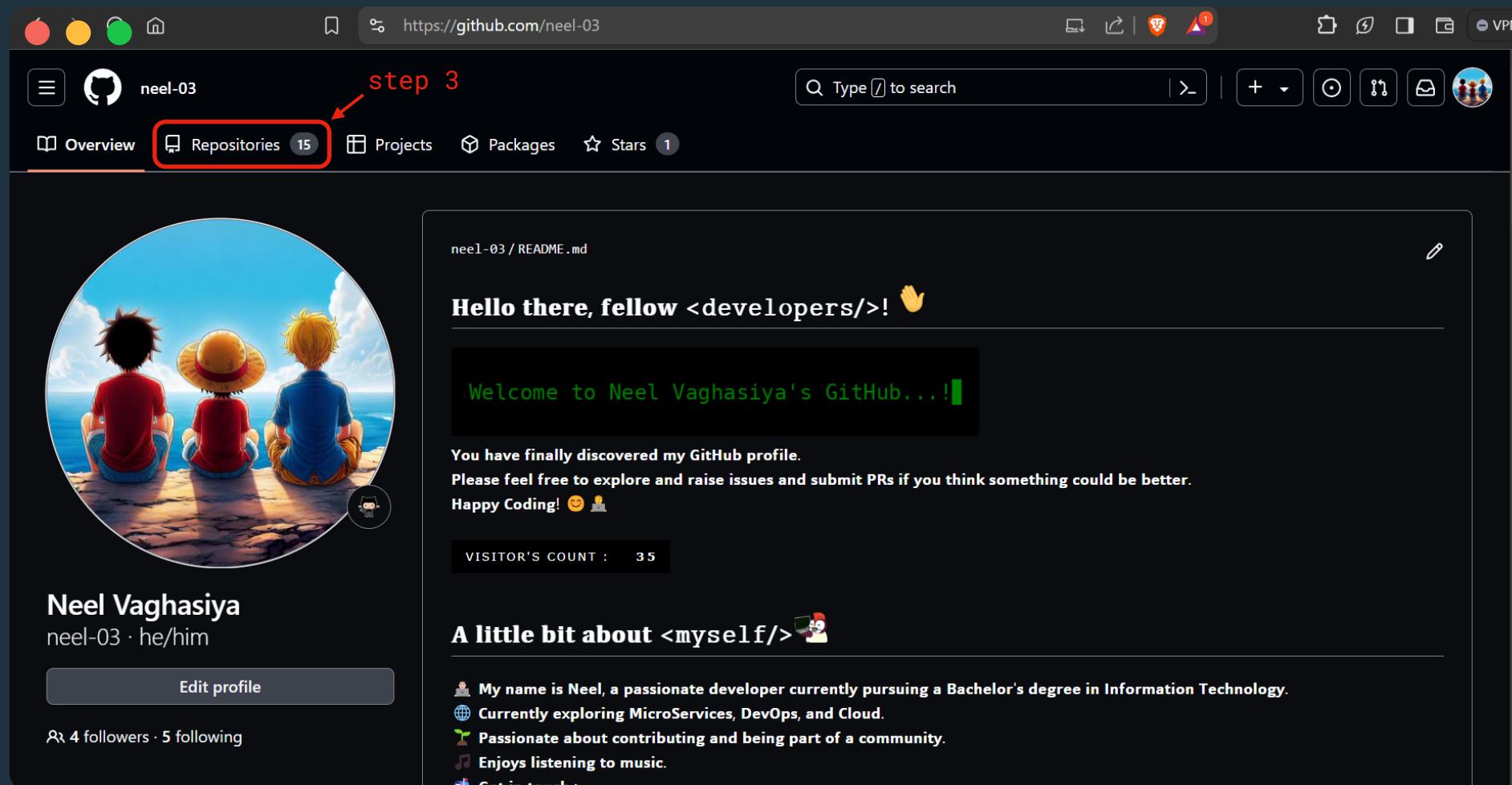
02

CREATING REMOTE REPOSITORY & MANAGING CODEBASE

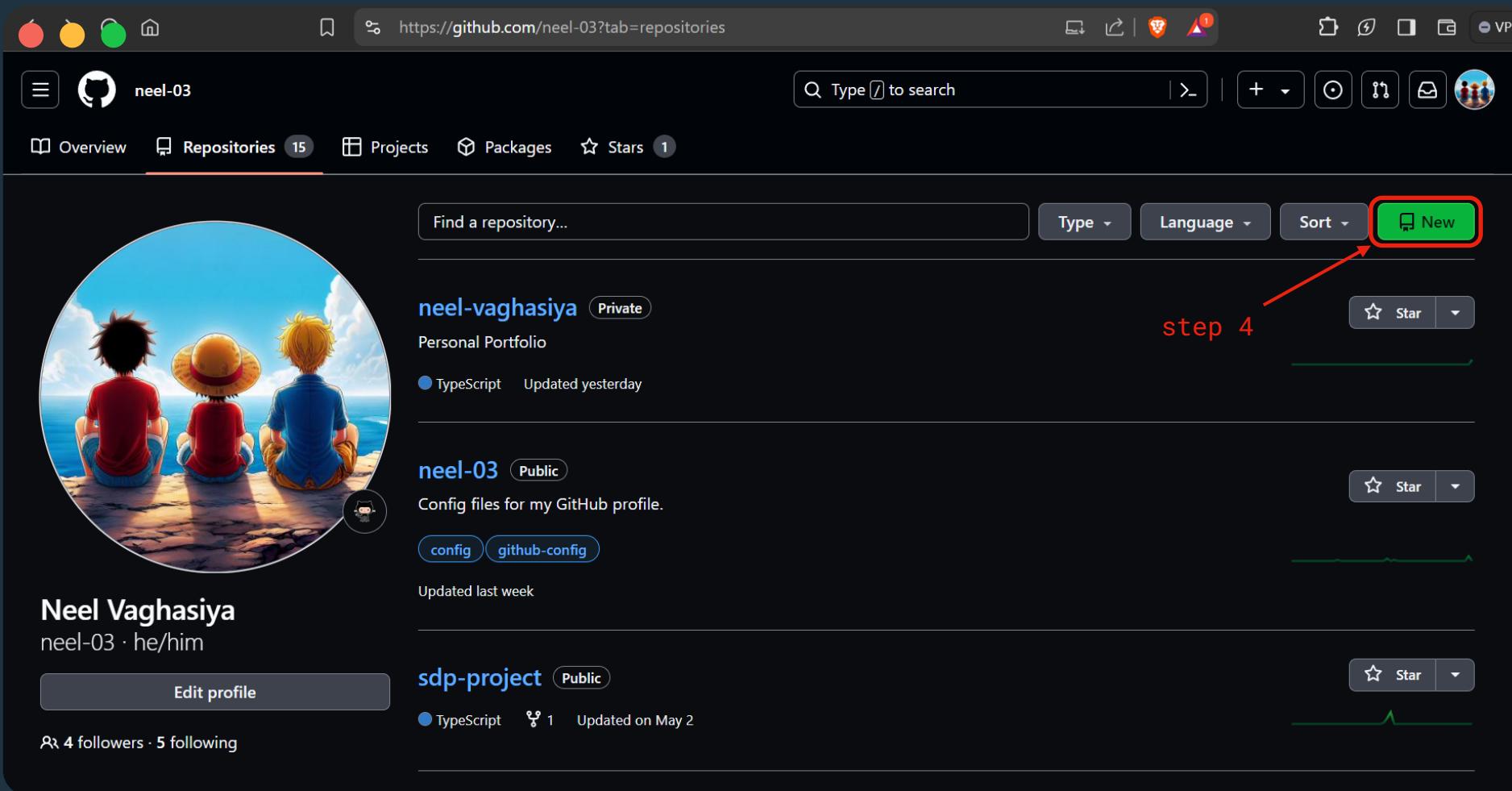
- STEP 1 : Visit <https://github.com>
- STEP 2 : Create a new account or sign in into existing one.



- STEP 3 : In Profile page, click on Repositories.



- STEP 4 : In Repositories tab, click on New button to create empty remote repository.



- STEP 5 : Fill the necessary details and click on Create repository button.

New repository

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Required fields are marked with an asterisk (*).

Owner *	Repository name *
 neel-03	/ exp-3  exp-3 is available.

Great repository names are short and memorable. Need inspiration? How about [special-bassoon](#) ?

Description (optional)

Working with remote repo

Select the repository owner and specify the unique repository name

Select visibility of repository

New repository

Public Anyone on the internet can see this repository. You choose who can commit.

Private You choose who can see and commit to this repository.

Initialize this repository with:

Add a README file This is where you can write a long description for your project. [Learn more about READMEs](#).

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. [Learn more about ignoring files](#).

Choose a license

License: None

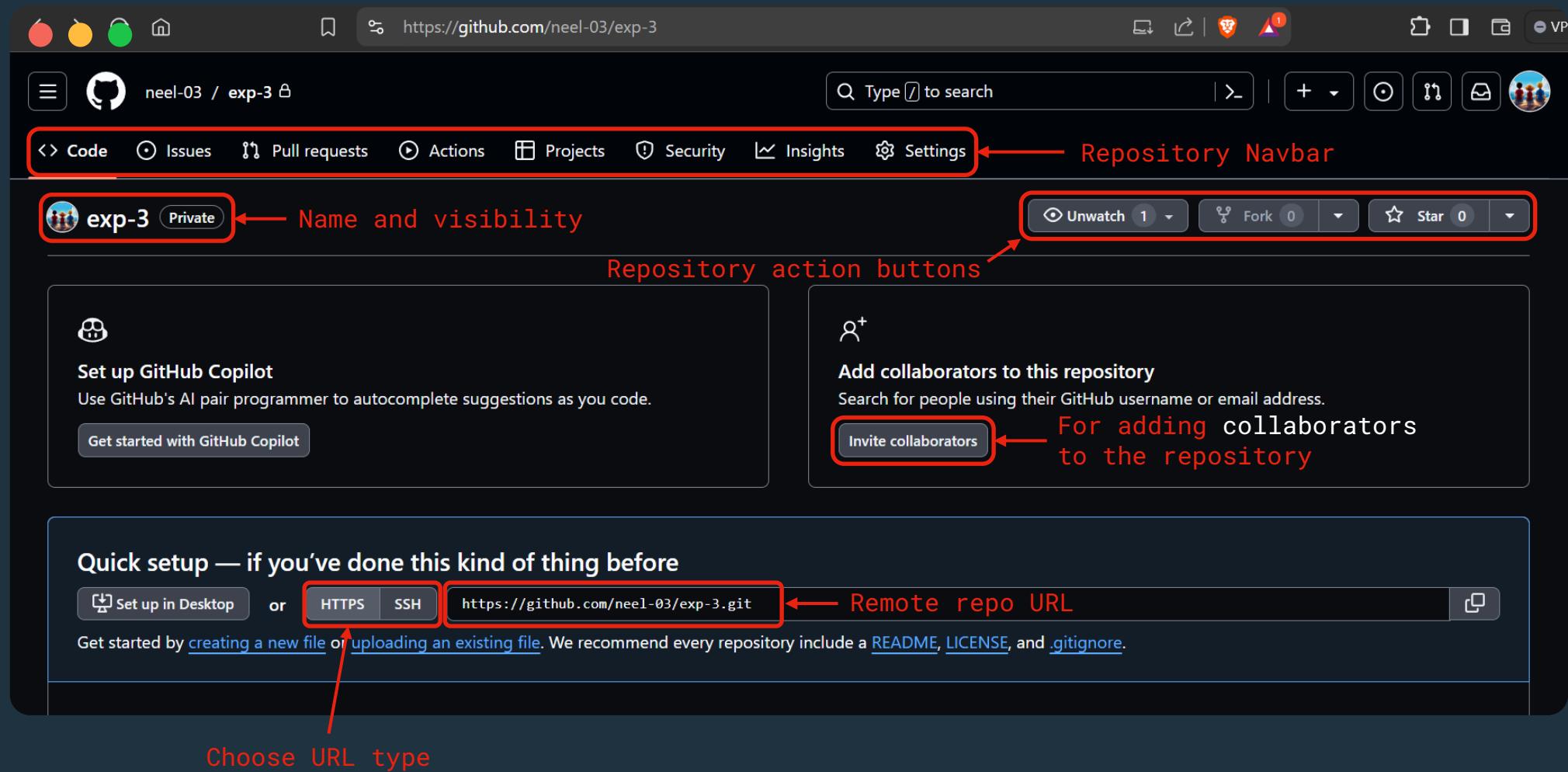
A license tells others what they can and can't do with your code. [Learn more about licenses](#).

 You are creating a private repository in your personal account.

Click here to create repository

Create repository

- After clicking the button, we will be directed to below page.
- Let's explore this page:



- Below that we can see the list of commands. Let's understand:

The screenshot shows a GitHub repository setup page for a repository named 'exp-3'. The URL in the address bar is <https://github.com/neel-03/exp-3>. The page provides options to 'Set up in Desktop' or 'HTTPS / SSH' and includes a link to the repository's URL (<https://github.com/neel-03/exp-3.git>). A note at the top says: 'Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#)'.

...or create a new repository on the command line

```
echo "# exp-3" >> README.md  
git init  
git add README.md  
git commit -m "first commit"  
git branch -M main  
git remote add origin https://github.com/neel-03/exp-3.git  
git push -u origin main
```

Annotations explain the steps:

- `git init` → Initialize Git
- `git add README.md` → (Optional) Add Readme or any other file and commit local changes
- `git commit -m "first commit"` → (Optional) Rename branch from master to main
- `git remote add origin https://github.com/neel-03/exp-3.git` → Add remote and push code to the remote repository

...or push an existing repository from the command line

```
git remote add origin https://github.com/neel-03/exp-3.git  
git branch -M main  
git push -u origin main
```

ProTip! Use the URL for this page when adding GitHub as a remote.

- Let's push the sample project we created in the previous experiment.

- Open Git Bash in project directory.
- Check status
- (optional) Rename master branch
- Set remote
- Push the code

```
MINGW4:/c/Users/neel/Desktop/exp-2
neel@NeelVaghasiya MINGW64 ~/Desktop/exp-2 (master)
$ git status
On branch master
nothing to commit, working tree clean

neel@NeelVaghasiya MINGW64 ~/Desktop/exp-2 (master)
$ git branch -M main
neel@NeelVaghasiya MINGW64 ~/Desktop/exp-2 (main)
$ git branch
bugfix/hero-section
feature/footer
feature/hero-section
feature/nav-options
feature/navbar
* main

neel@NeelVaghasiya MINGW64 ~/Desktop/exp-2 (main)
$
```

```
MINGW4:/c/Users/neel/Desktop/exp-2
neel@NeelVaghasiya MINGW64 ~/Desktop/exp-2 (main)
$ git remote add origin https://github.com/neel-03/exp-3.git
neel@NeelVaghasiya MINGW64 ~/Desktop/exp-2 (main)
$ git remote -v
origin https://github.com/neel-03/exp-3.git (fetch)
origin https://github.com/neel-03/exp-3.git (push)

neel@NeelVaghasiya MINGW64 ~/Desktop/exp-2 (main)
$ |
```

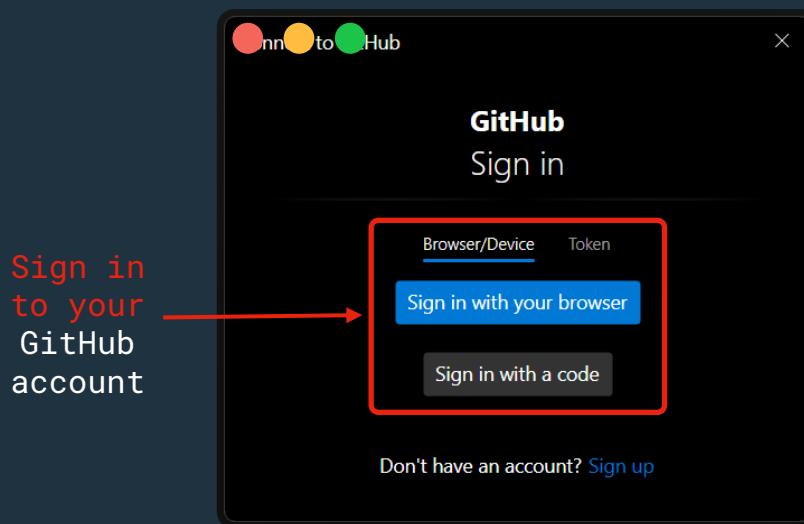
Compare this with syntax from the previous experiment

List all the remote



```
MINGW64:/c/Users/neel/Desktop/exp-2
neel@NeelVaghasiya MINGW64 ~/Desktop/exp-2 (main)
$ git push origin main
```

- On the first push to the remote repo, Git will prompt you to sign in



After sign in

```
MINGW64:/c/Users/neel/Desktop/exp-2
neel@NeelVaghasiya MINGW64 ~/Desktop/exp-2 (main)
$ git push origin main
Enumerating objects: 44, done.
Counting objects: 100% (44/44), done.
Delta compression using up to 8 threads
Compressing objects: 100% (43/43), done.
Writing objects: 100% (44/44), 39.98 KiB | 1.29 MiB/s, done.
Total 44 (delta 16), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (16/16), done.
To https://github.com/neel-03/exp-3.git
 * [new branch] main -> main
```

```
neel@NeelVaghasiya MINGW64 ~/Desktop/exp-2 (main)
$
```



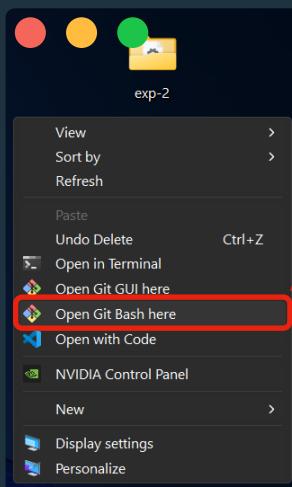
- Now refresh the repository webpage to see that we have successfully pushed our code.

The screenshot shows a GitHub repository page for 'exp-3'. The repository is private and has 1 branch and 0 tags. The 'Code' tab is selected. A list of 7 commits is shown, all made by 'neel-03' in the last 4 days. The commits include adding a logo to the navbar, initial commits for .eslintrc.cjs, .gitignore, README.md, index.html, package-lock.json, package.json, and vite.config.js. The sidebar on the right provides information about the repository, including 'Working with remote repo' (Readme, Activity, 0 stars, 1 watching, 0 forks), 'Releases' (No releases published, Create a new release), and 'Packages' (No packages published, Publish your first package).

- We have seen the `remote` and `push` commands. Now, let's understand the `clone` and `pull` commands with a simple scenario :
 1. `Clone` remote repo
 2. Make changes and `push` to remote repo
 3. Pull changes in original repo (named `exp-2`)

A screenshot of a GitHub repository page for 'exp-3'. The 'Code' button in the top right corner is highlighted with a red box. A dropdown menu is open from this button, showing options like 'Local' and 'Codespaces', and a 'Clone' section. Within the 'Clone' section, the URL 'https://github.com/neel-03/exp-3.git' is highlighted with a red box and has a 'Copy url to clipboard' button next to it. The right side of the dropdown shows repository statistics: 1 branch, 0 tags, 1 watching, 0 forks, and no releases or packages.

Open remote repo,
Click on 'Code' button and
Copy remote repo URL



Open git bash where we want to
clone the repo

Clone remote repo by running
'git clone <URL>' command

A screenshot of a terminal window titled 'MINGW64:/c/Users/neel/Desktop'. It shows the command '\$ git clone https://github.com/neel-03/exp-3.git' being run. The output of the command is displayed, showing the progress of cloning the repository, including object enumeration, compression, and receiving deltas.

```
neel@NeelVaghasiya MINGW64 ~/Desktop
$ git clone https://github.com/neel-03/exp-3.git
Cloning into 'exp-3'...
remote: Enumerating objects: 44, done.
remote: Counting objects: 100% (44/44), done.
remote: Compressing objects: 100% (27/27), done.
remote: Total 44 (delta 16), reused 44 (delta 16), pack-reused 0
Receiving objects: 100% (44/44), 39.98 KiB | 307.00 KiB/s, done.
Resolving deltas: 100% (16/16), done.
```



Open **cloned repo**,
Make changes,
and push changes to remote repo

```
const App = () => {
  return (
    <div>
      <Navbar/>
      Hello world..!
      Changes in main branch
      Changes in cloned repo...!
    </div>
  )
}

neel@NeelVaghasiya MINGW64 ~/Desktop/exp-3 (main)
$ git add .

neel@NeelVaghasiya MINGW64 ~/Desktop/exp-3 (main)
$ git commit -m "modify app.jsx in cloned repo"
[main 40f21c9] modify app.jsx in cloned repo
  1 file changed, 1 insertion(+)

neel@NeelVaghasiya MINGW64 ~/Desktop/exp-3 (main)
$ git push origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 448 bytes | 224.00 KiB/s, done.
```

- Let's take a look at our remote repo:

Commit	Message	Time
neel-03	modify app.jsx in cloned repo	40f21c9 · 2 minutes ago
src	modify app.jsx in cloned repo	2 minutes ago
.eslintrc.cjs	Initial commit	4 days ago

- Now, we've modified the **cloned repo (exp-3)** and pushed changes to the **remote repo**, so our **cloned repo** is **up to date**. But what about the **original repo (exp-2)**?
- When another developer pushes changes to the **remote repo**, we need to **pull those changes** to update our **original repo**.

- Let's open our original repo and pull the remote changes.

The image shows two side-by-side instances of the Visual Studio Code (VS Code) interface, illustrating a git pull operation between two branches of a repository.

Left Pane (Before pull):

- Explorer:** Shows the file structure of the `EXP-2` branch. The `src` folder contains `App.jsx`, `Footer.jsx`, `HeroSection.jsx`, and `Navbar.jsx` files.
- Editor:** Displays the `App.jsx` file with the following code:


```
1 import React from 'react'
2 import Footer from './components/Footer'
3 import Navbar from './components/Navbar'
4 import HeroSection from './components/HeroSection'
5
6 const App = () => {
7   return (
8     <div>
9       <Navbar/>
10      Hello world..!
11      Changes in main branch
12      <HeroSection/>
13      <Footer/>
14    </div>
15  )
16 }
```
- Terminal:** Shows the command `neel@NeelVaghasiya MINGW64 ~/Desktop/exp-2 (main)` followed by a blank line.

Right Pane (After pull):

- Explorer:** Shows the file structure of the `EXP-2` branch after pulling changes. The `src` folder now includes the `App.jsx` file from the main branch.
- Editor:** Displays the `App.jsx` file with the following updated code:


```
1 import React from 'react'
2 import Footer from './components/Footer'
3 import Navbar from './components/Navbar'
4 import HeroSection from './components/HeroSection'
5
6 const App = () => {
7   return (
8     <div>
9       <Navbar/>
10      Hello world..!
11      Changes in main branch
12      Changes in cloned repo..!
13      <HeroSection/>
14      <Footer/>
15    </div>
16  )
17 }
```
- Terminal:** Shows the command `neel@NeelVaghasiya MINGW64 ~/Desktop/exp-2 (main)` followed by the output of the `git pull` command:


```
$ git pull origin main
From https://github.com/neel-03/exp-3
 * branch            main      -> FETCH_HEAD
Updating a039c2e..40f21c9
Fast-forward
 src/App.jsx | 1 +
 1 file changed, 1 insertion(+)
```

Enough solo work -- let's collaborate and contribute..!



03 COLLABORATING & CONTRIBUTING ON GITHUB

03 COLLABORATING & CONTRIBUTING



Primary Roles in Repository Management

1. Owner:

- Has the highest level of control.
- Can change repo settings, invite and manage collaborators, manage access control, merge the pull request and manage issues, transfer repository ownership, delete the repository

2. Collaborator:

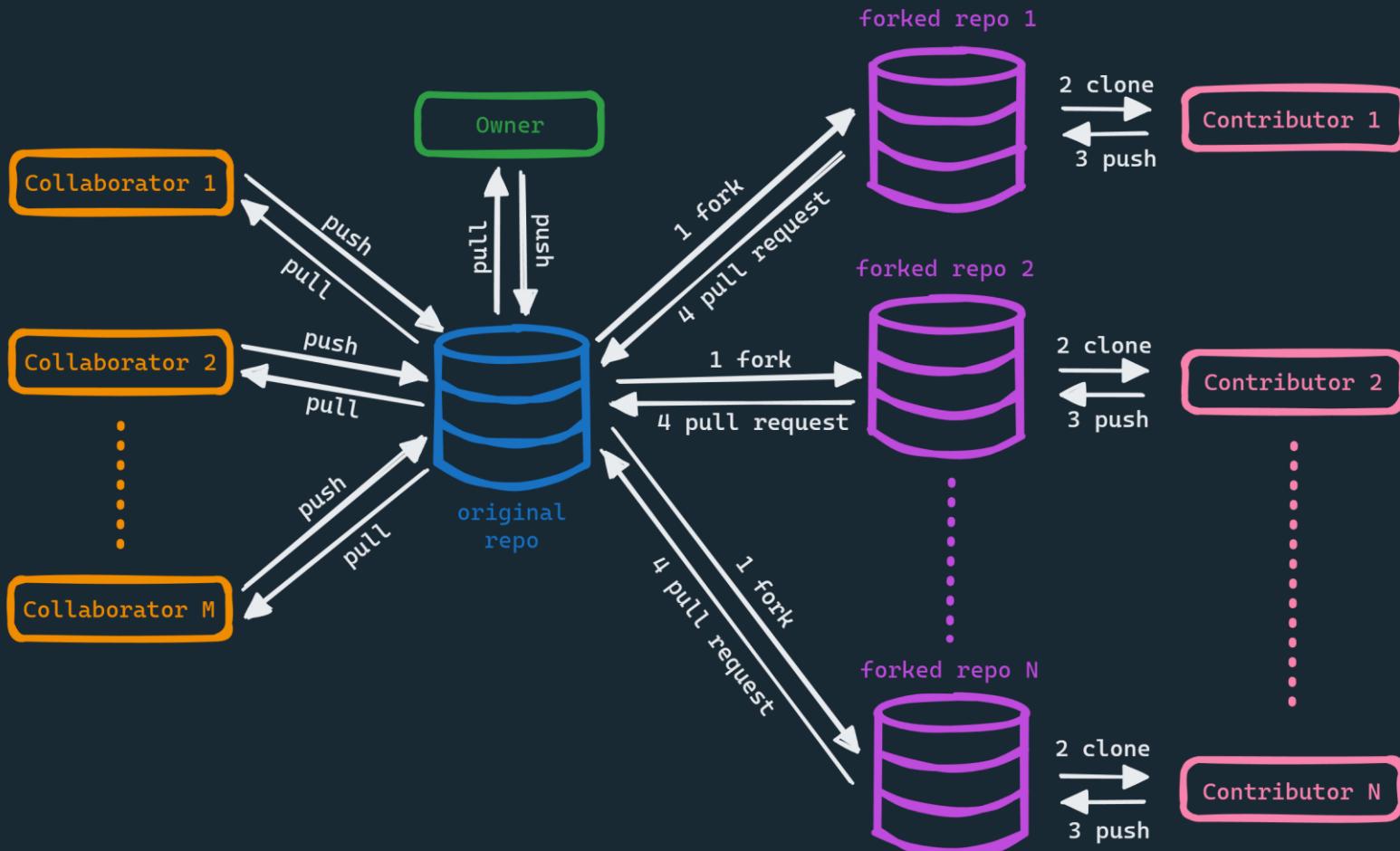
- Has write access to the repo. (can directly push the changes)
- Can change repo settings, Merge the pull requests and manage issues.

3. Contributor:

- Only has read access. (cannot directly push the changes)
- Can suggest changes by forking, and submitting pull request.



GitHub workflow for open-source and collaborative projects





Contribute to open source project

What are Open source projects?

- Projects in which source code is made available to the public.
- Anyone can inspect, modify, and contribute to the project.

Why Contribute to Open Source?

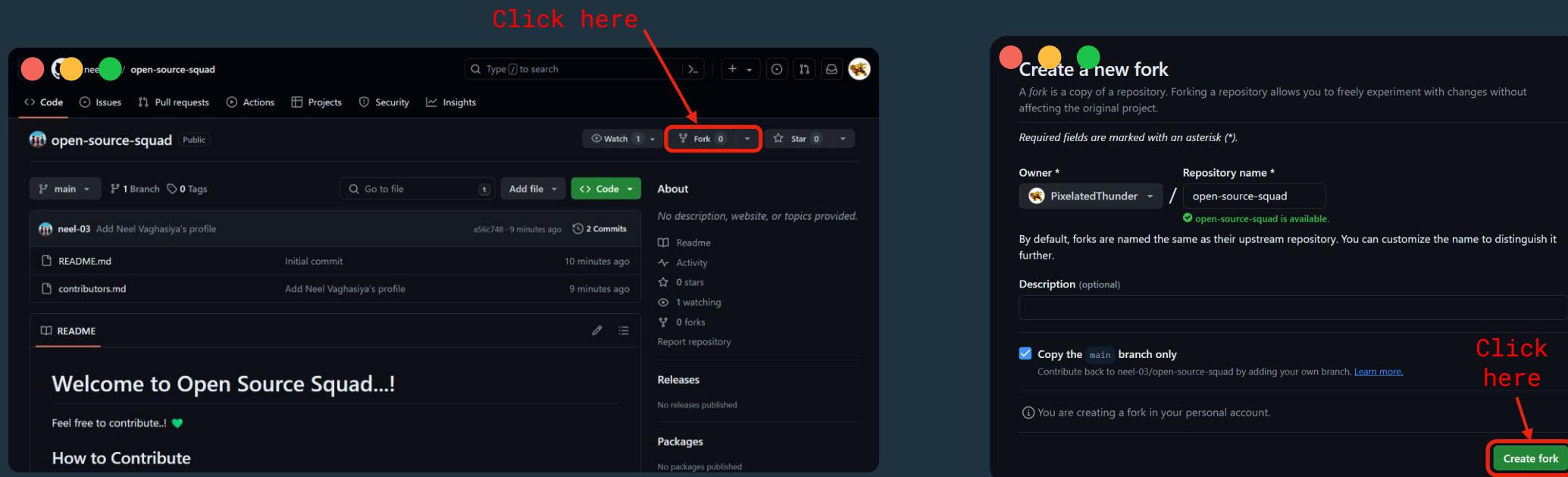
- Skill Development: Learn new technologies, and gain practical experience
- Best Practices: Learn industry practices by accessing high-quality codebases
- Expand Network: Connect with other developers
- Learning opportunity: Stay updated with industry trends

Let's Contribute to a Sample Repository

I've set up a repository for this section. Feel free to explore and contribute! Just follow the steps provided.

Step 1: Fork the repository

- Navigate to the repository you want to contribute to. (in this scenario, navigate to '<https://github.com/neel-03/open-source-squad>')
- Click 'Fork' button at the top-right corner of the page.



- Fork creates a copy of the repository under your GitHub account.

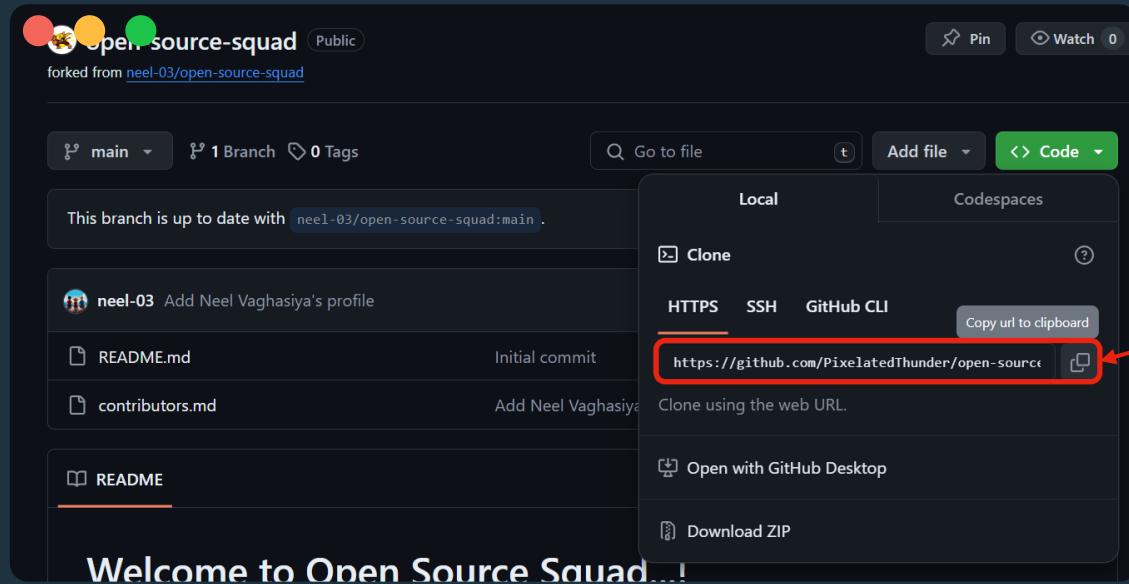


- Open your profile page, and you will see the forked repository.

The screenshot shows a GitHub profile page for a user named Neel Vaghasiya. The profile picture is a pixelated illustration of a character from a video game. The user's name is Neel Vaghasiya and their GitHub handle is PixelatedThunder. Below the profile picture, there is an 'Edit profile' button and a note that the user joined yesterday. The main content area displays 'Popular repositories' with one repository listed: 'open-source-squad' (Forked from neel-03/open-source-squad). A red box highlights this repository, and a red arrow points to it with the label 'Forked repo'. Below the repository list, there is a 'Contribution activity' section showing contributions for the year 2024, with a timeline from June to June. The contributions are represented by colored dots (Mon, Wed, Fri) and a legend indicating 'Less' and 'More' contributions.

Step 2: Clone the Forked repository

- Open that forked repository.
- Click the 'Code' button and copy the URL.



- Open Git bash and clone forked repo.
- After that open that folder in VS code.

A screenshot of a terminal window titled 'MINGW64;/c/Users/neel/Desktop'. The command 'git clone https://github.com/PixelatedThunder/open-source-squad.git' is being run. The output shows the cloning process, including object enumeration, compression, and receiving objects, all completed successfully at 100%.

```
neel@NeelVaghasiya MINGW64 ~/Desktop
$ git clone https://github.com/PixelatedThunder/open-source-squad.git
Cloning into 'open-source-squad'...
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 7 (delta 1), reused 7 (delta 1), pack-reused 0
Receiving objects: 100% (7/7), done.
Resolving deltas: 100% (1/1), done.
```



Step 3: Create new branch and Make your changes

- Instead of modifying the 'main' branch, create a new branch.
- Run '`git checkout -b add-profile/<your-name>`' command to create a new branch and switch to it.

The screenshot shows a terminal window with the following content:

```
neel@NeelVaghasiya MINGW64 ~/Desktop/open-source-squad (main)
$ git checkout -b add-profile/PixelatedThunder
Switched to a new branch 'add-profile/PixelatedThunder'

neel@NeelVaghasiya MINGW64 ~/Desktop/open-source-squad (add-profile/PixelatedThunder)
$
```

The command `git checkout -b add-profile/PixelatedThunder` and its output are highlighted with a red box.

- Open the remote repo, copy the snippet from 4th bullet from 'README.md' and paste it into 'contributors.md'. Replace your profile URL, avatar URL and your name as mentioned.

The screenshot shows a GitHub code editor with the file `contributors.md` open. The content of the file is:

```
You, 1 second ago | 1 author (You)
# List of Contributors
|Profile picture|Contributor Name|
|:--:|:--:|
|<img src='https://avatars.githubusercontent.com/u/96440861?v=4' height='25' width='25'>|<a href='https://github.com/neel-03'>Neel Vaghasiya</a>|
|<img src='https://avatars.githubusercontent.com/u/173437291?v=4' height='25' width='25'>|<a href='https://github.com/PixelatedThunder'>PixelatedThunder</a>|
```

The URLs and names in the last two rows are highlighted with a red box. A red arrow points to the last row, and the text 'Replace the URLs and name with your actual ones' is displayed below it.



Step 4: Commit your changes

- Stage, and commit the changes with meaningful message: (Make sure you are on **your branch**)

```
neel@NeelVaghasiya MINGW64 ~/Desktop/open-source-squad (add-profile/PixelatedThunder)
$ git add .

neel@NeelVaghasiya MINGW64 ~/Desktop/open-source-squad (add-profile/PixelatedThunder)
$ git commit -m "Add PixelatedThunder's profile"
[master/branchname c8bbc5c] Add PixelatedThunder's profile
 1 file changed, 2 insertions(+), 1 deletion(-)

neel@NeelVaghasiya MINGW64 ~/Desktop/open-source-squad (add-profile/PixelatedThunder)
$
```

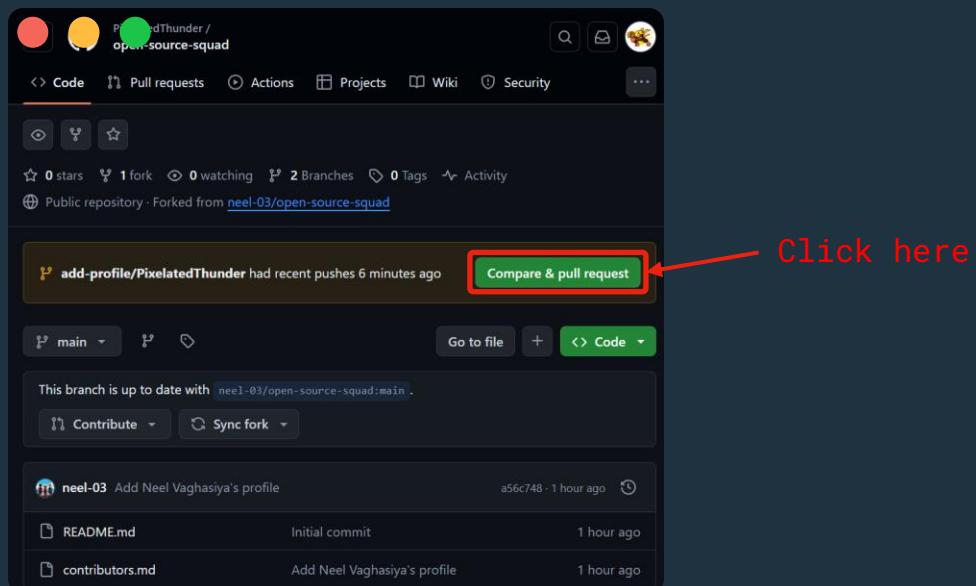
Step 5: Push the changes and Create a Pull Request

- Push **your branch** to your forked repository.
- Run this command : **git push origin add-profile/<your-name>**

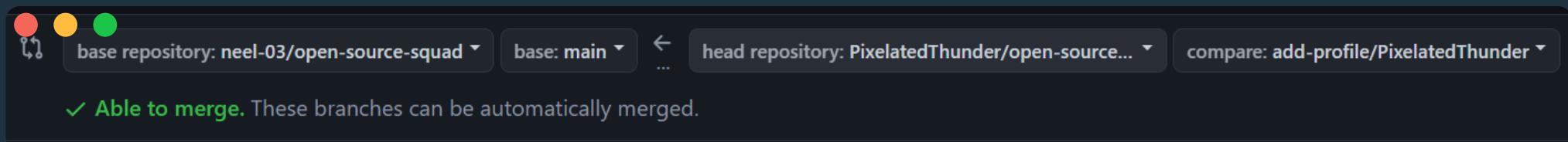
```
neel@NeelVaghasiya MINGW64 ~/Desktop/open-source-squad (add-profile/PixelatedThunder)
$ git push origin add-profile/PixelatedThunder
info: please complete authentication in your browser...
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 374 bytes | 374.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'add-profile/PixelatedThunder' on GitHub by visiting:
remote:   https://github.com/PixelatedThunder/open-source-squad/pull/new/add-profile/PixelatedThunder
remote:
To https://github.com/PixelatedThunder/open-source-squad.git
 * [new branch]      add-profile/PixelatedThunder -> add-profile/PixelatedThunder
```

After pushing,
Git suggests
Creating
pull request

- Now, go to your forked repository.
- You will see the prompt for creating pull request, just click on '**Compare & pull request**' button.

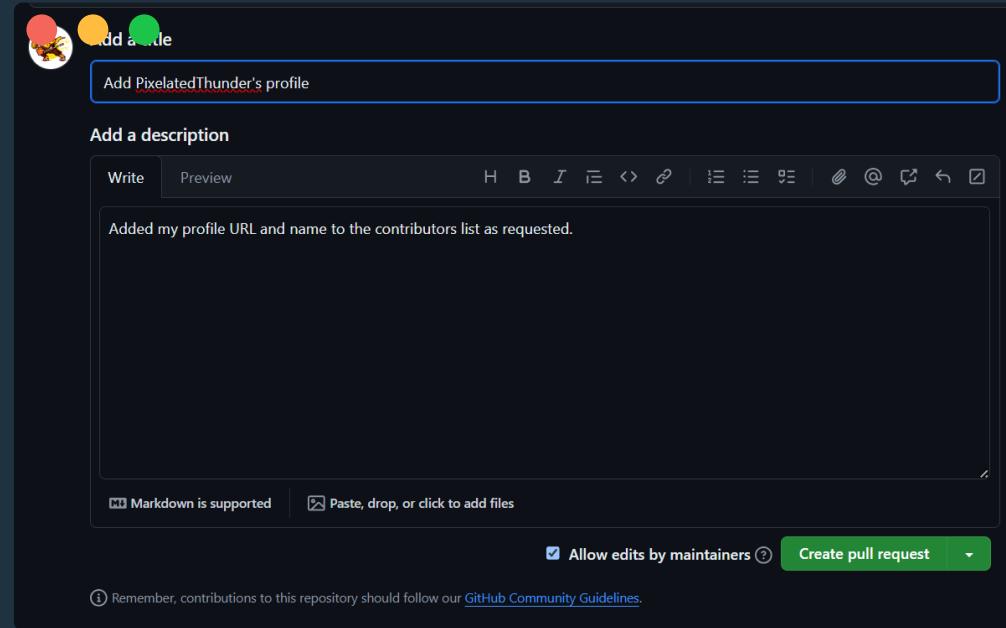


- In next page, ensure you are merging the correct branch from your forked repository into the correct branch of the original repository.





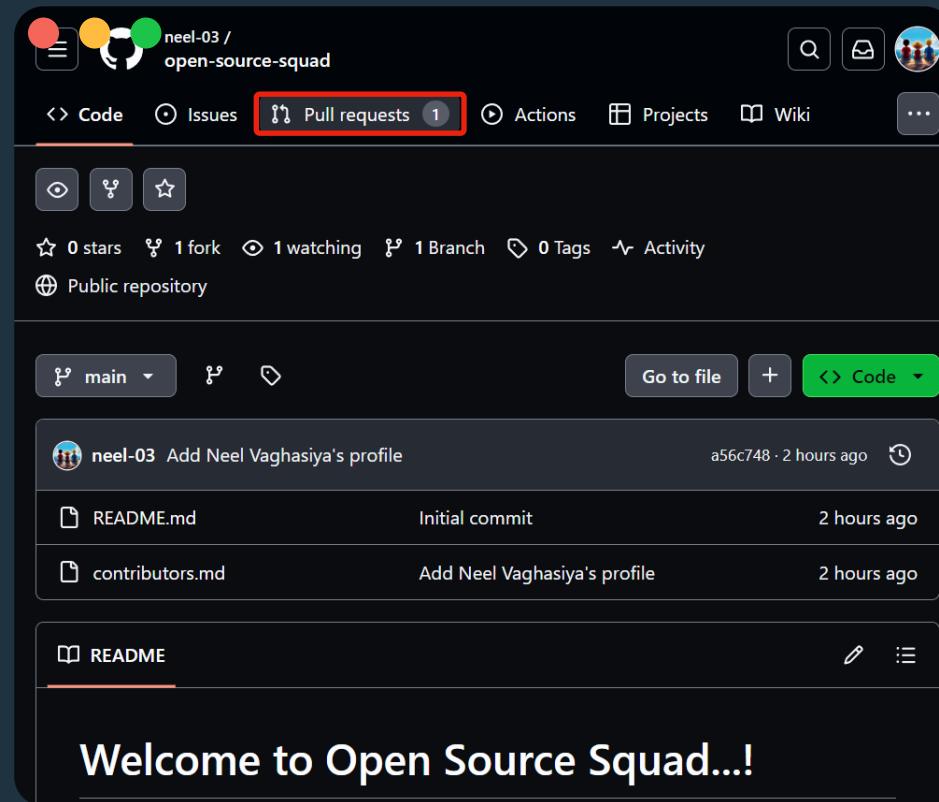
- Provide a descriptive title and detailed description of your changes.
- Click on ‘Create pull request’ button.



- **Congratulations..!** You have successfully created pull request...!
- After submitting a pull request, the pull request page will open. From there, contributor can add comment, or close pull request.
- Now, wait for the owner or collaborator to review and merge your pull request.

Let's merge the pull request created by contributors

- Only repository **owner**, and **collaborators** can review contributors' pull requests.
- Open the original repo and go to 'Pull request' tab.





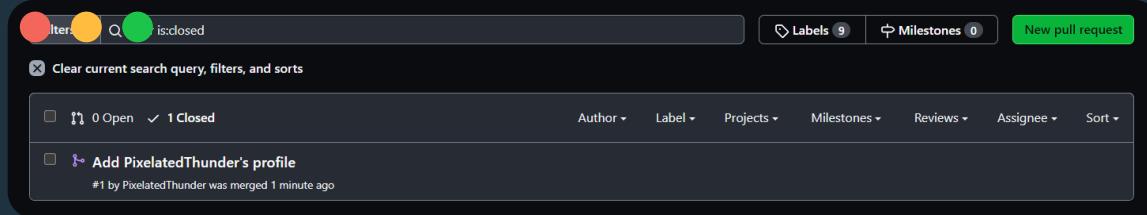
- We can see all the opened pull request.

A screenshot of the GitHub interface showing the 'Pull requests' tab. A single pull request is listed under the 'Open' section. The pull request is titled 'Add PixelatedThunder's profile' and was opened 44 minutes ago by 'PixelatedThunder'. A red arrow points from the text 'Select the pull request for review or merge' to the title of the pull request card.

- In next page, collaborator can see PR details, Conversation tab, Options for merging pull request, and button to close pull request.

A screenshot of the GitHub Pull Request details page for pull request #1. The title is 'Add PixelatedThunder's profile #1'. It shows a conversation from 'PixelatedThunder' stating they added their profile URL and name to the contributors list. Below the conversation, there are merge options: 'Continuous integration has not been set up' and 'This branch has no conflicts with the base branch'. A red arrow points from the text 'Click here to merge PR' to the 'Merge pull request' button at the bottom of the page.

- We can see pull request has been closed.



- Observe the last commit, and 'contributors.md' file.

A screenshot of a GitHub repository page for "open-source-squad". The repository is public and has one branch ("main") and no tags. The commit history shows a merge commit from "PixelatedThunder" and a commit to "contributors.md". The "contributors.md" commit message is "Add PixelatedThunder's profile". Both commits were made 3 minutes ago by user "neel-03".

A screenshot of the "contributors.md" file content on GitHub. The file is titled "List of Contributors" and contains two entries:

Profile picture	Contributor Name
	Neel Vaghasiya
	PixelatedThunder

Congratulations on clearing the basics of Git and GitHub..!

-- Remember "*Every commit tells a story of progress.*"



THANK YOU!



neel-03