How to contribute to Open-Source

(Open-Source contribution workflow)

Note: The discussion on this article assumes that you already know about git and GitHub and you are familiar with git branches etc.

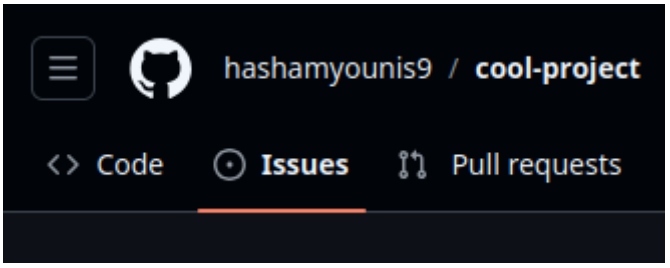
1. Find an Open-Source project

The first step is to find an open-source project of your interest. You can search on Google or GitHub your favourite topics and in the search on Google add “github” in the end of the topic, it will GitHub repos. For now, you’ll go to the mock repo created in the video named as cool-project .

2. Navigate through the project

Next step is to navigate through the project, try and understand the project. What is the project about? What tech stack are they using? How is the overall code working altogether? What is the solution they are trying to build and what problem are they solving and what is their target audience?

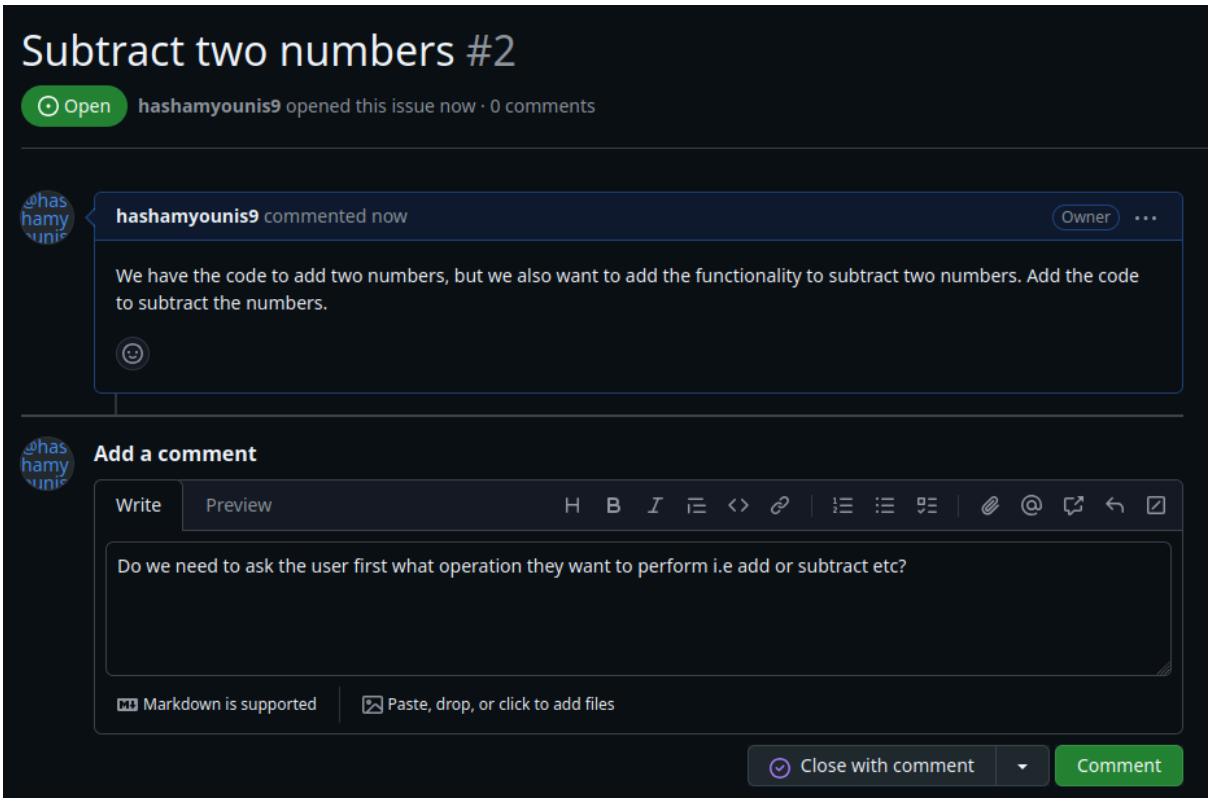
3. Go to the issues



After understanding the project, its working, and most importantly its purpose, when you decide you want to work on this project, you go to the **issues** tab. Now these issues are what’s wrong with this project and what needs attention. These are the problems with the current project that you as an open-source

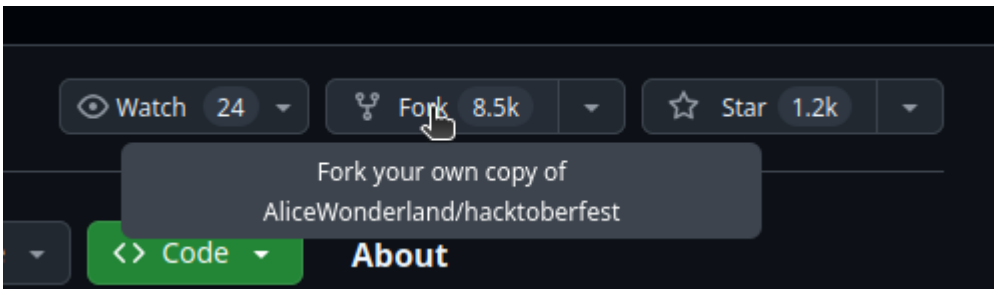
contributor can work on and resolve. Now find the issue which seems interesting to you or your expertise allows you to work on. Issues can be labelled with different kinds of keywords such as bug, feature, enhancement, good first issue etc. Newcomers may be more interested in issues labelled as **good first issue** as issues with this tag require less technical expertise and are intended to allow beginners to be able to work on open-source projects.

After finding an issue of your choice/interest, try to understand the problem from the description of the issue and if you have any queries regarding the issue, you can add a comment to ask your query as shown:

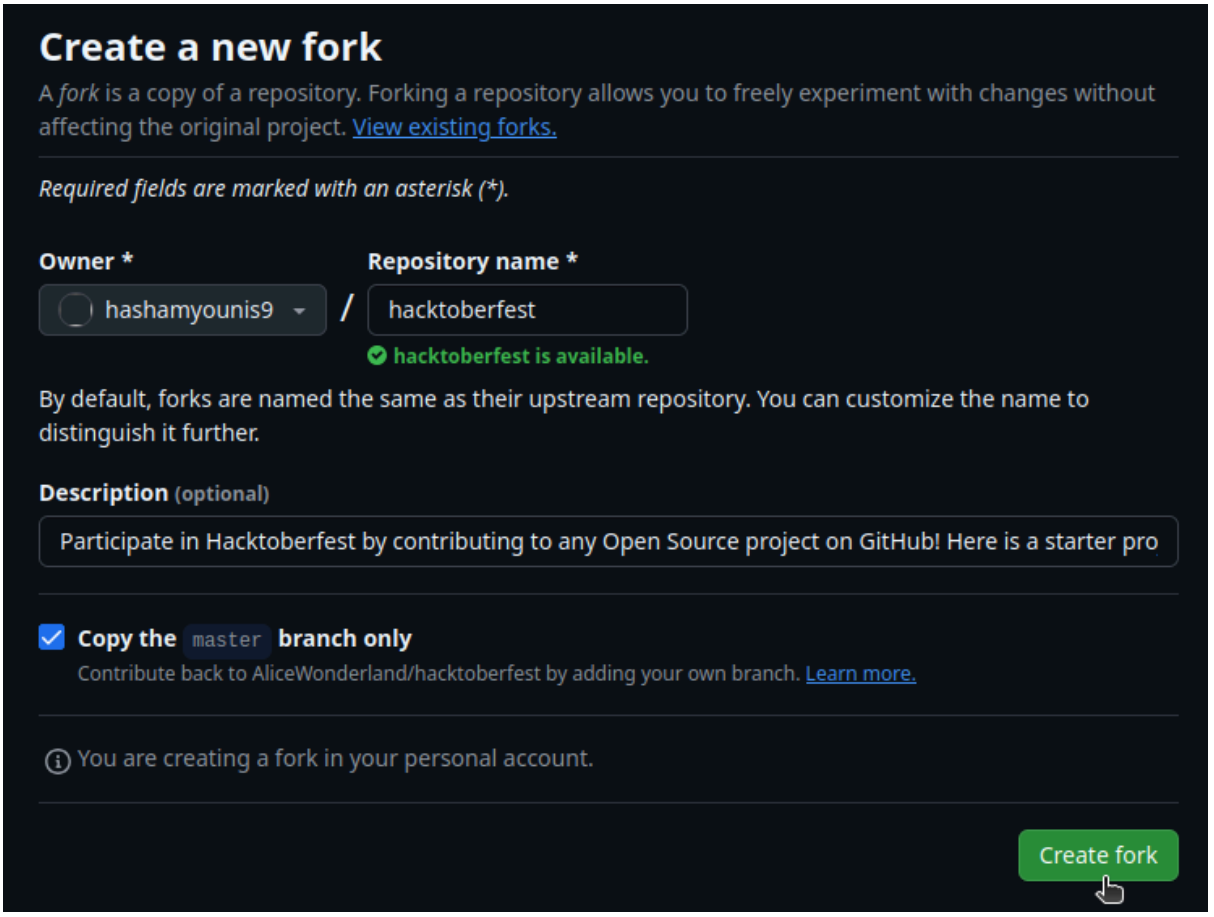


4. Fork the project

Click on the fork button located on top right corner



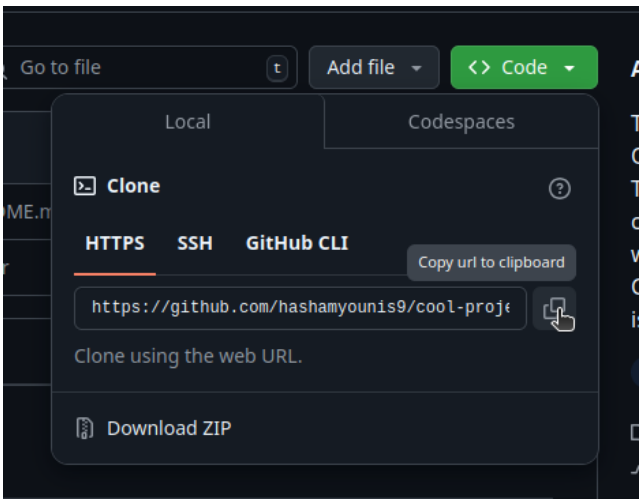
And then click on Create Fork on next page



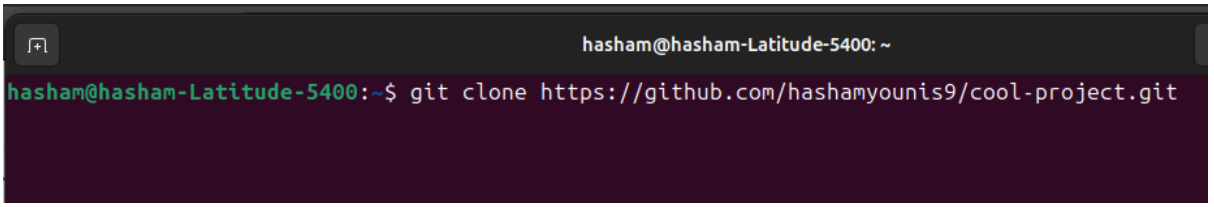
This will create a copy of this project on your GitHub account, which you can modify and make changes to.

5. Clone the project on you local machine

Copy the project https url



Go to the terminal and run the following command
(make sure you have git installed on your machine, if not, you can follow any video on youtube to download and install git according to your platform i.e. Linux, Mac and Windows.)



Each project will have its own unique url, the above one is of our mock project, cool-project . This command should clone/download the project on your machine, make sure you are connected to the internet and also keep careful about the location where you are cloning the project.

6. Setup the project on you local machine

Now this is sometimes the most difficult part, when you need to set up the project locally. Open the project in an IDE and then the setting up the project locally includes:

- Downloading the dependencies

  Suppose the project is in python language and the project uses some package/library i.e. matplotlib (useful for plotting graphs / data visualisation). Now you need to have this package installed on your machine in order to run

the project on your device. All the different tech stacks will have their own way of defining and downloading the dependencies.

- Having the compiler/interpreter installed

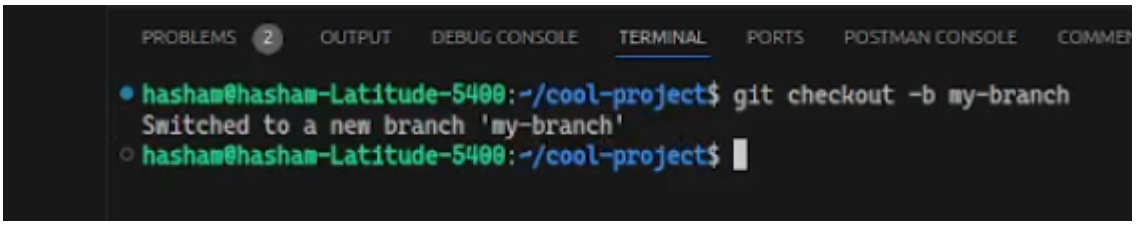
  You also need to have the compiler or interpreter installed on your machine i.e C++, Java Python etc.

After downloading dependencies and having the compiler installed on your machine, run the project and resolve any issue if there is any while trying to run the project.

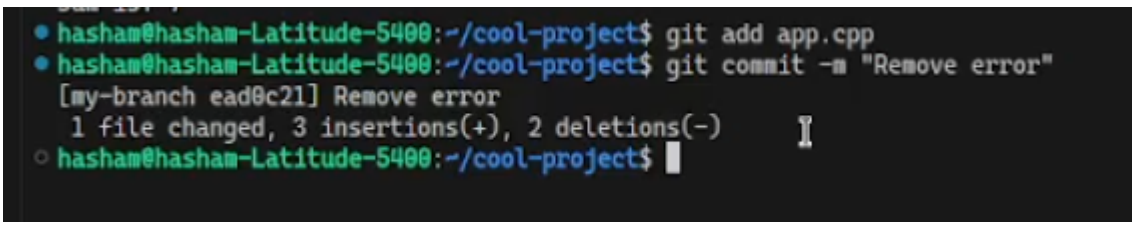
7. Make the changes

After you successfully run the project locally, you are ready to move on to the part where you can make your own changes and solve the issue you chose.

Before making any changes, you need to first create your own branch (assuming you already know about branches). After creating the branch with a meaningful name using the command:



You can make changes or solve the issue that you chose before and then commit your changes:

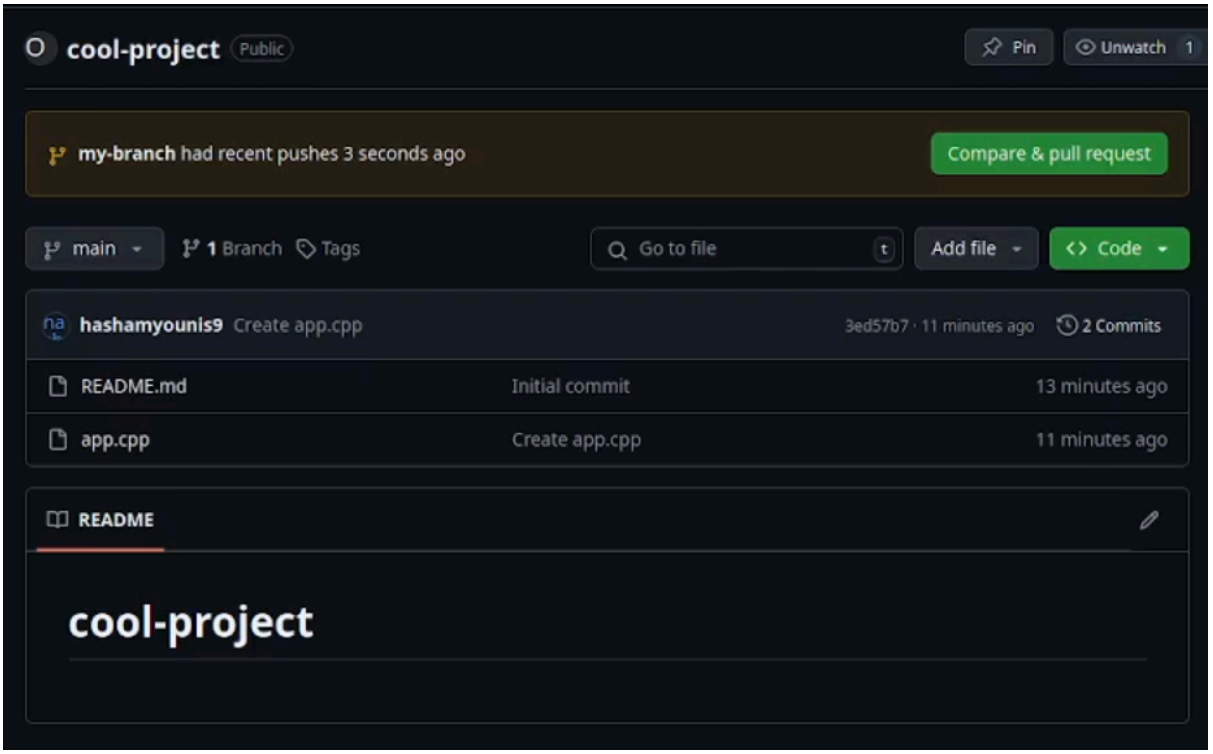


And then finally push them:

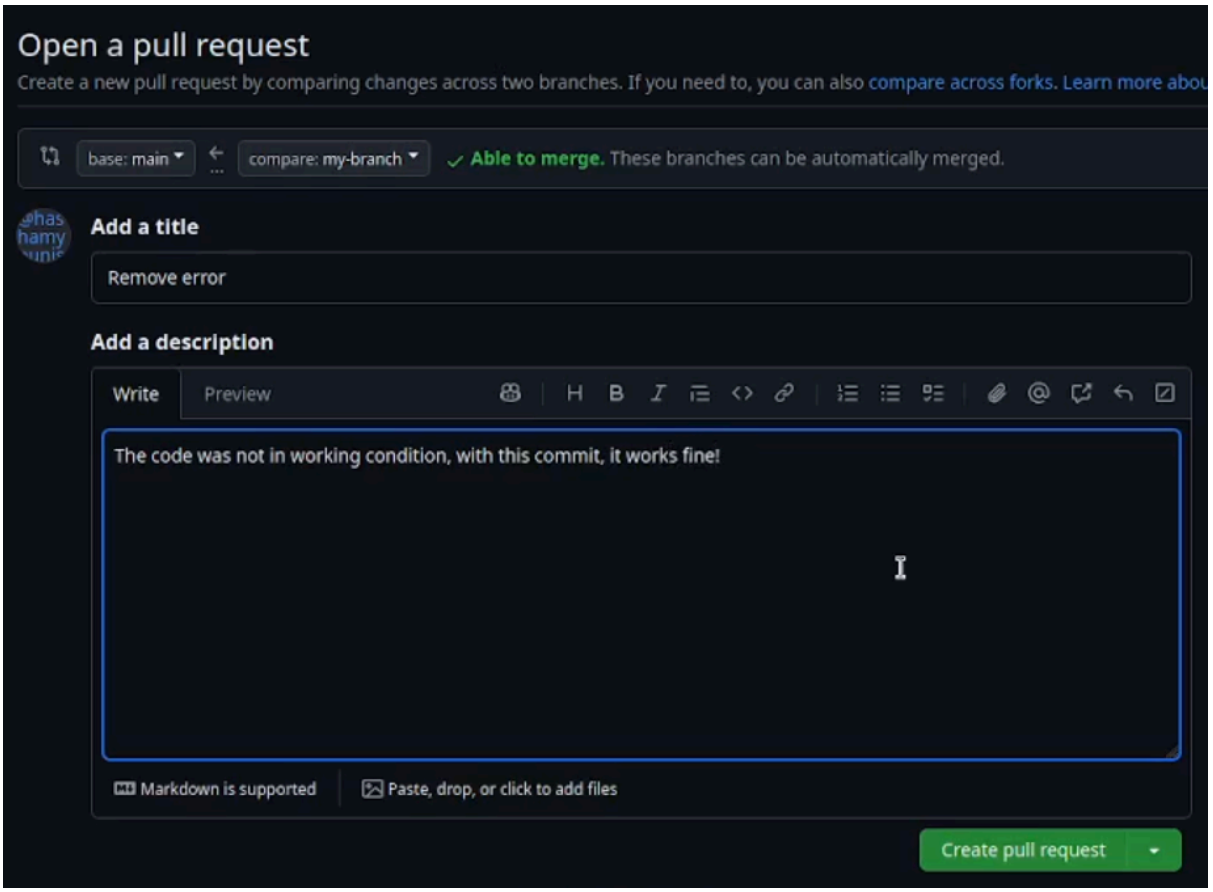
```
hasham@hasham-Latitude-5400:~/cool-project$ git push --set-upstream origin my-branch
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 406 bytes | 406.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'my-branch' on GitHub by visiting:
remote:      https://github.com/hashamyounis9/cool-project/pull/new/my-branch
remote:
To https://github.com/hashamyounis9/cool-project.git
 * [new branch]      my-branch -> my-branch
branch 'my-branch' set up to track 'origin/my-branch'.
hasham@hasham-Latitude-5400:~/cool-project$
```

8. Create the pull request!

Now on GitHub, you'll see something like:



Click on Create & pull request, it will take you to this page:



Fill in the appropriate title and give a nice description of what you did while solving the issue and create the pull request.

Now, after you create the pull request the owner will receive the notification from GitHub that someone created a pull request. Then after reviewing the PR, they can decide to either merge it or decline the pull request. If they merge your PR, then Congratulations! And if not, you can modify the PR again according to their requirement and try again.

Note:

The document is still not very Formal and may contain some inconsistencies but no worries, I'll be writing an article soon on Medium by enhancing this document and explaining the process in a better way.

Happy Coding!