A Project report on

# “Task 2”

with

# Source Code Management

(22CS003)

**Submitted by:** Submitted To:

Faculty Name:

Department of Computer Science & Engineering, Chitkara University Institute of engineering and Technology Rajpura, Punjab

Team Member 1 Name: AKSHI MITAL  
 Roll No. : 2210991232

Team Member 2 Name: AKSHI TRIPATHI   
 Roll No.: 2210991233

Team Member 3 Name: AKSHIT   
 Roll No.:2210991234



Institute/School Name: -**Chitkara University Institute of Engineering and Technology**

Department Name: - **Department of Computer Science & Engineering**

Program Name:

**Table of Content**

|  |  |  |
| --- | --- | --- |
| **S.NO.** | **EXPERIMENT NAME** | **Page No.** |
|  | **Introduction** |  |
|  | **Problem Statement** |  |
|  | **Solution** |  |
|  | **Objective** |  |
|  | **Create a distributed repository and add**  **members in a project team** |  |
|  | **Open and close pull request** |  |
|  | **Create a pull request on a team members repo**  **and close pull requests generated by team**  **members on own Repo as a maintainer** |  |
|  | **Publish and print network graphs** |  |

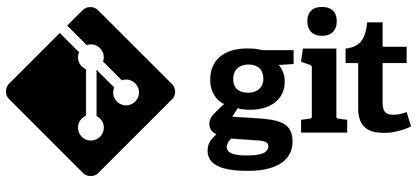
**INTRODUCTION**

**What is GIT and why is it used?**

Git is a version control system that is widely used in the programming world. It is used for tracking changes in the source code during software development. It was developed in 2005 by Linus Torvalds, the creator of the Linux operating system kernel.

Git is a speedy and efficient distributed [VCS](https://www.w3docs.com/learn-git/version-control-software.html) tool that can handle projects of any size, from small to very large ones. Git provides cheap local branching, convenient staging areas, and multiple workflows. It is free, open-source software that lowers the cost because developers can use Git without paying money. It provides support for non-linear development. Git enables multiple developers or teams to work separately without having an impact on the work of others.

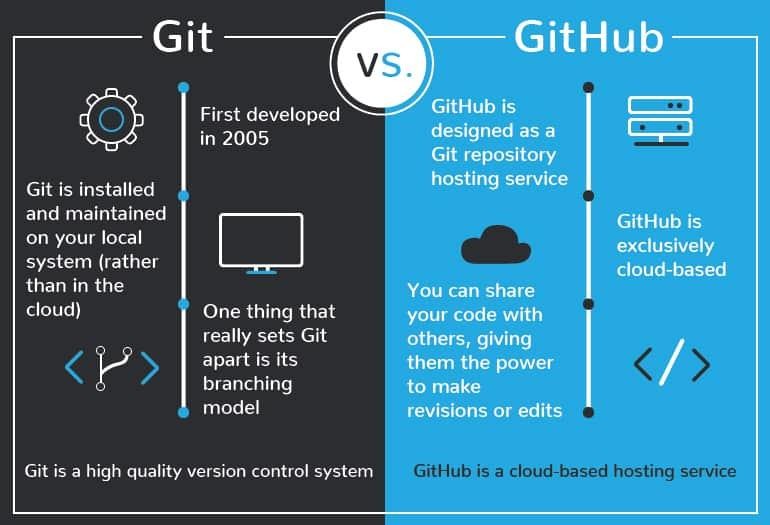
Git is an example of a distributed version control system (DVCS) (hence Distributed Version Control System).



**What is GITHUB?**

It is the world’s largest open-source software developer community platform where the users upload their projects using the software Git.

**What is the difference between GIT and GITHUB?**



**What is Repository?**

A repository is a directory or storage space where your projects can live. Sometimes GitHub users shorten this to “repo.” It can be local to a folder on your computer, or it can be a storage space on GitHub or another online host. You can keep code files, text files, image files, you name it, inside a repository.

**What is Version Control System (VCS)?**

A version control system is a tool that helps you manage “versions” of your code or changes to your code while working with a team over remote distances. Version control keeps track of every modification in a special kind of database that is accessible to the version control software. Version control software (VCS) helps you revert back to an older version just in case a bug or issue is introduced to the system or fixing a mistake without disrupting the work of other team members.

Types of VCS

1. Local Version Control System
2. Centralized Version Control System
3. Distributed Version Control System

I. **Local Version Control System:** Local Version Control System is located in your local machine. If the local machine crashes, it would not be possible to retrieve the files, and all the information will be lost. If anything happens to a single version, all the versions made after that will be lost.

AI. **Centralized Version Control System:** In the Centralized Version Control Systems, there will be a single central server that contains all the files related to the project, and many collaborators checkout files from this single server (you will only have a working copy). The problem with the Centralized

Version Control Systems is if the central server crashes, almost everything related to the project will be lost.

BI. **Distributed Version Control System:** In a distributed version control system, there will be one or more servers and many collaborators similar to the centralized system. But the difference is, not only do they check out the latest version, but each collaborator will have an exact copy of the main repository on their local machines. Each user has their own repository and a working copy. This is very useful because even if the server crash we would not lose everything as several copies are residing in several other computers.

**Problem statement**

“Develop a game and upload it to GitHub”

In all our day-to-day life we are tired and want something fun in our life, so isn’t playing a game a best option. I know in today’s world physical outdoor games are mind refreshing but due to lack of time we can’t play it. So, we have come up with a game which is fun and mind refreshing to play which also helps to increase mind stability and help children to grow up their mind.

[According to some studies, there are many factors that affect children’s opportunities to play outside, such as the physical environment of their day-care centers, the educators’ attitudes and the policy context](https://bing.com/search?q=children+are+unable+to+play+outdoors+due+to+lack+of+time). [Some of the benefits of outdoor play include sunshine, exercise, executive function, taking risks and socialization](https://www.health.harvard.edu/blog/6-reasons-children-need-to-play-outside-2018052213880). [Some of the consequences of spending less time outdoors include health problems, emotional repercussions and impaired social skills](https://www.tandfonline.com/doi/full/10.1080/21594937.2019.1643979)

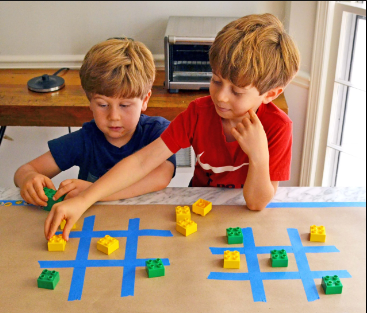


**SOLUTION**

Games can be a useful escape from boredom!!!

Playing games can have many benefits for your brain and well-being. According to some sources, playing games can:

* Promote a number of cognitive skills including counting and spatial  skills
* contributes to children's developmental growth in numerous ways including their understanding of predictability, problem solving, spatial reasoning, hand-eye coordination, turn taking, and strategizing



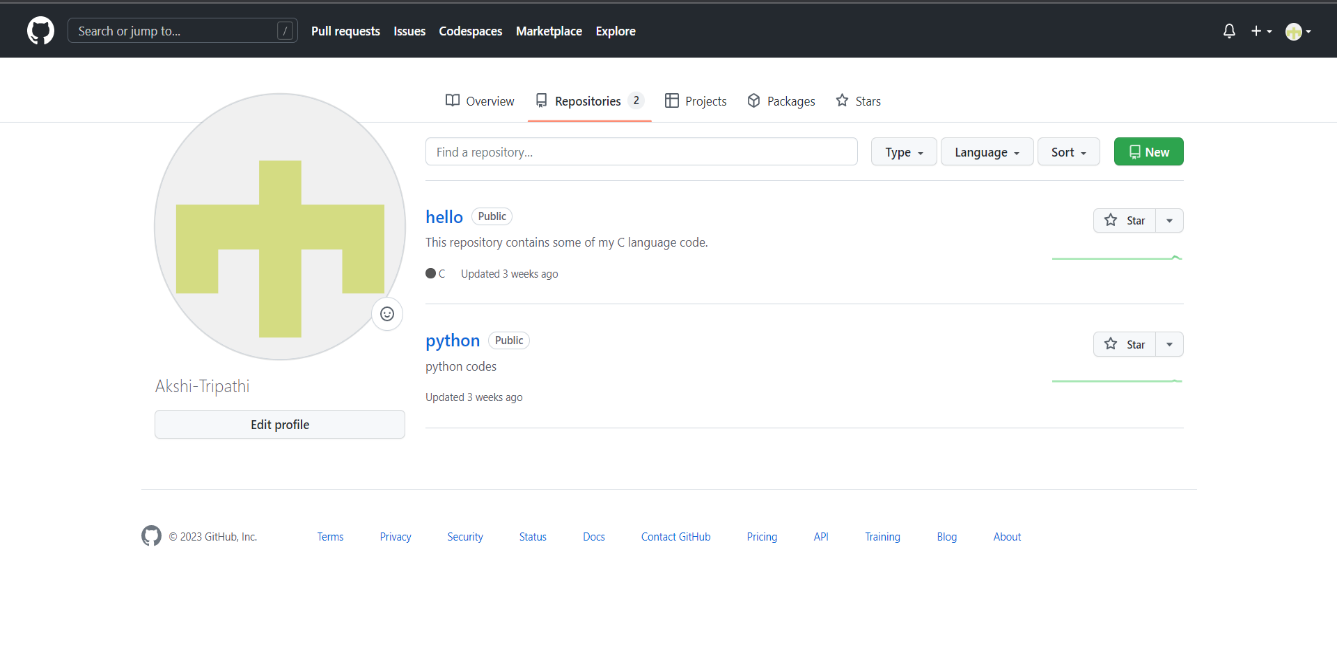
**OBJECTIVE**

The objective of this project is to associate programming with git because:

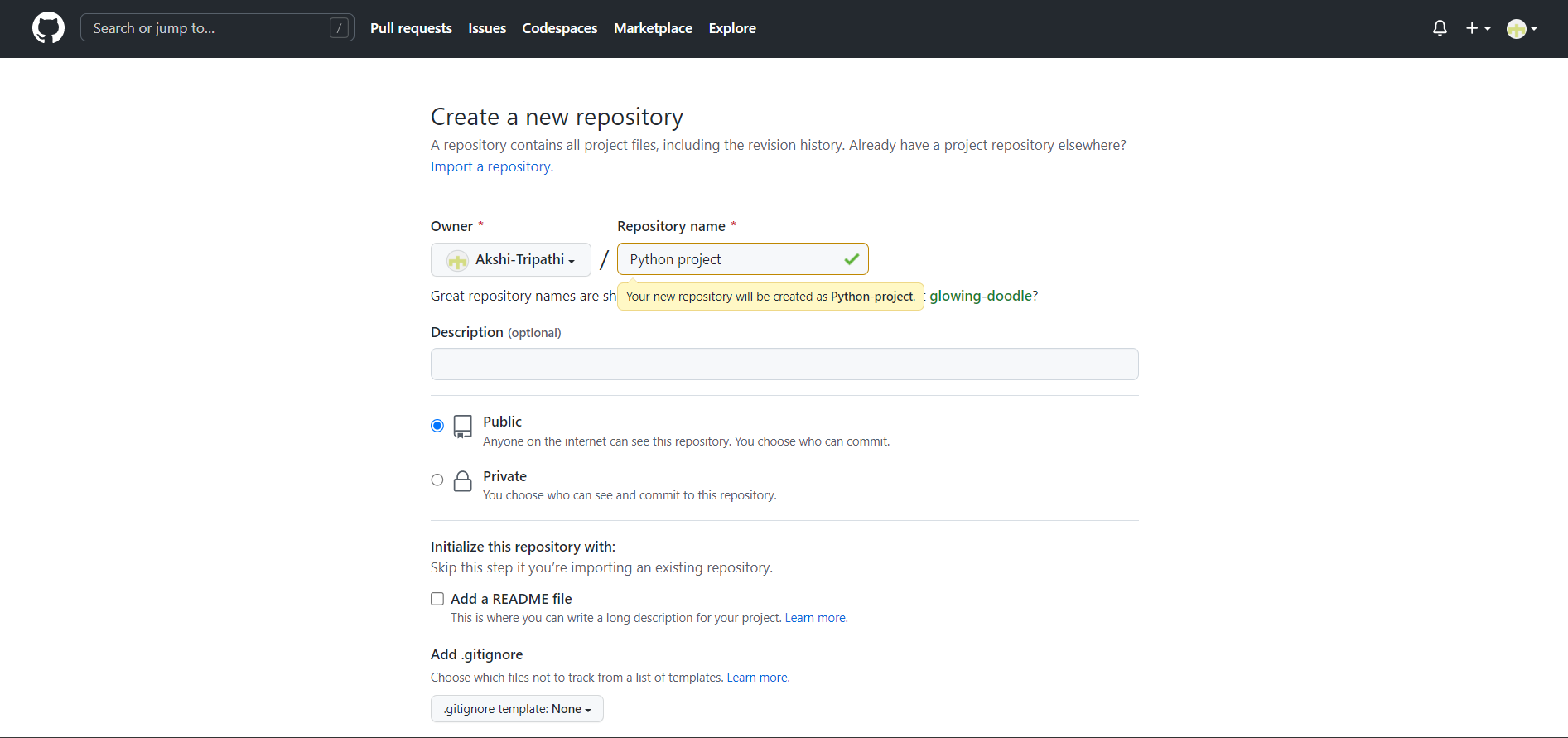
1. This is required because the collaboration makes the team work easy.
2. The code becomes manageable and we can build a clean repository.
3. Tracking and resolving of the errors is quite feasible in this process.
4. Moreover, we can make our locally available projects, globally available.

**CREATE A DISTRIBUTED REPOSITORY AND ADD MEMBERS IN PROJECT TEAM**

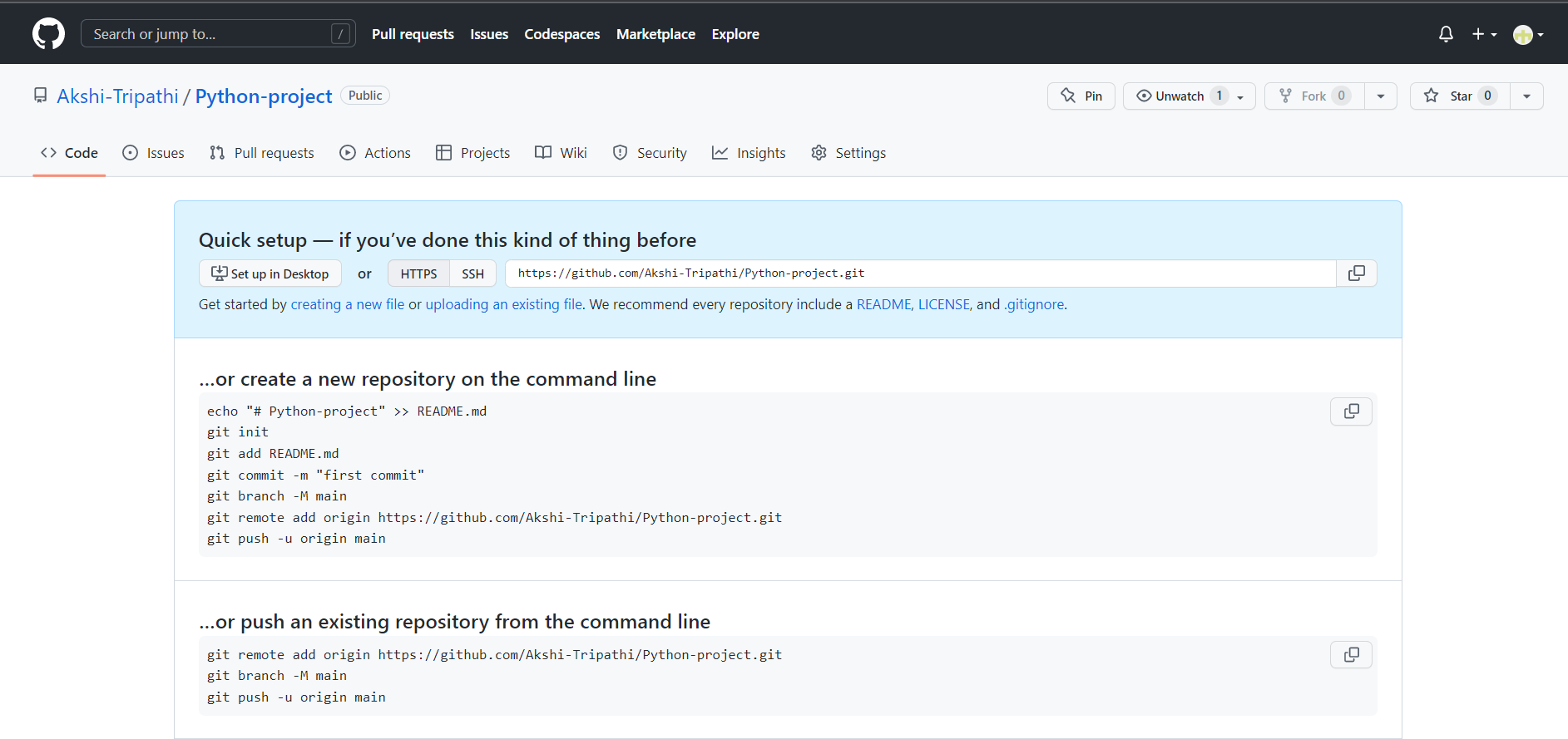
* Login to your GitHub account and you will land on the homepage as shown below. Click on Repositories option in the menu bar.



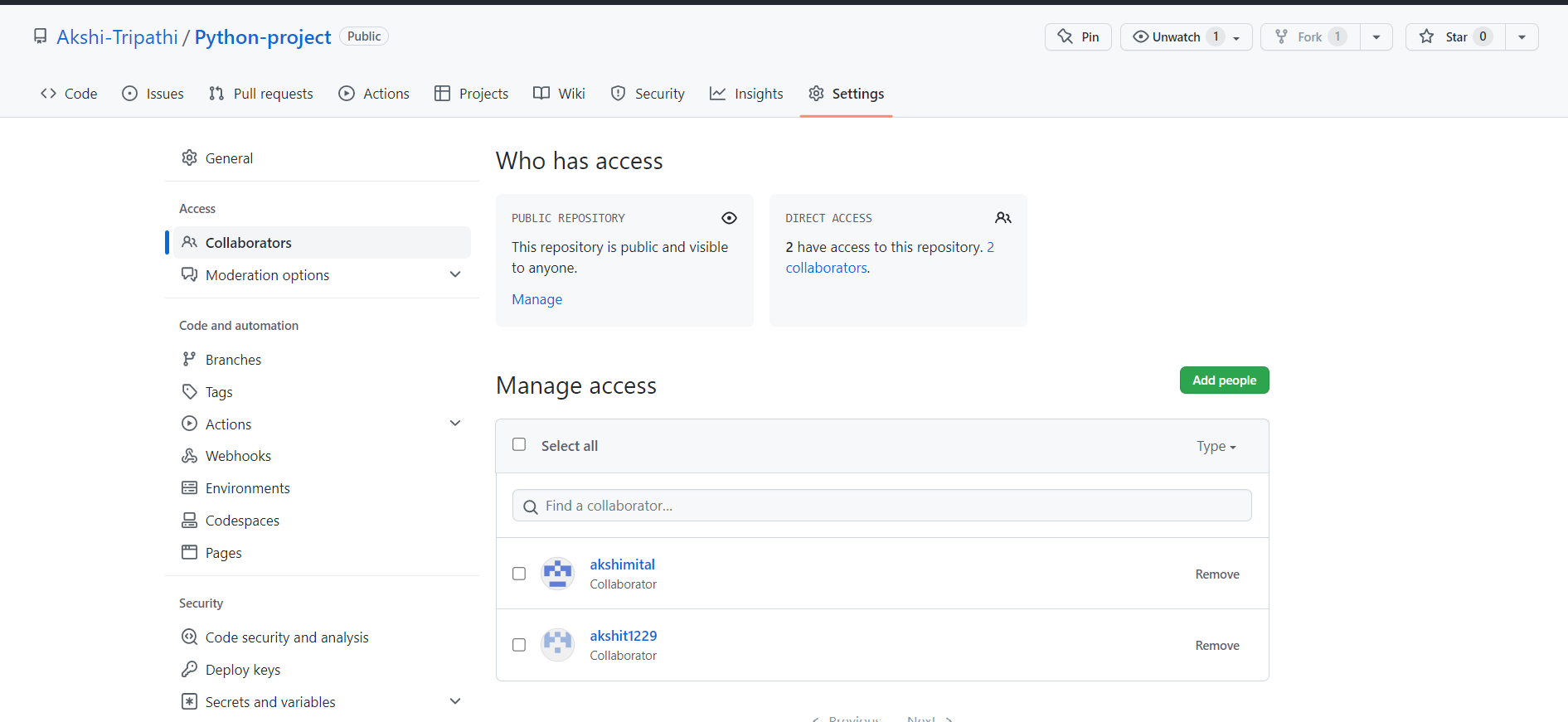
* Click on the “New” button in the top right corner. Enter the repository name and add description. Select if you want a public or private repository.

*(Repository created by owner: Akshi)*

* If you want to import code from an existing repository select the import code option. Noe successfully you have created repository.



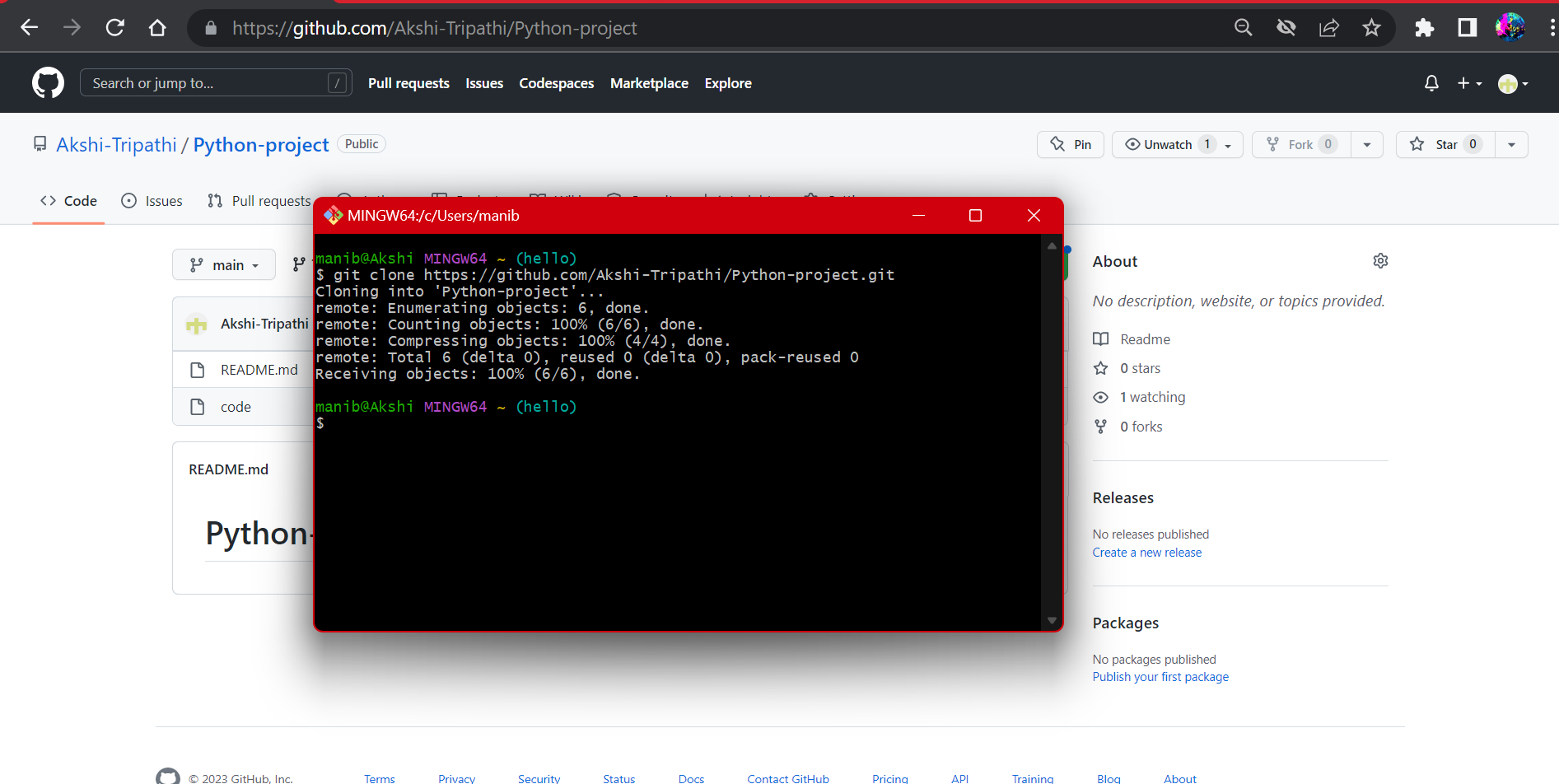
* To add members to your repository, open your repository and select settings option in navigation bar. Click on collaborators option under access tab. After clicking on collaborators Github asks you to enter your password to confirm the access to the repository. After entering password, you can add collaborators.
* To remove any member, click on remove option available in the last column of member’s respective row.



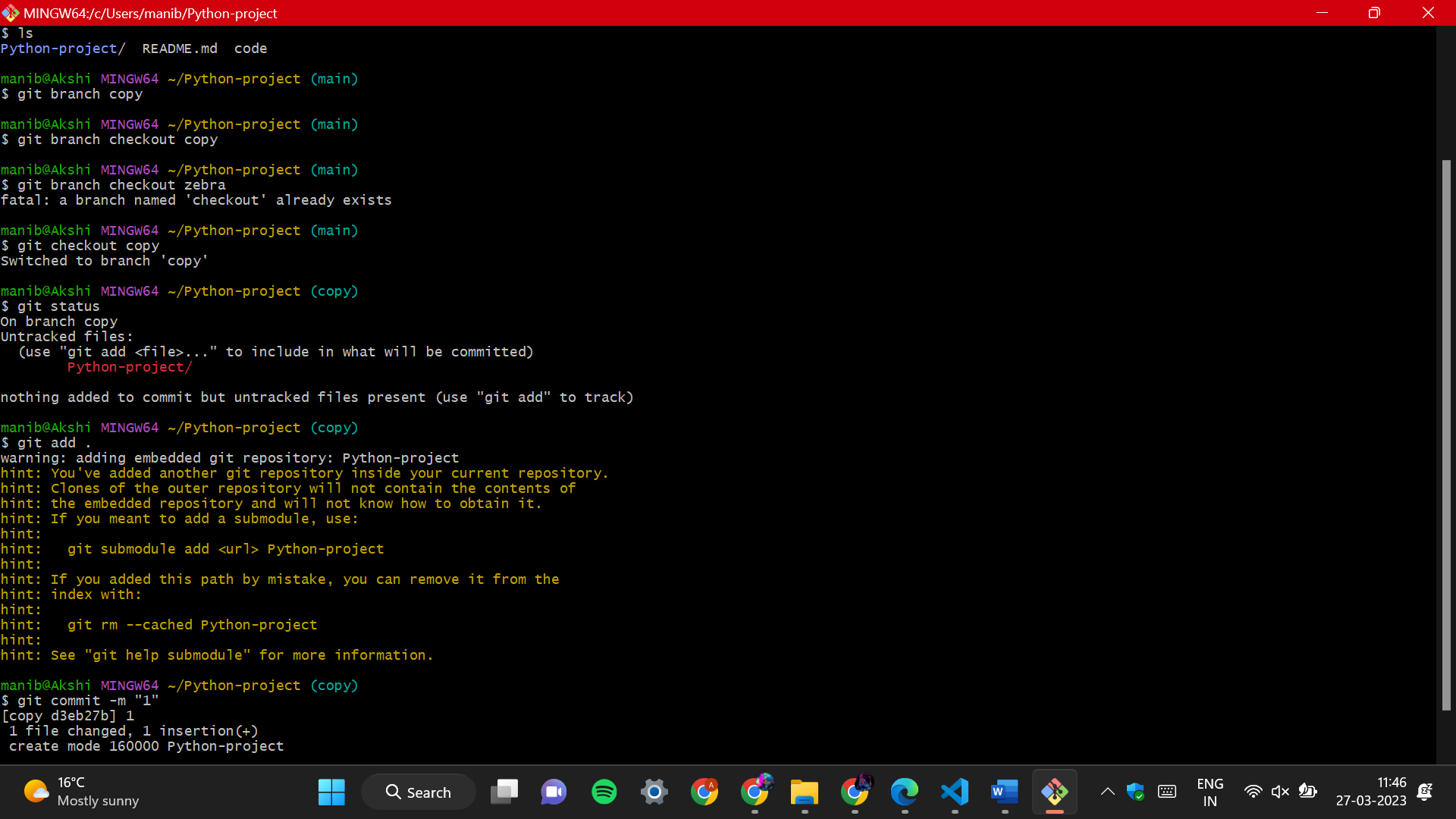
* Accept the invitation and hence you are ready to contribute to the project.

**OPEN AND CLOSE A PULL REQUEST**

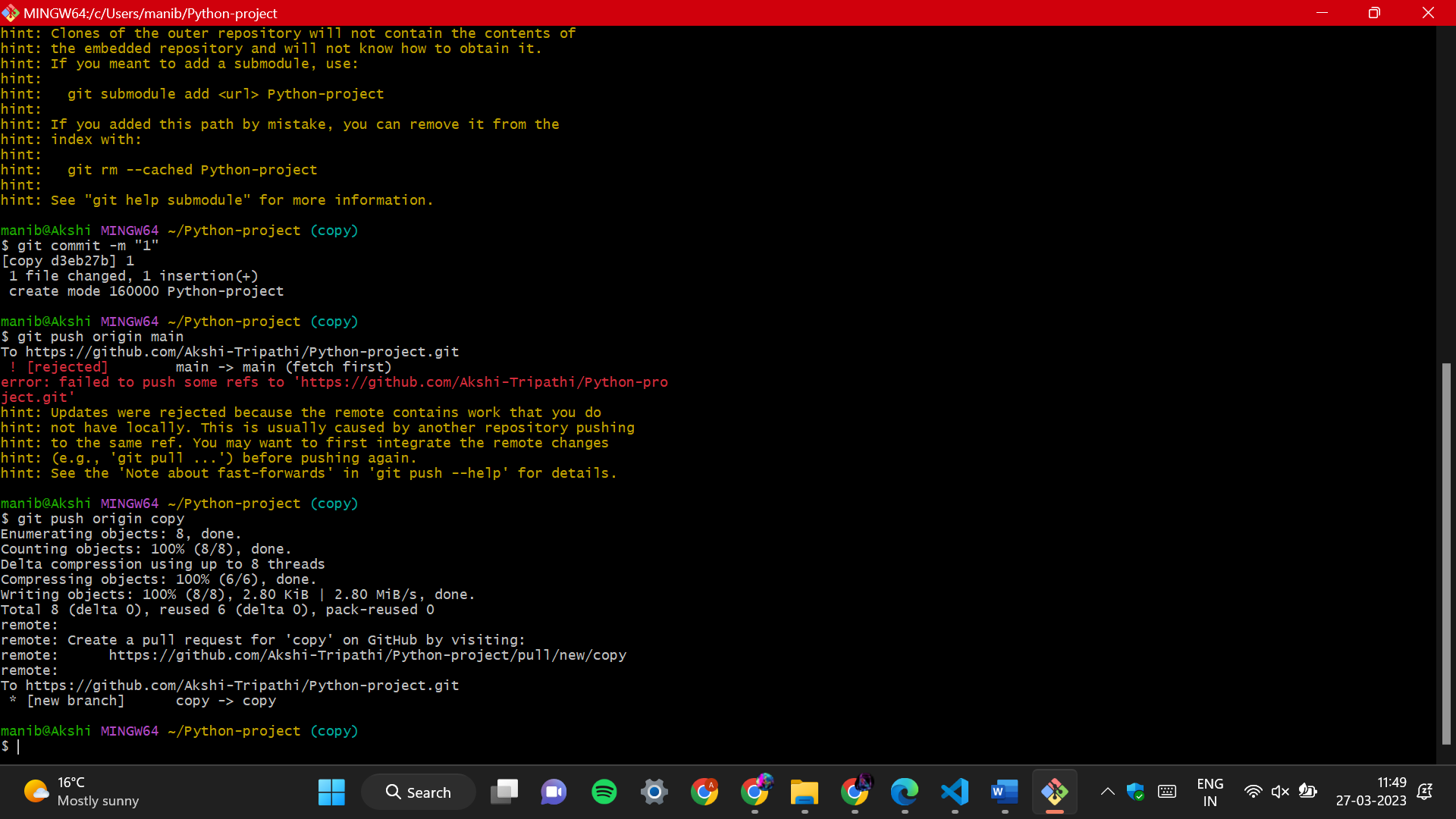
* + To open a pull request, we first have to make a new branch, by using git branch *branchname* option.



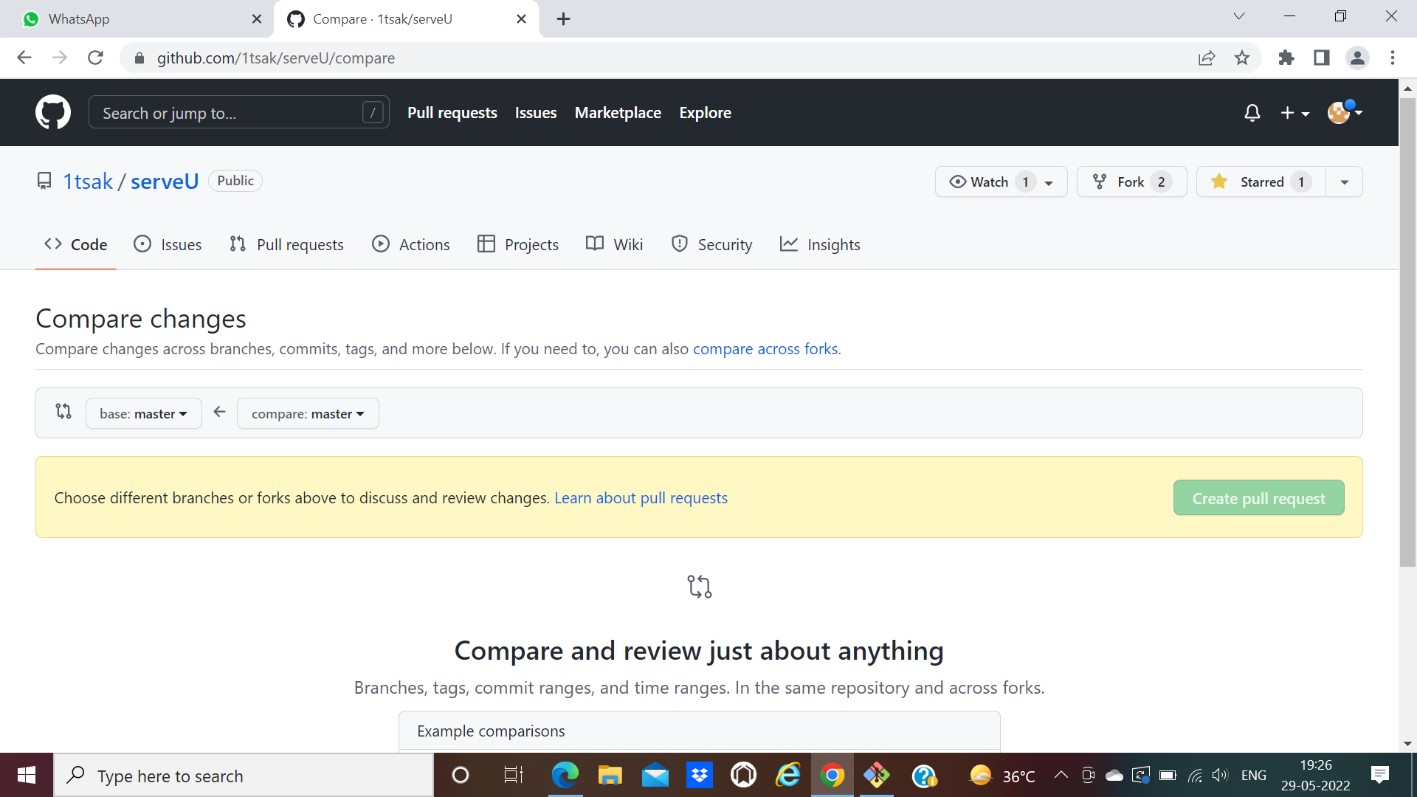
* + After making new branch we add a file to the branch or make changes in the existing file.



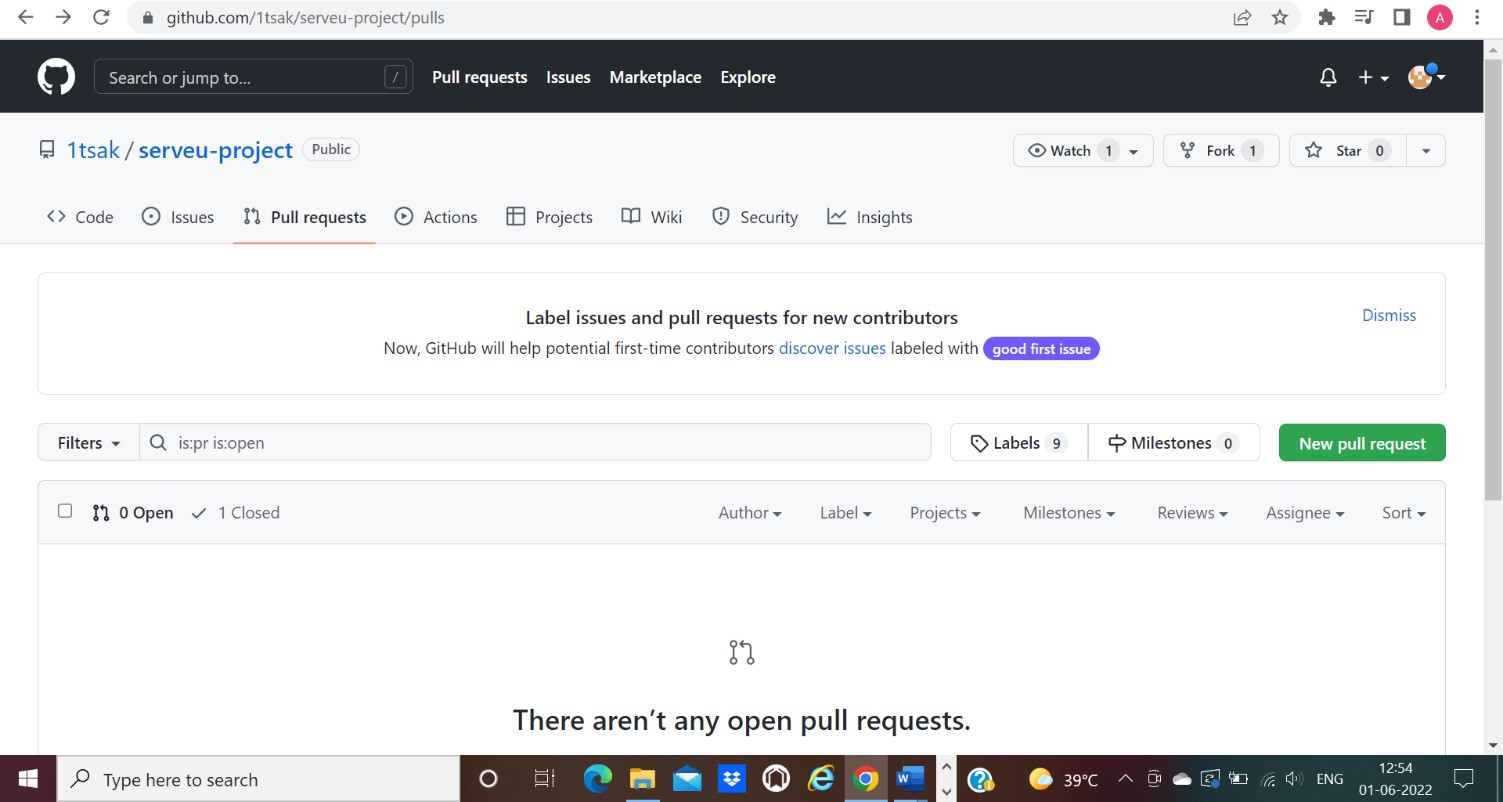
* + Add and commit the changes to the local repository.
  + Use git push origin *branchname* option to push the new branch to the main repository.



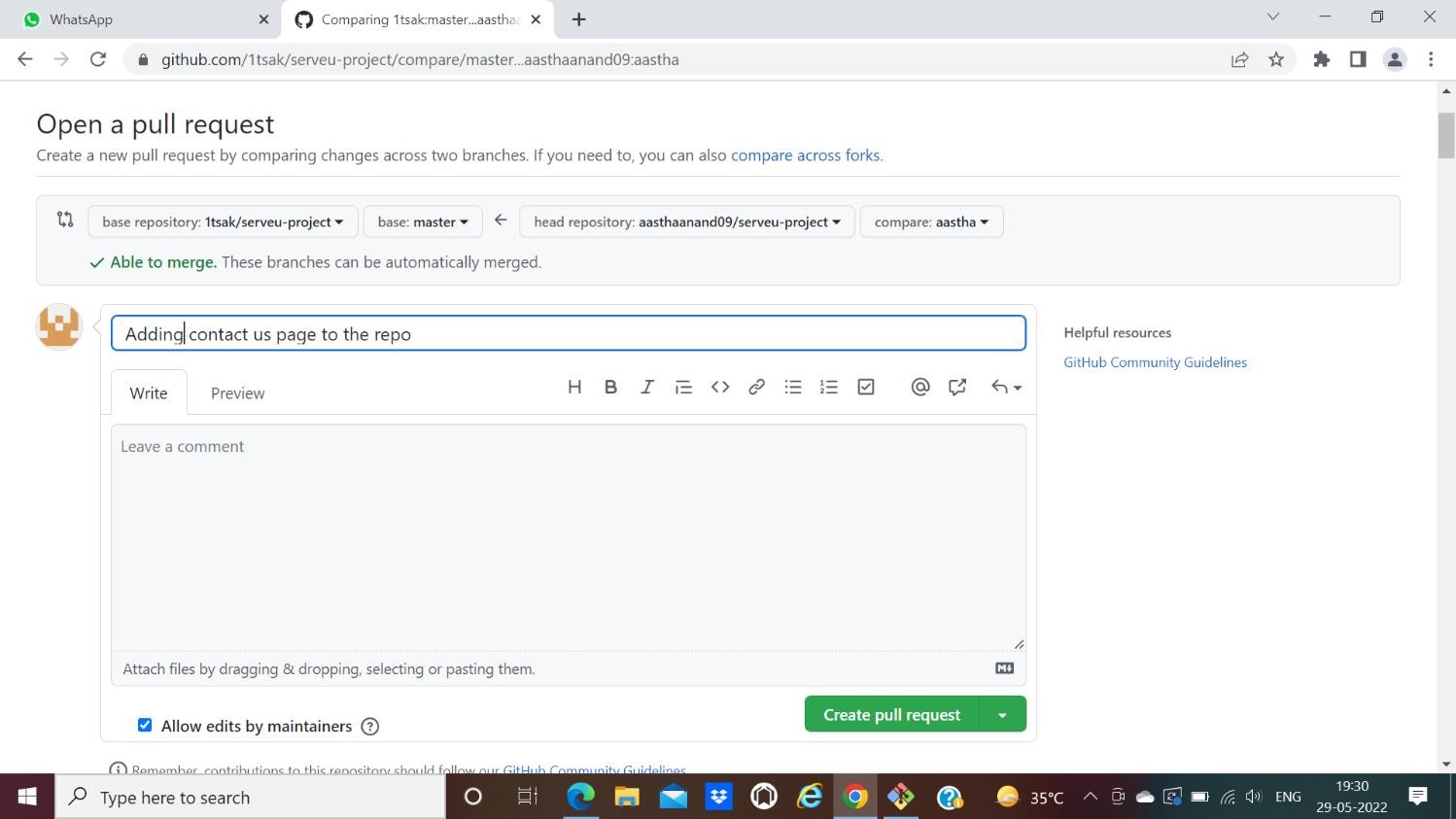
* After pushing new branch GitHub will either automatically ask you to create a pull request or you can create your own pull request.



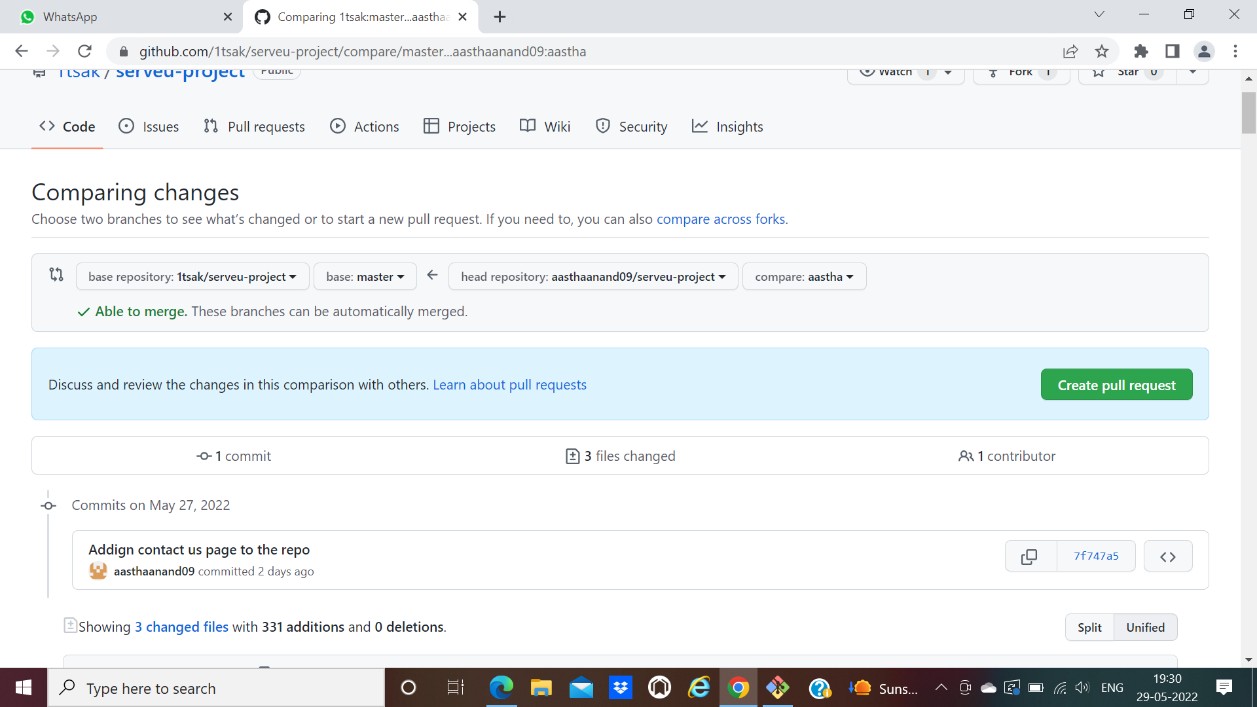
* To create your own pull request, click on pull request option.



* GitHub will detect any conflicts and ask you to enter a description of your pull request.



* After opening a pull request all the team members will be sent the request if they want to merge or close the request.



* If the team member chooses not to merge your pull request, they will close your pull request.
* To close the pull request simply click on close pull request and add comment/ reason why you closed the pull request.
* You can see all the pull request generated and how they were dealt with by clicking on pull request option.



