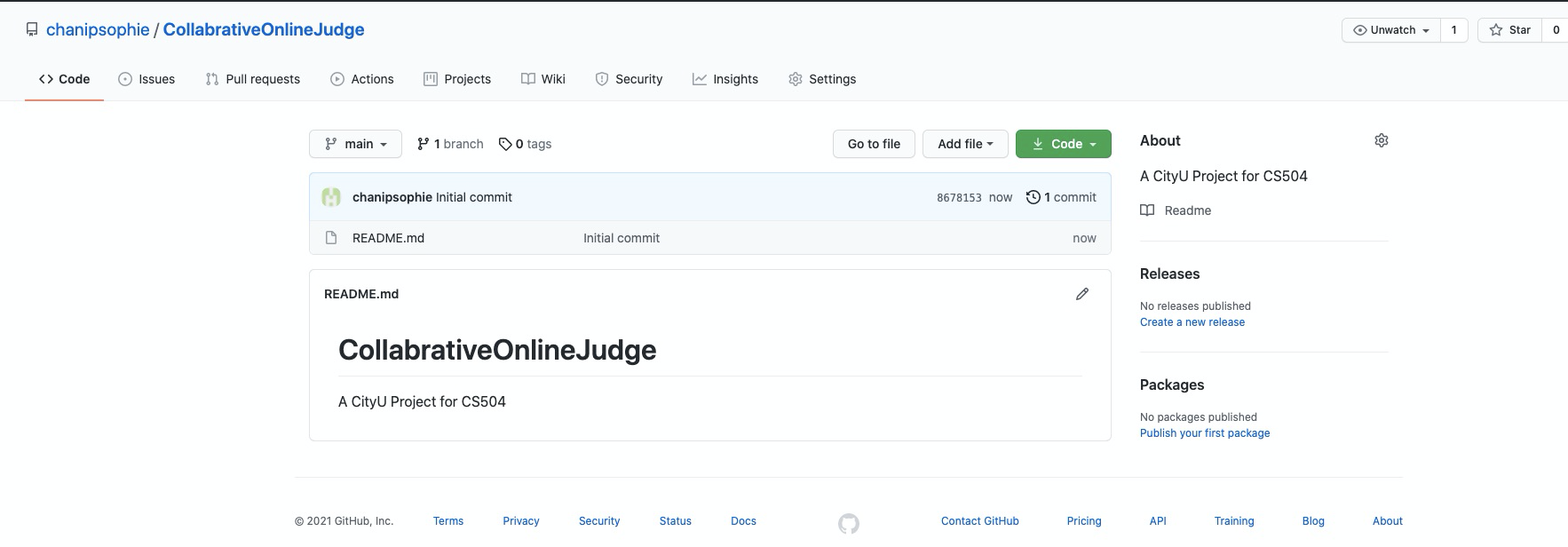
**CS504\_Independent Project1**

**Ip Chan**

* **Code check out**
* **Code check in**
* **Code snapshot (or release point)**
* **Code in development by two users, first one checks out code, second checks out code, makes changes and checks in, first makes changes and needs to check in.**
* **Code changes need to be rolled back to a previous level.**
* **One developer is working on a new version, one is fixing bugs in a previous release.**

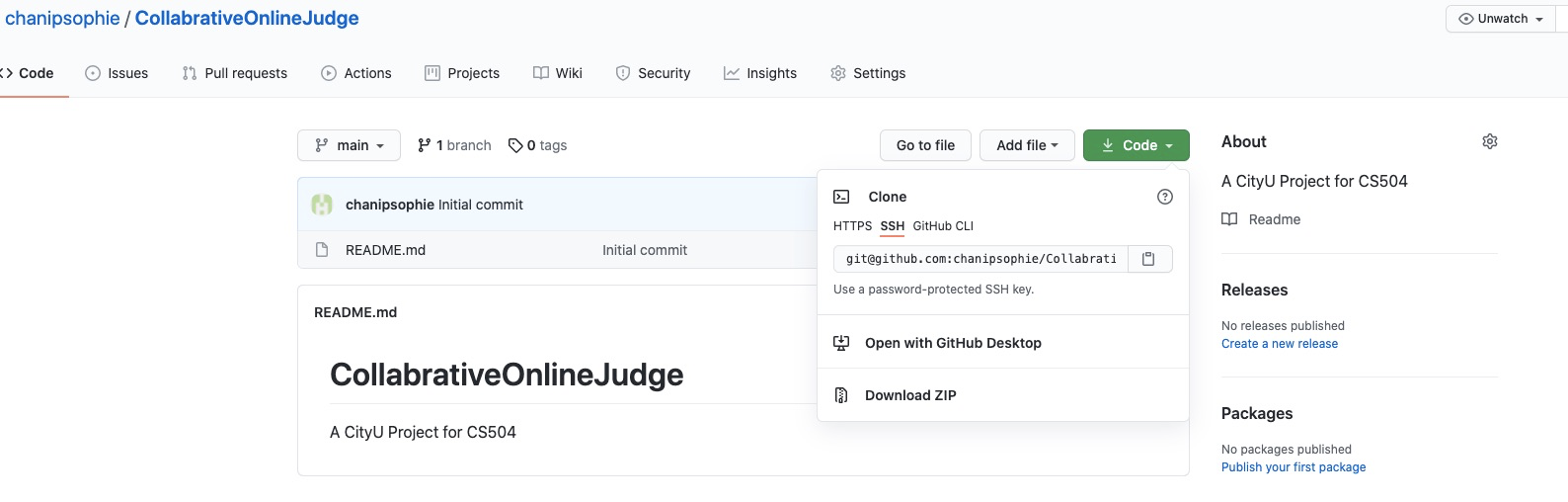
**I have created this repository “CollabrativeOnlineJudge” . This is a project to build a web-based collaborative code editor which supports multiple users editing simultaneously such as Google doc and Leetcode.**



Code checkout

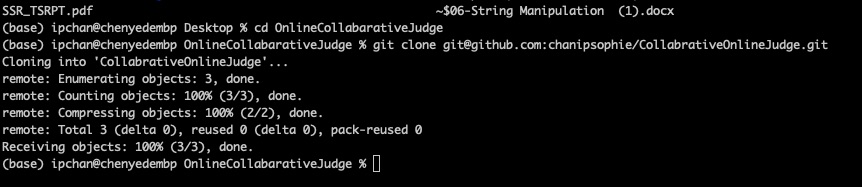
The first steps for this project is to configure the repository so that everyone in the project is able to check out/in the code. For code check out, I have created the repo with only one file, Readme.md, for now, and am demonstrating here how to checkout the code from GitHub to my local computer.

The screenshot below is an example of how I did “git clone” to my local Macbook. I have setup the ssh keys (public and private key pairs) so that I am able to do git related commands without entering certain credentials such as password manually. Therefore, the url is ssh url instead of https url.



The screenshot below is to show that I was able to clone the repo to my local Macbook.

chanipsophie checks out code



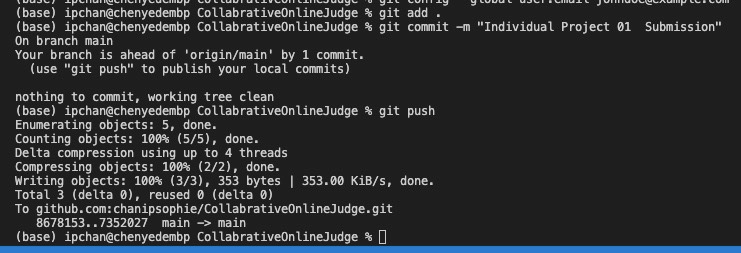
The screenshot below is to show that I was changing the file, and then I did git add and git commit to commit the changes. The commit messages themselves are also called “code snapshots”.

chanipsophie makes changes



The screenshot below is to show that I did git push to “check in” the code on to Github after I have committed the code changes locally.

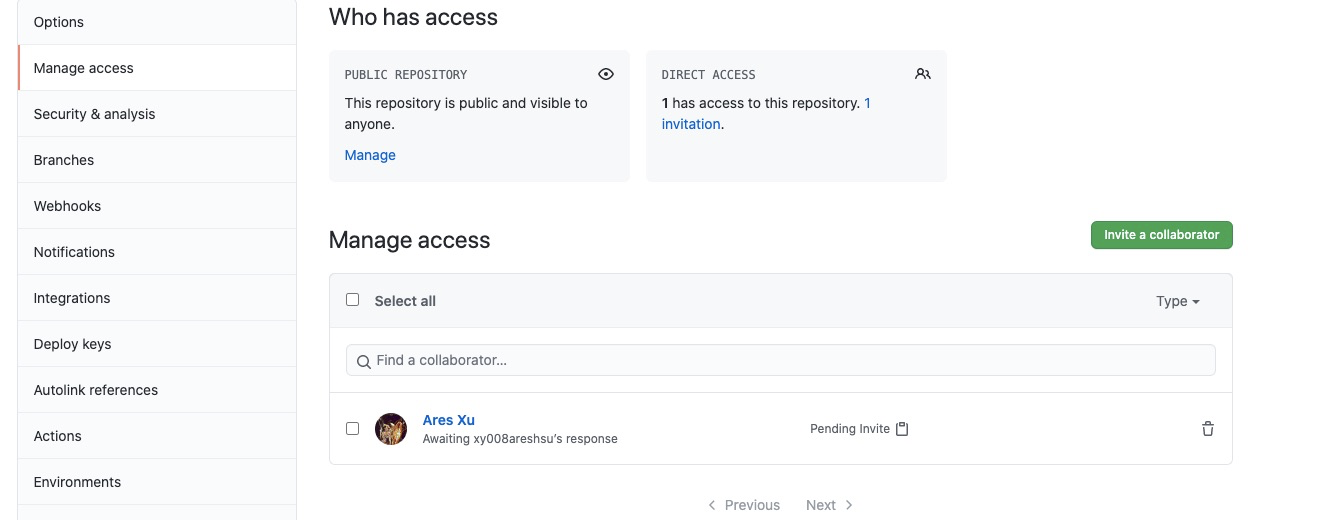
chanipsophie Code checkin

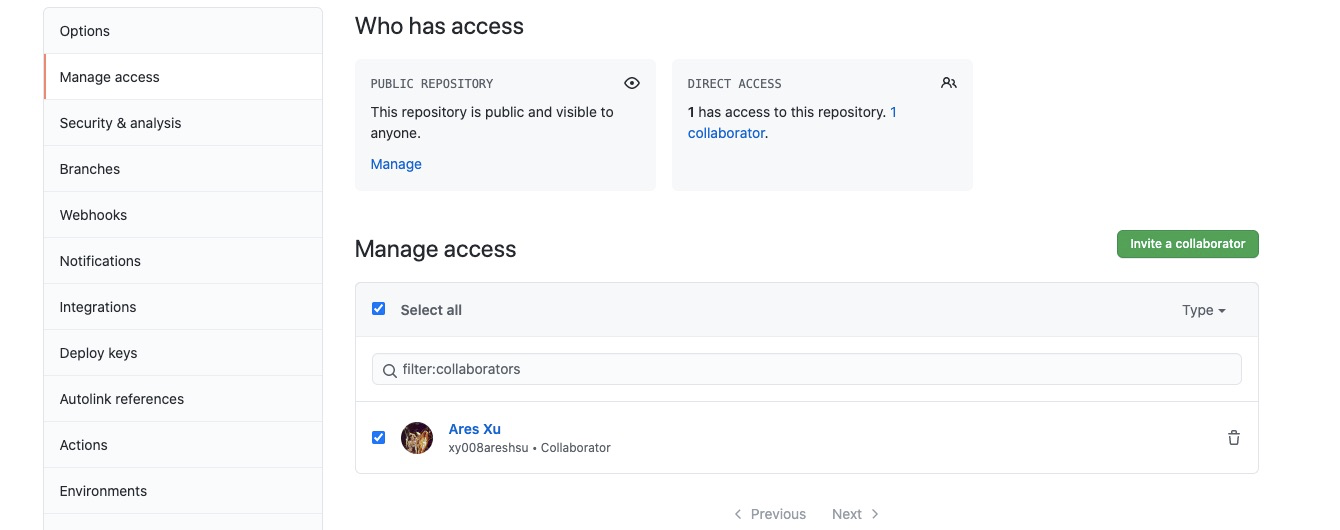


**Code in development by two users, first one checks out code, second checks out code, makes changes and checks in, first makes changes and needs to check in.**

At first I invited another collaborator, with the username xy008areshsu, to the project, like below.

Invite xy008areshsu as collabrator





This collaborator xy008areshsu checked out the code into his local computer, and then made some changes as well.

xy008areshsu checks out code:



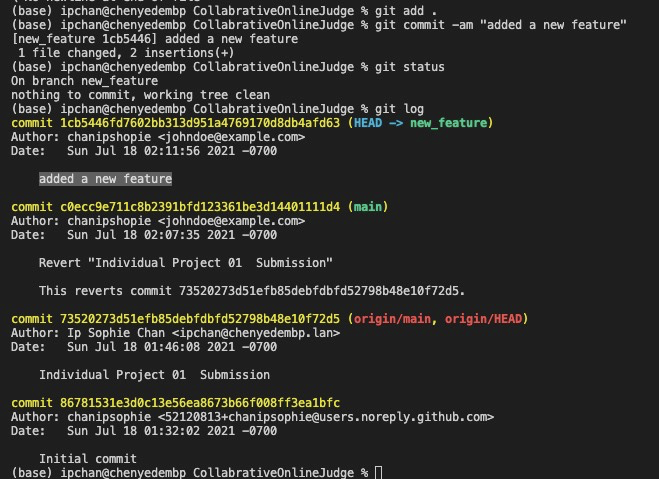
**Code changes need to be rolled back to a previous level**

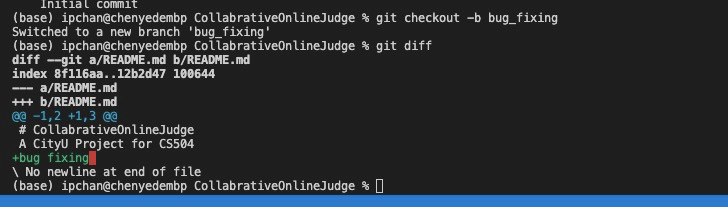
**With git, it is very easy to revert or rollback to a previous commit. One can execute “git revert CommitID” to revert that specific commit. The following screenshot shows that I reverted one of the committed code changes. Since git is an append only system for all commit histories, this “revert” commit also was appended to the previous commit, so that people in the future can easily track back what was going on.**

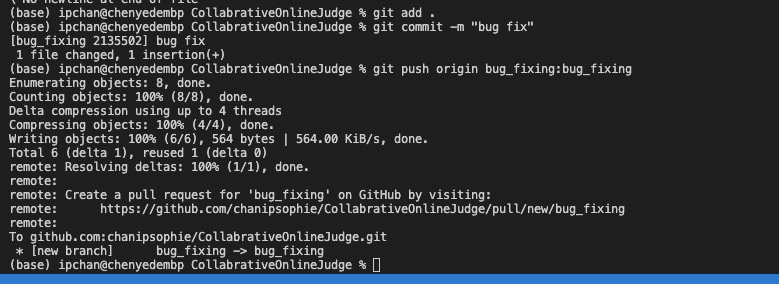


* One developer is working on a new version, one is fixing bugs in a previous release.

suppose the previous release is the main branch, below is the screenshot of one is working on a new feature in a new version. Here we can see that all of the changes are in the “new\_feature” branch. We used two different branches, new\_feature and bug\_fixing, to demonstrate that multiple developers can work together on different tasks without conflicting with the current release branch. bug\_fixing branch is for bug fixing, whereas new\_feature is for the task of adding a new feature. The changes in these two branches will not show up in the main branch until we manually request a “Pull Request” to merge the changes into the main branch.







Below is the screenshot of a sample pull request to merge the code changes from the bug\_fixing branch into the main branch.

