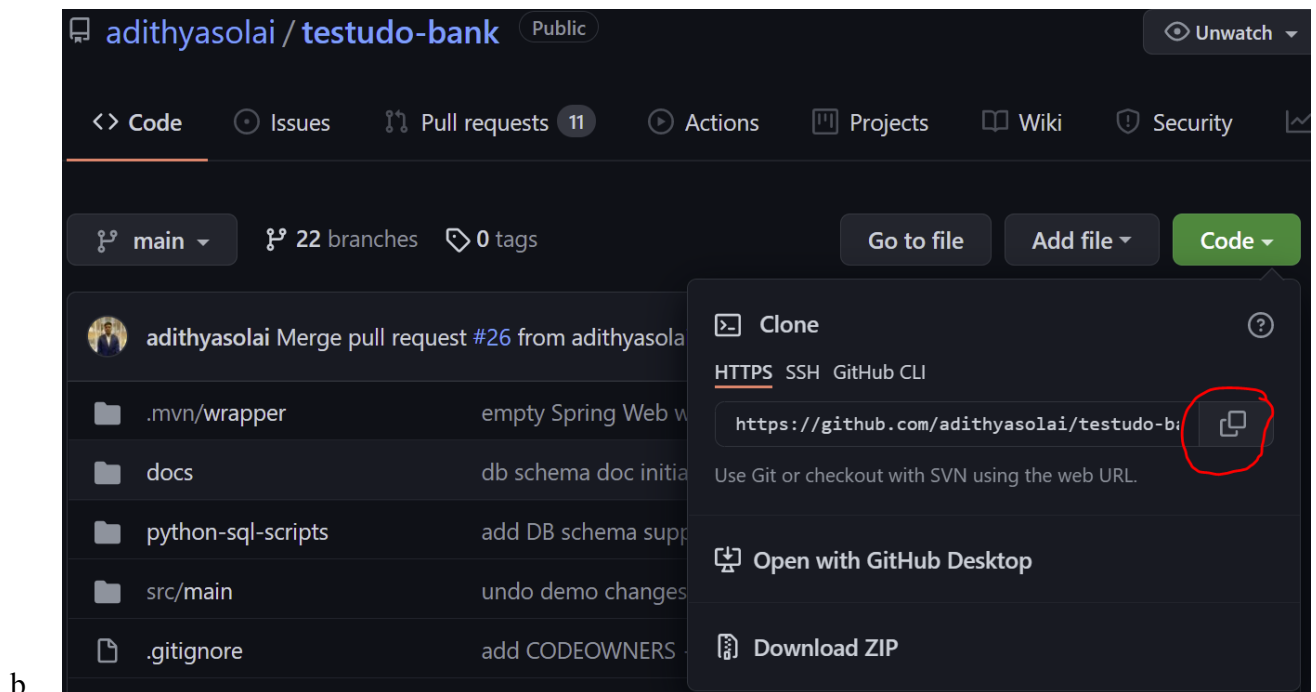
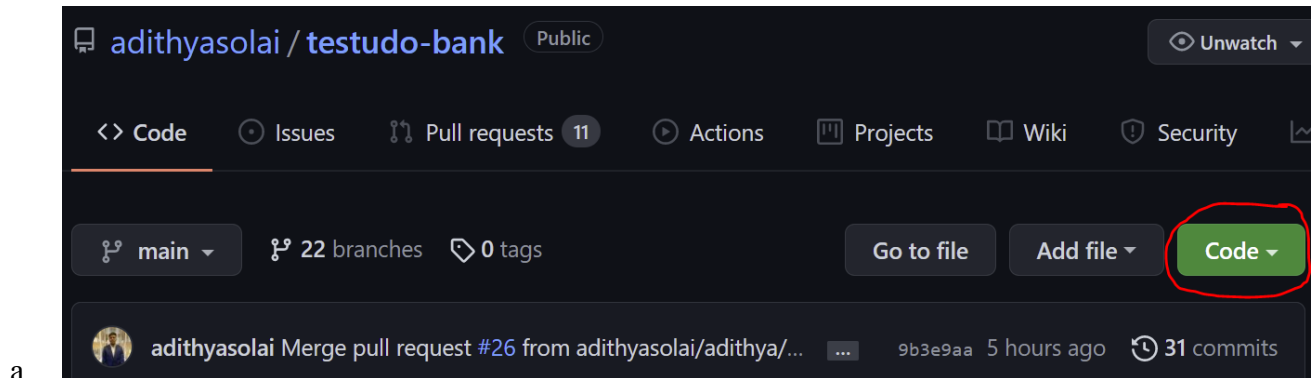
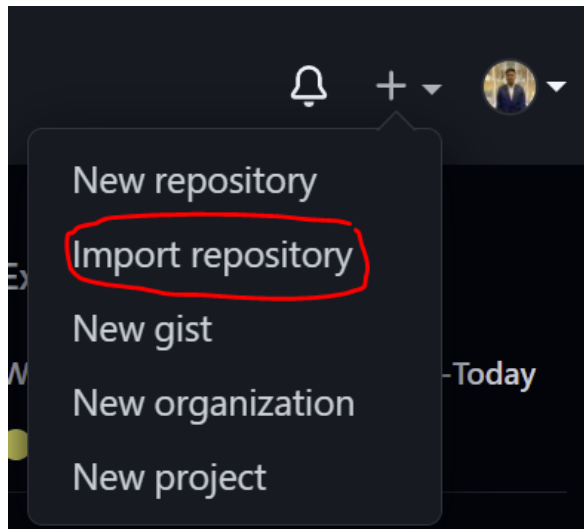


1. Go to <https://github.com/adithyasolai/testudo-bank>
2. Clone the public testudo-bank repo's Clone URL:



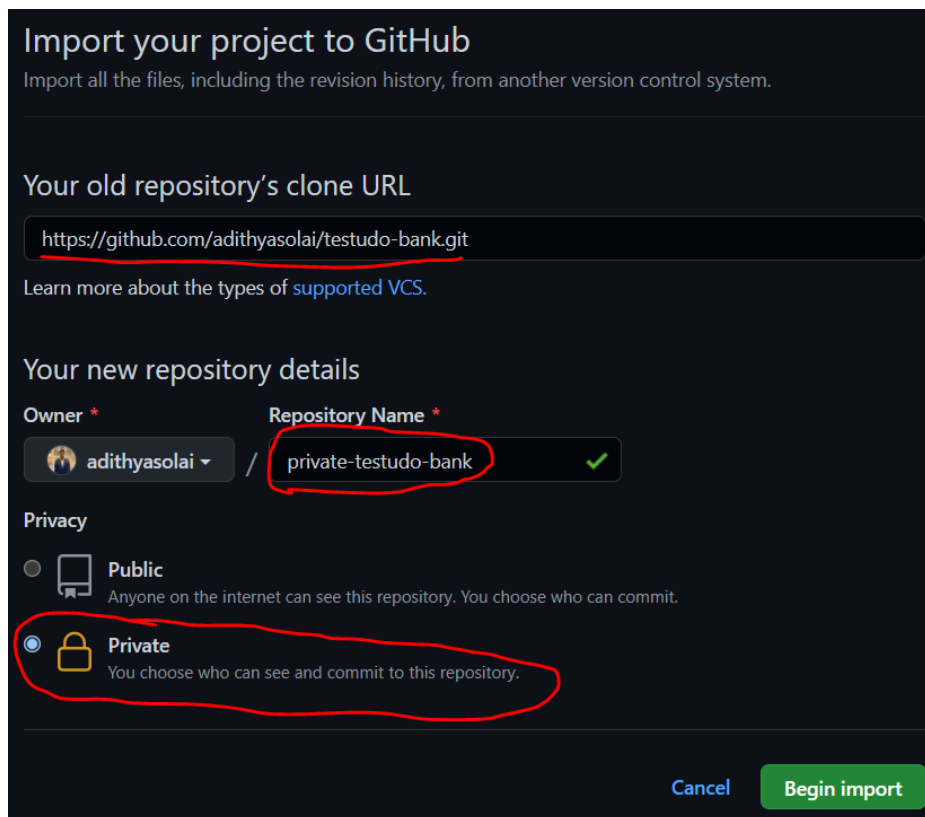
- c. (Or just use this clone URL: <https://github.com/adithyasolai/testudo-bank.git>)

3. Click on “Import repository” on the top right of GitHub



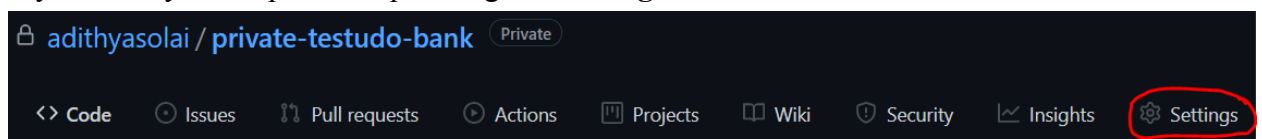
a.

4. Fill out the form **exactly** as described below, and then click “Begin Import”

A screenshot of the 'Import your project to GitHub' form. The form has a dark background with white text. The 'Your old repository's clone URL' field contains 'https://github.com/adithyasolai/testudo-bank.git' (underlined in red). The 'Your new repository details' section shows 'Owner' as 'adithyasolai' and 'Repository Name' as 'private-testudo-bank' (circled in red). The 'Privacy' section shows 'Public' and 'Private' options, with 'Private' selected (circled in red). At the bottom right, there are 'Cancel' and 'Begin import' buttons.

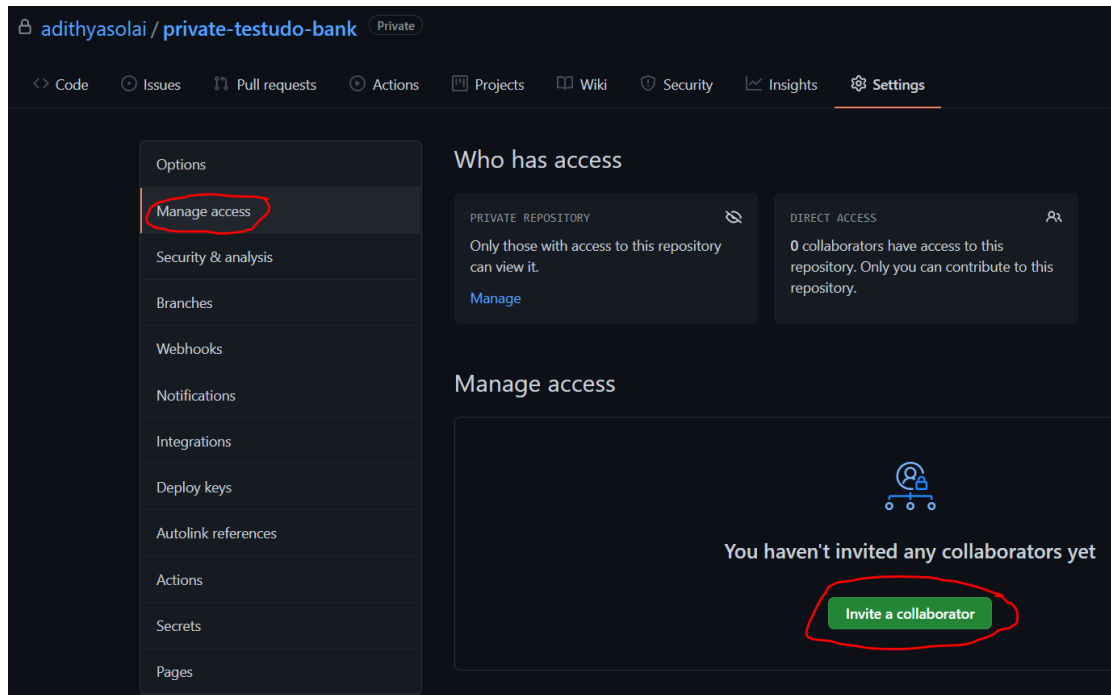
a.

5. Go to your newly-made private repo and go to **Settings**.



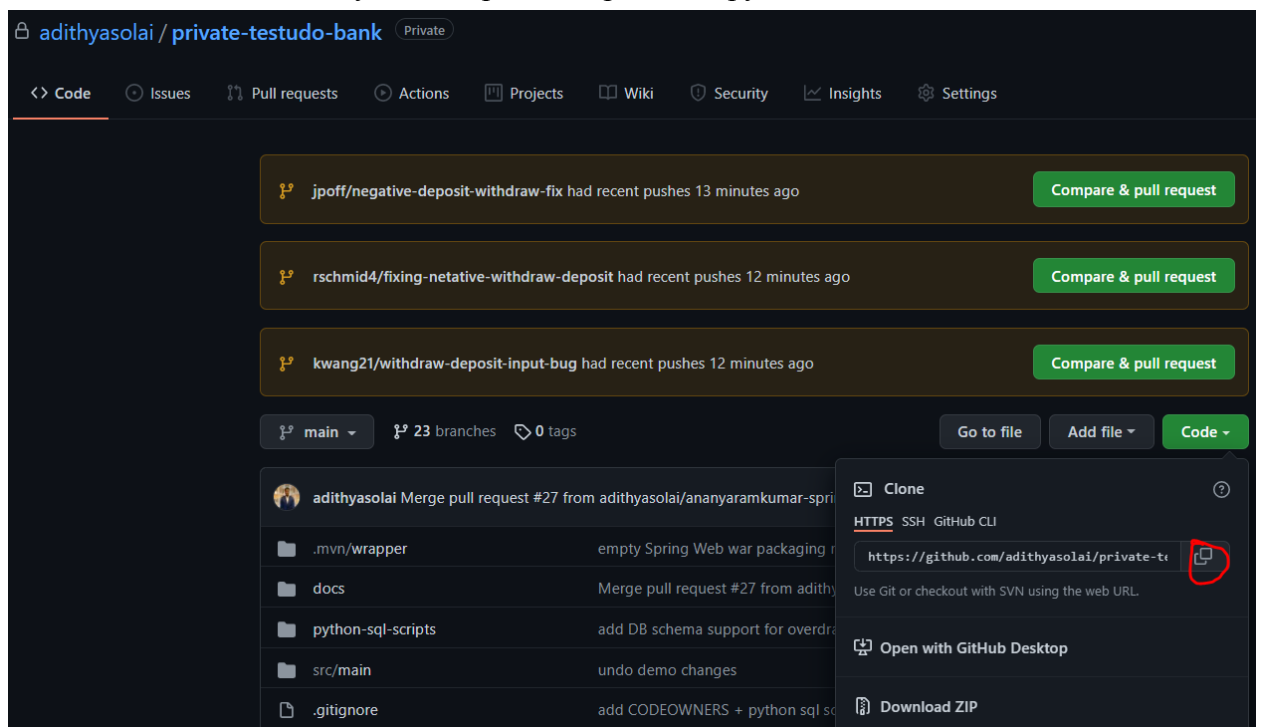
a.

6. Go to **Manage Access**, Click “Invite a collaborator”, and invite Ananya (GitHub username: ananyaramkumar) and Adithya (GitHub username: adithyasolai).



a.

7. Go back to the home screen of your new private repo and copy the Clone URL



a.

8. Open up the local repository you have been working in, and open up the bash terminal. Type **git remote**, and you should see only the **origin** remote.

```
pratb@DESKTOP-FSG5HP4 MINGW64 ~/CMSC389G-STIC/testudo-bank (adithya/demo)
$ git remote
origin
```

a.

9. Type '**git remote add private <Your Private Repo Clone URL>**' into the bash terminal. Type **git remote**, and you should now see a **private** remote.

```
pratb@DESKTOP-FSG5HP4 MINGW64 ~/CMSC389G-STIC/testudo-bank (adithya/demo)
$ git remote add private https://github.com/adithyasolai/private-testudo-bank.git

pratb@DESKTOP-FSG5HP4 MINGW64 ~/CMSC389G-STIC/testudo-bank (adithya/demo)
$ git remote
origin
private
```

a.

10. For Assignment #2, you will make your local changes, commit them, and then push to your **private** remote instead of the **origin** (public) remote.
- a. Ex: **git push private adithya/overdraft-feature**

IMPORTANT NOTES ON USING THE PRIVATE REPO:

- DO NOT CREATE A BRANCH OR PULL REQUEST IN THE PUBLIC REPO (<https://github.com/adithyasolai/testudo-bank>). OTHER STUDENTS WILL BE ABLE TO SEE AND COPY YOUR CODE!
 - Course Instructors will do their best to monitor and delete any branches opened on the public Testudo Bank Repo.
- NEVER EVER TYPE '**git push private main**'! There are no branch protections you can create on the **main** branch like we have in the public repo since GitHub doesn't support that feature for private branches unless you have a paid subscription.
 - Therefore, you will have to monitor yourself and never push a local **main** branch to your private repo's remote **main** branch.
 - If you ever type '**git push private main**', your code will get immediately merged into the codebase without review from Course Instructors, and you will also have to go through a messy process to revert the push.
- We will ask you to do a '**git pull origin main**' in your local repo if some major update is made to the public repo while you are working on Assignment #2.
 - In this case, you will then have to '**git push private <local branch name>**' to transfer these changes to your private repo.
 - You will have to open a PR in your private repo and request a PR review from a Course Instructor to merge in the changes from the public repo.
- Your final submission for Assignment #2 will be a URL link to your PR in your PRIVATE REPO that implements the Assignment #2 feature.