

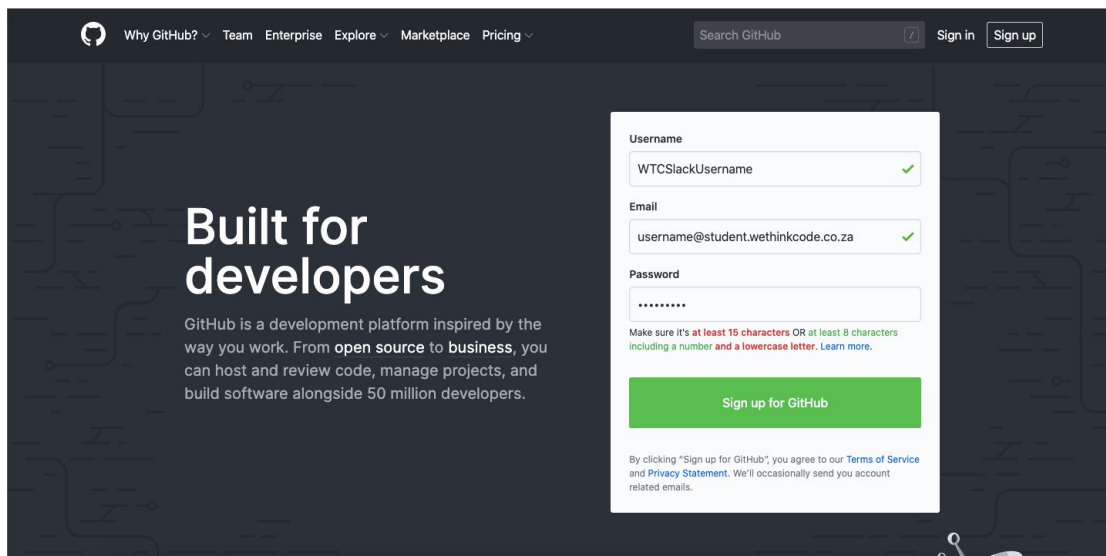
Getting Started

For the purposes of this document note the following terms:

Evaluator: The person doing the marking, who made a slot on intra and was assigned to the evaluatee when that evaluatee subscribed for defense.

Evaluatee: The person being marked.

1. Open Google Chrome (or any other preferred browser).
2. Type in the URL for Github, and hit enter.
3. On the github page, fill in the required fields using your student information. **Please use the format usernameCAMPUS. Eg jwolfJHB.**
(is your name is already in use you may use a series of numbers after the campus part until it is original)



Why GitHub? Team Enterprise Explore Marketplace Pricing

Search GitHub Sign in Sign up

Built for developers

GitHub is a development platform inspired by the way you work. From **open source** to **business**, you can host and review code, manage projects, and build software alongside 50 million developers.

Username
WTCSlackUsername ✓

Email
username@student.wethinkcode.co.za ✓

Password
.....

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)

Sign up for GitHub

By clicking "Sign up for GitHub", you agree to our [Terms of Service](#) and [Privacy Statement](#). We'll occasionally send you account related emails.

4. Answer the Recaptcha message.
5. Answer the preference questions based on your own interests or reasons for using Github, and once you're done, navigate to the next page.
6. Verify the email address you have linked to your Github account by accessing your student email and clicking on the link that has been sent to you.



Please verify your email address

Before you can contribute on GitHub, we need you to verify your email address.

An email containing verification instructions was sent to zain@wethinkcode.co.za.

[Resend verification email](#)[Change your email settings](#)

GitHub

Subscribe to our newsletter

Get product updates, company news, and more.

[Subscribe](#)

Product

[Features](#)[Security](#)[Team](#)[Enterprise](#)[Customer stories](#)[Pricing](#)[Resources](#)

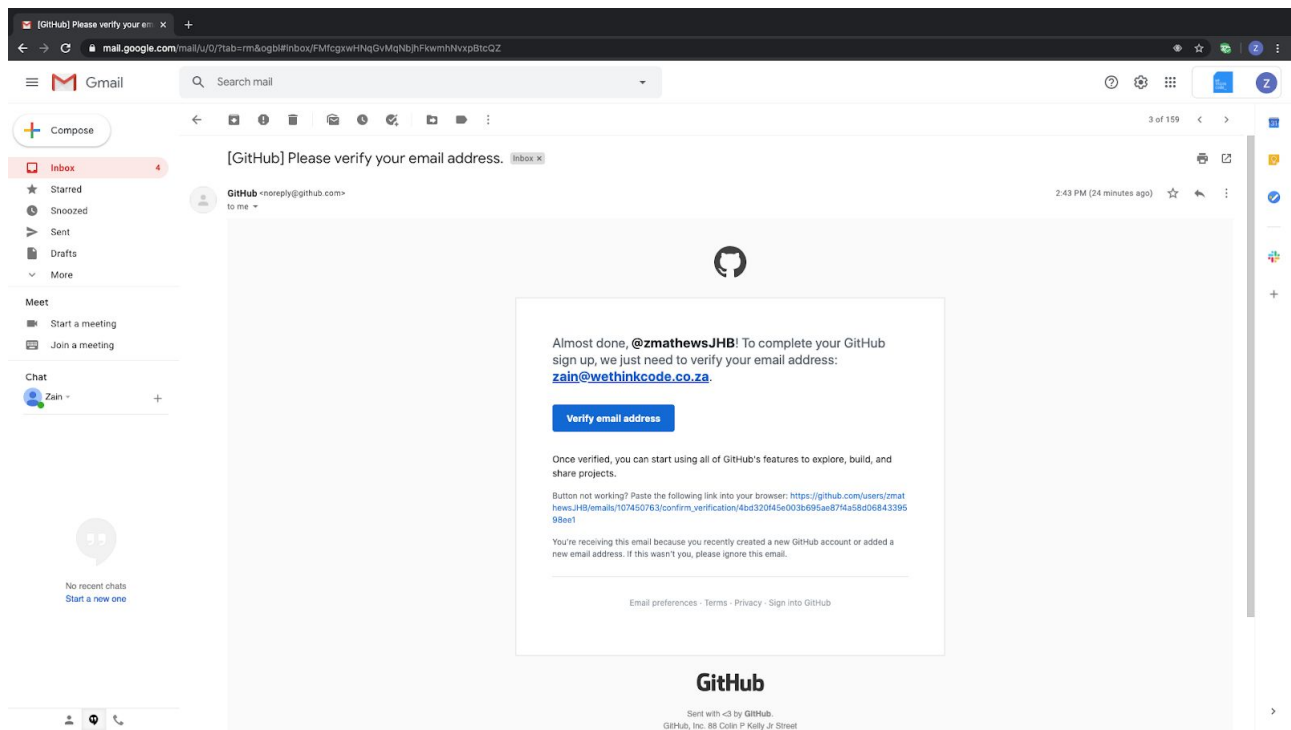
Platform

[Developer API](#)[Partners](#)[Atom](#)[Electron](#)[GitHub Desktop](#)

Support

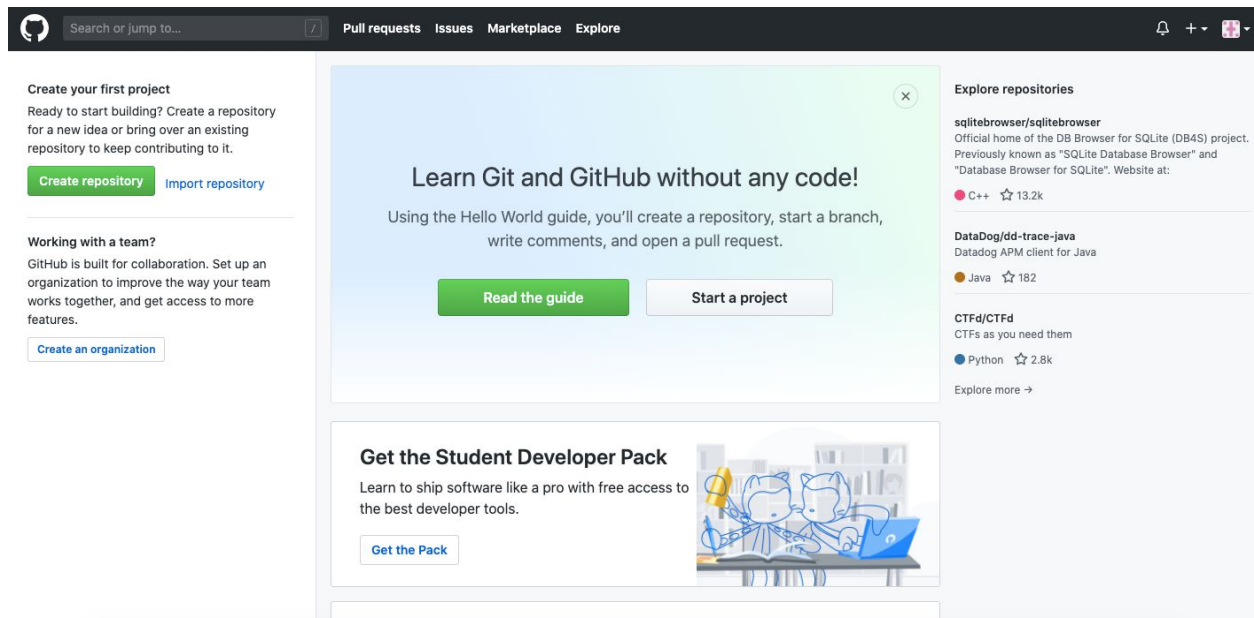
[Help](#)[Community Forum](#)[Professional Services](#)[Learning Lab](#)[Status](#)[Contact GitHub](#)

Company

[About](#)[Blog](#)[Careers](#)[Press](#)[Social Impact](#)[Shop](#)

Creating a Repository

1. Click on the button “Create Repository” on the left hand side of the home page.

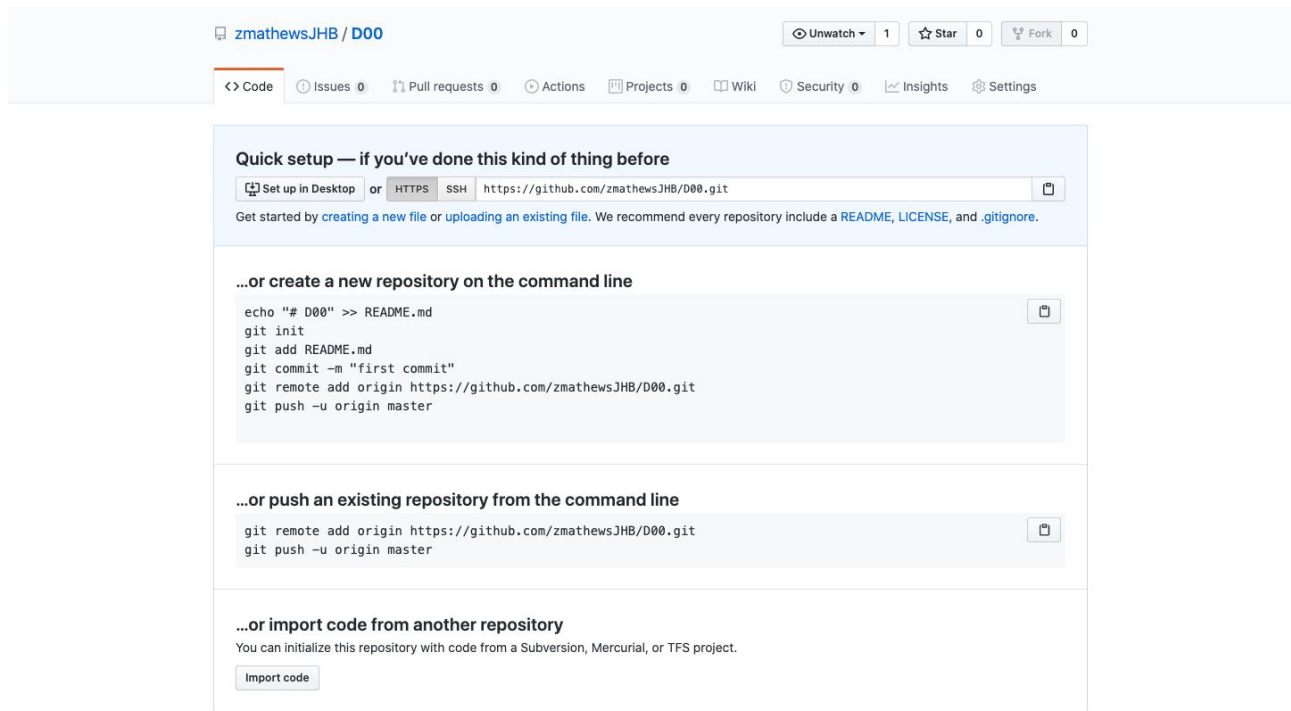


2. Add the name of your repository, and a description of what your repository holds to make it easier for both yourself and your peers to navigate through your project.

A screenshot of the 'Create a new repository' form on GitHub. The form includes fields for 'Owner' (zmathewsJHB) and 'Repository name' (D00). Below these is a 'Description (optional)' field containing 'WTC Bootcamp Day00 exercises'. There are radio buttons for 'Public' (selected) and 'Private'. At the bottom, there are checkboxes for 'Initialize this repository with a README' and 'Add .gitignore' and 'Add a license' dropdown menus. A green 'Create repository' button is at the bottom.

Cloning a Repository

1. Copy the link to your repository by clicking on the small clipboard icon to the right of your repo link.



2. Select and or create the directory you would like to work in, and change into that directory.
3. In the command line of your text editor, type **git clone** followed by the url of your repository and hit enter to begin downloading. You may rename the folder at this point should you wish to do so as follows: **git clone [repository link] name_of_folder**

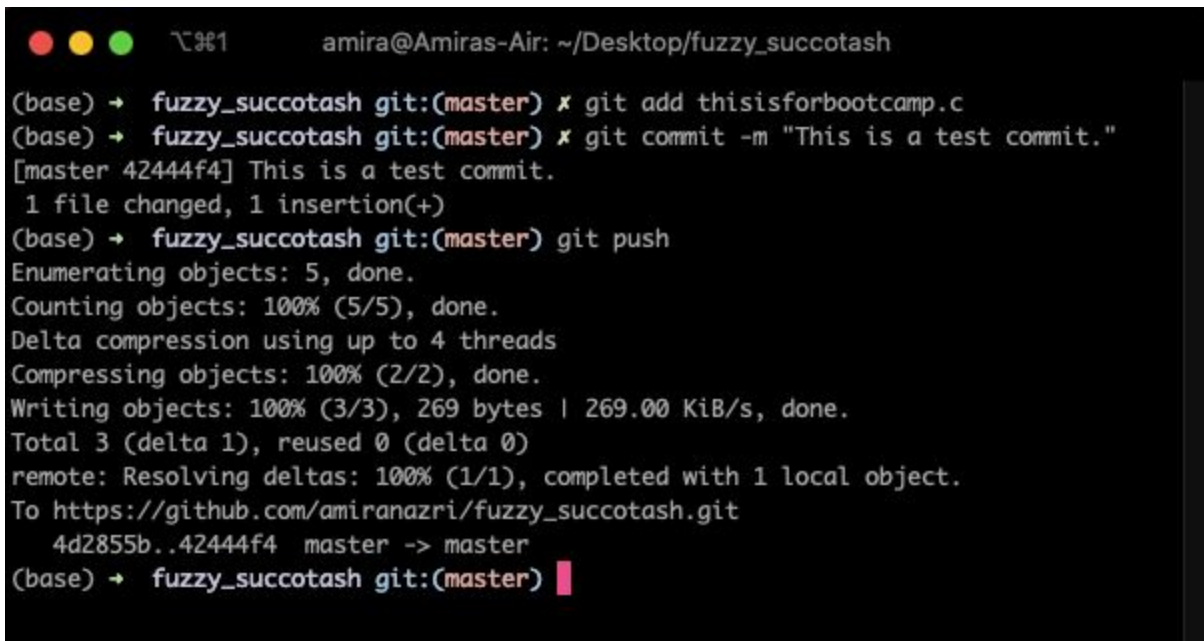
Please note: you must create/save your files within the same repository you have just downloaded in order to push successfully.

```
amira@Amiras-Air: ~/Desktop
(base) → ~ cd Desktop
(base) → Desktop git clone https://github.com/amiranazri/fuzzy_succotash.git
Cloning into 'fuzzy_succotash'...
remote: Enumerating objects: 106, done.
remote: Counting objects: 100% (106/106), done.
remote: Compressing objects: 100% (99/99), done.
remote: Total 106 (delta 6), reused 96 (delta 4), pack-reused 0
Receiving objects: 100% (106/106), 6.41 MiB | 1.54 MiB/s, done.
Resolving deltas: 100% (6/6), done.
(base) → Desktop
```

Commit To Your Repository

There are three main steps used to commit work into your repository, however, git is not limited to these three commands.

1. Add your work → this is done by typing **git add** followed by the name of the file you would like to push. If you would like to push all items in your repository, you can use the "." character to indicate the adding of **all files in your current working directory**.
2. Commit your work → this is done by typing **git commit -m** followed by a (helpful) comment or message for yourself and or anyone else viewing your project and its commits. Make sure to write this comment within quotation marks, (ex: **git commit -m "First commit: created print function"**).
3. Push your work → this is simply done by typing **git push**, and hitting enter.

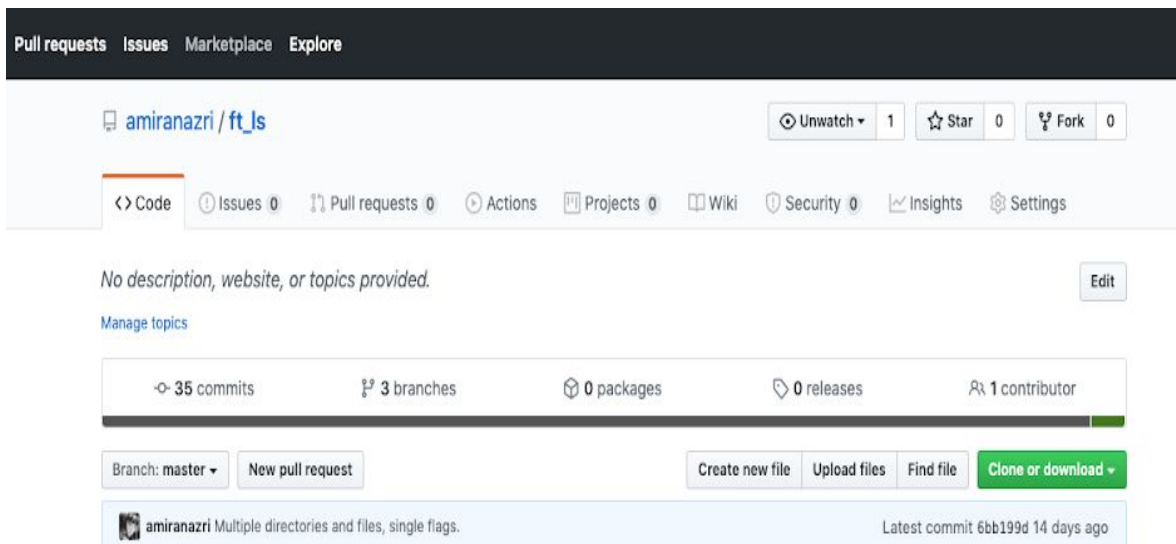
A terminal window with a dark background and light-colored text. The window title is 'amira@Amiras-Air: ~/Desktop/fuzzy_succotash'. The terminal shows a series of git commands and their outputs. The commands are: 'git add thisisforbootcamp.c', 'git commit -m "This is a test commit."', and 'git push'. The outputs include commit hashes, file change statistics, and push progress details. The terminal ends with a prompt for the next command.

```
amira@Amiras-Air: ~/Desktop/fuzzy_succotash
(base) → fuzzy_succotash git:(master) x git add thisisforbootcamp.c
(base) → fuzzy_succotash git:(master) x git commit -m "This is a test commit."
[master 42444f4] This is a test commit.
 1 file changed, 1 insertion(+)
(base) → fuzzy_succotash git:(master) git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 269 bytes | 269.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/amiranazri/fuzzy_succotash.git
 4d2855b..42444f4 master -> master
(base) → fuzzy_succotash git:(master) █
```

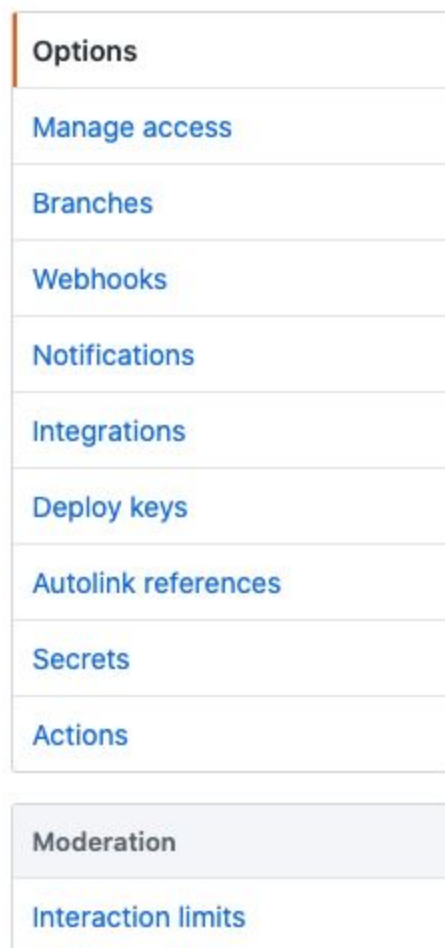
Using Github For An Evaluation

It is important you follow these steps in setting up your github for evaluations. During your Eval you'll be using 2 extra tools outside of intra and your terminal. Github, and meets.

1. Once you've created your repository, you can access the settings where you should set your repository to private. Your evaluator cannot view it unless you add her as a collaborator.

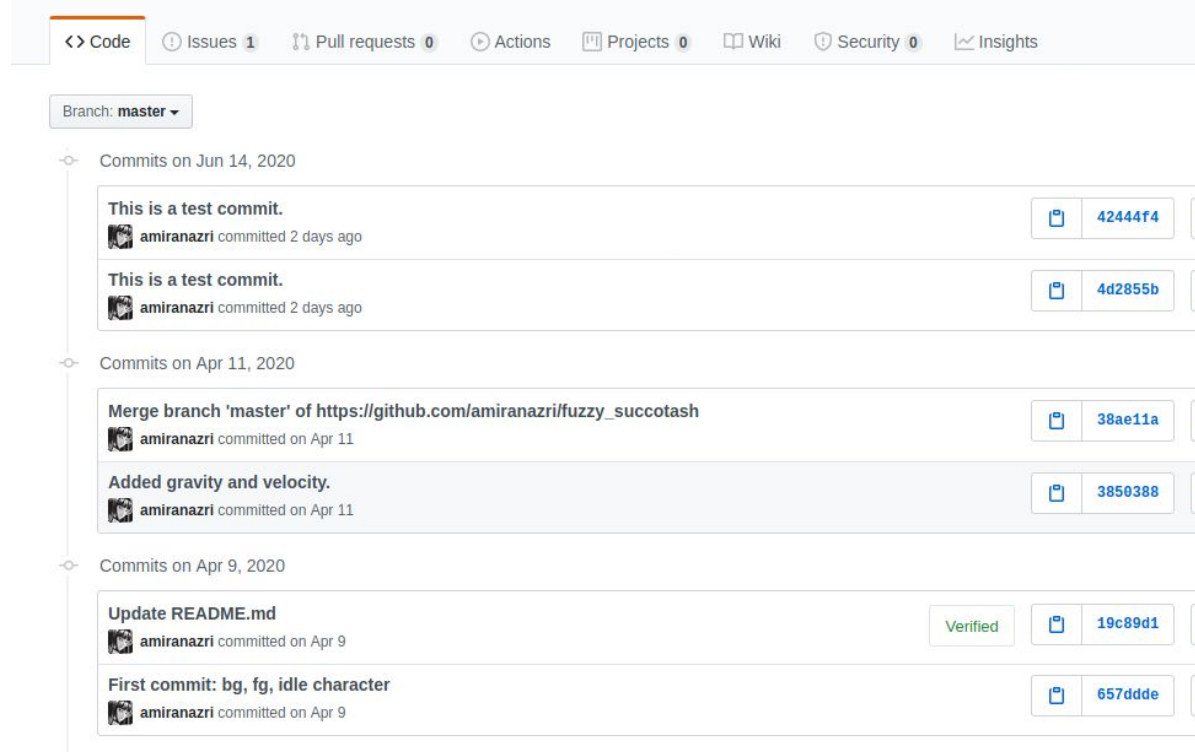


- Start by clicking the settings button on the right hand side of your repository menu bar.
- Click Options on the menu on the left side of the page.



- Navigate to the bottom of the page, to the section named **Danger Zone** and click on the button **Make Private**.

2. Under the same settings menu there is an option to “Manage access”. Click on it
 - Add staff members **anazri**, **zedmathews**, **Imkambela**, and **dstiekem** to your repository.
 - Also add the usernames of your evaluators **when the time comes**. Be careful about who you share your repository with and for what reason.
3. During eval you may want to clone your evaluatee’s repository from the time it was “set as finished” and not some time after.
 - Go to the project page of your evaluatee on intra. If you hover over the string “...was locked...and closed” you should get the exact time the project was set as finished. Note the time.
 - Head over to their repository on github and click on “commits” while the branch is on master. Find the commit with the timestamp that best matches their one on intra.



-
- Clone that commit.



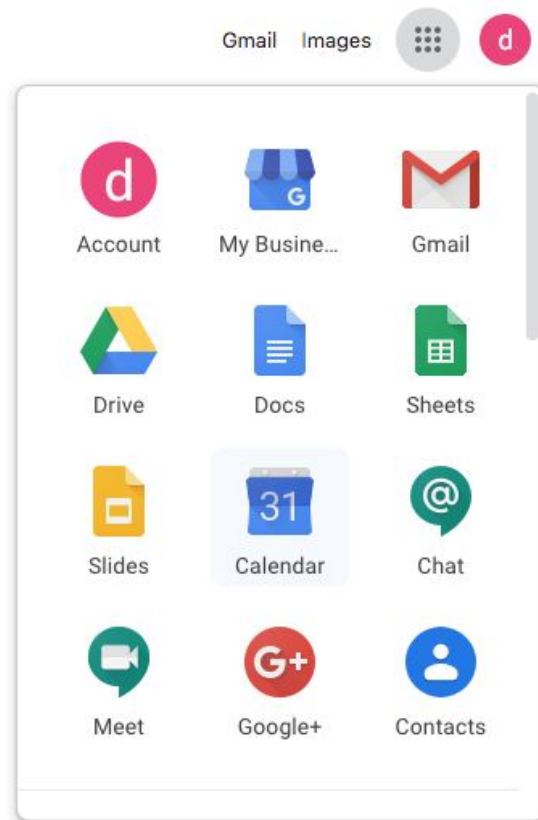
-

Next...The Meet

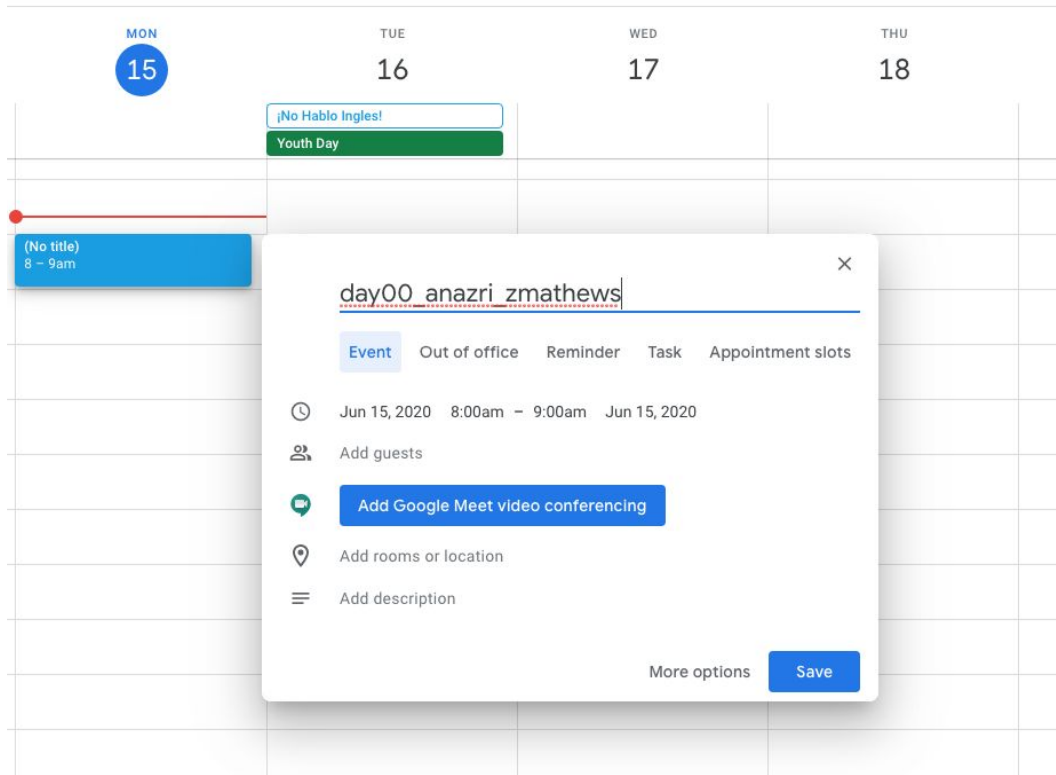
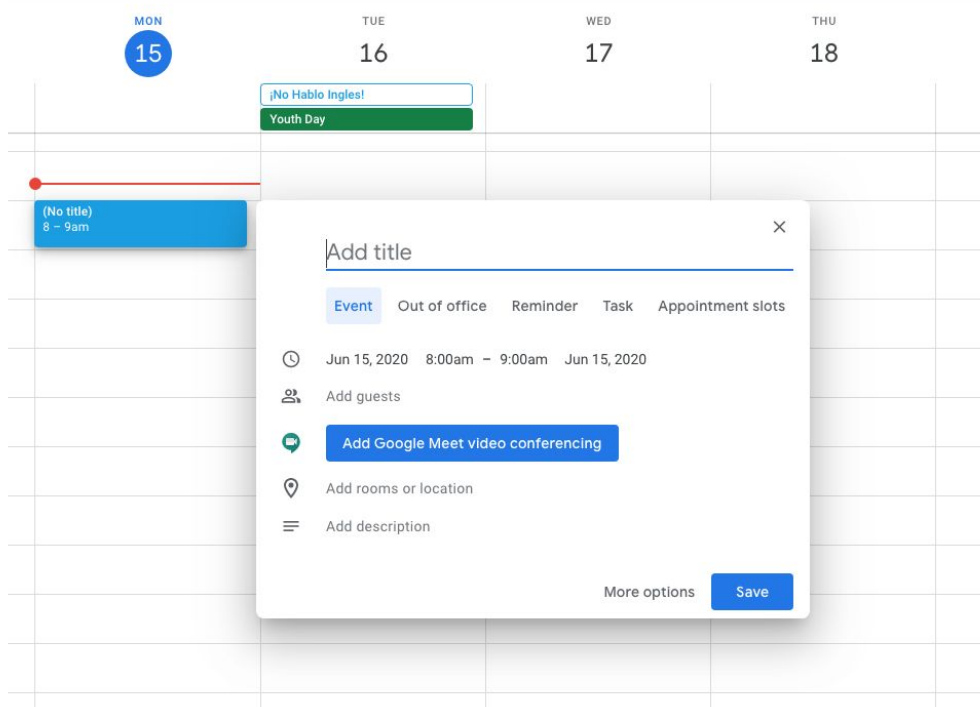
As the evaluatee you'll want to be sure you can actually interact with your evaluator or else we will have little evidence your evaluation took place at all. You'll use meets for this.

1. Once you have pushed your project, set up a calendar event for your evaluation. **It is your job to do this as soon as possible**. In order for the project to be considered complete you

need to evaluate 3 different people on their projects and be evaluated three times yourself. Both you and your evaluator must be present 15 minutes before the evaluation starts.



- create a meeting with a meets link for you and your evaluator. Name the event with the format **Project_evaluator_evaluatee**. Eg. **Day00_Imkambela_anazri**
- If the meeting is cut short or needs to be scheduled again do so with another calendar event and start versioning them **Day00_Imkambela_anazri1**, **Day00_Imkambela_anazri2**.
-





Create

SUN

14

June 2020



GMT+02

S	M	T	W	T	F	S
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	1	2	3	4
5	6	7	8	9	10	11

Meet with...

7 AM

8 AM

9 AM

10 AM

11 AM



day00_anazri_zmathews

Event

Out of office

Reminder

Task

Appointment slots



Jun 15, 2020 8:00am - 9:00am Jun 15, 2020



anazri@student.wethinkcode.co.za



Add Google Meet video conferencing



Add rooms or location



Add description

More options

Save

day00_anazri_zmathews



Event

Out of office

Reminder

Task

Appointment slots



Jun 15, 2020 8:00am – 9:00am Jun 15, 2020



Add guests

[See guest availability](#)

[Suggested times](#)



dominique@wethinkcode.co.za

Organizer



anazri@student.wethinkcode.co.za



Join with Google Meet



meet.google.com/roo-pubw-ije

Up to 250 participants [?](#)



Add rooms



Add location



Add description

More options

Save

Youth Day

day00_anazri_zmathews

Event

Out of office

Reminder

Task

Appointment slots



Jun 15, 2020 8:00am - 9:00am Jun 15, 2020



zmathews@student.wethinkcode.co.za

[See guest availability](#)

[Suggested times](#)



dominique@wethinkcode.co.za
Organizer



anazri@student.wethinkcode.co.za



Join with Google Meet



meet.google.com/roo-pubw-ije
Up to 250 participants ⓘ



Add rooms



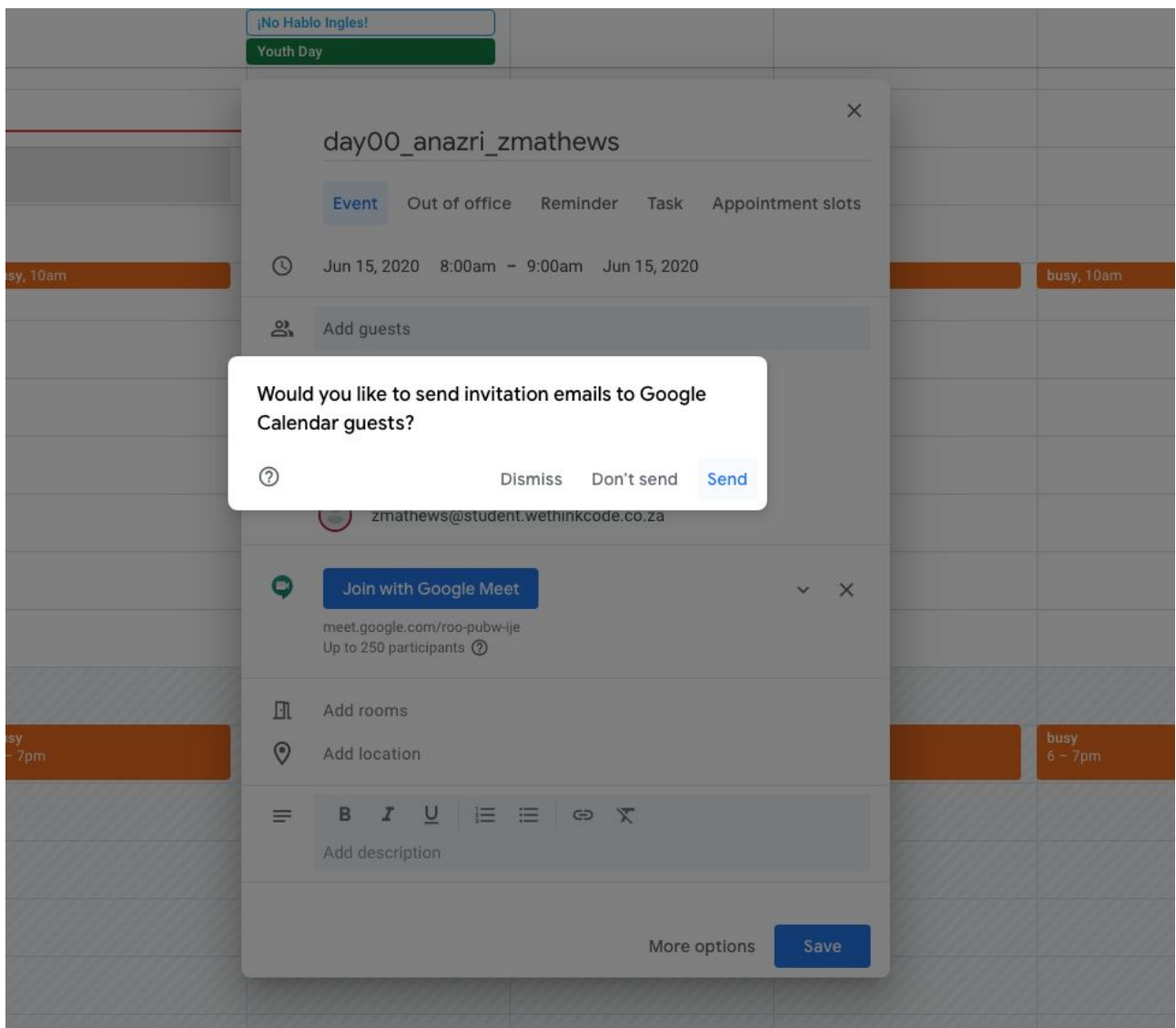
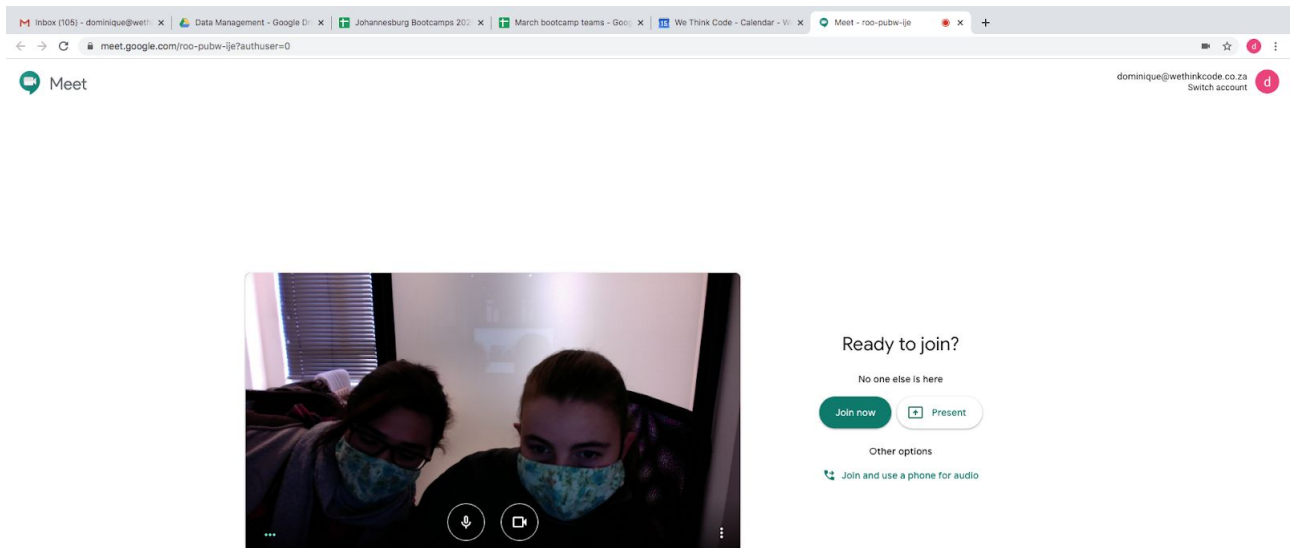
Add location



Add description

More options

Save



Missing evaluating people

1. Make sure to send the request to their student email to staff which includes evidence that the person missed and evidence that they didn't respond to attempts to reach them. Screenshots to be exact.