Tutorial: Setting up GitHub Education and Codespaces

Important: by doing this tutorial you agree that you've read and understood the document: AI Privacy, Safety, and Ethics. If you haven't read it yet, please do so before you continue.

This document was made with help from students like you. If you notice anything inaccurate, please let me know and I'll fix it.

In this course we'll use **GitHub Codespaces** as our online, cloud-based IDE. **Codespaces** allows us to create, run, and share applications within our browser. It also makes source and version control easy and provides a uniform development experience for all of us regardless of which computer and operating system we use.

As a student at Langara College, you are entitled to 90 hours per month of Codespaces use for free and 20 GB of storage. (This ought to be more than enough.)

Important: *using this service will require that you register for a* ***GitHub Education*** *account, and you'll have to submit some personal information which may be stored on servers outside the country and therefore accessible to authorities in those countries. If you have concerns about the privacy of your data, please talk to me ASAP.*

Also Important: *you are responsible for keeping your data safe and backing it up. As GitHub Codespaces is a third-party service that we don't control, there is no guarantee that it will keep your data safe, either from permanent loss or unauthorized access. By using it, you agree to accept responsibilities for all risks*

Task 1: Create a GitHub Account

GitHub is a service that provides a hub for developers to collaborate and share their code. If you don't already have an account, navigate to https://github.com/ and click **Sign Up** in top-right corner. Follow the steps to create a free account with your **Langara email address**. Note that if you use a non-Langara address you may not be approved for the education plan.

As a free GitHub member, you now can access most of their services, with some limits.

Note: *Do NOT choose a paid plan; you are entitled to free use of GitHub's Pro services as a student, which we'll arrange in the next step.*

Task 2: Set Up Account

GitHub has a strict process for verifying eligibility for their education program, and it's easy to be rejected. The requirements seem to be a bit different for each student and maybe based on the details of your documents, but I've included a list of steps here that I think are likely to work for everyone.

Even if you are rejected, don't worry; you can try again!

Task 2 A: Prepare Your Documents

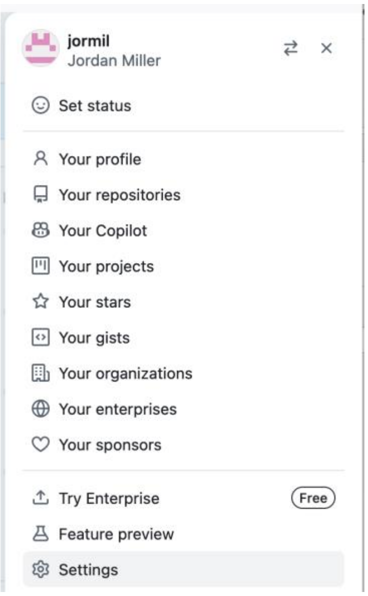
Github Education requires ONE document proving your status as a student with the college. These are recommended documents from their website:

- A picture of your school ID with current enrollment date
- Your class schedule
- Your transcript
- An affiliation or enrollment verification letter

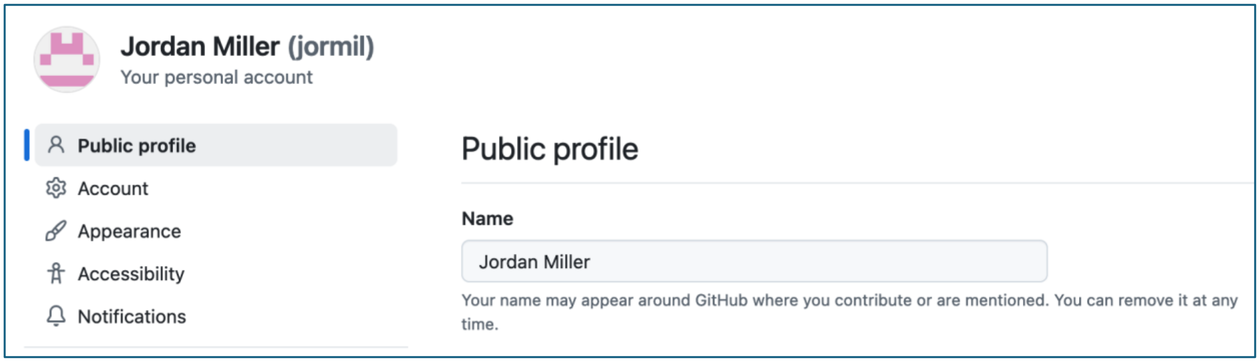
It seems that many students are rejected when they use their student ID card. I myself was rejected when trying this document; instead, try one of the others. (FYI, I was able to use a screenshot of a webpage with the relevant information.)

Task 2 B: Update Your Profile and Billing Information

When logged into GitHub, click the circular profile icon in the top-right corner, and choose **Settings:**



Make sure the name on your **Public Profile** is your **full name, exactly as it appears on your academic affiliation document.**



Jordan Miller (jormil)
Your personal account

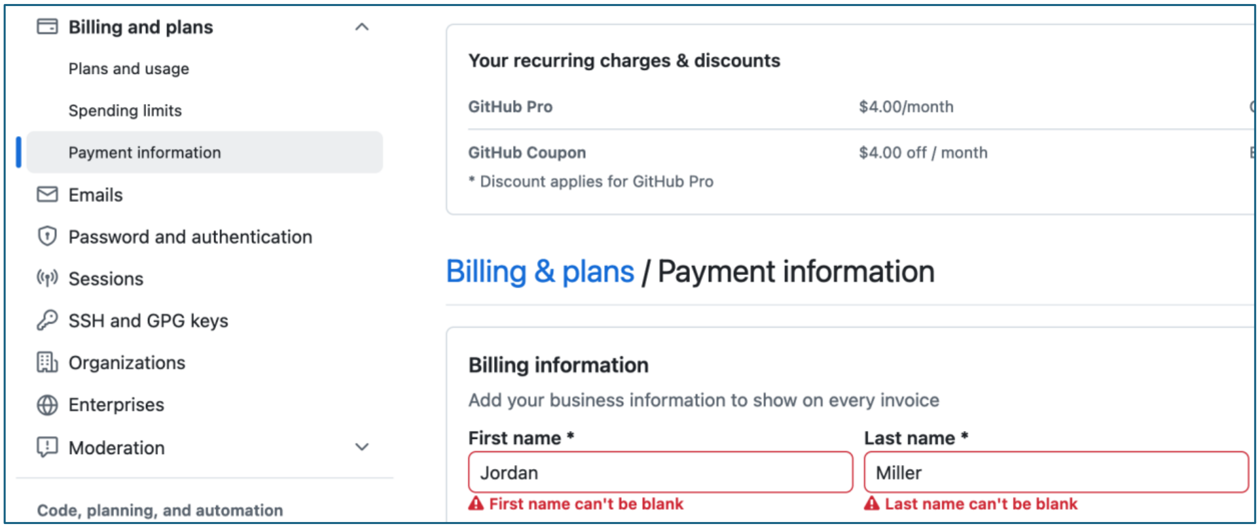
Public profile

Name

Jordan Miller

Your name may appear around GitHub where you contribute or are mentioned. You can remove it at any time.

Also, navigate to **Billing and Plans > Payment information** and fill in your **first and last name, exactly as it appears on your academic affiliation document.** If you only have a single legal name, enter it in both the first and last name fields. ***You do NOT have to enter a payment method!*** It appears that you'll also have to enter an address – if you don't feel comfortable entering your home address, you can try entering Langara College's address, ***although I'm not sure if GitHub will accept this*.**



Your recurring charges & discounts

GitHub Pro	$4.00/month
GitHub Coupon * Discount applies for GitHub Pro	$4.00 off / month

Billing & plans / Payment information

Billing information

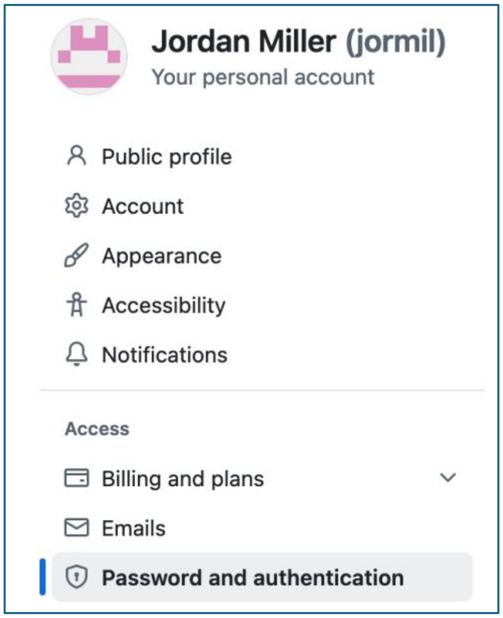
Add your business information to show on every invoice

First name *
Jordan
First name can't be blank

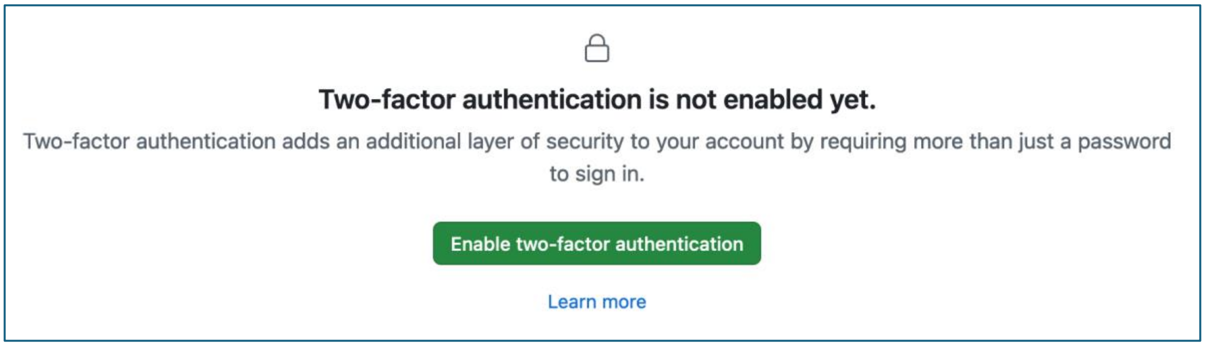
Last name *
Miller
Last name can't be blank

Task 2 C: Enable Two-Factor Authentication

From the **Settings** panel, navigate to **Password and authentication.**

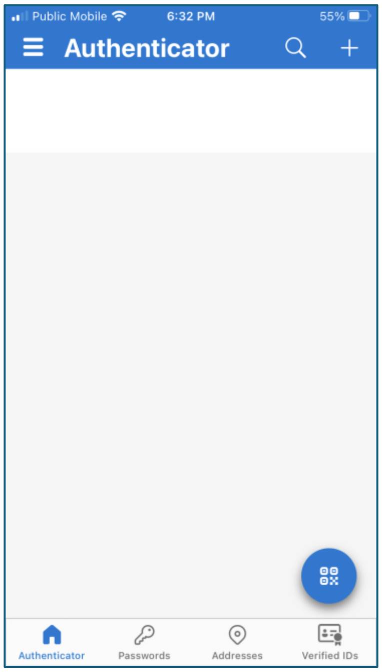


Click the green **Enable two-factor authentication** button:

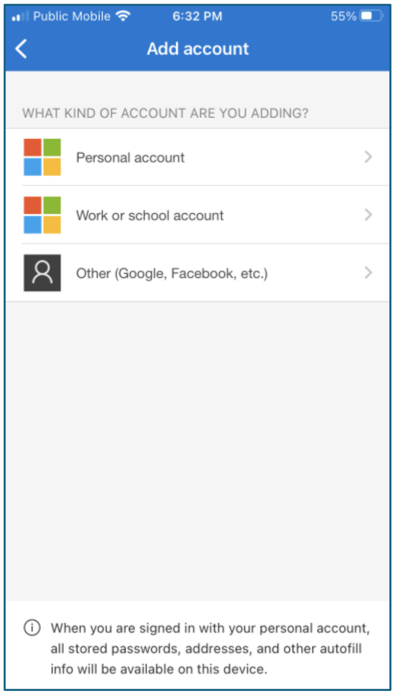

Two-factor authentication is not enabled yet.

Two-factor authentication adds an additional layer of security to your account by requiring more than just a password to sign in.

You'll have to install an application on your mobile phone for two-factor authentication. The one I use is **Microsoft Authenticator;** you can download and install this from the App store on your phone. When you open the authenticator app, you'll see a **+** sign in the top-right corner that will allow you to add another account (iOS interface is shown below):

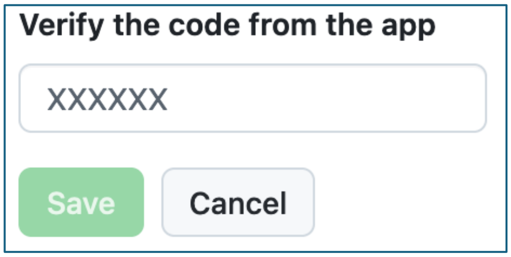


Tap the + button, and then tap **Other (Google, Facebook, etc.)**



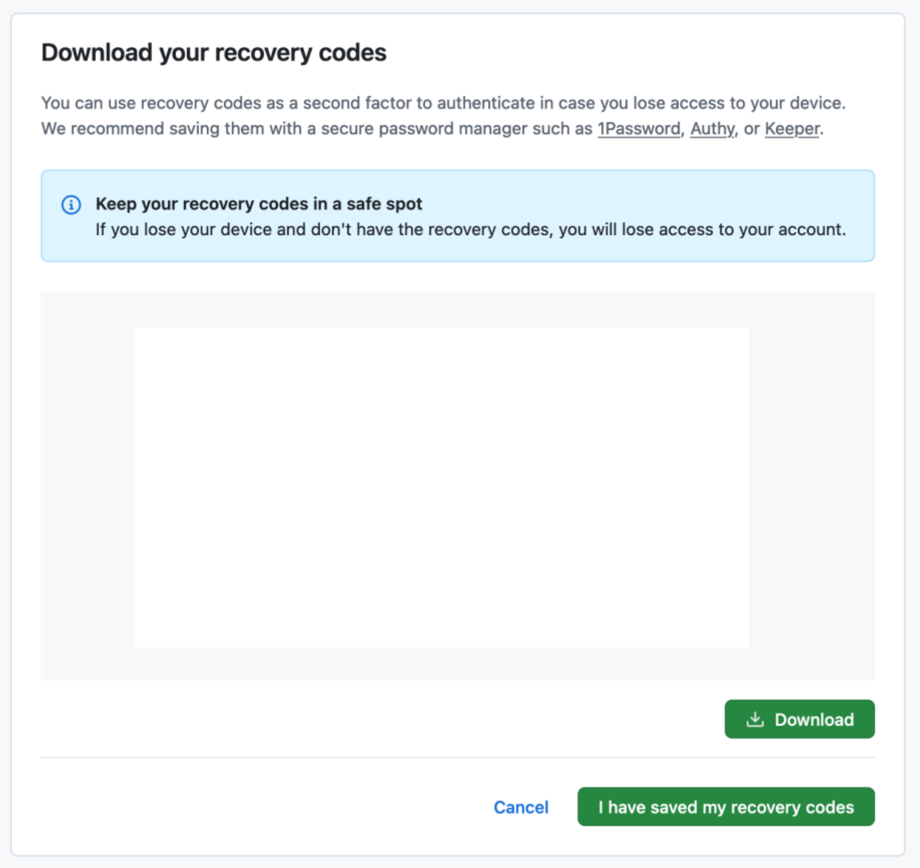
When you are signed in with your personal account, all stored passwords, addresses, and other autofill info will be available on this device.

The camera interface on your phone will be activated, and you can scan the **QR Code** displayed on the GitHub page on your computer. A 6-digit code will then appear on your phone that you can type into the interface on your computer:



Verify the code from the app

You'll be prompted to download your recovery codes: go ahead and do that, as they may be the only way to recover your account if you lose your phone or even if you lose the authenticator app (you are responsible for maintaining access to your account):



Download your recovery codes

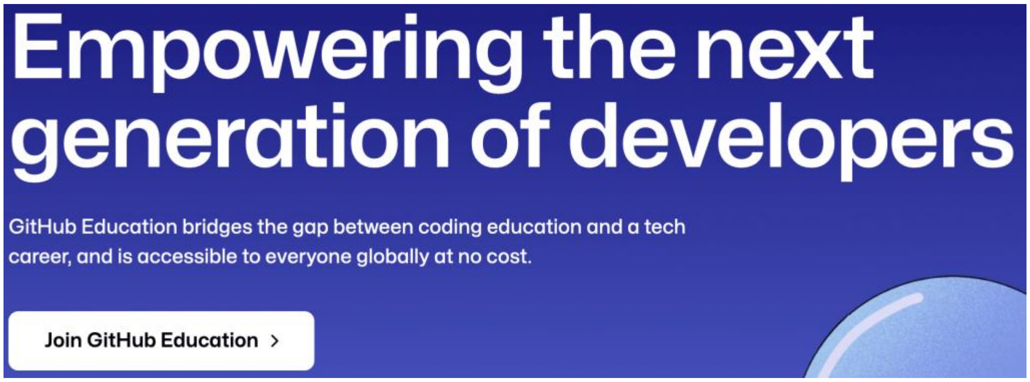
You can use recovery codes as a second factor to authenticate in case you lose access to your device. We recommend saving them with a secure password manager such as 1Password, Authy, or Keeper.

Keep your recovery codes in a safe spot
If you lose your device and don't have the recovery codes, you will lose access to your account.

At this point, you should **log out** of GitHub and login again. You'll have to enter the code from the authenticator app on your phone.

Task 3: Apply for GitHub Education

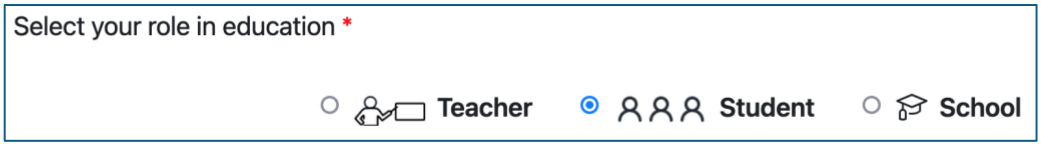
Navigate to https://github.com/education and click **Join GitHub Education**.



Empowering the next generation of developers

GitHub Education bridges the gap between coding education and a tech career, and is accessible to everyone globally at no cost.

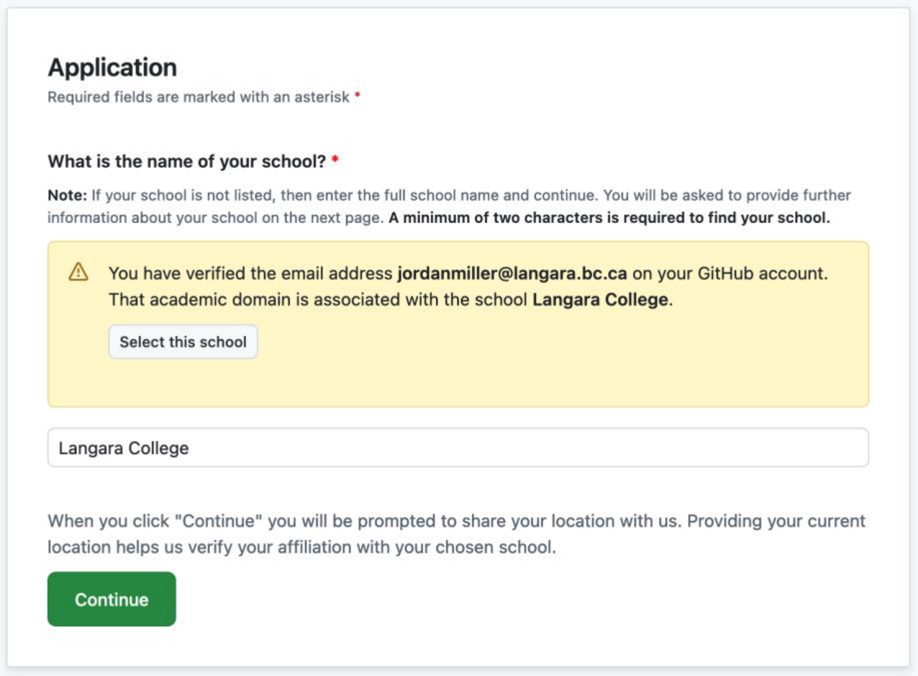
Select **Student** as your role:



Select your role in education *

○ Teacher ● Student ○ School

Make sure **Langara College** is chosen as the school – this should be automatically detected from your email address. Click **Continue.**



Application

Required fields are marked with an asterisk *

What is the name of your school? *

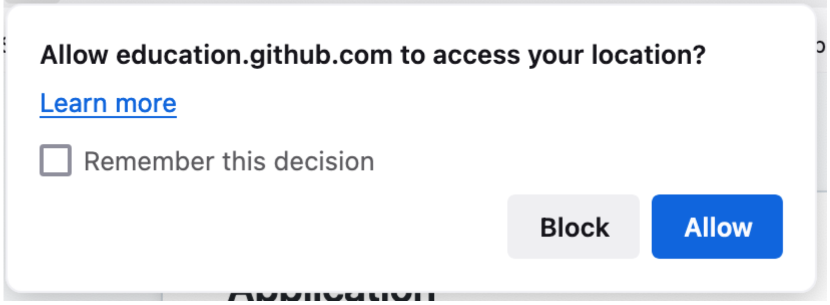
Note: If your school is not listed, then enter the full school name and continue. You will be asked to provide further information about your school on the next page. **A minimum of two characters is required to find your school.**

You have verified the email address **jordanmiller@langara.bc.ca** on your GitHub account. That academic domain is associated with the school **Langara College**.

Langara College

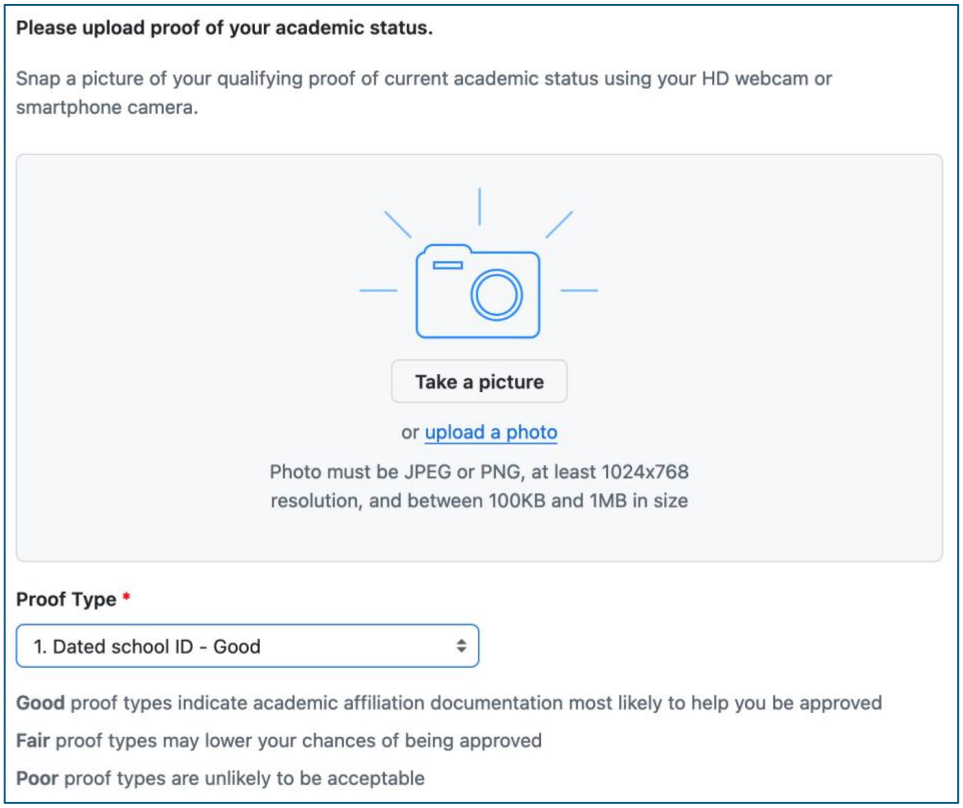
When you click "Continue" you will be prompted to share your location with us. Providing your current location helps us verify your affiliation with your chosen school.

You may be asked by your web browser to share your location; this is part of how GitHub confirms you're a real student at Langara College. Click **Allow.**



Allow education.github.com to access your location?

Upload a picture of your **academic affiliation document**.



Please upload proof of your academic status.

Snap a picture of your qualifying proof of current academic status using your HD webcam or smartphone camera.

Photo must be JPEG or PNG, at least 1024x768 resolution, and between 100KB and 1MB in size

Proof Type *

1. Dated school ID - Good

Good proof types indicate academic affiliation documentation most likely to help you be approved

Fair proof types may lower your chances of being approved

Poor proof types are unlikely to be acceptable

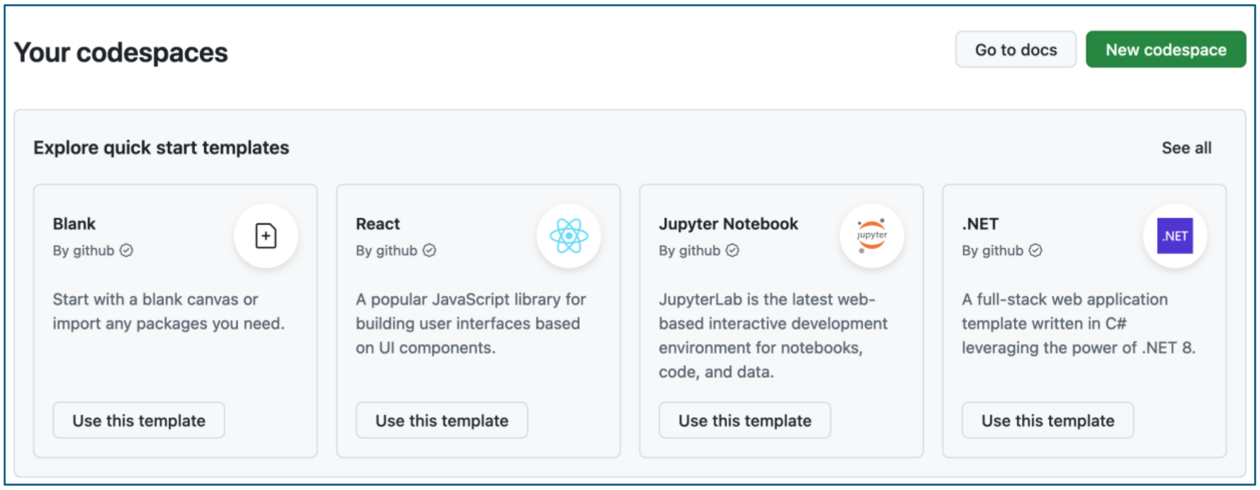
Click **Process my application.** It may take a few days for your application to be reviewed and accepted. You can check the status of your application here: https://education.github.com/discount_requests/application. If your application is rejected, please consult this document: https://docs.github.com/en/education/about-github-education/github-education-for-students/solving-problems-with-your-github-education-access. The rejection email itself may also have some other suggestions to try.

I you've tried everything and still haven't been approved, please talk to me.

Task 4: Using Codespaces

Even before your Education application has been accepted, you can get started using Codespaces with a temporarily reduced limit. Open the menu in the top-left corner of the screen and click **Codespaces,** or access it here: https://github.com/codespaces

To create a new project, you'll most often use the **Blank** template. Click **Use this template:**



Start with a blank canvas or import any packages you need.

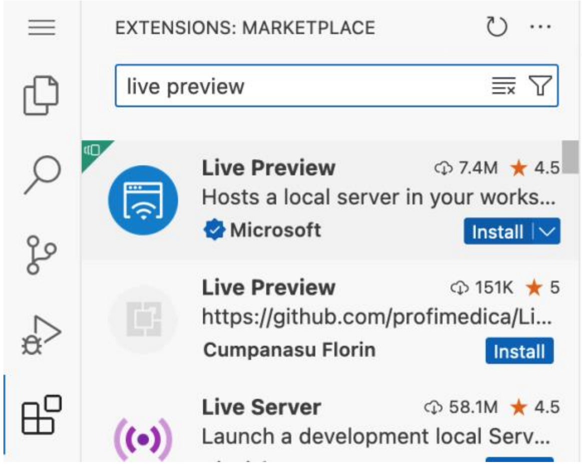
A popular JavaScript library for building user interfaces based on UI components.

JupyterLab is the latest web-based interactive development environment for notebooks, code, and data.

A full-stack web application template written in C# leveraging the power of .NET 8.

As you can see, Codespaces uses the familiar Visual Studio Code editor interface; very convenient! Not only can we write programs here, but our project has access to server infrastructure to run and serve web applications.

Go ahead and make a simple static HTML page to test the editor. Experiment with creating and saving files and folders. You can preview your new webpage in the browser by clicking the **Extensions** icon on the left, searching for "live preview", and installing Microsoft **Live Preview** extension:



Hosts a local server in your works...

https://github.com/profimedica/Li...

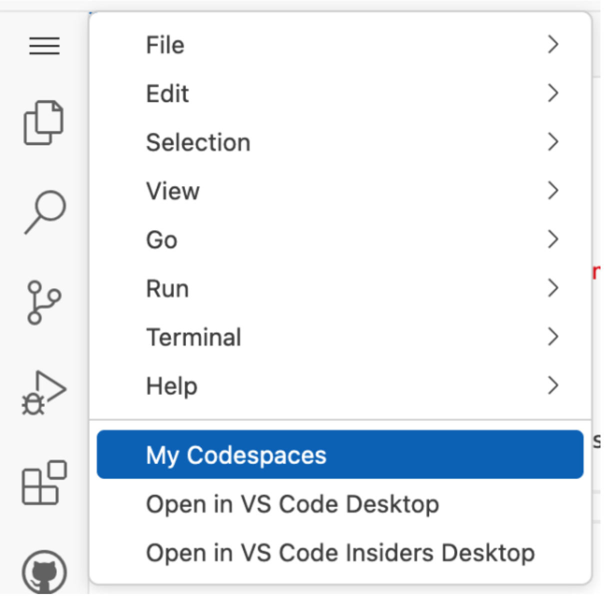
Launch a development local Serv...

Then simply click the **Show Preview** button in the top-right corner of the editor pane for your HTML page file:

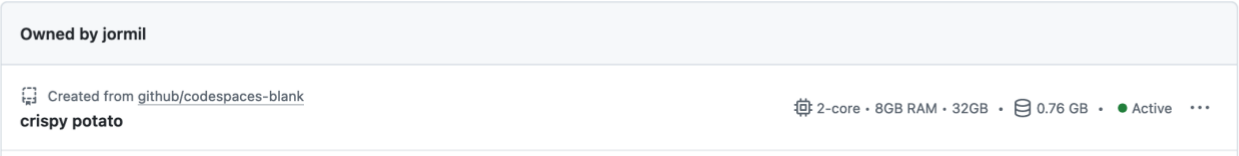
Important: *while your project is private by default, it is possible to configure it to be public. If you project becomes public, even accidentally, and maybe due to errors on GitHub's part without any action or knowledge on your part, the data stored in your project could be leaked, and I recommend NOT putting anything personal or sensitive here.*

Task 5: Management and Organization

To get back to the Codespaces dashboard, you can click the **hamburger menu** button in the top-left corner and choose **My Codespaces.**



Here you can see a list of all your projects, although we've only created one so far:

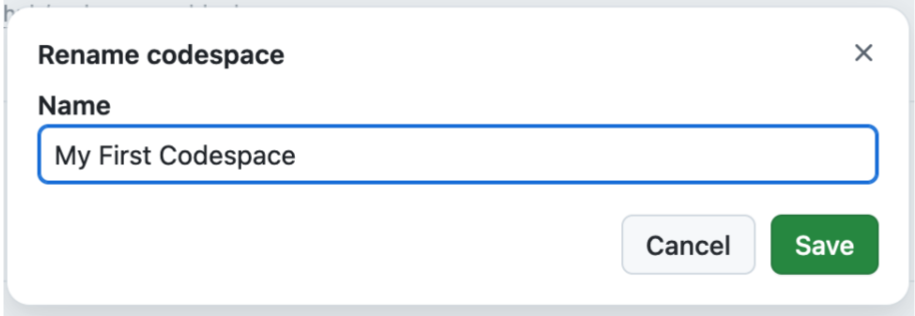


Owned by jormil

Created from github/codespaces-blank
crispy potato

2-core • 8GB RAM • 32GB • 0.76 GB • Active ···

Notice that your project is automatically given a random name; mine is "crispy potato", which sounds delicious, but isn't very descriptive. You can rename your project by clicking the **...** button, choosing **rename,** and typing a new name:

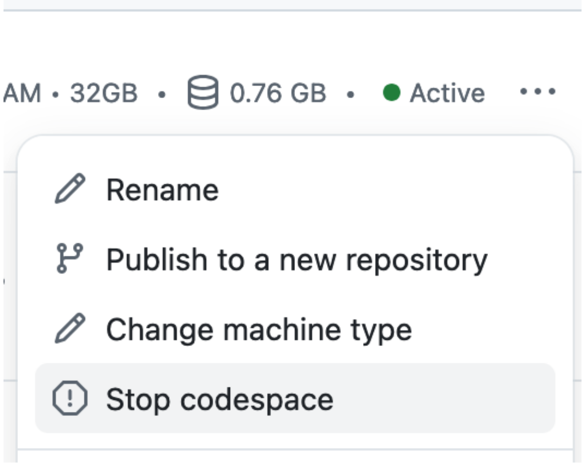


Rename codespace

Name

My First Codespace

Also notice that your project has a green **Active** indicator, which means it's currently running and using your limited allotted monthly time. When not working on a project, you should **stop** it to prevent it from using resources unnecessarily. You can do this by clicking the **...** button and choosing **Stop Codespace.** Don't worry, you can start it again when you're ready.

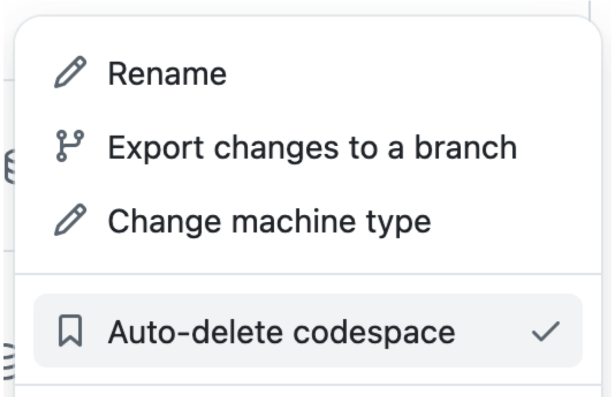


Task 6: Backup, Data Retention, and Source Control

Once again, you are responsible for backing up your data regularly. If a Codespace project is the only copy of your work, you are taking a huge risk. *In this section I'll give you some tips for keeping your work safe, although they're not guaranteed to be 100% foolproof, as error's in GitHub's system may lead to unexpected data loss. Use the service at your own risk.*

By default, GitHub will **delete projects that have been stopped for 30 days.** To prevent this from happening, you can do one of the following:

- Start and access your project before 30 days have passed to reset the clock, OR
- Disable automatic deletion. You can do this from the dashboard by clicking the **...** button beside the codespace and clicking **Auto-delete codespace** to **uncheck** that option. **Warning:** the official documentation for this feature differs from the current interface, so I wouldn't trust this method for now.



I strongly recommend backing up the data to your local computer (or at least syncing it with a repository) at the end of every coding session. There are a few ways to do this which I will describe now.

You can download individual files from the VS Code interface by simply right-clicking the file and choosing **Download...**

Weirdly, this doesn't work with folders. If you want to download a folder, you have to zip the folder first and download the zip archive. This will work if you want to download all the files and folders in the project, all at once; simply create a new folder, move all the files into that folder, and then zip the folder using this command in the terminal:

```
zip -r name-of-archive.zip name-of-folder
```

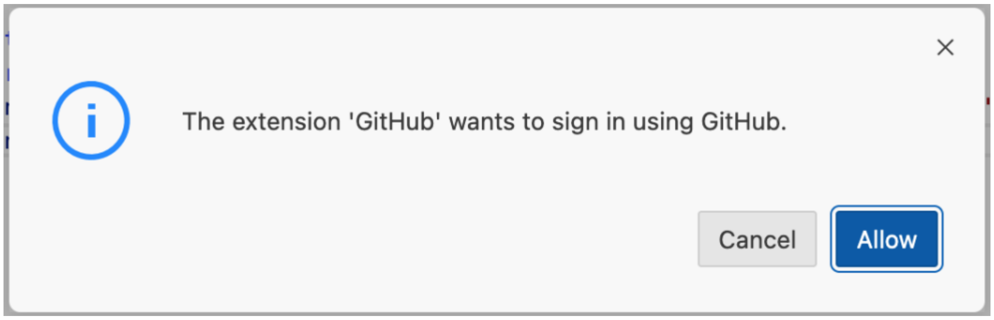
You can then right-click and download as usual.

Another thing you can do is use Visual Studio's source control and GitHub integration to create a new GitHub repository for your project. Start by creating a file in the root directory called **.gitignore** (don't forget the dot.) This file keeps track of all the files and folders we wish to be excluded from our repository. Inside this file, put the following:

node_modules
.env

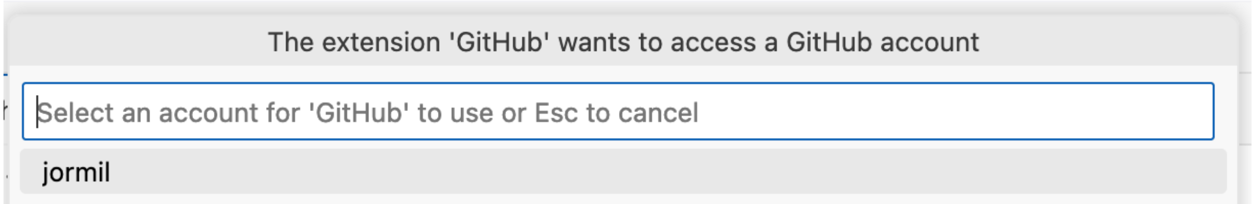
I won't explain this now, but you'll thank me later. Include a .gitignore file like this in every project; you may want to add other files and folders to it that should be ignored. Now do the following:

- Click the **source control** button on the left:
- Click **Publish to GitHub.**
- You'll be prompted to provide access to your GitHub account; click **Allow**



The extension 'GitHub' wants to sign in using GitHub.

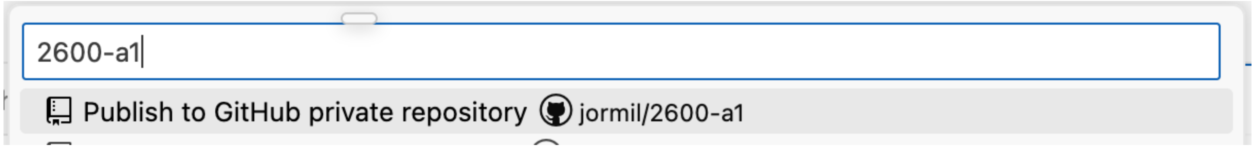
You'll be prompted to choose which GitHub account you want to publish to. Select your account name below. (Mine is "jormil" but you'll see your own username.)



The extension 'GitHub' wants to access a GitHub account

Select an account for 'GitHub' to use or Esc to cancel

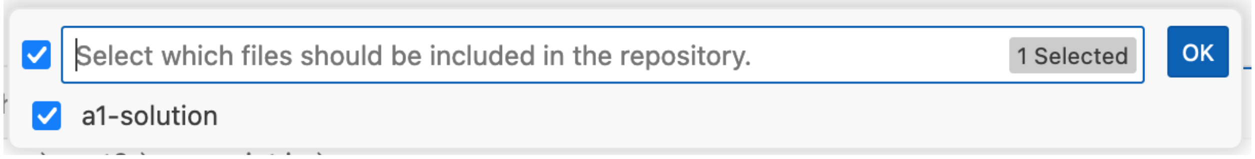
Click **Publish to GitHub** again. In top-middle part of the interface, a text field will appear; Type a name for your new code repository and click **"Publish to GitHub private repository".**



2600-a1

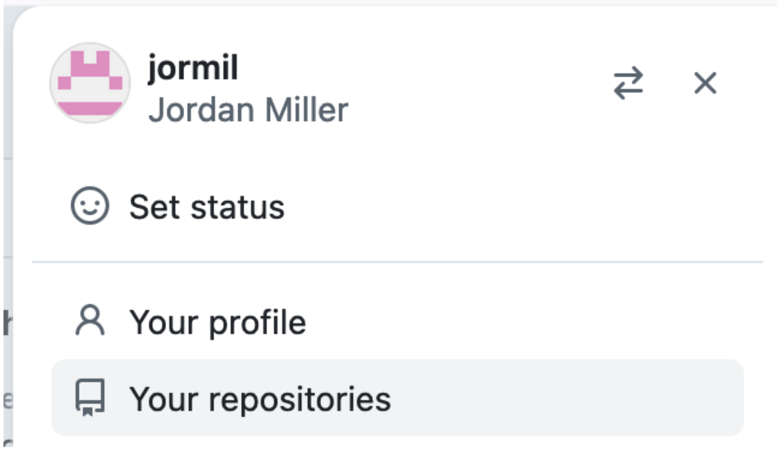
***(Warning:** if you choose the public option instead, your code will be accessible to literally everyone on the internet. Don't do this unless you're absolutely sure you want to.)*

Check the files and folders you want to include and click **OK**. (in my example, my project has a single folder called a1-solution that contains all the files, but you'll see a different set of files and folders depending on your project.)

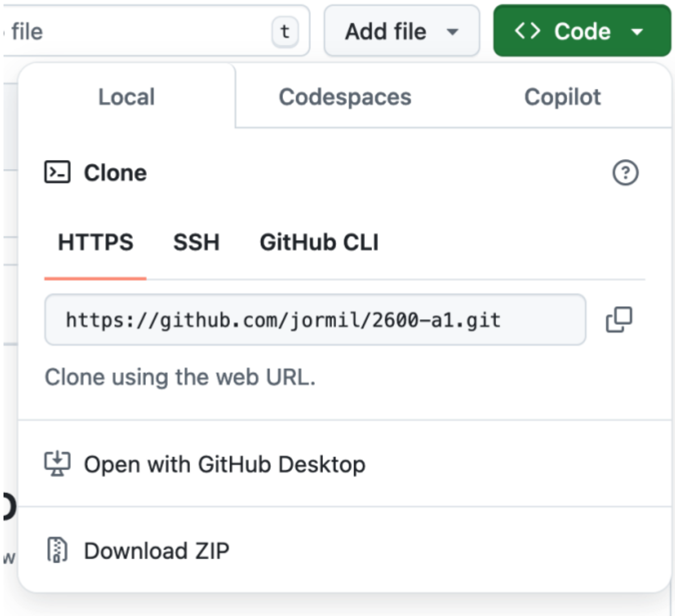


This will create a **repository** on your GitHub account with a **copy** of your Codespaces project. A repository is a copy of your project that can be shared with other developers or just kept to yourself for private use.

After creating the repository, a popup will appear in the bottom right corner of the window allowing you to **Open in GitHub**. If you miss the popup (it disappears quickly) you can navigate to https://github.com/ , click the **circular menu button** in the top right corner, and click **Your repositories:**



You should see your recently-created repo show up in a list. Click the link to visit the repository. To download the project, click the green **Code** button and choose **Download ZIP:**

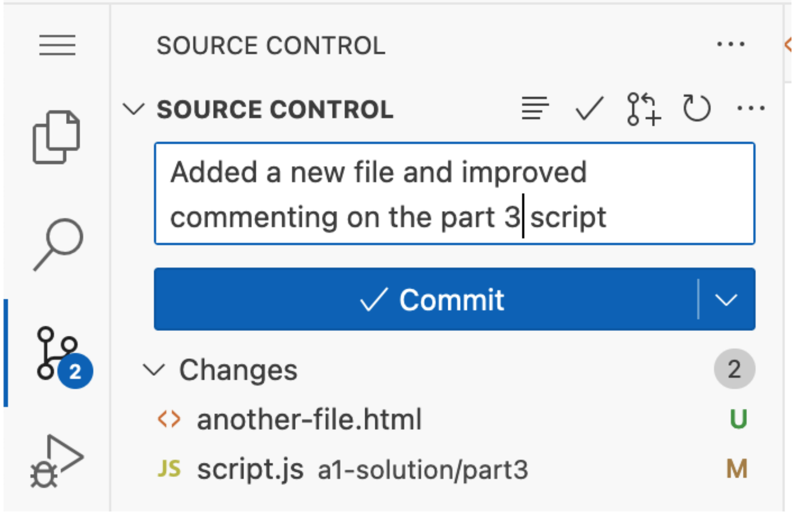


```
https://github.com/jormil/2600-a1.git
```

Clone using the web URL.

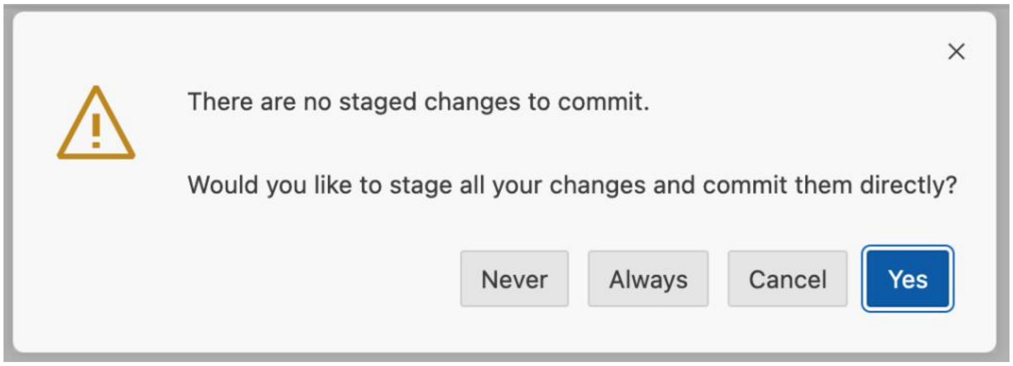
You're going to want to frequently record your progress in this repository every time you make significant changes. To test this, make some changes to your project right now (maybe just add some comments or a new file).

To record your recent changes, click the **source control** button again. You'll see a list of files that have changed since you created the repository, and a space to type a short message describing your changes. Type a message and click **Commit.**



Added a new file and improved commenting on the part 3 script

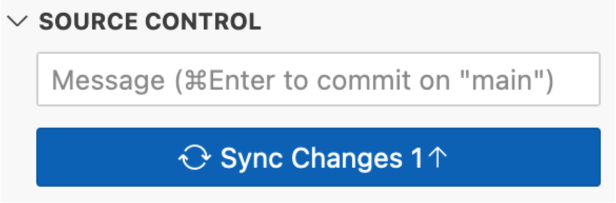
You'll see an alert indicating that you haven't "staged" any changes, and asking you if you'd like to stage all changes. Click **Yes.**



There are no staged changes to commit.

Would you like to stage all your changes and commit them directly?

Then click **Sync Changes,** and **OK:**



When you navigate back to your repository, you'll see that the changes have been updated there. Your fully up-to-date project can once again be downloaded.