

Work experience Nov 2022 – Build a website activity

Schedule:

Day 1 morning:

- Introduction by managers
- Talk to 2 –3 people in different roles – Mamata and Prashant

Day 1 afternoon:

- Start working on the web development tutorial

Day 2 morning:

- Interview with Dalton and Blake and Renzil

Day 2 afternoon:

- Start working on the website

Day 3 morning:

- Talk with mark

Part 1 - Tutorials

Get familiar with HTML, CSS and JavaScript programming languages for front-end web development.

<https://www.w3schools.com/whatis/default.asp>

The W3C Schools site has some tutorials and plenty of “Try it yourself” parts to familiarize yourself with the 3 languages that make up the foundations for most websites. The key sections are highlighted in red but feel free to do as much or as little as you like.

HTML Tutorial	CSS Tutorial	JS Tutorial	JS Arrays
HTML HOME	CSS HOME	JS HOME	JS Array Methods
HTML Introduction	CSS Introduction	JS Introduction	JS Array Sort
HTML Editors	CSS Syntax	JS Where To	JS Array Iteration
HTML Basic	CSS Selectors	JS Output	JS Array Const
HTML Elements	CSS How To	JS Statements	JS Dates
HTML Attributes	CSS Comments	JS Syntax	JS Date Formats
HTML Headings	CSS Colors	JS Comments	JS Date Get Methods
HTML Paragraphs	CSS Backgrounds	JS Variables	JS Date Set Methods
HTML Styles	CSS Borders	JS Let	JS Math
HTML Formatting	CSS Margins	JS Const	JS Random
HTML Quotations	CSS Padding	JS Operators	JS Booleans
HTML Comments	CSS Height/Width	JS Arithmetic	JS Comparisons
HTML Colors	CSS Box Model	JS Assignment	JS If Else
HTML CSS	CSS Outline	JS Data Types	JS Loop For
HTML Links	CSS Text	JS Functions	JS Loop For In
HTML Images	CSS Fonts	JS Objects	JS Loop For Of
		JS Events	JS Loop While
		JS Strings	JS Break
			JS Precedence
			JS Errors
			JS Scope
			JS Hoisting
			JS Strict Mode
			JS this Keyword
			JS Arrow Function
			JS Classes
			JS Modules
			JS JSON
			JS Debugging
			JS Style Guide
			JS Best Practices
			JS Mistakes

Other resources:

- Tips for debugging HTML CSS JS with Chrome dev tools <https://react.school/debugging/chrome-devtools>

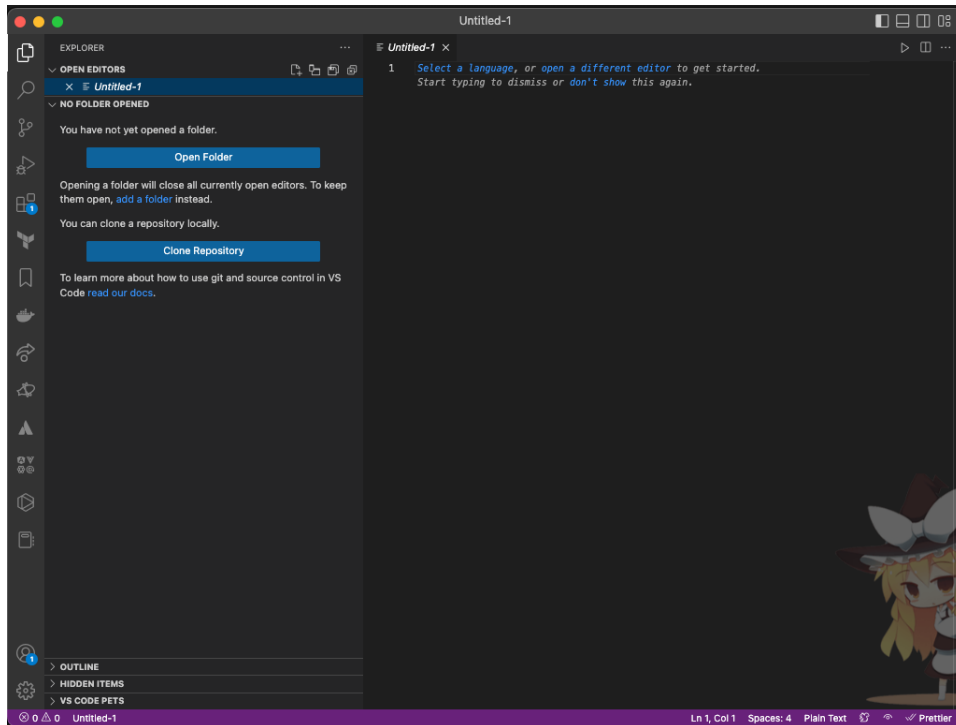
Part 2 - Setting up VSCode and Github

A few setup steps are needed before you start.

Visual Studio Code

Download and install the Visual Studio Code IDE <https://code.visualstudio.com/download>

Generally, you can use any text editor such as Notepad (Windows) or Notes (MacOS) to write code, but an IDE such as VSCode provides powerful tools to help you code.



Setup a Github account and fork the repository

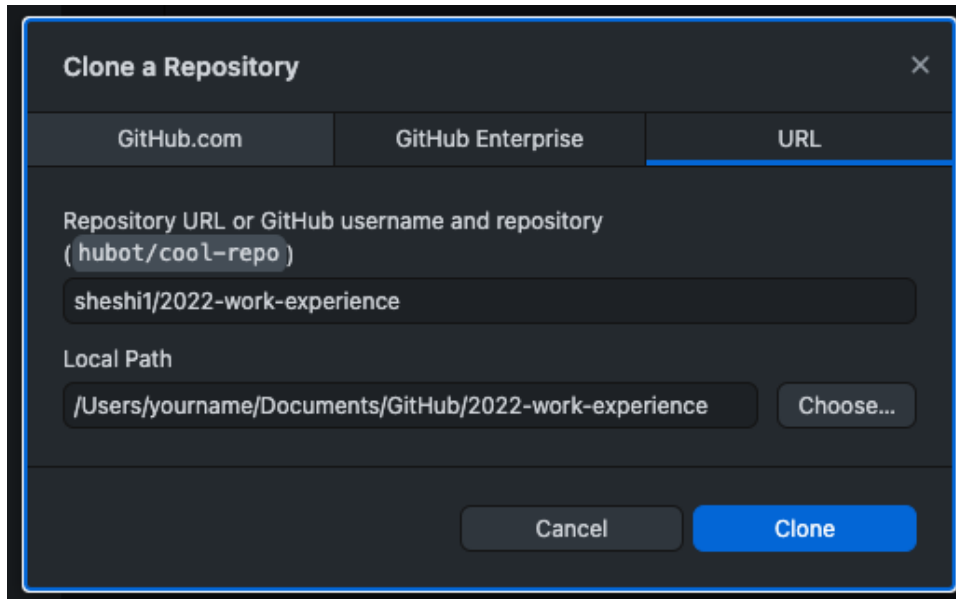
This repo contains some skeleton code we've written which you can copy into your Github account. This "forked" repo is now yours to own and do whatever you like without affecting the original repo.

1. Sign up for an account on <https://github.com>
2. Visit <https://github.com/sheshi1/2022-work-experience/>
3. Fork the repository via the "Fork" button on the top right. All the default values should be ok. <https://docs.github.com/en/get-started/quickstart/fork-a-repo#forking-a-repository>
4. You should end up with a copy of the repository under your account, eg. https://github.com/<YOUR_ACCOUNT>/2022-work-experience

Using Github for Desktop to download the forked repo to your laptop

1. Download and install Github for Desktop <https://desktop.github.com/>
2. Open Github for Desktop and sign in
3. Click the button "Clone a repository from the internet"

4. In the “Clone a Repository” modal that appears, switch to the URL tab and in Repository URL field, type in <YOUR ACCOUNT>/2022-work-experience or whatever you called it in the previous section.
5. Keep the “Local Path” location in mind for the next section when you open the project in VSCode.
6. Click “Clone”. This will download a copy of the repo’s files onto your laptop.

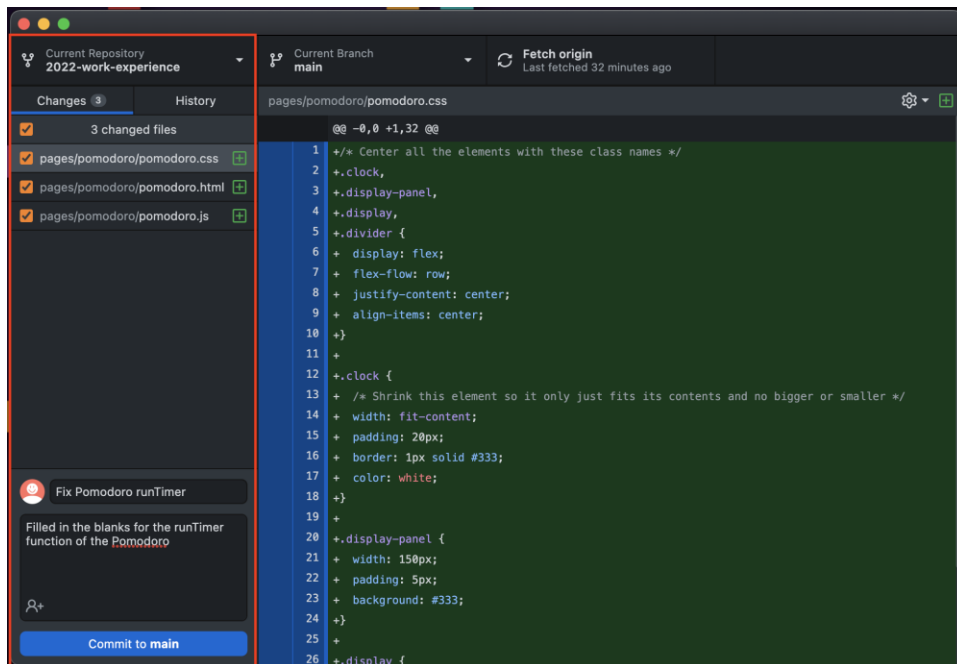


Using Github for Desktop to back up your code online (optional)

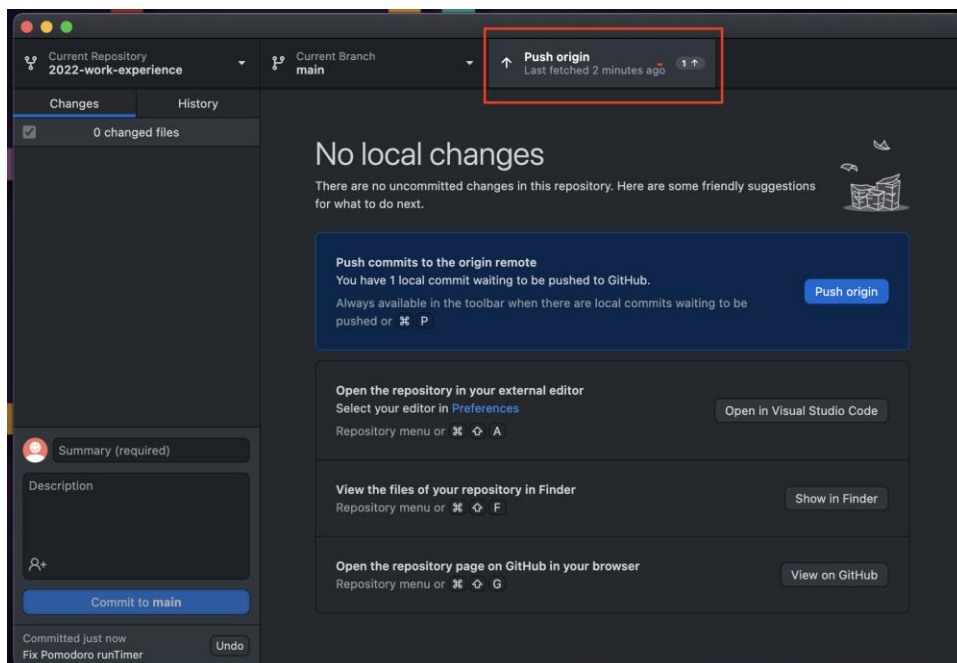
Now that you will be working on a project for the next few days, it might be wise to back up your files from time to time incase it gets lost or want to work on the same code base with other people.

BEFORE YOU PUSH ANY CODE TO GITHUB PLEASE DOUBLE CHECK THAT YOU DO NOT HAVE ANY PERSONAL INFORMATION ANYWHERE IN YOUR CODE SUCH AS YOUR FULL NAME, PHONE NUMBER, ADDRESS.

In this screenshot, I’ve been editing the code of 3 files and am about to create a commit. I gave my commit a title (and optional description) and clicked “Commit to main”.



Once the commit is created, there will be a number and up arrow indicating there are X numbers of commits sitting on the laptop that's yet to be "pushed" to Github. Click the "Push origin" button and that's it.



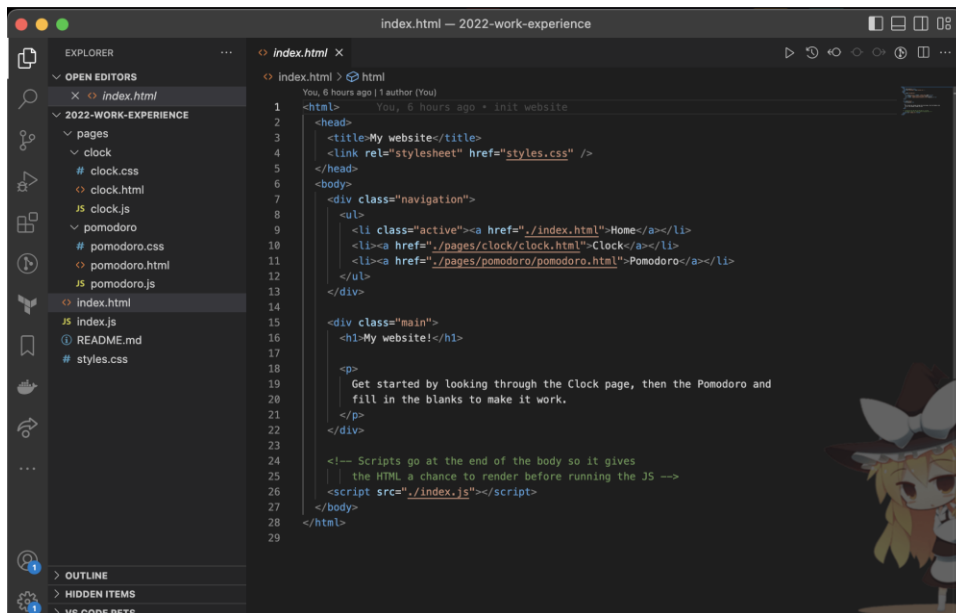
You should now be able to see that the latest revision of code is synced to Github
https://github.com/<YOUR_ACCOUNT>/2022-work-experience/commits/main

Part 3 – Coding your first website

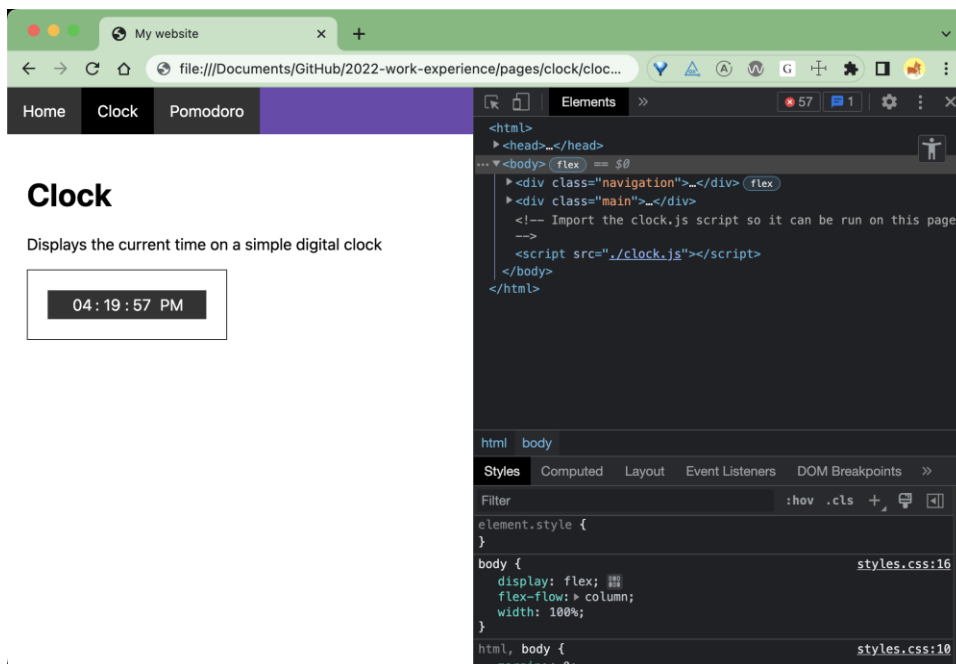
Explore the skeleton code in VSCode

1. In VSCode and in the Explorer panel (if not visible press the first icon on the top left bar) and click “Open folder”.
2. Find the folder of the skeleton code (in the previous step it would be saved in wherever you entered for the “Local Path” field)

With the Explorer view now open with the project, take a moment to look around the files provided and play with the website. You can run it by opening index.html in your browser.



```
1 <html>
2   <head>
3     <title>My website</title>
4     <link rel="stylesheet" href="styles.css" />
5   </head>
6   <body>
7     <div class="navigation">
8       <ul>
9         <li class="active"><a href="/index.html">Home</a></li>
10        <li><a href="/pages/clock/clock.html">Clock</a></li>
11        <li><a href="/pages/pomodoro/pomodoro.html">Pomodoro</a></li>
12      </ul>
13    </div>
14
15    <div class="main">
16      <h1>My website!</h1>
17
18      <p>
19        Get started by looking through the Clock page, then the Pomodoro and
20        fill in the blanks to make it work.
21      </p>
22    </div>
23
24    <!-- Scripts go at the end of the body so it gives
25         the HTML a chance to render before running the JS -->
26    <script src="/index.js"></script>
27  </body>
28 </html>
29
```

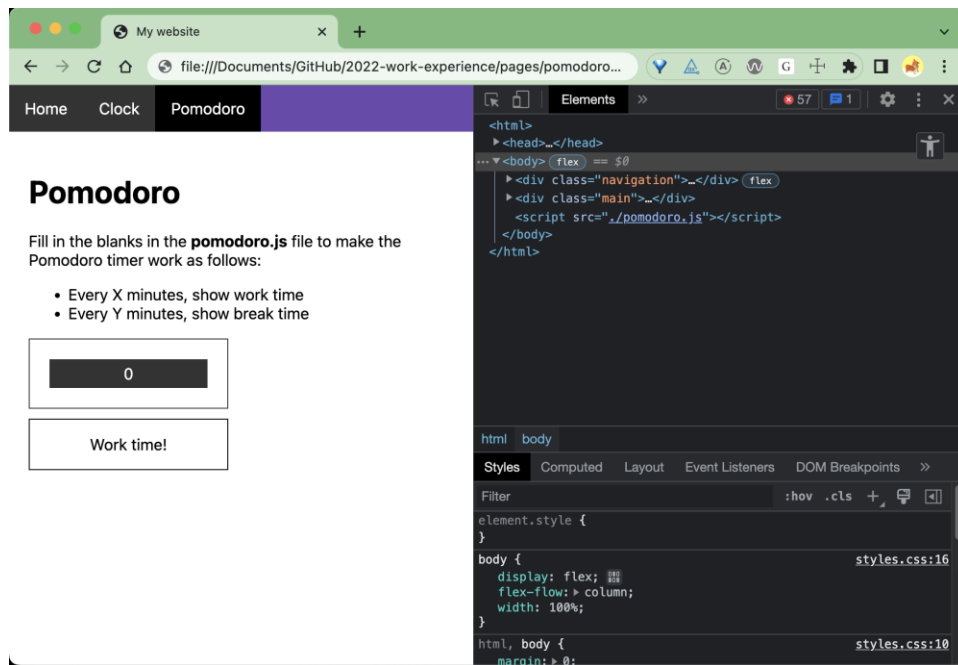


Completing the Pomodoro app

You may notice that there are some files prefixed with “Pomodoro” within the project, and the Pomodoro isn’t working on the website. Your first task is to fix the `pomodoro.js` file so that it works. Read the comments and use the pseudo-code to help you write some Javascript in each function.

Optionally, have a look at the `pomodoro.css` file and make CSS changes so the Pomodoro is more visually interesting.

With any code change, remember to save the file and refresh the page on your browser so it updates.



Part 4 – Extending your website

After you’ve gotten a hang of the tutorials and gotten the Pomodoro working, feel free to continue adding to your website and make it yours.

If you’re looking for some inspiration, below are some ideas on the theme of “productivity tools”:

- Website improvements
 - o Change the appearance of your website with HTML/CSS
 - o Designing a logo with CSS
 - o Add a link to your Github repo (once you’ve published your website in the next section)
- Polish up your Pomodoro. Here are some fully featured Pomodoros that have top search ratings on Google:
 - o <https://pomodorotimer.online>
 - o <https://pomofocus.io>
- Create more tools for your website. You should now have at least the Home, Clock and Pomodoro pages in your website. Create another page for the next tool you’re adding:
 - o Calculator <https://www.freecodecamp.org/news/how-to-build-an-html-calculator-app-from-scratch-using-javascript-4454b8714b98/>
 - o To-do list https://www.w3schools.com/howto/howto_js_todolist.asp
 - o Inspirational quotes generator
 - What is frontend VS backend? <https://www.udacity.com/blog/2020/12/frontend-vs-back-end-vs-full-stack-web-developers.html>
 - Use the /quotes REST endpoint of this fake server <https://dummyjson.com>
 - o More ideas <https://freshman.tech/random-quote-machine/>
- Turn your website into a portfolio showcasing your coding journey
 - o Web developers sometimes create a portfolio to show off their personal projects and add a bit of flair to their professional resumes
 - o See what different people have made here <https://www.hostinger.com/tutorials/web-developer-portfolio>

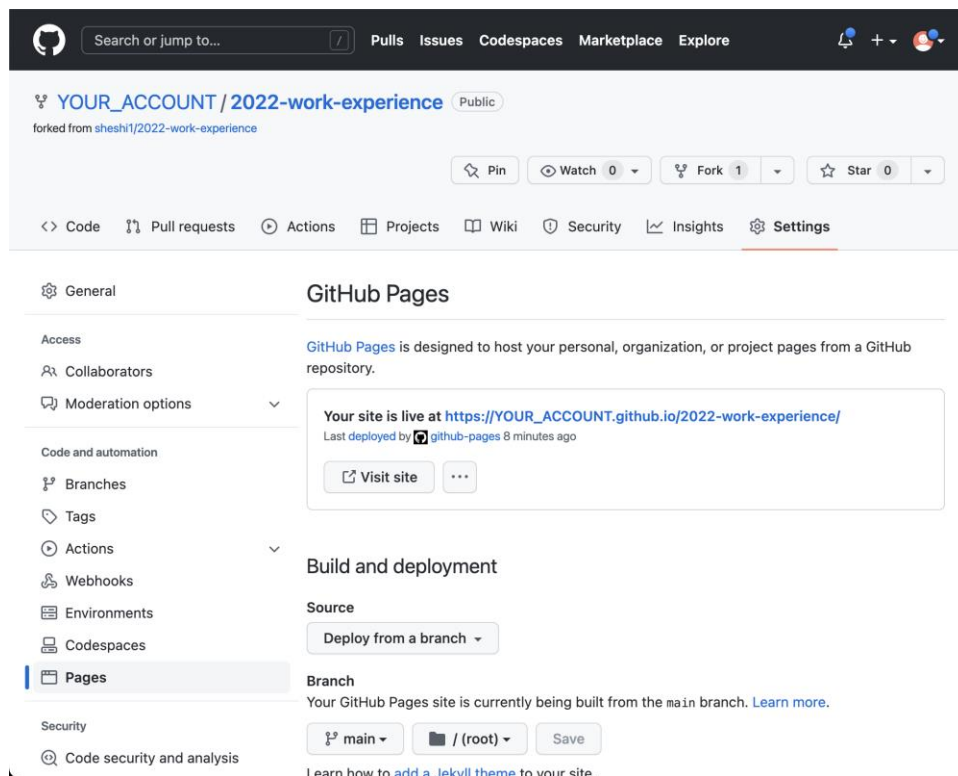
Part 5 - Publishing to Github Pages (optional)

Your website is technically not a real website unless it's published online 😊

BEFORE YOU PUSH ANY CODE TO GITHUB PLEASE DOUBLE CHECK THAT YOU DO NOT HAVE ANY PERSONAL INFORMATION ANYWHERE IN YOUR CODE SUCH AS YOUR FULL NAME, PHONE NUMBER, ADDRESS.

Assuming you've completed the previous section on how to back up your code to Github, you can also set it up so that your code gets published to Github Pages every time a push is made to your repo:

1. In your Github repo's settings, go to the Pages section
https://github.com/<YOUR_ACCOUNT>/2022-work-experience/settings/pages
2. Set the source to "Deploy from a branch"
3. Set the branch to "main" and "/" (root)"
4. You'll eventually get a notification saying your site is live on
https://<YOUR_ACCOUNT>.github.io/2022-work-experience



For any subsequent changes that you want to make to your live website, push your changes to your repo and wait a minute or so for it to update.

FYI Github Pages sometimes runs into an issue where it doesn't update – try again by pushing another code change to get it to redeploy.

