

README file

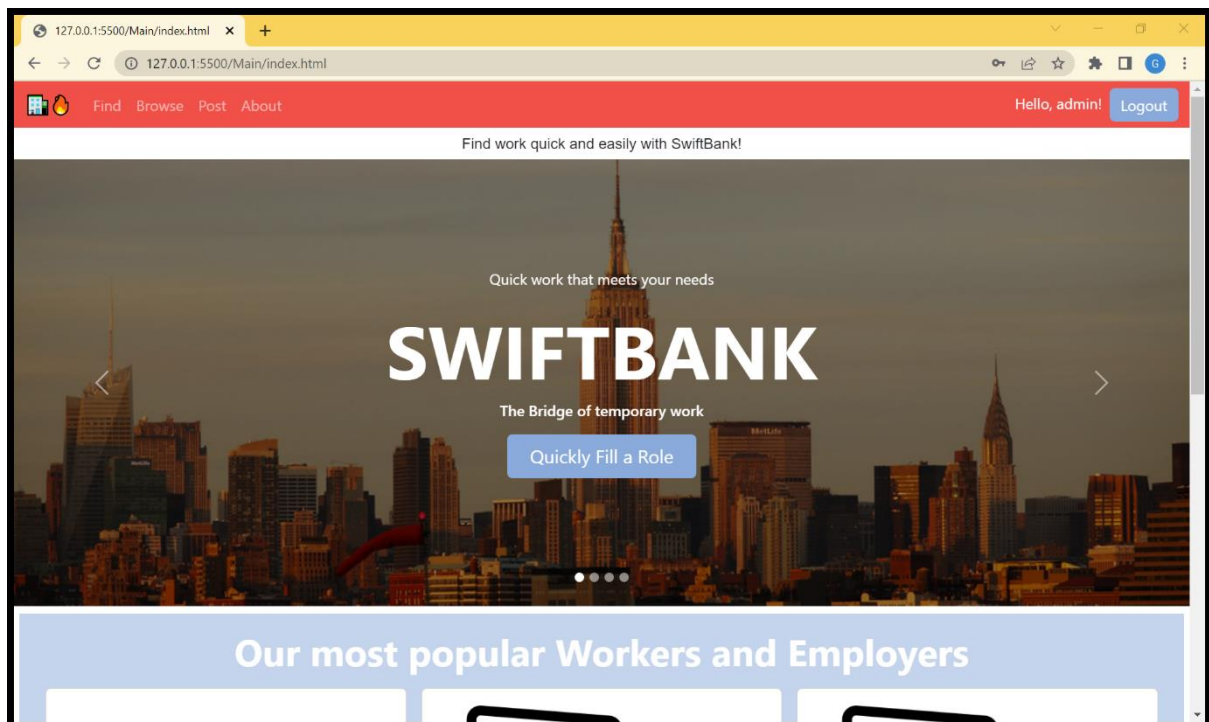
!!Just a disclaimer that most of the information on this website is spoofed. Please do not contact any of the contacts linked to this website as they are not associated. All the images were also found from outside sources. The links to these can be found from the web links in the source code.!!

I used Github copilot to help me while coding at points so some of the variables or procedures used may be slightly generic. This is all my own work.

To view the Webpage, you can run any of the pages. I would recommend running the index.html page. When a new session is started you will be taken to the login page. You can either use the navigation bar to traverse the website as a guest or you can use the username: "admin" and the password: "admin" to login.

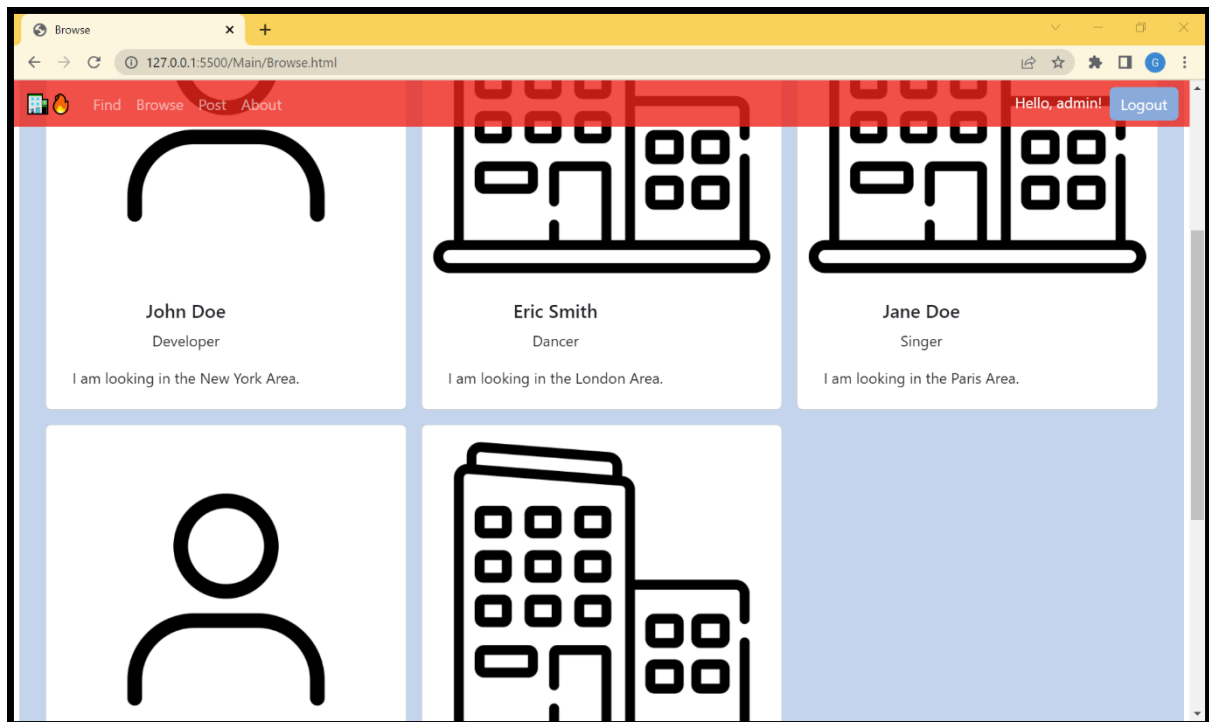
I have two different layout schemes for the website. Firstly, there is the display layout which is pages that are mainly used to display data to the user. In this case there is the index.html, Browse.html, About.html and Item.html pages. These can be seen below:

The index page:



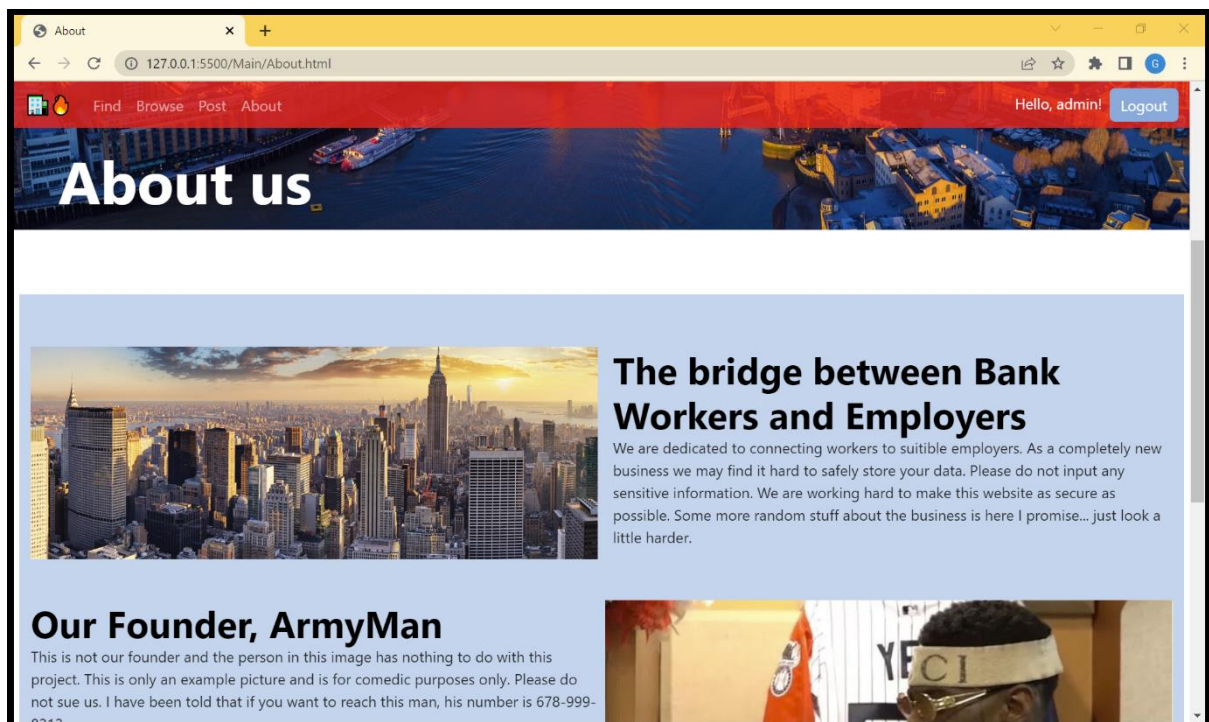
This includes a carousel, a table of 3 cards which are used as a preview, and a footer.

The Browse page:



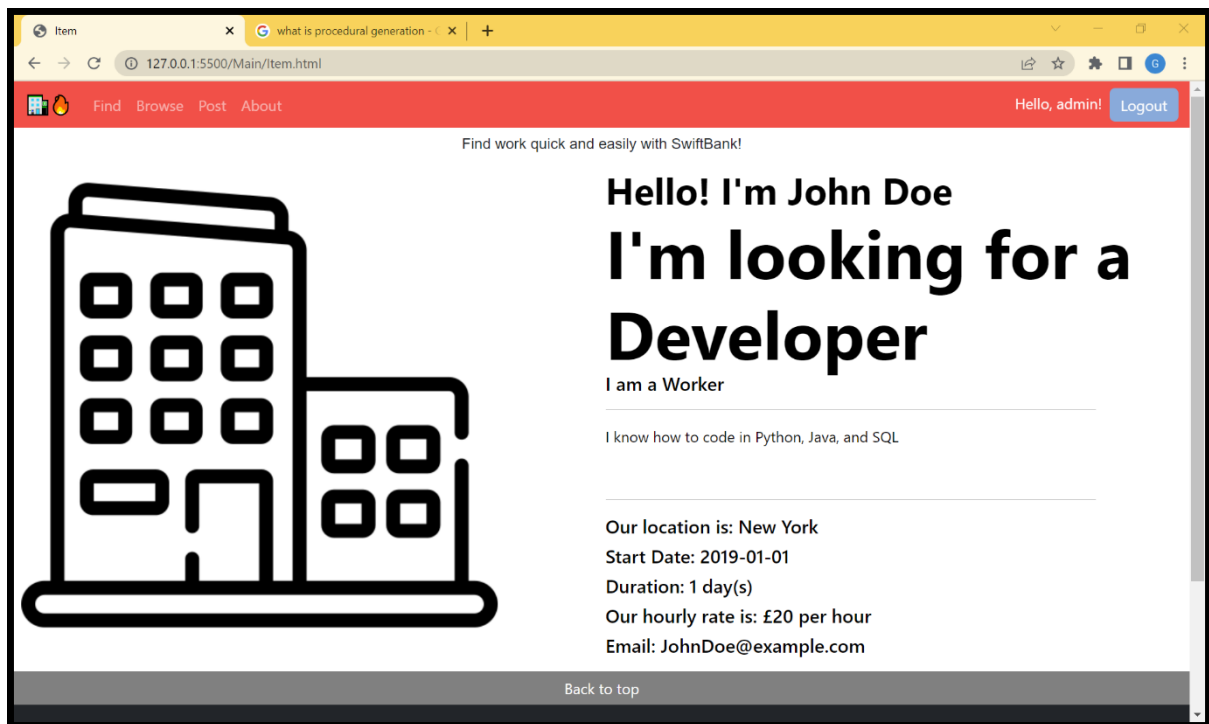
This page consists of procedurally generated table of cards and a footer. Initially, this page displays everything that is stored in the items.json file. This can be changed and filtered with the Find.html page so only certain content is produced. Each card links to a Item.html page.

The About page:



This page consists of a banner, then a table with information, and a footer. This page would be used to explain the website to new users.

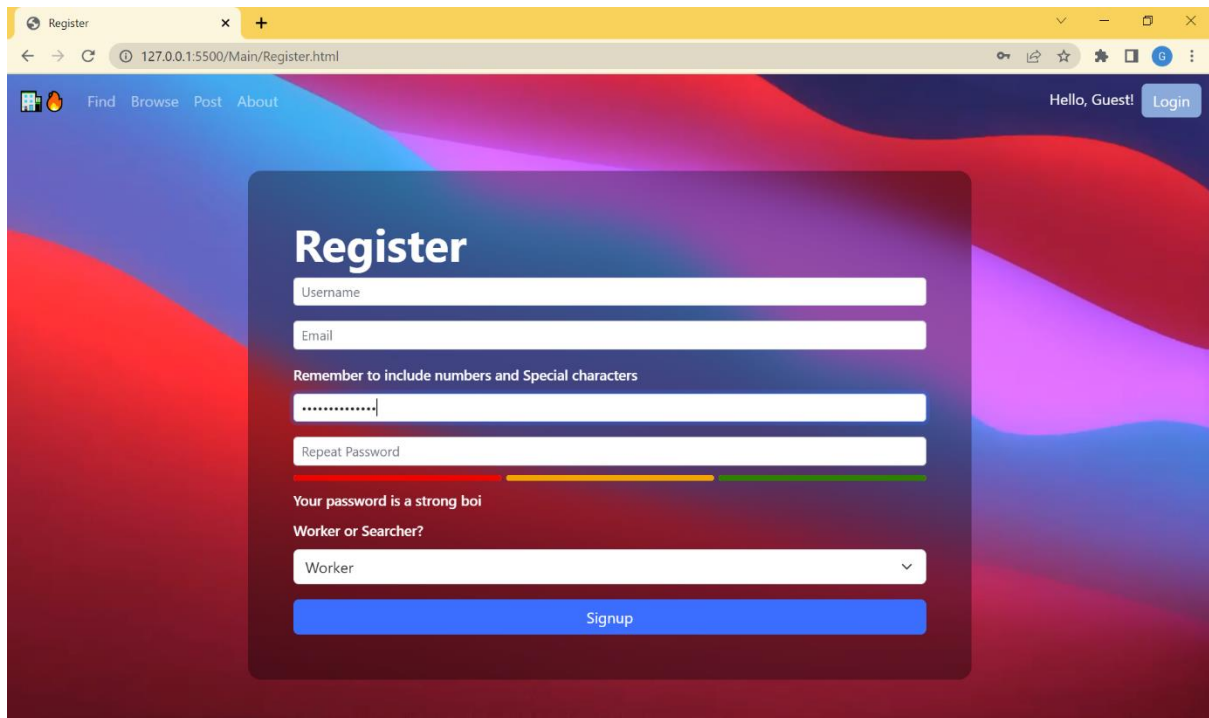
Next we have the Item.html page:



This page consists of a table that procedurally generates the data of the item that was selected on the previous page.

Next, we have the app layout pages, these are used to interact with the user and to receive data. These pages include the Login.html, Register.html, Find.html, Post.html pages. The login and register pages create an in-session login environment. These pages look very similar but are all functionally extremely different.

The register page:



Register

Username

Email

Remember to include numbers and Special characters

\*\*\*\*\*

Repeat Password

Your password is a strong boi

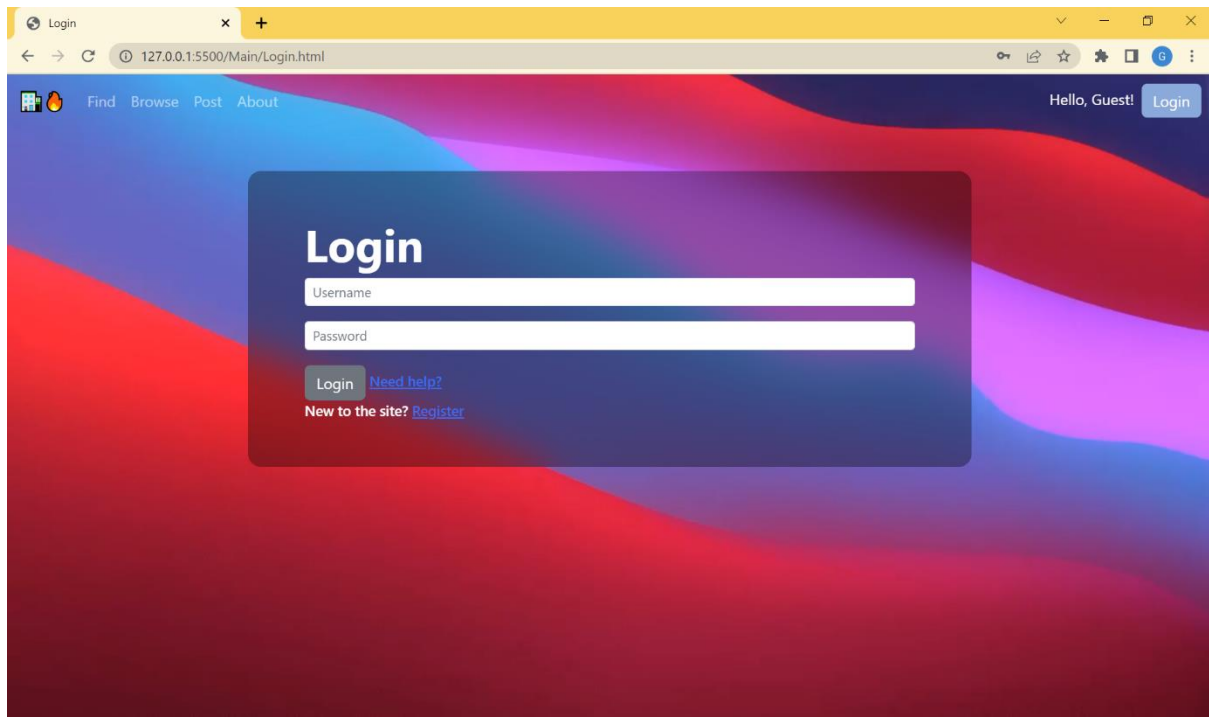
Worker or Searcher?

Worker

Signup

This page has input fields and a password strength checker. All the functions on this page work except the final step of adding the data to the json file. This is due to the time limitations and scope of this project, if I had more time then I would create a server to store the data and handle the POST request.

The login page:



Login

Username

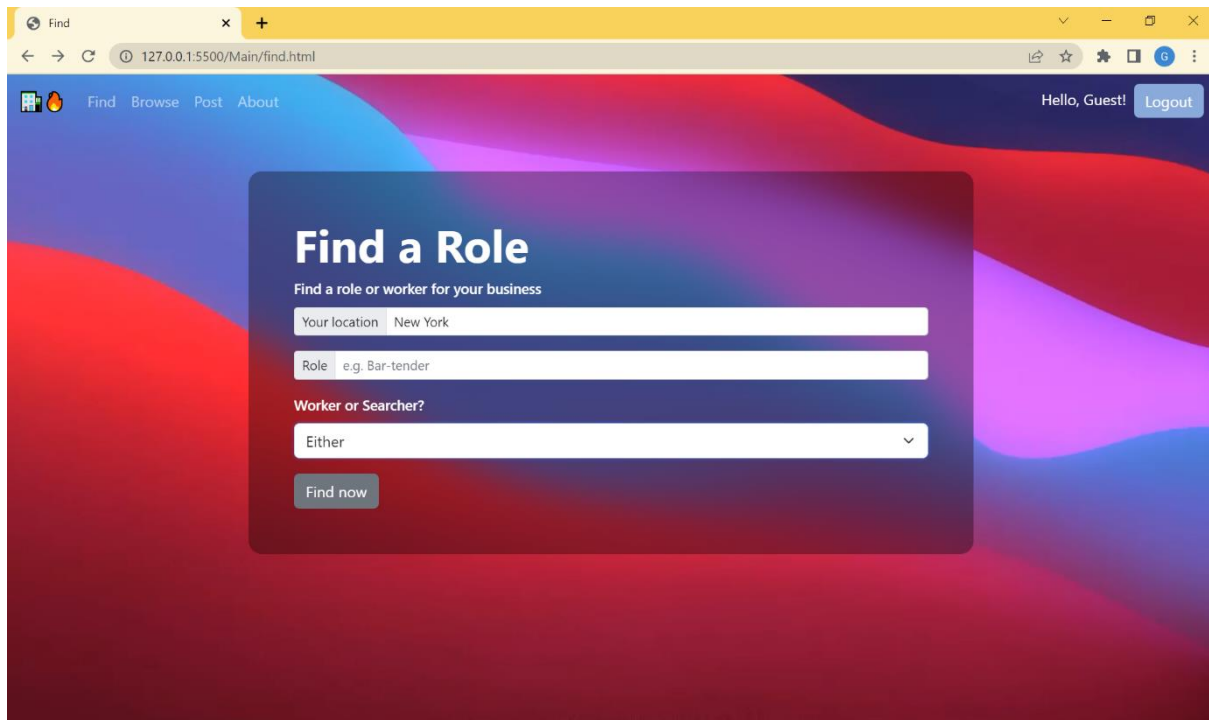
Password

Login [Need help?](#)

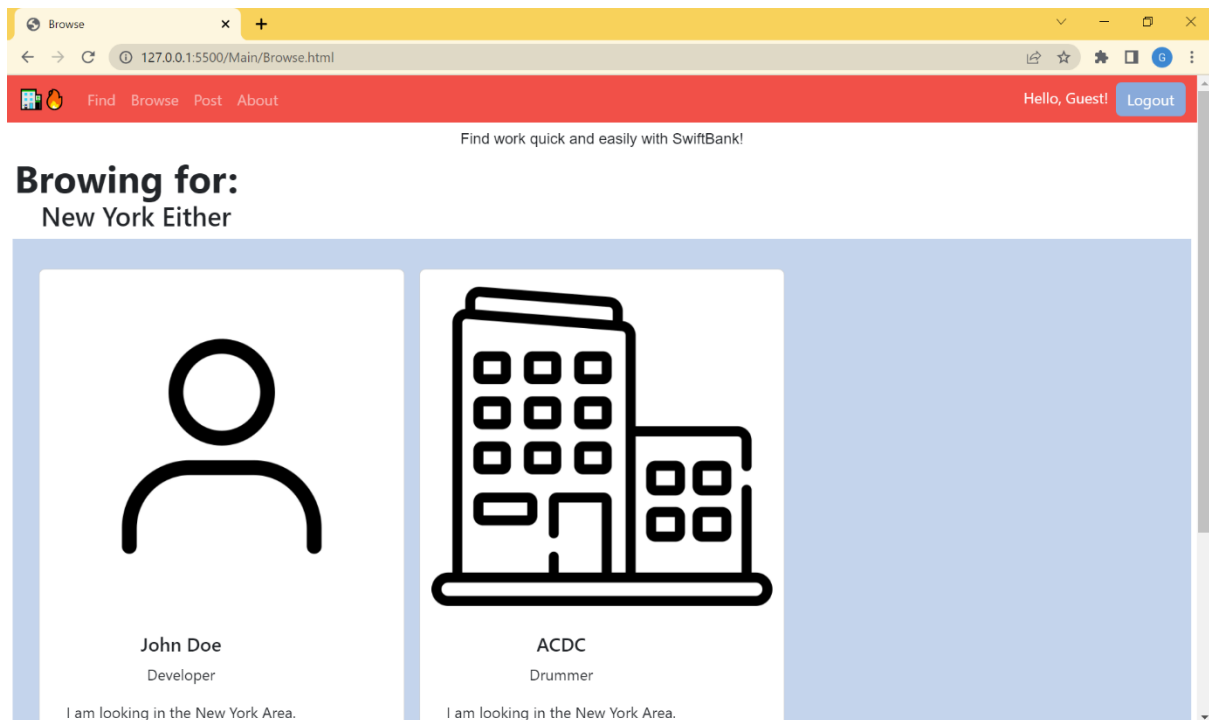
New to the site? [Register](#)

This allows a user to login to an account so the website is personalised. This is fully working. Use the username:"admin" and the password:"admin" to login.

The find page:



This page has input fields used to filter information shown on the Browse.html page. I have input a location into the field and chosen to display both workers and businesses. Here is the result after you press the find button:



The post page:

Post

127.0.0.1:5500/Main/Post.html

Find Browse Post About

Hello, Guest! Logout

## Post a Role

Post a job advertisement or your employee profile

Your location

Role e.g. Bar-tender

Hourly pay £ e.g. 17.50

Extra notes

Duration e.g. 3 days

Start Date e.g. 17/02/2024

Worker or Searcher?

Worker

Post now

This page is used for adding data to the items.json file. This page is mostly functional but once again I do not have a server to hose the json file so I cannot carry out POST requests.