WAPH-Web Application Programming and Hacking

Instructor: Dr. Phu Phung

Student

Name: Isaac Jacobsohn

Email: mailto:jacobsid@mail.uc.edu

Short-bio: Isaac Jacobsohn has keen interests in robotics engineering and software development.

Figure 1: Isaac's headshot

Overview

For project 1 I created a professional profile website and deployed it on github.io cloud service. Link to website: Website Link Link to GitHub repository: GitHub Repository

General Requirements

To create and deploy a personal website on Github, I had to first find a Bootstrap framework. Once I found it and fitted it to my needs, I also needed to create a HTML page that introduced the WAPH class along ith projects.

Non-Technical Requirements

To find a Bootstrap framework for my website, I looked up free Bootstrap frameworks and found one that I liked. I then had to include a page tracker, so I went with the flag counter. The flag counter was added by copying and pasting the necessary text to the website.

Technical Requirements

Using jQuery, I added the digital clock, analog clock, show/hide my email, and made it so the letters in my last name appear letter-by-letter. Using Vue.js, I added a public API with graphics. I chose to find random images of dogs when a button is pressed.

For my 2 public Web API integrations I added a random fact button and the most recent Canada-US currency exchange rate presented by the United States Treasury.

To integrate the jokeAPI where it will display a new joke on my page every minute, I adjusted the codew from Lab 2 to change the calling method from a button click to a set timer, similar to how the digital and analog clocks are updated.

To integrate a public API with graphics and display that image on my page, I first found an API that I liked, which was an API that would give a random photo of a dog. From that, I used Vue.js to make it so that when a button is pressed a new dog photo would be generated. I also had to check if the new dog photo was a mp4 file, so I adjusted the code to constantly call the API until it wasn't a mp4 file.

When using JavaScript cookies, I had to make sure that the person visiting the page was the same person, so I got the username of the user and stored that. I then check whether that person has visited before. If they have, then I welcome them back with when they last visited, and store their current visit. If they're new, then I welcome them to the website.