WAPH-Web Application Programming and Hacking

Instructor: Dr. Phu Phung

Name: Amaan Bilwar

Email: bilwarad@mail.uc.edu

Short-bio: Interested in backend software developement/devOps and enhancing cloud security.

Figure 1: Amaan Bilwar's headshot

Repository Information

Respository's URL: https://github.com/amaan/amaanbilwar.github.io

This is a private repository for Amaan Bilwar to store all code from the course.

Individual Project 1

Front-end Web Development with a Professional Profile Website on github.io cloud service

Overview

In order to receive free hosting on the Github cloud service, we built a front-end web application for this project, a professional portfolio website, and deployed it on `github.io`/github pages. There are three types of needs for this project: general, non-technical, and technical. Additional duties include setting cookies, integrating an API, and creating a flag counter.

For this Project, I have created a public repository that can be accessed through https://github.com/amaanbilwar/amaan.github.io

General Requirements

Made a public repository with the name amaanbilwar.github.io. Then I have created a html file with name `waph.html`. This file includes the course information and the overview of all the completed Labs, Hackathons and Individual Projects. This website can be accessed through https://amaanbilwar.github.io/waph.html
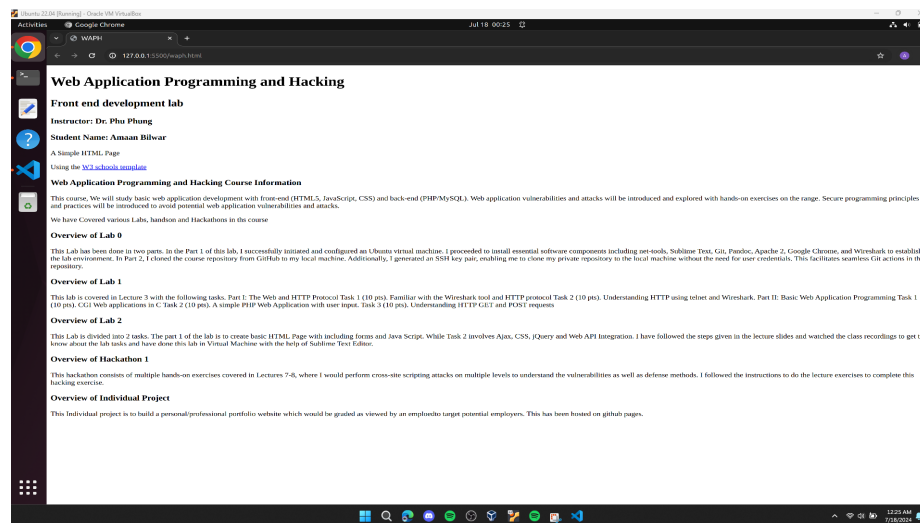

Figure 2: Screenshot of `waph.html`

After cloning,n I have created a html file `idex.html` for my personal website on

Github cloud as a professional profile to add to my resume. This website can be accessed through https://amaanbilwar.github.io/

My professional website includes the following sections: 1. About 2. Experience 3. Education 4. Skills 5. Projects 6. Additional Tasks

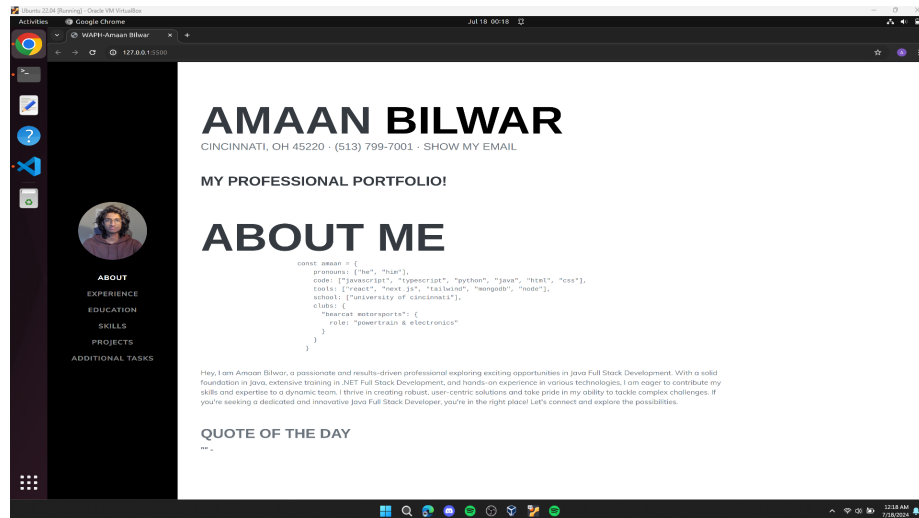Used BootStrap template and changed styling accordingly.

Figure 3: Home Page of `index.html`

You can use the navigation bar to quickly and swiftly navigate around the web page

At the end the website contains all the additional tasks required for this project.

Non-Technical Requirements

Next, for the Non-technical Requirements:

1. Open source bootstrap template.
2. Included a Page tracker, i.e flag counter, which notes the visits of the page and countries where this page has been opened.

Technical Requirements

Coming to the Technical Requirements:

1. The jQuery and Java Script code was introduced in Lab 2 for displaying Digital Clock, Analog Clock, Show/Hide My Email.
2. Used Vue and integrated a public API for fetching a random quote.

The Code for this functionality is in the script tag:

```
new Vue({
    el: '#app',
    data: {
        quote: {
            content: '',
            originator: {
                name: ''
            }
        }
    },
    methods: {
        async fetchQuote() {
            const options = {
                method: 'GET',
                url: 'https://random-quote-generator2.p.rapidapi.com/randomQuote',
                headers: {
                    'X-RapidAPI-Key': 'b9b5a91c2emsh000a90d32a87145p1c5729jsne626d5f90d2
                    'X-RapidAPI-Host': 'random-quote-generator2.p.rapidapi.com'
                }
            };
            try {
                const response = await axios.request(options);

                if (Array.isArray(response.data) && response.data.length > 0) {
                    const quote = response.data[0];

                    this.quote.content = quote.Quote;
                    this.quote.originator.name = quote.Author;
                } else {
                    console.error('Invalid data received from the API:', response.data);
                }
            } catch (error) {
                console.error(error);
            }
        }
    },
    created() {
        // Fetch a quote when the app is created
        this.fetchQuote();

        // Fetch a new quote every 12 hours
        setInterval(this.fetchQuote, 12 * 60 * 60 * 1000);
    }
});
```

Code in HTML to display the Quote:

```
<div class="mt-5">   <p>      <h3 class="text-success">Quote of the
day</h3>     <strong>"{{ quote.content }}"</strong>     <span
style="font-weight: bold;"> - {{ quote.originator.name }}</span>
</p> </div>
```

This functionality is inserted in the home page of the website.

Next, I have integrated 2 public webAPI's into my website.

They are:

1. Integrated the joke API https://v2.jokeapi.dev/joke/Any to display any random joke into the website and displays a new joke for every minute. For this calling, I have set interval for calling the function every minute.

Code:

```
function getJoke() {            $.get("https://v2.jokeapi.dev/joke/Any?type=single",
function (result) {              $("#response").html("A Random
Joke: " + result.joke);          });            } setInterval(getJoke,
60000); getJoke();
```

2. Integrated a public API which is to display the current weather using `https://www.weatherbit.io`. I have generated a API Key and integrated it into my website with all the JSON data returned from the API to display the data I have styled it using CSS and displayed the Temperature, Icon, Location and the weather description.
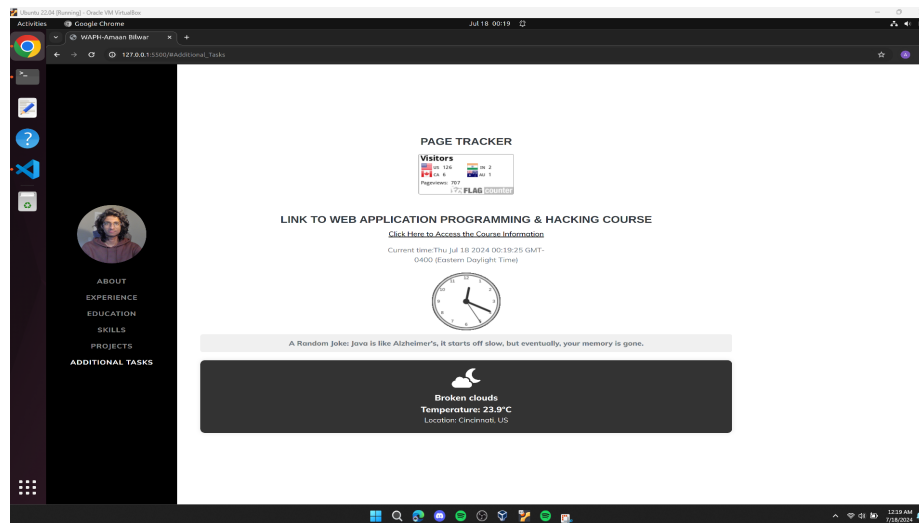

Figure 4: Screenshot of all above Technical Req

Additionally, as per the requirement, I have set cookies to remember the user, So, for the first time a pop up will be displayed saying Welcome to my Home page, and for the next visit it displays a pop up saying Welcome back and the date and time when the user have visited last.
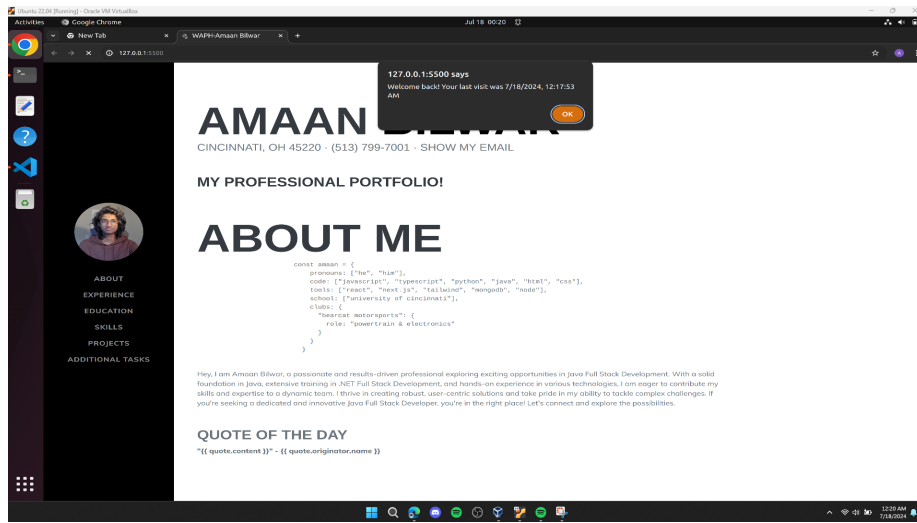

Figure 5: First Time user and Saved Cookies