

waph-meesalji

WAPH-Web APpLiCation Programming and Hacking

Instructor: Dr. PHu Phung

Student

Name: Jahnavi Meesala

Email: meesalji@mail.uc.edu

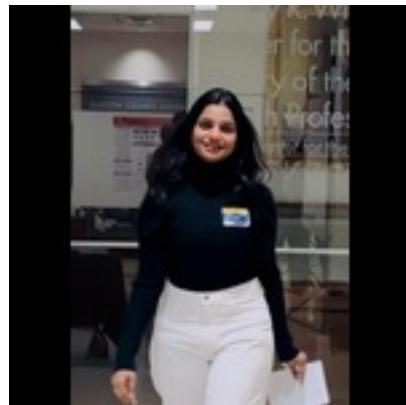


Figure 1: images

Repository links:

url

Here, is the link of my portfolio url

Individual Project 1 : Front-end Web Development with a Professional Profile Website on github.io cloud service

Overview and Requirements

Our goal in this project is to improve our front-end web development skills by building a professional profile website and hosting it on the github.io cloud service. Each of the three requirements—general, non-technical, and technical—contributes to the final grade distribution of the project. Making a personal website showcasing our professional profile—which includes our résumé, contact

details, education, experiences, and skills—is our primary duty. We also need to include a website that introduces the course “Web Application Programming and Hacking” and its associated projects. We’ll integrate page trackers for site traffic monitoring and use open-source CSS frameworks like Bootstrap to ensure a polished appearance and feel. Additionally, the technical specifications cover things like the usage of JavaScript, web API integration, and JavaScript functionality.

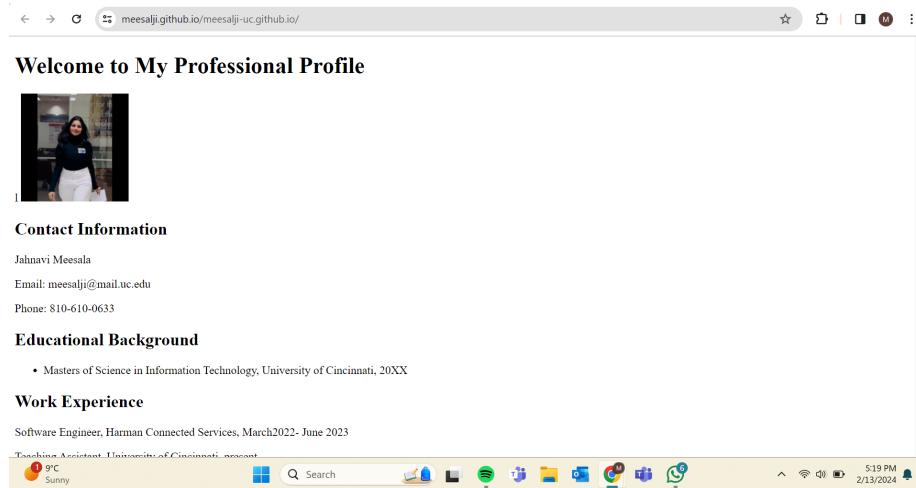
General requirements

Task 1:

Our job in this section of the project is to use GitHub’s cloud service, github.io, to create and launch a personal website. The website will function as an online resume in the sense of a professional profile. Important facts like our name, our professional headshot photo, our contact information, and an overview of our past that includes our training, experiences, and qualifications should all be included.

We hope to effectively introduce ourselves to potential employers through our website by showcasing our skills, experiences, and qualifications in an eye-catching and approachable way. In this task I’m adding the screenshots of the code, resume and all the things I have created from the starting

Here, I’m adding my resume link url



meesalji.github.io/meesalji-uc.github.io/

Contact Information

Jahnnavi Meesalji
Email: meesalji@mail.uc.edu
Phone: 810-610-0633

Educational Background

- Masters of Science in Information Technology, University of Cincinnati, 20XX

Work Experience

Software Engineer, Harman Connected Services, March2022- June 2023
Teaching Assistant, University of Cincinnati, present

Skills

- HTML
- CSS
- JavaScript
- Java, C, Python
- Github

[View Resume](#)

© 20XX My Professional Profile

[meesalji-uc.github.io](#) / index.html ↑ Top

Code Blame 434 lines (390 loc) · 16.2 KB Code 55% faster with GitHub Copilot Code Chat Raw Copy Download Edit More

```
101   <div class="col-lg-5 order-lg-2">
102     <div class="p-5">
103       
104     </div>
105   </div>
106
107   <div class="col-lg-7 order-lg-1">
108     <div class="p-5">
109       <h3 class="name">Hello, I'm Jahnavi Meesala.</h3>
110       <p>A passionate IT professional with a knack for crafting innovative digital solutions. I hold a Master's in Information Technology from the University of Cincinnati, honing my skills in C, Java, Python, and HTML/CSS along the way. Currently, I'm serving as a Graduate Teaching Assistant at Harman Connected Services saw me playing a pivotal role in product development, where I led the development of diverse products, from predictive ML models to Flask-based applications, I'm ready to make a meaningful impact in the tech world.</p>
111       <a href="jahnavi_resume.pdf" class="btn1 btn-lg btn-outline-light">
112         <i class="fa fa-download" aria-hidden="true" id="download"></i>
113         My Resume!</a>
114     </div>
115   </div>
```

JAHNAVI MEESALA

810-610-0653 | meesala.jahnnavi7@gmail.com | <https://www.linkedin.com/in/jahnnavimeesala/>

EDUCATION

University of Cincinnati
Master of Science in Information Technology | GPA: 3.9/4.0

Malla Reddy Engineering College for Women
Bachelor of Technology in Information Technology | GPA: 8.58/10

Aug 2023 — Dec 2024
Cincinnati, Ohio

July 2018 — July 2022
Hyderabad, India

PROGRAMMING SKILLS

- Languages: C, Java, Python, HTML/CSS.
- Developer Tools: Postman, docker
- IT Constructs: Data Structures and Algorithms, SQL

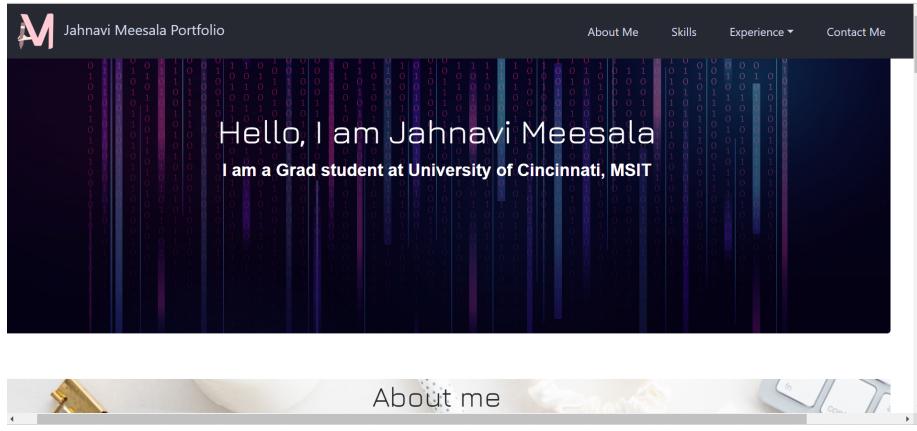
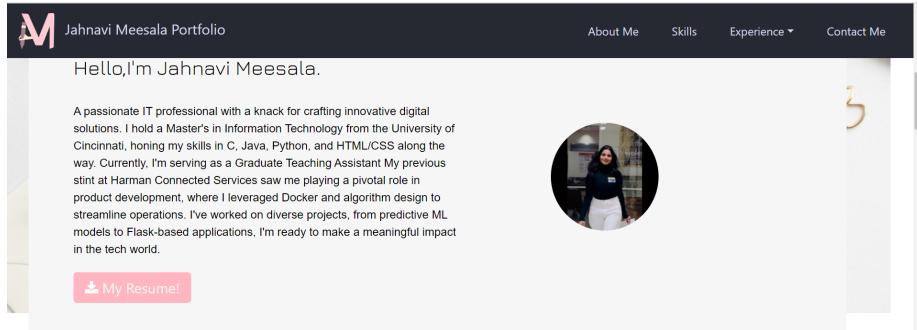
EXPERIENCE

University of Cincinnati
Graduate Teaching Assistant Jan 2024 — Present
Cincinnati, OH

- Assisted in teaching courses related to Human-Computer Interaction and Usability and provided support to students in understanding key concepts and theories in HCI.
- Mentored undergraduate students and projects related to HCI and usability.
- Provided detailed feedback to help students understand strengths and areas for improvement.

University of Cincinnati
Graduate Teaching Assistant Aug 2023 — Dec 2023
Cincinnati, OH

- Engaged as a Teaching Assistant for undergraduate courses such as Computer Programming, Contemporary Programming, and Fundamentals of Web Development.
- Provided individualized guidance to students by grading assignments, addressing queries, and guiding problem-solving methodologies, fostering an environment conducive to effective learning.
- Organized and conducted review sessions, assisting students in understanding complex programming concepts and reinforcing practical application.



I'm adding my waph.html(view portfolio) link

url

Task 2:

In an effort to improve the website's level of content, we've added a link to a fresh HTML page that introduces the "Web Application Programming and Hacking" course and the practical projects that go along with it. With its practical activities pertaining to web application development and hacking techniques, as well as its insights into the course curriculum, this supplement enhances the visitor's experience. Users can easily examine the course specifics and delve into the nuances of web development and security by embedding this link into the navigation bar. This deliberate inclusion strengthens our reputation and relevance in the tech community by highlighting our dedication to openness and thorough depiction of our professional endeavors.

The screenshot shows the GitHub code editor interface for the file `index.html`. The code is as follows:

```

225     <div class="imageContainer col-3">
226         
227     </div>
228     <div class="text3 col-8">
229         <h3>Project</h3>
230         <p>
231             The portfolio showcases a private repository for coursework and links to individual assignments and projects.
232             It offers a comprehensive overview of Jahnavi Meesala's progress and accomplishments in the course.
233         </p>
234         <a href="https://meesalji.github.io/meesalji-uc.github.io/waph.html" class="btn btn-primary">View Portfolio</a>
...

```



Web Application Programming and Hacking

Front-end Web Development Individual Project

Instructor: Dr. Phu Phung

Student: Jahnavi Meesala

A simple HTML page

Using the [W3Schools template](#)

- [Lab 0: Development Environment Setup](#) - Set up your development environment for web programming.
- [Lab 1: Foundations of the Web](#) - Learn the basics of web technologies such as HTML, CSS, and JavaScript.
- [Lab 2: Front-end Web Development](#) - Dive deeper into front-end web development concepts and techniques.
- [Lab Hackathon 1: Cross-site Scripting Attacks and Defenses](#) - Participate in a hackathon focused on XSS attacks and defenses.
- [Individual Project 1: Professional Profile Website](#) - Create and deploy a personal website on GitHub Pages showcasing your professional profile, including your resume, background, experiences, and skills.

Non-technical requirements

Task 3:

When developing our webpages, we used an open-source CSS template called Bootstrap. With its abundance of pre-designed components and ability to streamline responsive design, Bootstrap makes it easier to create aesthetically pleasing and intuitive user interfaces. Furthermore, we have included a page tracker tool on our webpage because we understand how important it is to monitor and evaluate visitor involvement. Through the collection of insightful data on visitor behavior and engagement indicators, we are able to make well-informed judgments on the optimization of content and user experience. By using resources like Google Analytics

Below is the code i have used :

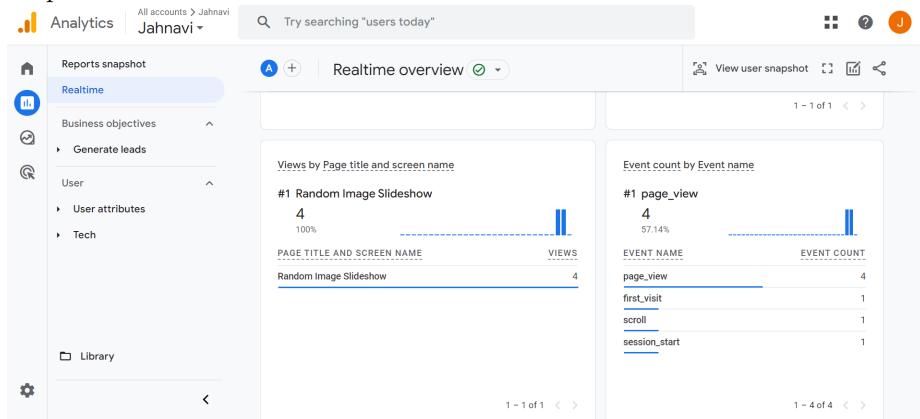
code screenshot

```

25 </style>
26 <!-- Bootstrap CSS -->
27 <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR8iXcbMQv3Xipma34MD+dh/1fQ784/j6cY/lJTQ0hCh"
28 <link href="https://fonts.googleapis.com/css?family=Anton|Jura&display=swap" rel="stylesheet">
29 <link rel="stylesheet" href="css/style.css">
30 <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
31
32
33 <title>Jahnavi Meesala's Portfolio</title>
34
35 <!-- Google Tag (gtag.js) -->
36 <script async src="https://www.googletagmanager.com/gtag/js?id=G-9030MR8VRT"></script>
37 <script>
38   window.dataLayer = window.dataLayer || [];
39   function gtag(){dataLayer.push(arguments);}
40   gtag('js', new Date());
41
42   gtag('config', 'G-9030MR8VRT');
43 </script>
44

```

output



Technical requirements

Basic JavaScript code

Task 4

Together with another open-source JavaScript framework/library, we've used jQuery to improve user experience and interactivity. Expanding on ideas presented in Lab 2, we've added a number of features, such as an analog and digital clock, the option to display or conceal email addresses, and additional functionality catered to our particular need.

Show my email

The following code snippet demonstrates how to integrate standard JavaScript features like showing and hiding email addresses and digital and analog clocks. These features improve our webpage's interaction and user engagement while showcasing our expertise with JavaScript and jQuery usage.

code

meesalji-uc.github.io / index.html

Code Blame 434 lines (390 loc) • 16.2 KB Code 55% faster with GitHub Copilot ↑ Top

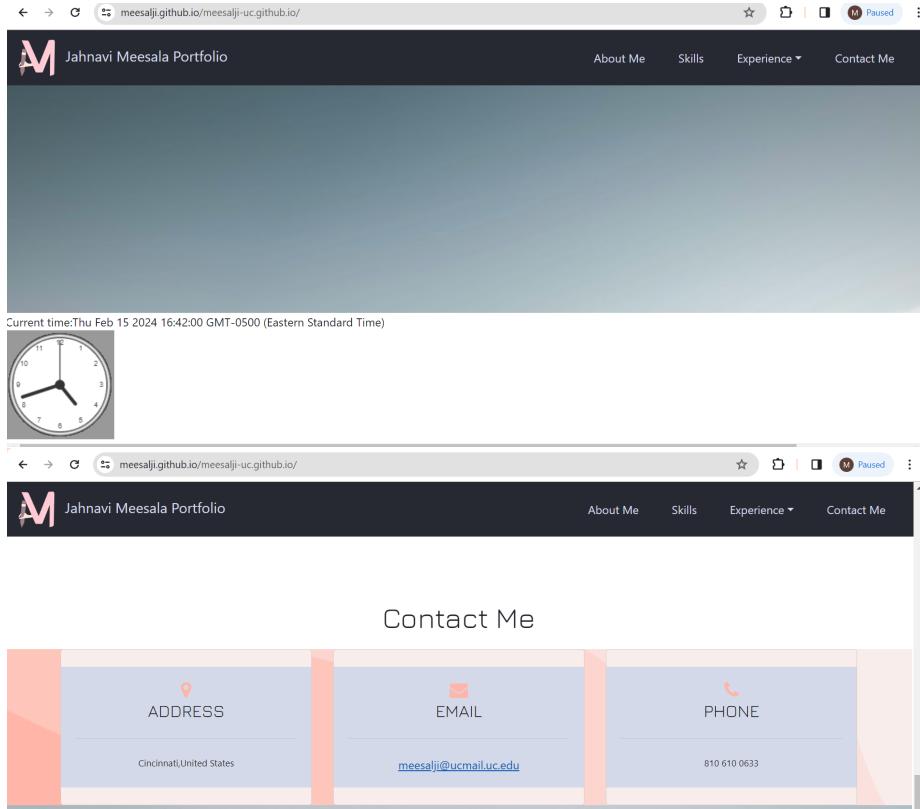
```
267     
288   </div>
289   <!-- Digital Clock -->
290   <div id="digital-clock"></div>
291   <canvas id="analog-clock" width="150" height="150" style="background-color:#999">
292     <script src="https://waph-uc.github.io/clock.js"></script>
293     <script type="text/javascript">function displayTime(){
294       document.getElementById('digital-clock').innerHTML="Current time:"+new Date();
295     }
296     setInterval(displayTime,500);
297     var canvas=document.getElementById("analog-clock");
298     var ctx = canvas.getContext("2d");
299     var radius = canvas.height / 2;
300     ctx.translate(radius,radius);
301     radius=radius * 0.90;
302     setInterval(drawClock,1000);
303     function drawClock() {
304       drawFace(ctx, radius);
305       drawNumbers(ctx, radius);
306       drawTime(ctx,radius);
307     }
308   </script>
309 </div>
```

meesalji-uc.github.io / index.html

Code Blame 434 lines (390 loc) • 16.2 KB Code 55% faster with GitHub Copilot ↑ Top

```
223   <hr class="text-uppercase m-0" data-eq="142">
256   <hr class="my-4" />
257   <div class="small">cincinnati,United States</div>
258   </div>
259   </div>
260   </div>
261   <div class="col-md-4 mb-3 mb-md-0">
262     <div class="card py-4 h-100">
263       <div class="card-body text-center"> <!-- Corrected class name here -->
264         <i class="fa fa-envelope" aria-hidden="true" id="icon"></i>
265         <h4 class="text-uppercase m-0">Email</h4>
266         <hr class="my-4" />
267         <div id="email" onclick="showhideEmail()">Show my email</div>
268         <script src="email.js"></script>
269       </div>
270     </div>
```

output



Our webpage's background color is dynamically changed by a jQuery feature that we've included. This feature improves the user experience overall by introducing a little personalization and interaction. The jQuery function that listens for a button click event is implemented in this snippet of code. The chosen element's (color-element) background color dynamically changes to a randomly generated hexadecimal color when clicking the button.

Change Background Color

screenshot

```

meesalji-uc.github.io / index.html
Code Blame 434 lines (390 loc) · 16.2 KB Code 55% faster with GitHub Copilot ↑ Top
393     displayWelcomeMessage();
394   });
395 


meesalji.github.io/meesalji-uc.github.io/ Paused



M Jahnnavi Meesala Portfolio About Me Skills Experience Contact Me



What do you call a witch at the beach?  
A Sandwich.



Change Background Color



Contact Me


```

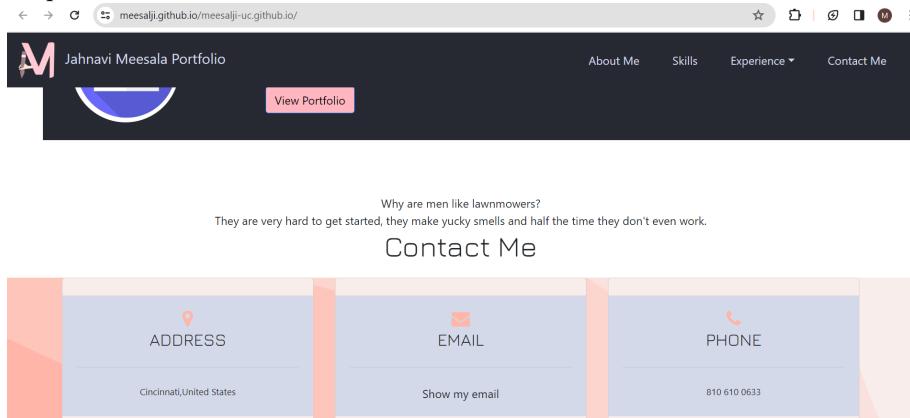
Two public Web APIs integration

Task 5

The jokeAPI has been incorporated into our current HTML code. With the help of this integration, we can retrieve a fresh joke every minute from the Any category of the jokeAPI and show it on our page. We've developed a function that uses JavaScript to retrieve a random joke from the API. We've included JavaScript in our HTML code to retrieve data from the jokeAPI endpoint using the fetch API. We update the relevant portion of our webpage with the retrieved joke content after retrieving the joke data.

code

output



Task 6:

I have used this public API with graphics <https://unsplash.com/oauth/applications/567692>, I've included JavaScript routines that communicate with the Unsplash API into the current HTML code. These routines retrieve random images from the Unsplash library in response to their requests, which include parameters like image size and category, sent to the Unsplash server. The JavaScript code instantly refreshes the image source on your webpage as soon as it gets a response from the Unsplash API. This increases the visual appeal of the website and maintains content freshness by displaying a fresh, randomly chosen image to visitors each time they load or refresh it.

code

meesalji-uc.github.io / index.html

Code Blame 434 lines (390 loc) • 16.2 KB Code 55% faster with GitHub Copilot Code Chat Raw ⌂ ⌄ ⌅ ⌆ ⌇ ⌈ ⌉ ↑ Top

```
322     </script>
323
324     <script>
325         // Function to fetch a random image from Unsplash API
326         function fetchRandomImage() {
327             fetch('https://api.unsplash.com/photos/random?client_id=2bAwIZZZXN0s5rDV4grqBxmFD3hdigLSh_IBeWqxQY')
328                 .then(response => response.json())
329                 .then(data => {
330                     // Update image source with the fetched random image
331                     document.getElementById('random-image').src = data.urls.regular;
332                 })
333                 .catch(error => console.error('Error fetching random image:', error));
334             ||
335         }
```

meesalji-uc.github.io / index.html

Code Blame 434 lines (390 loc) • 16.2 KB Code 55% faster with GitHub Copilot Code Chat Raw ⌂ ⌄ ⌅ ⌆ ⌇ ⌈ ⌉ ↑ Top

```
329     .then(data => {
330         // Update image source with the fetched random image
331         document.getElementById('random-image').src = data.urls.regular;
332     })
333     .catch(error => console.error('Error fetching random image:', error));
334 }
335
336 // Function to change image every 2 seconds
337 function changeImage() {
338     fetchRandomImage();
339     setTimeout(changeImage, 2000); // Call the function recursively after 2 seconds.
340 }
341
342 // Call the changeImage function to start the slideshow
343 changeImage();
344 </script>
```

meesalji.github.io/meesalji-uc.github.io/index.html

Jahnavi Meesala Portfolio About Me Skills Experience Contact Me Cincinnati, United States Show my email 810 610 0633



Current time: Thu Feb 15 2024 14:28:44 GMT-0500 (Eastern Standard Time)

A. Welcome to my homepage

We'll develop JavaScript code that looks for the existence of a cookie containing the client's most recent visit in order to provide this capability. It's the user's first visit if the cookie is absent, in which case a "Welcome to my homepage!" message will appear. If not, the timestamp will be retrieved from the cookie and a "Welcome back!" message will be displayed.

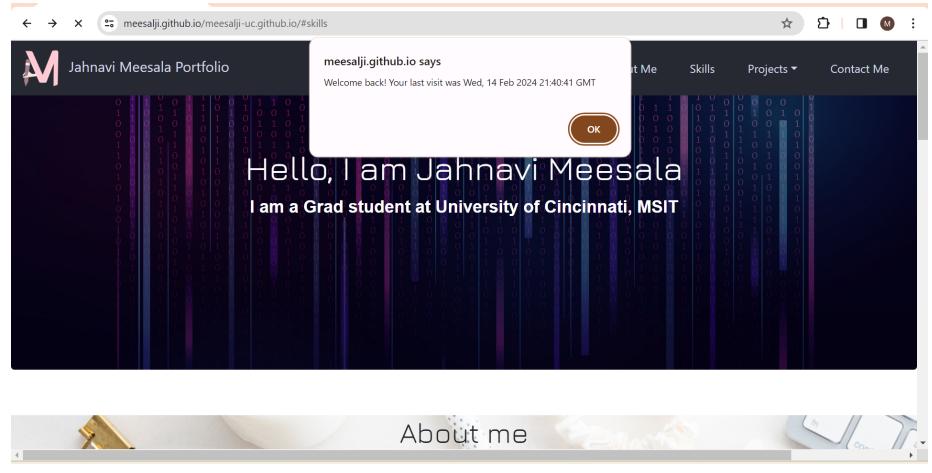
The screenshot shows the GitHub Copilot interface for a file named index.html. The code is written in JavaScript and includes functions for setting and getting cookies, and displaying a welcome message based on the cookie value. The GitHub Copilot interface has tabs for 'Code', 'Blame', and 'Raw'. It also shows statistics like 434 lines (390 loc) and 16.2 KB. A note indicates that the code is 55% faster with GitHub Copilot. The code itself is as follows:

```

355 <script>
356 // Function to set a cookie
357 function setCookie(name, value, days) {
358   const expires = new Date();
359   expires.setTime(expires.getTime() + days * 24 * 60 * 60 * 1000);
360   document.cookie = `${name}=${value};expires=${expires.toUTCString()};path=/`;
361 }
362
363 // Function to get the value of a cookie by name
364 function getCookie(name) {
365   const keyValue = document.cookie.match(`(^|;) ?${name}=([^;]+)(;|$)`);
366   return keyValue ? keyValue[2] : null;
367 }
368
369 // Function to display a welcome message based on cookie
370 function displayWelcomeMessage() {
371   const lastVisit = getCookie('lastVisit');
372
373   if (!lastVisit) {
374     // First-time visit
375     const currentDate = new Date();
376     const day = currentDate.getDate();
377     const month = currentDate.getMonth() + 1; // Months are zero-based
378     const year = currentDate.getFullYear();
379     const hours = currentDate.getHours();
380     const minutes = currentDate.getMinutes();
381     const seconds = currentDate.getSeconds();
382     const formattedDate = `${day}/${month}/${year} ${hours}:${minutes}:${seconds}`;
383     alert('Welcome to my homepage!');
384     setCookie('lastVisit', formattedDate, 365); // Set cookie to expire in 365 days
385   } else {
386     // Returning visit
387     alert('Welcome back! Your last visit was ${lastVisit}');
388   }
389 }
390
391 // Call the function on page load
392 $(document).ready(function() {
393   displayWelcomeMessage();
394 });
395 </script>
396
397 <!-- jQuery -->
398 <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
399 <script>
```

Below is the screenshot of the code:

This screenshot shows the same GitHub Copilot interface as the previous one, but with syntax highlighting applied to the code. The code remains the same as above, but the tokens are highlighted in different colors. The GitHub Copilot interface includes tabs for 'Code', 'Blame', and 'Raw', and shows the same file statistics and GitHub Copilot integration note.



output of the code:

Submission

Pandoc tool is used to generate the report in pdf format from the README.md file. The pdf is named as meesalji-waph-project.pdf