

# **WAPH-Web Application Programming and Hacking**

**Instructor: Dr. Phu Phung**

**Name: Lohit Naidu Lalam**

**ID: 101825796**



## **Lab 1 - Foundations of the Web**

### **Report**

#### **The lab's overview**

- The goal of this assignment was to create and launch a professional portfolio website that showcases my project work, abilities, experiences, and academic background. In addition, this website illustrates fundamental ideas covered in the Web Application Programming and Hacking course, including web development, GitHub deployment, JavaScript interactivity, API integration, and cookie handling.

I picked up information about how to:

Build a personal website with Bootstrap and HTML5/CSS3.

Utilize APIs to incorporate real-time content and dynamic JavaScript features.

Keep your layout responsive and your design up to date.

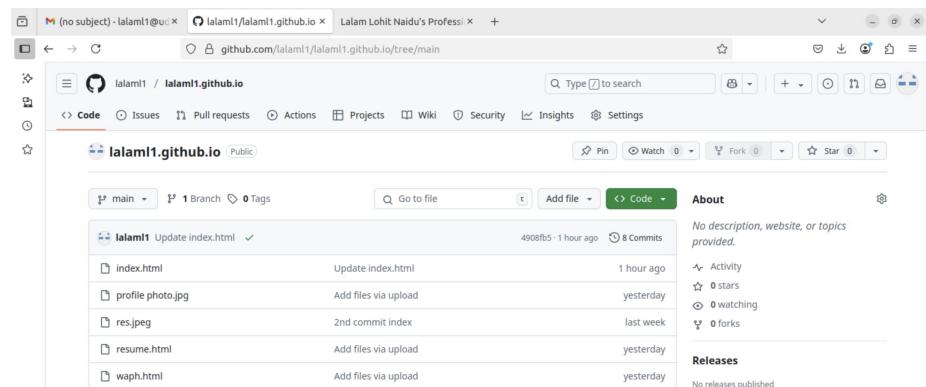
Website in real time: <https://lalam1.github.io/index.html>

This is the GitHub repository: <https://github.com/lalam1/lalam1.github.io>

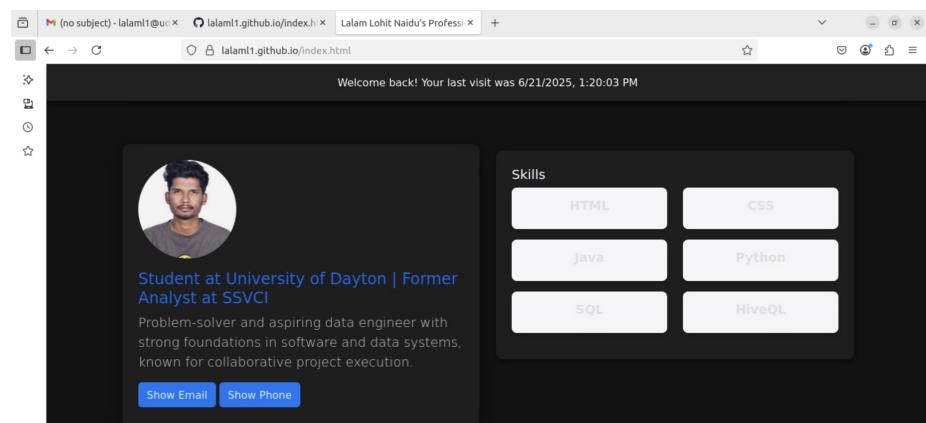
## Part 1- General Requirements

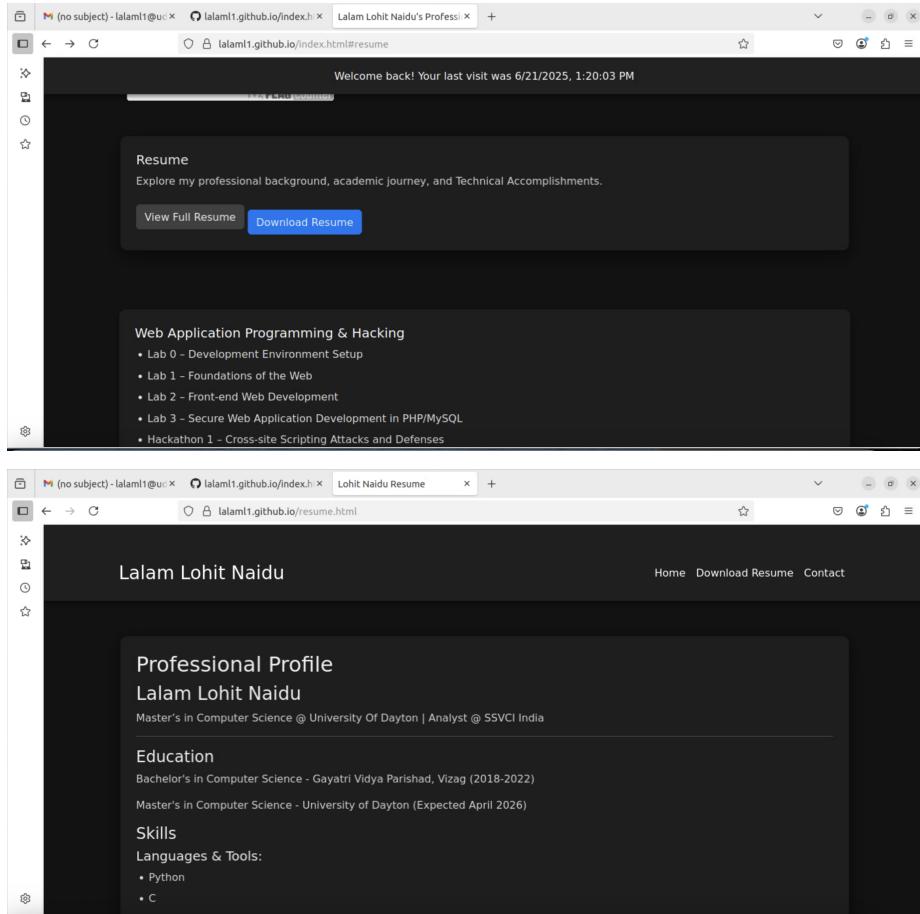
### 1. Create and Deploy a Personal Website

- I develop and host a personal website with bootstrap responsiveness which implemented
- And i added sections like About Me, Resume, skills, Projects, eResume is on-page and also downloadable as a PDF. Headshot image and real contact information included.
- Webpage hosted at GitHub Pages so the public can view it.



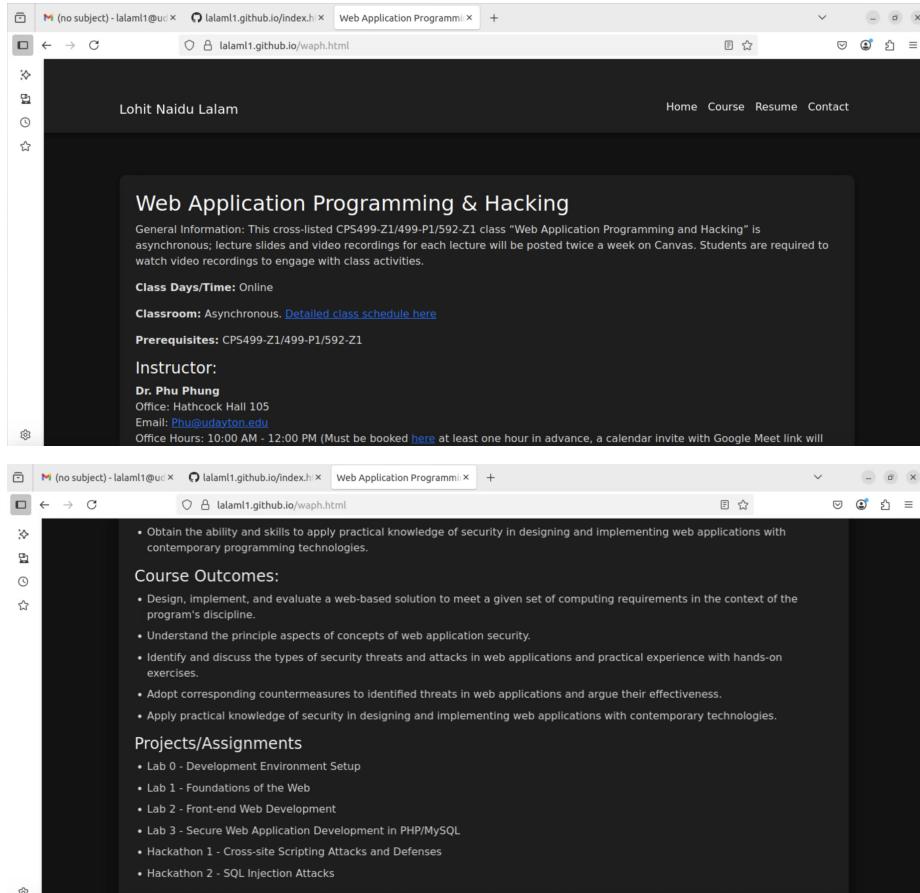
- I added hyperlinks for my skills which if we touch on the skills it will redirects to the skill related data.
- I have added hyperlinks to index.html and resume.html





## 2 Add Course Intro Page

- I have created a new page for web application course details and I named it as waph.html and I added the course details into that
- I also added the hyperlinks for some of them like professor mail and TA mail IDs which is easy by clicking on it redirects easily.



## Task 2: Non-Technical Requirements

### part 1

- I have used bootstrap 5 from CDN to design different kind of representative layout which looks good and it also designs the navigation bars, profile cards, buttons and grids.
- Basically it was inserted in the `<body>` block of the html code.
- I have Included this bootstrap `https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css` code in the head tag in every html code individually, so it runs the code good.
- And this made the website friendly and easy to navigate for everyone.

-/waph/lalam1.github.io/resume.html - Sublime Text (UNREGISTERED)

```

File Edit Selection Find View Goto Tools Project Preferences Help
4 ► waph.html resume.html index.html # WAPH-Web Application Programming and Hacking README.md
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Lalit Naidu's Professional Profile</title>
7   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
8 </head>
9 <body>
10  font-family: 'Poppins', sans-serif;
11  background-color: #121212;
12  color: #ebedeb;
13  padding: 0;
14  overflow-x: hidden;
15  perspective: 1500px;
16  }
17
18 .navbar {
19   font-family: 'Poppins', sans-serif;
20   background-color: #121212;
21   color: #ebedeb;
22   padding: 0;
23   margin: 0;
24   overflow-x: hidden;
25   perspective: 1500px;
26
27 @keyframes slideIn {
28   0% { transform: translateY(100%); }
29   100% { transform: translateY(0); }
30 }
31
32 .navbar a {
33   color: #ffffff !important;
34   text-decoration: none;
35   transition: color 0.3s ease, transform 0.3s ease;
36 }
37
38 .navbar a:hover {
39   color: #ff1f1f !important;
40   transform: translateX(10px) scale(1.05);
41 }
42

```

-/waph/lalam1.github.io/waph.html - Sublime Text (UNREGISTERED)

```

File Edit Selection Find View Goto Tools Project Preferences Help
4 ► waph.html resume.html index.html # WAPH-Web Application Programming and Hacking README.md
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Lalit Naidu's Professional Profile</title>
7   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
8 </head>
9 <body>
10  font-family: 'Poppins', sans-serif;
11  background-color: #121212;
12  color: #ebedeb;
13  padding: 0;
14  margin: 0;
15  overflow-x: hidden;
16  perspective: 1500px;
17
18 .navbar {
19   font-family: 'Poppins', sans-serif;
20   background-color: #121212;
21   color: #ebedeb;
22   padding: 0;
23   margin: 0;
24   overflow-x: hidden;
25   perspective: 1500px;
26
27 @keyframes slideIn {
28   0% { transform: translateY(100%); }
29   100% { transform: translateY(0); }
30 }
31
32 .navbar a {
33   color: #ffffff !important;
34   text-decoration: none;
35   transition: color 0.3s ease, transform 0.3s ease;
36 }
37
38 .navbar a:hover {
39   color: #ff1f1f !important;
40   transform: translateX(10px) scale(1.05);
41 }
42

```

-/waph/lalam1.github.io/index.html - Sublime Text (UNREGISTERED)

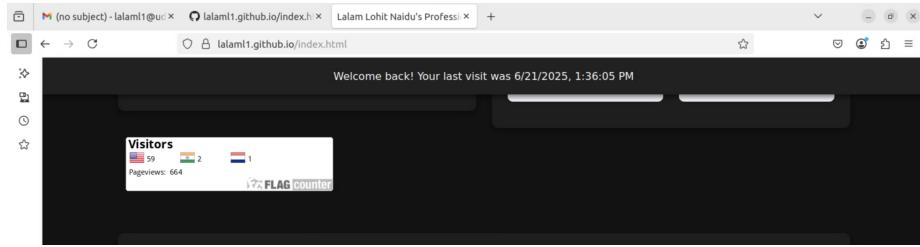
```

File Edit Selection Find View Goto Tools Project Preferences Help
4 ► waph.html resume.html index.html # WAPH-Web Application Programming and Hacking README.md
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Lalit Naidu's Professional Profile</title>
7   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
8 </head>
9 <body>
10  <div class="container" style="text-align: center; margin-top: 100px;>
11    <h3>Lalit Naidu</h3>
12    
13    <p>Software Developer & Web Application Developer<br/>Dayton, OH, USA<br/>Email: maitalialam1@dayton.edu</p>
14    <hr style="margin: 10px 0; border: 0; border-top: 1px solid #121212; width: 200px; margin-left: auto; margin-right: auto;">
15    <div style="background-color: #121212; color: #ebedeb; padding: 5px; border-radius: 5px; margin-bottom: 10px; width: fit-content; margin-left: auto; margin-right: auto;">
16      <span>About Me</span>
17      <span>Skills</span>
18      <span>Education</span>
19      <span>Experience</span>
20      <span>Projects</span>
21      <span>Contact</span>
22    </div>
23    <div style="background-color: #121212; color: #ebedeb; padding: 5px; border-radius: 5px; width: fit-content; margin-left: auto; margin-right: auto; margin-top: 10px;">
24      <span>Cookie Alert ...</span>
25      <span>Cookie Alert</span>
26    </div>
27  </div>
28  <div style="font-family: 'Poppins', sans-serif; background-color: #121212; color: #ebedeb; padding: 0; overflow-x: hidden; perspective: 1500px; position: relative; height: 100%; width: 100%;>
29    <div style="position: absolute; top: 0; left: 0; width: 100%; height: 100%; background: linear-gradient(to bottom right, transparent 45%, black 45%, black 55%, transparent 55%);>
30    </div>
31    <div style="position: absolute; top: 0; left: 0; width: 100%; height: 100%; background: linear-gradient(to bottom right, transparent 45%, black 45%, black 55%, transparent 55%);>
32    </div>
33  </div>
34

```

## part 2 : Page Tracker

- I used flag counter in the code and I also added tracker at the bottom of the code which helps that how many people visited the page.
- It shows the stats of the home page.



- I have used the url to integrate the flag counter is [www.flagcounter.com](http://www.flagcounter.com)
- We can customize different kinds of ways for the flag counter by defining the url `bg_FFFFFF/txt_000000`

### Task 3: Technical Requirements

#### part 1: Digital clock and Analog clock

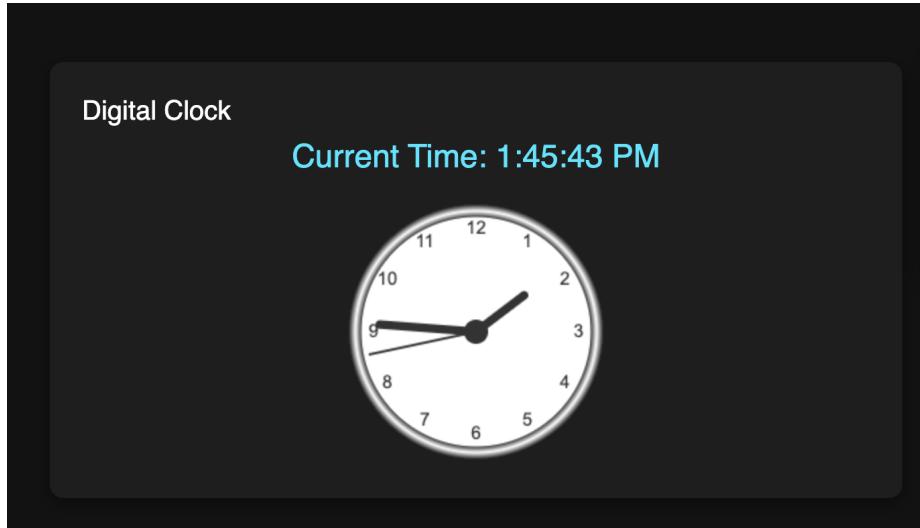
- In this we are creating the digital clock and analog clock for the website in which I used the same code.
- It displays both analog and digital clock at one place.
- The time updates every second and it shows in the home page

Screenshot of a GitHub repository showing the file `index.html`. The code includes a digital clock, an analog clock, and a joke API call.

```

lalamit.github.io / index.html
Code Blame 677 lines (668 loc) · 17.8 KB ⚡ Code 55% faster with GitHub Copilot
449 </div>
450 <small class="text-muted">Weather data provided by OpenWeatherMap. We are not responsible for the accuracy of this information.</small>
451 </div>
452 </div>
453 </div>
454 </div>
455 <!-- Footer -->
456 <div class="container mt-5 fadeInIn">
457 <div class="row">
458 <div class="col-md-6">
459 <div class="profile-card scroll-animate">
460 <div class="profile-card scroll-animate">
461 <h5>Digital Clock</h5>
462 <h1 id="digital-clock" class="text-center mb-3" style="text-align: center;"></h1>
463 <div class="row justify-content-center" style="margin-top: 20px;">
464 <div id="analog-clock" width="200" height="200" style="display: block; margin: 0 auto;"></div>
465 <script src="https://waph-uc.github.io/clock.js"></script>
466 </div>
467 </div>
468 </div>
469 <div class="col-md-6">
470 <div class="profile-card scroll-animate">
471 <h5>Random Joke</h5>
472 <p id="jokeText">Loading joke...</p>
473 <small class="text-muted">Powered by JokeAPI.dev. We are not responsible for the content of this joke.</small>
474 </div>
475 <div class="profile-card scroll-animate mt-4">
476 <h5>Stopwatch</h5>
477 <div id="stopwatch-display">00:00:00</div>
478 <div class="stopwatch-controls mt-3">
479 <button id="start-stopwatch" class="btn btn-primary">Start</button>

```



## Show/Hide Email

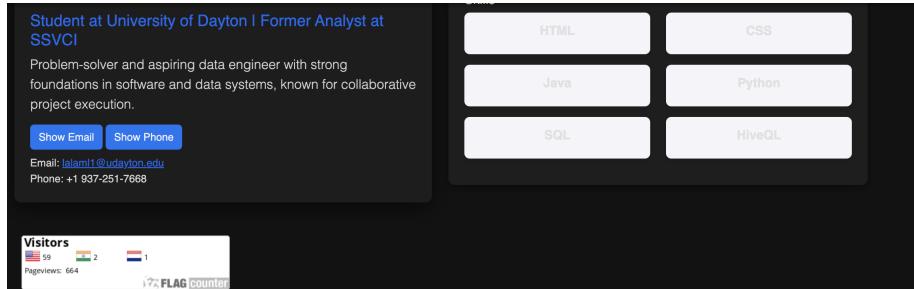
- In this I wrote a function that allows users to show/Hide Email I'd and phone number.

```

Code | Blame | 677 lines (608 loc) · 17.8 KB | Code 55% faster with GitHub Copilot
281 <!-- Home-->
282 <div class="container mt-5 fadeIn" id="home">
283   <div class="row">
284     <div class="col-md-6">
285       <div class="profile-card scroll-animate">
286         Student at University of Dayton | Former Analyst at SSVCI</h4>
288         <p class="lead">Problem-solver and aspiring data engineer with strong foundations in software and data systems, known for collaborative project
289         <button class="btn btn-primary mb-2" onclick="toggleEmail()>Show Email</button>
290         <button class="btn btn-primary mb-2" onclick="togglePhone()>Show Phone</button>
291         <div id="email-info" style="display: none;">
292           Email: <a href="mailto:lalam1@udayton.edu">lalam1@udayton.edu</a>
293         </div>
294         <div id="phone-info" style="display: none;">
295           Phone: +1 937-251-7668
296         </div>
297
298 <script>
  ...
</script>

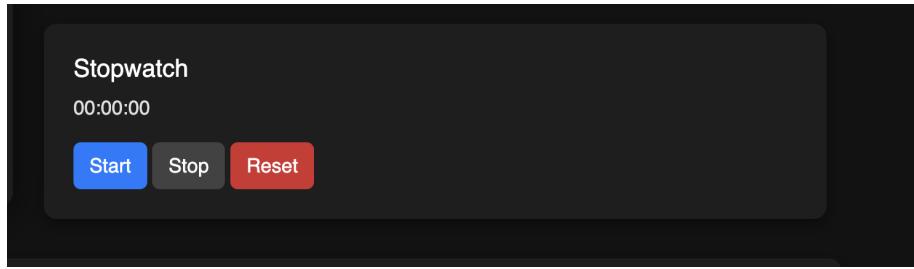
```

- I have created a button and under that button I have added email id and mobile number inside the div tags.
- The link I injected in the code is



## Stop Watch

- I had used the functionality stop Watch for the code.
- It includes the start, stop and reset buttons in the code which interacts with the display.



[lalam1.github.io / index.html](https://lalam1.github.io/index.html) ↑ Top

Code Blame 677 lines (608 loc) · 17.8 KB ⚡ Code 55% faster with GitHub Copilot

```

605 });
606
607 document.addEventListener('DOMContentLoaded', function () {
608     let startTime;
609     let elapsedTime = 0;
610     let isRunning = false;
611     let animationFrameId;
612
613     const stopwatchDisplay = document.getElementById('stopwatch-display');
614     const startButton = document.getElementById('start-stopwatch');
615     const stopButton = document.getElementById('stop-stopwatch');
616     const resetButton = document.getElementById('reset-stopwatch');
617
618     function updateStopwatchDisplay() {
619         const totalMilliseconds = elapsedTime;
620         const totalSeconds = Math.floor(totalMilliseconds / 1000);
621         const milliseconds = Math.floor((totalMilliseconds % 1000) / 10);
622         const hours = Math.floor(totalSeconds / 3600);
623         const minutes = Math.floor((totalSeconds % 3600) / 60);
624         const seconds = totalSeconds % 60;
625
626         stopwatchDisplay.textContent =
627             `${String(hours).padStart(2, '0')}:${String(minutes).padStart(2, '0')}:${String(seconds).padStart(2, '0')}.${String(milliseconds).padStart(2, '0')}`;
628     }
629
630     function step(timestamp) {
631         if (!startTime) startTime = timestamp - elapsedTime;
632         elapsedTime = timestamp - startTime;
633         updateStopwatchDisplay();
634     }
635
636     startButton.addEventListener('click', () => {
637         if (!isRunning) {
638             isRunning = true;
639             step(timestamp);
640             animationFrameId = requestAnimationFrame(step);
641         }
642     });
643
644     stopButton.addEventListener('click', () => {
645         if (isRunning) {
646             isRunning = false;
647             cancelAnimationFrame(animationFrameId);
648         }
649     });
650
651     resetButton.addEventListener('click', () => {
652         isRunning = false;
653         startTime = null;
654         elapsedTime = 0;
655         updateStopwatchDisplay();
656     });
657
658     updateStopwatchDisplay();
659 }

```

## Joke API

- I integrated the joke API which helps to provide a new joke for every minute in the website

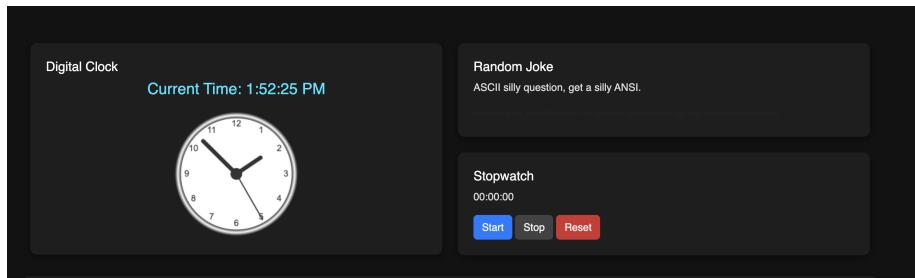


lalam1.github.io / index.html

Code Blame 677 lines (608 loc) · 17.8 KB ⚡ Code 55% faster with GitHub Copilot

```
533     function drawClock() {
534       drawFace(ctx, radius);
535       drawNumbers(ctx, radius);
536       drawTime(ctx, radius);
537     }
538
539
540   // Joke API
541   function fetchJoke() {
542     fetch('https://v2.jokeapi.dev/joke/Any?type=single')
543       .then(response => response.json())
544       .then(data => {
545         document.getElementById("jokeText").innerText = data.joke;
546       });
547     }
548   fetchJoke();
549   setInterval(fetchJoke, 60000); // Refresh every 1 minute
550
551
```

- I included a disclaimer that the jokes and content are generated and we are not responsible for that.
- A new joke appears for every minute by using set interval and fetch API.



## XKCD comics

- I used a which shows new comic image for every time when we refresh the page.
- It includes the image and comic tile in the page.
- It also added a note indicating that the content is from API .

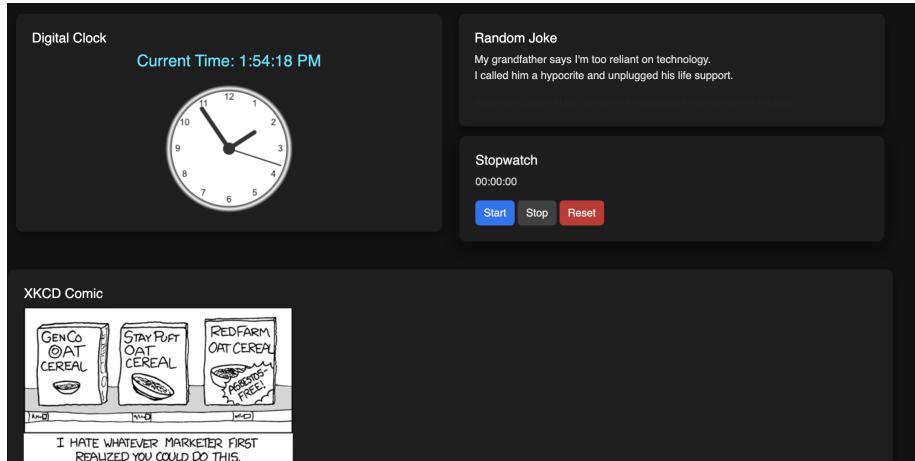
lalam1.github.io / index.html

Code Blame 677 lines (608 loc) · 17.8 KB ⚡ Code 55% faster with GitHub Copilot

```

551     function getXKCDComic() {
552       const minComicId = 1;
553       const maxComicId = 2500;
554       const randomComicId = Math.floor(Math.random() * (maxComicId - minComicId + 1)) + minComicId;
555
556       fetch(`https://xkcd.vercel.app/?comic=${randomComicId}`)
557         .then(response => {
558           if (!response.ok) {
559             throw new Error(`HTTP error! Status: ${response.status}`);
560           }
561           return response.json();
562         })
563         .then(data => {
564           const comicContainer = document.getElementById('xkcd-container');
565           if (data.img) {
566             comicContainer.innerHTML = `
567               
568               <p>${data.title}</p>
569             `;
570           } else {
571             comicContainer.innerHTML = '<p>Failed to load XKCD comic image.</p>';
572           }
573         })
574       .catch(error => {
575         console.error(`Error fetching XKCD comic:`, error);
576         const comicContainer = document.getElementById('xkcd-container');
577         comicContainer.innerHTML = '<p>Failed to load XKCD comic.</p>';
578       });
579     }
580   }

```



## Weather API

- I have done integration of weather API using the Java script code in which it shows the current weather in your particular area
- I used a Java script fetch to get a current temperature, humidity and description of the current City.
- for creating the container I used

alam1.github.io / index.html

Code Blame 677 lines (608 loc) · 17.8 KB ⚡ Code 55% faster with GitHub Copilot

```

584     document.addEventListener("DOMContentLoaded", function () {
585         function getWeather() {
586             const apiKey = "bd5e378503939ddae76f12ad7a97608";
587             const city = "Dayton";
588             fetch(`https://api.openweathermap.org/data/2.5/weather?q=${city}&appid=${apiKey}&units=imperial`)
589                 .then(response => response.json())
590                 .then(data => {
591                     const { temp, humidity } = data.main;
592                     const weatherDesc = data.weather[0].description;
593                     document.getElementById('weather-info').innerHTML = `
594                         <p>City: ${data.name}</p>
595                         <p>Temperature: ${temp} °F</p>
596                         <p>Humidity: ${humidity}%</p>
597                         <p>Weather: ${weatherDesc}</p>
598                     `;
599                 })
600                 .catch(() => {
601                     document.getElementById('weather-info').innerText = 'Unable to fetch weather data';
602                 });
603             }
604             getWeather();
605         });
606     });

```

- The weather which shows in the website I accurate and It is in real time.

The screenshot shows a dark-themed web application interface. At the top, there is a navigation bar with three items: 'Hackathon 1 – Cross-site Scripting Attacks and Defenses', 'Hackathon 2 – SQL Injection Attacks', and 'Individual Project 1 – Front-end Web Development with a Professional Profile Website on GitHub.io'. Below this is a large button labeled 'Go to Course Page'.

In the center, there is a section titled 'Current Weather' containing the following information:

- City: Dayton
- Temperature: 87.55 °F
- Humidity: 60%
- Weather: clear sky

At the bottom of the page, there are two smaller sections: 'Digital Clock' (showing 'Current Time: 1:55:56 PM') and 'Random Joke' (with the question 'What does the MacBook have in common with Donald Trump?').

## Java script code

- I used a cookie alert in the code to show the first and last visit where the time I shown in the top of the website
- cookie remembers the last visit of the website and I used the html to write the cookie alert to show inside a tag

[lalam1.github.io / index.html](https://lalam1.github.io/index.html) ↑ Top

Code Blame 677 lines (608 loc) · 17.8 KB ⚡ Code 55% faster with GitHub Copilot

```

488 <h5>XKCD Comic</h5>
489 <div id="xkcd-container"></div>
490 </div>
491
492 <script>
493
494   function checkFirstVisit() {
495     const lastVisit = getCookie("lastVisit");
496     const now = new Date();
497     const currentDate = now.toLocaleString();
498     const alert = document.getElementById("cookieAlert");
499     alert.style.display = "block";
500     alert.innerText = lastVisit
501     ? `Welcome back! Your last visit was ${lastVisit}`
502     : "Welcome to my homepage for the first time!";
503     setCookie("lastVisit", currentDate, 30);
504   }
505
506   function setCookie(name, value, days) {
507     const expires = new Date(Date.now() + days * 864e5).toUTCString();
508     document.cookie = `${name}=${encodeURIComponent(value)}; expires=${expires}; path=/`;
509   }
510
511   function getCookie(name) {
512     return document.cookie.split('; ').reduce((r, c) => {
513       const [key, v] = c.split('=');
514       return key === name ? decodeURIComponent(v) : r;
515     }, '');
516   }
517   checkFirstVisit();
...

```

Lalam Lohit Naidu

Welcome back! Your last visit was 6/21/2025, 1:36:43 PM

[Home](#) [Resume](#) [Course](#) [Contact](#)

**Professional Profile**  
Lalam Lohit Naidu  
Master's Student @ University of Dayton | Ex-Analyst @ SSVCI

**Education**  
Bachelor of Computer Science – Gayatri Vidy Parishad, India (Graduated: 2022)  
Master of Computer Science – University of Dayton (Expected: April 2026)

**Skills**  
• Languages: Python, Java, C++, SQL, HTML  
• Tools: DBMS, GitHub, Power BI  
• Other: Problem Solving, Time Management, Leadership

**Experience**  
• Analyst, SSVCI (Oct 2022 – Aug 2024)  
Worked on Java, Spring Boot, SQL, Microservices, and front-end development. Delivered web application as part of internship deliverables.