

# WAPH-Web Application Programming and Hacking

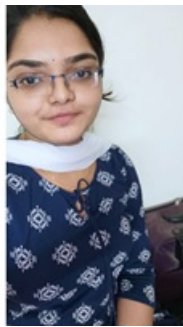
**Instructor: Dr. Phu Phung**

**Name:** Udayani Mandadi

**Email:** mandadui@mail.uc.edu

## **Short Bio :**

I am Udayani Mandadi. I have enrolled in the Web Application Programming and hacking course to enhance my technical skills in web development languages such as HTML5, JavaScript, CSS, PHP, and MySQL. And I am interested to learn about secure programming practices, web and web application development, client-side and server-side programming, Git version control, and DevOps procedures, to gain a well-rounded understanding of the field.



## **Individual Project 1**

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

### **Repository Information**

Repository's URL: <https://github.com/mandadui/mandadui.github.io>

### **Overview and Requirements**

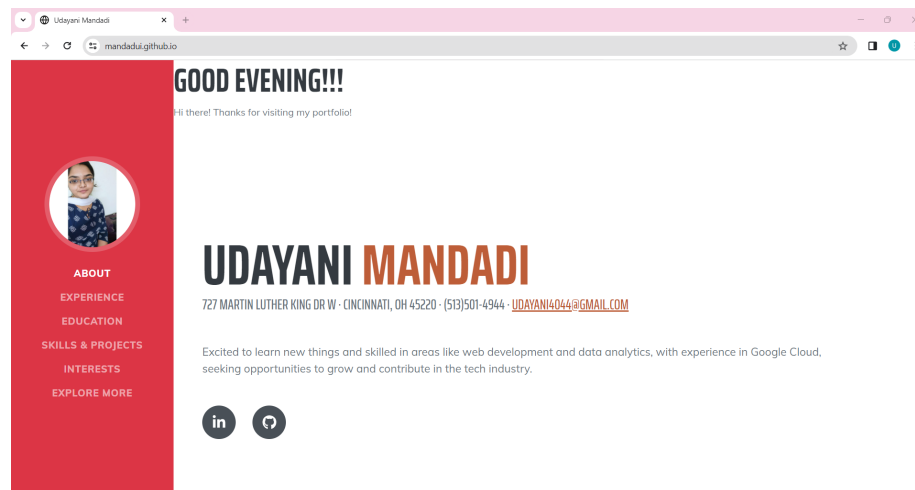
The project's main goal is to build a professional profile website on the GitHub cloud that features a new HTML page creating the "Web Application Programming and Hacking" course in addition to a detailed profile with key information. Using free and open-source CSS frameworks such as Bootstrap and customizing the profile for potential employers are standard needs. The technical parts include integrating two public Web APIs (jokeAPI and a graphics API), using jQuery and a framework/library, creating JavaScript features, and using JavaScript

cookies for personalized interactions. The goal is to effortlessly merge these aspects into a single HTML page while emphasizing advanced web development skills for potential job applications.

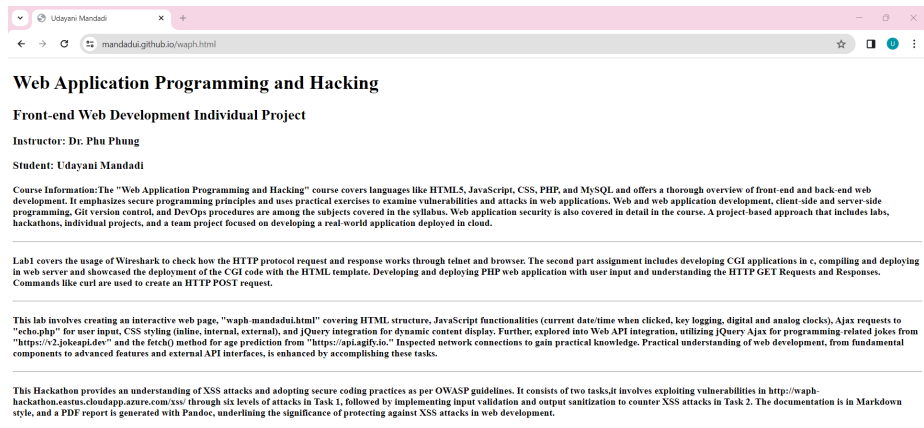
## General Requirements

I created and deployed a personal website on GitHub (github.io). The website included my name, photo, contact information, educational background, work experience, and skills. The goal was to create a comprehensive and visually appealing resume, enhancing my digital profile for potential employment prospects.

link to the page : <https://mandadui.github.io/>

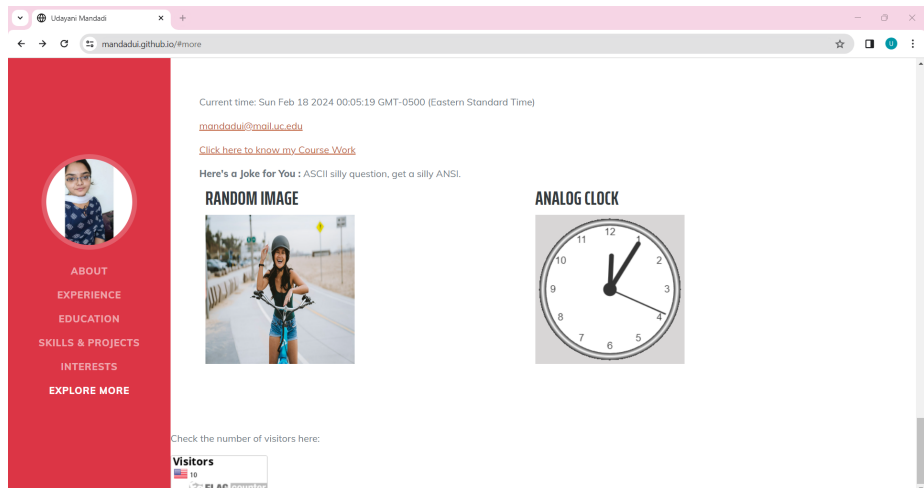


To provide viewers with a comprehensive summary of the completed coursework for the Web Application Programming and Hacking course, I have included a link in the “More” section that says, “Click here to know my coursework.” This link takes you to a page that explains all the work I’ve done in the Web Application Programming and Hacking course. On that page, you can see what topics we’ve covered, giving you a good idea of how the course is going and what I’ve completed so far. It’s an easy way for you to check out the different parts of the Web Application Programming and Hacking course.



## Non Technical Requirements

I used a Bootstrap-based portfolio design and enhanced its appearance with features such as navigation and a page tracker. I highlighted my professional experience to draw recruiters in, and I organized my portfolio into sections to make it easier for them to access. This streamlined approach ensures recruiters can quickly access and assess specific areas of interest in my resume.



I added a page tracker at the bottom of the webpage using Flag Counter Analytics. It keeps track of the number of visitors, their countries of origin, visit times, and devices used. The details are accessible through a provided link, similar to trackers such as <https://info.flagcounter.com/K8DY>



<p>Check the number of visitors here:</p>

<div><a href="https://info.flagcounter.com/K8DY"></div>

```

<script>
  function displayTime() {
    document.getElementById('digit-clock').innerHTML = " Current time : " +
  }
  setInterval(displayTime, 500);
  function validateInput(inputId) {
    var input = document.getElementById(inputId).value;
    if (input.length === 0) {
      alert("Please enter some text");
      return false;
    }
    return true;
  }
  function encodeInput(input) {
    const encodedData = document.createElement('div');
    encodedData.innerText = input;
    return encodedData.innerHTML;
  }
</script>
<div id="email" onclick="showhideEmail()">Show Udayani's mail</div>
<a href="/waph.html">Click here to know my Course Work</a>

// wishes.js
const Greeting = () => {
  const currentHour = new Date().getHours();
  let greetingMessage;
  if (currentHour >= 5 && currentHour < 12) {
    greetingMessage = 'Good Morning!!!';
  } else if (currentHour >= 12 && currentHour < 17) {
    greetingMessage = 'Good Afternoon!!!';
  } else {
    greetingMessage = 'Good Evening!!!';
  }
  alert(greetingMessage); // Alert moved outside of the conditional statement
  return React.createElement('h2', null, greetingMessage);
};
ReactDOM.render(React.createElement(Greeting, null),
  document.getElementById('greeting-root'));

```

## Two Public APIs integration

In the index.html page, I've incorporated two joke APIs. The first one, <https://v2.jokeapi.dev/joke/Any>, displays a random joke on the page and automatically refreshes every minute. Additionally, I've used the 'https://source.unsplash.com/random' API to showcase random graphic images

on the webpage

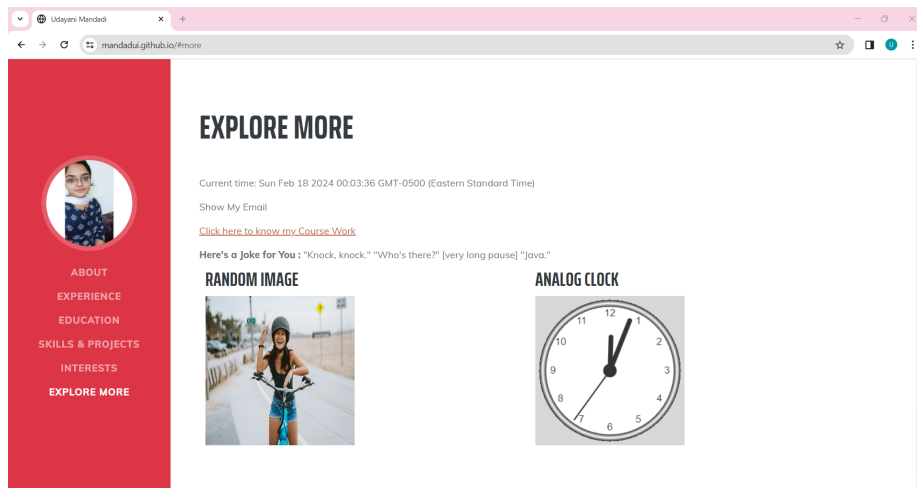
Code for Joke api :

```
<script>

    function fetchJoke() {
        fetch('https://v2.jokeapi.dev/joke/Programming?type=single')
            .then(response => response.json())
            .then(data => {
                document.getElementById('joke').innerHTML = "<b>Here's a Joke for You :</b>";
            })
            .catch(error => {
                console.error('Error fetching joke:', error);
            });
    }
    fetchJoke();
    setInterval(fetchJoke, 60000);
</script>
```

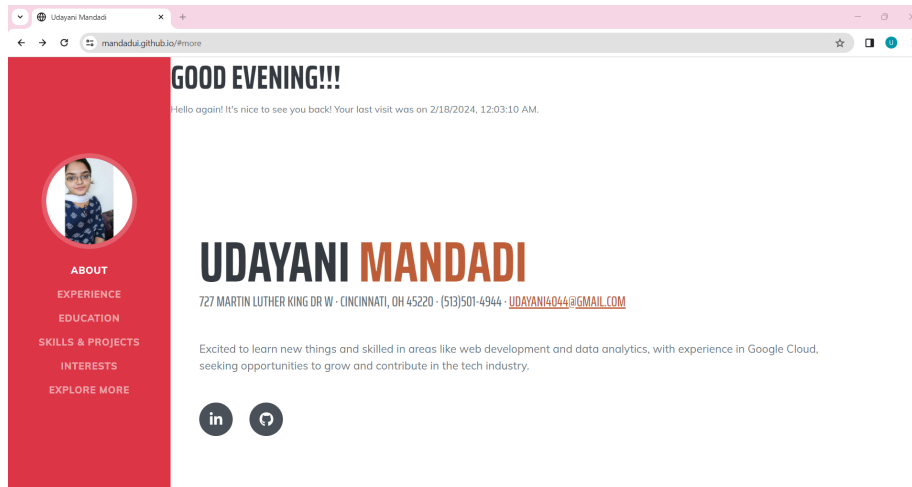
Code for Random Image:

```
fetch("https://source.unsplash.com/random")
    .then((response) => {
        if (response.redirected) {
            const redirectedUrl = response.url;
            document.getElementById("random-image").src = redirectedUrl;
        } else {
            throw new Error("Unexpected response from the server");
        }
    })
    .catch((error) => console.error("Error fetching data:", error));
```



## Integration of cookies

I added “getCookie()” and “setCookie()” functions to personalize messages on my portfolio website. Initially welcoming users with “Hi there! Thanks for visiting my portfolio!”. Upon returning, the website displays a message indicating the user’s last visit along with the date



```
function welcomeUser() {  
  const lastVisitDate = getCookie("lastVisit");  
  const greetingElement = document.getElementById("greetingCookie");  
  if (lastVisitDate) {  
    greetingElement.innerHTML =  
      "Hello again! It's nice to see you back! Your last visit was on " +  
      lastVisitDate +  
      ".";  
  } else {  
    greetingElement.innerHTML = "Hi there! Thanks for visiting my portfolio!";  
  }  
  const currentDate = new Date();  
  setCookie("lastVisit", currentDate.toLocaleString(), 365);  
}  
welcomeUser();
```

## Submission

I used Pandoc to generate a PDF report from the README.md file for lab 1, ensuring that all content was accurately rendered. The resulting PDF file is named ‘mandadui-waph-project1.pdf’.