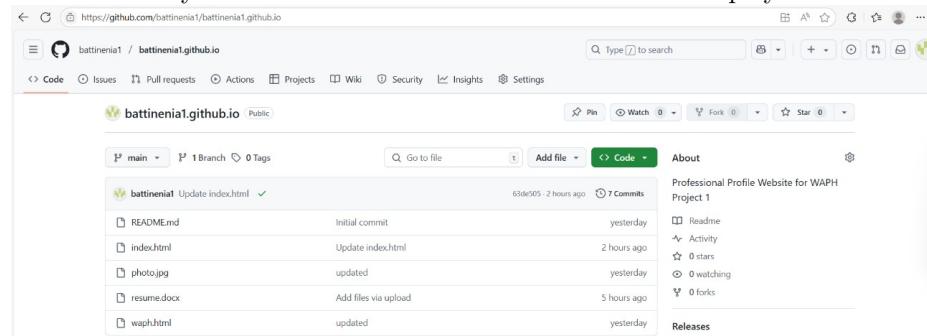


WAPH-Web Application Programming and Hacking

Individual Project 1 Front-end Web Development with a Professional Profile Website on Bitbucket.io Cloud service Course Number : CPS592 Course Name : WAPH(Web Application Programming And Hacking) ## Instructor/Professor : Dr. Phu Phung **Presented By:** Battineni Anvitha **Email :** battinenia1@udayton.edu Website url: file:///home/anvitha/battinenia1.github.io/index.html file:///home/anvitha/battinenia1.github.io/waph.html

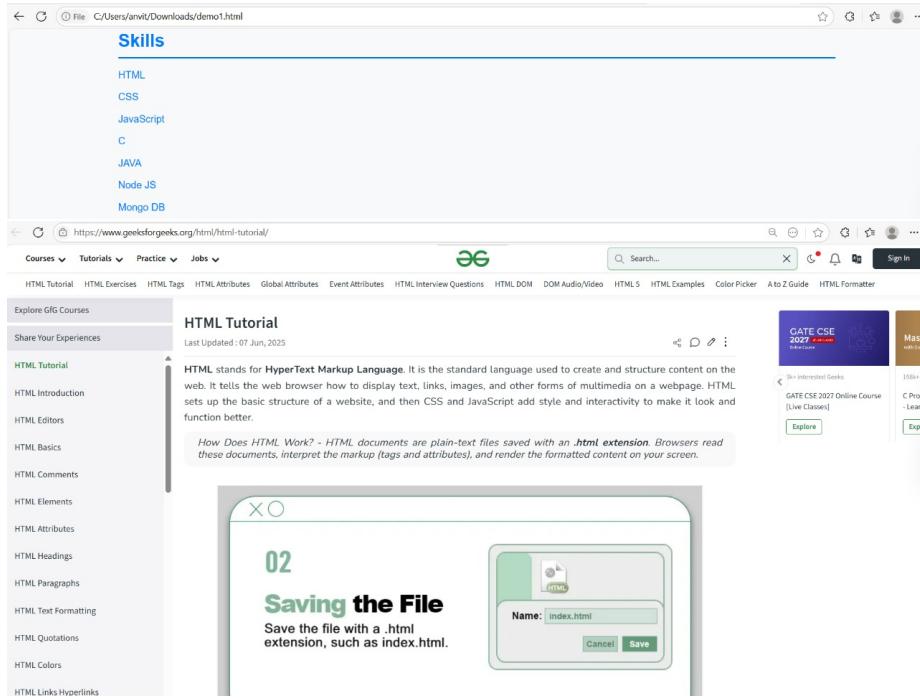
Repository url: **Part I :** General Requirements **Task I:** Using my headshot, I made a professional website that included my resume, academic accomplishments, professional experiences, and personal information. Additionally, I have contributed my skill set and used Bitbucket.io Cloud to deploy the website.



I made a second HTML file called waph and connected it to the Index.html file so that they are interconnected so that the resume can display my background. In order to inform visitors about the skill they are viewing, I have also included hyperlinks to the skills tab. Clicking on the skill will direct them to the skill's dedicated website or educational resource.



I've also included links to the skills I've acquired over the years. They will be taken to either an educational website to learn more about the skill or a website where they can download it after clicking on it. I'm clicking on the HTML now, which will take me to the geeksforgeeks page about HTML.



Task II: Provide a link to a fresh HTML page that introduces the course “Web Application Programming and Hacking” and its associated practical exercises. Additionally, I made another HTML page called Web-Application-Hacking.html and included the course information there. The webpage contains the following information: the instructor, the course results, tentative topics, and the goals and objectives of the course. I’ve also included links to the course page on Canvas, the instructor’s email addresses, and the comprehensive class schedule. Additionally, I have included a link to the course page at the bottom of the page. By clicking on the link, users and visitors can view comprehensive course information: <https://bit.ly/waph-s25>

The screenshot shows a web browser window with the URL `file:///home/anvitha/battinenia1.github.io/waph.html#`. The page title is "Battineni Anvitha". The main content area has a light blue header bar with the title "Web Application Programming & Hacking". Below the header, there is a section titled "General Information:" which includes:

- Class Days/Time:** TR 3:35-4:50 PM
- Classroom:** JHH 180. [Detailed class schedule here](#)
- Prerequisites:** CPS 341 and CPS 350/CPS 501

Below this, there is an "Instructor:" section with:

- Dr. Phu Phung**
- Office: Hathcock Hall 105
- Email: Phu@udayton.edu

There is also a "Course Webpage:" section with a note that all course materials can be found on the course page located on the Canvas server: [Canvas Course Page](#).

Course Description: In this course, students will study basic web application development with front-end (HTML5, JavaScript, CSS) and back-end (PHP/MySQL). Web application vulnerabilities and attacks will be introduced and explored with hands-on exercises on the range. Secure programming principles and practices will be introduced to avoid potential web application vulnerabilities and attacks.

Non-Technical Requirements Task I: For the HTML pages, I used the open-source CSS bootstrap template, which can be found at "<https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css>". I have included the bootstrap in the head tag before the body tag, which will help me get the respective bootstrap functionalities to design the website. Usually, this is inserted in the

block of the html code to gain all the bootstrap css frameworks in the webpage. Using this framework, the page can instantly get the necessary css styles, components to design interactive and responsive websites. Additionally, the flag counter features geographic tracking to display the nation from which the website was accessed. The image displays the flag counter that is incorporated into the index.html page. The flag counter indicates that index.html has been viewed 43 times, with 3 users from the USA and three from India. This code is utilized to incorporate the code into the website. Additionally, if ad-blockers are enabled on the website, the flag counter occasionally vanishes and performs poorly.

The screenshot shows a web browser window with the URL `file:///home/anvitha/battinenia1.github.io/index.html`. The page displays an analog clock with hands pointing to 10:10, a yellow circular sun-like icon below it, and the text "Temperature: 31.7°C, Clear sky". A "Visitors" section shows 3 Indian and 3 American flags with 43 pageviews. A "FLAG counter" button is present. The Sublime Text editor window below shows the corresponding HTML code for the page.

```

253     </div>
254     <div class="flagcounter mt-3">
255       <a href="https://flagcounter.com/" target="
256         blank" title="Flag Counter">
257           
262     </a>
263   </div>

```

Technical Requirements Task-1 Basic Java Script Code JQuery and JavaScript will be used in this task to implement a number of features, including an analog and digital clock, the ability to show or hide your email, and a current date/time. Initially, we will use JavaScript to design both the digital and analog clocks. I used HTML and JavaScript code to design the digital and analog clocks.

Current Date and Time: Saturday, June 21, 2025 at 11:32:25 PM GMT+5:30

Digital Clock

11:36:11 PM

Analog Clock



← → ⌛

file:///home/anvitha/battinenia1.github.io/index.html



Professional Profile

A prospective student of computer science who has worked as a package specialist in SAP and C languages in addition to SAP modules. Looking to become a Full Stack Developer in a dema

Contact Information

✉️ battinenia1@udayton.edu

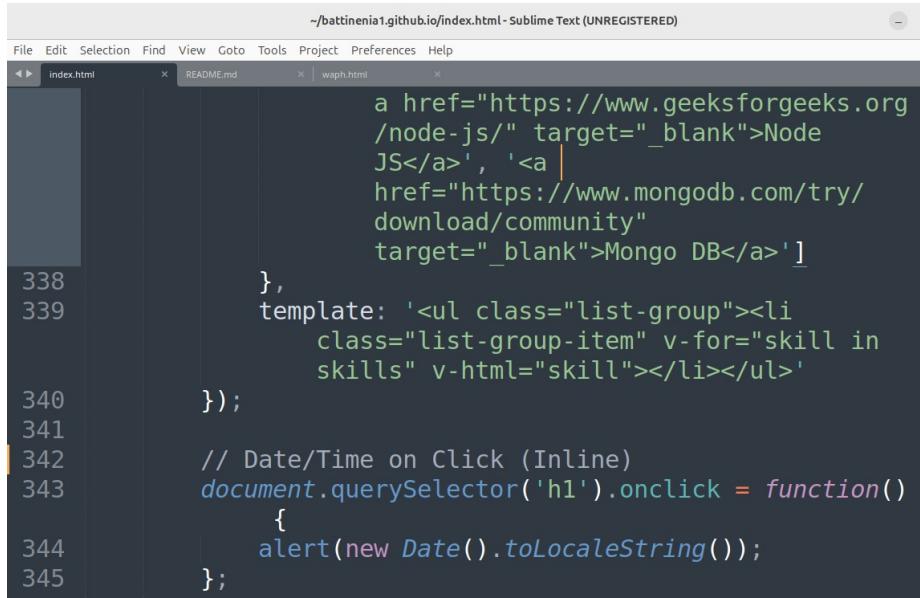
```
~/battinenia1.github.io/index.html - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
index.html README.md waph.html

        '✉ battinenia1@udayton.edu' : 'Click to
        show' );
289     }
290
291     // Digital Clock
292     function updateDigitalClock() {
293         const now = new Date();
294         $('#digitalClock').text(now.
        toLocaleTimeString());
295     }
296     $(document).ready(function() {
297         updateDigitalClock();
298         setInterval(updateDigitalClock, 1000);
299     });
300

~/battinenia1.github.io/index.html - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
index.html README.md waph.html

302     function updateAnalogClock() {
303         const now = new Date();
304         const hours = now.getHours() % 12;
305         const minutes = now.getMinutes();
306         const seconds = now.getSeconds();
307         const hourDeg = (hours * 30) + (minutes * 0.5
            );
308         const minuteDeg = (minutes * 6);
309         const secondDeg = (seconds * 6);
310         $('#hourHand').css({ transform: `rotate(${{
            hourDeg}deg)` });
311         $('#minuteHand').css({ transform: `rotate(${{
            minuteDeg}deg)` });
312         $('#secondHand').css({ transform: `rotate(${{
            secondDeg}deg)` });
313         positionClockNumbers();
314     }


```



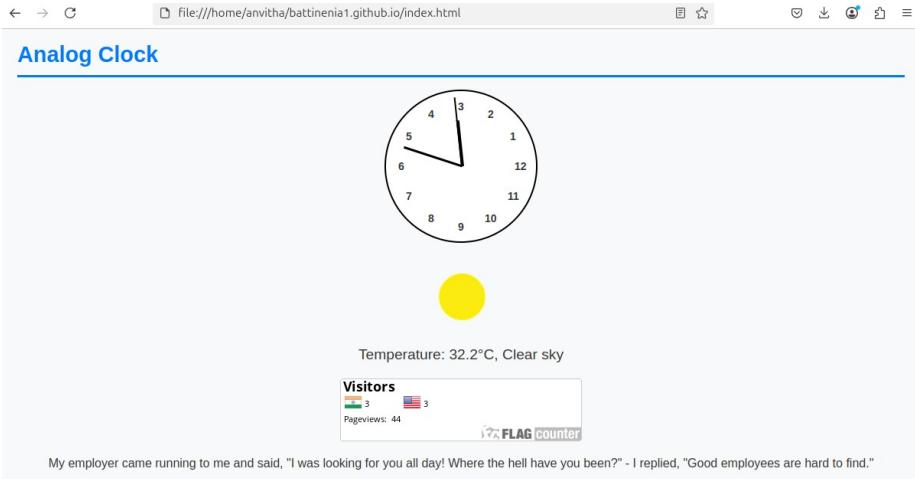
A screenshot of the Sublime Text editor interface. The title bar reads: ~/battinenia1.github.io/index.html - Sublime Text (UNREGISTERED). The menu bar includes File, Edit, Selection, Find, View, Goto, Tools, Project, Preferences, and Help. There are three tabs open: index.html, README.md, and waph.html. The index.html tab contains the following code:

```
a href="https://www.geeksforgeeks.org/node-js/" target="_blank">Node JS</a>', '<a href="https://www.mongodb.com/try/download/community" target="_blank">Mongo DB</a>']

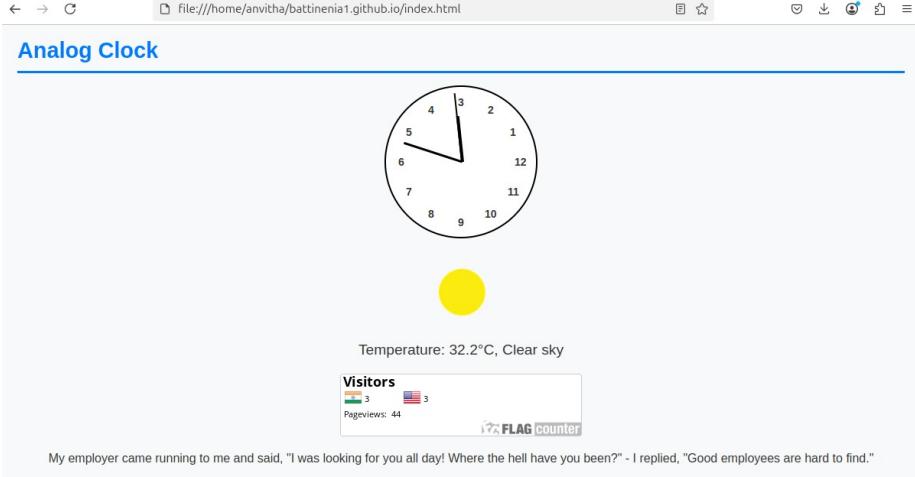
338
339
340
341
342
343
344
345
```

Public Web API Integration I have used a container with

tags and Java script to integrate the joke api, which delivers a new joke every minute, into my website through a link. Using the div tag from the HTML entity, I first made a container. That is displayed below. Additionally, I have included a disclaimer that we are not liable for the jokes or content that is generated. Using the fetch keyword, I have written the Java script code to retrieve the joke from the corresponding user interface. Additionally, I have established a time interval that calls for updates every minute. This is the code. This is how the joke api appears on the website and also disclaimer is included in the second joke picture.



Every minute, a new joke will be presented to us. The jokes in the last two images are different, indicating that they are updated every minute. Public API to display weather- Using JavaScript and the openweathermap api key, the weather api integration was completed, displaying the local weather. To make it simple for users to view the data, I made a container to store the weather. I used the div class="container mt-5 fadeIn" id="weather" to create the container. After that, I wrote javascript code to retrieve the data from the Open Weather API key to ensure that the data was correct.



```

417     async function fetchWeather() {
418         const weatherbitApiKey = 'dd51a99ec1924d4cb8f700421f201f88'; // Your Weatherbit key
419         const city = 'Dayton,OH';
420         const url = `https://api.weatherbit.io/v2.0/current?city=${city}&key=${weatherbitApiKey}`;
421         const maxRetries = 3;
422         for (let attempt = 1; attempt <= maxRetries; attempt++) {
423             try {
424                 const response = await fetch(url);
425                 if (!response.ok) throw new Error(`HTTP error! status: ${response.status}`);
426                 const data = await response.json();

```



```

249         <div id="weather" class="text-center mt-3">
250             <img id="weatherIcon" src="" alt="Weather Icon">
251             <p id="weatherText"></p>
252             <p id="weatherError" class="text-danger" style="display: none;">Failed to load weather data.</p>
253         </div>

```

JavaScript cookies to keep the client in mind: I used a cookie alert in the CSS styling to display the user's first or last visit time as an alert at the top of the website. It uses cookies to remember the user's last visit or if it is the first visit. I wrote the cookie alert in HTML to display inside a `<div>` tag like this: `div class="cookie-alert" id="cookieAlert"` Below is the javascript code that makes the cookie function.

```
function updateWelcomeMessage() {
    let visitCount = parseInt(getCookie('visitCount')) || 0;
    const lastVisit = getCookie('lastVisit');
    visitCount += 1;
    setCookie('visitCount', visitCount, 365);
    if (visitCount === 1) {
        $('#welcomeMessage').html('Welcome to my
            homepage for the first time! <br>
            Current Date and Time: ' +
            getCurrentDateTime());
    } else {
        $('#welcomeMessage').html(`Welcome back!
            Your last visit was ${new Date(
            lastVisit).toLocaleString()} <br>
            Current Date and Time: ` +
            getCurrentDateTime());
    }
}
```

NPTEL issued "Programming Solving Through Programming in C"

Web Application Programming and Hacking Course

Welcome to my homepage for the first time!

Certifications

AWS Certified Cloud Practitioner

UDEMY issued "Learn JAVA Programming - Beginner to Master"

NPTEL issued "Programming Solving Through Programming in C"

Web Application Programming and Hacking Course

Welcome back! Your last visit was 6/21/2025, 11:48:55 PM

Current Date and Time: Sunday, June 22, 2025 at 12:02:02 AM GMT+5:30