

WAPH-Web Application Programming and Hack-ing

Instructor: Dr. Phu Phung

Name: Shivani Eli

Email: elisi@mail.uc.edu

Short-bio: This is Shivani Eli currently pursuing Master's in Information Technology at UC. An IT graduate with a passion for problem solving, and learning new technology to drive innovation.



Figure 1: Shivani Eli

Repository Information

Respository's URL: <https://github.com/ShivaniEli/ShivaniEli.github.io/blob/main/Project1/README.md>

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

Overview and Requirements

In this individual project 1, I have created a professional portfolio webpage. In this webpage, it has my details like, education, work experience, project, contact details, my email. Along with my downloadable resume,headshot. I have added few features like, 2D analog clock, digital clock, show/hide email, along with integration of web apis like Joke, weather api. As special feature I have integrated a stock data web api which is developed in a new framework React JS, remaining features were added using HTML,CSS,JavaScript, JQuery. The main focus of this project is development of the front-end part with api integration and cookies. I have added a page tracker. I have added linked link, with linked in icon. This all were included in a new public repository.

The link to access my website is: <https://shivanieli.github.io/Project1/index.html>.

The link to access Individual Project-1 is: <https://github.com/ShivaniEli/ShivaniEli.github.io>.

General Requirements

Personal Website on Github.io

For this website, I have created a new public repository ShivaniEli.github.io, where my webpage is available.

The link to access my website is: <https://shivanieli.github.io/Project1/index.html>.

“Web Application Programming and Hacking” course and related hands-on projects on waph.html file

Another html file ‘waph.html’ is created with the web application programming and hacking course details. I created the separate page with the name waph.html on my repository introducing the “Web Application Programming and Hacking” course and its related hands-on projects. In this I includes overviews of Lab0, Lab1, Lab2, Hackathon 1 and Individual Project 1.

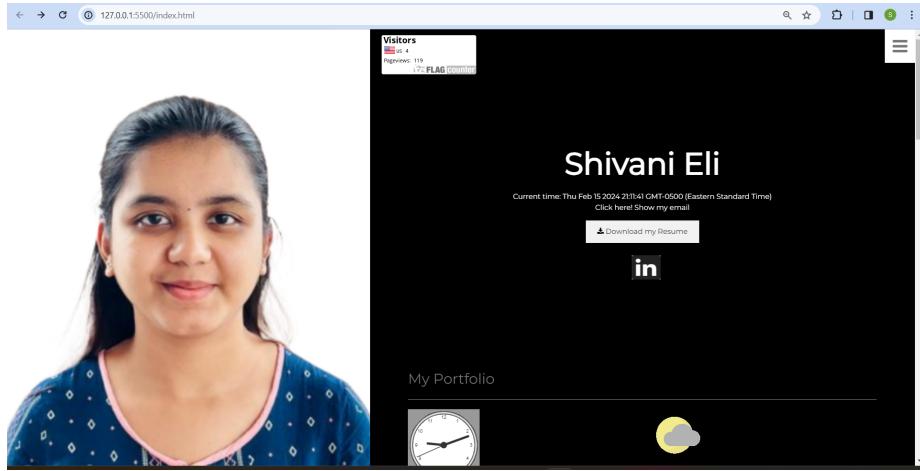


Figure 2: Portfolio Website

The link to access index.html is: <https://shivanieli.github.io/Project1/index.html>.

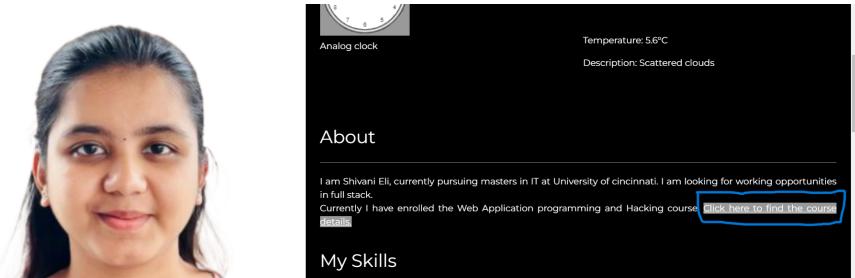


Figure 3: waph.html page link on website

Non-technical requirements

Bootstrap Template

I have used a bootstrap template free open source template for my portfolio. Then I have made modifications, accordingly, as per the requirements.

Page Tracker

I have included a flag counter, page tracker on the top, for displaying no. of visitors, with website <https://flagcounter.com/>, with the key included within the code.

Code for Flag Counter tracker:

Course Webpage

All course materials, such as the syllabus, major assignments, and handouts, can be found on the course page located on Canvas <https://sc.instructure.com/courses/1683040> (ECE3895-001, cross-listed for all other sections).

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.

Topics:

- ❖ The Web and Web Application
- ❖ Client-side Web Development with HTML5, CSS, and JavaScript
- ❖ Server-side Web Application Development with PHP and Database (MySQL)
- ❖ Web Application Security
- ❖ Web Application Security: Vulnerabilities and Attacks with Hands-on Exercises
- ❖ Secure Web Programming: Principles and Practices

Hands-on Projects:

In this we done 1 labs, 1 hackathon, and 1 Individual project. In summary, In the labs had worked hands-on cloning and creating git hub repository
In lab 1 we created a basic html page, with analog clock, written in jquery and digital clock. With joke api integration, covered HTML, JS, JQuery, JSON, did some backend with echo.php file for GET POST requests and with AJAX. In Hackathon did hands-on Level 06 XSS attacks and gave input validation for defense of the lab 1 webpage.
In this individual project I created a professional profile webpage with some features from lab1 and added 2 webapis with jquery, js, another web app with react js. Used bootstrap for template.

Labs:

- ❖ Lab_0
- ❖ Lab_1
- ❖ Lab_2

Hackathon 1:

- ❖ Hackathon_1

Individual Project 1:

- ❖ Project_1

Figure 4: waph.html page

```
<div style="text-align:left;">
  <a href="https://info.flagcounter.com/szVl"></a>
</div>
```

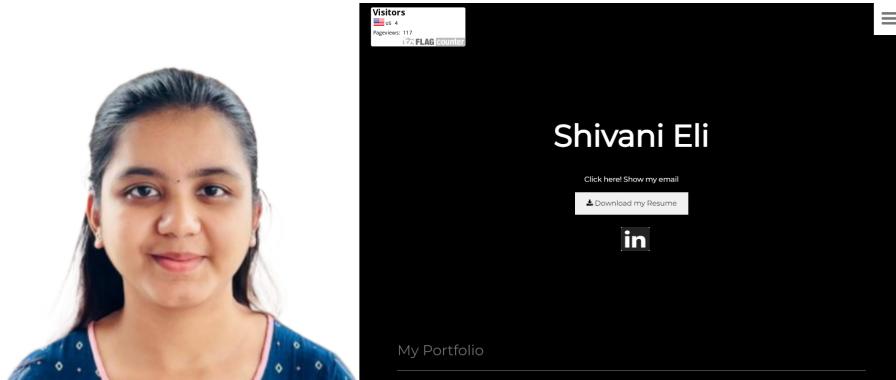


Figure 5: Flag Counter

Technical requirements

A digital clock; An analog clock; show/hide your email:

From lab2, added the analog clock, digital clock and show/hide email in the website.

Source Code for digital clock:

```
function displayTime() {
    document.getElementById('digital-clock').innerHTML = "Current time: "+new Date();
}
setInterval(displayTime, 500);
function validateInput(inputId){
    var input=document.getElementById(inputId).value;
    if(input.length==0){
        alert("please enter text");
        return false;
    }
    return true;
}
function encodeInput(input){
    const encodeData=document.createElement('div');
    encodeData.innerText=input;
    return encodeData.innerHTML;
}
//end digital-clock
```

Source Code for 2D Analog clock:

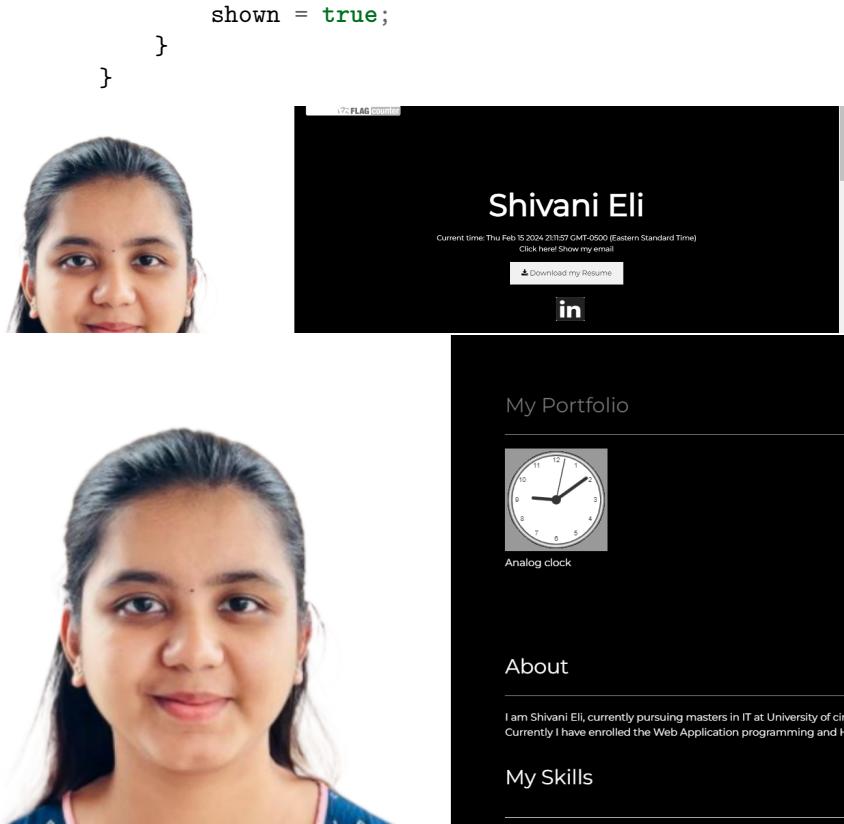
```
var canvas = document.getElementById("analog-clock");
var ctx = canvas.getContext("2d");
var radius = canvas.height / 2;
ctx.translate(radius, radius);
radius = radius * 0.90;
setInterval(drawClock, 1000);

function drawClock() {
    drawFace(ctx, radius);
    drawNumbers(ctx, radius);
    drawTime(ctx, radius);
}
```

Source Code for show/hide your email:

```
var shown = false;

function showhideEmail() {
    if (shown) {
        document.getElementById('email').innerHTML = "Show my email.";
        shown = false;
    }
    else {
        var mye = "<a href='mailto:elisi" + "@" + "mail.uc.edu'>elisi" + "@" +
        document.getElementById('email').innerHTML = mye;
```



One more Functionality of my choice

For a new functionality I have added another web api, stock web api with REACT JS framework. And added a linkedin link with icon. And added a button by clicking it will download my resume, with JS code. For integrating REACTJS I have installed the react-dom, jsx libraries by CND.

Source code for Stock code api:

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/react/17.0.2/umd/react.development.js">
<script src="https://cdnjs.cloudflare.com/ajax/libs/react-dom/17.0.2/umd/react-dom.development.js">
<script src="https://cdnjs.cloudflare.com/ajax/libs/babel-standalone/6.26.0/babel.min.js"></script>

function StockData() {
    const [stockData, setStockData] = React.useState(null);
    const [loading, setLoading] = React.useState(true);

    React.useEffect(() => {
        fetch("https://api.polygon.io/v2/aggs/ticker/AAPL/range/1/day/2024-02-15?sort=desc&limit=1000")
            .then(response => response.json())
            .then(data => setStockData(data))
            .catch(error => console.error(error));
    }, []);
}

const App = () => {
    return (
        <div>
            <h1>Stock Data</h1>
            {loading ? <p>Loading...</p> : <pre>{JSON.stringify(stockData, null, 2)}</pre>}
        </div>
    );
}

export default App;
```

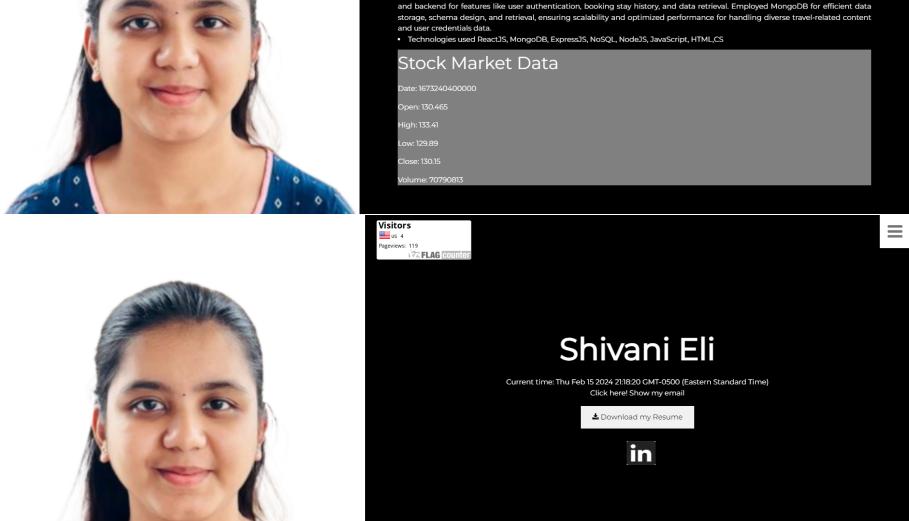
```

        .then(data => {
          setStockData(data);
          setLoading(false);
        })
        .catch(error => console.log("Error fetching stock data:", error));
      }, []);
    }

    return (
      <div>
        <h1>Stock Market Data</h1>
        {loading ? (
          <p>Loading...</p>
        ) : (
          <div>
            <p>Date: {stockData.results[0].t}</p>
            <p>Open: {stockData.results[0].o}</p>
            <p>High: {stockData.results[0].h}</p>
            <p>Low: {stockData.results[0].l}</p>
            <p>Close: {stockData.results[0].c}</p>
            <p>Volume: {stockData.results[0].v}</p>
          </div>
        )}
      </div>
    );
  }
}

ReactDOM.render(<StockData />, document.getElementById("stock"));

```



Joke API

From lab2 included the joke api and modified such that new joke is displayed every one minute without reloading the webpage. Source code for Joke API:

```
function getJoke(){
    $.get("https://v2.jokeapi.dev/joke/Programming?type=single", function (result) {
        if (result && result.joke) {
            var encodeJoke = encodeInput(result.joke);
            $("#response").html("Programming joke of the day: " + encodeJoke);
        }
        else {
            $("#response").text("couldnot retrieve a joke at this time. ");
        }
    });
}

getJoke();
setInterval(getJoke, 60000);
```

Weather API

Included a weatherbit API to get current weather of Cincinnati with graphics.

```
$(document).ready(function () {
    var apiKey = '74516cb77ec742629b8fa2c6bead25ad';

    $.get("https://api.weatherbit.io/v2.0/current", {
        key: apiKey,
        city: 'Cincinnati'
    })
    .done(function (data) {

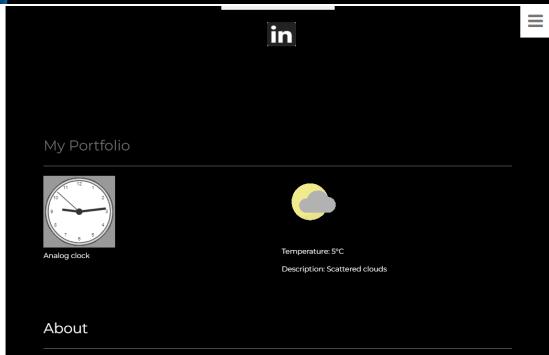
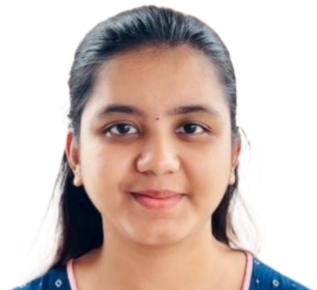
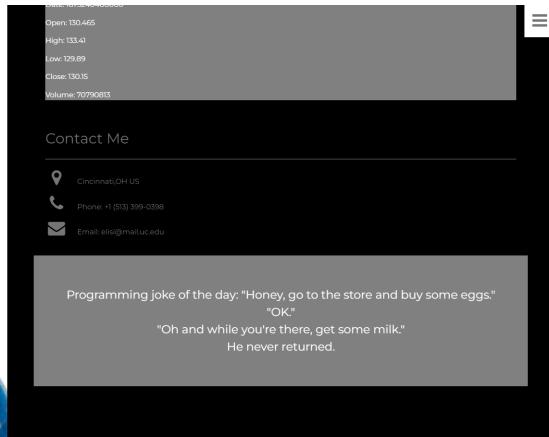
        if (data && data.data && data.data.length > 0) {
            var weatherData = data.data[0];
            var temperature = weatherData.temp;
            var description = weatherData.weather.description;
            var iconCode = weatherData.weather.icon;

            var iconUrl = "https://www.weatherbit.io/static/img/icons/" + iconCode +
                ".png";
            // Display
            $("#weather-icon").attr("src", iconUrl);
        }
    });
});
```

```

        $($("#weather-details").html("<p>Temperature: " + temperature + "°C</p>" +
            "<p>Description: " + description + "</p>"));
    } else {
        $($("#weather-info")).html("<p>Unable to fetch weather data.</p>");
    }
}
.fail(function () {
    $($("#weather-info")).html("<p>Error occurred while fetching weather data.</p>");
});
);

```



Javascript Cookies

Then deployed the cookies javascript code for displaying “Welcome back!” alert message along with last visited date and time if visited the website more than once else alert with “welcome to my home page” message.

```

function setCookie(name, value, days) {
    var expires = "";
    if (days) {
        var date = new Date();

```

```

        date.setTime(date.getTime() + (days * 24 * 60 * 60 * 1000));
        expires = "; expires=" + date.toUTCString();
    }
    document.cookie = name + "=" + (value || "") + expires + "; path=/";
}

function getCookie(name) {
    var nameEQ = name + "=";
    var ca = document.cookie.split(';');
    for (var i = 0; i < ca.length; i++) {
        var c = ca[i];
        while (c.charAt(0) === ' ') c = c.substring(1, c.length);
        if (c.indexOf(nameEQ) === 0) return c.substring(nameEQ.length, c.length);
    }
    return null;
}

function displayWelcomeMessage() {
    var lastVisit = getCookie("lastVisit");
    if (!lastVisit) {
        // First
        setCookie("lastVisit", new Date().toISOString(), 30);
        alert("Welcome to my homepage!");
    } else {
        // Revisit
        var lastVisitDate = new Date(lastVisit);
        alert("Welcome back! Your last visit was " + lastVisitDate.toLocaleString());
    }
}

// function call
window.onload = displayWelcomeMessage;

```

