WAPH-Web Application Programming and Hacking

Instructor: Dr. Phu Phung

Student

Name: Ruthvik Suvarnakanti

Email: suvarnrk@mail.uc.edu

Figure 1: Ruthvik Suvarnakanti

Lab 2 - Front End Web Development

Overview: In this hands-on web development lab, we begin by constructing a simple HTML page, tossing in essential tags and forms to lay the groundwork. Then, we delve into JavaScript, tackling it from various angles: embedding it directly, using the script tag, linking to an external file, and even pulling code from a remote repository. To tackle up the look of our webpage, we play around with CSS – going for inline, internal, and external styles to make things visually appealing. Now, here comes jQuery, our trusty sidekick. It helps us pull off some slick AJAX calls, reaching out to the echo.php file that is needed to be reused from lab 1 for some asynchronous action. But we're not stopping there. We bring in two web services using jQuery and the fetch method – one for serving up random jokes and the other for taking a crack at guessing ages. And to put the finishing touches on our project, we enlist Pandoc to effortlessly transform our README.md into a published PDF file.

https://github.com/suvarnrk/waph-suvarnrk/blob/main/README.md

Part 1 : Basic HTML with forms, and JavaScript

Task 1. HTML

As part of this task, we developed a basic HTML webpage named "waph-nakkantm.html." The webpage incorporates essential tags like

,

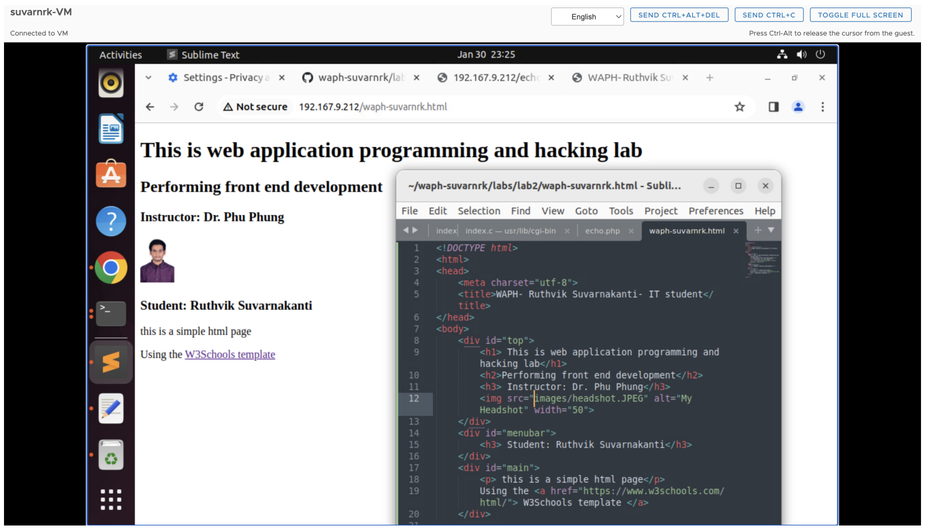
,

, , , and

. These tags structure the content, create headings, hyperlinks, and images, making the webpage interactive and visually appealing.

Included file `waph-suvarnrk.html`:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>WAPH- Ruthvik Suvarnakanti</title>
</head>
<body>
<div >
    <div id="top">
        <h1>Web Application Programming and Hacking</h1>
        <h2>Front End Development Lab </h2>
        <h3>Instructor : Dr Phu Phung</h3>
    </div>
    <div >
        <div id="menubar">
        <h3>Student : Ruthvik Suvarnakanti</h3>
        <img src="images/headshot.jpg" alt="Ruthvik Headshot image" width="50">
        </div>
        <div id="main">
            <p>A Simple HTML Page</p>
            Using the <a href="https://www.w3schools.com/html">W3 Schools Template</a>
            <hr>
            <b>Interaction with forms</b>
        <div>
            <i> Form with an HTTP GET request</i>
            <form action="/echo.php" method="GET">
                Your Input: <input name="input">
                <input type="submit" value="Submit">
            </form>

        </div>
        <div>
            <i> Form with an HTTP POST request</i>
            <form action="/echo.php" method="GET" name="echo_post">
                Your Input: <input name="input" onkeypress="console.log('You pressed a key'
                <input type="submit" value="Submit">

            </form>
            </div>
        </div>
    </div>
</div>
</body>
</html>
```



This is web application programming and hacking lab

Performing front end development

Instructor: Dr. Phu Phung

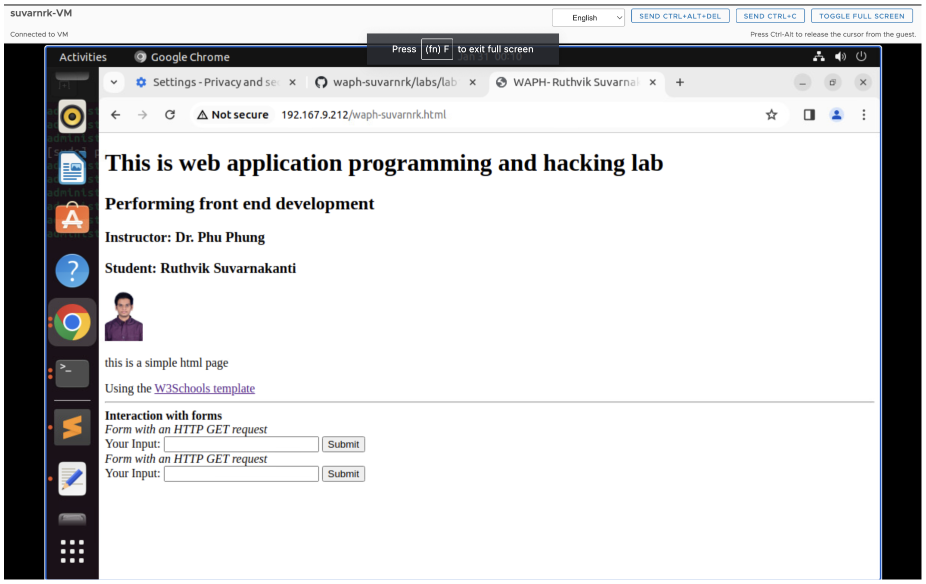
Student: Ruthvik Suvarnakanti

this is a simple html page

Using the W3Schools template

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>WAPH- Ruthvik Suvarnakanti- IT student</title>
</head>
<body>
    <div id="top">
        <h1> This is web application programming and hacking lab</h1>
        <h2>Performing front end development</h2>
        <h3> Instructor: Dr. Phu Phung</h3>
        <img src="images/headshot.JPEG" alt="My Headshot" width="50">
    </div>
    <div id="menubar">
        <h3> Student: Ruthvik Suvarnakanti</h3>
    </div>
    <div id="main">
        <p> this is a simple html page</p>
        Using the <a href="https://www.w3schools.com/html/"> W3Schools template </a>
    </div>
```

Figure 2: A simple HTML Page



This is web application programming and hacking lab

Performing front end development

Instructor: Dr. Phu Phung

Student: Ruthvik Suvarnakanti

this is a simple html page

Using the W3Schools template

Interaction with forms
Form with an HTTP GET request
Your Input: Submit
Form with an HTTP GET request
Your Input: Submit

Figure 3: A simple HTML Page

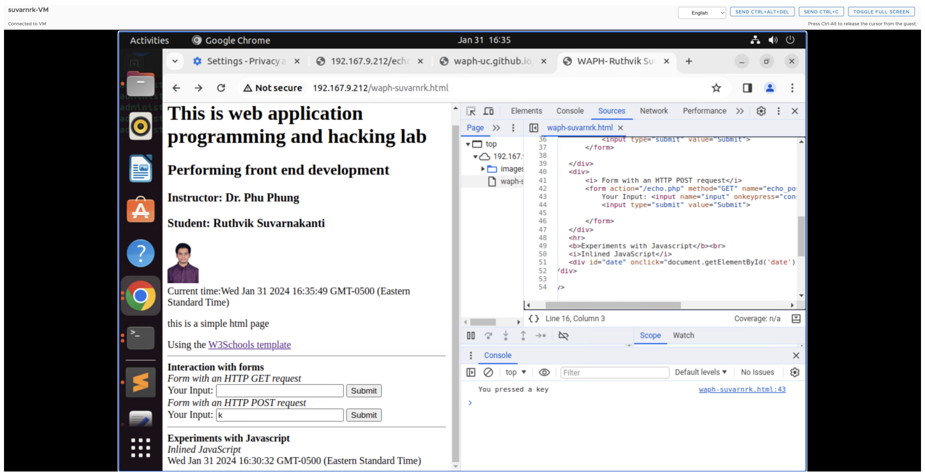
Task 2. Simple JavaScript

In this task, we got a simple introduction to JavaScript syntax and explored various methods of adding JavaScript code into an HTML file. We played around with Inline JS, where we wrote code to show the current date and time when clicked. Additionally, we logged the click event on the console for a bit of behind-the-scenes action.

```
     <div>
        <hr>
        <b>Experiments with Javascript</b><br>
        <i>Inlined JavaScript</i>
        <div id="date" onclick="document.getElementById('date').innerHTML= Date()">
    </div>
```

-JavaScript code in a

~~~
-JS code in JS file and and code in HTML page to show or hide email when clicked.
```JavaScript
    var shown=false;
function showhideEmail(){
    if(shown){
        document.getElementById('email').innerHTML="Show ny email";
        shown =false;
    }else{
~~~



This is web application programming and hacking lab

Performing front end development

Instructor: Dr. Phu Phung

Student: Ruthvik Suvarnakanti

Current time:Wed Jan 31 2024 16:35:49 GMT-0500 (Eastern Standard Time)

this is a simple html page

Using the W3Schools template

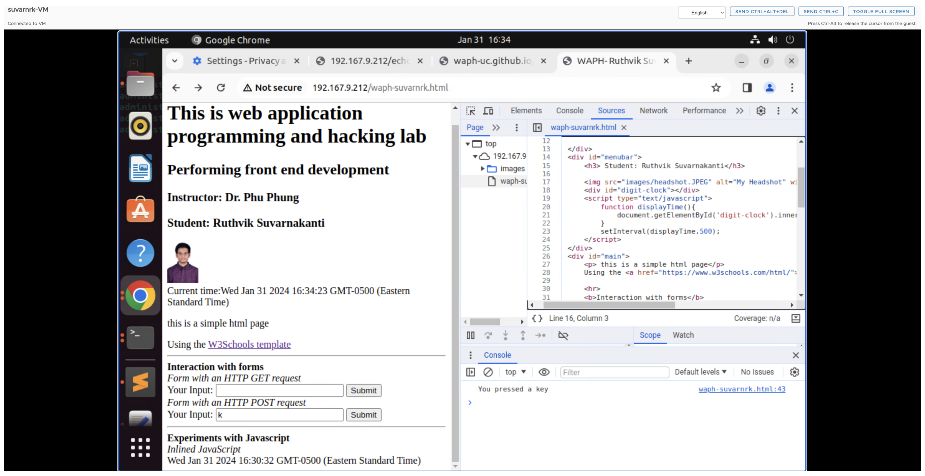
Interaction with forms
Form with an HTTP GET request
Your Input: Submit
Form with an HTTP POST request
Your Input: k Submit

Experiments with Javascript
Inlined JavaScript
Wed Jan 31 2024 16:30:32 GMT-0500 (Eastern Standard Time)

```
            <input type="submit" value="Submit">
        </form>

    </div>
    <div>
        <i> Form with an HTTP POST request</i>
        <form action="/echo.php" method="GET" name="echo_po
            Your Input: <input name="input" onkeypress="con
            <input type="submit" value="Submit">

        </form>
    </div>
    <hr>
    <b>Experiments with Javascript</b><br>
    <i>Inlined JavaScript</i>
    <div id="date" onclick="document.getElementById('date')
/div>

/>
```

```
You pressed a key                                   waph-suvarnrk.html:43
```

Figure 4: Display date/time when clicked



This is web application programming and hacking lab

Performing front end development

Instructor: Dr. Phu Phung

Student: Ruthvik Suvarnakanti

Current time:Wed Jan 31 2024 16:34:23 GMT-0500 (Eastern Standard Time)

this is a simple html page

Using the W3Schools template

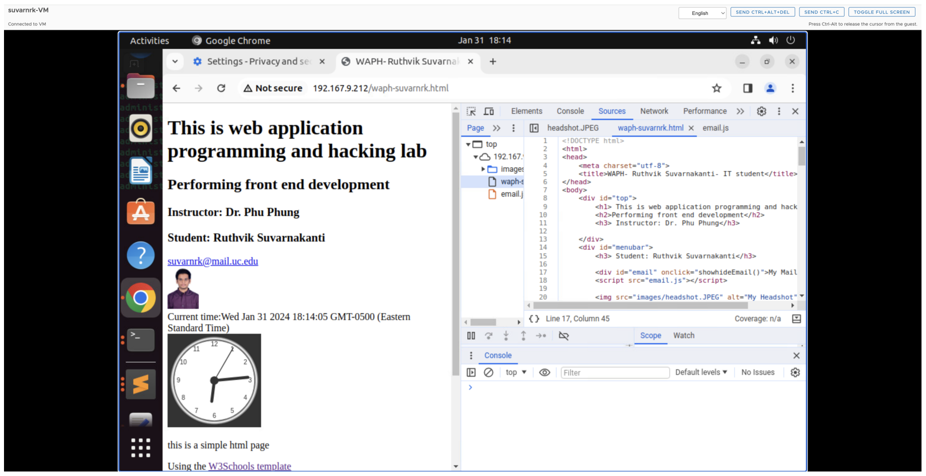
Interaction with forms
Form with an HTTP GET request
Your Input: Submit
Form with an HTTP POST request
Your Input: k Submit

Experiments with Javascript
Inlined JavaScript
Wed Jan 31 2024 16:30:32 GMT-0500 (Eastern Standard Time)

```

</div>
<div id="menubar">
    <h3> Student: Ruthvik Suvarnakanti</h3>

    <img src="images/headshot.JPEG" alt="My Headshot" wi
    <div id="digit-clock"></div>
    <script type="text/javascript">
        function displayTime(){
            document.getElementById('digit-clock').inner
        }
        setInterval(displayTime,500);
    </script>
</div>
<div id="main">
    <p> this is a simple html page</p>
    Using the <a href="https://www.w3schools.com/html/">

    <hr>
    <b>Interaction with forms</b>
```

```
You pressed a key                                   waph-suvarnrk.html:43
```

Figure 5: Display digital clock



This is web application programming and hacking lab

Performing front end development

Instructor: Dr. Phu Phung

Student: Ruthvik Suvarnakanti

suvarnrk@mail.uc.edu

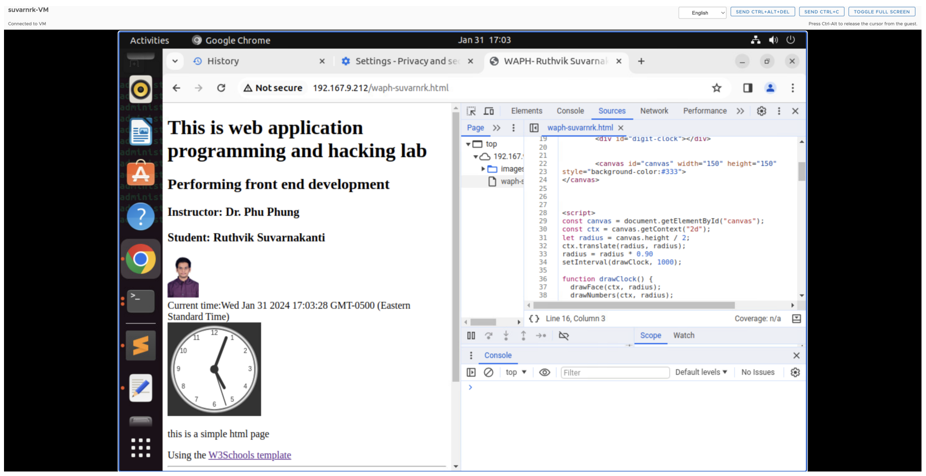
Current time:Wed Jan 31 2024 18:14:05 GMT-0500 (Eastern Standard Time)

this is a simple html page

Using the W3Schools template

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>WAPH- Ruthvik Suvarnakanti- IT student</title>
</head>
<body>
    <div id="top">
        <h1> This is web application programming and hack
        <h2>Performing front end development</h2>
        <h3> Instructor: Dr. Phu Phung</h3>

    </div>
    <div id="menubar">
        <h3> Student: Ruthvik Suvarnakanti</h3>

        <div id="email" onclick="showhideEmail()">My Mail
        <script src="email.js"></script>

        <img src="images/headshot.JPEG" alt="My Headshot"
```

Figure 6: show email when clicked



This is web application programming and hacking lab

Performing front end development

Instructor: Dr. Phu Phung

Student: Ruthvik Suvarnakanti

Current time:Wed Jan 31 2024 17:03:28 GMT-0500 (Eastern Standard Time)

this is a simple html page

Using the W3Schools template

```
        <div id="digit-clock"></div>


        <canvas id="canvas" width="150" height="150"
style="background-color:#333">
</canvas>



<script>
const canvas = document.getElementById("canvas");
const ctx = canvas.getContext("2d");
let radius = canvas.height / 2;
ctx.translate(radius, radius);
radius = radius * 0.90
setInterval(drawClock, 1000);

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

Figure 7: Display analog clock

```
        var myemail="<a href='mailto:suvarnrk"+"@"+"mail.uc.edu'>suvarnrk"+"@"+"mail.uc.edu<
        document.getElementById('email').innerHTML=myemail;
        shown=true;
    }
}
```

```
    <div id="email" onclick="showOrHideEmail()">Show my email</div>
    <script type="text/javascript" src="email.js"></script>
```

-Displaying an Analog clock with an external Javascript code and code in HTML page.

```
<canvas id="analog-clock" width="150" height="150" style="background-color:#999"></canvas>
<script src="https://waph-uc.github.io/clock.js"></script>
<script type=text/javascript>
            const canvas = document.getElementById("canvas");
const ctx = canvas.getContext("2d");
let 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);
}
</script>
```

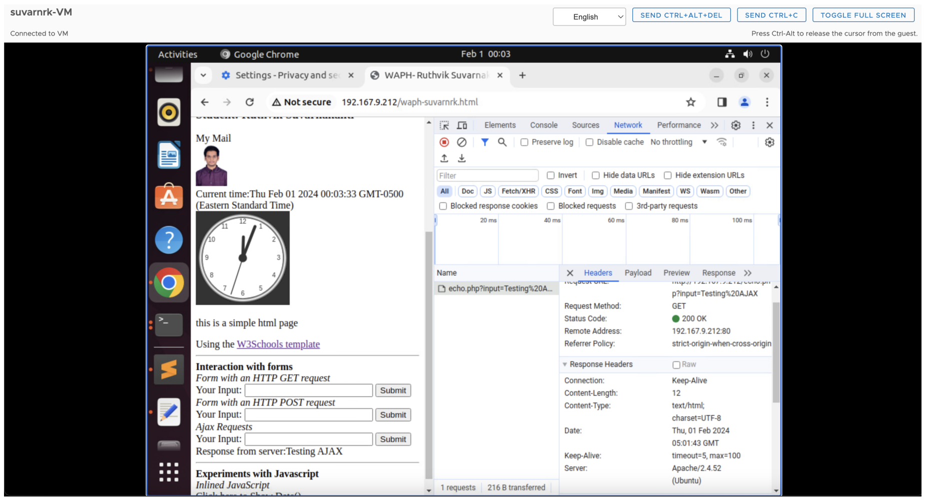
Part II - Ajax, CSS, jQuery, and Web API integration

Task 1: Ajax

We wrote HTML code that captures user input and uses AJAX to make a GET call to echo.php. The received response is then showcased within a designated div. Since it's a GET call, the input was transmitted as a path variable in the URL.

```
<div>
    <i> Ajax Requests</i><br>
    Your Input:
    <input name="data"
    onkeypress="console.log('You have pressed a key ')" id="data">
    <input type="button" class="button round" value="Ajax Echo" onclick="getEcho()">
    <div id="response"></div>
        <input class="button round" type="submit" value="JQuery Ajax Echo" onclick="get
            <input class="button round" type="submit" value="JQuery Ajax Echo Post" oncl
            <input class="button round" type="submit" value="Guess Age" onclick="guessAg
            <div id="response"></div>
</script>
```

We examined the Ajax call response in the inspect view, observing that the request method was GET, the status code indicated a successful 200 OK, and the input data was transmitted within the URL.



```
192.167.9.212/waph-suvarnrk.html

My Mail
Current time:Thu Feb 01 2024 00:03:33 GMT-0500 (Eastern Standard Time)

this is a simple html page

Using the W3Schools template

Interaction with forms
Form with an HTTP GET request
Your Input: [          ] Submit
Form with an HTTP POST request
Your Input: [          ] Submit
Ajax Requests
Your Input: [          ] Submit
Response from server:Testing AJAX

Experiments with Javascript
Inlined JavaScript
```

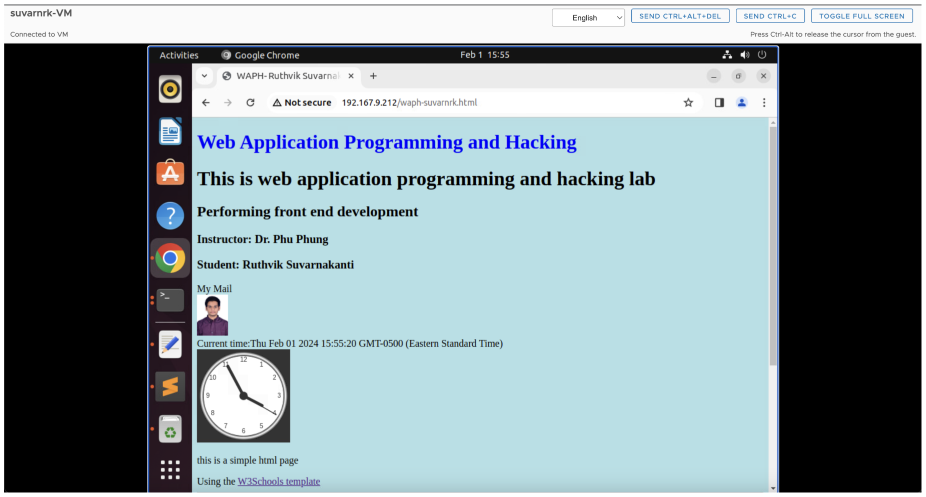
```
Name: echo.php?input=Testing%20A...
                     p?input=Testing%20AJAX
Request Method:      GET
Status Code:         ● 200 OK
Remote Address:      192.167.9.212:80
Referrer Policy:     strict-origin-when-cross-origin

▼ Response Headers   ☐ Raw
Connection:          Keep-Alive
Content-Length:      12
Content-Type:        text/html;
                     charset=UTF-8
Date:                Thu, 01 Feb 2024
                     05:01:43 GMT
Keep-Alive:          timeout=5, max=100
Server:              Apache/2.4.52
                     (Ubuntu)

1 requests   216 B transferred
```

Figure 8: Making an Ajax get call and inspecting respponse

Task 2: CSS

a) Inline CSS

```
<body style="background-color: powderblue;">
<h1 style="color: blue;">Web Application Programming and Hacking</h1>
```



suvarnrk-VM

Connected to VM

Press Ctrl-Alt to release the cursor from the guest.

192.167.9.212/waph-suvarnrk.html

Web Application Programming and Hacking

This is web application programming and hacking lab

Performing front end development

Instructor: Dr. Phu Phung

Student: Ruthvik Suvarnakanti

My Mail

Current time:Thu Feb 01 2024 15:55:20 GMT-0500 (Eastern Standard Time)

this is a simple html page

Using the W3Schools template

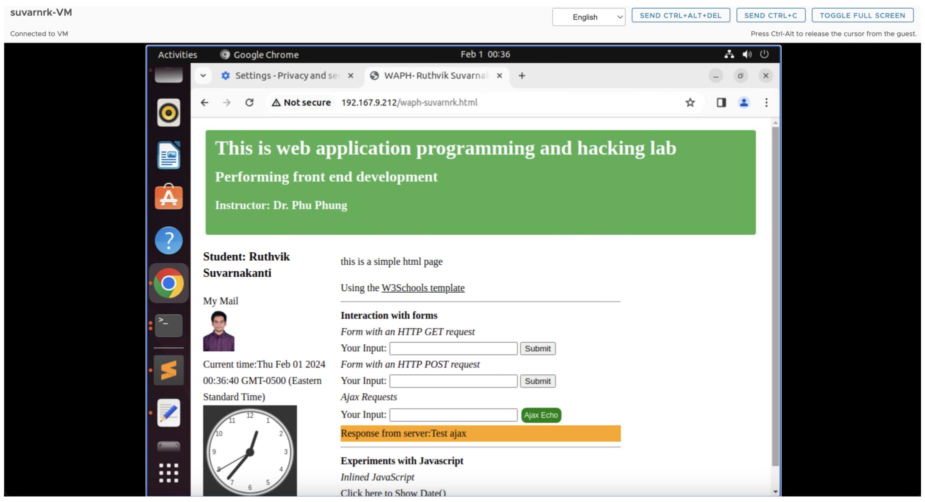
Figure 9: webpage after adding inline CSS

b) Internal CSS.

```
    <style>
        .button{
            background-color:green;
            border: none;
            color: white;
            padding: 5px;
            text-align: center;
            text-decoration: none;
            display: inline-block;
            font-size: 12px;
            margin: 4px 2px;
            cursor: pointer;
        }
        .round{border-radius: 8px;}
        #response{background-color: orange;}
        }
    <!-- HTML code -->
    </style>
    input class="button round" type="submit" value="JQuery Ajax Echo" onclick="getJqueryAja
                <input class="button round" type="submit" value="JQuery Ajax Echo Post" oncl
                <input class="button round" type="submit" value="Guess Age" onclick="guessAg
    <div id="response"></div>
```

c) External CSS from the remote repository provided in the lecture.https://waph-uc.github.io/style1.css.

```
<link rel="stylesheet" type="text/css" href="https://waph-uc.github.io/style1.css">
<!-- HTML code -->
<div class="container wrapper">
<!-- HTML code -->
    <div class="wrapper">
<!-- HTML code -->
    </div>
</div>
```



This is web application programming and hacking lab

Performing front end development

Instructor: Dr. Phu Phung

Student: Ruthvik Suvarnakanti

My Mail

Current time:Thu Feb 01 2024 00:36:40 GMT-0500 (Eastern Standard Time)

this is a simple html page

Using the W3Schools template

Interaction with forms

Form with an HTTP GET request

Your Input: Submit

Form with an HTTP POST request

Your Input: Submit

Ajax Requests

Your Input: Ajax Echo

Response from server:Test ajax

Experiments with Javascript

Inlined JavaScript

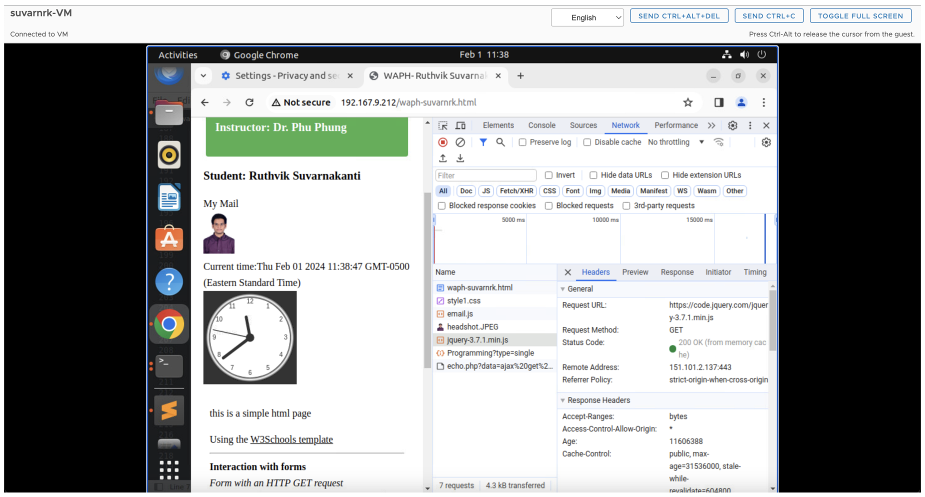
Click here to Show Date()

Figure 10: web page after adding internal CSS and external CSS

Task 3: JQuery

We included the jQuery library in the HTML code, incorporating two buttons one for jQuery Ajax GET and the other for jQuery Ajax POST. These buttons are designed to initiate GET and POST calls, respectively, to echo.php using jQuery. In the case of the Ajax GET request to echo.php, we inspected the response in the view, noting that it was a GET call with a status code of 200 OK. **i.** Ajax GET request to echo.php , the response is analyzed in the inpect view. The call was GET and status code was 200OK.

```
<!-- HTML code -->
<input class="button round" type="submit" value="JQuery Ajax Echo" onclick="getJqueryAja
<!-- HTML code -->
<script>
    function getJqueryAjax(){
        var input=$("#data").val();
            if(input.length==0)
                return;
        $.get("echo.php?data="+input,
                  function(result){
                      printResult(result);
                    });
        $("#data").val("");
        }
    function printResult(result){
        $("#response").html(result);
        }
</script>
```



suvarnrk-VM

Connected to VM

Press Ctrl-Alt to release the cursor from the guest.

Not secure 192.167.9.212/waph-suvarnrk.html

Instructor: Dr. Phu Phung

Student: Ruthvik Suvarnakanti

My Mail

Current time:Thu Feb 01 2024 11:38:47 GMT-0500 (Eastern Standard Time)

this is a simple html page

Using the W3Schools template

Interaction with forms

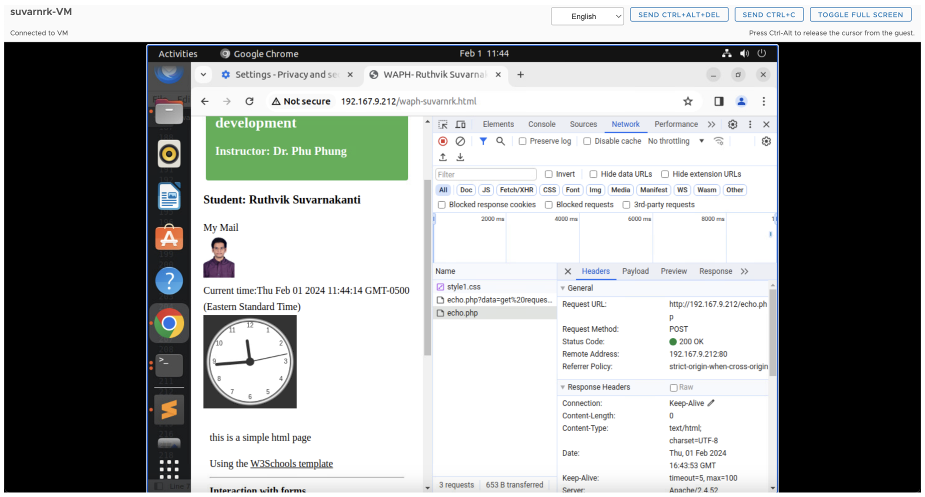
Form with an HTTP GET request

```
Name
waph-suvarnrk.html
style1.css
email.js
headshot.JPEG
jquery-3.7.1.min.js
Programming?type=single
echo.php?data=ajax%20get%2...

7 requests | 4.3 kB transferred

General
Request URL:                  https://code.jquery.com/jquery-3.7.1.min.js
Request Method:               GET
Status Code:                  200 OK (from memory cache)
Remote Address:               151.101.2.137:443
Referrer Policy:              strict-origin-when-cross-origin

Response Headers
Accept-Ranges:                bytes
Access-Control-Allow-Origin:  *
Age:                          11606388
Cache-Control:                public, max-age=31536000, stale-while-revalidate=604800
```

Figure 11: JQuery Ajax GET request to echo.php

i. Ajax POST request to echo.php , the response is analyzed in the inpect view. The call was POST and status code was 200OK.

```
<!-- HTML code -->
<input class="button round" type="submit"
    value="JQuery Ajax Echo Post" onclick="getJqueryAjaxPost()">
<!-- HTML code -->
<script>
    function getJqueryAjaxPost(){
        var input=$("#data").val();
        if(input.length==0)
            return;
        $.post("echo.php",{data:input},function(result){
                printResult(result);
                });
        $("#data").val("");
        }
    function printResult(result){
        $("#response").html(result);
        }
</script>
```



Student: Ruthvik Suvarnakanti

Current time:Thu Feb 01 2024 11:44:14 GMT-0500 (Eastern Standard Time)

this is a simple html page

Using the W3Schools template

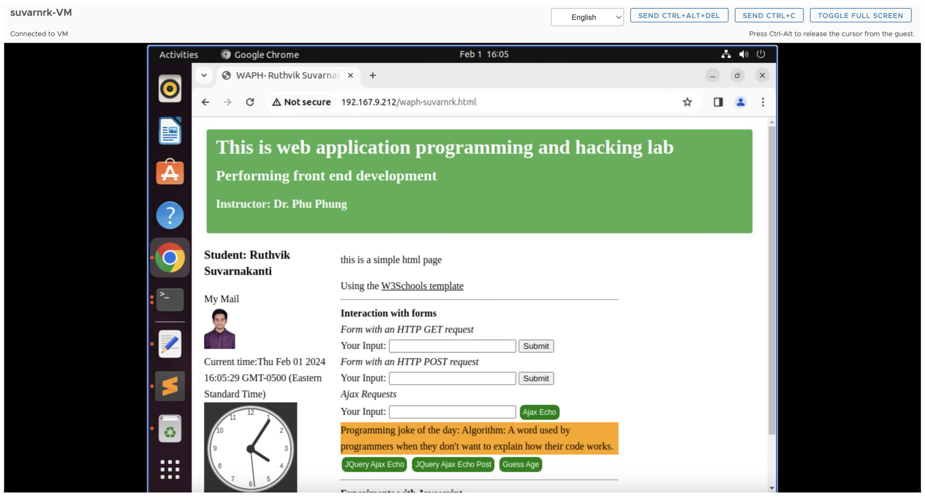
```
Name
style1.css
echo.php?data=get%20reques...
echo.php

3 requests | 653 B transferred

▼ General
Request URL:        http://192.167.9.212/echo.php
Request Method:     POST
Status Code:        ● 200 OK
Remote Address:     192.167.9.212:80
Referrer Policy:    strict-origin-when-cross-origin

▼ Response Headers
Connection:         Keep-Alive
Content-Length:     0
Content-Type:       text/html;
                    charset=UTF-8
Date:               Thu, 01 Feb 2024
                    16:43:53 GMT
Keep-Alive:         timeout=5, max=100
Server:             Apache/2.4.52
```

Figure 12: JQuery Ajax POST request to echo.php

Task 4: WEB API Integration.

i. Using Ajax on https://v2.jokeapi.dev/joke/Programming?type=single

We wrote JavaScript code with jQuery Ajax to make a GET call to the specified web service. After receiving the response in JSON format, we converted it to a string and displayed it in the console. To extract the joke from this response, we used the result.joke property. This service generates a different random joke each time the webpage is refreshed, adding a touch of humor to the user experience.



This is web application programming and hacking lab

Performing front end development

Instructor: Dr. Phu Phung

Student: Ruthvik Suvarnakanti

My Mail

Current time:Thu Feb 01 2024 16:05:29 GMT-0500 (Eastern Standard Time)

this is a simple html page

Using the W3Schools template

Interaction with forms

Form with an HTTP GET request

Your Input: Submit

Form with an HTTP POST request

Your Input: Submit

Ajax Requests

Your Input: Ajax Echo

Programming joke of the day: Algorithm: A word used by programmers when they don't want to explain how their code works.

JQuery Ajax Echo JQuery Ajax Echo Post Guess Age

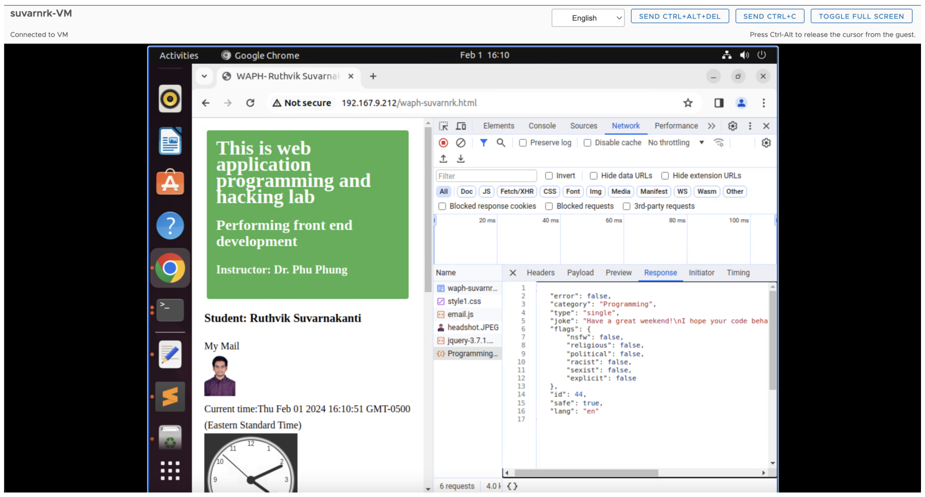
Figure 13: Random Joke displayed when the page is loaded

The below picture represents image of randomly created joke when page is loaded.

```
<!-- HTML code -->
<script>
$.get("https://v2.jokeapi.dev/joke/Programming?type=single",function(result){
            console.log("from joke API: "+ JSON.stringify(result));
            $("#response").html("Programming joke of the day: " +result.joke);
            });
</script>
<!-- HTML code -->
```

ii. Using the `fetch` API on https://api.agify.io/?name=input

In JavaScript, the fetch method is employed to initiate an HTTP request to the mentioned web service. Since this is an asynchronous call, the function is marked with the async keyword, and await is utilized to coordinate the response. The HTTP request executed is of the GET type, and a 200 OK status code confirms the success of the operation.


suvarnrk-VM
Connected to VM
SEND CTRL+ALT+DEL
SEND CTRL+C
TOGGLE FULL SCREEN
Activities
Google Chrome
Feb 1 16:10
WAPH-Ruthvik Suvarna
Not secure 192.167.9.212/waph-suvarnrk.html
This is web application programming and hacking lab
Performing front end development
Instructor: Dr. Phu Phung
Student: Ruthvik Suvarnakanti
My Mail
Current time:Thu Feb 01 2024 16:10:51 GMT-0500 (Eastern Standard Time)
Elements
Console
Sources
Network
Performance
Headers
Payload
Preview
Response
Initiator
Timing
6 requests

```
{
    "error": false,
    "category": "Programming",
    "type": "single",
    "joke": "Have a great weekend!\nI hope your code beha
    "flags": {
        "nsfw": false,
        "religious": false,
        "political": false,
        "racist": false,
        "sexist": false,
        "explicit": false
    },
    "id": 44,
    "safe": true,
    "lang": "en"
}
```

Figure 14: Response of the webservice in inspect view

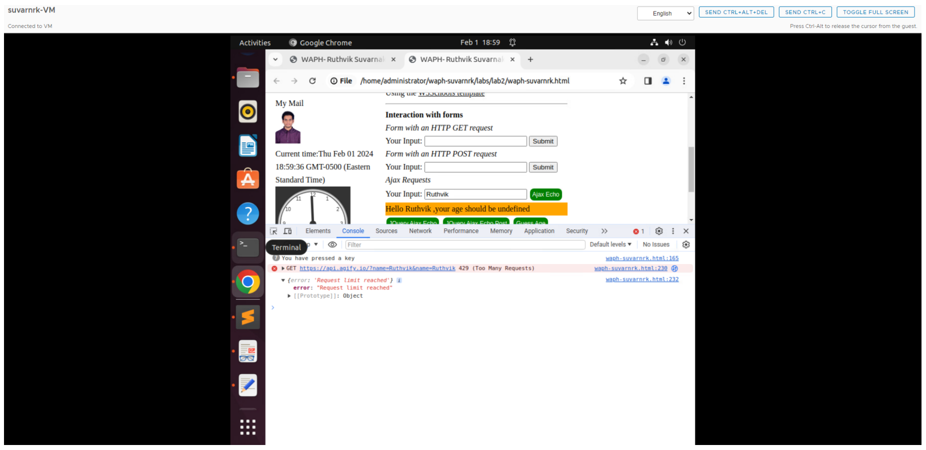
```
<script>
async function guessAge(name){
            const response= await fetch("https://api.agify.io/?name="+name);
            const result= await response.json();
            $("#response").html("Hello "+name+" ,your age should be "+result.age);
    }
</script>
```

The pictures of API calls are not responsive as there are too many request to API and gettting a status of 429 which is meant for request limit reached. I am attaching the pictures of the same here.

Below is the response of the API call.

Below is the final webPage after completing all the tasks and following the lectures.

Following the completion of the lab, a directory named “Lab2” was established to house both the project report and related files. Subsequently, these modifications were pushed to the repository. The project report was generated from the README.md file using the Pandoc tool.



/home/administrator/waph-suvarnrk/labs/lab2/waph-suvarnrk.html

My Mail

Current time:Thu Feb 01 2024 18:59:36 GMT-0500 (Eastern Standard Time)

Interaction with forms

Form with an HTTP GET request

Your Input:

Form with an HTTP POST request

Your Input:

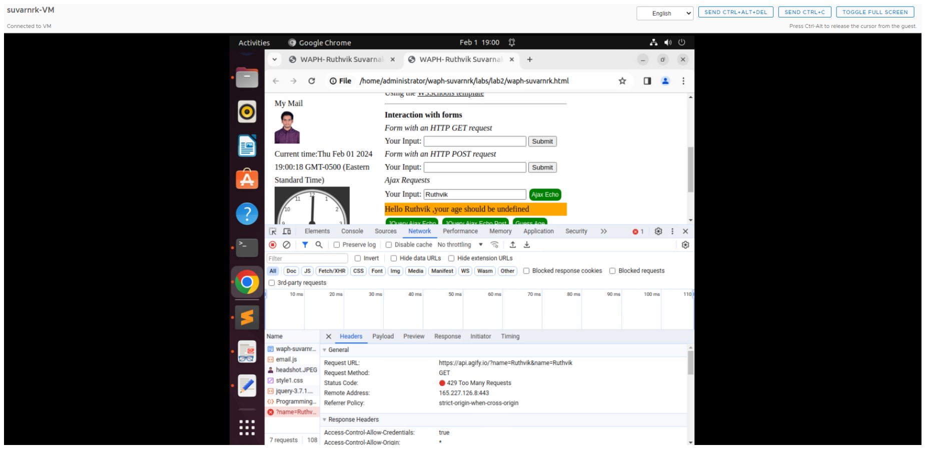
Ajax Requests

Your Input: Ruthvik

Hello Ruthvik ,your age should be undefined

```
You have pressed a key                                                      waph-suvarnrk.html:165
▶ GET https://api.agify.io/?name=Ruthvik&name=Ruthvik 429 (Too Many Requests)  waph-suvarnrk.html:230
▼ {error: 'Request limit reached'}                                          waph-suvarnrk.html:232
    error: "Request limit reached"
  ▶ [[Prototype]]: Object
```

Figure 15: HTTP request to API



/home/administrator/waph-suvarnrk/labs/lab2/waph-suvarnrk.html

My Mail

Current time:Thu Feb 01 2024 19:00:18 GMT-0500 (Eastern Standard Time)

Your Input: Ruthvik

Hello Ruthvik ,your age should be undefined

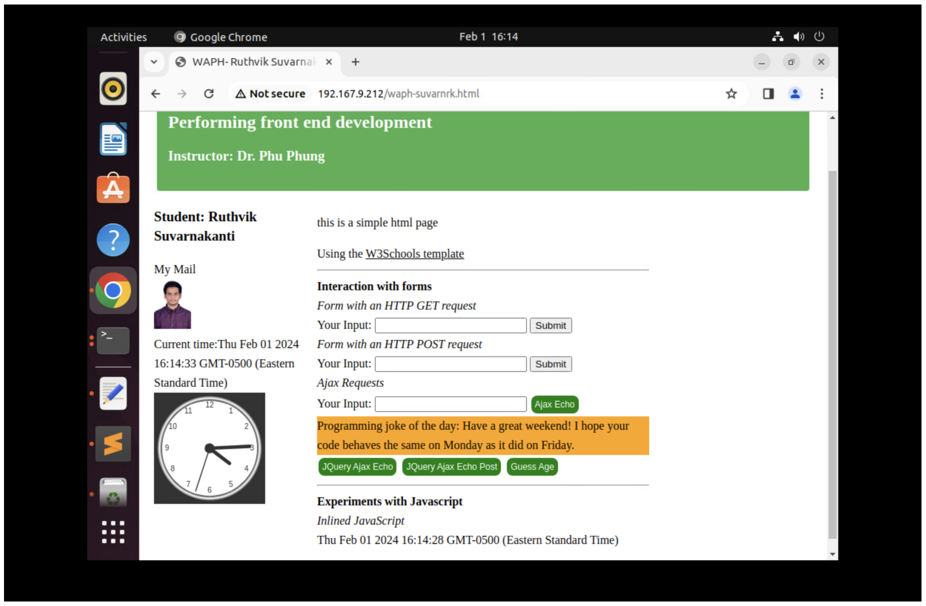
General

Request URL: https://api.agify.io/?name=Ruthvik&name=Ruthvik
Request Method: GET
Status Code: 429 Too Many Requests
Remote Address: 165.227.126.8:443
Referrer Policy: strict-origin-when-cross-origin

Response Headers

Access-Control-Allow-Credentials: true
Access-Control-Allow-Origin: *

Figure 16: Response from API



Performing front end development

Instructor: Dr. Phu Phung

Student: Ruthvik Suvarnakanti

My Mail

Current time:Thu Feb 01 2024 16:14:33 GMT-0500 (Eastern Standard Time)

this is a simple html page

Using the W3Schools template

Interaction with forms

Form with an HTTP GET request

Your Input: Submit

Form with an HTTP POST request

Your Input: Submit

Ajax Requests

Your Input: Ajax Echo

Programming joke of the day: Have a great weekend! I hope your code behaves the same on Monday as it did on Friday.

JQuery Ajax Echo | JQuery Ajax Echo Post | Guess Age

Experiments with Javascript

Inlined JavaScript

Thu Feb 01 2024 16:14:28 GMT-0500 (Eastern Standard Time)

Figure 17: Ruthvik Suvarnakanti Final Page