WAPH - Web Application Programming and Hacking

Instructor: Dr. Phu Phung

Student

Name: Venkata Ramana Rao Devara
Email: devaravo@mail.uc.edu

Repository

Repository URL: https://www.github.com/devaravo/waph-devaravo/tree/main/labs/lab2.

Lab 2 - Front-end Web Development

This lab covers development of basic HTML web page with forms and adding simple javascript to it to display current date/time, digital and analog cloxks, displaying email when clicked etc. After that, it covers Ajax and JQuery requests. Finally, it covers integration of Web APIs using Ajax and fetch API.

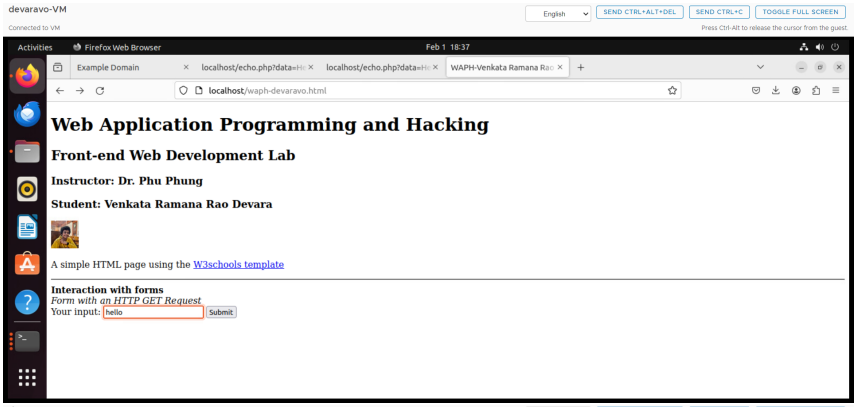
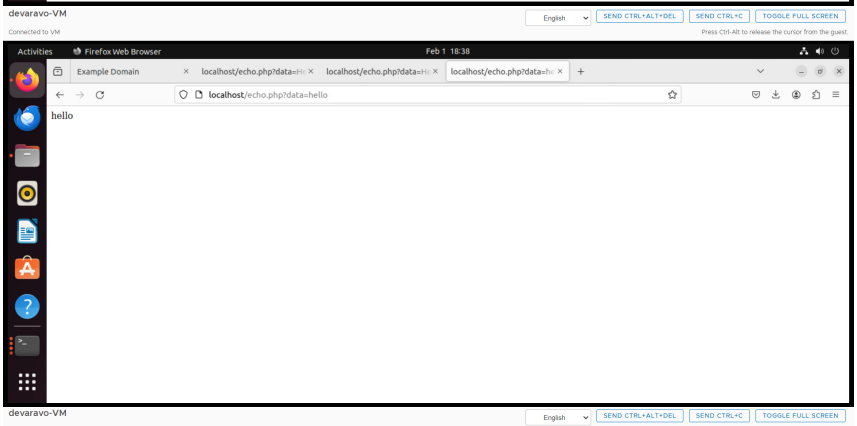
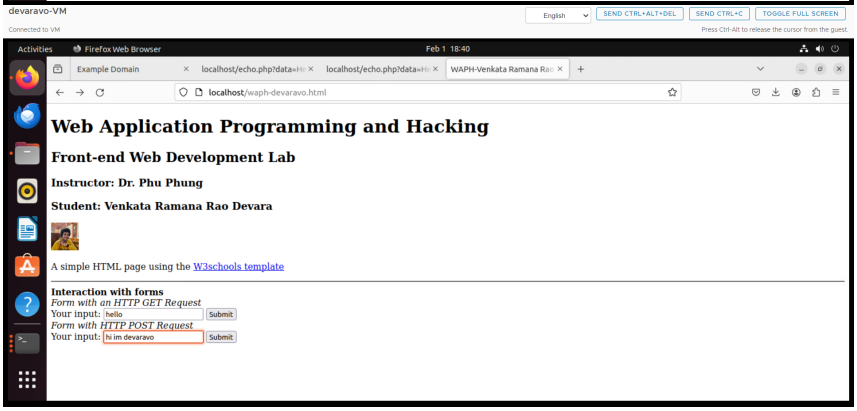
Task 1: Basic HTML with forms, and JavaScript

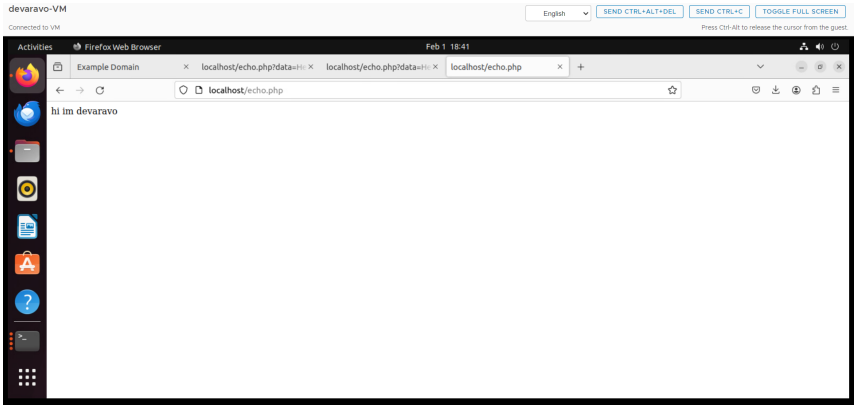
a) HTML

- Created a basic html file "waph-devaravo.html" using W3schools template.
- Included basic html tags (img, a, form). The forms use echo.php for GET and POST requests.
- Sent the HTML page to server and observed the result

```
<!DOCTYPE html>
```

```
<html>
<head>
    <meta charset="utf-8">
    <title>WAPH-Venkata Ramana Rao Devara</title>
</head>
<body>
    <div id="top">
        <h1>Web Application Programming and Hacking</h1>
        <h2>Front-end Web Development Lab</h2>
        <h3>Instructor: Dr. Phu Phung</h3>
    </div>
    <div id="menubar">
        <h3>Student: Venkata Ramana Rao Devara</h3>
        <img src="images/headshot_ramana.png" alt="my headshot" width="50">
    </div>
    <div id="main">
        <p> A simple HTML page using the <a href="https://www.w3schools.com/html">W3schools
        </p>
        <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="data">
                <input type="submit" value="Submit">
            </form>
        </div>
        <div>
            <i>Form with HTTP POST Request</i>
            <form action="/echo.php" method="POST" name="echo_post">
                Your input: <input name="data" >
                <input  type="Submit" value="Submit">
            </form>
        </div>

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

- 
devaravo-VM
Activities
Firefox Web Browser
Feb 1 18:37
Example Domain
WAPH-Venkata Ramana Rao
localhost/waph-devaravo.html
Web Application Programming and Hacking
Front-end Web Development Lab
Instructor: Dr. Phu Phung
Student: Venkata Ramana Rao Devara
A simple HTML page using the W3schools template
Interaction with forms
Form with an HTTP GET Request
Your input: hello
Submit
- 
devaravo-VM
Activities
Firefox Web Browser
Feb 1 18:38
Example Domain
localhost/echo.php?data=hello
hello
- 
devaravo-VM
Activities
Firefox Web Browser
Feb 1 18:40
Example Domain
WAPH-Venkata Ramana Rao
localhost/waph-devaravo.html
Web Application Programming and Hacking
Front-end Web Development Lab
Instructor: Dr. Phu Phung
Student: Venkata Ramana Rao Devara
A simple HTML page using the W3schools template
Interaction with forms
Form with an HTTP GET Request
Your input: hello
Submit
Form with HTTP POST Request
Your input: hi im devaravo
Submit

- 

b) Simple Javascript

- Added JS code to display current date/time when clicked (inline)

```
<div id="date" onclick="document.getElementById('date').innerHTML = Date()">Click here to sh
```

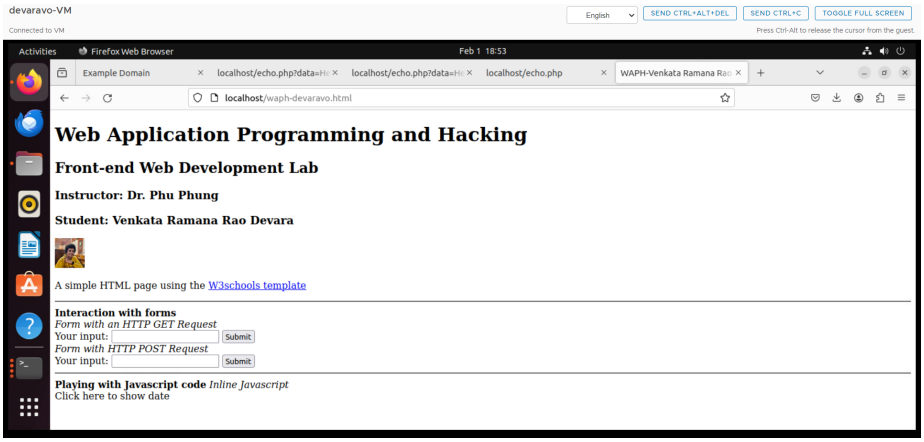


Figure 1: Show Date

-
-
- Added JS code to log when a key is pressed (inline)

```
<input name="data" onkeyup="console.log('You have pressed a key')">
```

-
- Added JS code in script tag to display digital clock

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

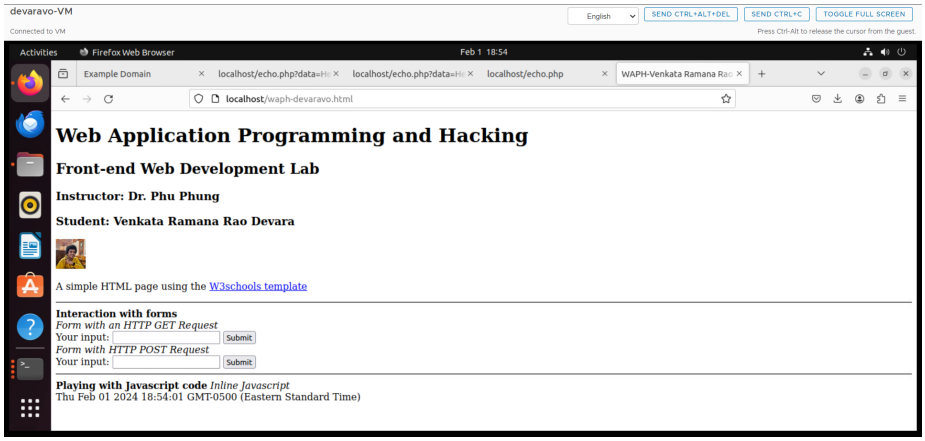


Figure 2: Show Date

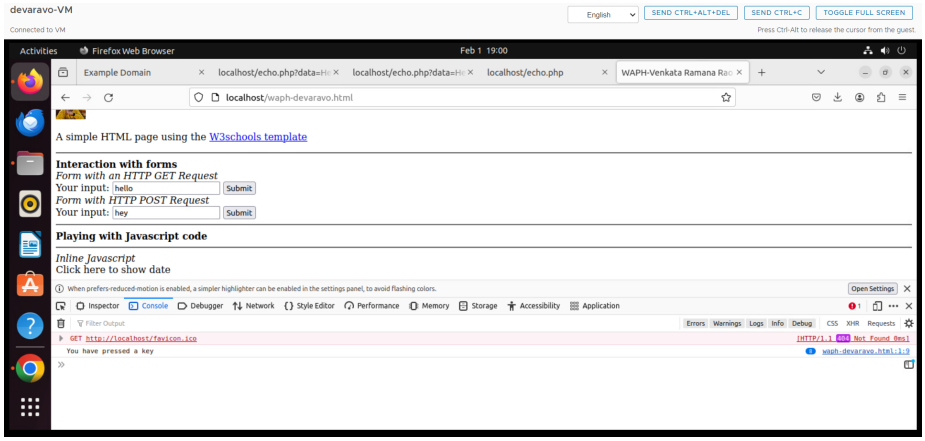


Figure 3: Js to show key is pressed

```
<script>
    function displayTime() {
    document.getElementById("digital-clock").innerHTML="Current time: "+ new Date();
    }
    setInterval(displayTime, 500);
</script>
```

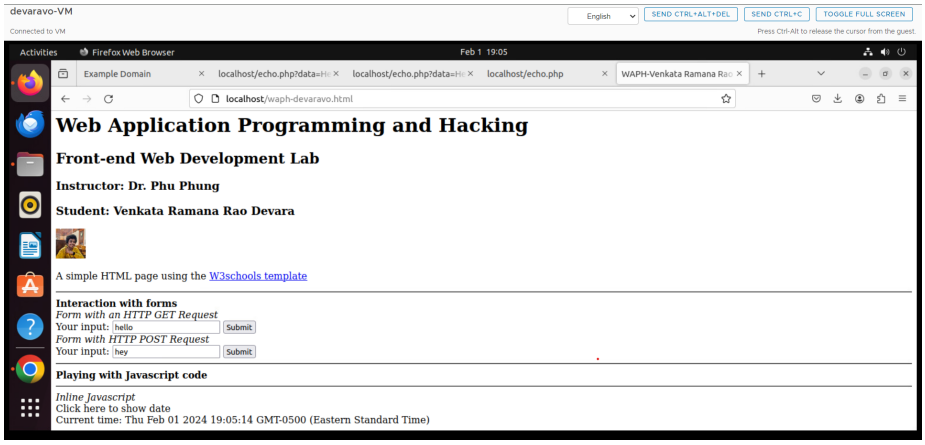


Figure 4: Digital clock

-
- Added JS code in JS file to display or hide email when clicked and included the file in html

email.js:

```
var isShown = false;

function showHideEmail(){
    if(isShown){
        document.getElementById('email').innerHTML = "Show my email";
        isShown = false;
    }
    else{
        var myEmail = "<a href='mailto:devaravo" + "@" + "mail.uc.edu'>devaravo" + "@" + "ma
        document.getElementById('email').innerHTML = myEmail;
        isShown = true;
    }
}
```

html:

```
<div id="email" onclick="showHideEmail()">Show my email</div>
<script src="email.js"></script>
```

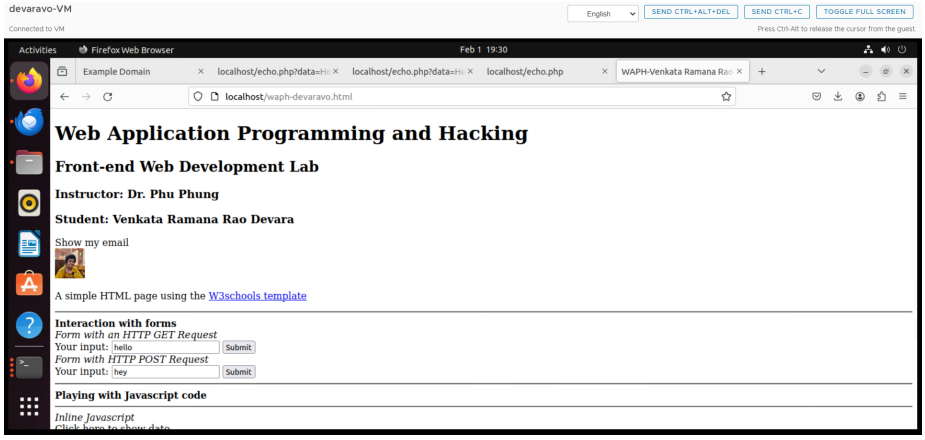


Figure 5: Show email

-

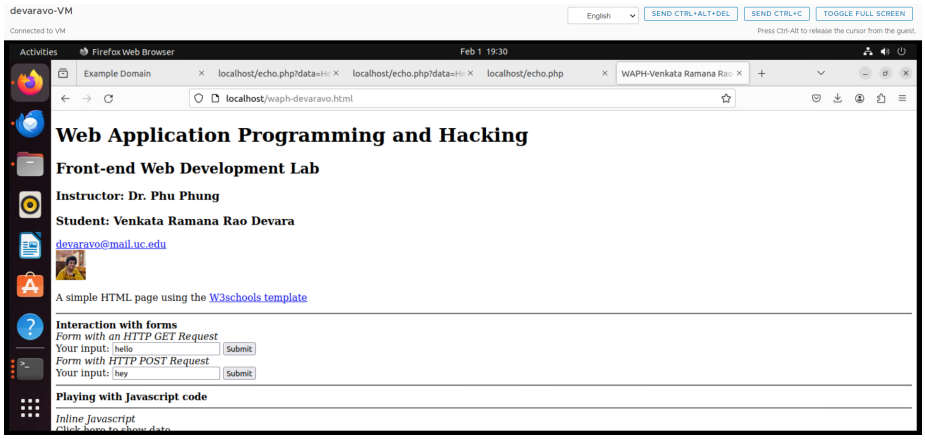
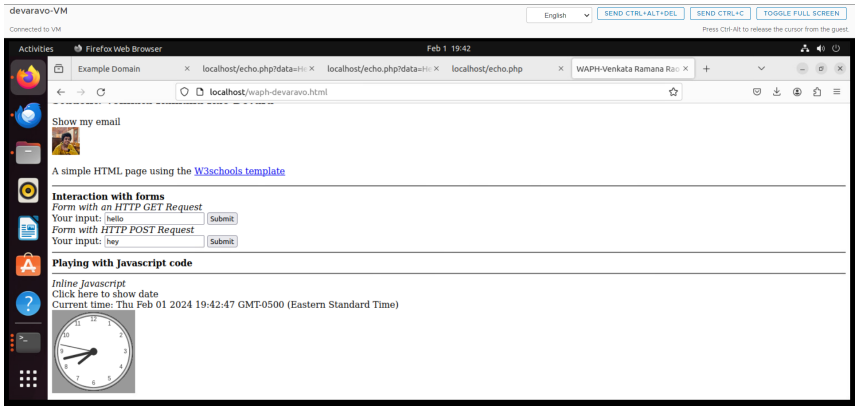


Figure 6: Show email

-
- Added external JS code and displayed analog clock using it.

```
<canvas id="analog-clock" width="150"  height="150" style="background-color:#999"></canvas>

<script src="https://waph-uc.github.io/clock.js"></script>
<script>
```

```
            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);
            }
</script>
```

- 

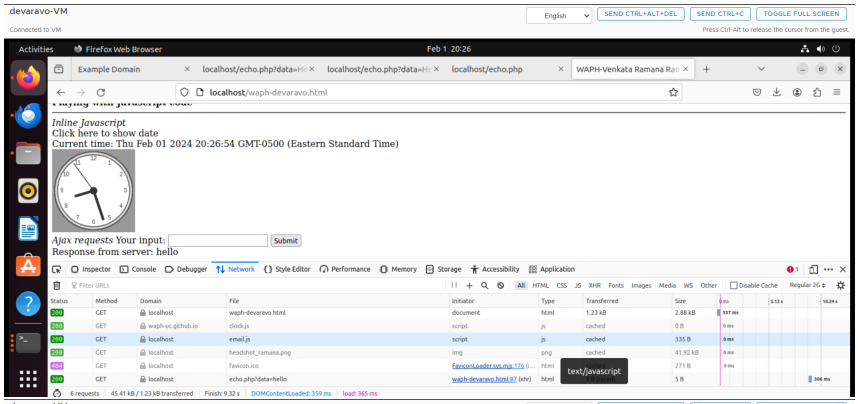
Task 2: Ajax, CSS, jQuery, Web API integration

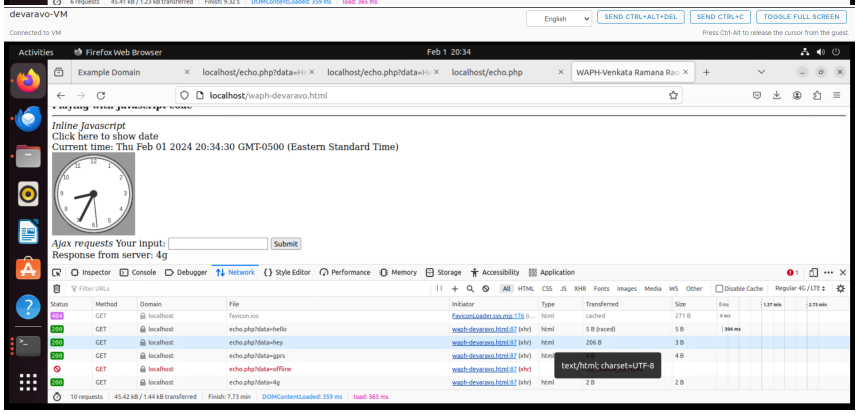
a) Ajax

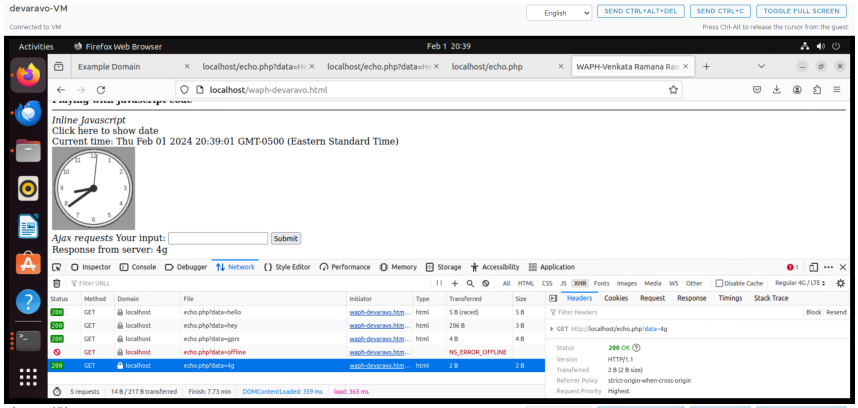
- Added code for user input and to send Ajax GET request to echo.php web application
- Added code to display the result in response element

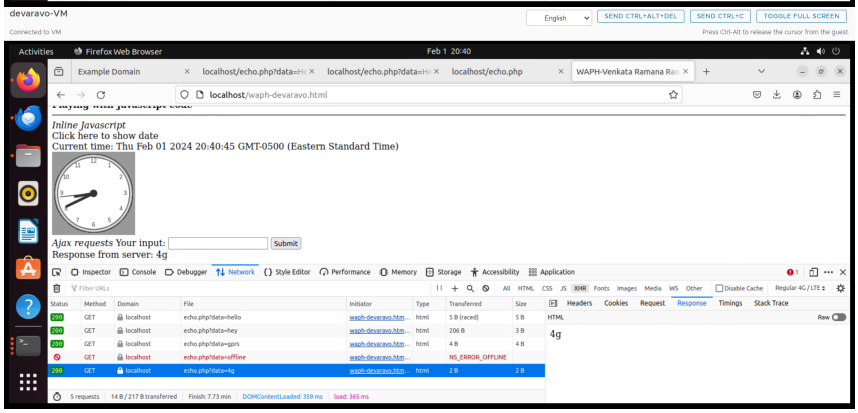
```
Your input: <input name="data" onkeypress="console.log('You have pressed a key')" id="data">
<input type="button" value="Ajax Echo" onclick="getEcho()">
<script>
function getEcho() {
    var input = document.getElementById("data").value;
    if(input.length == 0){
        return
    }

    var xhttp = new XMLHttpRequest();

    xhttp.onreadystatechange = function() {
```

```
        if (this.readyState == 4 && this.status == 200){
            console.log("Received data= "+xhttp.responseText);
            document.getElementById("response").innerText = "Response from server: " + xhttp
        }
    }

    xhttp.open("GET", "echo.php?data="+input, true);
    xhttp.send();
    document.getElementById("data").value="";
}
</script>
```

- 

- 

- 

- 

b) CSS

- Added code for inline CSS

```
<h1 style="color:green;">Web Application Programming and Hacking</h1>
```

-

- Added code for external CSS

```
<link rel="stylesheet" href="https://waph-uc.github.io/style2.css">
```

```
Added corresponding classes in HTML page
```

-

- Added code for internal CSS

```
<style>
        .button{
            background-color: green;
            border: none;
            color: white;
```

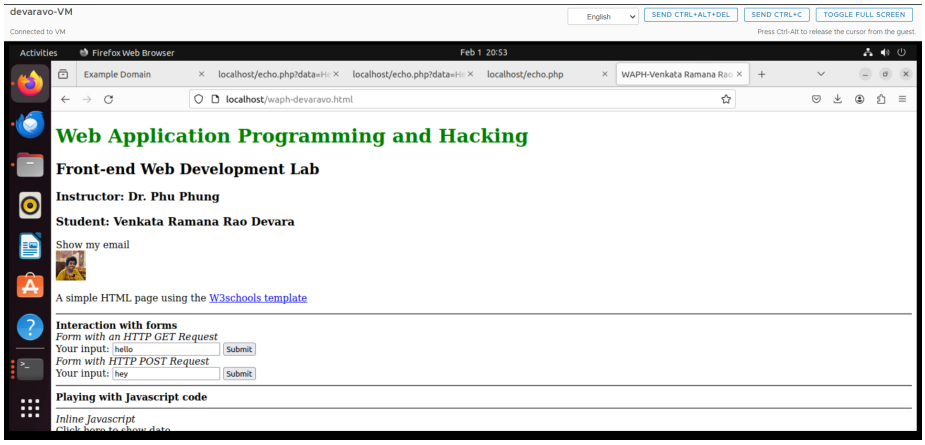


Figure 7: Inline CSS

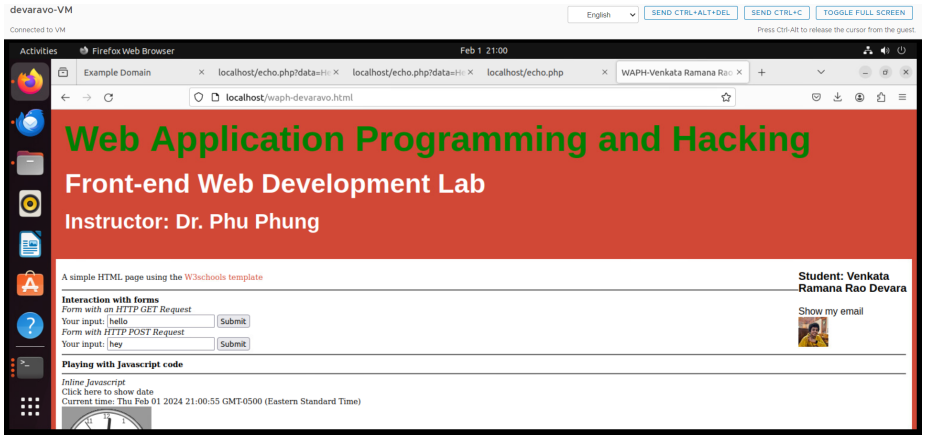
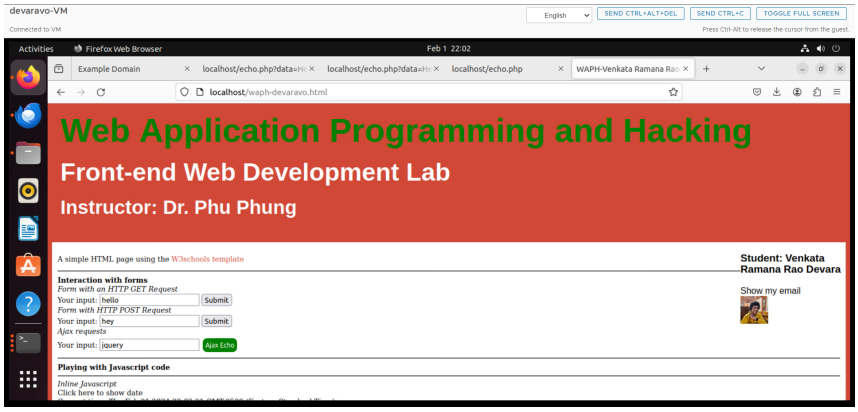


Figure 8: External CSS

```
        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: #FF9800;
    }
```

Added corresponding classes to

- 

c) jQuery

- Added jQuery library to the page

```
<script src="https://code.jquery.com/jquery-3.7.1.min.js"
                integrity="sha256-/JqT3SQfawRcv/BIHPThkBvs0OEvtFFmqPF/lYI/Cxo="
                crossorigin="anonymous"></script>
```

- i) Added code to send Ajax GET request to echo.php and display the result

```
<input class="button round" type="button" value="jQuery Ajax GET Echo" onclick="jQueryAjax()

<script>
function jQueryAjax(){
```

```
    var input = $('#data').val();
    if(input.length == 0)
        return;
    $.get("echo.php?data="+input, function(result){
        $("#response").html("Response from server: "+ result);
    });

$("#data").val("");
}
</script>
```

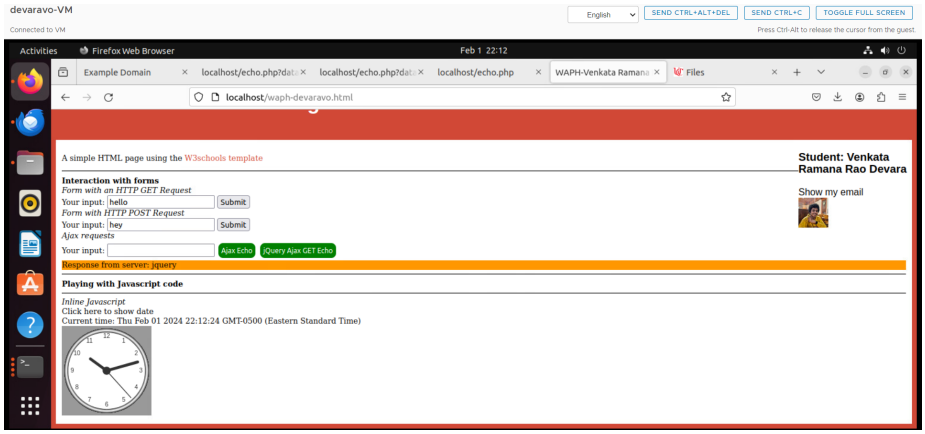
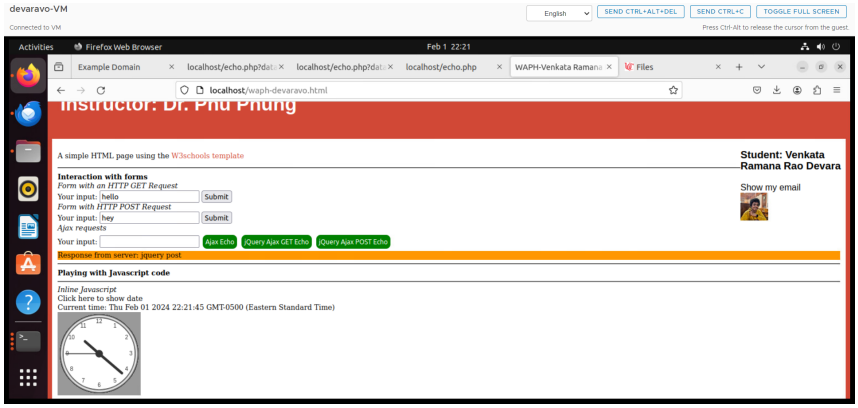


Figure 9: jQuery GET

-
- ii) Added code to send Ajax POST request to echo.php and display the result

```
<input class="button round" type="button" value="jQuery Ajax POST Echo" onclick="jQueryAjaxP
<script>
function jQueryAjaxPost(){
    var input = $('#data').val();
    if(input.length == 0)
        return;
    $.post("echo.php",{data: input}, function(result){
        $("#response").html("Response from server: "+result);
    });
$("#data").val("");
}
</script>
```

- 

d) Web API Integration

- i) Added JS code to send request to Web API (joke API) when the page is loaded and examined the result in browser console , added result to response element

```
$.get("https://v2.jokeapi.dev/joke/Programming?type=single", function(result){
    console.log("From jokeAPI"+result.joke);
    $("#response").html("Programming joke: "+ result.joke);
    });
```

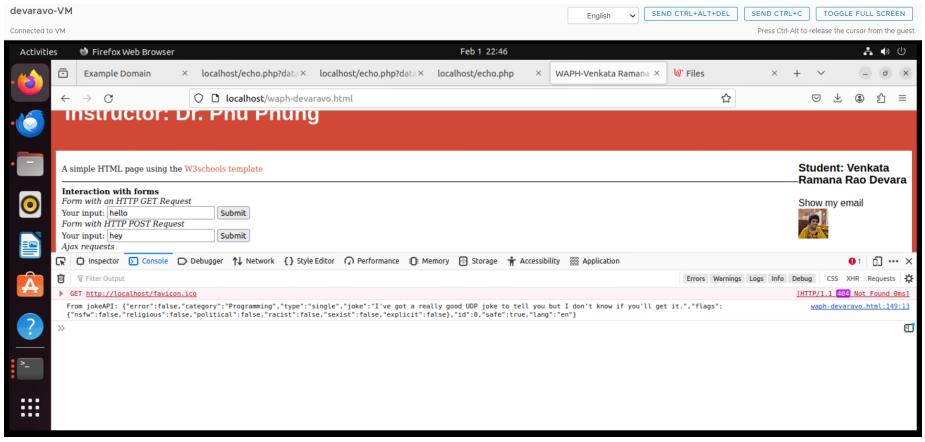


Figure 10: Joke API

-
-
- Observed request and response in browser (network)
-

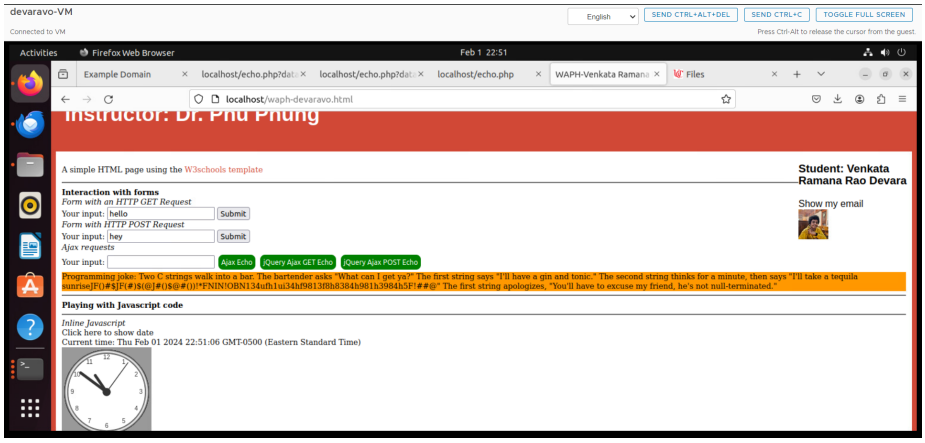


Figure 11: Joke API

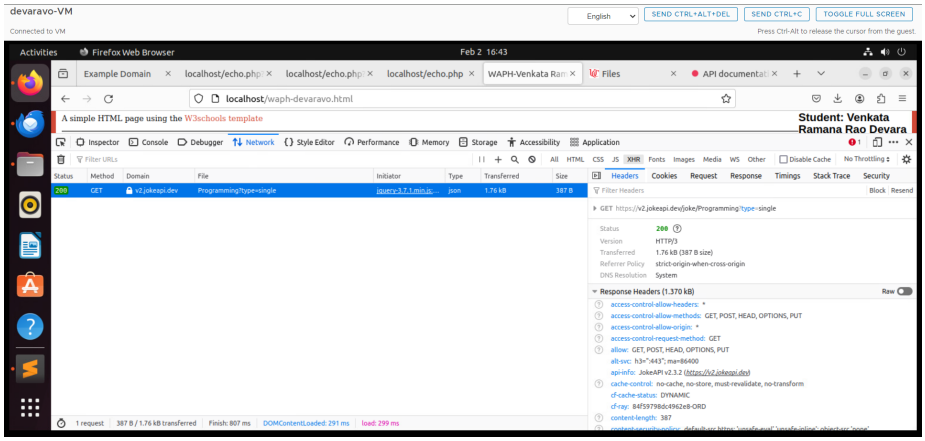


Figure 12: Joke API request

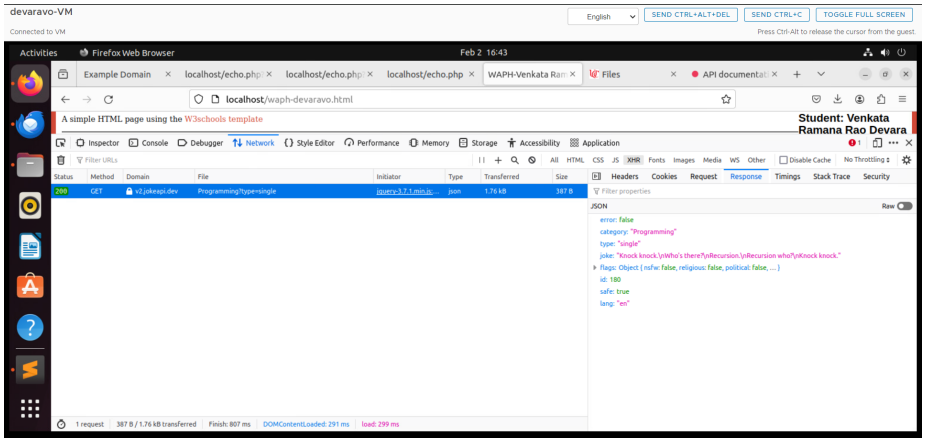


Figure 13: Joke API response

-

- ii) Added JS code to use fetch method to call age guessing api and displayed the result in browser “‘

async function guessAge(name){ const response = await fetch(“https://api.agify.io/?name=”+name); const result = await response.json(); console.log(result); $(“#response”).html(“Hi”+ name + “, your age should be”+result.age); }

“‘s

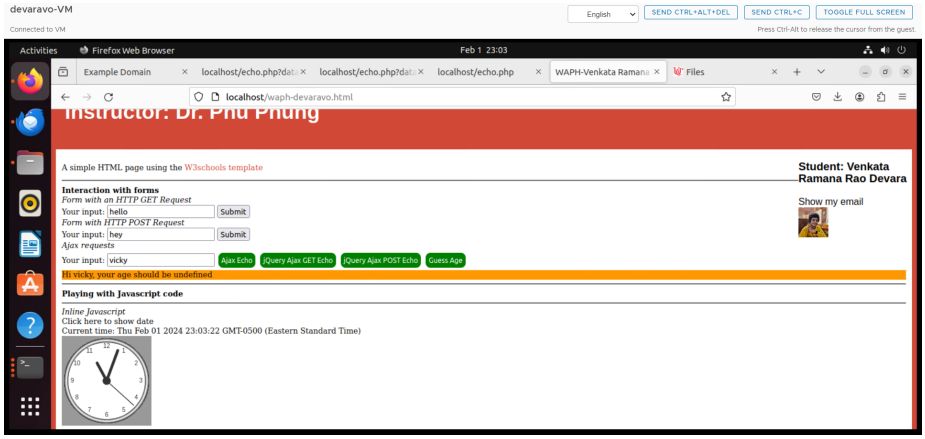


Figure 14: fetch API

-

- Observed request and response in browser (network)

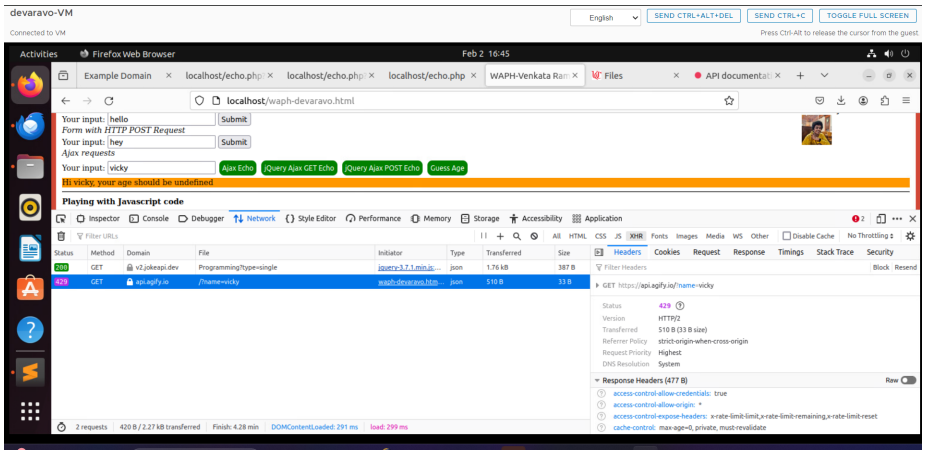


Figure 15: Fetch request

-

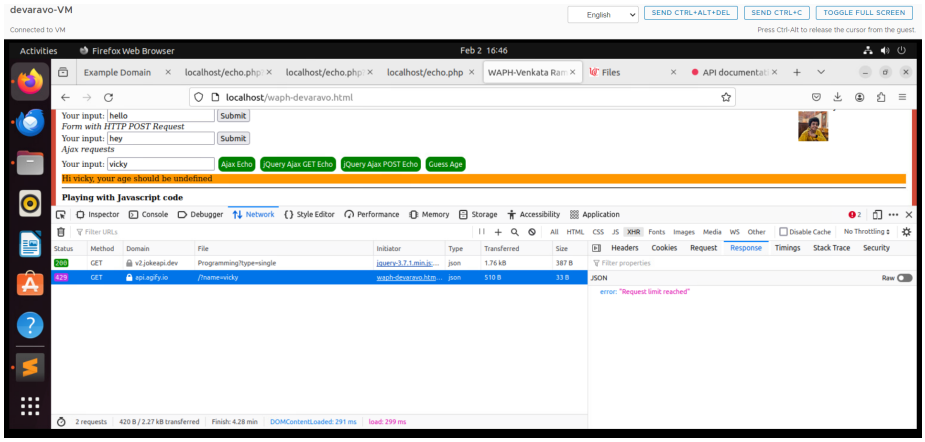


Figure 16: Fetch response

-