Assignment No: - 1

* **Understand basic PHP syntax, use variables, and explore different data types.**

**Task:**

Create a PHP script that displays "Hello, World!" on the webpage.

Define variables of different data types (integer, string, Boolean, float).

Print out the variables and their data types using appropriate PHP functions.

Perform basic arithmetic operations using variables and display the results.

**Code:**

<!DOCTYPE html>

<html>

<head>

<title>Hello, World!</title>

</head>

<body>

<?php

// Display "Hello, World!"

echo "<h1>Hello, World!</h1>";

// Define variables of different data types

$integerVar = 10;

$stringVar = "Hello";

$booleanVar = true;

$floatVar = 3.14;

// Print out variables and their data types

echo "<p>Integer Variable: $integerVar (Type: " . gettype($integerVar) . ")</p>";

echo "<p>String Variable: $stringVar (Type: " . gettype($stringVar) . ")</p>";

echo "<p>Boolean Variable: $booleanVar (Type: " . gettype($booleanVar) . ")</p>";

echo "<p>Float Variable: $floatVar (Type: " . gettype($floatVar) . ")</p>";

// Perform basic arithmetic operations using variables and display the results

$resultAddition = $integerVar + $floatVar;

$resultSubtraction = $integerVar - $floatVar;

$resultMultiplication = $integerVar \* $floatVar;

$resultDivision = $integerVar / $floatVar;

echo "<p>Arithmetic Operations:</p>";

echo "<p>Addition: $resultAddition</p>";

echo "<p>Subtraction: $resultSubtraction</p>";

echo "<p>Multiplication: $resultMultiplication</p>";

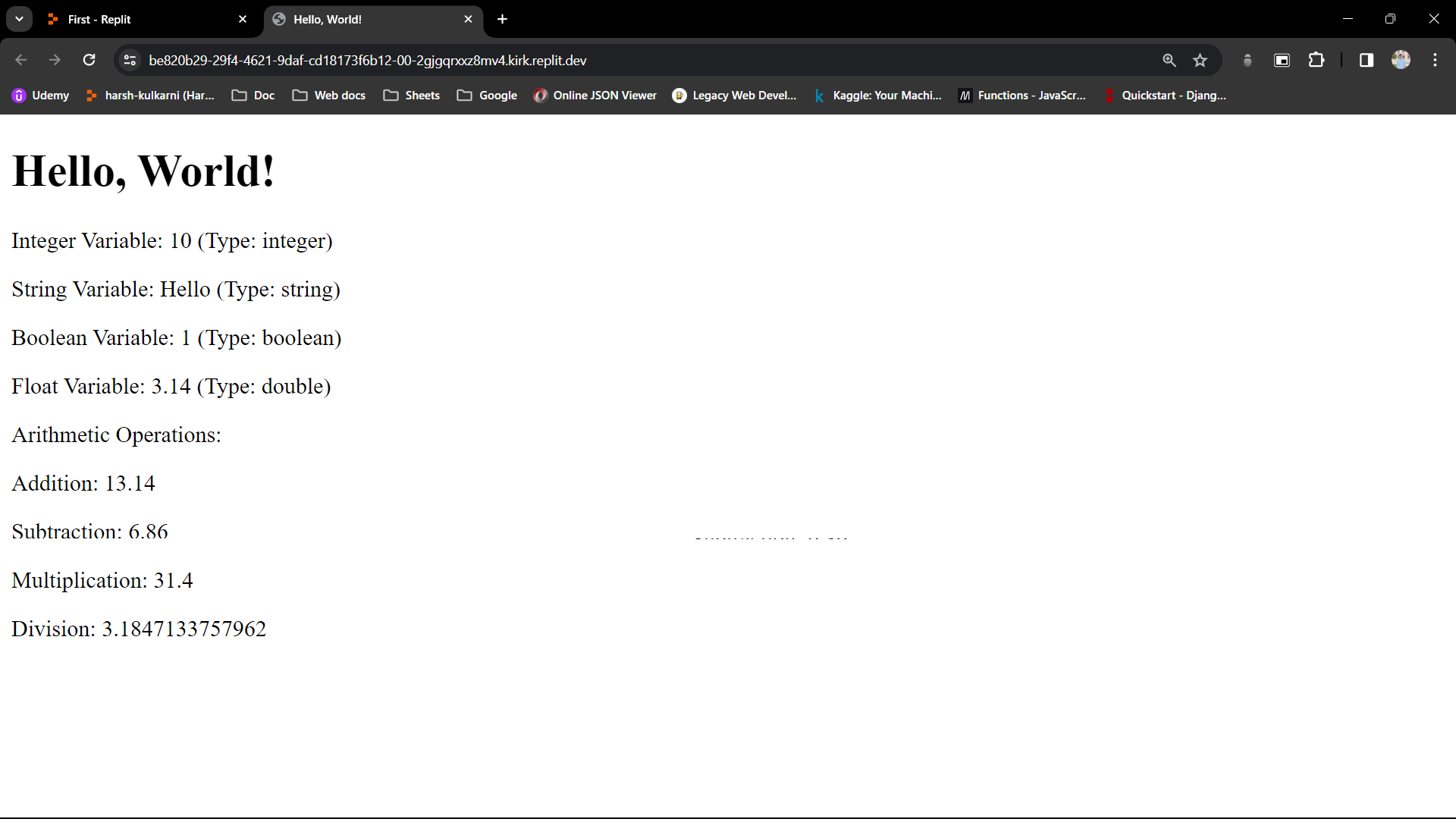
echo "<p>Division: $resultDivision</p>";

?>

</body>

</html>

**Output:**



Assignment No: - 2

* **Practice using conditional statements in PHP.**

Task:

Write a PHP script that takes a number as input.

Use an if-else statement to determine if the number is positive, negative, or zero.

Display an appropriate message based on the input.

**Code:**

<!DOCTYPE html>

<html>

<head>

<title>Number Analysis</title>

</head>

<body>

<form method="post" action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]);?>">

Enter a number: <input type="number" name="number">

<input type="submit" name="submit" value="Submit">

</form>

<?php

// Check if the form is submitted

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

// Check if the number is set and not empty

if (isset($\_POST["number"]) && !empty($\_POST["number"])) {

$number = $\_POST["number"];

// Determine if the number is positive, negative, or zero

if ($number > 0) {

echo "<p>The number $number is positive.</p>";

} elseif ($number < 0) {

echo "<p>The number $number is negative.</p>";

} else {

echo "<p>The number is zero.</p>";

}

} else {

echo "<p>Please enter a valid number.</p>";

}

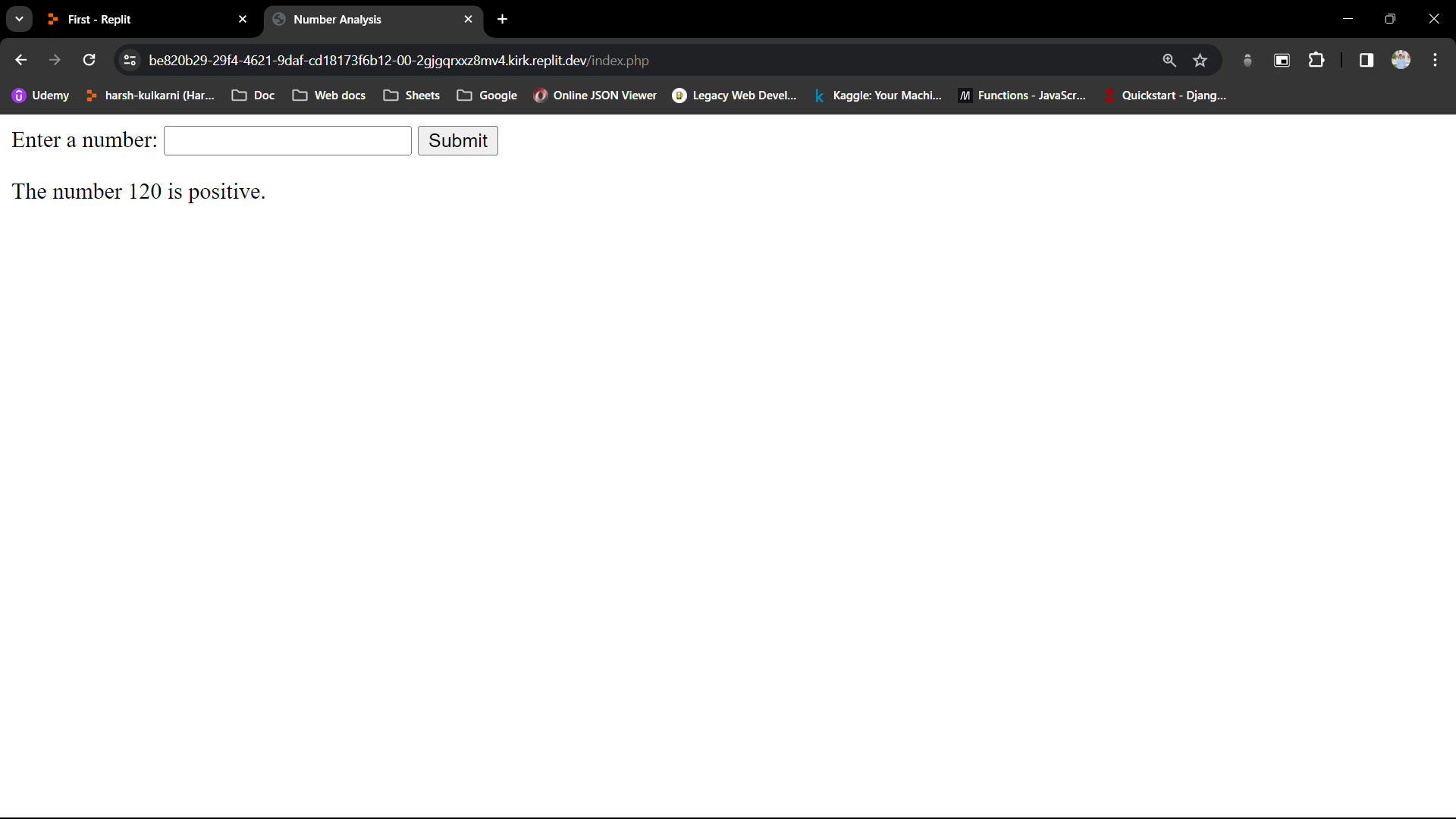
}

?>

</body>

</html>

**Output:**



Assignment No: - 3

* **Understand different types of loops in PHP.**

Task:

Write a PHP script that prints numbers from 1 to 10 using a for loop.

Create another script that prints even numbers from 1 to 20 using a while loop.

Use a do-while loop to print the first 5 multiples of a number entered by the user.

**Code:**

**// Using a for loop**

<!DOCTYPE html>

<html>

<head>

<title>Printing Numbers with for Loop</title>

</head>

<body>

<h2>Numbers from 1 to 10 using for Loop:</h2>

<?php

for ($i = 1; $i <= 10; $i++) {

echo "$i ";

}

?>

</body>

</html>

**// Using a while loop**

<!DOCTYPE html>

<html>

<head>

<title>Printing Even Numbers with while Loop</title>

</head>

<body>

<h2>Even Numbers from 1 to 20 using while Loop:</h2>

<?php

$num = 1;

while ($num <= 20) {

if ($num % 2 == 0) {

echo "$num ";

}

$num++;

}

?>

</body>

</html>

**// Using do-while loop**

<!DOCTYPE html>

<html>

<head>

<title>Printing Multiples with do-while Loop</title>

</head>

<body>

<h2>Print the First 5 Multiples of a Number:</h2>

<form method="post" action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]);?>">

Enter a number: <input type="number" name="number">

<input type="submit" name="submit" value="Submit">

</form>

<?php

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

if (isset($\_POST["number"]) && !empty($\_POST["number"])) {

$num = $\_POST["number"];

$count = 1;

echo "<p>The first 5 multiples of $num are:</p>";

do {

$multiple = $num \* $count;

echo "$multiple ";

$count++;

} while ($count <= 5);

} else {

echo "<p>Please enter a valid number.</p>";

}

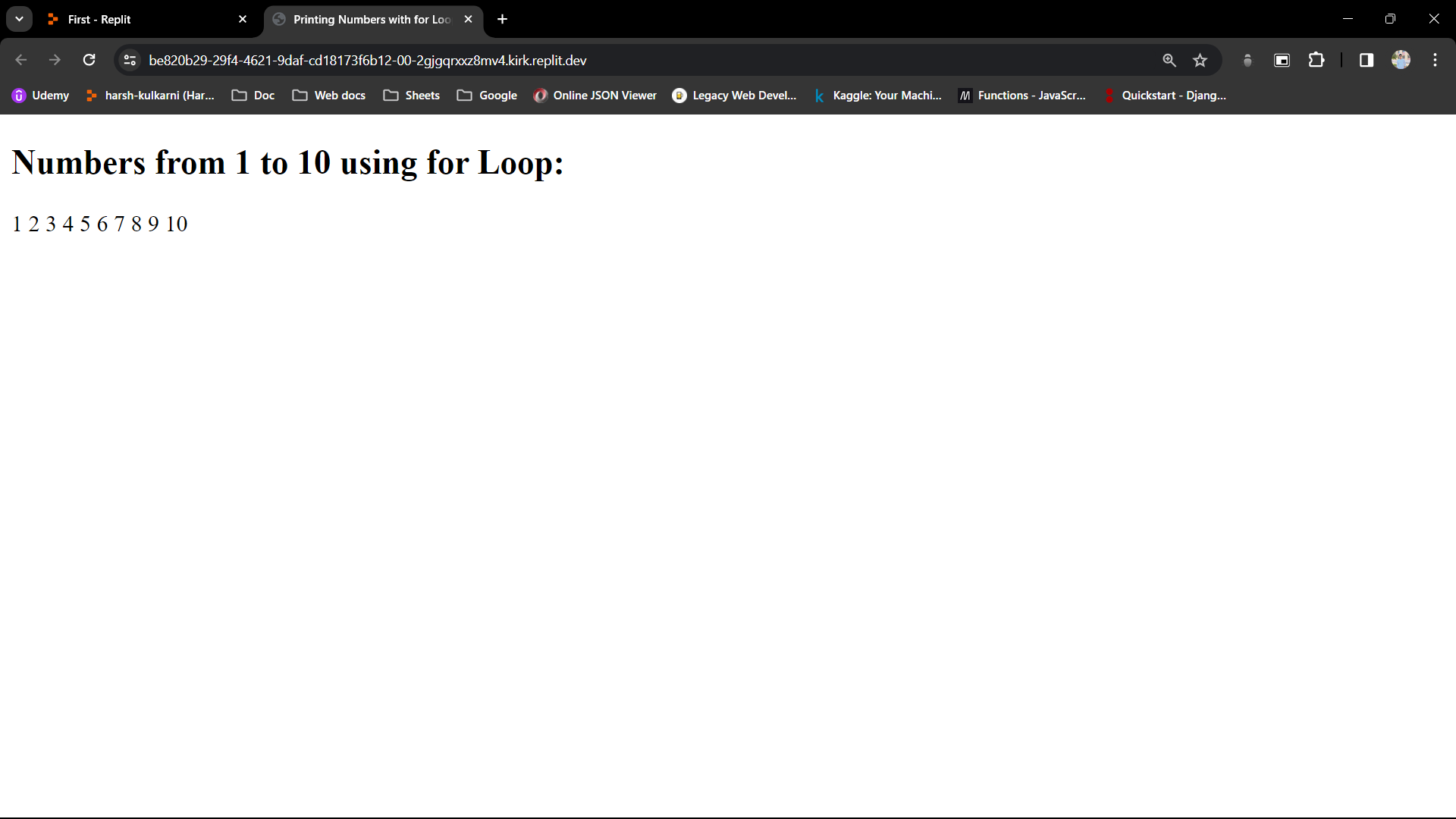
}

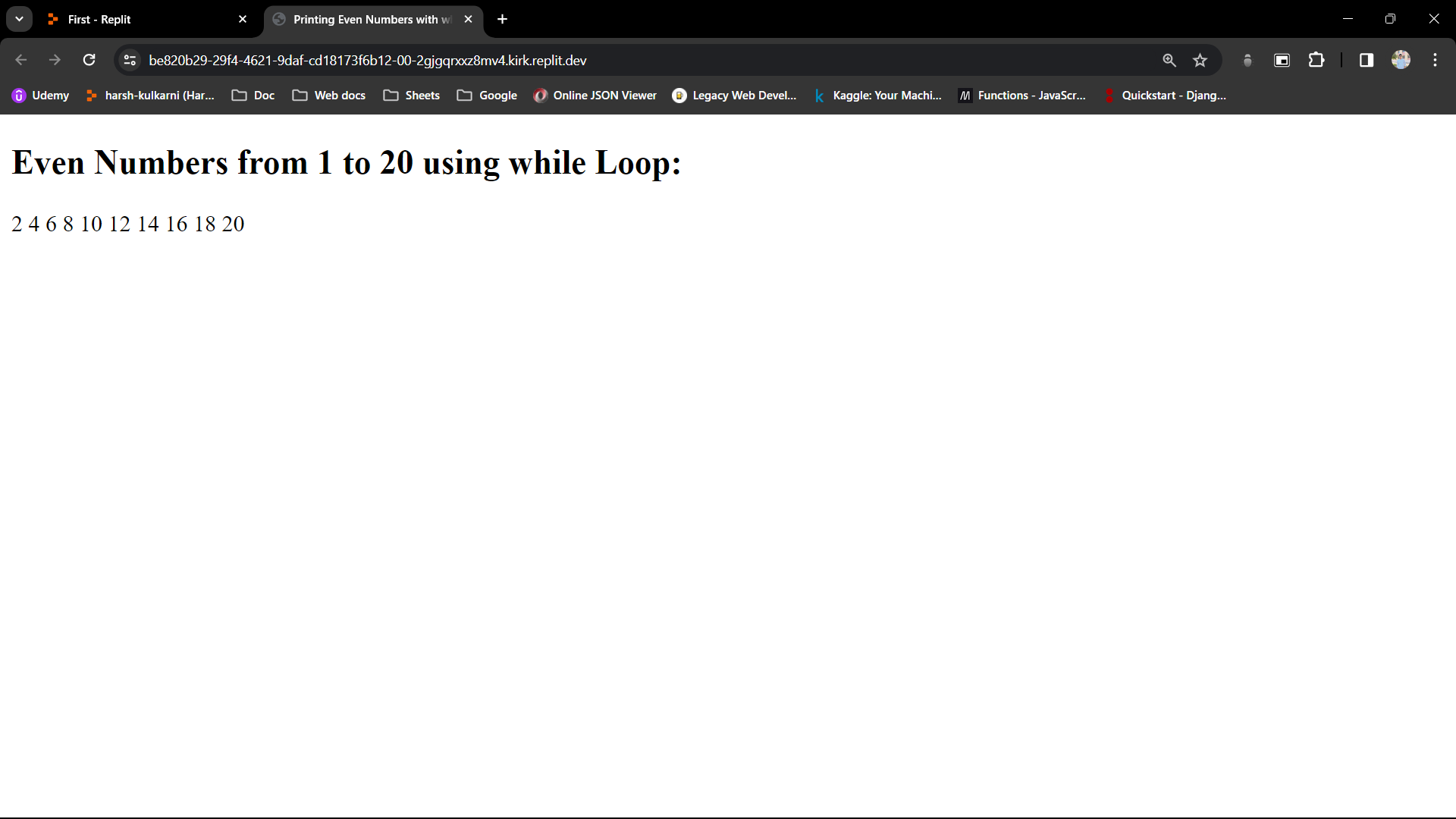
?>

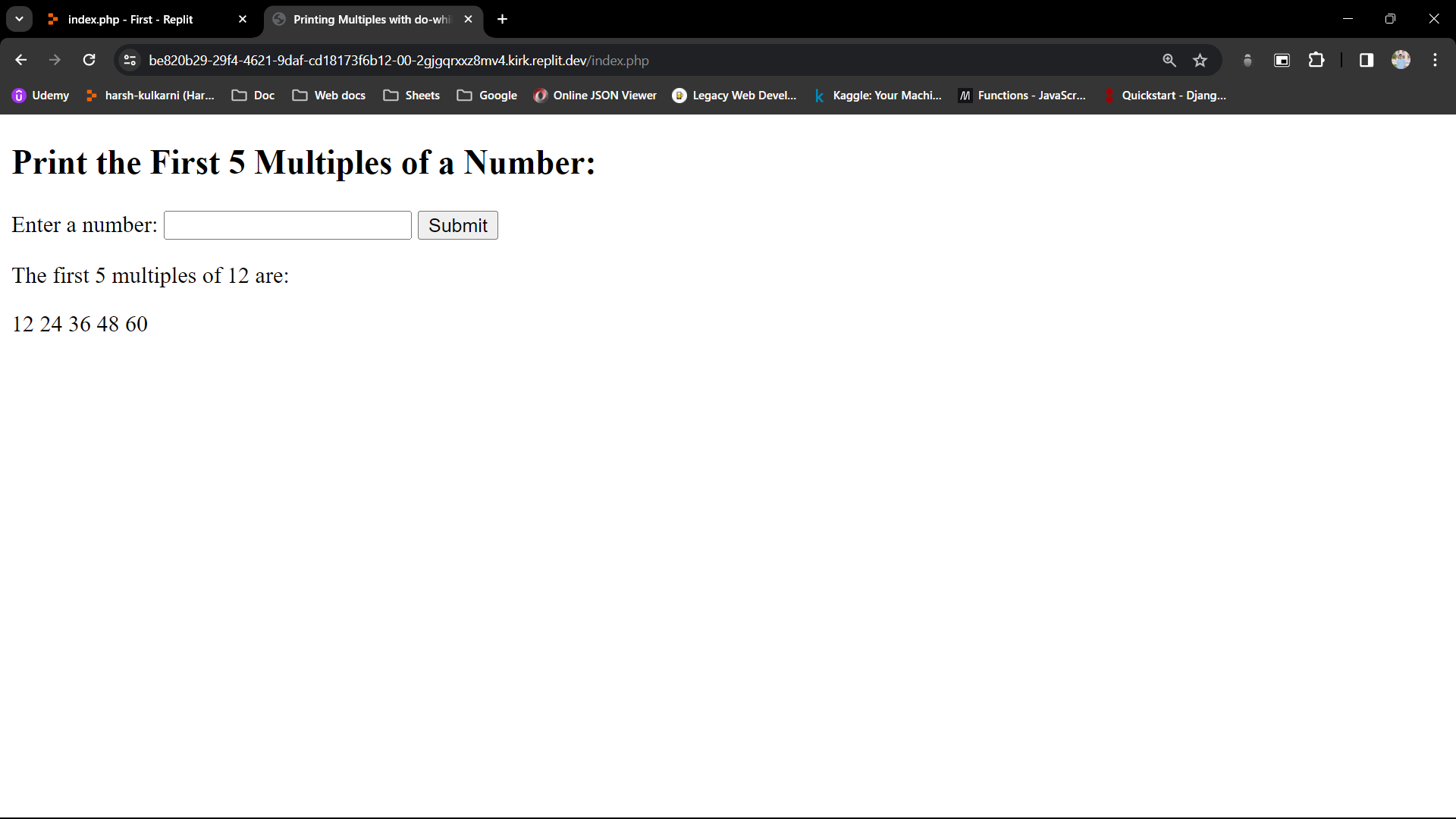
</body>

</html>

**Output:**







Assignment No: - 4

* **Practice using switch statements in PHP.**

Task:

Write a PHP script that takes a day of the week (e.g., Monday, Tuesday, etc.) as input.

Use a switch statement to display a message based on the input day.

Include cases for weekdays and a default case for weekends.

**Code:**

<!DOCTYPE html>

<html>

<head>

<title>Day of the Week Message</title>

</head>

<body>

<h2>Day of the Week Message</h2>

<form method="post" action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]);?>">

Enter a day of the week: <input type="text" name="day">

<input type="submit" name="submit" value="Submit">

</form>

<?php

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

if (isset($\_POST["day"]) && !empty($\_POST["day"])) {

$day = $\_POST["day"]; // Retrieve the day input

switch (strtolower($day)) { // Convert the input to lowercase for case-insensitive comparison

case 'monday':

case 'tuesday':

case 'wednesday':

case 'thursday':

case 'friday':

echo "<p>It's a weekday. Enjoy your work!</p>";

break;

case 'saturday':

case 'sunday':

echo "<p>It's a weekend. Relax and have fun!</p>";

break;

default:

echo "<p>Invalid input. Please enter a valid day of the week.</p>";

}

} else {

echo "<p>Please enter a day of the week.</p>";

}

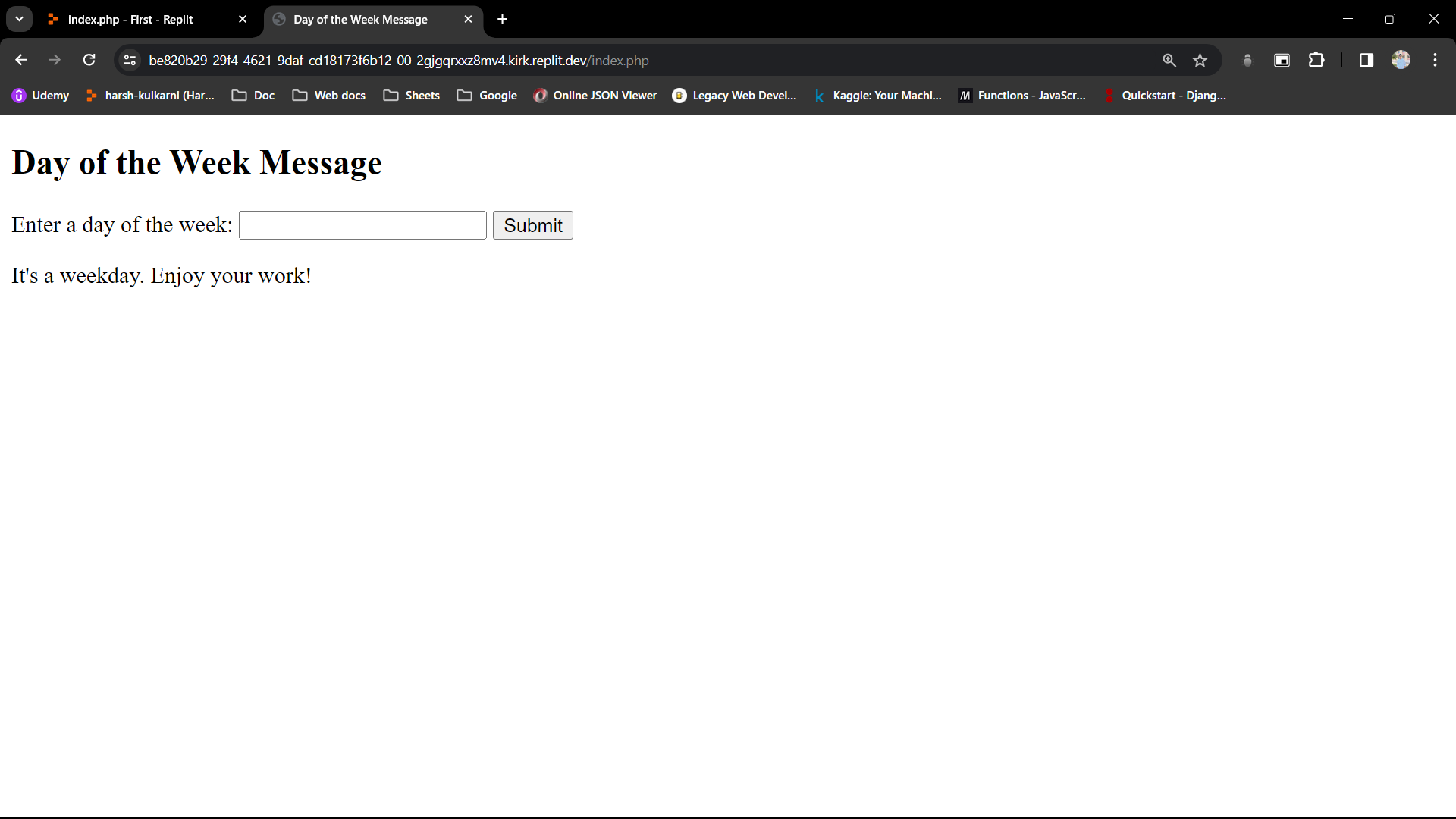
}

?>

</body>

</html>

**Output:**

****

Assignment No: - 5

* **Understand the concept of functions, declaration, definition, and function calling in PHP.**

Task:

Create a PHP function named **greet()** that takes a name as a parameter and prints a greeting message.

Call the **greet()** function with different names as arguments.

Explore the scope of variables inside and outside the function.

**Code:**

<!DOCTYPE html>

<html>

<head>

<title>Greeting</title>

</head>

<body>

<h2>Greeting</h2>

<?php

// Define the greet() function

function greet($name) {

// Variable inside the function

$greeting = "Hello, $name! How are you?";

echo $greeting . "<br>";

}

// Call the greet() function with different names

greet("Alice");

greet("Bob");

// Attempting to access the $greeting variable outside the function will result in an error

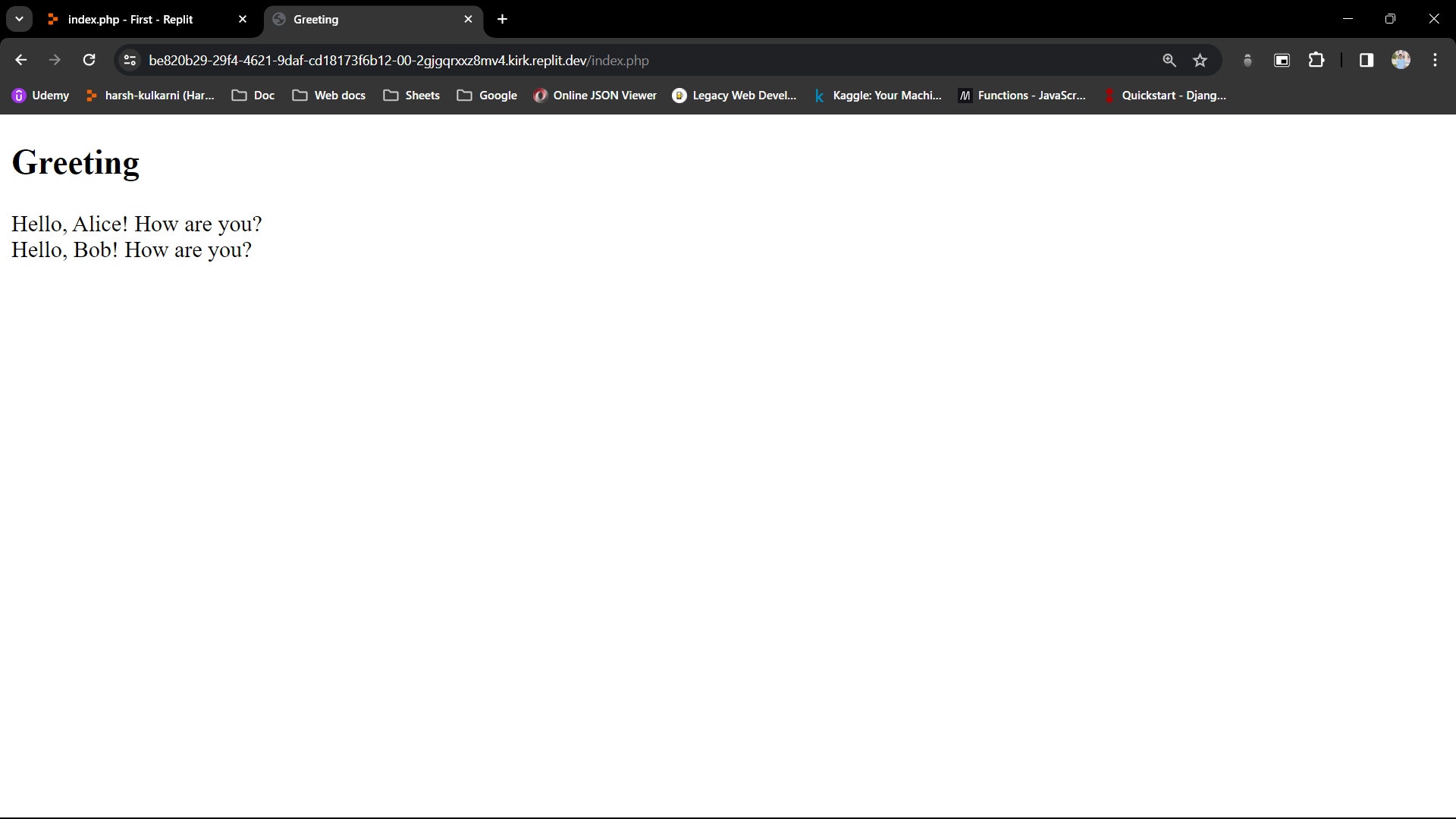
// echo $greeting; // Uncommenting this line will result in an "Undefined variable" error

?>

</body>

</html>

**Output:**

****

Assignment No: - 6

* **Explore user-defined functions and understand variable scope in PHP.**

Task:

Create a PHP function named **calculateSum()** that calculates the sum of two numbers passed as parameters.

Call the **calculateSum()** function with different numbers as arguments.

Attempt to access variables declared inside the function from outside and vice versa to understand variable scope.

**Code:**

<!DOCTYPE html>

<html>

<head>

<title>Sum Calculation</title>

</head>

<body>

<h2>Sum Calculation</h2>

<?php

// Define the calculateSum() function

function calculateSum($num1, $num2) {

// Variable inside the function

$sum = $num1 + $num2;

echo "Sum inside function: $sum<br>";

return $sum;

}

// Call the calculateSum() function with different numbers

$result1 = calculateSum(5, 3);

$result2 = calculateSum(10, 20);

// Attempting to access the $sum variable outside the function will result in an error

// echo "Sum outside function: $sum<br>"; // Uncommenting this line will result in an "Undefined variable" error

// Attempting to access variables declared outside the function inside the function

$num3 = 15;

$num4 = 25;

$sumOutside = calculateSum($num3, $num4);

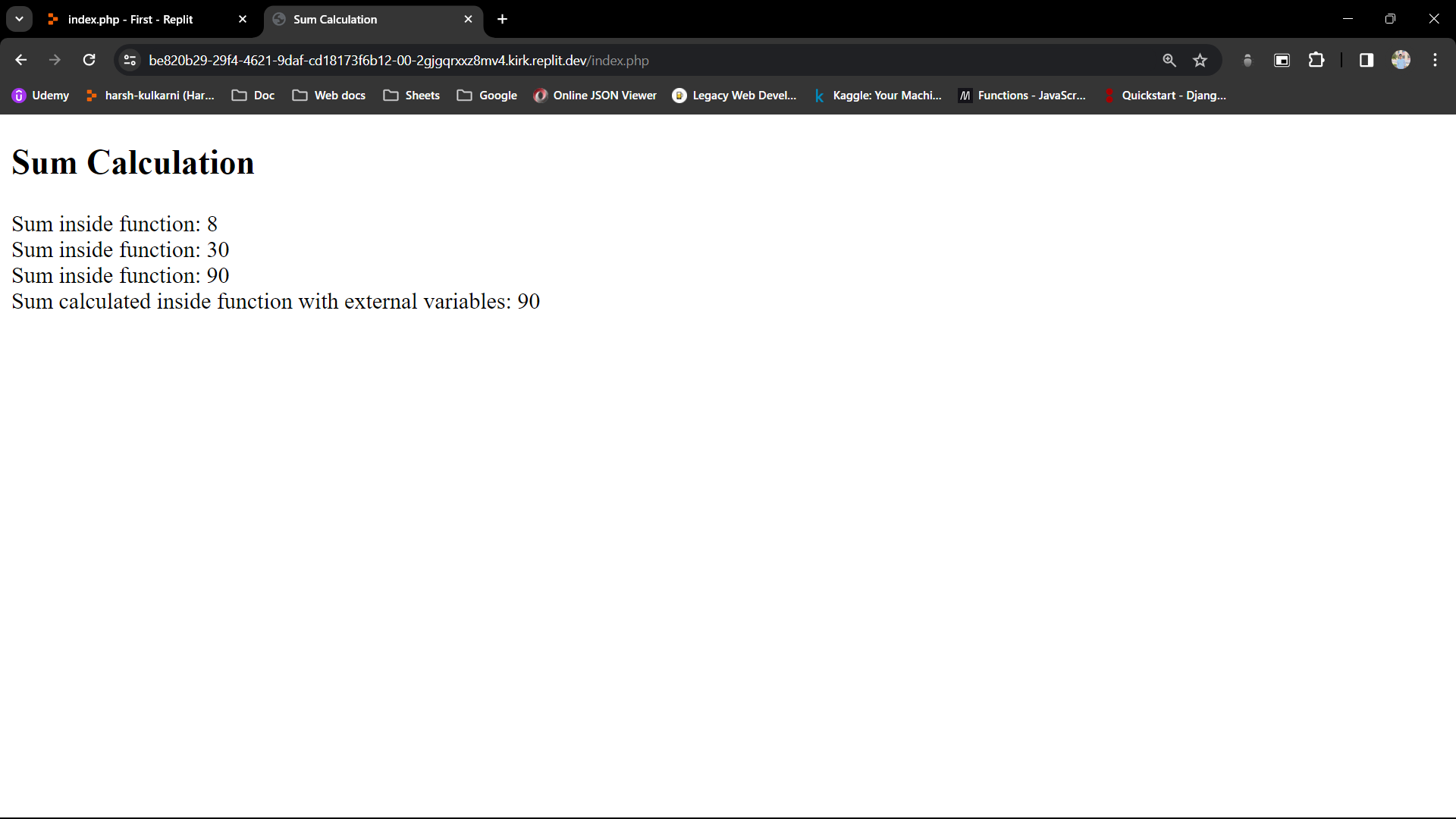
echo "Sum calculated inside function with external variables: $sumOutside<br>";

?>

</body>

</html>

**Output:**



Assignment No: - 7

* **Practice sorting and reordering arrays in PHP.**

Task:

Create a PHP script that declares an array of numbers.

Sort the array in ascending and descending order using array functions.

Reorder the array elements randomly.

**Code:**

<!DOCTYPE html>

<html>

<head>

<title>Array Manipulation</title>

</head>

<body>

<h2>Array Manipulation</h2>

<?php

// Declare an array of numbers

$numbers = array(5, 3, 9, 2, 7, 1, 8, 4, 6);

// Sort the array in ascending order

sort($numbers);

echo "<p>Sorted array in ascending order: ";

foreach ($numbers as $num) {

echo "$num ";

}

echo "</p>";

// Sort the array in descending order

rsort($numbers);

echo "<p>Sorted array in descending order: ";

foreach ($numbers as $num) {

echo "$num ";

}

echo "</p>";

// Reorder the array elements randomly

shuffle($numbers);

echo "<p>Array elements after shuffling: ";

foreach ($numbers as $num) {

echo "$num ";

}

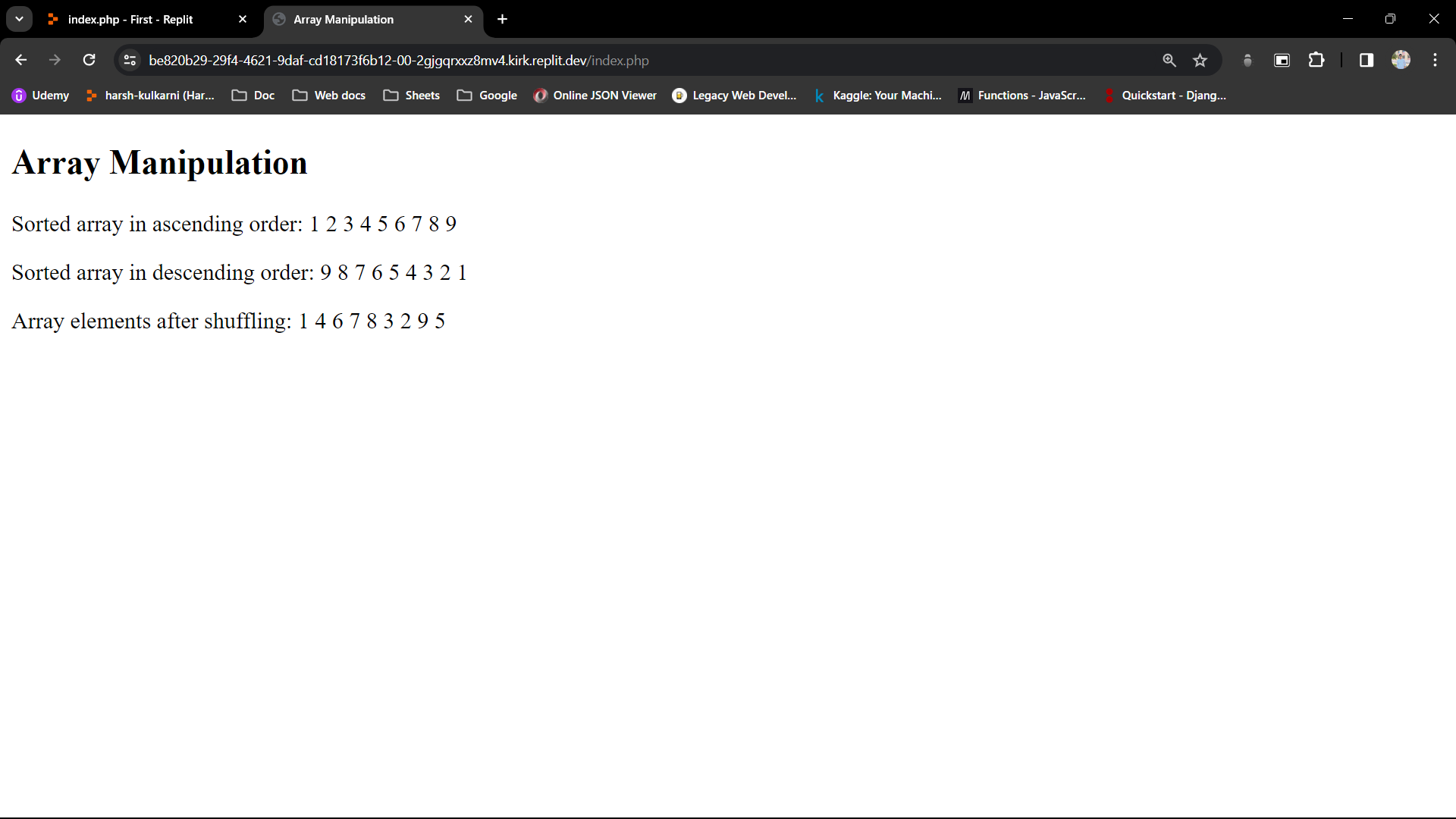
echo "</p>";

?>

</body>

</html>

**Output:**



Assignment No: - 8

* **Understand and utilize PHP's date and time functions.**

Task:

Create a PHP script that displays the current date and time.

Convert a specific date and time format to another format.

Calculate and display the difference between two dates.

**Code:**

<!DOCTYPE html>

<html>

<head>

<title>Date and Time Operations</title>

</head>

<body>

<h2>Current Date and Time</h2>

<?php

// Display the current date and time

$currentDateTime = date("Y-m-d H:i:s");

echo "<p>Current Date and Time: $currentDateTime</p>";

// Convert a specific date and time format to another format

$specificDateTime = "2022-01-15 14:30:00";

$formattedDateTime = date("F j, Y, g:i a", strtotime($specificDateTime));

echo "<p>Formatted Date and Time: $formattedDateTime</p>";

// Calculate and display the difference between two dates

$date1 = strtotime("2023-05-20");

$date2 = strtotime("2023-03-10");

$difference = abs($date1 - $date2);

$daysDifference = floor($difference / (60 \* 60 \* 24));

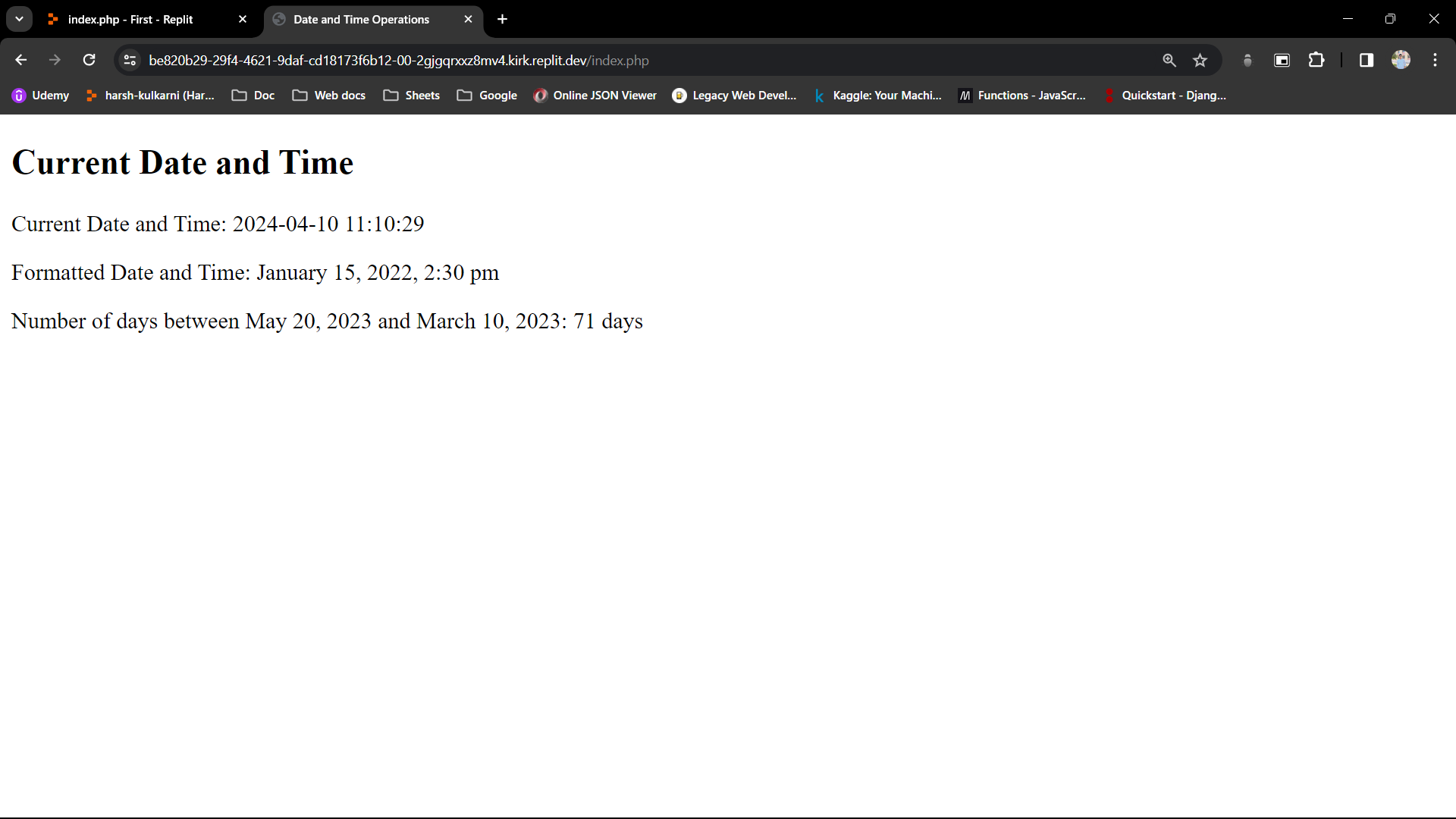
echo "<p>Number of days between May 20, 2023 and March 10, 2023: $daysDifference days</p>";

?>

</body>

</html>

**Output:**



Assignment No: - 9

* **Learn to handle file operations in PHP, including saving and retrieving data.**

Task:

Create an HTML form to collect Bob's order details such as product name, quantity, and price.

Upon submission, save the order details to a text file named "orders.txt".

Implement a PHP script to read and display Bob's order details from the file.

**Code:**

**// Index.php**

<!DOCTYPE html>

<html>

<head>

<title>Bob's Order Form</title>

</head>

<body>

<h2>Bob's Order Form</h2>

<form action="save\_order.php" method="post">

<label for="product\_name">Product Name:</label><br>

<input type="text" id="product\_name" name="product\_name"><br><br>

<label for="quantity">Quantity:</label><br>

<input type="number" id="quantity" name="quantity"><br><br>

<label for="price">Price:</label><br>

<input type="number" id="price" name="price" step="0.01"><br><br>

<input type="submit" value="Submit">

</form>

</body>

</html>

**// save\_order.php**

<?php

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

// Collect order details

$product\_name = $\_POST["product\_name"];

$quantity = $\_POST["quantity"];

$price = $\_POST["price"];

// Format order details

$order\_details = "Product Name: $product\_name, Quantity: $quantity, Price: $price";

// Save order details to a text file named "orders.txt"

$file = fopen("orders.txt", "a");

fwrite($file, $order\_details . "\n");

fclose($file);

echo "<p>Order details saved successfully!</p>";

} else {

echo "<p>Invalid request!</p>";

}

?>

**// read\_order.php**

<!DOCTYPE html>

<html>

<head>

<title>Bob's Orders</title>

</head>

<body>

<h2>Bob's Orders</h2>

<?php

// Read and display Bob's order details from the file

$orders\_file = "orders.txt";

if (file\_exists($orders\_file)) {

$orders = file($orders\_file);

if (!empty($orders)) {

echo "<ul>";

foreach ($orders as $order) {

echo "<li>$order</li>";

}

echo "</ul>";

} else {

echo "<p>No orders found.</p>";

}

} else {

echo "<p>Orders file not found.</p>";

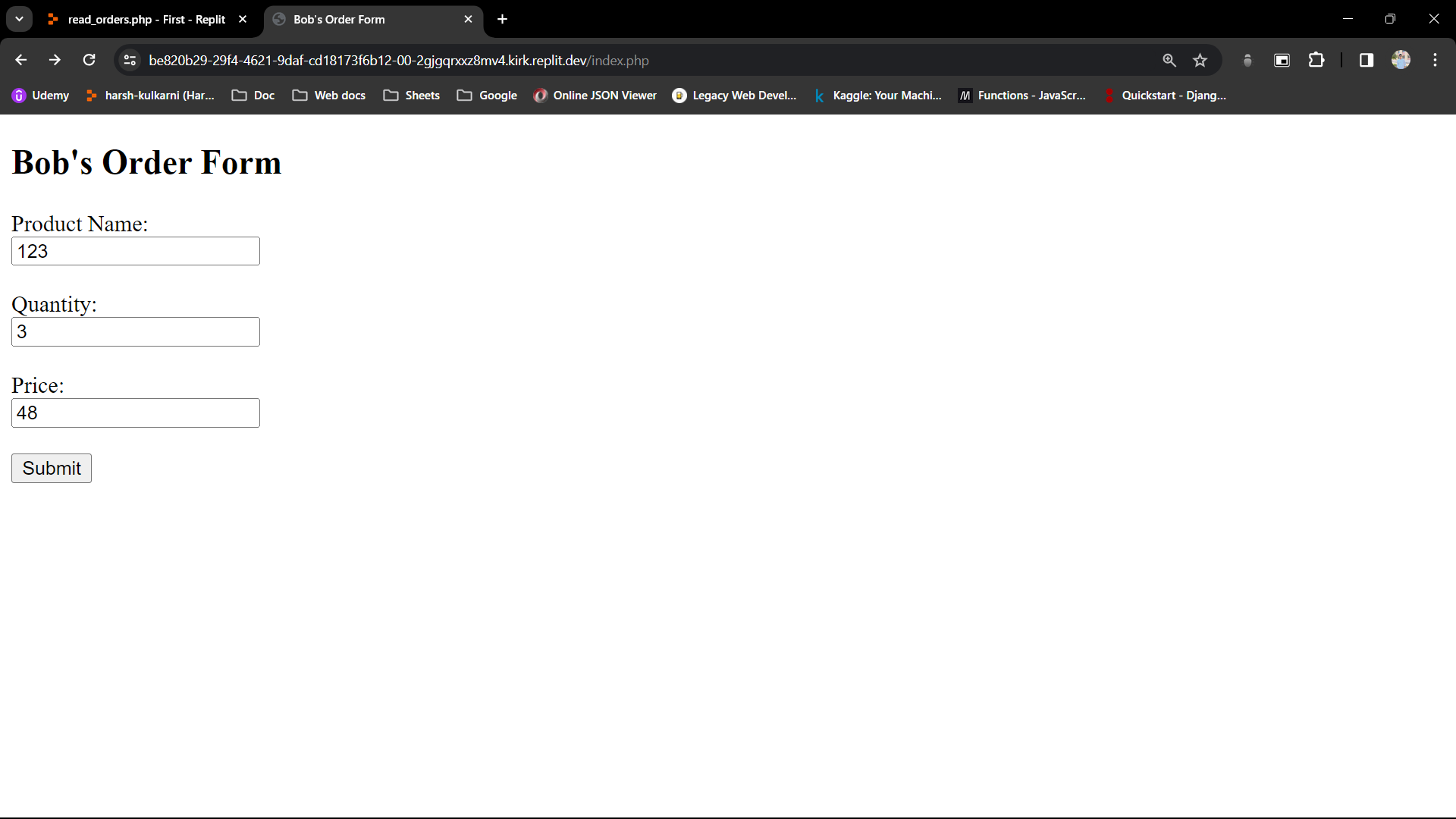
}

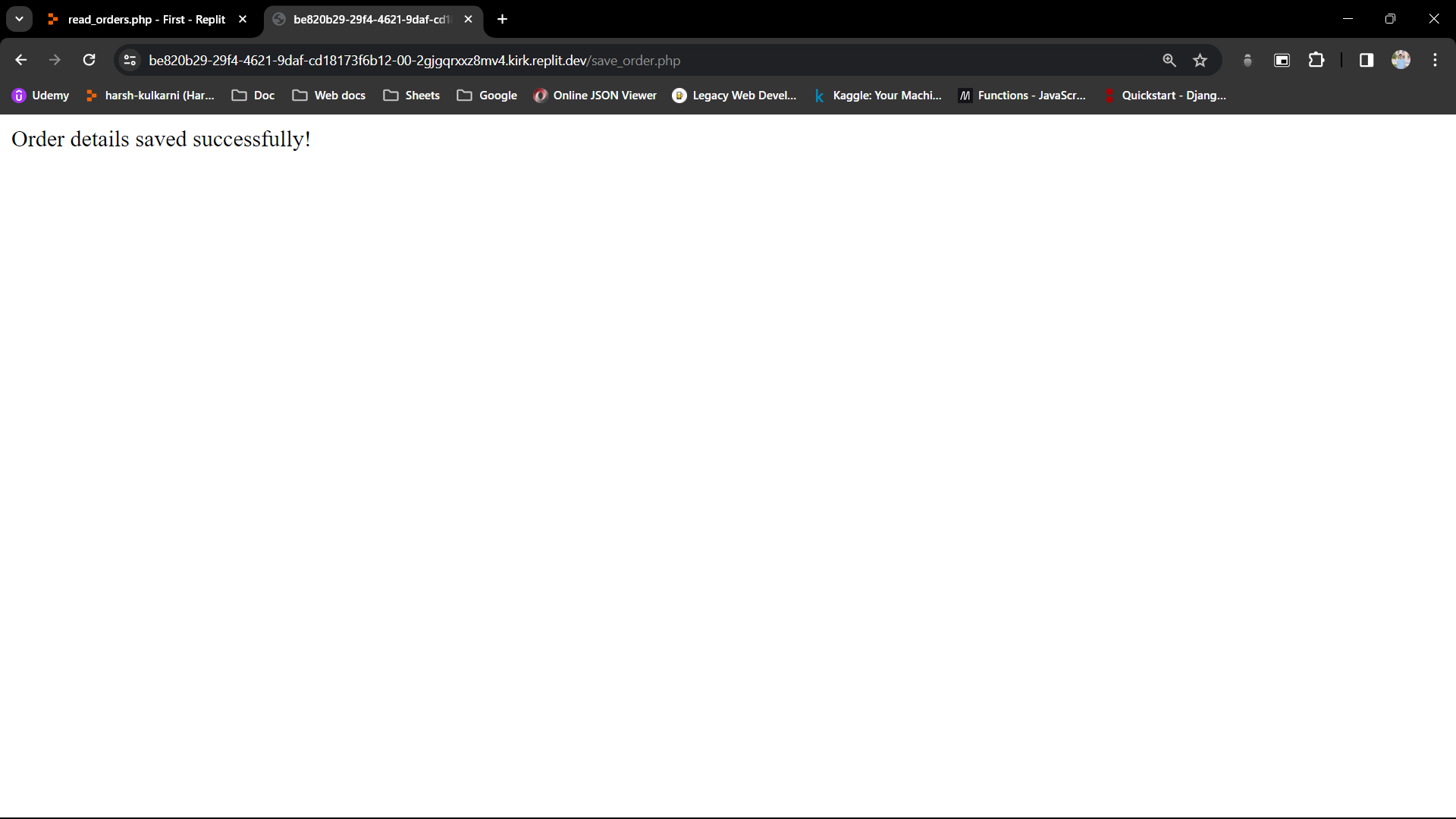
?>

</body>

</html>

**Output:**





// order.txt

Product Name: 123, Quantity: 3, Price: 48

Assignment No: - 10

* **Understand the basics of inserting and retrieving data in MySQL.**

Task:

Create a table named "users" within the "web\_database" to store user information (e.g., id, name, email).

Insert several records into the "users" table.

Retrieve all data from the "users" table using a SELECT query.

**Code:**

<?php

try {

$db = new PDO('sqlite:database.sqlite');

$db->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$res = $db->exec(

"CREATE TABLE users (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(255) NOT NULL,

email VARCHAR(255) NOT NULL

)"

);

$stmt = $db->prepare(

"INSERT INTO users (name, email) VALUES

('John Doe', 'john@example.com'),

('Jane Smith', 'jane@example.com'),

('Alice Johnson', 'alice@example.com')"

);

// // Bind values directly to statement variables

// $stmt->bindValue(':title', 'message title', SQLITE3\_TEXT);

// $stmt->bindValue(':message', 'message body', SQLITE3\_TEXT);

// // Format unix time to timestamp

// $formatted\_time = date('Y-m-d H:i:s');

// $stmt->bindValue(':time', $formatted\_time, SQLITE3\_TEXT);

// Execute statement

$stmt->execute();

$messages = $db->query("SELECT \* FROM users");

// Garbage collect db

$db = null;

} catch (PDOException $ex) {

echo $ex->getMessage();

}

?>

<html>

<head>

<title>PHP Test</title>

</head>

<body>

<?= '<h1>Messages</h1>'; ?>

<?php foreach ($messages as $msg) {

echo '<p>';

echo '<h4>' . $msg['name'] . '</h4>';

echo $msg['email'];

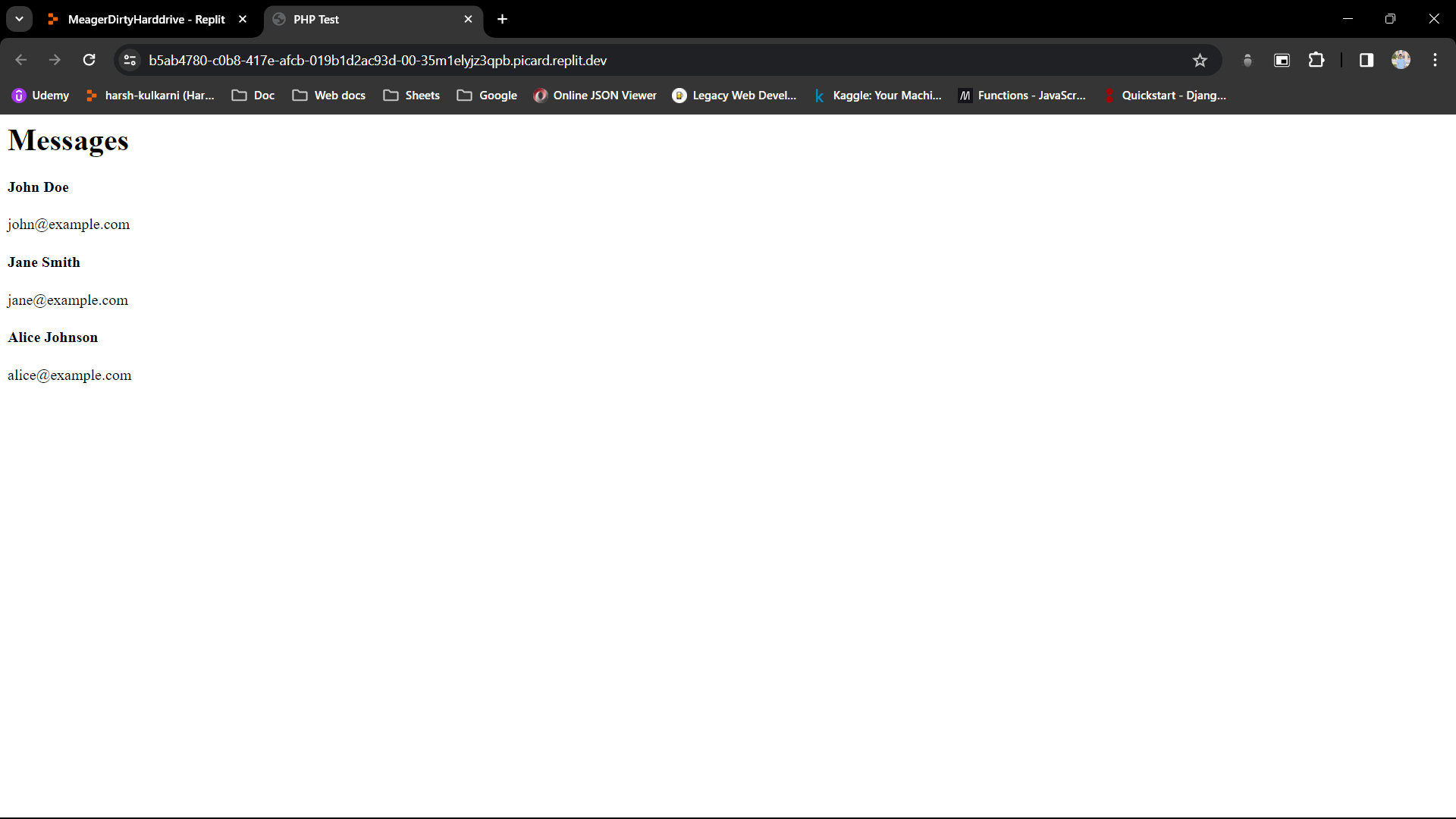
echo '</p>';

} ?>

</body>

</html>

**Output:**

****

Assignment No: - 11

* **Understand how to retrieve data from multiple tables using joins.**

Task:

Create a new table named "orders" to store user orders (e.g., order\_id, user\_id, product\_id, quantity).

Insert sample order data into the "orders" table.

Write a SQL query to retrieve user orders along with user information by joining the "users" and "orders" tables.

Display the results.

**Code:**

**// Database**

-- the "users" table

CREATE TABLE users (

user\_id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(255) NOT NULL,

email VARCHAR(255) NOT NULL

);

-- the "products" table

CREATE TABLE products (

product\_id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(255) NOT NULL,

price DECIMAL(10, 2) NOT NULL

)

-- the "orders" table

CREATE TABLE orders (

order\_id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT,

product\_id INT,

quantity INT,

FOREIGN KEY (user\_id) REFERENCES users(user\_id),

FOREIGN KEY (product\_id) REFERENCES products(product\_id)

);

-- Insert sample data into the "users" table

INSERT INTO users (name, email) VALUES

('John Doe', 'john@example.com'),

('Jane Smith', 'jane@example.com'),

('Alice Johnson', 'alice@example.com');

-- Insert sample data into the "products" table

INSERT INTO products (name, price) VALUES

('Product A', 10.00),

('Product B', 20.00),

('Product C', 30.00);

-- Insert sample data into the "orders" table

INSERT INTO orders (user\_id, product\_id, quantity) VALUES

(1, 1, 2),

(2, 2, 1),

(3, 3, 3);

-- retrieve user orders along with user information

SELECT u.name AS user\_name, u.email, p.name AS product\_name, o.quantity

FROM orders o

JOIN users u ON o.user\_id = u.user\_id

JOIN products p ON o.product\_id = p.product\_id;

**// PHP code**

<?php

// Database connection parameters

$servername = "localhost";

$username = "root";

$password = "password";

$database = "web\_database";

// Create connection

$conn = new mysqli($servername, $username, $password, $database);

// Check connection

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

// Write SQL query to retrieve user orders along with user information by joining the "users" and "orders" tables

$sql = "SELECT users.name AS user\_name, users.email, orders.order\_id, orders.product\_id, orders.quantity

FROM orders

JOIN users ON orders.user\_id = users.id";

// Execute the query

$result = $conn->query($sql);

// Check if there are any results

if ($result->num\_rows > 0) {

// Output data of each row

echo "<h2>User Orders:</h2>";

echo "<table border='1'><tr><th>User Name</th><th>Email</th><th>Order ID</th><th>Product ID</th><th>Quantity</th></tr>";

while($row = $result->fetch\_assoc()) {

echo "<tr><td>" . $row["user\_name"]. "</td><td>" . $row["email"]. "</td><td>" . $row["order\_id"]. "</td><td>" . $row["product\_id"]. "</td><td>" . $row["quantity"]. "</td></tr>";

}

echo "</table>";

} else {

echo "0 results";

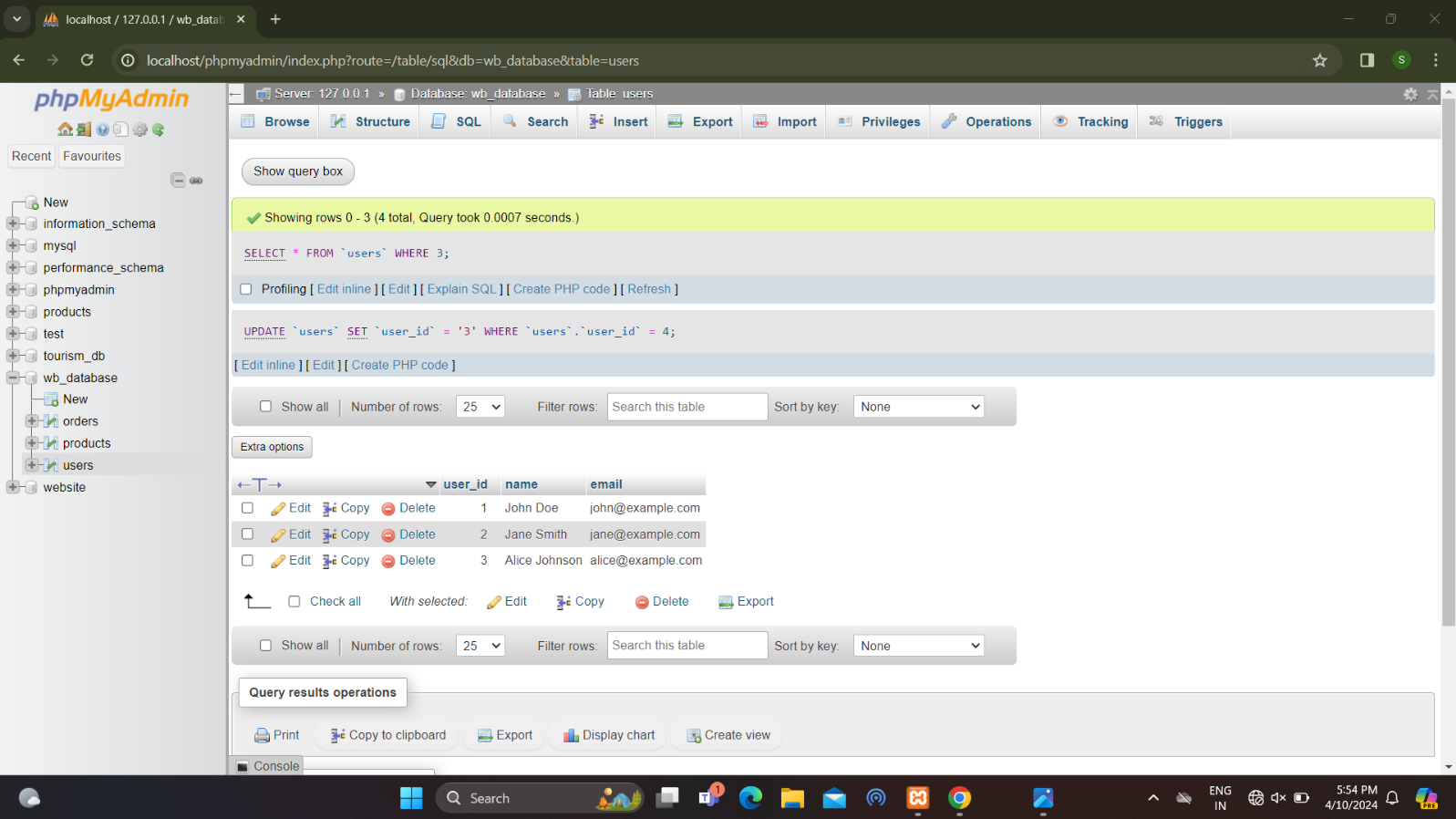
}

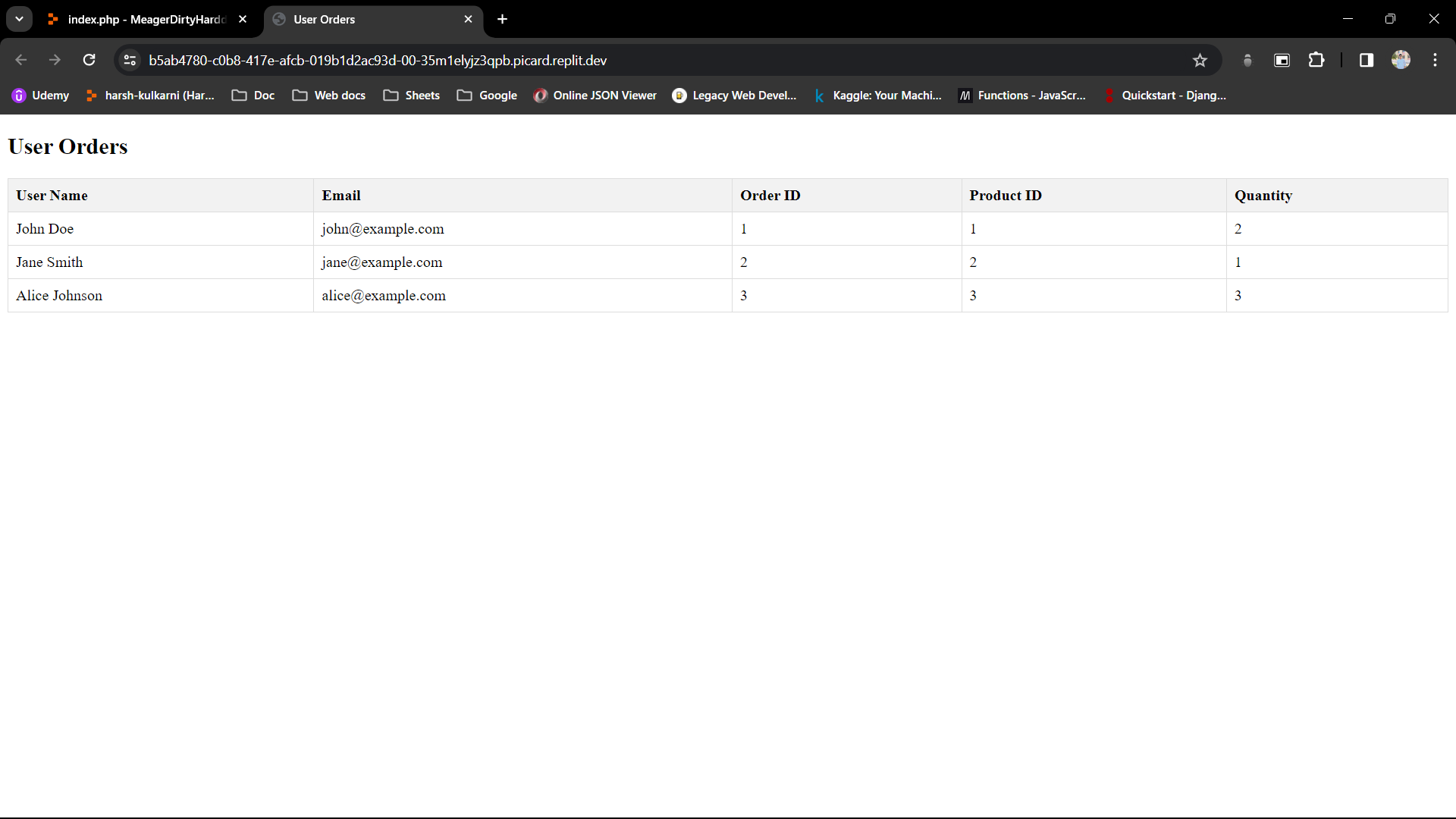
// Close connection

$conn->close();

?>

**Output:**





Assignment No: - 12

* **Practice updating and deleting records in MySQL.**

Task:

Write SQL queries to update the email address of a specific user and delete a user record from the "users" table.

Verify that the changes have been applied correctly.

**Code:**

<?php

// Database connection parameters

$servername = "localhost";

$username = "root";

$password = "password";

$database = "web\_database";

try {

// Create connection

$conn = new PDO("mysql:host=$servername;dbname=$database", $username, $password);

// Set the PDO error mode to exception

$conn->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

// SQL query to update the email address of a specific user

$update\_sql = "UPDATE users

SET email = 'new\_email@example.com'

WHERE user\_id = 1"; // Specify the user\_id of the user whose email you want to update

// Execute the update query

$conn->exec($update\_sql);

// SQL query to delete a user record from the "users" table

$delete\_sql = "DELETE FROM users

WHERE user\_id = 2"; // Specify the user\_id of the user you want to delete

// Execute the delete query

$conn->exec($delete\_sql);

// SQL query to verify the updated email address

$verify\_update\_sql = "SELECT \* FROM users WHERE user\_id = 1";

$stmt = $conn->query($verify\_update\_sql);

$updated\_user = $stmt->fetch(PDO::FETCH\_ASSOC);

// SQL query to verify if the user record has been deleted

$verify\_delete\_sql = "SELECT \* FROM users WHERE user\_id = 2";

$stmt = $conn->query($verify\_delete\_sql);

$deleted\_user = $stmt->fetch(PDO::FETCH\_ASSOC);

// Display verification results

echo "<h2>Verification Results:</h2>";

echo "<p>Updated User:</p>";

print\_r($updated\_user);

echo "<p>Deleted User:</p>";

print\_r($deleted\_user);

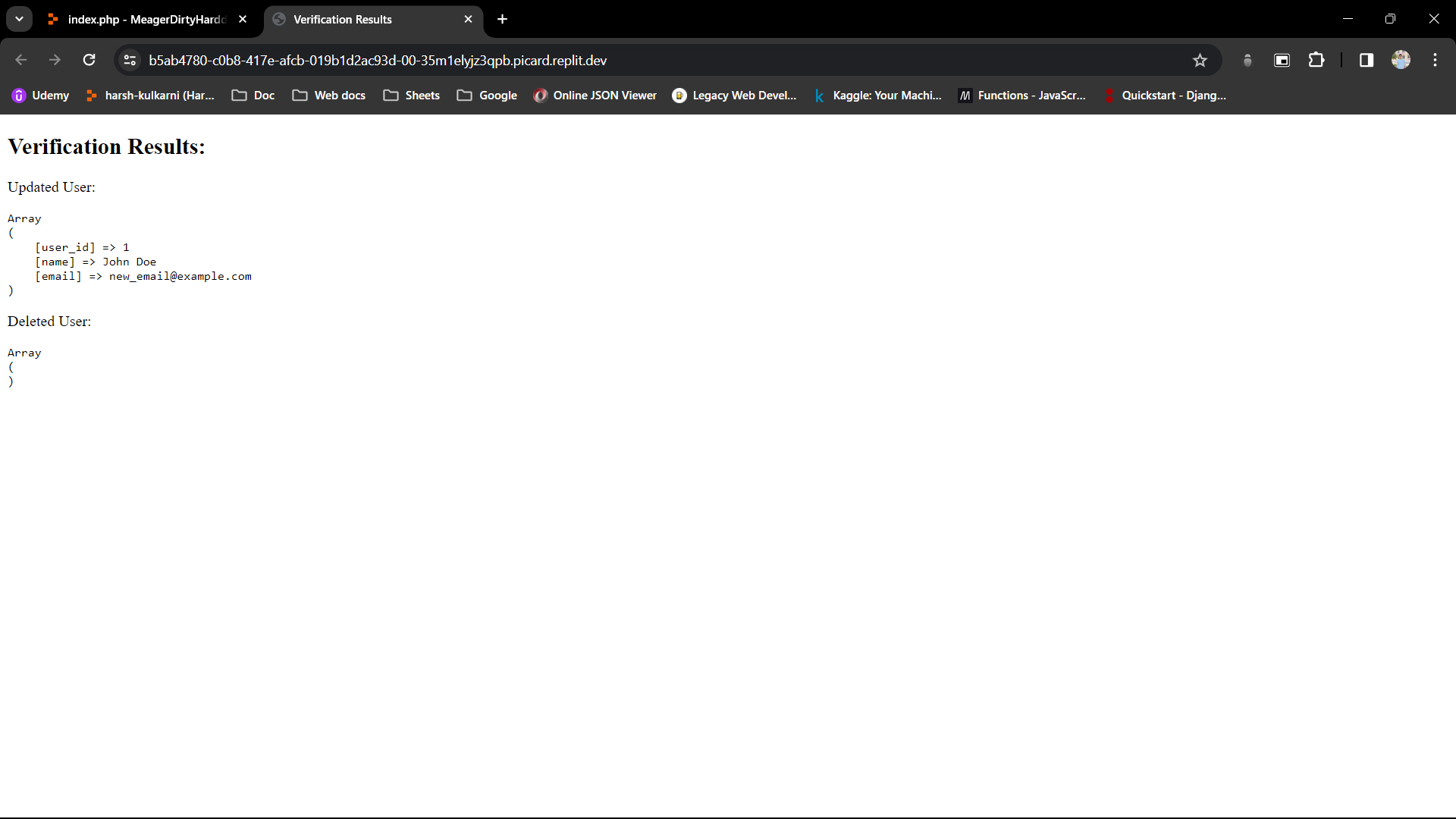
} catch (PDOException $e) {

echo "Connection failed: " . $e->getMessage();

}

?>

**Output:**

****