**ASSIGNMENT NO:.1**

**Que.:** 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.

<!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>";

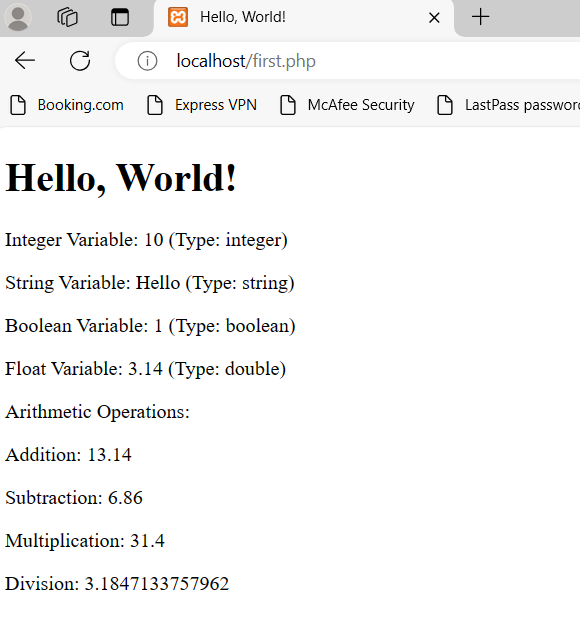
echo "<p>Division: $resultDivision</p>";

?>

</body>

</html>

**Output:**



**ASSIGNMENT NO:.2**

**Que**. 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.

<?php

// Function to sanitize user input

function sanitizeInput($input) {

// Remove whitespace from the beginning and end of input

$input = trim($input);

// Remove slashes

$input = stripslashes($input);

// Convert special characters to HTML entities

$input = htmlspecialchars($input);

return $input;

}

// Check if the form is submitted

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

// Get the input number and sanitize it

$inputNumber = sanitizeInput($\_POST["number"]);

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

if ($inputNumber > 0) {

$message = "The number $inputNumber is positive.";

} elseif ($inputNumber < 0) {

$message = "The number $inputNumber is negative.";

} else {

$message = "The number is zero.";

}

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Number Checker</title>

</head>

<body>

<h2>Enter a Number</h2>

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

<label for="number">Number:</label>

<input type="number" id="number" name="number" required>

<button type="submit">Check</button>

</form>

<?php

// Display the output message if it's set

if (isset($message)) {

echo "<p>$message</p>";

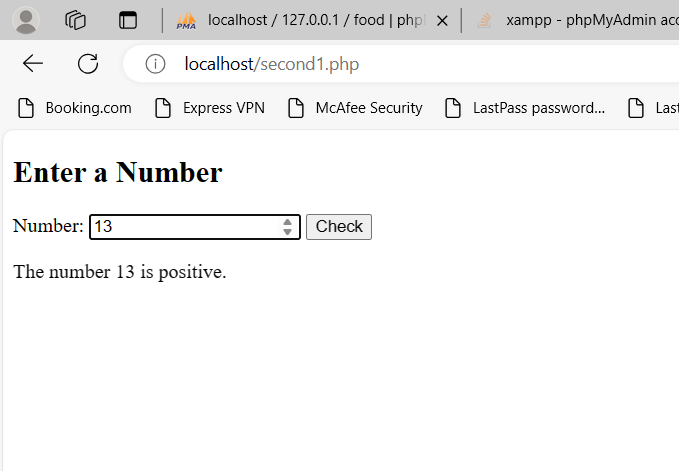
}

?>

</body>

</html>

**Output:**

: 

**ASSIGNMENT NO.:3**

Que. Understand different types of loops in PHP.Task

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

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

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

A]

<?php

echo "Prime numbers from 1 to 10: ";

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

$isPrime = true;

if ($i == 1) {

continue;

}

for ($j = 2; $j <= sqrt($i); $j++) {

if ($i % $j == 0) {

$isPrime = false;

break;

}

}

if ($isPrime) {

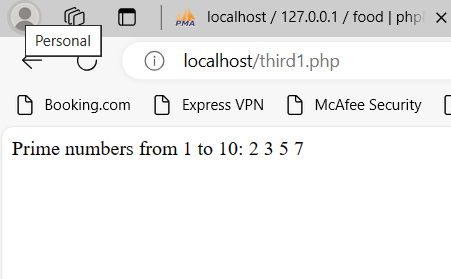
echo $i . " ";

}

}

?>

**Output:**



B]

<?php

echo "Even numbers from 1 to 12: ";

$num = 1;

while ($num <= 12) {

if ($num % 2 == 0) {

echo $num . " ";

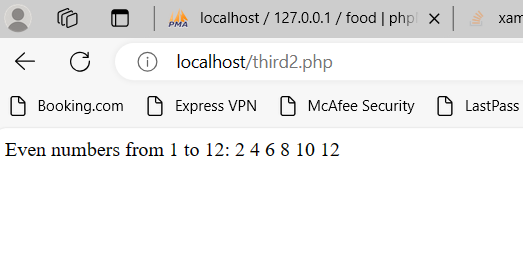
}

$num++;

}

?>

**Output:**



C]

<!DOCTYPE html>

<html>

<body>

<h2> first 5 multiples of a number</h2>

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

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

<input type="submit">

</form>

<?php

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

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

$i = 1;

$count = 0;

do {

$multiple = $num \* $i;

echo "$num x $i = $multiple<br>";

$i++;

$count++;

} while ($count < 5);

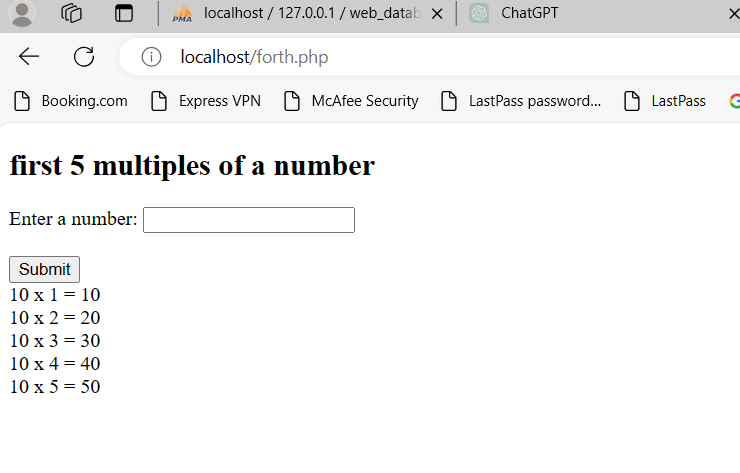
}

?>

</body>

</html>

**Output:**

****

**ASSIGNMENT NO.:4**

**Que.** 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.

<!DOCTYPE html>

<html>

<body>

<h2>PHP Switch Statement </h2>

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

Day: <input type="text" name="day">

<input type="submit">

</form>

<?php

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

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

switch ($day) {

case "Monday":

case "Tuesday":

case "Wednesday":

case "Thursday":

case "Friday":

echo "This is a weekday.";

break;

case "Saturday":

case "Sunday":

echo "This is the weekend.";

break;

default:

}

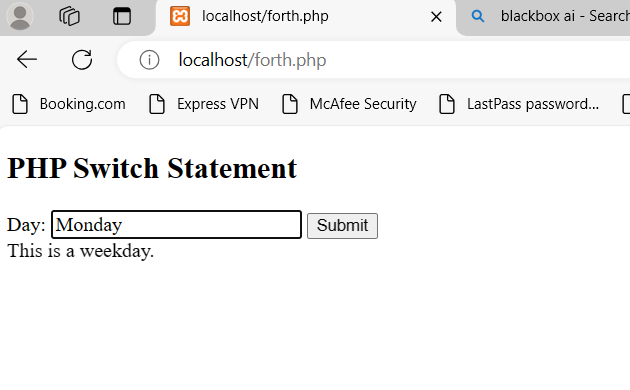
}

?>

</body>

</html>

**Output:**



**ASSIGNMENT NO.:5**

**Que**. 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.

<?php

// Define the greet() function

function greet($name) {

// Access the global $greeting variable

global $greeting;

// Print a greeting message

echo $greeting . " " . $name . "!\n";

}

// Define a global $greeting variable

$greeting = "Hello";

// Call the greet() function with different arguments

greet("Alice");

greet("Bob");

greet("Charlie");

// Change the value of the global $greeting variable

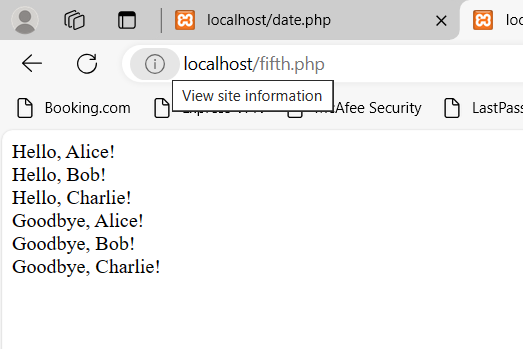
$greeting = "Hi";

// Call the greet() function again with the new value of $greeting

greet("David");

?>

**Output:**

****

**ASSIGNMENT NO.:6**

**Que.** : 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 understandvariable scope.

<?php

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

function calculateSum($num1, $num2) {

// The variables declared inside the function are only accessible within the function

$sum = $num1 + $num2;

return $sum;

}

// Call the calculateSum() function with different numbers as arguments

$result1 = calculateSum(5, 10);

$result2 = calculateSum(20, 30);

// Attempt to access variables declared inside the function from outside

// This will result in an error because variables declared inside the function are not accessible outside of it

// echo $sum;

// However, we can access the result of the function call because it returns a value

echo "The sum of 5 and 10 is: " . $result1 . "<br>";

echo "The sum of 20 and 30 is: " . $result2 . "<br>";

// Attempt to access variables declared outside the function from inside

// This will work because variables declared outside the function are accessible inside it

$external\_var = 25;

function addToExternalVar() {

global $external\_var;

$external\_var += 10;

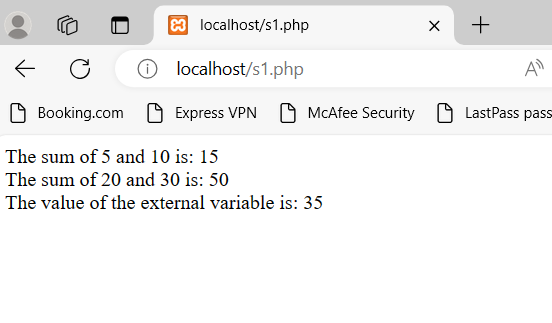
}

addToExternalVar();

echo "The value of the external variable is: " . $external\_var . "\n";

?>

**Output:**

****

**ASSIGNMENT NO.:7**

**Que:** 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.

<?php

// Declare an array of numbers

$numbers = [5, 2, 9, 1, 7, 3];

// Sorting the array in ascending order

sort($numbers);

echo "Sorted Array (Ascending Order):";

foreach ($numbers as $number) {

echo $number . " ";

}

echo "\n";

// Sorting the array in descending order

rsort($numbers);

echo "<br> Sorted Array (Descending Order):\n";

foreach ($numbers as $number) {

echo $number . " ";

}

echo "\n";

// Reordering the array elements randomly

shuffle($numbers);

echo "<br> Randomly Reordered Array:\n";

foreach ($numbers as $number) {

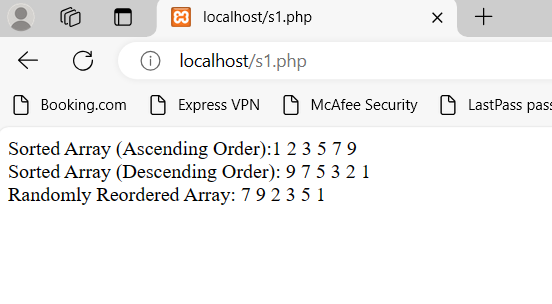
echo $number . " ";

}

echo "\n";

?>

**Output:**

****

**ASSIGNMENT NO.:8**

**Que:** 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.

<?php

// Display the current date and time

$now = new DateTime();

echo "Current date and time: " . $now->format('Y-m-d H:i:s') . "<br>";

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

$date = DateTime::createFromFormat('m/d/Y', '01/22/2023');

echo "Original date format: " . $date->format('m/d/Y') . "<br>";

echo "Converted date format: " . $date->format('F j, Y') . "<br>";

// Calculate and display the difference between two dates

$date1 = DateTime::createFromFormat('Y-m-d H:i:s', '2022-01-01 00:00:00');

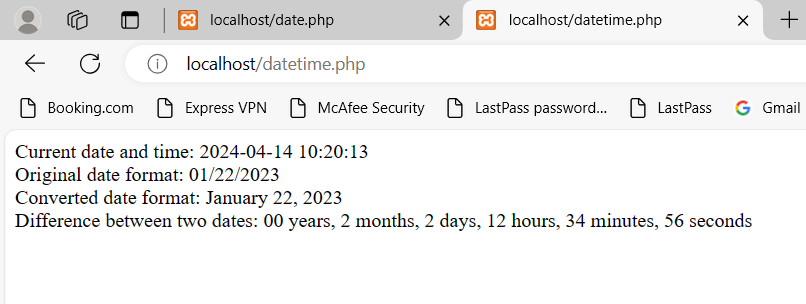
$date2 = DateTime::createFromFormat('Y-m-d H:i:s', '2022-03-03 12:34:56');

$interval = $date1->diff($date2);

echo "Difference between two dates: " . $interval->format('%Y years, %m months, %d days, %h hours, %i minutes, %s seconds') . "<br>";

?>

**Output:**

****

**ASSIGNMENT NO.:9**

**Que.** : 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.

<!DOCTYPE html>

<html>

<head>

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

</head>

<body>

<h2>Bob's Order Form</h2

<?php

// Define variables to store form input values

$productName = $quantity = $price = "";

// Define variable to store file name

$fileName = "orders.txt";

// Function to sanitize input data

function sanitize\_input($data) {

$data = trim($data);

$data = stripslashes($data);

$data = htmlspecialchars($data);

return $data;

}

// Handle form submission

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

// Sanitize and store form input values

$productName = sanitize\_input($\_POST["product\_name"]);

$quantity = sanitize\_input($\_POST["quantity"]);

$price = sanitize\_input($\_POST["price"]);

// Save order details to a text file

$orderDetails = "Product Name: $productName, Quantity: $quantity, Price: $price";

file\_put\_contents($fileName, $orderDetails . PHP\_EOL, FILE\_APPEND);

echo "Order details saved successfully.";

}

?>

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

Product Name: <input type="text" name="product\_name"><br><br>

Quantity: <input type="number" name="quantity"><br><br>

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

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

</form>

<hr>

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

<?php

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

if (file\_exists($fileName)) {

$orders = file($fileName);

foreach ($orders as $order) {

echo $order . "<br>";

}

} else {

echo "No orders yet.";

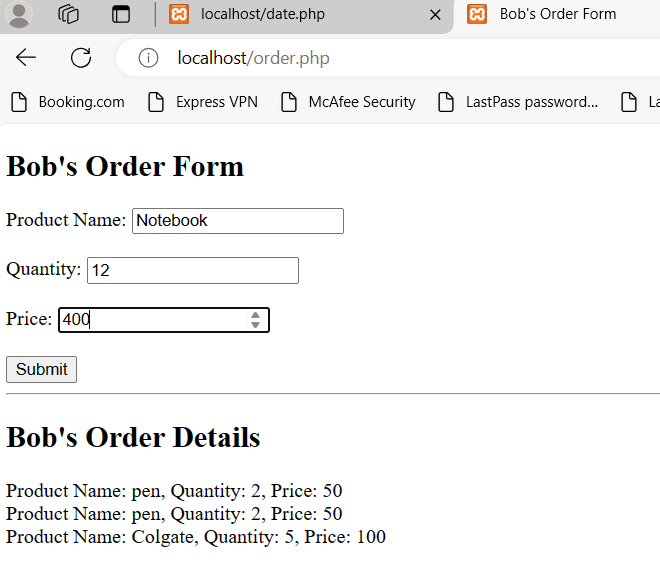
}

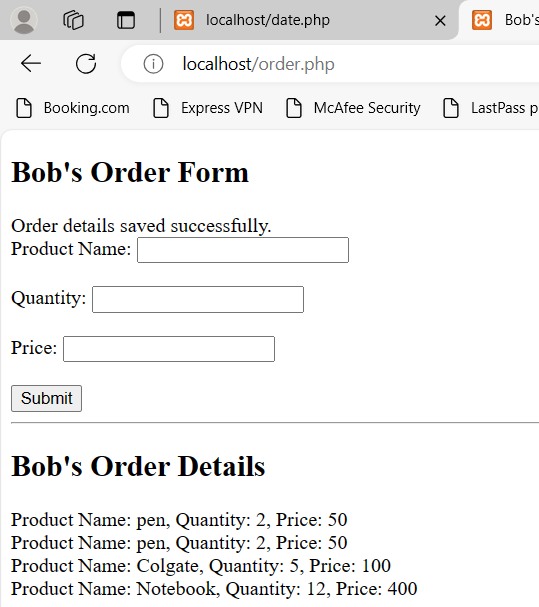
?>

</body>

</html>

**Output:**

****

****

**ASSIGNMENT NO.:10**

**Que.** Objective: 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.

1)Create Query:

CREATE TABLE web\_database.users (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(50),

email VARCHAR(100)

);

2)Insert Query:

Insert several records into the "users" table

INSERT INTO web\_database.users (name, email) VALUES

(‘sakshi’, ‘sakshi@gmail.com’),

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

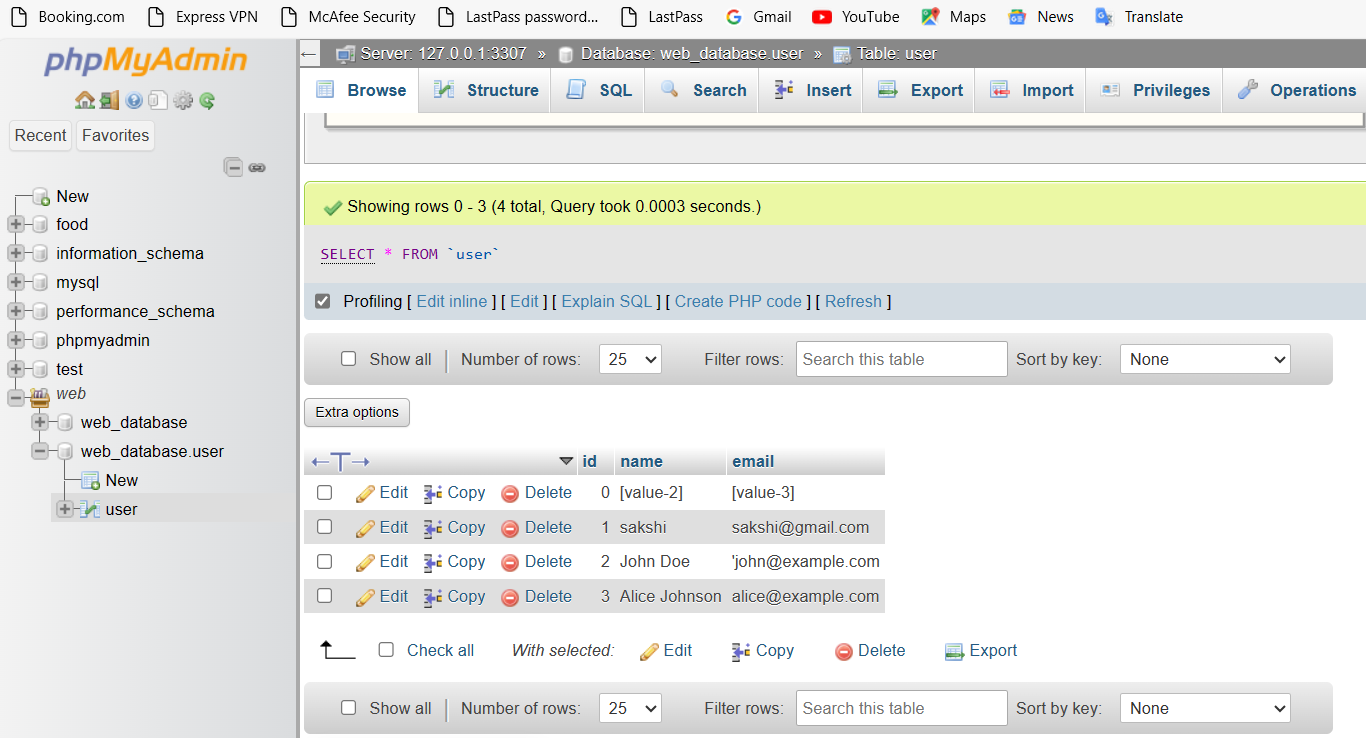
('Alice Johnson', 'alice@example.com');

3)Select Query:

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

SELECT \* FROM web\_database.users;

**Output:**

****

**ASSIGNMENT NO .:11**

**Que.** : 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.

1)Create Query:

CREATE TABLE orders (

order\_id INT PRIMARY KEY,

user\_id INT,

product\_id INT,

quantity INT,

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

);

2)Insert Query:

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

(1, 101, 201, 2),

(2, 102, 202, 1),

(3, 103, 201, 3),

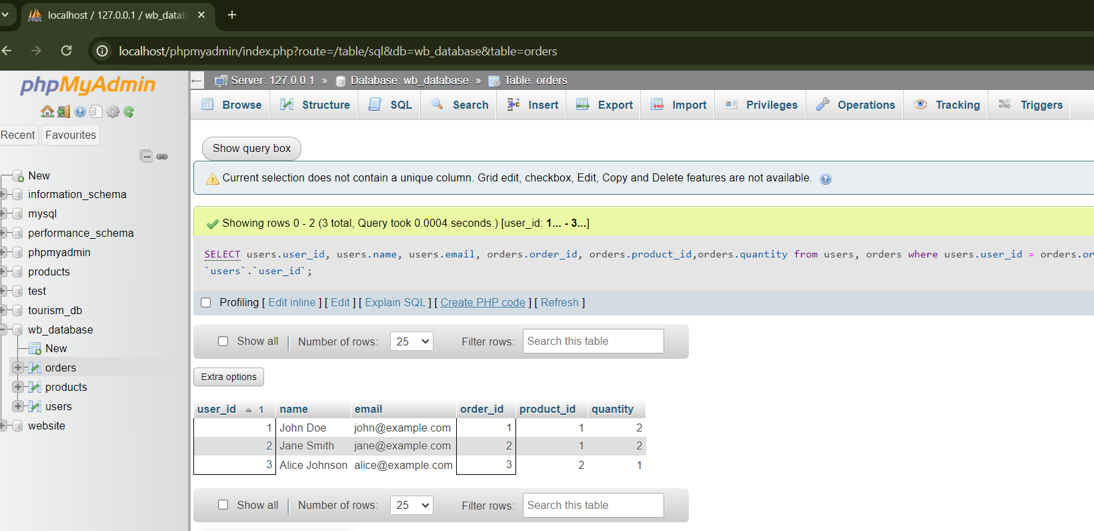
(4, 104, 203, 2),

(5, 105, 204, 1);

3) select Query:

SELECT order.order\_id, order.user\_id, user.rname, user.email, order.product\_id, oorder.quantity FROM orders o JOIN users u ON o.user\_id = u.user\_id;

**Output:**

****

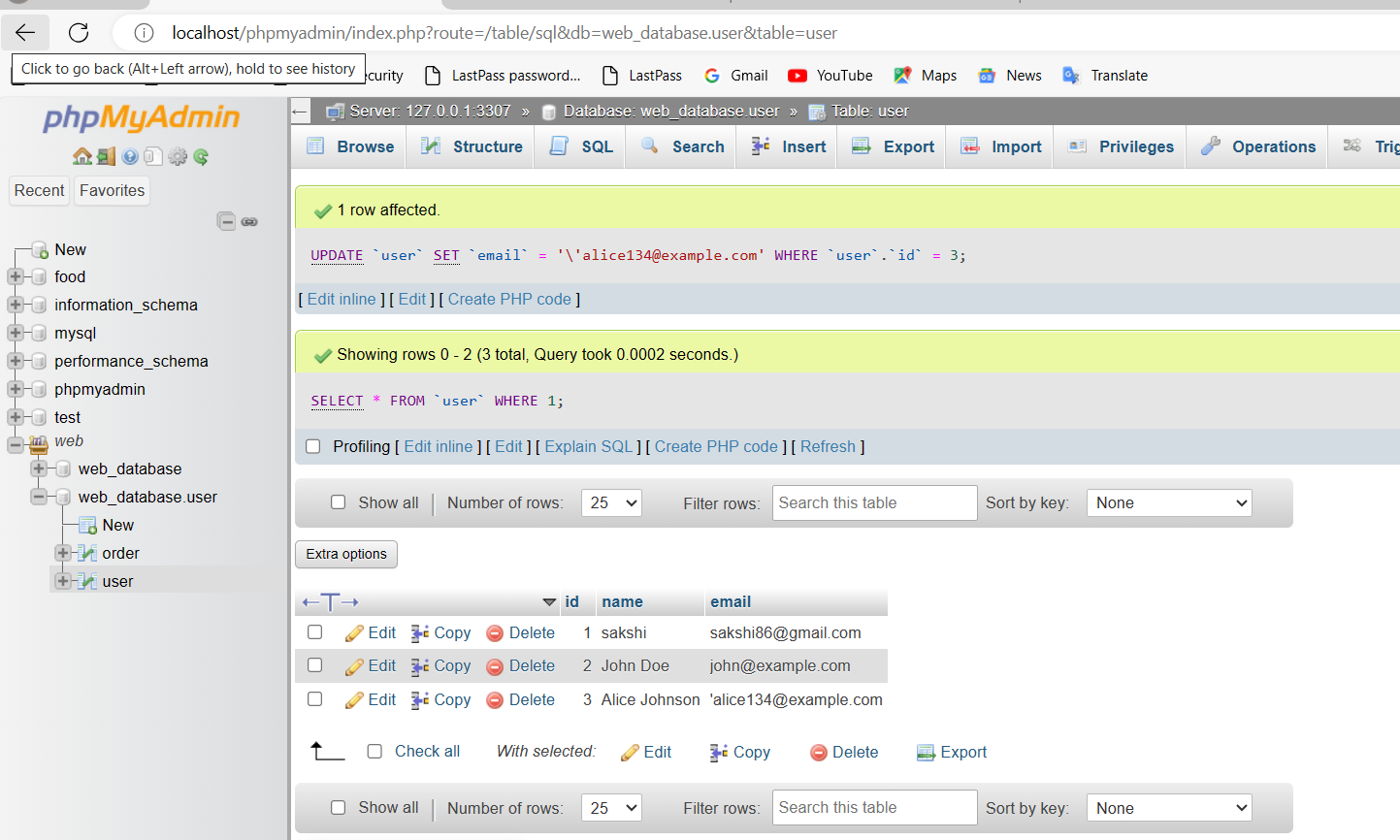
**ASSIGNMENT NO.:12**

**Que.** 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.

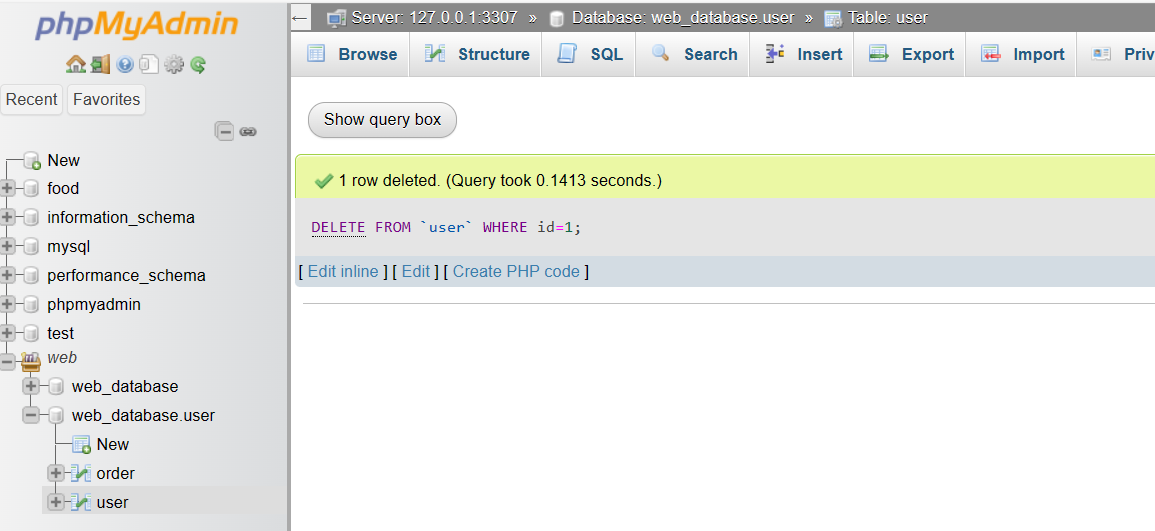
1).Update Record:

UPDATE `user` SET `email` = 'alice134@example.com' WHERE `user`.`id` = 3;



2)Delete record:

DELETE FROM `user` WHERE id=1;



3) Select:

SELECT \* FROM `user`;

