Lab/Project Report

Only for course Teacher						
		Needs Improvement	Developing	Sufficient	Above Average	Total Mark
Allocate mark & Percentage		25%	50%	75%	100%	25
Problem understanding & Analysis	7					
Implementation	8					
Report writing	10					
Total obtained mark						
Comments						

Semester: Fall 2023

Student Name: Hanium sakiba

Student ID: 213-35-771

Batch: 37th Section: E

Course Code: SE322 Course Name: Software Engineering and Web Application Lab

Course Teacher Name: Md. Ashikur Rahman

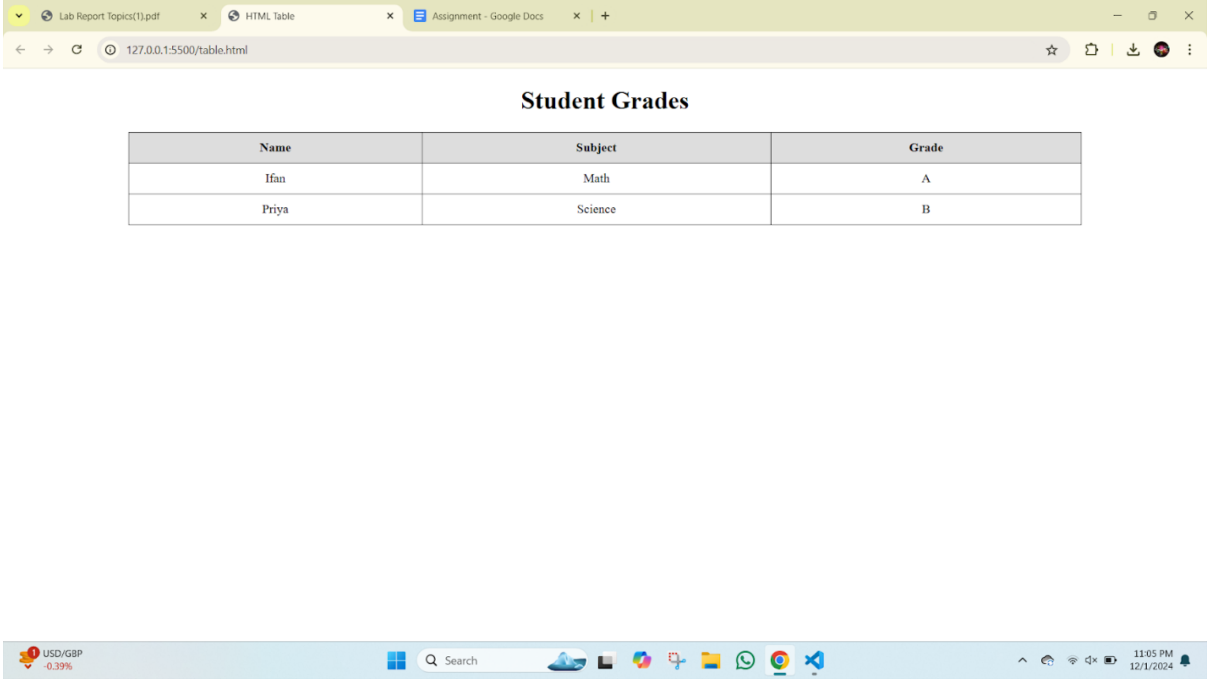
Designation: Lecture
Department of Software Engineering

Submission Date: 05 December, 2024

1. Design a table with HTML and internal CSS.
Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>HTML Table</title>
    <style>
        table {
            border-collapse: collapse;
            width: 80%;
            margin: 20px auto;
        }
        th, td {
            border: 1px solid black;
            padding: 10px;
            text-align: center;
        }
        th {
            background-color: #ddd;
        }
    </style>
</head>
<body>
    <h1 style="text-align: center;">Student Grades</h1>
    <table>
        <tr>
            <th>Name</th>
            <th>Subject</th>
            <th>Grade</th>
        </tr>
        <tr>
            <td>Ifan</td>
            <td>Math</td>
            <td>A</td>
        </tr>
        <tr>
            <td>Priya</td>
            <td>Science</td>
            <td>B</td>
        </tr>
    </table>
</body>
</html>
```

Output:



Student Grades

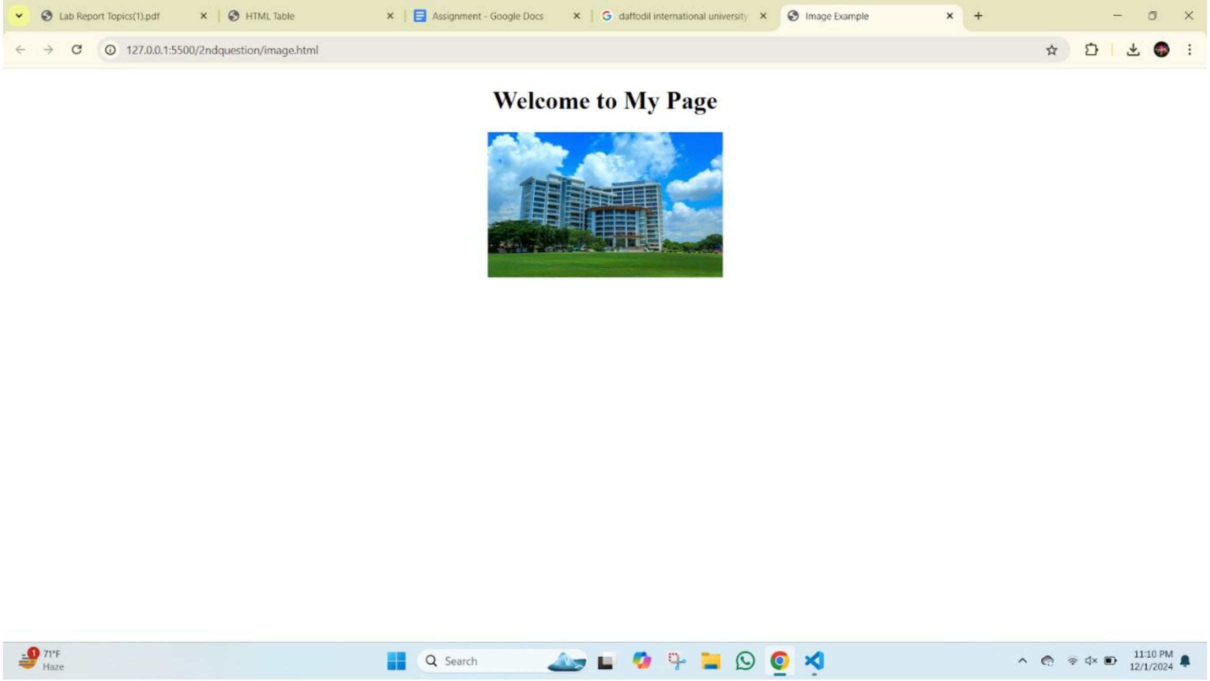
Name	Subject	Grade
Ifan	Math	A
Priya	Science	B

2. Add image to your webpage.

Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Image Example</title>
</head>
<body>
    <h1 style="text-align: center;">Welcome to My Page</h1>
    <img
src="https://studentshub.daffodilvarsity.edu.bd/assets/img/about/1719679319.jpeg"
alt="Sample Image" style="display: block; margin: 0 auto; width: 300px;">
</body>
</html>
```

Output:



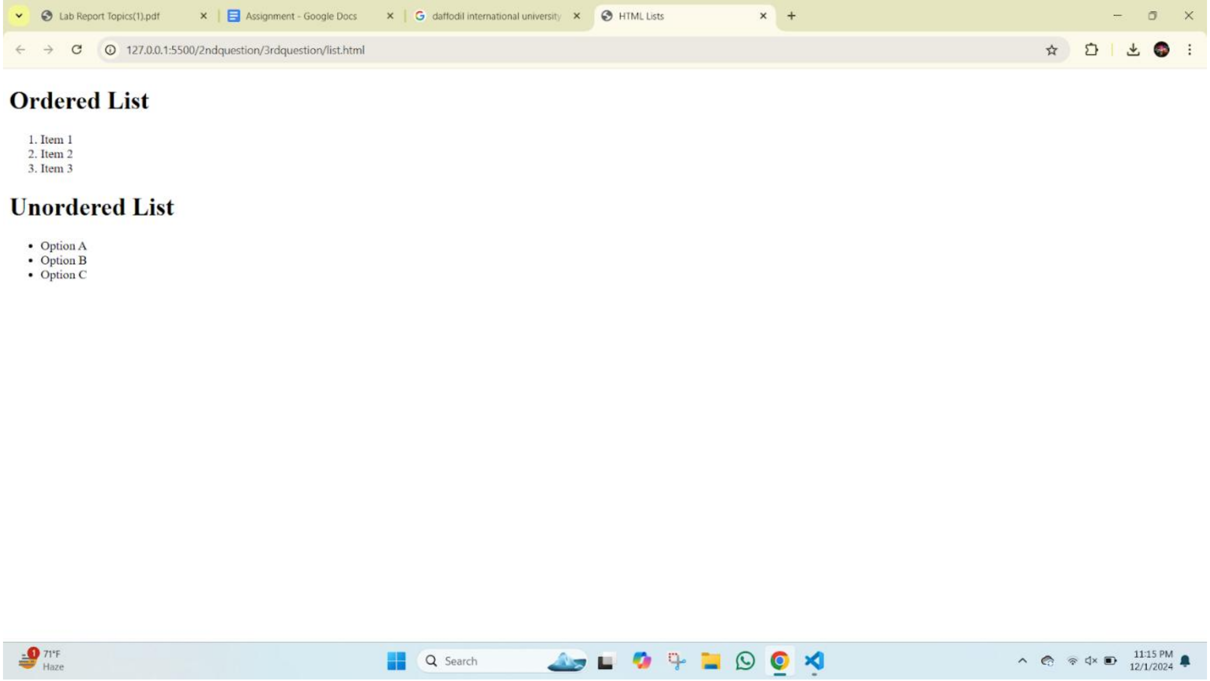
Welcome to My Page

3. HTML list (ordered list and unordered list)

Code:

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>HTML Lists</title>
</head>
<body>
    <h1>Ordered List</h1>
    <ol>
        <li>Item 1</li>
        <li>Item 2</li>
        <li>Item 3</li>
    </ol>

    <h1>Unordered List</h1>
    <ul>
        <li>Option A</li>
        <li>Option B</li>
        <li>Option C</li>
    </ul>
</body>
</html>
```

Output:



Ordered List

1. Item 1
2. Item 2
3. Item 3

Unordered List

- Option A
- Option B
- Option C

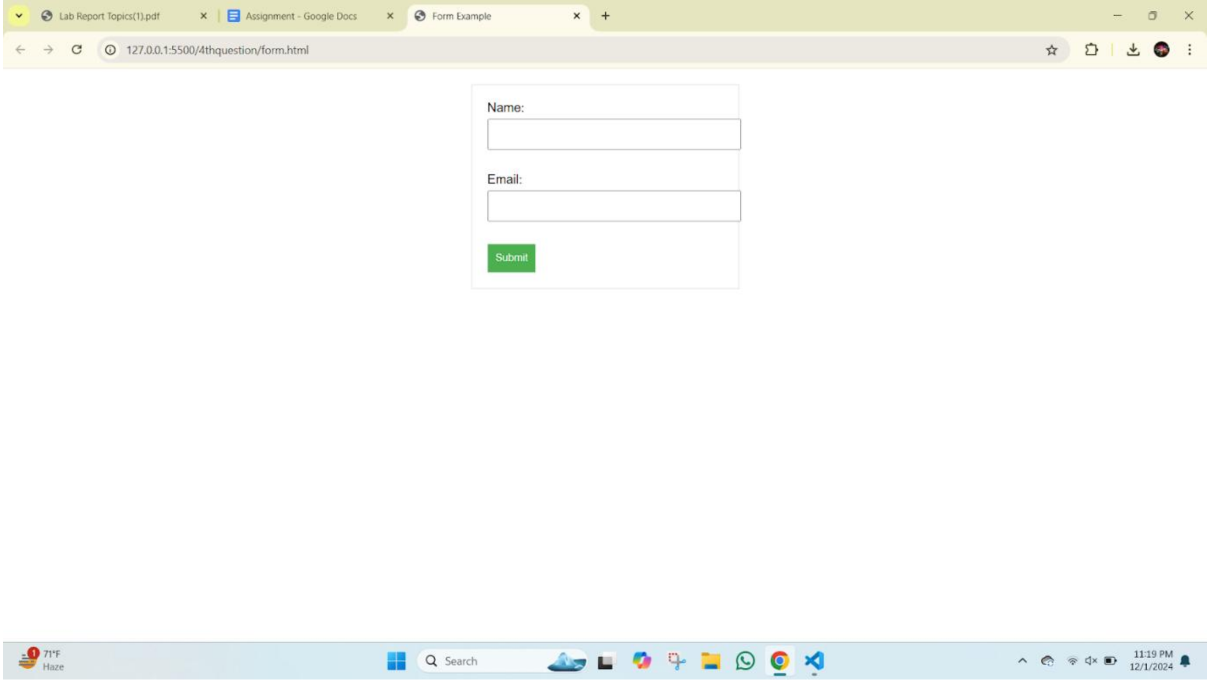
4. HTML form design with external CSS.

Code:

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Form Example</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
```

```
    <form>
        <label for="name">Name:</label>
        <input type="text" id="name" name="name"><br><br>
        <label for="email">Email:</label>
        <input type="email" id="email" name="email"><br><br>
        <button type="submit">Submit</button>
    </form>
</body>
</html>

body {
    font-family: Arial, sans-serif;
    margin: 20px;
}
form {
    max-width: 300px;
    margin: 0 auto;
    padding: 20px;
    border: 1px solid #ddd;
}
label {
    display: block;
    margin-bottom: 5px;
}
input {
    width: 100%;
    padding: 10px;
    margin-bottom: 10px;
}
button {
    background-color: #4CAF50;
    color: white;
    padding: 10px;
    border: none;
    cursor: pointer;
}
```

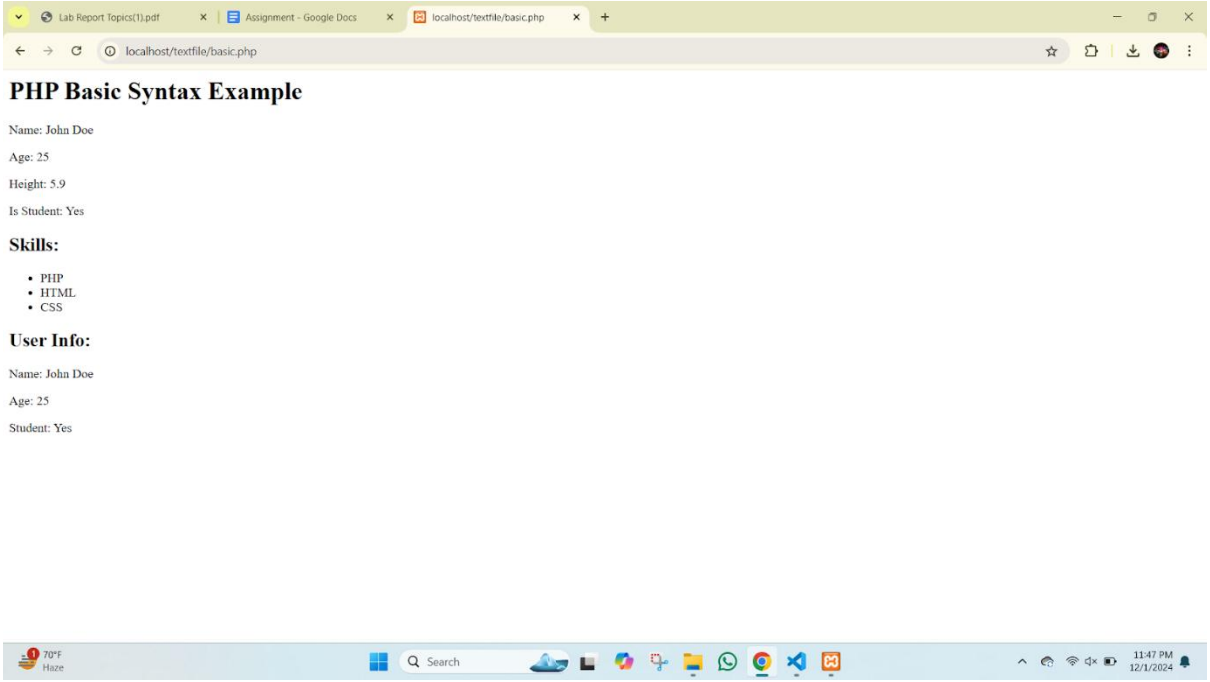
Output:



5. PHP basic syntax, data types, and variables.
Code:

```php
<?php
// Basic Syntax, Data Types, and Variables in PHP

// String
$name = "John Doe";

// Integer
$age = 25;

// Float
$height = 5.9;

// Boolean
$isStudent = true;

// Array
$skills = array("PHP", "HTML", "CSS");

// Associative Array
$userInfo = array(
    "Name" => $name,
    "Age" => $age,
    "Student" => $isStudent
);

// Output using echo
echo "<h1>PHP Basic Syntax Example</h1>";
echo "<p>Name: $name</p>";
echo "<p>Age: $age</p>";
echo "<p>Height: $height</p>";
echo "<p>Is Student: " . ($isStudent ? "Yes" : "No") . "</p>";

// Output Array Elements
echo "<h2>Skills:</h2>";
echo "<ul>";
foreach ($skills as $skill) {
    echo "<li>$skill</li>";
}
echo "</ul>";

// Output Associative Array
echo "<h2>User Info:</h2>";
foreach ($userInfo as $key => $value) {
    echo "<p>$key: " . (is_bool($value) ? ($value ? "Yes" : "No") : $value) .
"</p>";
}
?>
```

Output:



PHP Basic Syntax Example

Name: John Doe

Age: 25

Height: 5.9

Is Student: Yes

Skills:

- PHP
- HTML
- CSS

User Info:

Name: John Doe

Age: 25

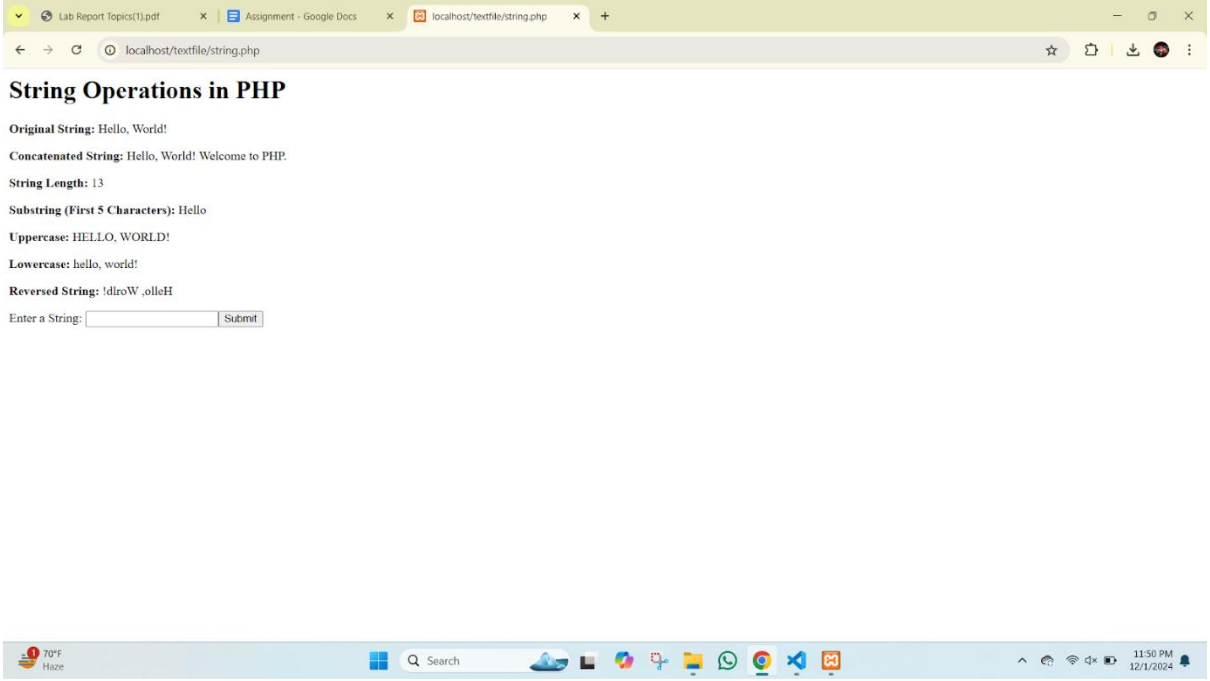
Student: Yes

6. String operation in PHP.

Code:

```php
<?php
// String Operations in PHP

// Input String
$originalString = "Hello, World!";

// Perform Various String Operations
$concatenatedString = $originalString . " Welcome to PHP.";
$stringLength = strlen($originalString);
$substring = substr($originalString, 0, 5);
$uppercaseString = strtoupper($originalString);
$lowercaseString = strtolower($originalString);
$reversedString = strrev($originalString);

// Output Results
echo "<h1>String Operations in PHP</h1>";
echo "<p><strong>Original String:</strong> $originalString</p>";
echo "<p><strong>Concatenated String:</strong> $concatenatedString</p>";
echo "<p><strong>String Length:</strong> $stringLength</p>";
echo "<p><strong>Substring (First 5 Characters):</strong> $substring</p>";
echo "<p><strong>Uppercase:</strong> $uppercaseString</p>";
echo "<p><strong>Lowercase:</strong> $lowercaseString</p>";
echo "<p><strong>Reversed String:</strong> $reversedString</p>";

// Interactive User Input (Optional)
echo '<form method="POST">';
echo '<label for="inputString">Enter a String: </label>';
echo '<input type="text" id="inputString" name="inputString">';
echo '<button type="submit">Submit</button>';
echo '</form>';

// Handle Form Submission
if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['inputString'])) {
    $userInput = $_POST['inputString'];
    echo "<h2>Operations on Your Input</h2>";
    echo "<p><strong>Original Input:</strong> $userInput</p>";
    echo "<p><strong>Reversed Input:</strong> " . strrev($userInput) . "</p>";
    echo "<p><strong>Uppercase:</strong> " . strtoupper($userInput) . "</p>";
    echo "<p><strong>Substring (First 3 Characters):</strong> " .
substr($userInput, 0, 3) . "</p>";
}
?>
```

Output:



String Operations in PHP

Original String: Hello, World!

Concatenated String: Hello, World! Welcome to PHP.

String Length: 13

Substring (First 5 Characters): Hello

Uppercase: HELLO, WORLD!

Lowercase: hello, world!

Reversed String: !dlroW ,olleH

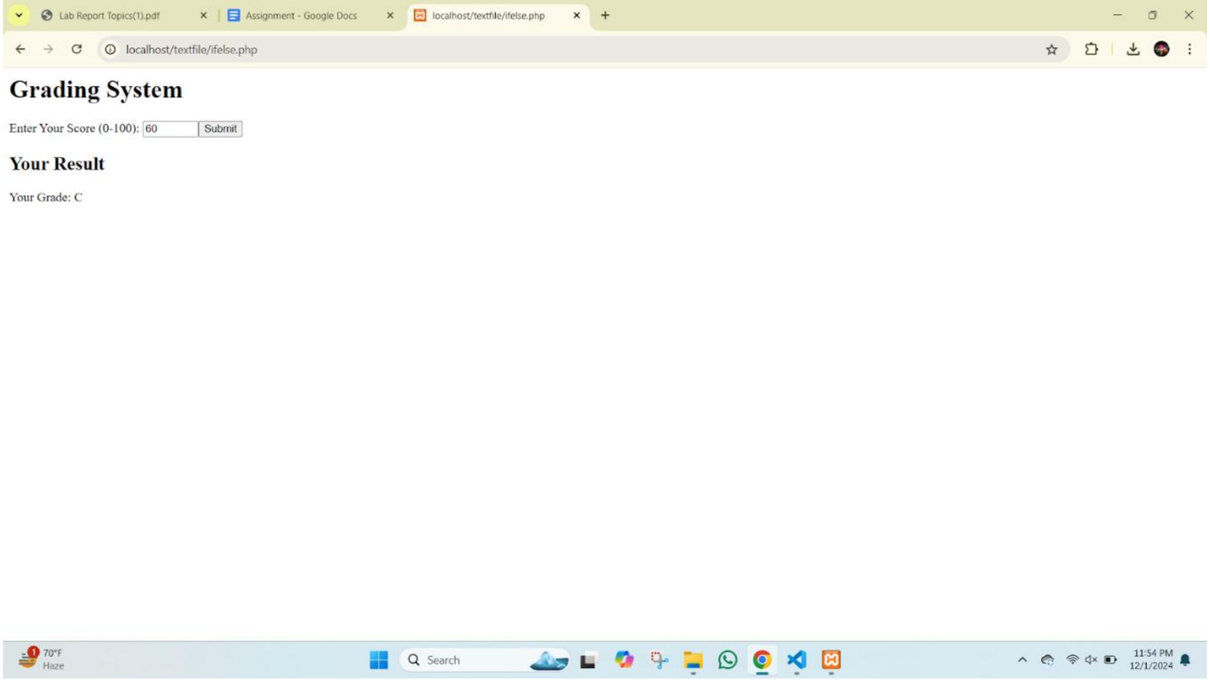
Enter a String: [ ] Submit

7. If-Else condition in PHP.

Code:

```
<?php
// If-Else Condition in PHP - Grading System

echo "<h1>Grading System</h1>";

// Form for User Input
echo '<form method="POST">';
echo '<label for="score">Enter Your Score (0-100): </label>';
echo '<input type="number" id="score" name="score" min="0" max="100">';
echo '<button type="submit">Submit</button>';
echo '</form>';

// Handle Form Submission
if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['score'])) {
    $score = $_POST['score'];

    echo "<h2>Your Result</h2>";

    // If-Else Conditions for Grading
    if ($score >= 90) {
        echo "<p>Your Grade: A+</p>";
    } elseif ($score >= 80) {
        echo "<p>Your Grade: A</p>";
    } elseif ($score >= 70) {
        echo "<p>Your Grade: B</p>";
    } elseif ($score >= 60) {
        echo "<p>Your Grade: C</p>";
    } elseif ($score >= 50) {
        echo "<p>Your Grade: D</p>";
    } else {
        echo "<p>Your Grade: F (Failed)</p>";
    }
}
?>
```

Output:



Grading System

Enter Your Score (0-100): 60 Submit

Your Result

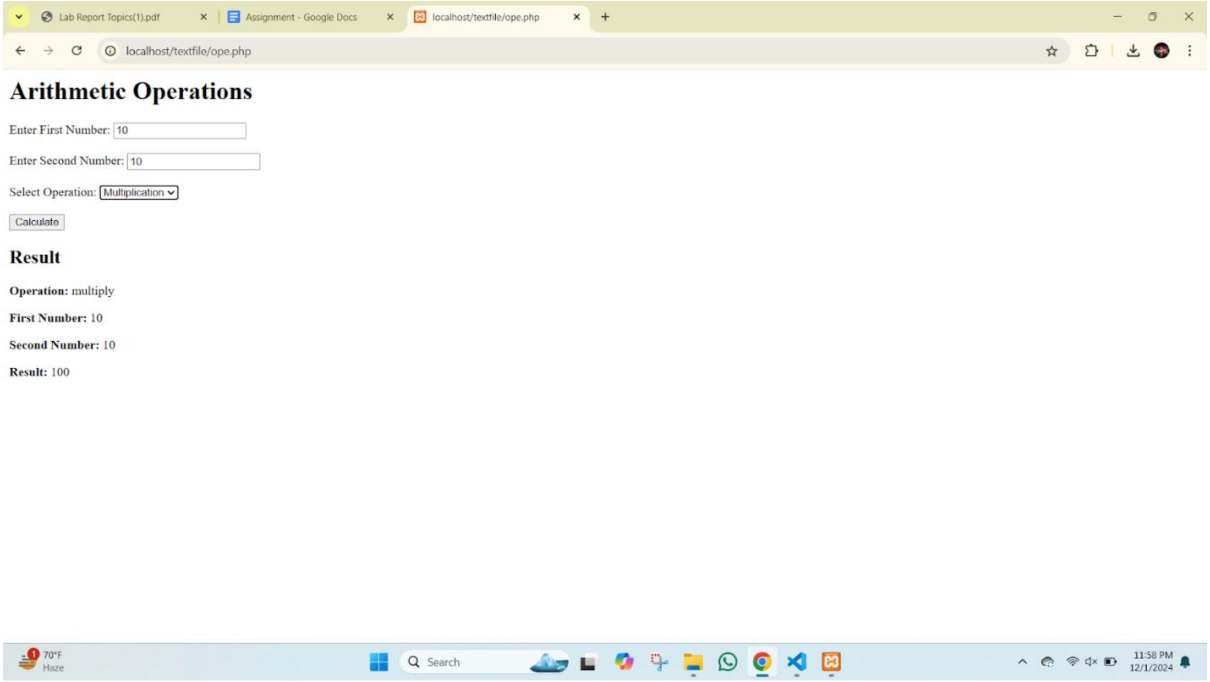
Your Grade: C

8. Different types of arithmetic operation using function in PHP (user defined function).

Code:

```
<?php
// Arithmetic Operations Using User-Defined Functions in PHP

// User-Defined Functions for Arithmetic Operations
function add($num1, $num2) {
    return $num1 + $num2;
}

function subtract($num1, $num2) {
    return $num1 - $num2;
}

function multiply($num1, $num2) {
    return $num1 * $num2;
}

function divide($num1, $num2) {
    if ($num2 != 0) {
        return $num1 / $num2;
    } else {
        return "Error: Division by zero";
    }
}

// Form for User Input
echo "<h1>Arithmetic Operations</h1>";
echo '<form method="POST">';
echo '<label for="num1">Enter First Number: </label>';
echo '<input type="number" id="num1" name="num1" step="any" required><br><br>';
echo '<label for="num2">Enter Second Number: </label>';
echo '<input type="number" id="num2" name="num2" step="any" required><br><br>';
echo '<label for="operation">Select Operation: </label>';
echo '<select id="operation" name="operation" required>';
echo '<option value="add">Addition</option>';
echo '<option value="subtract">Subtraction</option>';
echo '<option value="multiply">Multiplication</option>';
echo '<option value="divide">Division</option>';
echo '</select><br><br>';
echo '<button type="submit">Calculate</button>';
echo '</form>';

// Handle Form Submission
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $num1 = $_POST['num1'];
    $num2 = $_POST['num2'];
    $operation = $_POST['operation'];

    // Perform the selected operation
    $result = null;
    switch ($operation) {
        case 'add':
            $result = add($num1, $num2);
            break;
        case 'subtract':
            $result = subtract($num1, $num2);
            break;
        case 'multiply':
            $result = multiply($num1, $num2);
            break;
        case 'divide':
            $result = divide($num1, $num2);
            break;
    }

    // Display the result
    echo "<h2>Result</h2>";
    echo "<p><strong>Operation:</strong> $operation</p>";
    echo "<p><strong>First Number:</strong> $num1</p>";
    echo "<p><strong>Second Number:</strong> $num2</p>";
    echo "<p><strong>Result:</strong> $result</p>";
}
?>
```

Output:



Arithmetic Operations

Enter First Number: 10

Enter Second Number: 10

Select Operation: Multiplication

Calculate

Result

Operation: multiply

First Number: 10

Second Number: 10

Result: 100

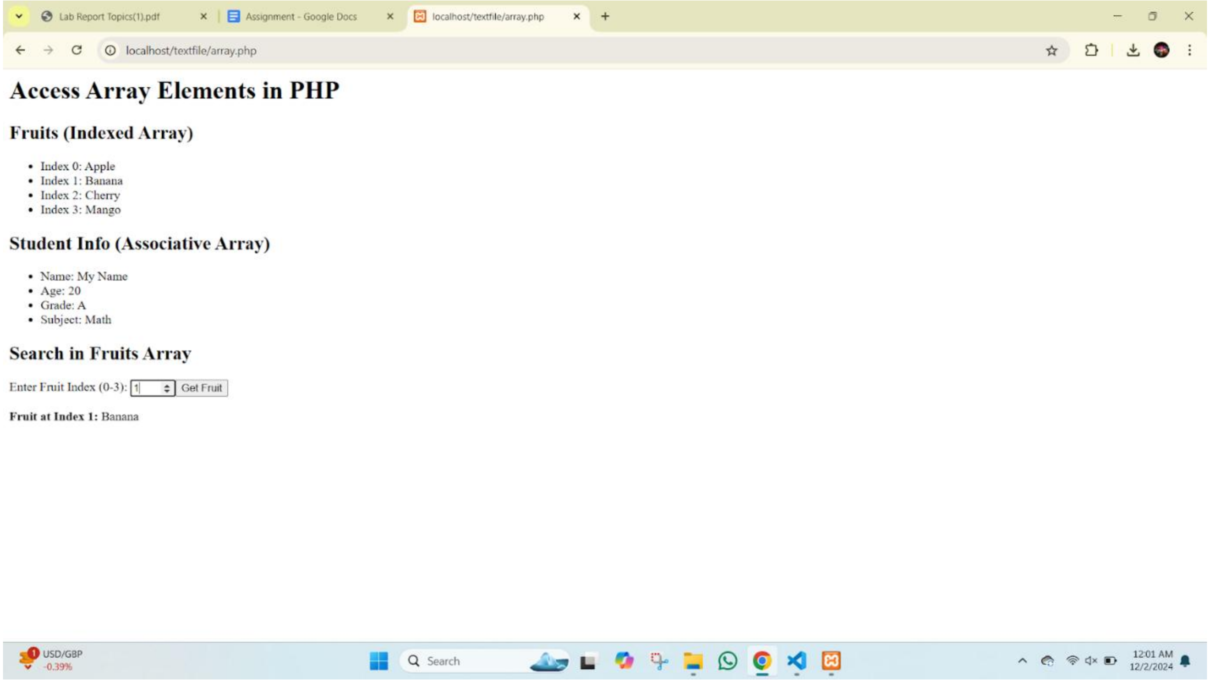
9. Access array elements and printout the value in PHP.

Code:

```
<?php
// Indexed Array
$fruits = ["Apple", "Banana", "Cherry", "Mango"];

// Associative Array
$student = [
    "Name" => "My Name",
    "Age" => 20,
    "Grade" => "A",
    "Subject" => "Math"
];

echo "<h1>Access Array Elements in PHP</h1>";

// Display Indexed Array
echo "<h2>Fruits (Indexed Array)</h2>";
echo "<ul>";
foreach ($fruits as $index => $fruit) {
    echo "<li>Index $index: $fruit</li>";
}
echo "</ul>";

// Display Associative Array
echo "<h2>Student Info (Associative Array)</h2>";
echo "<ul>";
foreach ($student as $key => $value) {
    echo "<li>$key: $value</li>";
}
echo "</ul>";

// Add User Interaction (Optional)
echo '<h2>Search in Fruits Array</h2>';
echo '<form method="POST">';
echo '<label for="fruitIndex">Enter Fruit Index (0-3): </label>';
echo '<input type="number" id="fruitIndex" name="fruitIndex" min="0" max="3">';
echo '<button type="submit">Get Fruit</button>';
echo '</form>';
```

```
if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['fruitIndex'])) {
    $fruitIndex = $_POST['fruitIndex'];
    if (isset($fruits[$fruitIndex])) {
        echo "<p><strong>Fruit at Index $fruitIndex:</strong>
{$fruits[$fruitIndex]}</p>";
    } else {
        echo "<p><strong>Error:</strong> Invalid index!</p>";
    }
}
?>
```

Output:



Access Array Elements in PHP

Fruits (Indexed Array)

- Index 0: Apple
- Index 1: Banana
- Index 2: Cherry
- Index 3: Mango

Student Info (Associative Array)

- Name: My Name
- Age: 20
- Grade: A
- Subject: Math

Search in Fruits Array

Enter Fruit Index (0-3): 1 Get Fruit

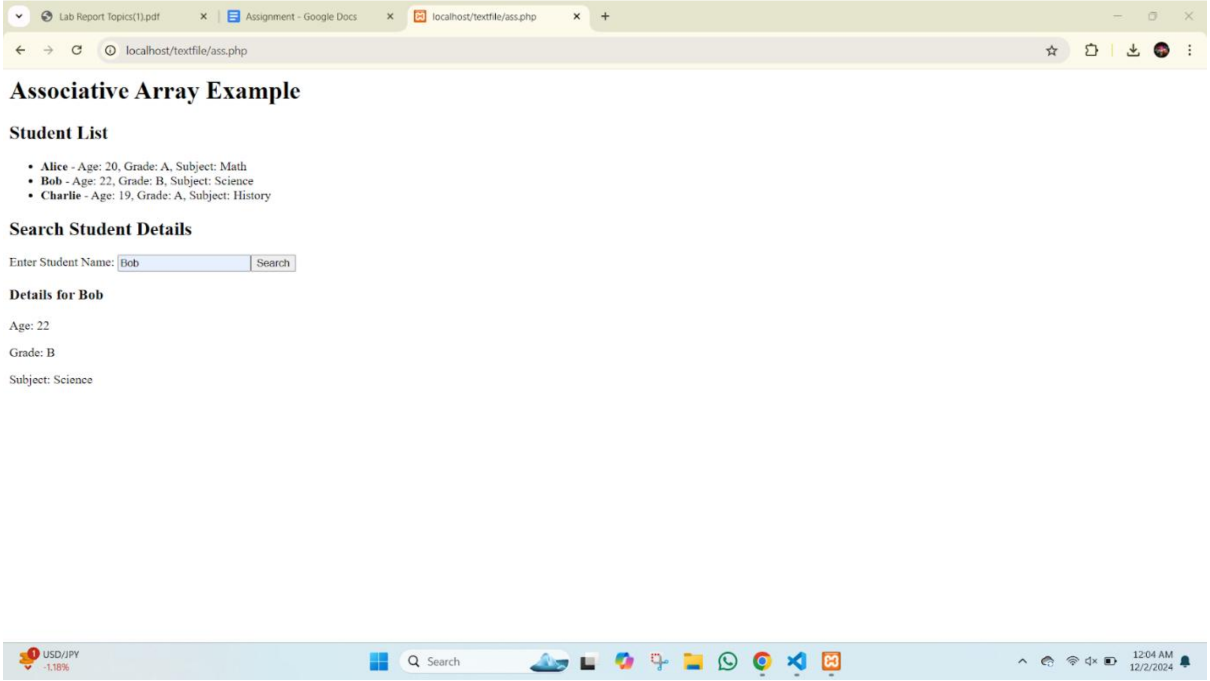
Fruit at Index 1: Banana

10. Associative array in PHP.

Code:

```
<?php
// Associative Array of Students
$students = [
    "Alice" => ["Age" => 20, "Grade" => "A", "Subject" => "Math"],
    "Bob" => ["Age" => 22, "Grade" => "B", "Subject" => "Science"],
    "Charlie" => ["Age" => 19, "Grade" => "A", "Subject" => "History"]
];

echo "<h1>Associative Array Example</h1>";

// Display All Students
echo "<h2>Student List</h2>";
echo "<ul>";
foreach ($students as $name => $details) {
    echo "<li><strong>$name</strong> - Age: {$details['Age']}, Grade:
{$details['Grade']}, Subject: {$details['Subject']}</li>";
}
echo "</ul>";

// User Input to Search for a Student
echo '<h2>Search Student Details</h2>';
echo '<form method="POST">';
echo '<label for="studentName">Enter Student Name: </label>';
echo '<input type="text" id="studentName" name="studentName" required>';
echo '<button type="submit">Search</button>';
echo '</form>';

// Handle Search Submission
if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['studentName'])) {
    $studentName = $_POST['studentName'];
    if (array_key_exists($studentName, $students)) {
        echo "<h3>Details for $studentName</h3>";
        echo "<p>Age: {$students[$studentName]['Age']}</p>";
        echo "<p>Grade: {$students[$studentName]['Grade']}</p>";
        echo "<p>Subject: {$students[$studentName]['Subject']}</p>";
    } else {
        echo "<p><strong>Error:</strong> Student not found!</p>";
    }
}
?>
```

Output:



Associative Array Example

Student List

- **Alice** - Age: 20, Grade: A, Subject: Math
- **Bob** - Age: 22, Grade: B, Subject: Science
- **Charlie** - Age: 19, Grade: A, Subject: History

Search Student Details

Enter Student Name: Bob Search

Details for Bob

Age: 22

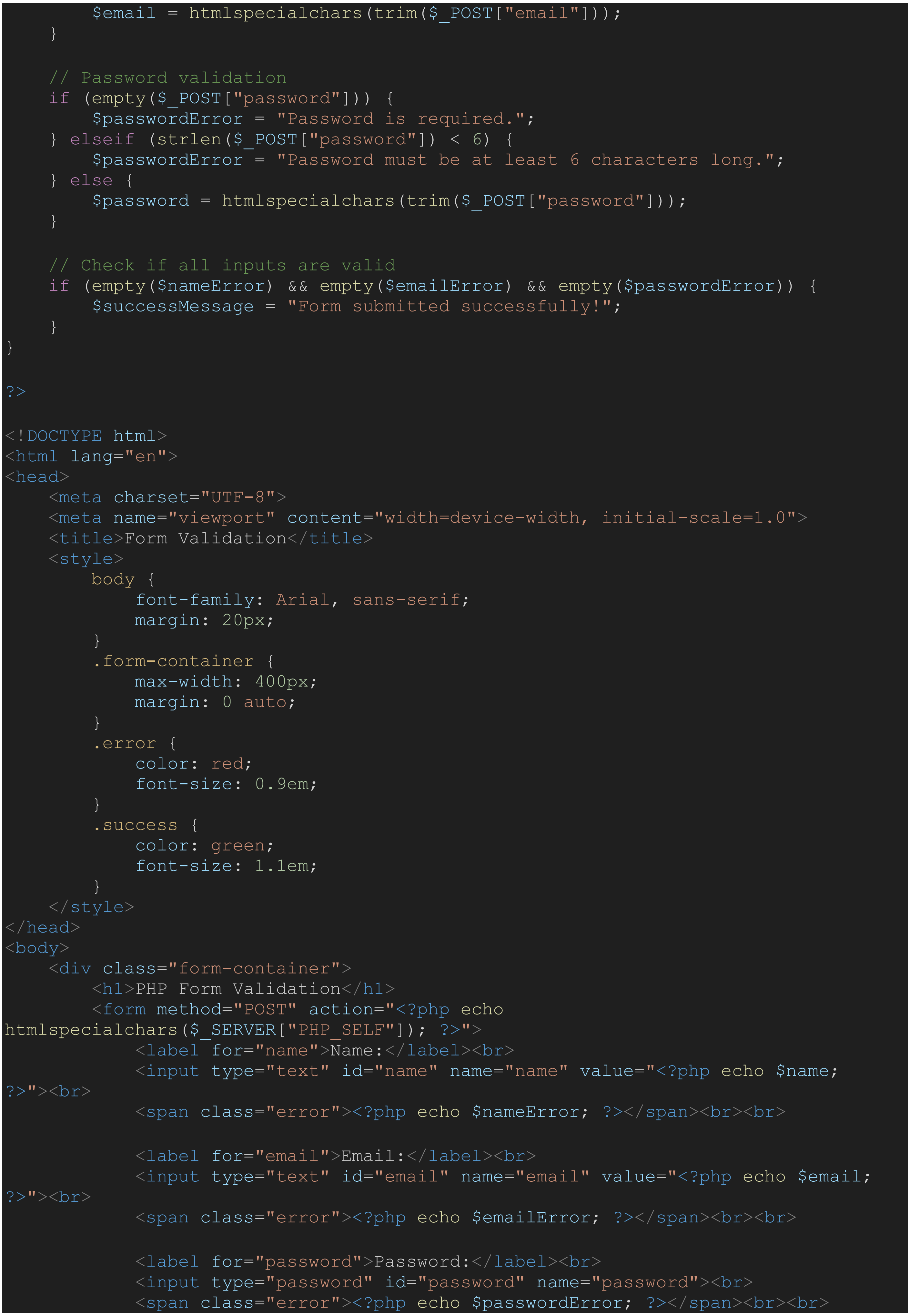
Grade: B

Subject: Science

11. Form validation in PHP.

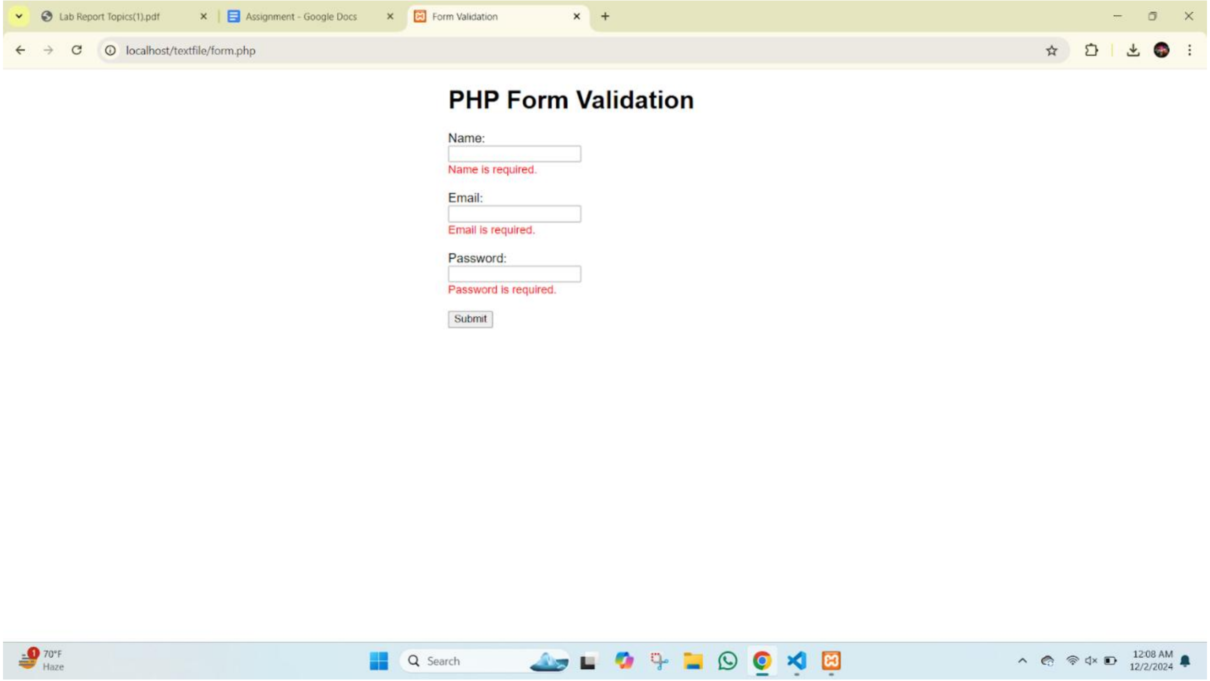
Code:

```
<?php
// Initialize variables for form values and error messages
$name = $email = $password = "";
$nameError = $emailError = $passwordError = $successMessage = "";

// Validate the form upon submission
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Name validation
    if (empty($_POST["name"])) {
        $nameError = "Name is required.";
    } else {
        $name = htmlspecialchars(trim($_POST["name"]));
    }

    // Email validation
    if (empty($_POST["email"])) {
        $emailError = "Email is required.";
    } elseif (!filter_var($_POST["email"], FILTER_VALIDATE_EMAIL)) {
        $emailError = "Invalid email format.";
    } else {
```



```
        $email = htmlspecialchars(trim($_POST["email"]));
    }

    // Password validation
    if (empty($_POST["password"])) {
        $passwordError = "Password is required.";
    } elseif (strlen($_POST["password"]) < 6) {
        $passwordError = "Password must be at least 6 characters long.";
    } else {
        $password = htmlspecialchars(trim($_POST["password"]));
    }

    // Check if all inputs are valid
    if (empty($nameError) && empty($emailError) && empty($passwordError)) {
        $successMessage = "Form submitted successfully!";
    }
}

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Form Validation</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 20px;
        }
        .form-container {
            max-width: 400px;
            margin: 0 auto;
        }
        .error {
            color: red;
            font-size: 0.9em;
        }
        .success {
            color: green;
            font-size: 1.1em;
        }
    </style>
</head>
<body>
    <div class="form-container">
        <h1>PHP Form Validation</h1>
        <form method="POST" action="<?php echo
htmlspecialchars($_SERVER["PHP_SELF"]); ?>">
            <label for="name">Name:</label><br>
            <input type="text" id="name" name="name" value="<?php echo $name;
?>"><br>
            <span class="error"><?php echo $nameError; ?></span><br><br>

            <label for="email">Email:</label><br>
            <input type="text" id="email" name="email" value="<?php echo $email;
?>"><br>
            <span class="error"><?php echo $emailError; ?></span><br><br>

            <label for="password">Password:</label><br>
            <input type="password" id="password" name="password"><br>
            <span class="error"><?php echo $passwordError; ?></span><br><br>
            <button type="submit">Submit</button>
        </form>

        <?php
        if (!empty($successMessage)) {
            echo "<p class='success'>$successMessage</p>";
        }
        ?>
    </div>
</body>
</html>
```

Output:



PHP Form Validation

Name:

Name is required.

Email:

Email is required.

Password:

Password is required.

Submit

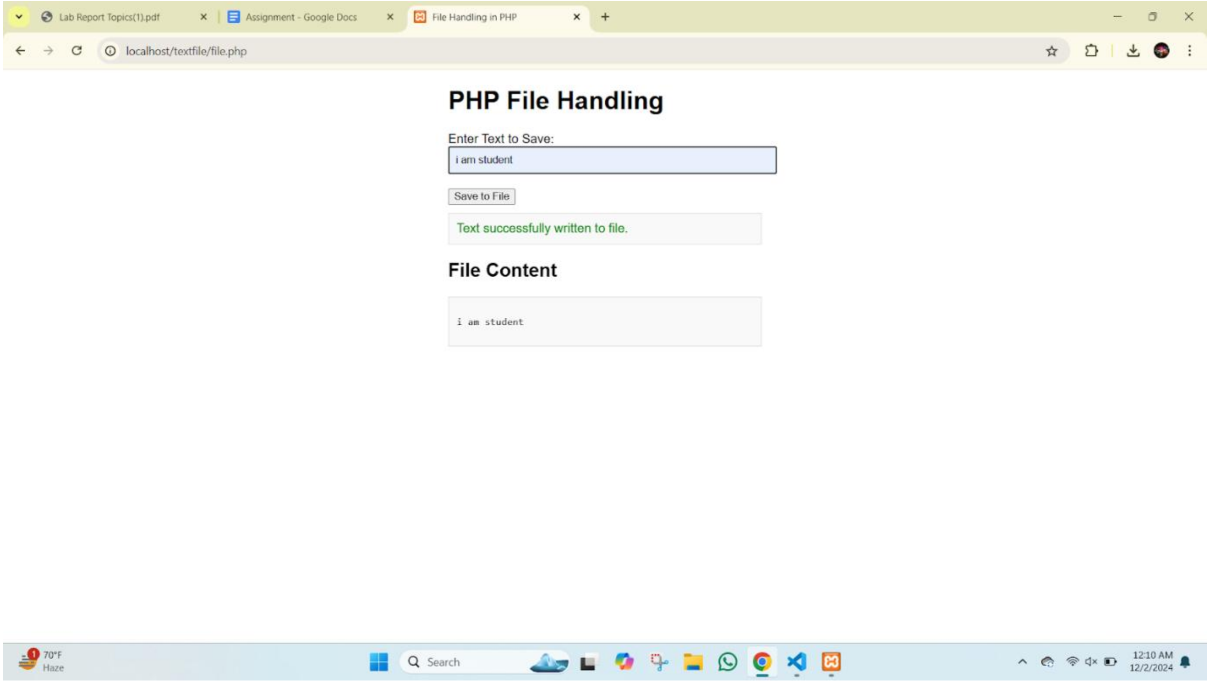
12. File handling in PHP.

Code:

```
<?php
// File path to save data
$filePath = "data.txt";
$writeMessage = $readMessage = "";

// Handle form submission to write to file
if ($_SERVER["REQUEST_METHOD"] === "POST") {
    if (isset($_POST["textInput"]) && !empty(trim($_POST["textInput"]))) {
        $textInput = htmlspecialchars(trim($_POST["textInput"]));
        // Write the input to the file
        file_put_contents($filePath, $textInput . PHP_EOL, FILE_APPEND);
        $writeMessage = "Text successfully written to file.";
    } else {
        $writeMessage = "Please enter some text to save.";
    }
}

// Read content from the file
if (file_exists($filePath)) {
    $readMessage = file_get_contents($filePath);
} else {
    $readMessage = "The file does not exist yet.";
}
```

```
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>File Handling in PHP</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 20px;
        }
        .form-container {
            max-width: 400px;
            margin: 0 auto;
        }
        .message {
            margin-top: 10px;
            padding: 10px;
            border: 1px solid #ddd;
            background-color: #f9f9f9;
        }
        .success {
            color: green;
        }
        .error {
            color: red;
        }
    </style>
</head>
<body>
    <div class="form-container">
        <h1>PHP File Handling</h1>
        <form method="POST" action="<?php echo
htmlspecialchars($_SERVER["PHP_SELF"]); ?>">
            <label for="textInput">Enter Text to Save:</label><br>
            <input type="text" id="textInput" name="textInput" style="width: 100%;
padding: 8px;"><br><br>
            <button type="submit">Save to File</button>
        </form>
        <?php if (!empty($writeMessage)): ?>
            <p class="message <?php echo strpos($writeMessage, 'successfully') !==
false ? 'success' : 'error'; ?>">
                <?php echo $writeMessage; ?>
            </p>
        <?php endif; ?>

        <h2>File Content</h2>
        <div class="message">
            <pre><?php echo $readMessage; ?></pre>
        </div>
    </div>
</body>
</html>
```

Output:



PHP File Handling

Enter Text to Save:

i am student

Text successfully written to file.

File Content

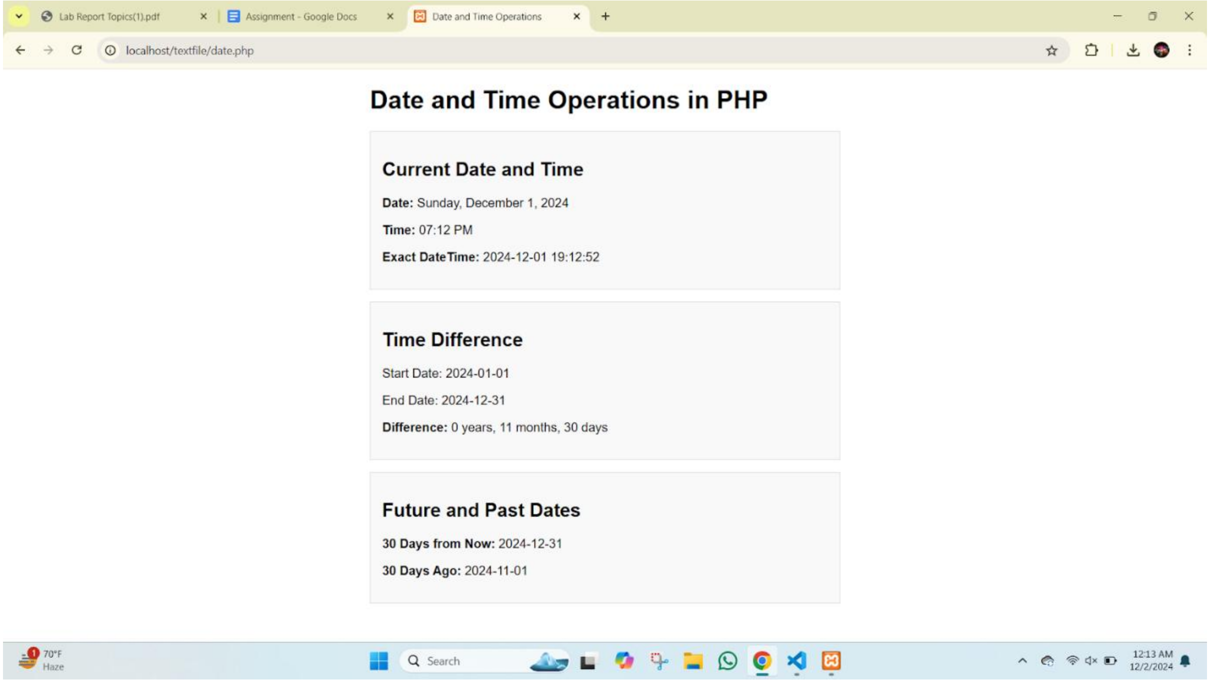
```
i am student
```

13. Date and Time related problem in PHP.

Code:

```
<?php
// Date and Time Operations in PHP

// Current Date and Time
$currentDateTime = date("Y-m-d H:i:s");

// Formatted Date and Time
$formattedDate = date("l, F j, Y");
$formattedTime = date("h:i A");

// Time Difference Example
$startDate = "2024-01-01";
$endDate = "2024-12-31";
$date1 = new DateTime($startDate);
$date2 = new DateTime($endDate);
$interval = $date1->diff($date2);

// Future and Past Date Calculation
$futureDate = date("Y-m-d", strtotime("+30 days"));
$pastDate = date("Y-m-d", strtotime("-30 days"));
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Date and Time Operations</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 20px;
        }
        .container {
            max-width: 600px;
            margin: 0 auto;
        }
        .box {
            border: 1px solid #ddd;
            padding: 15px;
            margin-bottom: 15px;
            background-color: #f9f9f9;
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>Date and Time Operations in PHP</h1>

        <div class="box">
            <h2>Current Date and Time</h2>
            <p><strong>Date:</strong> <?php echo $formattedDate; ?></p>
            <p><strong>Time:</strong> <?php echo $formattedTime; ?></p>
            <p><strong>Exact DateTime:</strong> <?php echo $currentDateTime; ?></p>
        </div>

        <div class="box">
            <h2>Time Difference</h2>
            <p>Start Date: <?php echo $startDate; ?></p>
            <p>End Date: <?php echo $endDate; ?></p>
            <p><strong>Difference:</strong>
                <?php echo $interval->y . " years, " . $interval->m . " months, " .
$interval->d . " days"; ?>
            </p>
        </div>

        <div class="box">
            <h2>Future and Past Dates</h2>
            <p><strong>30 Days from Now:</strong> <?php echo $futureDate; ?></p>
            <p><strong>30 Days Ago:</strong> <?php echo $pastDate; ?></p>
        </div>
    </div>
</body>
</html>
```

Output:



Date and Time Operations in PHP

Current Date and Time

Date: Sunday, December 1, 2024

Time: 07:12 PM

Exact DateTime: 2024-12-01 19:12:52

Time Difference

Start Date: 2024-01-01

End Date: 2024-12-31

Difference: 0 years, 11 months, 30 days

Future and Past Dates

30 Days from Now: 2024-12-31

30 Days Ago: 2024-11-01

14. MySQL database connection in PHP.

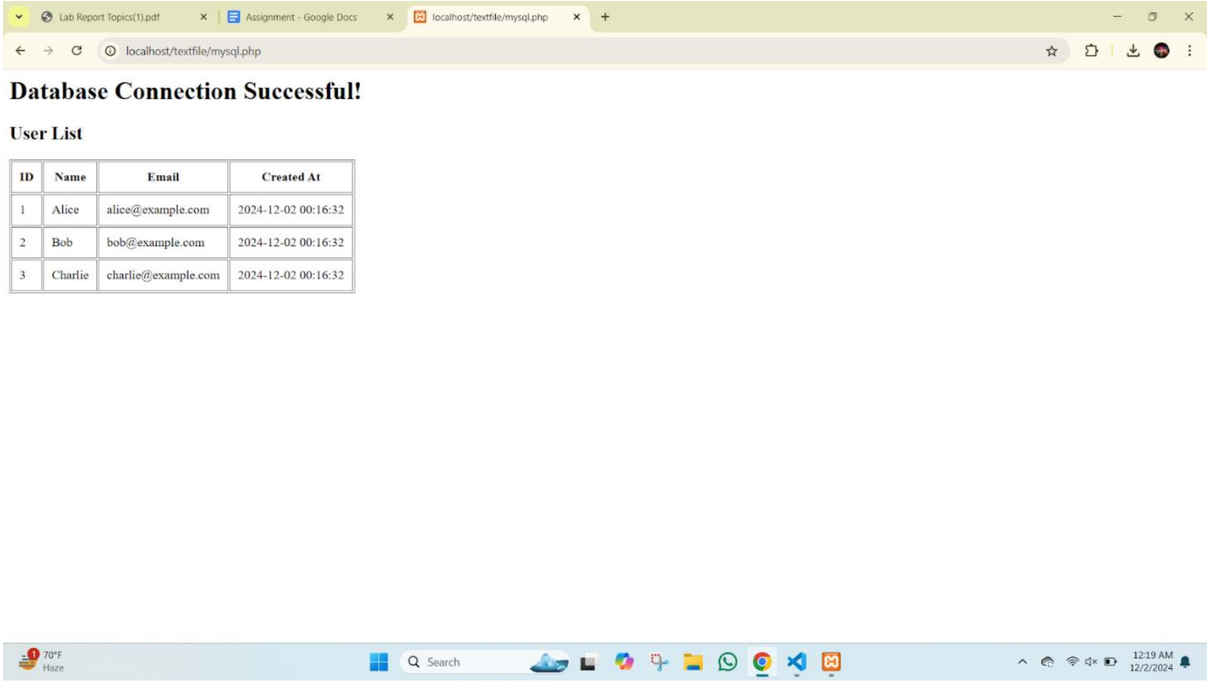
MySql Code:

```
CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(50) NOT NULL,
    email VARCHAR(100) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

INSERT INTO users (name, email) VALUES
('Alice', 'alice@example.com'),
('Bob', 'bob@example.com'),
('Charlie', 'charlie@example.com');
```

Php Code:

```
<?php
// Database credentials
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "testdb";

// Create a connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Check the connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
} else {
    echo "<h1>Database Connection Successful!</h1>";
}

// Fetch data from the 'users' table
$sql = "SELECT id, name, email, created_at FROM users";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    echo "<h2>User List</h2>";
    echo "<table border='1' cellpadding='10'>";
    echo "<tr><th>ID</th><th>Name</th><th>Email</th><th>Created At</th></tr>";

    // Output data of each row
    while ($row = $result->fetch_assoc()) {
        echo "<tr>";
        echo "<td>" . $row['id'] . "</td>";
        echo "<td>" . $row['name'] . "</td>";
        echo "<td>" . $row['email'] . "</td>";
        echo "<td>" . $row['created_at'] . "</td>";
        echo "</tr>";
    }

    echo "</table>";
} else {
    echo "<p>No records found in the database.</p>";
}

// Close the connection
$conn->close();
?>
```

Output:



Database Connection Successful!

User List

ID	Name	Email	Created At
1	Alice	alice@example.com	2024-12-02 00:16:32
2	Bob	bob@example.com	2024-12-02 00:16:32
3	Charlie	charlie@example.com	2024-12-02 00:16:32

15. CRUD operation in PHP.

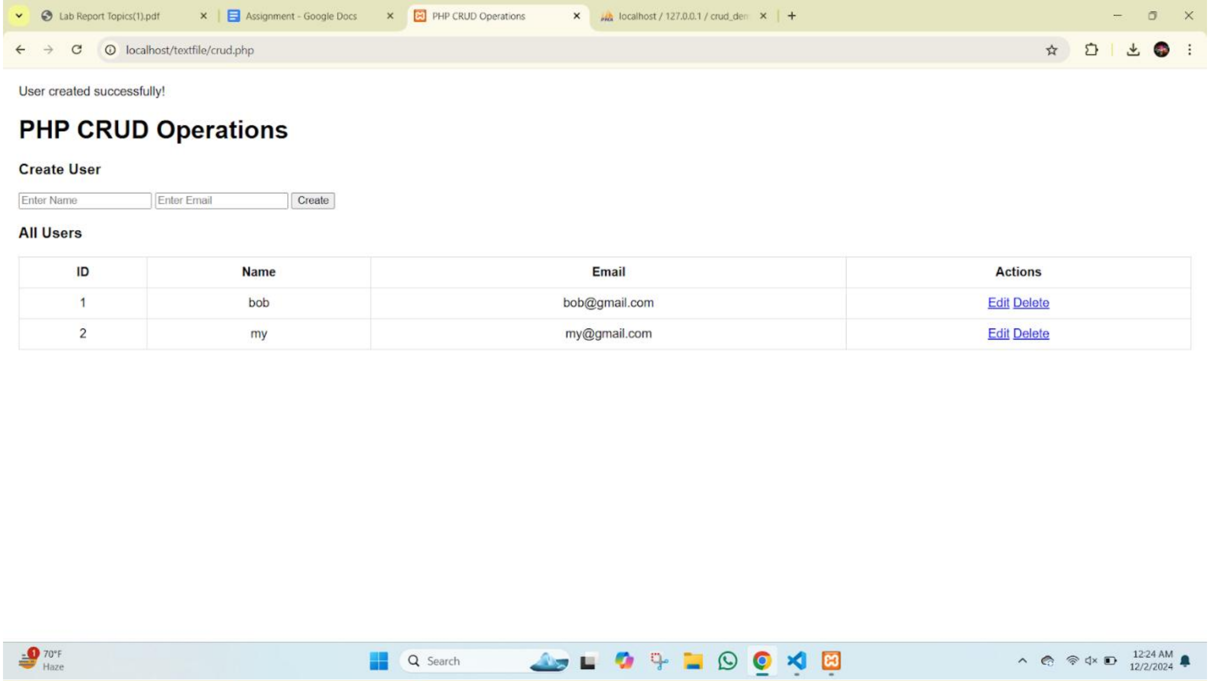
MySql Code:

```
CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(50) NOT NULL,
    email VARCHAR(100) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Php Code:

```php
<?php
// Database connection
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "crud_demo";

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

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Handle Create and Update
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (isset($_POST["create"])) {
        // Create operation
        $name = htmlspecialchars($_POST["name"]);
        $email = htmlspecialchars($_POST["email"]);
        $sql = "INSERT INTO users (name, email) VALUES ('$name', '$email')";
        if ($conn->query($sql)) {
            echo "User created successfully!";
        } else {
            echo "Error: " . $conn->error;
        }
    } elseif (isset($_POST["update"])) {
        // Update operation
        $id = $_POST["id"];
        $name = htmlspecialchars($_POST["name"]);
        $email = htmlspecialchars($_POST["email"]);
        $sql = "UPDATE users SET name='$name', email='$email' WHERE id=$id";
        if ($conn->query($sql)) {
            echo "User updated successfully!";
        } else {
            echo "Error: " . $conn->error;
        }
    }
}

// Handle Delete
if (isset($_GET["delete"])) {
    $id = $_GET["delete"];
    $sql = "DELETE FROM users WHERE id=$id";
    if ($conn->query($sql)) {
        echo "User deleted successfully!";
    } else {
        echo "Error: " . $conn->error;
    }
}

// Fetch all users
$sql = "SELECT * FROM users";
$result = $conn->query($sql);
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>PHP CRUD Operations</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 20px;
        }
        table {
            border-collapse: collapse;
            width: 100%;
            margin-bottom: 20px;
        }
        table, th, td {
            border: 1px solid #ddd;
        }
        th, td {
            padding: 10px;
            text-align: center;
        }
        form {
            margin-bottom: 20px;
        }
    </style>
</head>
<body>
    <h1>PHP CRUD Operations</h1>

    <!-- Create Form -->
    <form method="POST">
        <h3>Create User</h3>
        <input type="text" name="name" placeholder="Enter Name" required>
        <input type="email" name="email" placeholder="Enter Email" required>
        <button type="submit" name="create">Create</button>
    </form>

    <!-- Display Users -->
    <h3>All Users</h3>
    <table>
        <thead>
            <tr>
                <th>ID</th>
                <th>Name</th>
                <th>Email</th>
                <th>Actions</th>
            </tr>
        </thead>
        <tbody>
            <?php while ($row = $result->fetch_assoc()): ?>
                <tr>
                    <td><?php echo $row["id"]; ?></td>
                    <td><?php echo $row["name"]; ?></td>
                    <td><?php echo $row["email"]; ?></td>
                    <td>
                        <a href="crud.php?edit=<?php echo $row['id']; ?>">Edit</a>
                        <a href="crud.php?delete=<?php echo $row['id']; ?>"
onclick="return confirm('Are you sure?')">Delete</a>
                    </td>
                </tr>
            <?php endwhile; ?>
        </tbody>
    </table>

    <!-- Update Form -->
    <?php if (isset($_GET["edit"])): ?>
        <?php
        $id = $_GET["edit"];
        $sql = "SELECT * FROM users WHERE id=$id";
        $result = $conn->query($sql);
        $row = $result->fetch_assoc();
        ?>
        <form method="POST">
            <h3>Update User</h3>
            <input type="hidden" name="id" value="<?php echo $row['id']; ?>">
            <input type="text" name="name" value="<?php echo $row['name']; ?>"
required>
            <input type="email" name="email" value="<?php echo $row['email']; ?>"
required>
            <button type="submit" name="update">Update</button>
        </form>
    <?php endif; ?>
</body>
</html>
```

Output:



User created successfully!

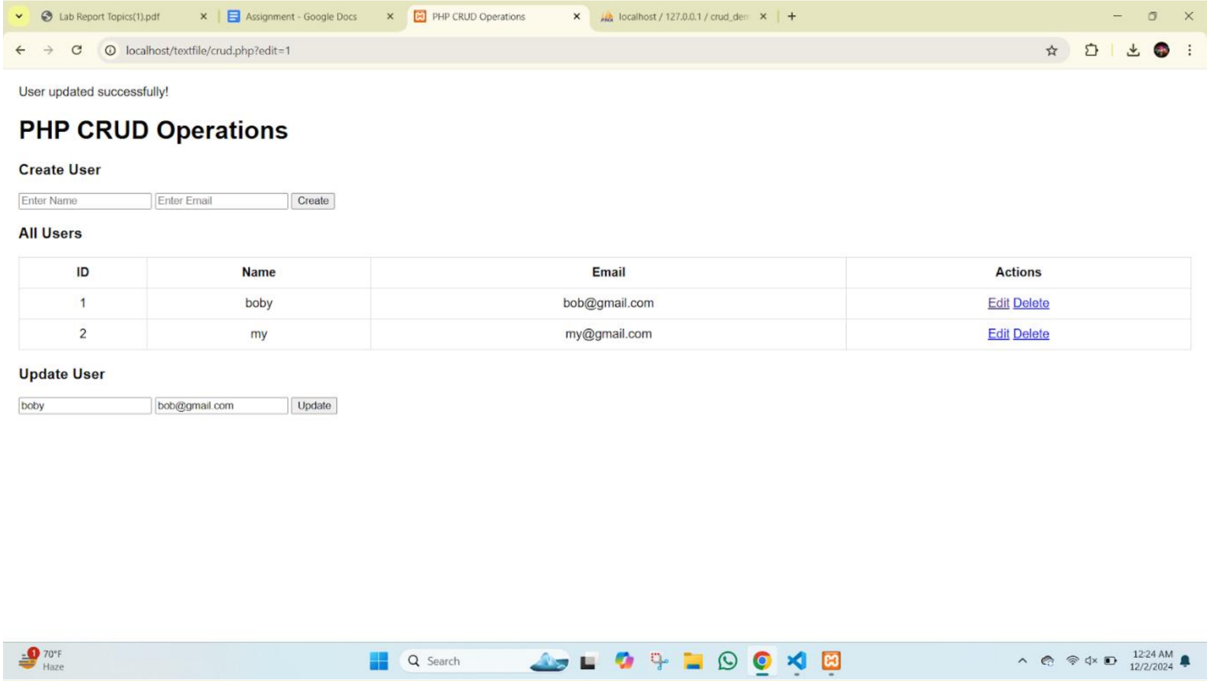
PHP CRUD Operations

Create User

Enter Name | Enter Email | Create

All Users

ID	Name	Email	Actions
1	bob	bob@gmail.com	Edit Delete
2	my	my@gmail.com	Edit Delete



User updated successfully!

PHP CRUD Operations

Create User

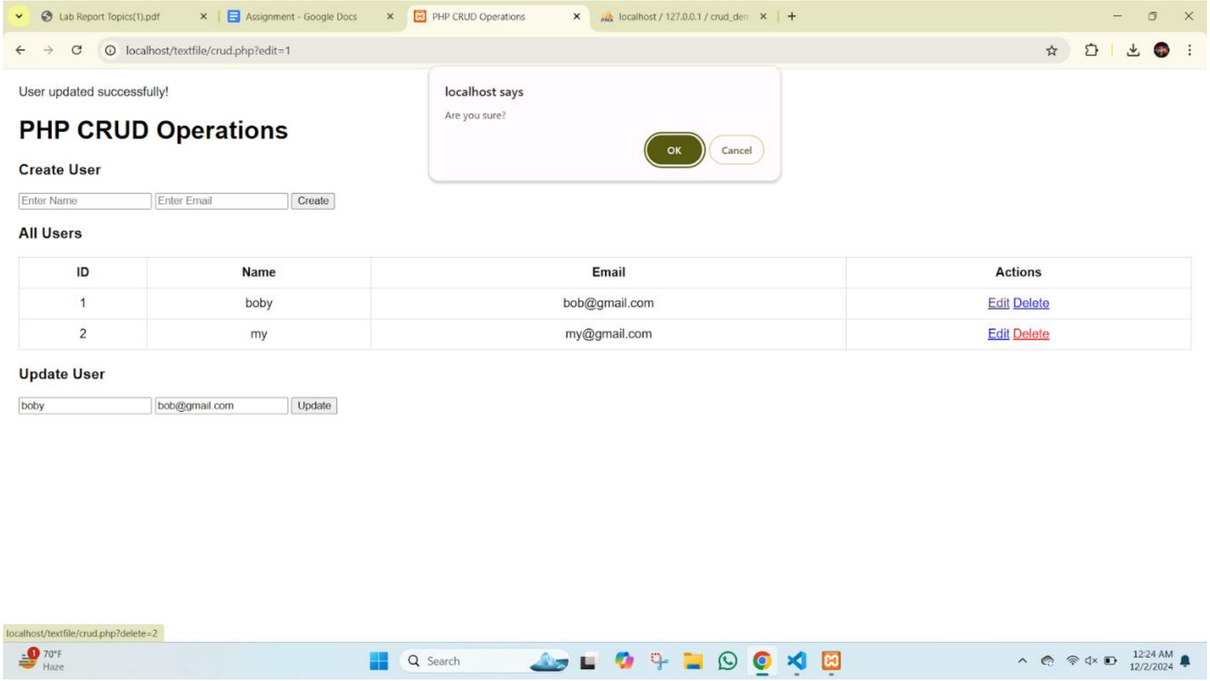
Enter Name | Enter Email | Create

All Users

ID	Name	Email	Actions
1	boby	bob@gmail.com	Edit Delete
2	my	my@gmail.com	Edit Delete

Update User

boby | bob@gmail.com | Update



localhost says

Are you sure?

User updated successfully!

PHP CRUD Operations

Create User

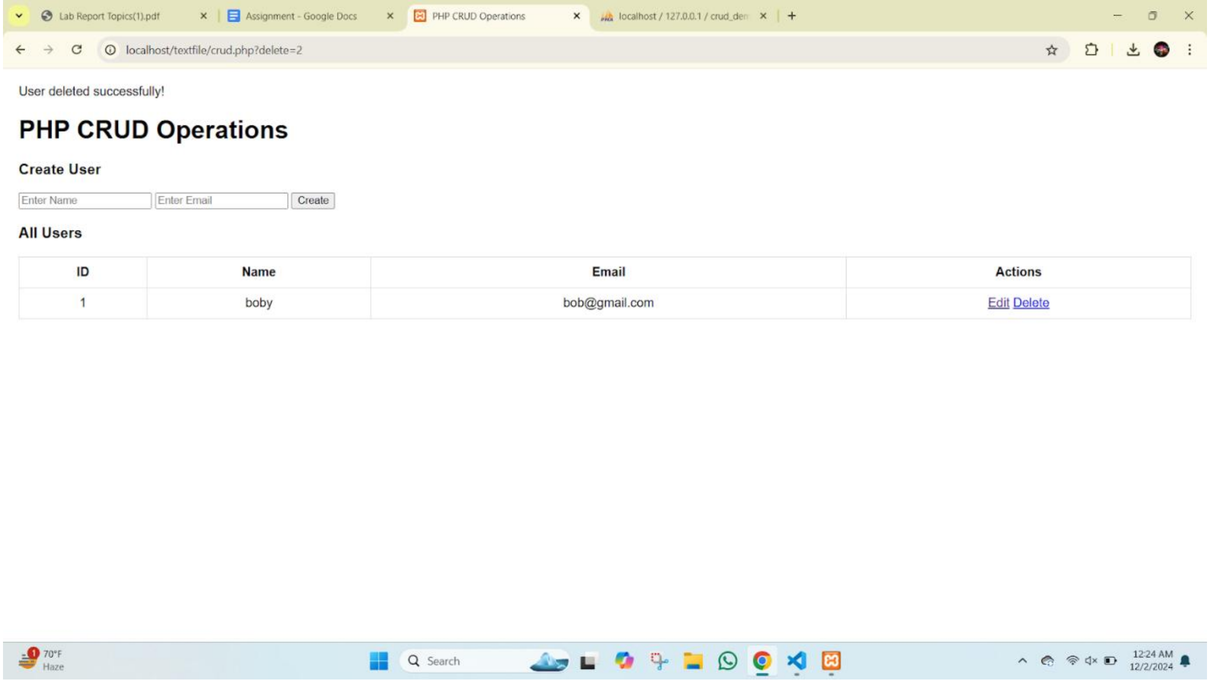
Enter Name | Enter Email | Create

All Users

ID	Name	Email	Actions
1	boby	bob@gmail.com	Edit Delete
2	my	my@gmail.com	Edit Delete

Update User

boby | bob@gmail.com | Update



Lab Report Topics(1).pdf
Assignment - Google Docs
PHP CRUD Operations
localhost / 127.0.0.1 / crud_den
localhost/textfile/crud.php?delete=2

User deleted successfully!

PHP CRUD Operations

Create User

All Users

ID	Name	Email	Actions
1	boby	bob@gmail.com	Edit Delete