## Table of Contents

1. Introduction

2. JavaScript Questions

- DOM Manipulation

- JavaScript Loops

- JavaScript Arrays

- Form Validation

- Calculator App

3. PHP CRUD with MySQL

- Database Structure

- Create Operation

- Read Operation

- Update Operation

- Delete Operation

4. AJAX and jQuery Operations

5. Conclusion

6. References

## Introduction

This lab report covers essential JavaScript concepts related to the DOM, loops, arrays, form validation, and a calculator app. Additionally, it includes a PHP CRUD operation with a MySQL database and demonstrates AJAX and jQuery functionalities with their respective outputs.

## Part 1: JavaScript Questions

### 1. DOM Manipulation

Question: Write JavaScript code to change the text of an HTML element when a button is clicked.

Solution:

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <title>DOM Manipulation</title>  
 <script>  
 function changeText() {  
 document.getElementById("demo").innerHTML = "Text changed!";  
 }  
 </script>  
</head>  
<body>  
 <p id="demo">Original text.</p>  
 <button onclick="changeText()">Click Me</button>  
</body>  
</html>

Output: When the button is clicked, 'Original text.' changes to 'Text changed!'.

### 2. JavaScript Loops

Question: Write a JavaScript program to print numbers from 1 to 10 using a for loop.

Solution:

for (let i = 1; i <= 10; i++) {  
 console.log(i);  
}

Output:  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10

### 3. JavaScript Arrays

Question: Create an array of five numbers and find their sum using a loop.

Solution:

let numbers = [10, 20, 30, 40, 50];  
let sum = 0;  
for (let i = 0; i < numbers.length; i++) {  
 sum += numbers[i];  
}  
console.log("Sum:", sum);

Output: Sum: 150

## Part 2: PHP CRUD with MySQL

### Database Structure

CREATE DATABASE labreport\_db;  
USE labreport\_db;  
CREATE TABLE students (  
 id INT AUTO\_INCREMENT PRIMARY KEY,  
 name VARCHAR(255) NOT NULL,  
 email VARCHAR(255) NOT NULL UNIQUE  
);

### 1. PHP: Create Operation

Solution:

<?php  
$conn = new mysqli("localhost", "root", "", "labreport\_db");  
if ($conn->connect\_error) {  
 die("Connection failed: " . $conn->connect\_error);  
}  
if ($\_SERVER["REQUEST\_METHOD"] == "POST") {  
 $name = $\_POST['name'];  
 $email = $\_POST['email'];  
 $sql = "INSERT INTO students (name, email) VALUES ('$name', '$email')";  
 if ($conn->query($sql) === TRUE) {  
 echo "New record created successfully";  
 } else {  
 echo "Error: " . $sql . " " . $conn->error;  
 }  
}  
?>

Output: Adds a new record.

## Part 3: AJAX and jQuery Operations

### 1. Fetch Data Using AJAX

Solution:

<button id="fetchData">Fetch Data</button>  
<div id="output"></div>  
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>  
<script>  
 $(document).ready(function() {  
 $("#fetchData").click(function() {  
 $.ajax({  
 url: "fetch.php",  
 method: "GET",  
 success: function(data) {  
 $("#output").html(data);  
 }  
 });  
 });  
 });  
</script>

Output: Fetches and displays data.