St. Francis Institute of Technology

(Engineering College)

An Autonomous Institute, Affiliated to University of Mumbai

NAAC A+ Accredited | CMPN, EXTC, INFT NBA Accredited | ISO 9001:2015 Certified

Department of Artificial Intelligence and Machine Learning

Academic Year: 2025-2026 **Term**: Even (Jan. 2026 – Jun. 2026)
Class / Branch: SE – AIML **Semester**: IV
Course: Web Programming Lab. (AI4VS_LR4)
Date of Assignment: / /2026 **Date of Submission:** / /2026

Pre-Lab Exercises for Experiment-6

JavaScript Form Validation, Local Storage & UI Control

Before performing Experiment-6, understand HTML forms, basic validation logic, and local storage concepts.

Part A: Conceptual Exercises

Exercise 1: Form Validation Basics

Task: Answer the following:

1. What is form validation?

Ans. Form validation is the process of checking user input before submitting a form to ensure the data is correct and complete.

2. Why is client-side validation important?

Ans. It prevents incorrect data submission, improves user experience, and reduces server load.

3. Name any two form fields that require validation.

Ans.

➔ Email
➔ Password

Exercise 2: Error Handling

1. Explain how error messages help users during form submission.

Ans. Error messages guide users to correct mistakes during form submission.
They clearly indicate what is wrong (e.g., invalid email, short password) and help improve user experience.

Part B: Hands-On JavaScript Practice

Exercise 3: Email Validation

Task:
• Create an email input field.
• Check whether the input contains @.

CODE

HTML

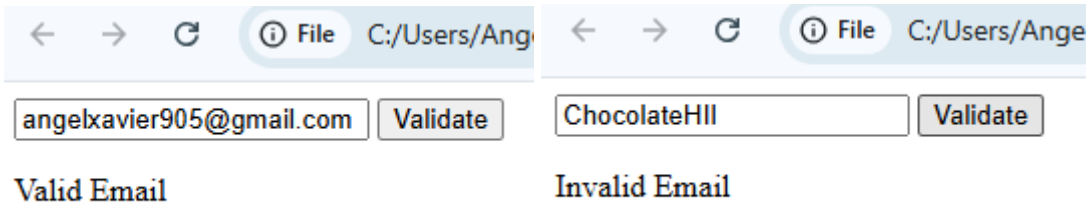
```
<!DOCTYPE html>
<html>
<head>
    <title>Email Validation</title>
```

```
</head>
<body>
<input type="text" id="email"
placeholder="Enter Email">
<button
id="validateBtn">Validate</button>
<p id="result"></p>
<script src="6.js"></script>
</body>
</html>
```

JS

```
document.getElementById("validateBtn")
.addEventListener("click", function()
{
    let email =
document.getElementById("email").value
;
    if (email.includes("@")) {

document.getElementById("result").text
Content = "Valid Email";
    } else {

document.getElementById("result").text
Content = "Invalid Email";
    }
});
```

OUTPUT



Exercise 4: Phone Number Validation

Task:

• Validate that a phone number contains exactly 10 digits.

CODE

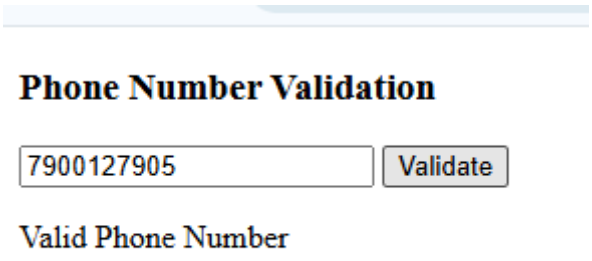
HTML

```
<!DOCTYPE html>
<html>
<head>
    <title>Phone Number
Validation</title>
</head>
<body>
<h3>Phone Number Validation</h3>
<input type="text" id="phone"
placeholder="Enter 10-digit phone
number">
<button
id="validateBtn">Validate</button>
<p id="result"></p>
<script src="6.js"></script>
</body>
</html>
```

JS

```
document.getElementById("validateBtn")
.addEventListener("click", function()
{
    let phone =
document.getElementById("phone").value
;
    if (/^\d{10}$/.test(phone)) {

document.getElementById("result").text
Content = "Valid Phone Number";
    } else {

document.getElementById("result").text
Content = "Invalid Phone Number";
    }
});
```

OUTPUT



Exercise 5: Password Strength Check

Task:

• Display a message if the password length is less than 6 characters.

CODE

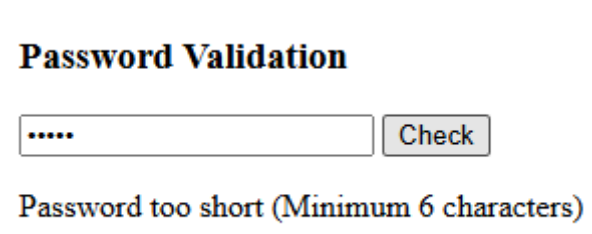
HTML

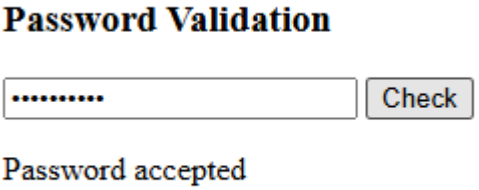
```
<!DOCTYPE html>
<html>
<head>
    <title>Password Strength
Check</title>
</head>
<body>
<h3>Password Validation</h3>
<input type="password" id="password"
placeholder="Enter Password">
<button id="checkBtn">Check</button>
<p id="result"></p>
<script src="6.js"></script>
</body>
</html>
```

JS

```
document.getElementById("checkBtn").ad
dEventListener("click", function() {
    let password =
document.getElementById("password").va
lue;
    if (password.length < 6) {

document.getElementById("result").text
Content = "Password too short (Minimum
6 characters)";
    } else {

document.getElementById("result").text
Content = "Password accepted";
    }

});
```

OUTPUT





Part C: Local Storage Practice

Exercise 6: Storing Data

Task:

• Store a username in local storage.

• Retrieve and display it on page load.

CODE

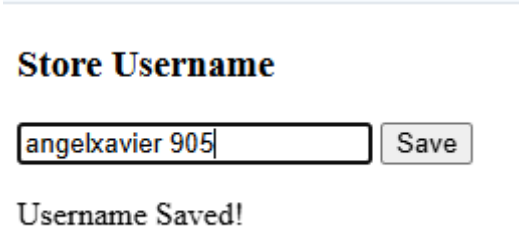
HTML

```
<!DOCTYPE html>
<html>
<head>
    <title>Local Storage Example</title>
</head>
<body>
<h3>Store Username</h3>
<input type="text" id="username" placeholder="Enter Username">
<button id="saveBtn">Save</button>
<p id="display"></p>
<script src="6.js"></script>
</body>
</html>
```

JS

```
// Save username to local storage
document.getElementById("saveBtn").addEventListener("click", function() {
    let user = document.getElementById("username").value;
    if (user !== "") {
localStorage.setItem("username", user);
document.getElementById("display").textContent = "Username Saved!";
    }
});
// Retrieve and display on page load
window.onload = function() {
    let savedUser = localStorage.getItem("username");
    if (savedUser) {
document.getElementById("display").textContent = "Welcome " + savedUser;
    }
};
```

OUTPUT



Exercise 7: Theme Preference

Task:

• Create a button.

• Toggle a CSS class on click.

• Save the preference in local storage

CODE

HTML

```
<!DOCTYPE html>
<html>
<head>
    <title>Theme Preference</title>
    <style>
        .dark {
            background-color: black;
            color: white;
        }
    </style>
</head>
<body>
<h3>Theme Toggle </h3>
<button id="themeBtn">Toggle Theme</button>
<script src="6.js"></script>
</body>
</html>
```

JS

```
let button = document.getElementById("themeBtn");
// Apply saved theme on page load
window.onload = function() {
    if (localStorage.getItem("theme") === "dark") {

document.body.classList.add("dark");
    }
};
// Toggle theme on button click
button.addEventListener("click", function() {

document.body.classList.toggle("dark");
    if (document.body.classList.contains("dark")) {
        localStorage.setItem("theme", "dark");
    } else {
        localStorage.setItem("theme", "light");
    }
});
```

OUTPUT