Pre-Lab Exercises – Experiment 6

JavaScript Form Validation, Local Storage & UI Control

Part A: Conceptual Exercises

Exercise 1: Form Validation Basics

What is form validation?
Form validation is the process of checking whether the user input in a form is correct, complete, and follows the required format before submitting the data.

Why is client-side validation important?
Client-side validation improves user experience by providing immediate feedback, reduces server load, and prevents incorrect data from being submitted.

Name any two form fields that require validation:

- Email address
- Phone number

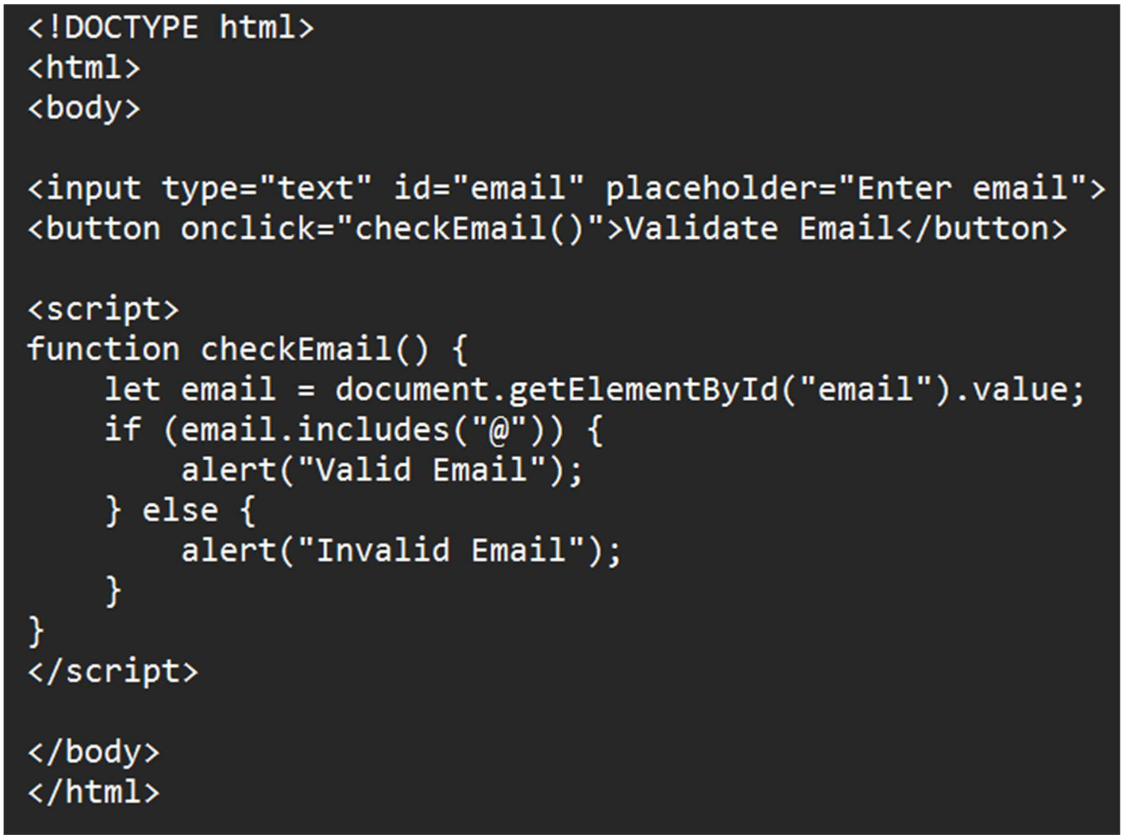
Exercise 2: Error Handling

How do error messages help users during form submission?
Error messages guide users by clearly indicating what input is incorrect or missing. They help users correct mistakes quickly and ensure accurate data submission.

Part B: Hands-On JavaScript Practice

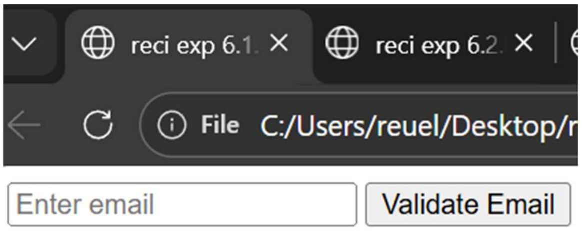
Exercise 3: Email Validation

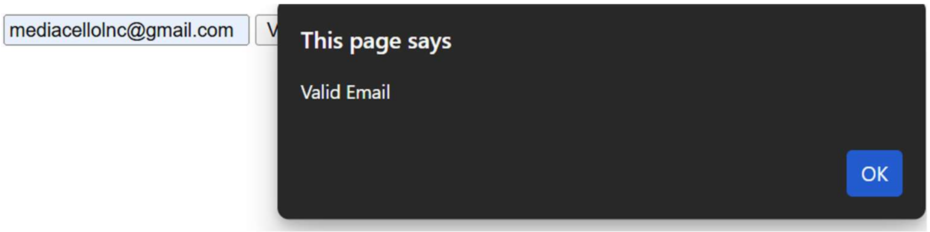
Code:



```
<!DOCTYPE html>
<html>
<body>

<input type="text" id="email" placeholder="Enter email">
<button onclick="checkEmail()">Validate Email</button>

<script>
function checkEmail() {
    let email = document.getElementById("email").value;
    if (email.includes("@")) {
        alert("Valid Email");
    } else {
        alert("Invalid Email");
    }
}
</script>

</body>
</html>
```

Output:





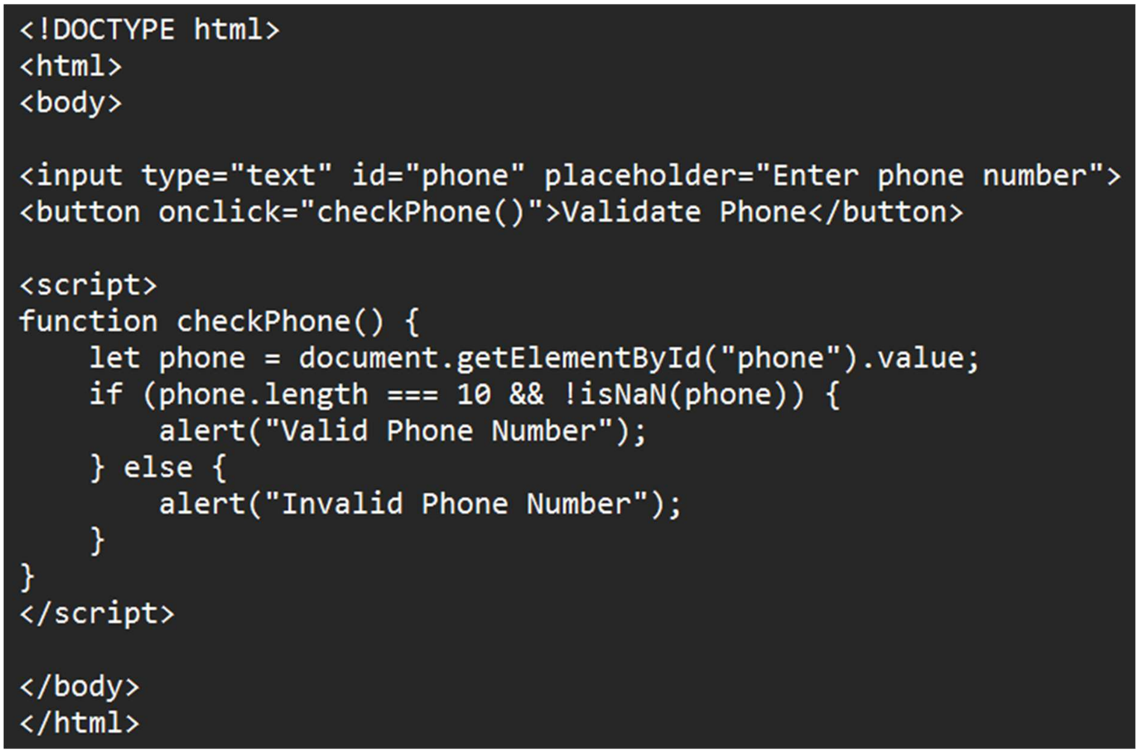
mediacellolnc@gmail.com

This page says

Valid Email

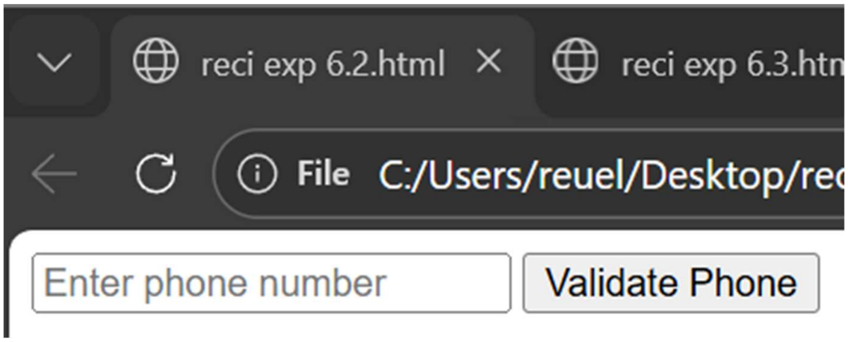
Exercise 4: Phone Number Validation

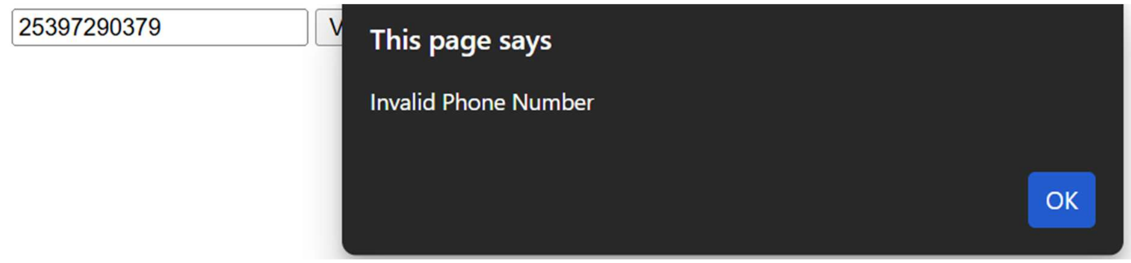
Code:



```
<!DOCTYPE html>
<html>
<body>

<input type="text" id="phone" placeholder="Enter phone number">
<button onclick="checkPhone()">Validate Phone</button>

<script>
function checkPhone() {
    let phone = document.getElementById("phone").value;
    if (phone.length === 10 && !isNaN(phone)) {
        alert("Valid Phone Number");
    } else {
        alert("Invalid Phone Number");
    }
}
</script>

</body>
</html>
```

Output:





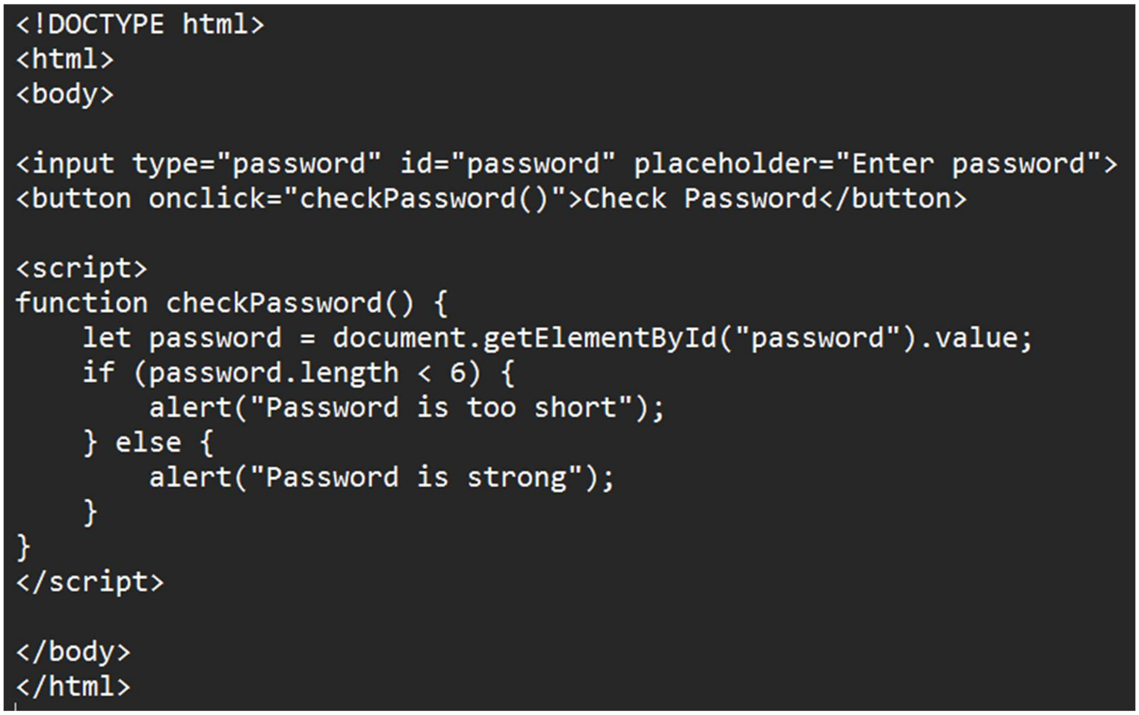
25397290379

This page says

Invalid Phone Number

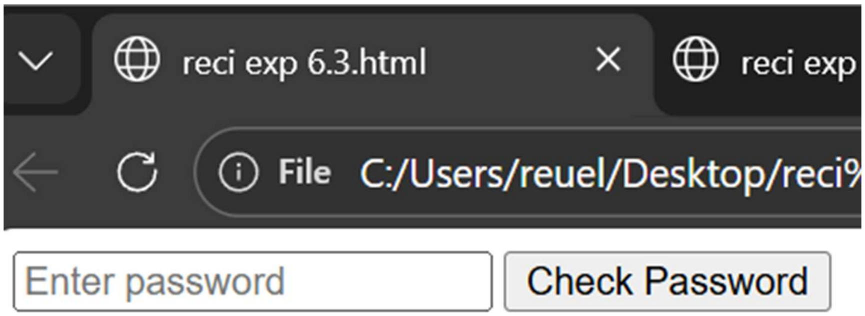
Exercise 5: Password Strength Check

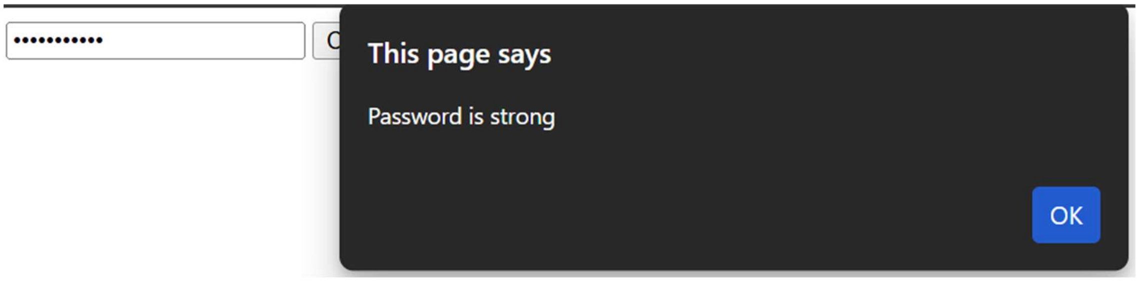
Code:



```
<!DOCTYPE html>
<html>
<body>

<input type="password" id="password" placeholder="Enter password">
<button onclick="checkPassword()">Check Password</button>

<script>
function checkPassword() {
    let password = document.getElementById("password").value;
    if (password.length < 6) {
        alert("Password is too short");
    } else {
        alert("Password is strong");
    }
}
</script>

</body>
</html>
```

Output:





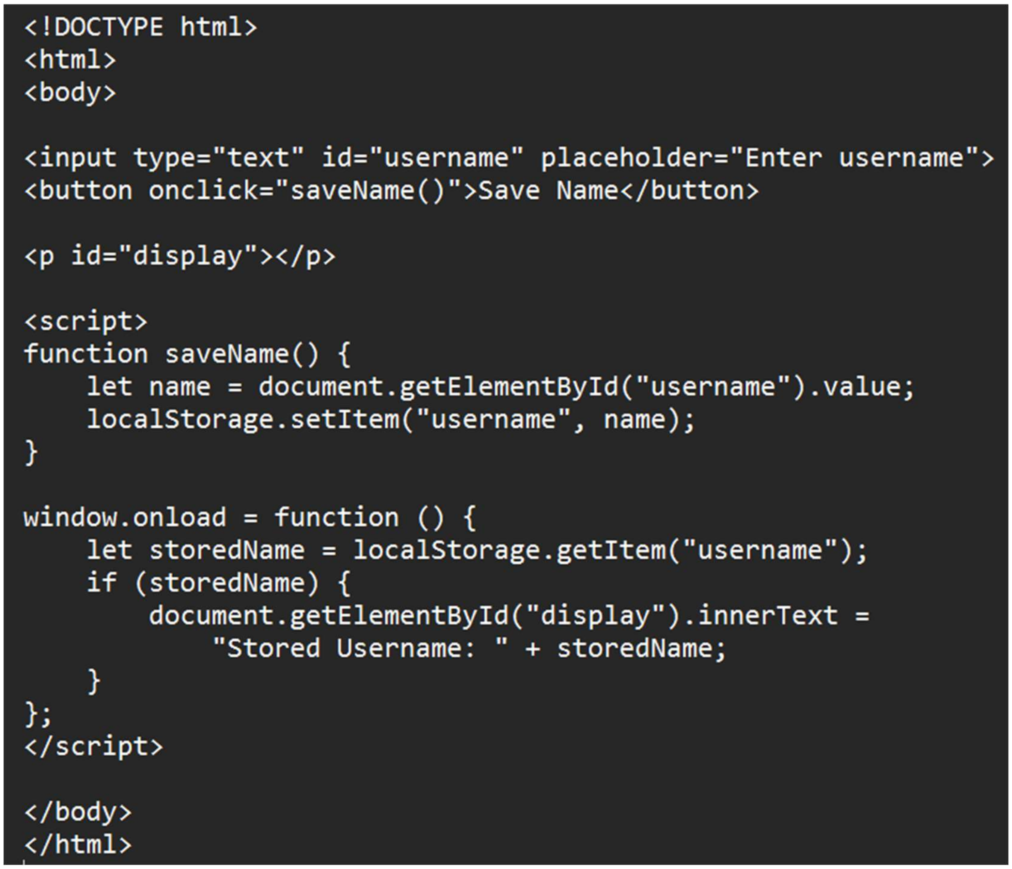
This page says

Password is strong

Part C: Local Storage Practice

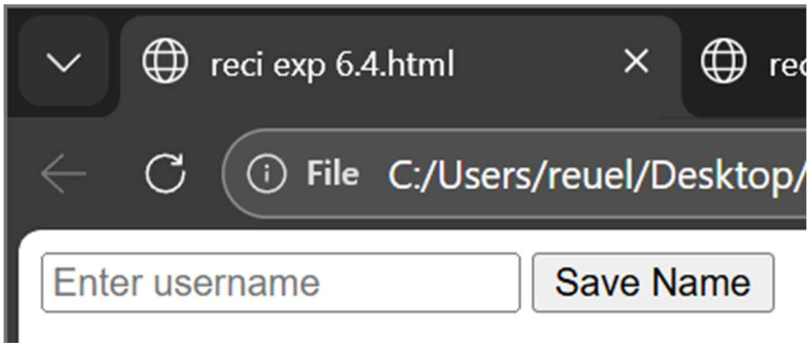
Exercise 6: Storing Data

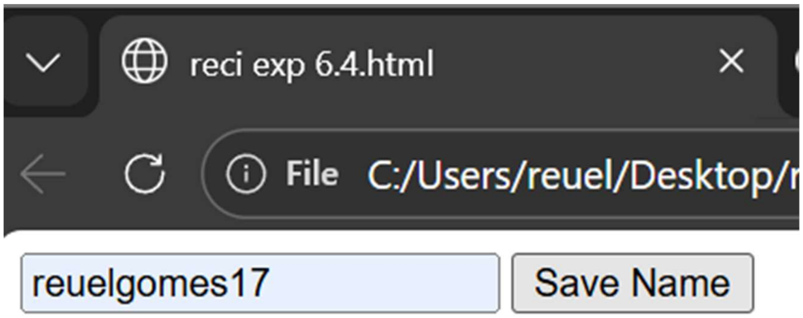
Code:



```
<!DOCTYPE html>
<html>
<body>

<input type="text" id="username" placeholder="Enter username">
<button onclick="saveName()">Save Name</button>

<p id="display"></p>

<script>
function saveName() {
    let name = document.getElementById("username").value;
    localStorage.setItem("username", name);
}

window.onload = function () {
    let storedName = localStorage.getItem("username");
    if (storedName) {
        document.getElementById("display").innerText =
            "Stored Username: " + storedName;
    }
};
</script>

</body>
</html>
```

Output:

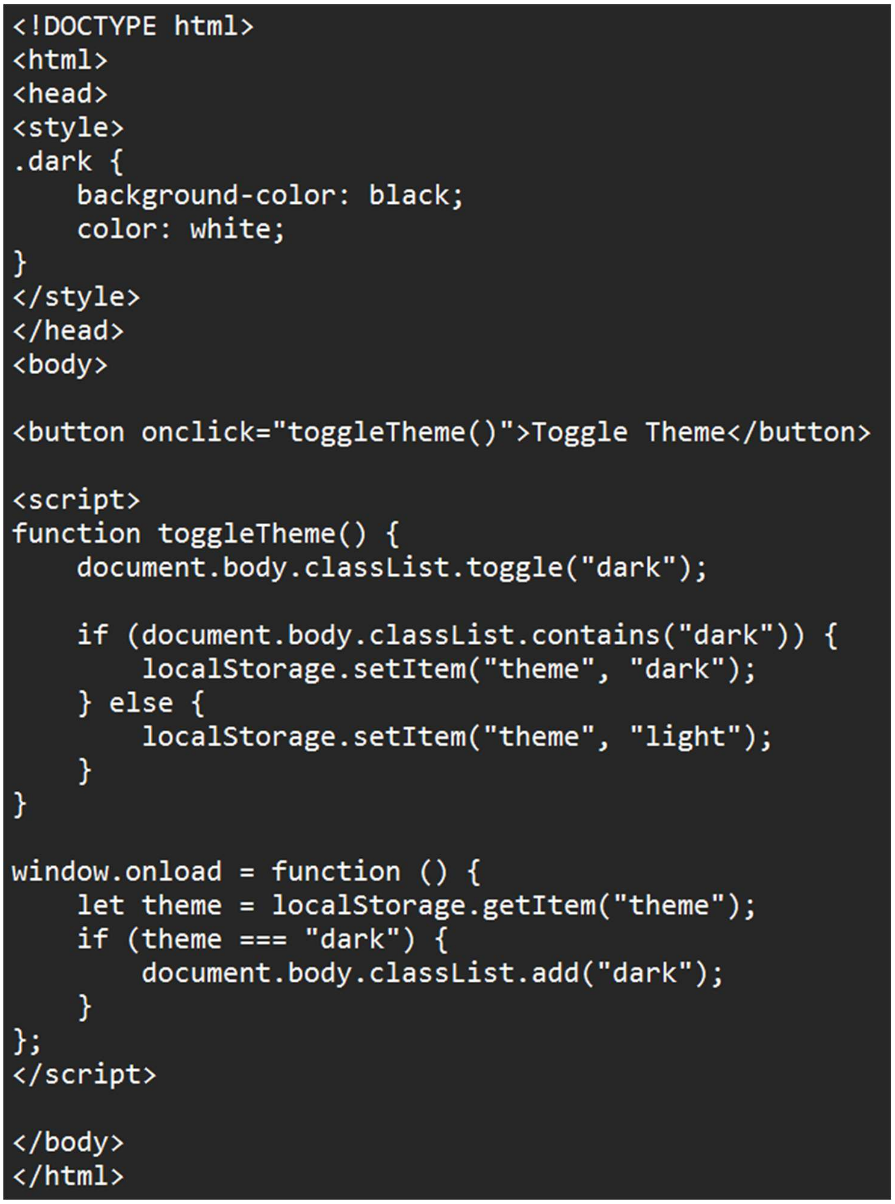




reuelgomes17

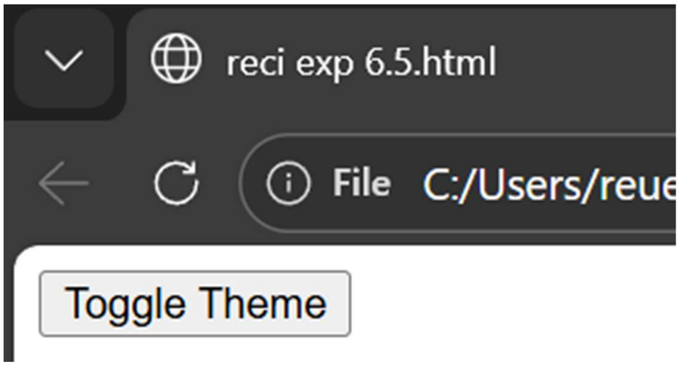
Exercise 7: Theme Preference

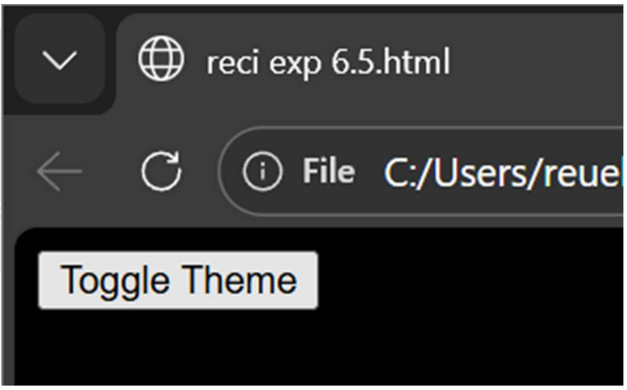
Code:



```
<!DOCTYPE html>
<html>
<head>
<style>
.dark {
    background-color: black;
    color: white;
}
</style>
</head>
<body>

<button onclick="toggleTheme()">Toggle Theme</button>

<script>
function toggleTheme() {
    document.body.classList.toggle("dark");

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

window.onload = function () {
    let theme = localStorage.getItem("theme");
    if (theme === "dark") {
        document.body.classList.add("dark");
    }
};
</script>

</body>
</html>
```

Output:


reci exp 6.5.html
File C:/Users/reue
Toggle Theme


reci exp 6.5.html
File C:/Users/reuel
Toggle Theme