# **EXP 2**

**AIM:** To implement JavaScript functionalities to manipulate the DOM and perform form validation.

**THEORY**

**1. Creating a Digital Clock on a Webpage**

A digital clock dynamically updates the time on a webpage using JavaScript’s setInterval function. The Date object is used to fetch the current time.

**2. Changing Background Color Every 5 Seconds**

This functionality involves modifying the webpage’s background color at intervals using JavaScript’s setInterval method. The color is randomly selected from a predefined array.

**3. Guess the Number Game**

A simple number guessing game where the user has limited attempts to guess a randomly generated number. JavaScript handles input validation and provides hints if the guess is too high or low.

**4. Implementing a Bulb ON/OFF Feature**

A program where clicking a button toggles the bulb’s state (on/off). JavaScript dynamically changes the image source and disables the clicked button until the other is pressed.

**5. Adding an Element to an Existing Layout**

Using JavaScript’s createElement() and appendChild() methods, a new element is added to the webpage dynamically when a button is clicked.

**6. Removing an Element from an Existing Layout**

JavaScript’s removeChild() method is used to delete an element from the DOM upon user interaction.

**7. Replacing an Element in an Existing Layout**

This involves replacing an existing DOM element with a new one using replaceChild().

**8. Form Validation with Highlighting Failed Fields**

JavaScript uses **Regular Expressions (Regex)** to validate form inputs. Invalid fields are highlighted using CSS by adding an invalid class dynamically.

**CODE IMPLEMENTATION**

**1. Digital Clock**

**HTML**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Digital Clock</title>

    <link rel="stylesheet" href="clock.css">

</head>

<body>

    <div class="clock-container">

        <div id="clock"></div>

    </div>

    <script>

        function updateClock() {

            const now = new Date();

            const timeString = now.toLocaleTimeString();

            document.getElementById('clock').textContent = timeString;

        }

        setInterval(updateClock, 1000);

        updateClock();

    </script>

</body>

</html>

**CSS**

body {

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

background-color: black;

color: white;

font-family: Arial, sans-serif;

}

.clock-container {

font-size: 3rem;

background: #222;

padding: 20px;

border-radius: 10px;

box-shadow: 0 0 10px white;

}

**2. Background Color Change**

**HTML**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Background Color Changer</title>

<link rel="stylesheet" href="background.css">

</head>

<body>

<div class="clock-container">

<div id="clock"></div>

</div>

<script>

function updateClock() {

const now = new Date();

const timeString = now.toLocaleTimeString();

document.getElementById('clock').textContent = timeString;

}

setInterval(updateClock, 1000);

updateClock();

function changeColor() {

const colors = ["red", "blue", "green", "yellow", "purple", "orange"];

document.body.style.backgroundColor = colors[Math.floor(Math.random() \* colors.length)];

}

setInterval(changeColor, 5000);

changeColor();

</script>

</body>

</html>

**CSS**

body {

height: 100vh;

display: flex;

justify-content: center;

align-items: center;

transition: background-color 1s ease-in-out;

font-family: Arial, sans-serif;

font-size: 2rem;

color: white;

}

.clock-container {

font-size: 3rem;

background: #222;

padding: 20px;

border-radius: 10px;

box-shadow: 0 0 10px white;

}

**3. Guess the Number Game**

**HTML**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Guess the Number</title>

<link rel="stylesheet" href="guess.css">

</head>

<body>

<div class="game-container">

<h1>Guess the Number</h1>

<p>Guess a number between 1 and 100:</p>

<input type="number" id="userGuess">

<button onclick="checkGuess()">Submit</button>

<p id="message"></p>

<p id="chances">Chances left: 10</p>

</div>

<script>

let randomNumber = Math.floor(Math.random() \* 100) + 1;

let attempts = 10;

function checkGuess() {

let userGuess = parseInt(document.getElementById('userGuess').value);

let message = document.getElementById('message');

let chances = document.getElementById('chances');

if (attempts > 1) {

if (userGuess === randomNumber) {

message.textContent = "Congratulations! You guessed the correct number.";

} else {

message.textContent = userGuess > randomNumber ? "Too high! Try again." : "Too low! Try again.";

attempts--;

chances.textContent = "Chances left: " + attempts;

}

} else {

message.textContent = "Game Over! The correct number was " + randomNumber;

chances.textContent = "No chances left.";

}

}

</script>

</body>

</html>

**CSS**

body {

text-align: center;

font-family: Arial, sans-serif;

background-color: #282c34;

color: white;

}

.game-container {

background: #444;

padding: 20px;

border-radius: 10px;

box-shadow: 0 0 10px white;

display: inline-block;

margin-top: 50px;

}

input {

padding: 10px;

font-size: 1rem;

}

button {

padding: 10px;

font-size: 1rem;

margin-left: 10px;

cursor: pointer;

background: lightblue;

border: none;

border-radius: 5px;

}

button:hover {

background: deepskyblue;

}

**4. Bulb ON/OFF**

**HTML**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Bulb ON/OFF</title>

<link rel="stylesheet" href="bulb.css">

</head>

<body>

<div class="container">

<h1>Light Bulb ON/OFF</h1>

<img id="bulb" src="https://www.w3schools.com/js/pic\_bulboff.gif" alt="Bulb">

<br>

<button id="onBtn" onclick="turnOn()">Turn ON</button>

<button id="offBtn" onclick="turnOff()">Turn OFF</button>

</div>

<script>

function turnOn() {

document.getElementById("bulb").src = "https://www.w3schools.com/js/pic\_bulbon.gif";

document.getElementById("onBtn").disabled = true;

document.getElementById("offBtn").disabled = false;

}

function turnOff() {

document.getElementById("bulb").src = "https://www.w3schools.com/js/pic\_bulboff.gif";

document.getElementById("offBtn").disabled = true;

document.getElementById("onBtn").disabled = false;

}

</script>

</body>

</html>

**CSS**

body {

text-align: center;

font-family: Arial, sans-serif;

background-color: #f4f4f4;

margin-top: 50px;

}

.container {

background: white;

padding: 20px;

border-radius: 10px;

box-shadow: 0 0 10px gray;

display: inline-block;

}

button {

padding: 10px 20px;

margin: 10px;

font-size: 1rem;

cursor: pointer;

border: none;

border-radius: 5px;

background: #007BFF;

color: white;

}

button:disabled {

background: gray;

cursor: not-allowed;

}

button:hover:enabled {

background: #0056b3;

}

**5.Add an Element**

**HTML**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Add Element</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<h1>Click to Add a New Element</h1>

<button onclick="addElement()">Add New Paragraph</button>

<div id="container"></div>

<script>

function addElement() {

let newPara = document.createElement("p");

newPara.textContent = "This is a newly added paragraph.";

document.getElementById("container").appendChild(newPara);

}

</script>

</body>

</html>

**6. Remove an Element**

**HTML**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Remove Element</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<h1>Click to Remove a Paragraph</h1>

<button onclick="removeElement()">Remove Last Paragraph</button>

<div id="container">

<p>This is paragraph 1.</p>

<p>This is paragraph 2.</p>

<p>This is paragraph 3.</p>

</div>

<script>

function removeElement() {

let container = document.getElementById("container");

if (container.lastElementChild) {

container.removeChild(container.lastElementChild);

}

}

</script>

</body>

</html>

**7. Replace an Element**

**HTML**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Replace Element</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<h1>Click to Replace an Element</h1>

<button onclick="replaceElement()">Replace Paragraph</button>

<div id="container">

<p id="oldPara">This is the original paragraph.</p>

</div>

<script>

function replaceElement() {

let newPara = document.createElement("p");

newPara.textContent = "This is the replaced paragraph!";

let oldPara = document.getElementById("oldPara");

oldPara.parentNode.replaceChild(newPara, oldPara);

}

</script>

</body>

</html>

**8. Form Validation**

**HTML**

<!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>

<link rel="stylesheet" href="form.css">

</head>

<body>

<div class="form-container">

<h1>Form Validation</h1>

<form id="myForm">

<label for="name">Name:</label>

<input type="text" id="name">

<span class="error" id="nameError"></span>

<label for="email">Email:</label>

<input type="text" id="email">

<span class="error" id="emailError"></span>

<label for="password">Password:</label>

<input type="password" id="password">

<span class="error" id="passwordError"></span>

<button type="button" onclick="validateForm()">Submit</button>

</form>

</div>

<script>

function validateForm() {

let name = document.getElementById("name");

let email = document.getElementById("email");

let password = document.getElementById("password");

let nameError = document.getElementById("nameError");

let emailError = document.getElementById("emailError");

let passwordError = document.getElementById("passwordError");

let nameRegex = /^[a-zA-Z\s]+$/;

let emailRegex = /^[a-zA-Z0-9.\_%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/;

let passwordRegex = /^(?=.\*\d)(?=.\*[a-z])(?=.\*[A-Z]).{6,}$/;

let valid = true;

// Name validation

if (!nameRegex.test(name.value)) {

name.classList.add("invalid");

nameError.textContent = "Name must contain only letters and spaces.";

valid = false;

} else {

name.classList.remove("invalid");

nameError.textContent = "";

}

// Email validation

if (!emailRegex.test(email.value)) {

email.classList.add("invalid");

emailError.textContent = "Enter a valid email address.";

valid = false;

} else {

email.classList.remove("invalid");

emailError.textContent = "";

}

// Password validation

if (!passwordRegex.test(password.value)) {

password.classList.add("invalid");

passwordError.textContent = "Password must be at least 6 characters long, contain a number, a lowercase, and an uppercase letter.";

valid = false;

} else {

password.classList.remove("invalid");

passwordError.textContent = "";

}

if (valid) {

alert("Form submitted successfully!");

}

}

</script>

</body>

</html>

**CSS**

body {

text-align: center;

font-family: Arial, sans-serif;

background-color: #f4f4f4;

}

.form-container {

background: white;

padding: 20px;

border-radius: 10px;

box-shadow: 0 0 10px gray;

display: inline-block;

margin-top: 50px;

text-align: left;

}

label {

display: block;

margin-top: 10px;

font-weight: bold;

}

input {

width: 100%;

padding: 10px;

margin-top: 5px;

border: 1px solid #ccc;

border-radius: 5px;

font-size: 1rem;

}

button {

padding: 10px 20px;

margin-top: 15px;

font-size: 1rem;

cursor: pointer;

border: none;

border-radius: 5px;

background: #007BFF;

color: white;

width: 100%;

}

button:hover {

background: #0056b3;

}

.error {

color: red;

font-size: 0.9rem;

}

.invalid {

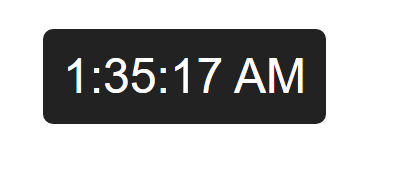
border-color: red;

background-color: #ffd1d1;

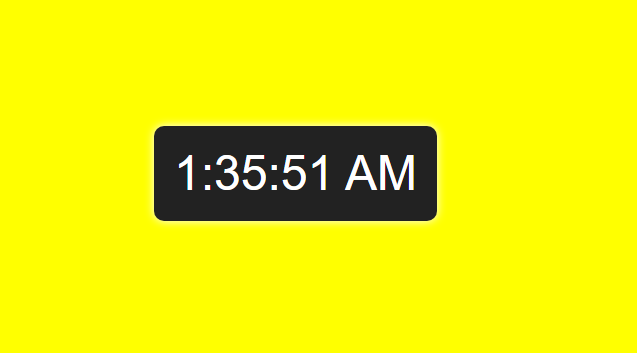
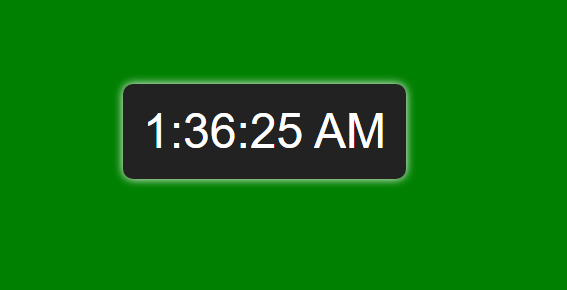
}

**OUTPUT**

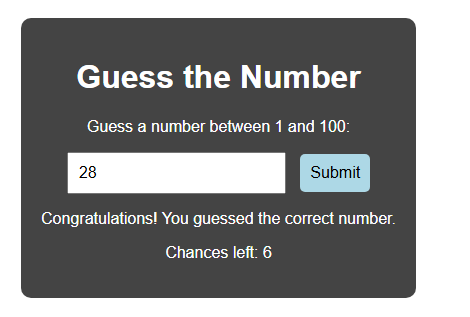
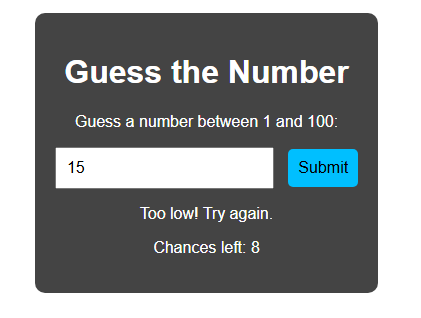
**1. Digital Clock**



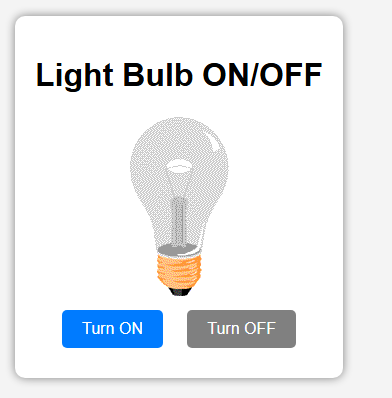
**2. Background Color Change**

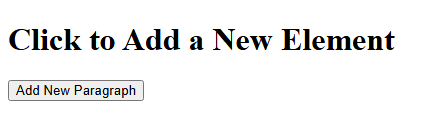
**3. Guess the Number Game**



**4. Bulb ON/OFF**

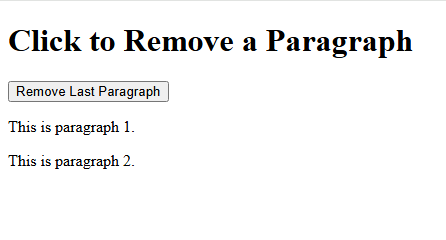
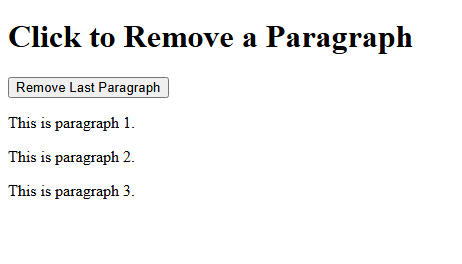


**5.Add an Element**

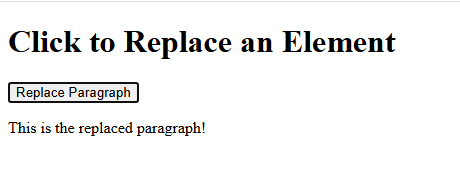
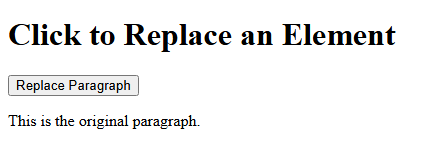




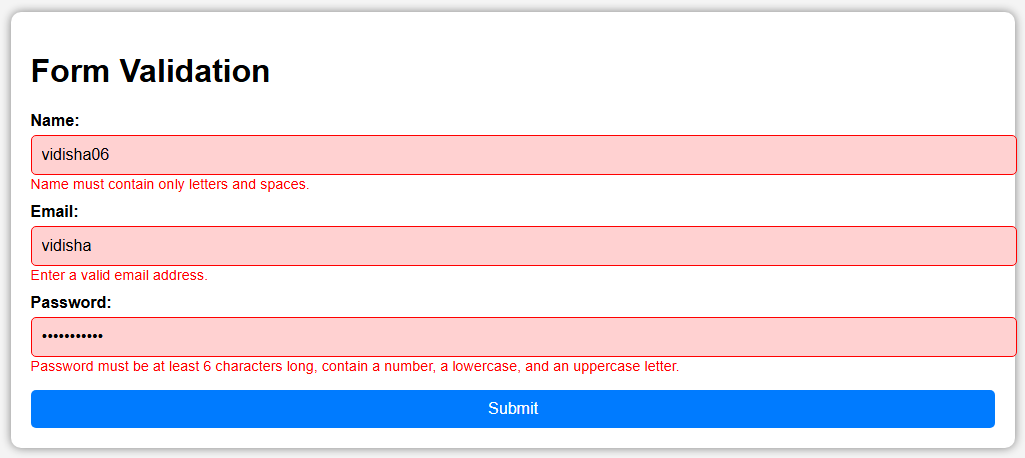
**6. Remove an Element**



**7. Replace an Element**



**8. Form Validation**



**CONCLUSION**

In this experiment, JavaScript was used to manipulate webpage elements dynamically. The programs demonstrated adding, removing, and replacing elements, toggling a bulb’s state, changing background colors, implementing a guessing game, and validating a form using regular expressions. This highlights the power of JavaScript in interactive web development.