**EXP NO: 9**

**Design input forms that validate data (e.g., email, phone number) and display error messages using HTML/CSS, JavaScript (with Validator.js)**

**To create a simple form that accepts an email and phone number from the user, and validates the inputs using JavaScript and the Validator.js library, displaying error messages for incorrect inputs and a success message for valid inputs**

**This was built using HTML to structure the form elements , CSS for styling the form layout , JavaScript to add validation logic and Validator.js which is a JavaScript library used to validate user input formats**

**Step 1: HTML Form Setup**

<!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="style.css">

</head>

<body>

<div class="container">

<form id="myForm">

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

<input type="email" id="email" name="email" required>

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

<label for="phone">Phone Number:</label>

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

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

<button type="submit">Submit</button>

</form>

</div>

<script

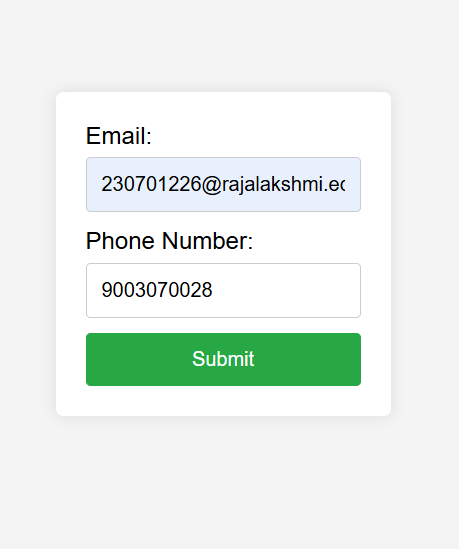
src="https://cdnjs.cloudflare.com/ajax/libs/validator/13.6.0/validator.min.js"></script>

<script src="script.js"></script>

</body>

</html>**Bottom of Form**

1. Created a form with two fields: **Email** and **Phone Number**.
2. Each field has a <label>, an <input> element, and a <span> tag for error display.
3. A **Submit** button is added to trigger validation.



**Step 2: CSS Styling**

body {

font-family: Arial, sans-serif;

background-color: #f4f4f4;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

margin: 0;

}

.container {

background-color: white;

padding: 20px;

border-radius: 5px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

form {

display: flex;

flex-direction: column;

}

label {

margin-bottom: 5px;

}

input {

margin-bottom: 10px;

padding: 10px;

border: 1px solid #ccc;

border-radius: 3px;

}

button {

padding: 10px;

background-color: #28a745;

color: white;

border: none;

border-radius: 3px;

cursor: pointer;

}

button:hover {

background-color: #218838;

}

.error {

color: red;

font-size: 0.875em;

}

Used flex display to center the form on the screen.

1. Designed a clean and minimal interface using:
2. Shadows for the card container
3. Rounded borders for inputs
4. Green color for the submit button
5. Error messages are shown in **red** using the .error class.

**Step 3: JavaScript + Validator.js for Validation**

document.getElementById('myForm').addEventListener('submit', function (e) {

e.preventDefault();

let email = document.getElementById('email').value;

let phone = document.getElementById('phone').value;

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

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

emailError.textContent = '';

phoneError.textContent = '';

if (!validator.isEmail(email)) {

emailError.textContent = 'Please enter a valid email address.';

}

if (!validator.isMobilePhone(phone, 'any')) {

phoneError.textContent = 'Please enter a valid phone number.';

}

if (validator.isEmail(email) && validator.isMobilePhone(phone, 'any')) {

console.log('Email:', email);

console.log('Phone:', phone);

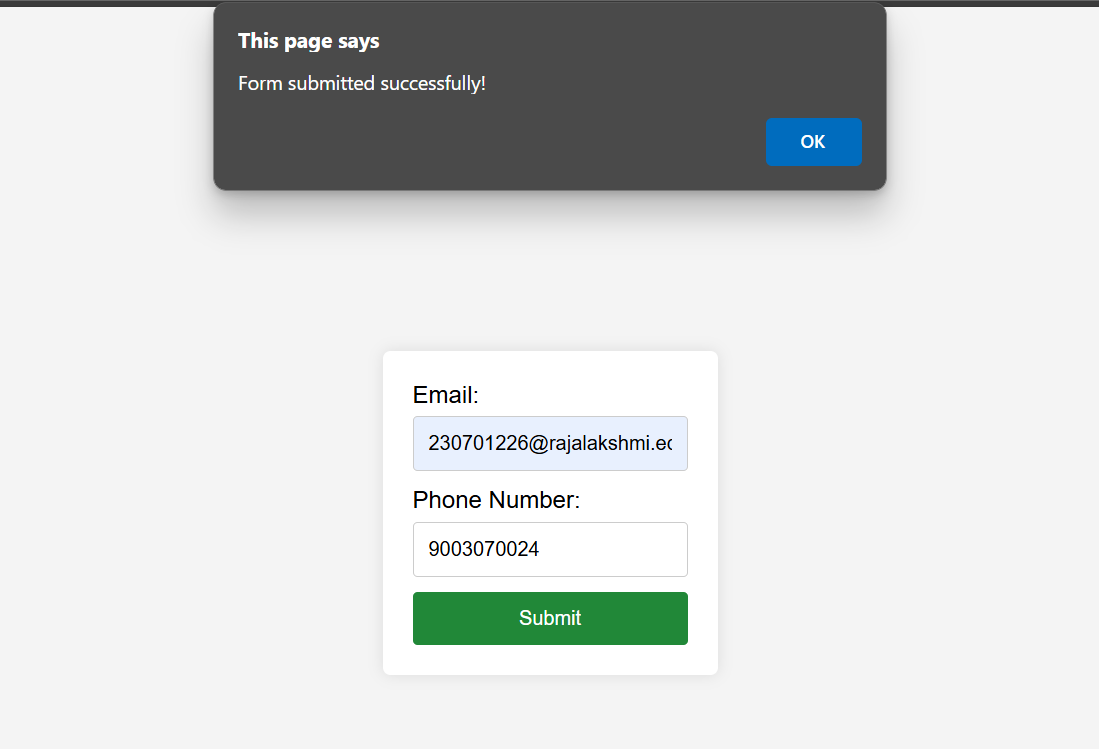
}

});

On form submission, we:

* 1. **Prevent default submission**
  2. **Grab values** from input fields
  3. Use validator.isEmail() to check if the email is in correct format
  4. Use validator.isMobilePhone() to validate the phone number
  5. Show respective error messages if validation fails
  6. If both validations pass, show a popup message using alert() that says:

“Form submitted successfully!”



The form validation project successfully validates email and phone number inputs using HTML, CSS, and JavaScript with Validator.js. It displays real-time error messages for invalid inputs and confirms successful submission with a pop-up alert. This enhances user experience by ensuring correct data entry before processing.