Experiment 9

Form Validation Using HTML, CSS, and JavaScript

Objective: To design a responsive and interactive web form that validates user input, specifically email addresses and phone numbers, utilizing HTML for structure, CSS for styling, and JavaScript integrated with the Validator.js library for input validation.

Tools and Technologies Used:

- **HTML5** – For structuring the input form
- **CSS3** – For styling and layout
- **JavaScript (ES6)** – For implementing validation logic
- **Validator.js** – A third-party JavaScript library for robust validation of string inputs

Procedure:

1. HTML Structure: An HTML form is created with input fields for an email address and phone number. The structure includes associated labels and span elements for displaying dynamic error messages.

```
<!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>
```

2. CSS Styling: Basic styling is applied to enhance the user interface. The form is centered and styled with modern UI principles for readability and user experience.

```
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;
}
```

3. JavaScript Validation Logic:

JavaScript intercepts the form submission, validates both fields using Validator.js, and displays errors if needed.

```
javascript
CopyEdit
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);
  }
});
```

Expected Output:

- **Valid inputs**:
    - No error messages.

    - Console logs:

makefile

CopyEdit

Email: test@example.com

Phone: 9876543210

- **Invalid inputs**:
    - Red error messages under respective fields.

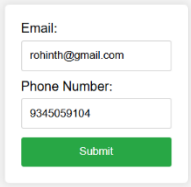
Examples:

- ✅ Email: user@example.com, Phone: 9876543210 → Valid
- ❌ Email: user.com → Shows: *"Please enter a valid email address."*
- ❌ Phone: abc123 → Shows: *"Please enter a valid phone number."*

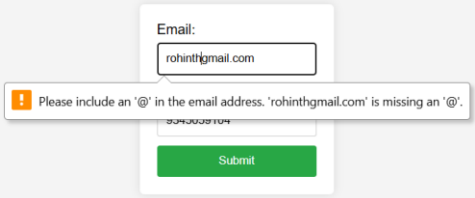
Conclusion:

This experiment demonstrates how to build a client-side form that validates user inputs for email and phone number, ensuring data integrity and providing immediate user feedback. It highlights best practices in modern web development and improves the usability of online forms using Validator.js.

Output



Output



Output