

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)

AIM:

The aim is to design input forms that validate data, such as email and phone number, and display error messages using HTML/CSS and JavaScript with Validator.js.

PROCEDURE:

Index.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="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>
```

style.css

```
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;  
}
```

```
script.js  
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');
```

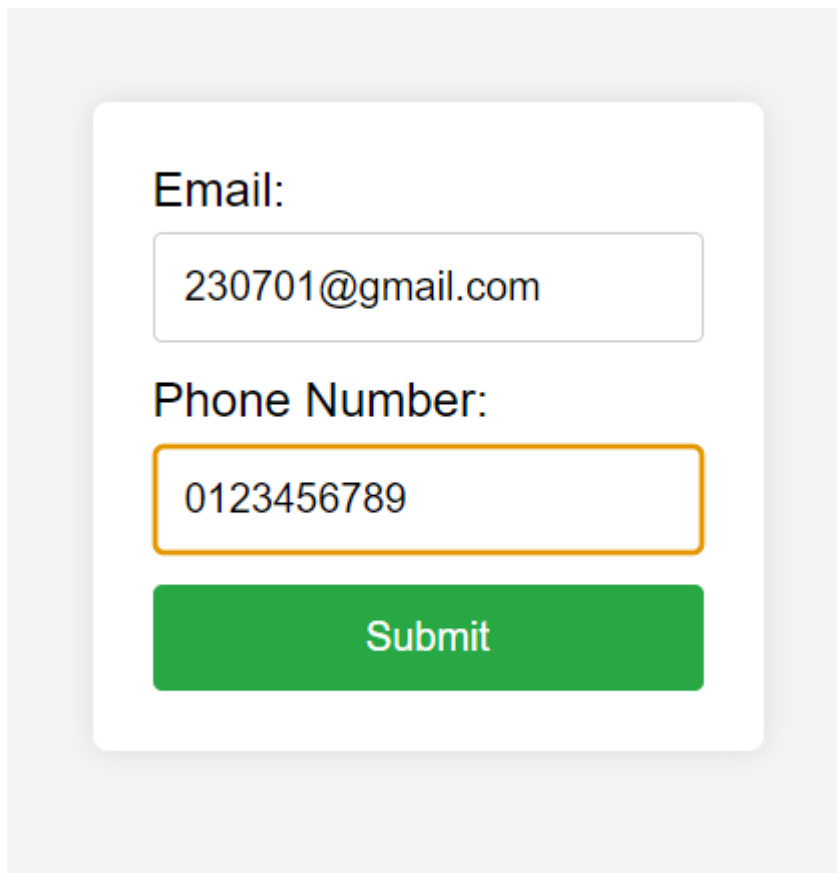
```
  // Clear previous error messages  
  emailError.textContent = "";  
  phoneError.textContent = "";
```

```
  // Validate email  
  if (!validator.isEmail(email)) {  
    emailError.textContent = 'Please enter a valid email address.';  
  }
```

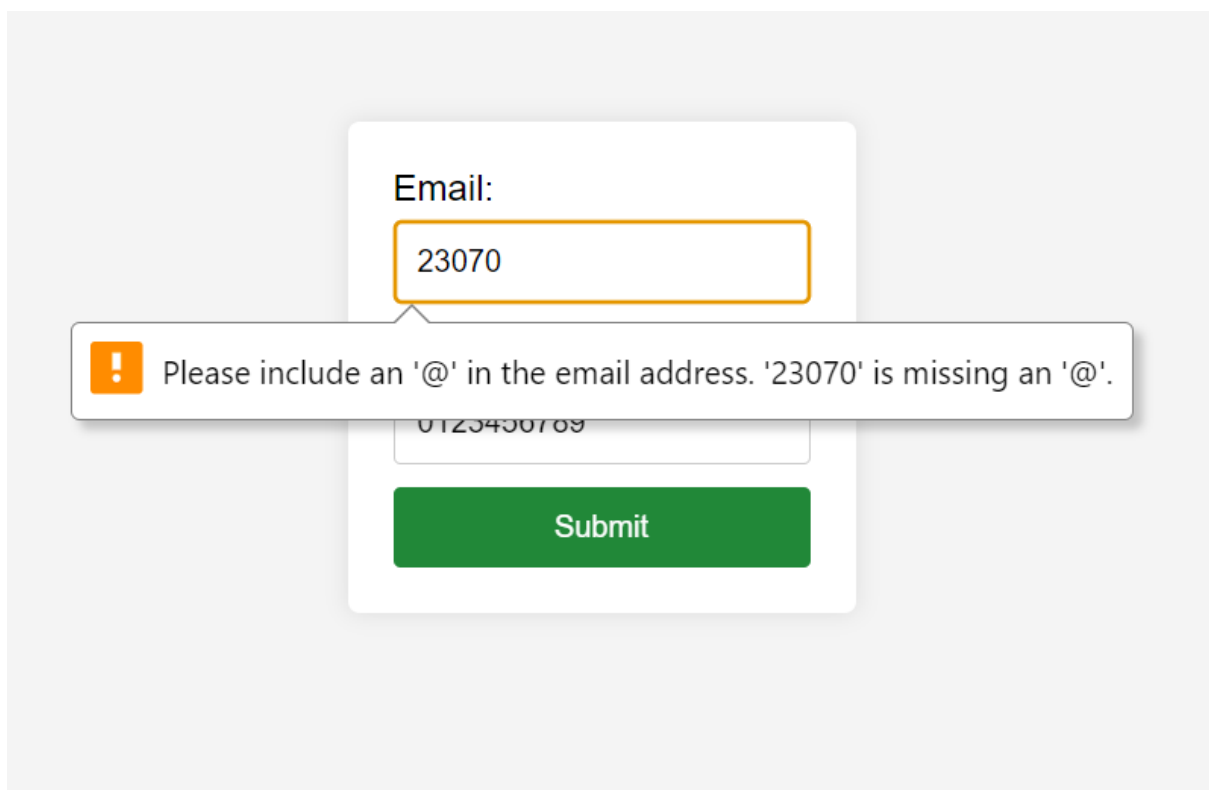
```
  // Validate phone number  
  if (!validator.isMobilePhone(phone, 'any')) {  
    phoneError.textContent = 'Please enter a valid phone number.';  
  }
```

```
  // If valid, log to console  
  if (validator.isEmail(email) && validator.isMobilePhone(phone, 'any')) {  
    console.log('Email:', email);  
    console.log('Phone:', phone);  
    alert('Form submitted successfully!');  
  }  
});
```

OUTPUT:



A form with a white background and rounded corners, set against a light gray background. It contains two input fields and a submit button. The first input field is labeled "Email:" and contains the text "230701@gmail.com". The second input field is labeled "Phone Number:" and contains the text "0123456789". Below the input fields is a green button with the text "Submit".



A form with a white background and rounded corners, set against a light gray background. It contains two input fields and a submit button. The first input field is labeled "Email:" and contains the text "23070". The second input field is labeled "Phone Number:" and contains the text "0123456789". Below the input fields is a green button with the text "Submit". An error message is displayed below the email input field, indicating that the email address is invalid because it is missing an '@' symbol.

Please include an '@' in the email address. '23070' is missing an '@'.

Email:

Please enter a valid email address.

Phone Number:

Please enter a valid phone number.

Submit

RESULT:

The input form was successfully designed using HTML and styled with CSS. Validation of user inputs such as email and phone number was effectively implemented using JavaScript with the Validator.js library. The form displayed appropriate error messages for invalid inputs and allowed submission only when all data was valid, thereby ensuring accurate and user-friendly data entry.