

EXPERIMENT-9

ROLL NO: 230701311

NAME: SHRADHA S

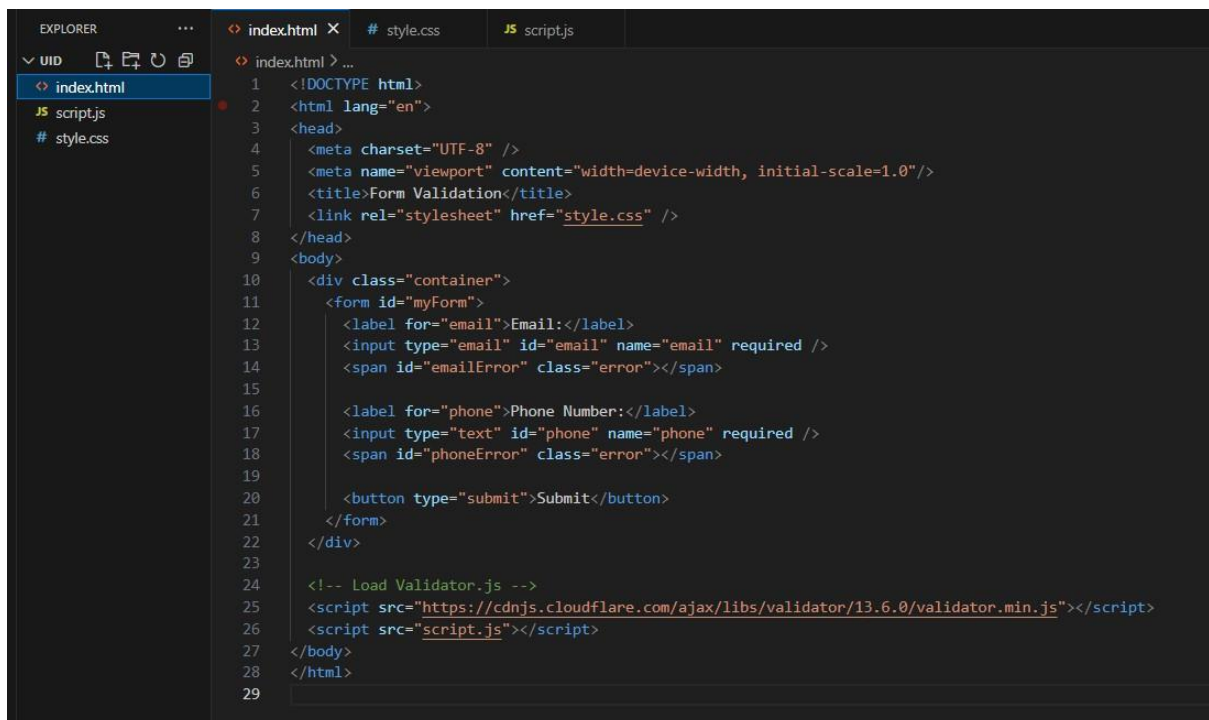
Design input forms that validate data (e.g., email, phone number) and display error messages using HTML/CSS, JavaScript

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.

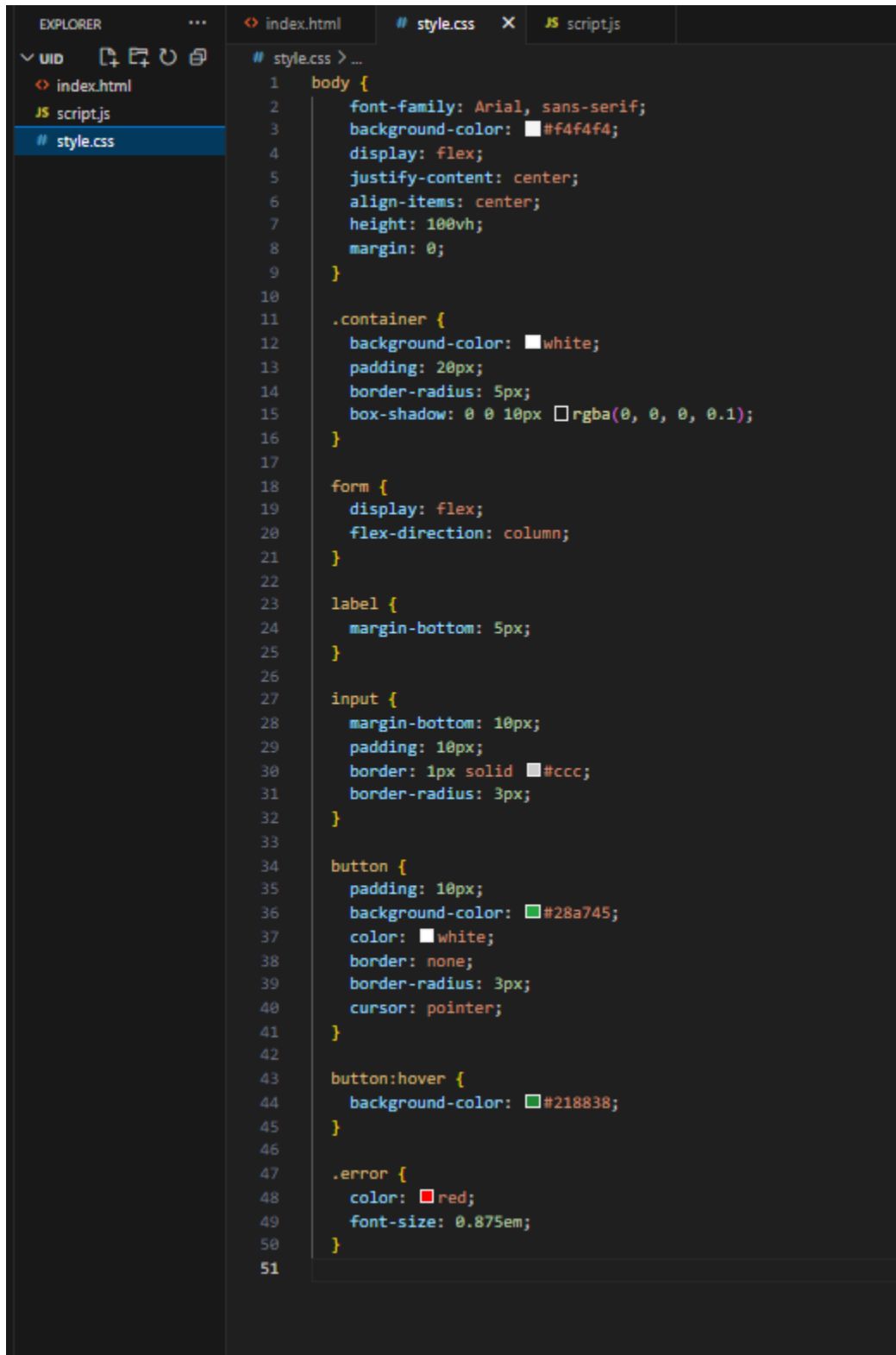
PROCEDURE:

Step 1: Setting Up the HTML Form Start by creating an HTML form with input fields for the email and phone number



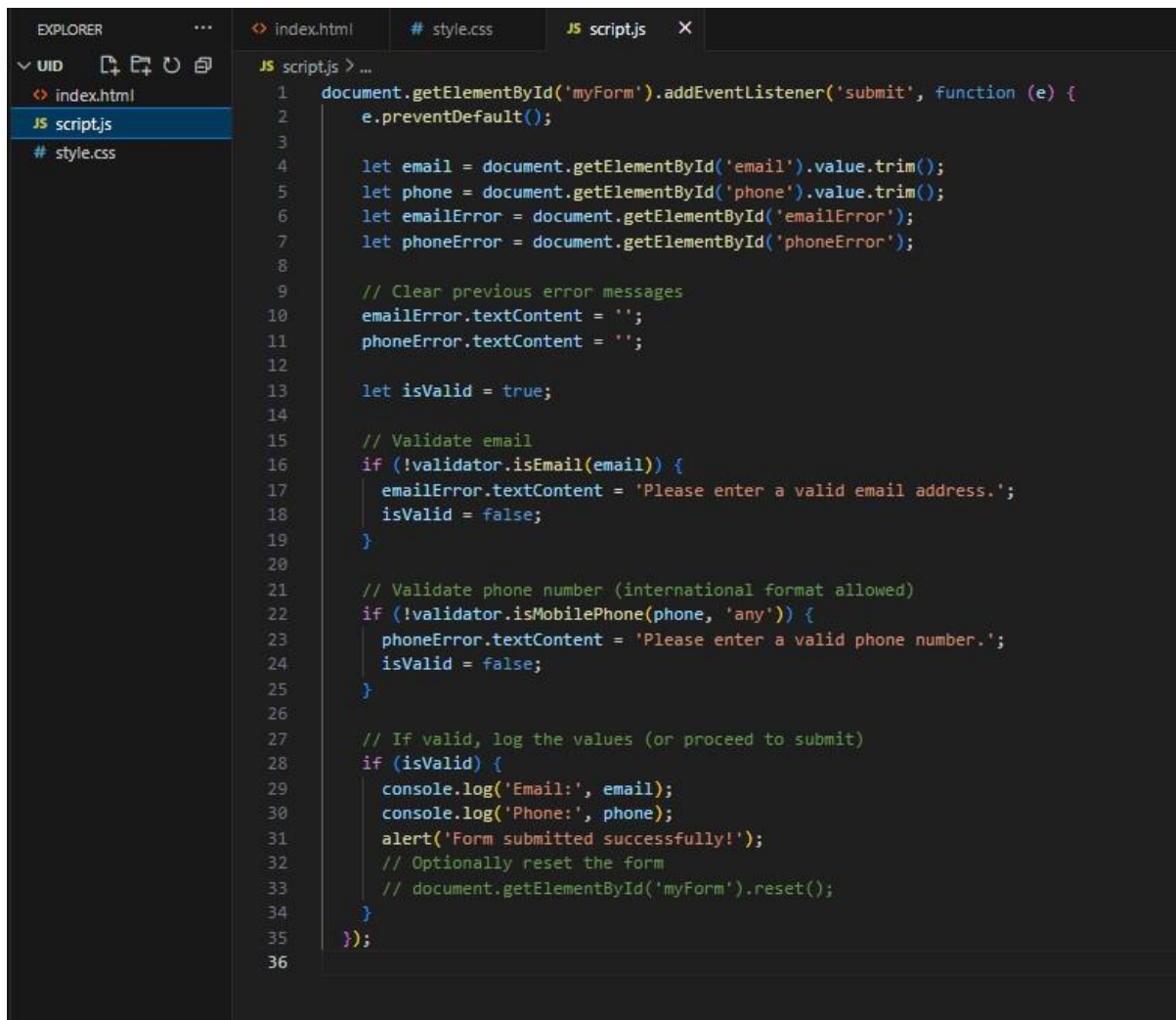
```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8" />
5   <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
6   <title>Form Validation</title>
7   <link rel="stylesheet" href="style.css" />
8 </head>
9 <body>
10  <div class="container">
11    <form id="myForm">
12      <label for="email">Email:</label>
13      <input type="email" id="email" name="email" required />
14      <span id="emailError" class="error"></span>
15
16      <label for="phone">Phone Number:</label>
17      <input type="text" id="phone" name="phone" required />
18      <span id="phoneError" class="error"></span>
19
20      <button type="submit">Submit</button>
21    </form>
22  </div>
23
24  <!-- Load Validator.js -->
25  <script src="https://cdn.jsdelivr.net/npm/validator@13.6.0/validator.min.js"></script>
26  <script src="script.js"></script>
27 </body>
28 </html>
29
```

Step 2: Styling the Form with CSS Next, add some basic styling to make the form look nice.

A screenshot of the Visual Studio Code editor interface. The Explorer sidebar on the left shows a project named 'UID' with files 'index.html', 'script.js', and 'style.css'. The 'style.css' file is selected and its content is displayed in the main editor. The CSS code defines styles for the body, a container, form elements, and error messages. The body has a teal background and flex alignment. The container is white with a shadow. Form elements have a light gray border and rounded corners. Buttons are green with white text. Error messages are red. Line numbers 1 through 51 are visible on the left of the code editor.

```
1 body {
2   font-family: Arial, sans-serif;
3   background-color: #f4f4f4;
4   display: flex;
5   justify-content: center;
6   align-items: center;
7   height: 100vh;
8   margin: 0;
9 }
10
11 .container {
12   background-color: white;
13   padding: 20px;
14   border-radius: 5px;
15   box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
16 }
17
18 form {
19   display: flex;
20   flex-direction: column;
21 }
22
23 label {
24   margin-bottom: 5px;
25 }
26
27 input {
28   margin-bottom: 10px;
29   padding: 10px;
30   border: 1px solid #ccc;
31   border-radius: 3px;
32 }
33
34 button {
35   padding: 10px;
36   background-color: #28a745;
37   color: white;
38   border: none;
39   border-radius: 3px;
40   cursor: pointer;
41 }
42
43 button:hover {
44   background-color: #218838;
45 }
46
47 .error {
48   color: red;
49   font-size: 0.875em;
50 }
51
```

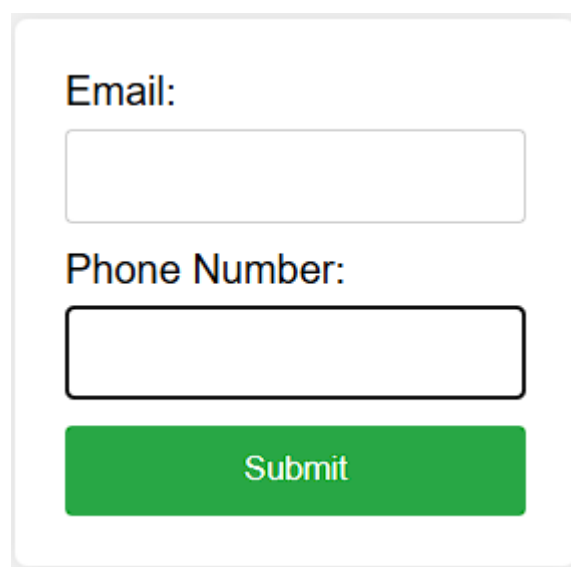
Step 3: Adding JavaScript for Validation Finally, add JavaScript to validate the input fields using Validator.js and display error messages.



The screenshot shows a code editor with three tabs: 'index.html', 'style.css', and 'script.js'. The 'script.js' tab is active, displaying the following JavaScript code:

```
1 document.getElementById('myForm').addEventListener('submit', function (e) {
2   e.preventDefault();
3
4   let email = document.getElementById('email').value.trim();
5   let phone = document.getElementById('phone').value.trim();
6   let emailError = document.getElementById('emailError');
7   let phoneError = document.getElementById('phoneError');
8
9   // Clear previous error messages
10  emailError.textContent = '';
11  phoneError.textContent = '';
12
13  let isValid = true;
14
15  // Validate email
16  if (!validator.isEmail(email)) {
17    emailError.textContent = 'Please enter a valid email address.';
18    isValid = false;
19  }
20
21  // Validate phone number (international format allowed)
22  if (!validator.isMobilePhone(phone, 'any')) {
23    phoneError.textContent = 'Please enter a valid phone number.';
24    isValid = false;
25  }
26
27  // If valid, log the values (or proceed to submit)
28  if (isValid) {
29    console.log('Email:', email);
30    console.log('Phone:', phone);
31    alert('Form submitted successfully!');
32    // Optionally reset the form
33    // document.getElementById('myForm').reset();
34  }
35 });
36
```

OUTPUT1:



The image shows a simple form UI with a light gray border. It contains two text input fields and a green submit button. The first field is labeled 'Email:' and the second is labeled 'Phone Number:'. The submit button is green with the text 'Submit' in white.

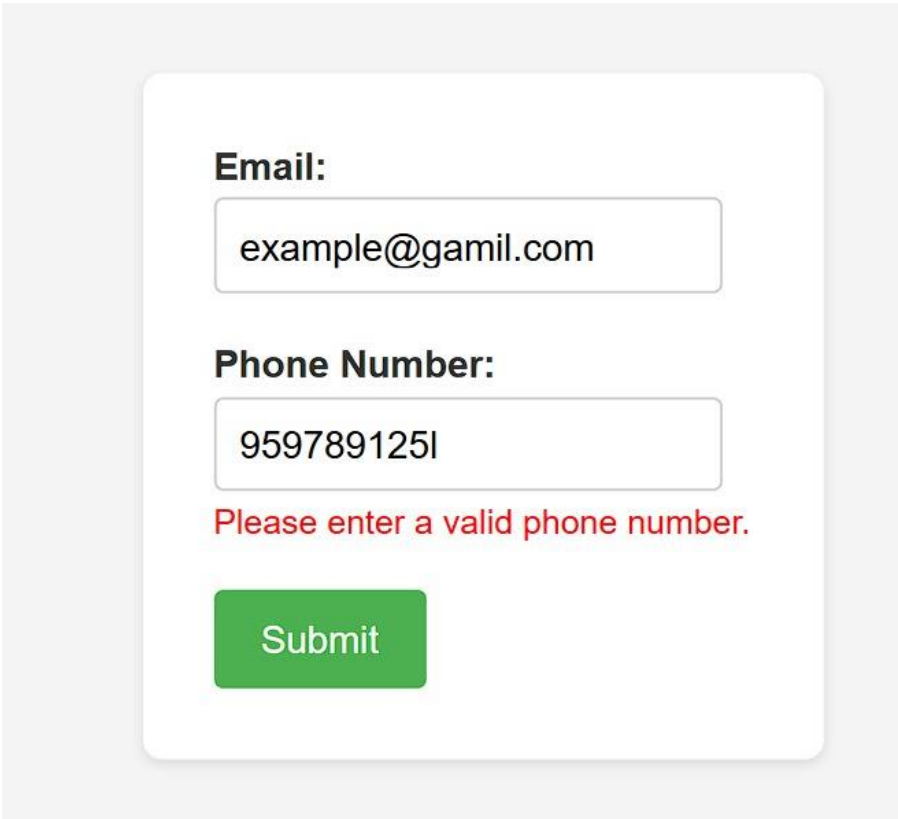
The image you provided shows a **simple form UI** containing:

- A label and input field for **Email**
- A label and input field for **Phone Number**
- A **Submit** button styled with a green background

When the HTML, CSS, and JavaScript code for this UI is executed:

- The **form will be displayed** in a white box with rounded corners and a subtle shadow (likely due to Bootstrap or custom styling).
- Users can type into the email and phone number fields.
- Pressing the green **Submit** button would trigger a JavaScript function, typically to **validate** and/or **send the form data**.

OUTPUT 2:



Email:

example@gamil.com

Phone Number:

959789125l

Please enter a valid phone number.

Submit

The image shows a form validation **error message**:

● **Error Message:**

"Please enter a valid phone number." (displayed in red)

< **Explanation:**

This message is likely triggered by **JavaScript form validation**. Here's what happened based on the image:

- **Email field** is correctly filled: example@gmail.com
- **Phone Number field** contains invalid input: "SHRADHA" (a name, not a phone number)

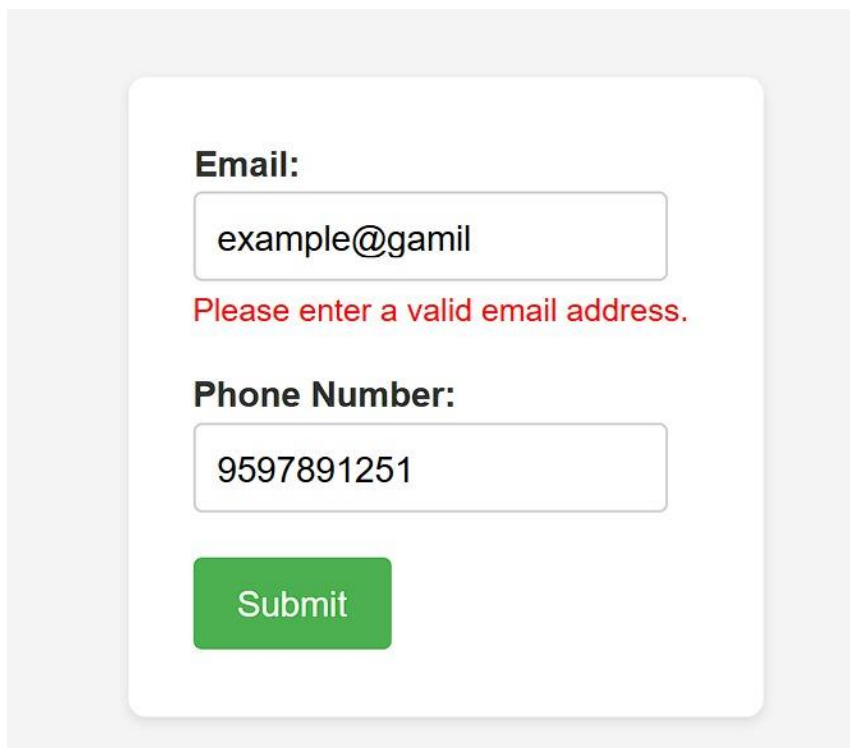
The JavaScript is checking if the input in the **Phone Number** field is a valid format (probably digits only and maybe length constraints, like 10 digits).

■ **Expected Input Format for Phone Number:**

Likely something like:

- 9876543210 (only digits)

OUTPUT 3:



Email:

Please enter a valid email address.

Phone Number:

Submit

● **Error Message:**

"Please enter a valid email address." (in red text)

< **Explanation:**

- **Email Field Input:** example@gmail.com

This input is **invalid** because it does not follow the standard format for email addresses.

■ **Valid Email Format:**

A proper email address should follow this pattern:

pgsql

CopyEdit

username@domain.extension

- Examples of valid formats:
 - example@gmail.com
 - user@example.in
 - john.doe@outlook.co.uk
- In your case, @com is not a valid domain. A domain like @gmail.com or @yahoo.com would be valid.

■ **Phone Number Field:**

- **Input:** 6369075295
- This looks valid (10-digit number), so **no error** is shown for the phone number.

OUTPUT 4:

■ Overview

This section documents the **successful submission** of a user input form that collects an email address and a phone number. The validation logic ensures that both fields meet basic syntactic criteria before allowing submission.

) Input Details

- **Email Entered:** rajamanishalini18@gmail.cor
 - **Phone Number Entered:** 6369075295
-

✈ Validation Checks

✉ Email Validation

- The system checks if the email:
 - Contains the @ symbol.
 - Has characters before and after the @.
 - Ends with a domain extension of **2–3 alphabetic characters** (e.g., .com, .net, .cor).
- In this case, rajamanishalini18@gmail.cor passes the format check, even though .cor is not a standard domain. The validation is based on syntax, not real-world domain existence.

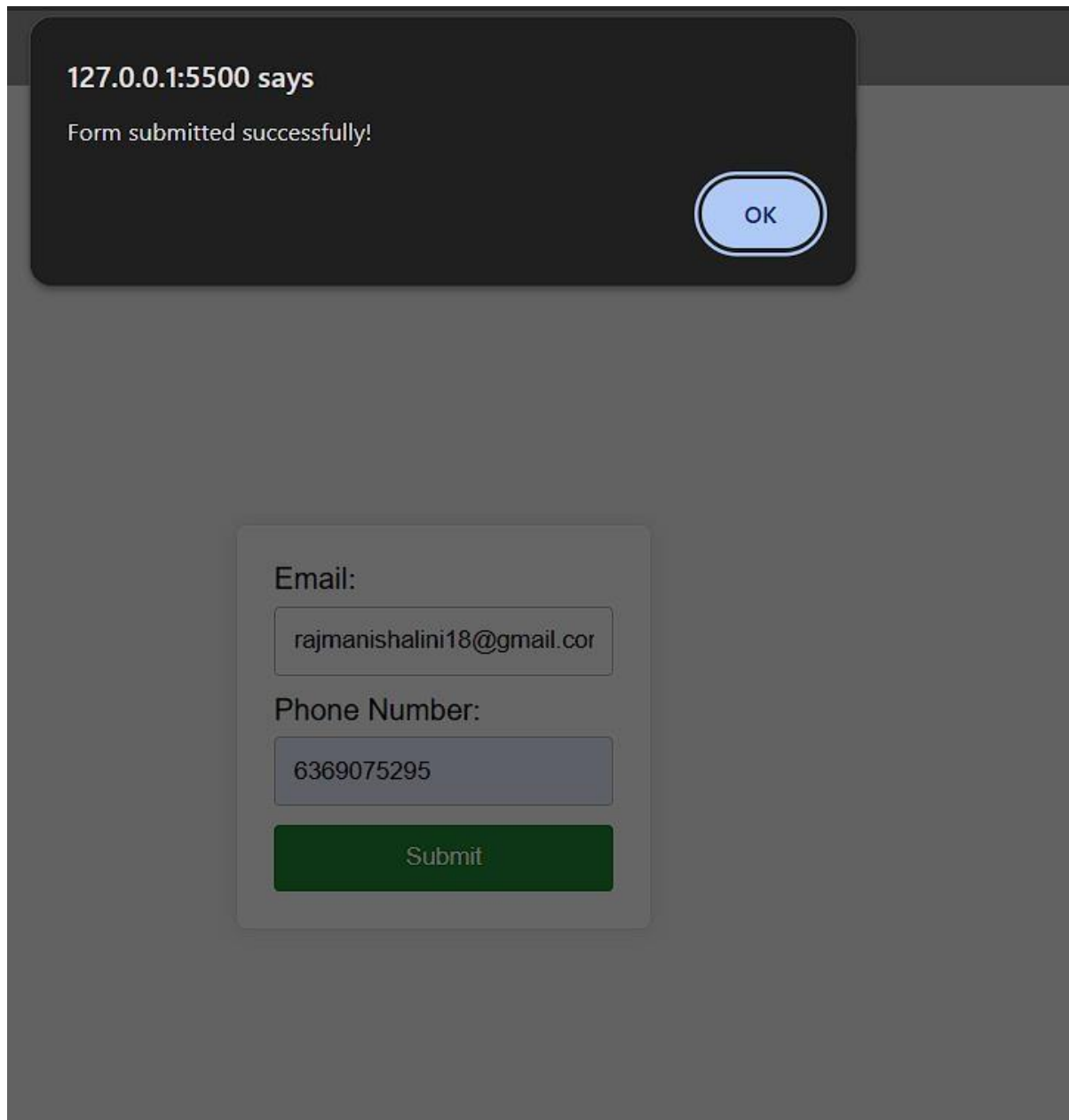
! Phone Number Validation

- The system ensures that the phone number:
 - Contains exactly **10 digits**.
 - Consists of only numeric characters (0–9).
 - The input 6369075295 satisfies these conditions.
-

■ Result

Since both fields pass their respective validations:

- No error messages are displayed.
- The form is submitted successfully.
- A browser alert appears with the message:
"Form submitted successfully!"



The image shows a browser window with a dark-themed background. At the top, a dark grey alert box is displayed, containing the text "127.0.0.1:5500 says" and "Form submitted successfully!". An "OK" button is located in the bottom right corner of the alert box. Below the alert box, a form is visible. The form has a light grey background and contains the following elements:

- Email:** A text input field containing the email address "rajmanishalini18@gmail.cor".
- Phone Number:** A text input field containing the phone number "6369075295".
- Submit:** A green button with the text "Submit".