

**UIT2602 WEB PROGRAMMING**  
**Exercise 2 HTML/ DHTML Form Validation**

**Name: S. Selcia**  
**Reg. No.: 3122215002098**  
**Section: IT B**

**1. Aim:**

The aim of this exercise is to design a website using HTML and DHTML to implement form validation using regular expressions and to perform HTML DOM manipulation.

**2. Required Web Tools and Methodology:**

- HTML: For structuring the web page and creating forms.
- CSS: For styling the web page and making it visually appealing.
- JavaScript: For implementing form validation using regular expressions and DOM manipulation.
- Web Browser: To test the functionality of the website.

**3. Implementation Procedure:**

1. Planning and Designing:
  - a. Decide on the layout and elements of the web page.
  - b. Plan the form fields and validation requirements.
2. HTML Structure:
  - a. Create the HTML structure using ``<form>``, ``<input>``, and ``<select>`` tags for form elements.
3. CSS Styling:
  - a. Style the form and other elements using CSS to make the website visually appealing.
4. JavaScript Implementation:
  - a. Write JavaScript functions for form validation using regular expressions.
  - b. Implement DOM manipulation to dynamically update the form elements.
5. Testing:
  - a. Test the website in different web browsers to ensure compatibility.
  - b. Verify form validation for different input scenarios.

**4. Code Snippets**

`index.html`

```

<html>
<head>
  <title>Form Validation</title>
  <link rel="stylesheet" href="styles.css">
  <script>
    function validateForm() {
      var name = document.getElementById('name').value;
      var creditCard = document.getElementById('creditCard').value;
      var cvv = document.getElementById('cvv').value;
      var expiryMonth = document.getElementById('expiryMonth').value;
      var expiryYear = document.getElementById('expiryYear').value;
      // Regular expressions for validation
      var nameRegex = /^[A-Za-z]+$/;
      var creditCardRegex = /^\d{4}-?\d{4}-?\d{4}-?\d{4}$/;
      var cvvRegex = /^\d{3,4}$/;
      // Validation checks
      if (name === "" || creditCard === "" || cvv === "" || expiryMonth === "" || expiryYear ===
"" ) {
        alert("Please fill in all fields");
        return false;
      }
      if (!name.match(nameRegex)) {
        alert("Name can only contain alphabets");
        return false;
      }
      if (!creditCard.match(creditCardRegex)) {
        alert("Invalid credit card number");
        return false;
      }
      if (!cvv.match(cvvRegex)) {
        alert("Invalid CVV");
        return false;
      }
      // Expiry date validation
      var currentDate = new Date();
      var currentYear = currentDate.getFullYear();
      var currentMonth = currentDate.getMonth() + 1; // Month is zero-indexed
      if (parseInt(expiryYear) < currentYear || (parseInt(expiryYear) === currentYear &&
parseInt(expiryMonth) < currentMonth)) {
        alert("Expiry date must be in the future");
        return false;
      }
      // If all validations pass, move focus to the next input field
      document.getElementById('cvv').focus();
      return true;
    }
  </script>
</head>

```

```
<body>
  <h2>Form Validation</h2>
  <form onsubmit="return validateForm()">
    <label for="name">Name:</label><br>
    <input type="text" id="name" name="name"><br><br>
    <label for="creditCard">Credit Card Number:</label><br>
    <input type="text" id="creditCard" name="creditCard"><br><br>
    <label for="cvv">CVV:</label><br>
    <input type="text" id="cvv" name="cvv"><br><br>
    <label for="expiryMonth">Expiry Month:</label>
    <select id="expiryMonth" name="expiryMonth">
      <option value="">--Select Month--</option>
      <option value="01">January</option>
      <option value="02">February</option>
      <option value="03">March</option>
      <option value="04">April</option>
      <option value="05">May</option>
      <option value="06">June</option>
      <option value="07">July</option>
      <option value="08">August</option>
      <option value="09">September</option>
      <option value="10">October</option>
      <option value="11">November</option>
      <option value="12">December</option>
    </select><br><br>
    <label for="expiryYear">Expiry Year:</label>
    <select id="expiryYear" name="expiryYear">
      <option value="">--Select Year--</option>
      <option value="2024">2024</option>
      <option value="2025">2025</option>
      <option value="2026">2026</option>
      <option value="2027">2027</option>
      <option value="2028">2028</option>
      <option value="2029">2029</option>
      <option value="2030">2030</option>
      <option value="2031">2031</option>
      <option value="2032">2032</option>
      <option value="2033">2033</option>
      <option value="2034">2034</option>
      <option value="2035">2035</option>
      <option value="2036">2036</option>
    </select><br><br>
    <input type="submit" value="Submit">
  </form>
</body>
</html>
```

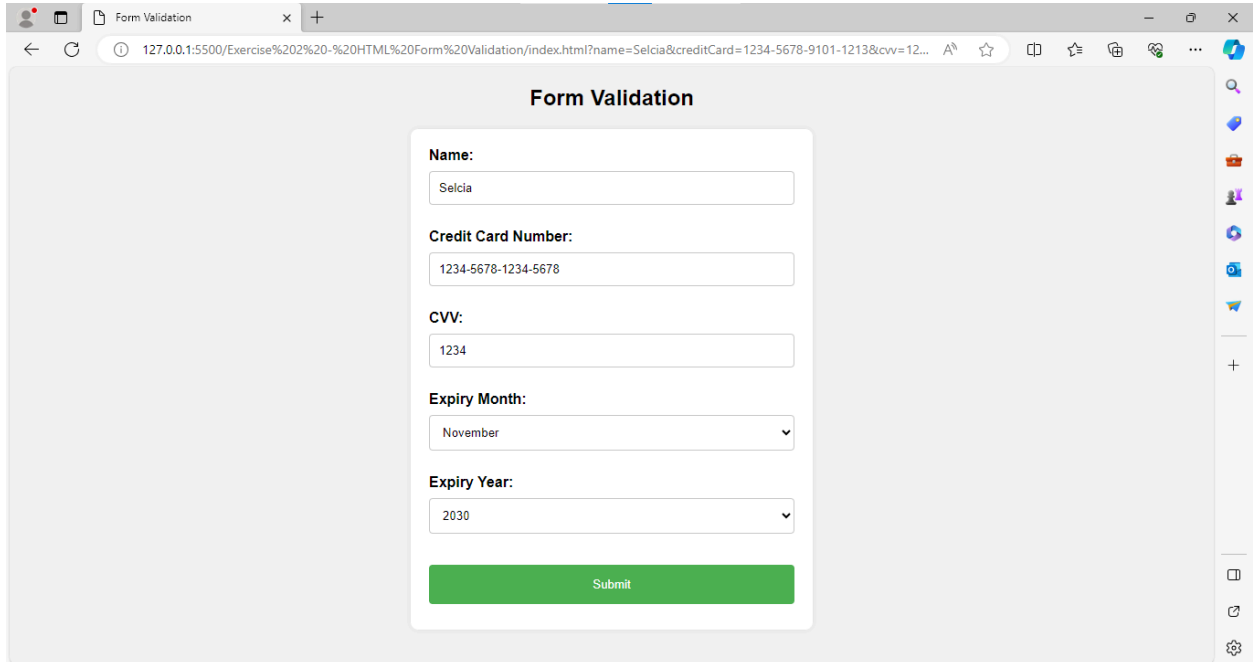
## styles.css

```
body {
  font-family: Arial, sans-serif;
  background-color: #f4f4f4;
}
h2 {
  text-align: center;
}
form {
  max-width: 400px;
  margin: 0 auto;
  background-color: #ffffff;
  padding: 20px;
  border-radius: 8px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}
label {
  font-weight: bold;
}
input[type="text"],
select {
  width: 100%;
  padding: 10px;
  margin: 8px 0;
  border: 1px solid #ccc;
  border-radius: 4px;
  box-sizing: border-box;
}
input[type="submit"] {
  width: 100%;
  background-color: #4caf50;
  color: white;
  padding: 14px 20px;
  margin: 8px 0;
  border: none;
  border-radius: 4px;
  cursor: pointer;
}
input[type="submit"]:hover {
  background-color: #45a049;
}
.error {
  color: red;
  font-size: 0.9em;
}
.success {
  color: green;
```

```
font-size: 0.9em;  
}
```

## 5. Output Screenshots:

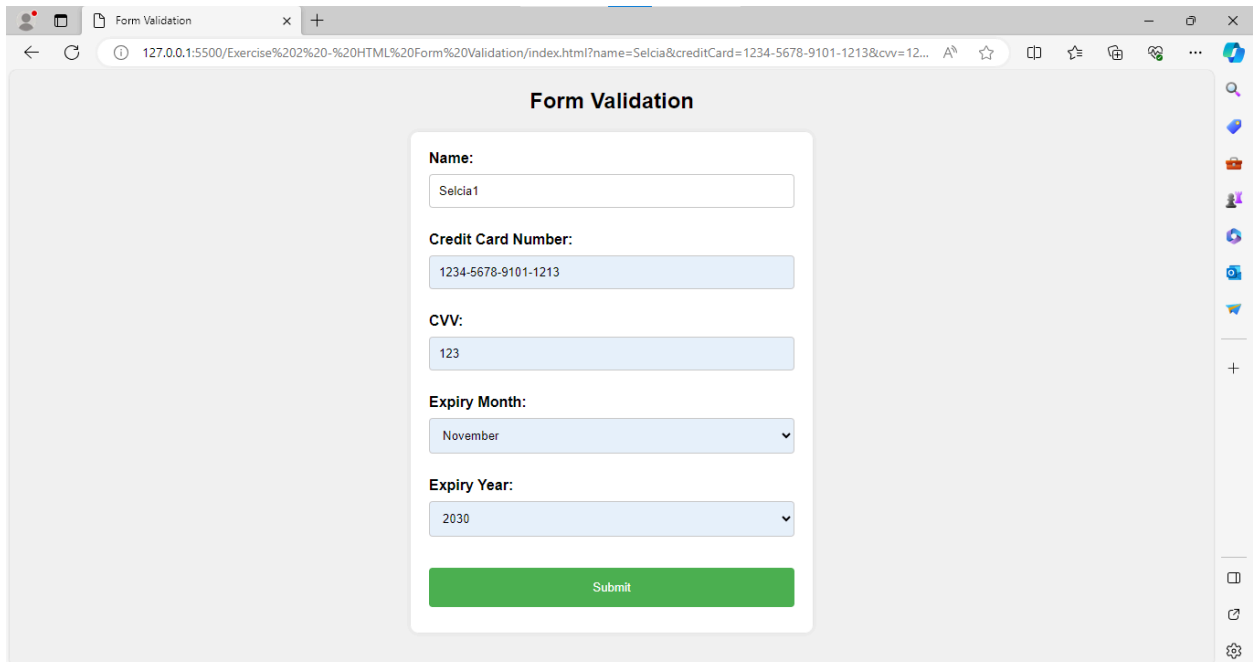
### Test Case 1 : All details entered correctly, no alert



The screenshot shows a web browser window with the title 'Form Validation'. The address bar displays the URL: `127.0.0.1:5500/Exercise%20%20-%20HTML%20Form%20Validation/index.html?name=Selcia&creditCard=1234-5678-9101-1213&cvv=1234&expiryMonth=November&expiryYear=2030`. The form itself is titled 'Form Validation' and contains the following fields:

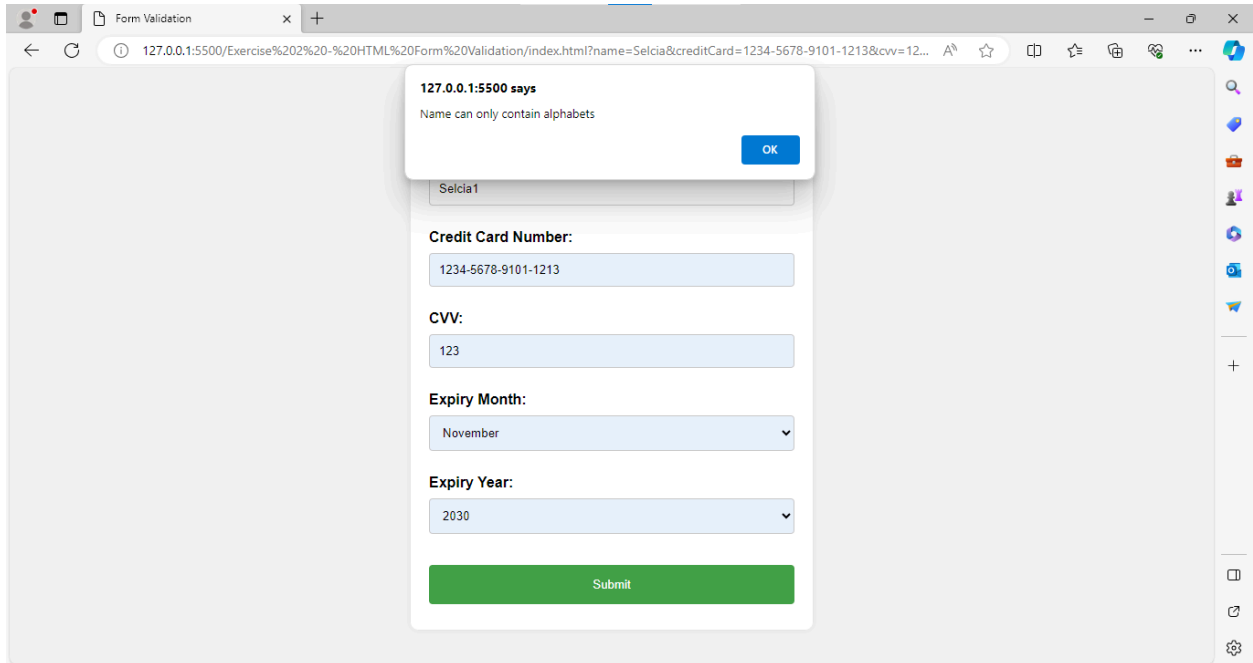
- Name:** A text input field containing 'Selcia'.
- Credit Card Number:** A text input field containing '1234-5678-1234-5678'.
- CVV:** A text input field containing '1234'.
- Expiry Month:** A dropdown menu with 'November' selected.
- Expiry Year:** A dropdown menu with '2030' selected.
- Submit:** A green button labeled 'Submit'.

### Test Case 2: Name has non-alphabetic characters, an alert is generated

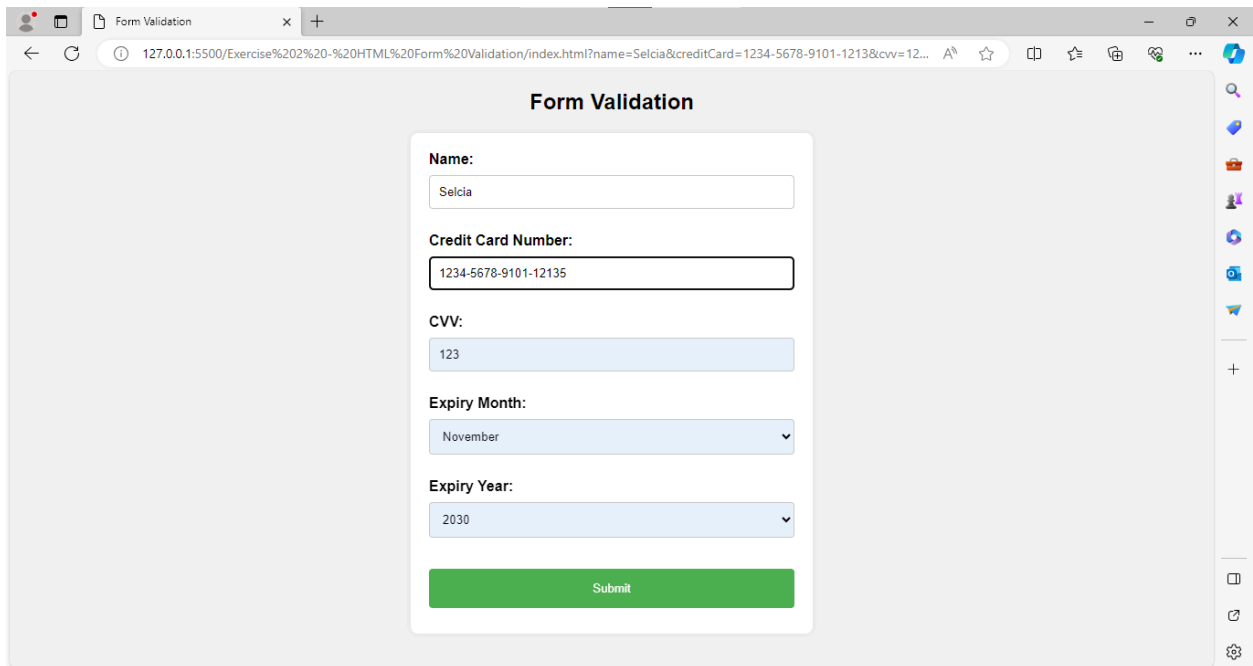


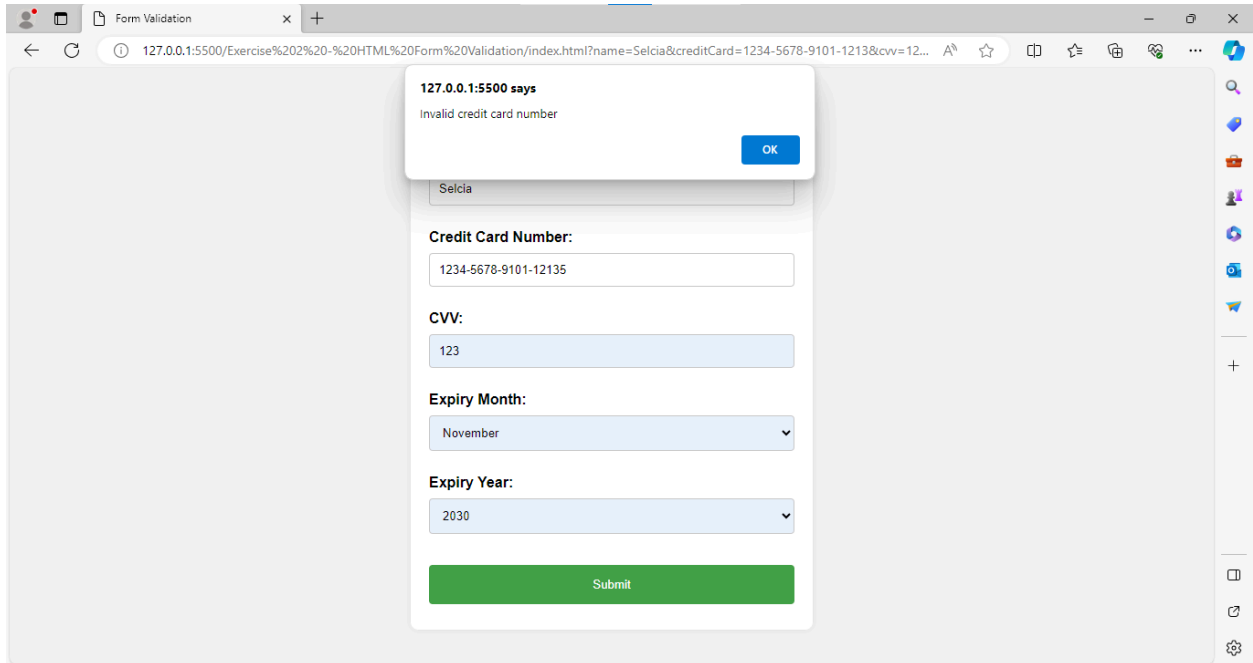
The screenshot shows the same web browser window as the previous one, but with the following changes:

- Name:** The text input field now contains 'Selcia1'.
- Credit Card Number:** The text input field contains '1234-5678-9101-1213'.
- CVV:** The text input field contains '123'.
- Expiry Month:** The dropdown menu still has 'November' selected.
- Expiry Year:** The dropdown menu still has '2030' selected.
- Submit:** The green button labeled 'Submit' remains at the bottom.

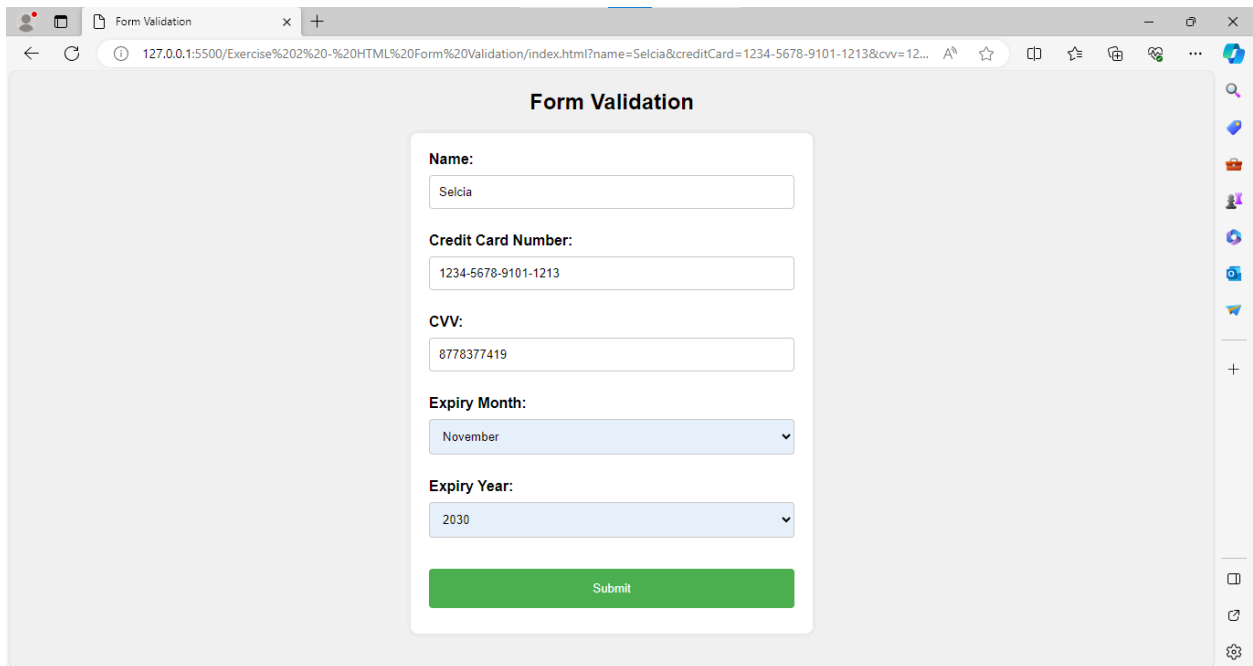


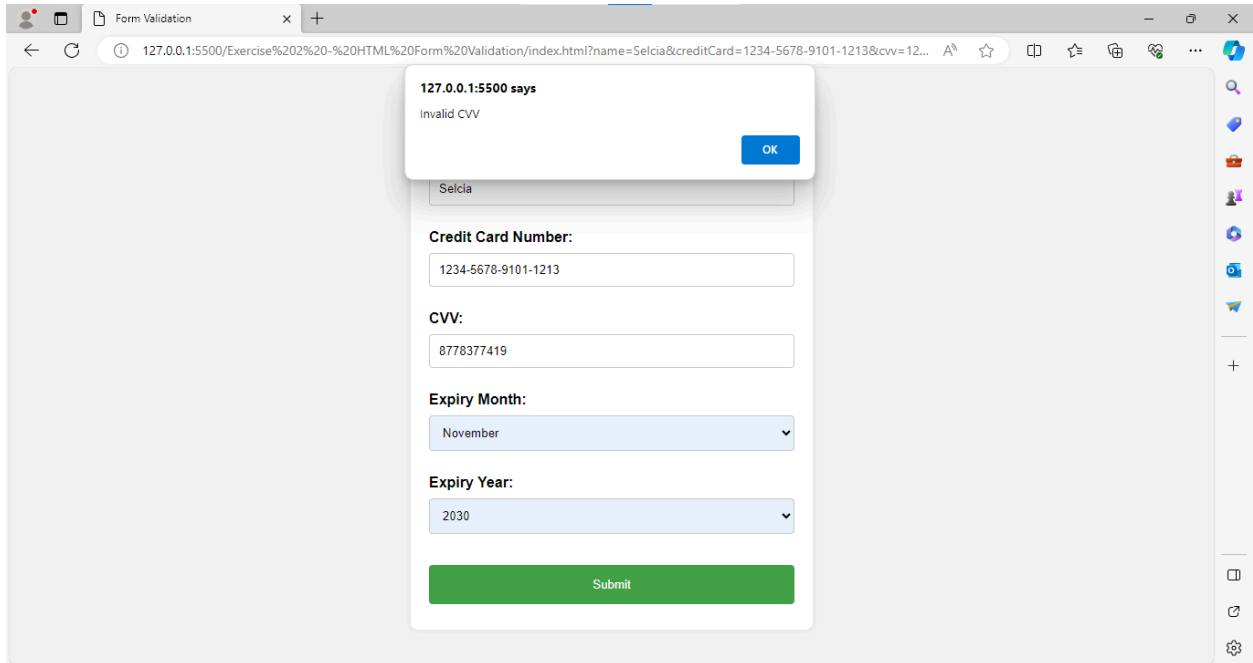
### Test Case 3: Credit Card Number Format is wrong, an alert is generated





#### Test Case 4: Invalid CVV Number, an alert is generated





## 6. Conclusion:

By implementing form validation using regular expressions and DOM manipulation, we learned how to enhance user experience and ensure data integrity.