

# **Web Stack Development**

## **Lab Exercise-6**

Submission Date:

**19<sup>th</sup> August 2024**

Submitted to:

**Dr. Cynthia T**

Submitted by:

**Eileen Maria Tom**

**(2447118)**

# form.html

```
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>Bentley User Registration</title>
7      <link href="https://cdnjs.cloudflare.com/ajax/libs/tailwindcss/2.2.19/tailwind.min.css" rel="stylesheet"
8      <link rel="stylesheet" href="form.css">
9  </head>
10 <body class="bg-gray-100 relative">
11
12     <nav class="bg-black p-4 text-white relative z-10">
13         <div class="max-w-6xl mx-auto flex justify-between items-center">
14             <ul class="flex space-x-4">
15                 <li><a href="#" class="hover:underline">Home</a></li>
16                 <li><a href="#" class="hover:underline">About</a></li>
17                 <li><a href="#" class="hover:underline">Contact</a></li>
18             </ul>
19         </div>
20     </nav>
21
22     <div class="max-w-lg mx-auto p-6 bg-white rounded-lg shadow-md mt-10">
23         <h2 class="text-2xl font-bold mb-6 text-center text-black">User Registration Form for booking</h2>
24
25         <div class="mb-6 text-center">
26             
27         </div>
28
29         <form id="registrationForm" novalidate>
30             <div class="mb-4">
31                 <label for="name" class="block text-gray-700">Enter your Name</label>
32                 <input type="text" id="name" name="name" class="w-full p-2 border rounded" required>
33                 <small id="nameError" class="text-red-600"></small>
34
35                 <input type="email" id="email" name="email" class="w-full p-2 border rounded" required>
36                 <small id="emailError" class="text-red-600"></small>
37             </div>
38             <div class="mb-4">
39                 <label for="password" class="block text-gray-700">Enter your Password</label>
40                 <input type="password" id="password" name="password" class="w-full p-2 border rounded" required>
41                 <small id="passwordError" class="text-red-600"></small>
42             </div>
43             <div class="mb-4">
44                 <label for="confirmPassword" class="block text-gray-700">Please confirm your Password</label>
45                 <input type="password" id="confirmPassword" name="confirmPassword" class="w-full p-2 border rounded" required>
46                 <small id="confirmPasswordError" class="text-red-600"></small>
47             </div>
48             <div class="mb-4">
49                 <label for="dob" class="block text-gray-700">Enter your Date of Birth</label>
50                 <input type="date" id="dob" name="dob" class="w-full p-2 border rounded" required>
51                 <small id="dobError" class="text-red-600"></small>
52             </div>
53             <div class="mb-4">
54                 <label for="age" class="block text-gray-700">Your Age</label>
55                 <input type="text" id="age" name="age" class="w-full p-2 border rounded" readonly>
56             </div>
57             <button type="submit" id="submitBtn" class="w-full bg-blue-600 text-white p-2 rounded mt-4">CLICK HERE</button>
58         </form>
59     </div>
60
61     <script src="form.js"></script>
62 </body>
63 </html>
```

Navigation Bar: A <nav> element is added to given a user-friendly approach to the website, it is styled using Tailwind CSS.

Form Container: The <div> with max-w-lg mx-auto p-6 bg-white rounded-lg shadow-md classes centers the form and gives it a white background with rounded corners and shadow.

Form Element: The <form> element has an id of registrationForm and novalidate to disable default browser validation and use custom JavaScript validation instead.

## form.css

```
1  input:valid {
2  |     border-color: green;
3  | }
4
5  input:invalid {
6  |     border-color: red;
7  | }
8
9  small {
10 |     display: none;
11 | }
12
13 input:invalid + small {
14 |     display: block;
15 | }
16
17 .error-visible {
18 |     display: block;
19 | }
20
```

Validation Styles: `input:valid` and `input:invalid` pseudo-classes change the border color of inputs based on their validation.

Error Message Display: `small` elements are hidden by default (`display: none`), and only displayed (`display: block`) if the input is invalid using the `input:invalid + small` selector. The `.error-visible` class is used to make error messages visible when there is an error.

## form.js

```
1  document.getElementById('registrationForm').addEventListener('submit', function(event) {
2  |     event.preventDefault();
3  |     validateForm();
4  | });
5
6  document.getElementById('name').addEventListener('input', validateName);
7  document.getElementById('email').addEventListener('input', validateEmail);
8  document.getElementById('password').addEventListener('input', validatePassword);
9  document.getElementById('confirmPassword').addEventListener('input', validateConfirmPassword);
10 document.getElementById('dob').addEventListener('input', validateDOB);
11
12 function validateForm() {
13 |     validateName();
14 |     validateEmail();
15 |     validatePassword();
16 |     validateConfirmPassword();
17 |     validateDOB();
18 |
19 |     if (isFormValid()) {
20 |         alert("Form submitted successfully!");
21 |         document.getElementById('registrationForm').submit();
22 |     }
23 }
```

```

25 function validateName() {
26     const nameInput = document.getElementById('name');
27     const nameError = document.getElementById('nameError');
28
29     const namePattern = /^[A-Za-z\s]{3,}$/;
30
31     if (!namePattern.test(nameInput.value)) {
32         nameError.textContent = "Name must be at least 3 characters long and contain only alphabets and spaces.";
33         nameInput.classList.add('border-red-500');
34         nameError.classList.add('error-visible');
35     } else {
36         nameError.classList.remove('error-visible');
37         nameInput.classList.remove('border-red-500');
38     }
39 }
40
41 function validateEmail() {
42     const emailInput = document.getElementById('email');
43     const emailError = document.getElementById('emailError');
44
45     const emailPattern = /^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,6}$/;
46
47     if (!emailPattern.test(emailInput.value)) {
48         emailError.textContent = "Please enter a valid email address.";
49         emailInput.classList.add('border-red-500');
50         emailError.classList.add('error-visible');
51     } else {
52         emailError.classList.remove('error-visible');
53         emailInput.classList.remove('border-red-500');
54     }
55 }

```

```

57 function validatePassword() {
58     const passwordInput = document.getElementById('password');
59     const passwordError = document.getElementById('passwordError');
60
61     const passwordPattern = /^(?=.*[A-Za-z])(?=.*\d)[A-Za-z\d]{8,}$/;
62
63     if (!passwordPattern.test(passwordInput.value)) {
64         passwordError.textContent = "Password must be at least 8 characters long and contain both letters and numbers.";
65         passwordInput.classList.add('border-red-500');
66         passwordError.classList.add('error-visible');
67     } else {
68         passwordError.classList.remove('error-visible');
69         passwordInput.classList.remove('border-red-500');
70     }
71 }
72
73 function validateConfirmPassword() {
74     const confirmPasswordInput = document.getElementById('confirmPassword');
75     const confirmPasswordError = document.getElementById('confirmPasswordError');
76     const passwordInput = document.getElementById('password');
77
78     if (confirmPasswordInput.value !== passwordInput.value) {
79         confirmPasswordError.textContent = "Passwords do not match.";
80         confirmPasswordInput.classList.add('border-red-500');
81         confirmPasswordError.classList.add('error-visible');
82     } else {
83         confirmPasswordError.classList.remove('error-visible');
84         confirmPasswordInput.classList.remove('border-red-500');
85     }
86 }

```

```

88  function validateDOB() {
89      const dobInput = document.getElementById('dob');
90      const dobError = document.getElementById('dobError');
91      const ageInput = document.getElementById('age');
92
93      const dob = new Date(dobInput.value);
94      const today = new Date();
95      const age = today.getFullYear() - dob.getFullYear();
96      const monthDifference = today.getMonth() - dob.getMonth();
97      const dayDifference = today.getDate() - dob.getDate();
98
99      if (monthDifference < 0 || (monthDifference === 0 && dayDifference < 0)) {
100          age--;
101      }
102
103      ageInput.value = age;
104
105      if (age < 18) {
106          dobError.textContent = "You must be at least 18 years old.";
107          dobInput.classList.add('border-red-500');
108          dobError.classList.add('error-visible');
109          document.getElementById('submitBtn').disabled = true;
110      } else {
111          dobError.classList.remove('error-visible');
112          dobInput.classList.remove('border-red-500');
113          document.getElementById('submitBtn').disabled = false;
114      }
115  }

```

```

116
117  function isFormValid() {
118      const nameInput = document.getElementById('name');
119      const emailInput = document.getElementById('email');
120      const passwordInput = document.getElementById('password');
121      const confirmPasswordInput = document.getElementById('confirmPassword');
122      const dobInput = document.getElementById('dob');
123
124      return (
125          nameInput.value.trim() !== '' &&
126          emailInput.value.trim() !== '' &&
127          passwordInput.value.trim() !== '' &&
128          confirmPasswordInput.value.trim() !== '' &&
129          dobInput.value.trim() !== '' &&
130          !document.querySelector('.border-red-500')
131      );
132  }

```

Form Submission Handling: Prevents default form submission with `event.preventDefault()` and calls `validateForm()` for custom validation.

`validateName()`: Ensures the name is at least 3 characters long and contains only letters and spaces and displays an error message if the input is invalid.

`validateForm()`: Calls all validation functions and successfully submits the form if all validations pass.

Email, Password, Confirm Password Validations: Ensure that format for the email id, password criteria is meet.

Date of Birth Validation: Calculates the age from the date of birth and checks if the user is at least 18 years of age, if not it disables the submit button.