

# Session 2

## HTML Forms

HTML Forms are a fundamental building block of interactive web development, designed to collect user input dynamically at runtime. In this session, you will gain a foundational understanding of how to create, structure, and implement HTML forms to gather information from users. The data entered through these forms is typically transmitted to a web server for processing, validation, and storage.

Forms are widely used to collect essential information such as usernames, passwords, contact details (e-mail IDs, mobile numbers), and other user preferences. This session introduces the core `<form>` container element and explores the various input controls it can hold, including:

- Text fields & text areas
- Radio buttons & checkboxes
- Submit & clickable buttons
- Labels for accessibility and clarity
- HTML5-specific input types (e.g., email, search, url, date, month, time)



### Suggestion

Before trying the solutions provided for the exercises, try to solve them on your own. This will help you understand the concepts better and improve your problem-solving skills.

Also, Lab manual includes great examples before exercises, make sure to go through them as well.

## Objective

✗ Do not copy. Read for understanding and the viva

After Completing this session, one should be able to:

- Understand the HTML form element
- use the form element in HTML
- use HTML to acquire user input and process it

- Create forms with basic elements like textboxes, checkboxes, radio buttons and submit buttons.
- Create forms using HTML5 elements like E-Mail address field and form validation

## Question 1

### Problem Statement

■ Write in lab record

Create an HTML webpage, as shown in the figure [2.1]. The user can specify any expression as input. Use form element to collect the user data.

### Figure 2.1 (Demo)

- Reference by question
- Working demo for preview

| Expression | Result |
|------------|--------|
| 1+1        | 2      |
| 1+1*2      | 3      |

### How the demo evaluates the expression

The lab manual solution uses `eval()`, and so does this demo. `eval` runs any string as JavaScript with access to everything in scope, which is why real applications avoid it. The `Function` constructor is the usual replacement: it still parses a string, but it runs in its own scope and cannot touch local variables.

eval.js

```
1 const expr = document.getElementById("expression-1").value; // e.g. "  
2 const result = eval(expr); // 3 - sees every local variable, runs any
```

function-constructor.js

```
1 const expr = document.getElementById("expression-1").value;  
2 const result = new Function(`return (${expr})`)(); // 3 - isolated sc
```

For a calculator that only needs numbers and `+ - * /`, the safest option is to validate the string first: `/^[\\d\\s+\\-*/()\\.]+$/ .test(expr)`. Write whichever the lab manual asks for in your record; mention the alternative in the viva.

## Code

■ Write in lab record

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>Vanilla Expression Calculator</title>
7     <style>
8       .demo-1 {
9         color: black;
10      }
11      /* Replicates: grid grid-cols-2 border p-1 w-1/3 mt-3 gap-1 */
12      .grid-container {
13        display: grid;
14        grid-template-columns: 1fr 1fr;
15        border: 1px solid #333;
16        color: black;
17        padding: 0.25rem;
18        width: 33.333%;
19        margin-top: 0.75rem;
20        gap: 0.25rem;
21        box-sizing: border-box;
22        font-family:
23          system-ui,
24          -apple-system,
25          sans-serif;
26      }
27
28      /* Replicates: border p-1 px-2 font-bold */
29      .grid-item,
30      .grid-input {
31        border: 1px solid #333;
32        padding: 0.25rem 0.5rem;
33        font-weight: bold;
34        box-sizing: border-box;
35        margin: 0;
36        color: black;
37      }
38
39      .grid-input {
40        outline: none;
41        background: #fff;
42      }
43
```

```
44         .grid-input:focus {
45             border-color: #2563eb;
46             box-shadow: 0 0 0 1px #2563eb;
47         }
48
49         .grid-header {
50             background-color: #f3f4f6;
51         }
52     </style>
53 </head>
54 <body class="demo-1">
55     <div class="grid-container">
56         <p class="grid-item grid-header">Expression</p>
57         <p class="grid-item grid-header">Result</p>
58
59         <input type="text" class="grid-input" id="expression-1" />
60         <input type="text" class="grid-input" id="result-1" readonly />
61
62         <input type="text" class="grid-input" id="expression-2" />
63         <input type="text" class="grid-input" id="result-2" readonly />
64     </div>
65
66     <script>
67         // 1. Mirror the $state object
68         const inputvalue = {
69             1: "1+1",
70             2: "1+1*2",
71             result1: "2",
72             result2: "3",
73         };
74
75         // 2. Cache DOM elements for performance
76         const els = {
77             expr1: document.getElementById("expression-1"),
78             res1: document.getElementById("result-1"),
79             expr2: document.getElementById("expression-2"),
80             res2: document.getElementById("result-2"),
81         };
82
83         // 3. Initialize inputs with default state
84         els.expr1.value = inputvalue[1];
85         els.res1.value = inputvalue.result1;
86         els.expr2.value = inputvalue[2];
87         els.res2.value = inputvalue.result2;
```

```

88
89 // 4. Core calculation function
90 function onInputChange(index) {
91     const exprEl = index === 1 ? els.expr1 : els.expr2;
92     const resEl = index === 1 ? els.res1 : els.res2;
93
94     // Sync DOM → State
95     inputValue[index] = exprEl.value;
96
97     try {
98         // Note: eval() is used as requested. In production, con
99         // like math.js or new Function() for better security &
100
101         const result = eval(inputValue[index]);
102
103         // Sync State → DOM
104
105         inputValue[`result${index}`] = result;
106
107         resEl.value = result;
108     } catch (error) {
109
110         inputValue[`result${index}`] = "Error";
111
112         resEl.value = "Error";
113     }
114 }
115
116 // 5. Attach event listeners
117
118 // Using 'input' fires on every keystroke (better UX for calcula
119
120 // Change to 'change' if you only want it to fire on blur/enter.
121
122 els.expr1.addEventListener("input", () => onInputChange(1));
123
124 els.expr2.addEventListener("input", () => onInputChange(2));

```

```
11
6      </script>
11
7      </body>
11
8 </html>
```

## Question 2

### Problem Statement

■ Write in lab record

Create a scrolling list that shows five items from the following list of PC processors: Intel i7, Motorola 68000, AMD Athlon , Intel 8088, AMD Ryzon, Intel Pentium MMX, Intel i3, Intel Pentium II, Intel Pentium III, Intel i5, Intel Celeron, PowerPC G3, PowerPC G4.

### Preview

✖ Do not copy. Read for understanding and the viva

1 Intel i7

2 Motorola 68000

3 AMD Athlon

4 Intel 8088

5 AMD Ryzen



6

Intel Pentium MMX

7

Intel i3

8

Intel Pentium II

9

Intel Pentium III

10

Intel i5

11

Intel Celeron

12

PowerPC G3

## Code

■ Write in lab record

list-scroller.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>PC Processors Scroll List</title>
7     <style>
8       * {
9         box-sizing: border-box;
10        margin: 0;
11        padding: 0;
12      }
13
14      body {
15        font-family:
16          system-ui,
17          -apple-system,
18          sans-serif;
19        background: #0f172a;
20        display: flex;
21        justify-content: center;
22        align-items: center;
23        height: 100svh;
24        padding: 12px;
25      }
26
27      .scroll-container {
28        width: 100%;
29        max-width: 420px;
30        height: 50svh;
31        background: #ffffff;
32        border-radius: 16px;
33        box-shadow: 0 12px 40px rgba(0, 0, 0, 0.35);
34        overflow-y: auto;
35        scroll-behavior: smooth;
36        position: relative;
37      }
38
39      .list-header {
40        position: sticky;
41        top: 0;
42        background: #1e293b;
43        color: #f8fafc;
```

```
44         padding: 18px;
45         text-align: center;
46         font-size: 1.15rem;
47         font-weight: 600;
48         z-index: 10;
49         box-shadow: 0 4px 12px rgba(0, 0, 0, 0.15);
50         letter-spacing: 0.5px;
51     }
52
53     .processor-list {
54         list-style: none;
55     }
56
57     .processor-item {
58         padding: 16px 20px;
59         border-bottom: 1px solid #e2e8f0;
60         display: flex;
61         align-items: center;
62         gap: 12px;
63         transition: background 0.2s ease;
64     }
65
66     .processor-item:last-child {
67         border-bottom: none;
68     }
69
70     .processor-item:hover {
71         background: #f1f5f9;
72     }
73
74     .item-number {
75         width: 32px;
76         height: 32px;
77         background: linear-gradient(135deg, #3b82f6, #8b5cf6);
78         border-radius: 8px;
79         display: flex;
80         align-items: center;
81         justify-content: center;
82         color: white;
83         font-size: 0.75rem;
84         font-weight: 700;
85         flex-shrink: 0;
86     }
87
```

```
88         .item-name {
89             font-size: 1rem;
90             color: #0f172a;
91             font-weight: 500;
92         }
93
94         /* Custom Scrollbar */
95         .scroll-container::-webkit-scrollbar {
96             width: 8px;
97         }
98         .scroll-container::-webkit-scrollbar-track {
99             background: #f8fafc;
100
101         }
102
103         .scroll-container::-webkit-scrollbar-thumb {
104
105             background: #cbd5e1;
106
107             border-radius: 4px;
108
109             border: 2px solid #f8fafc;
110
111         }
112
113         .scroll-container::-webkit-scrollbar-thumb:hover {
114
115             background: #94a3b8;
116
117         }
118
119     </style>
120
121 </head>
122
123 <body>
124
125     <div class="scroll-container">
126
127         <div class="list-header">PC Processors</div>
128
129         <ul class="processor-list" id="processorList"></ul>
130
131     </div>
```

```
11
6
11
7    <script>
11
8        const processors = [
11
9            "Intel i7",
12
10            "Motorola 68000",
12
11            "AMD Athlon",
12
12            "Intel 8088",
12
13            "AMD Ryzen",
12
14            "Intel Pentium MMX",
12
15            "Intel i3",
12
16            "Intel Pentium II",
12
17            "Intel Pentium III",
12
18            "Intel i5",
12
19            "Intel Celeron",
13
20            "PowerPC G3",
13
21            "PowerPC G4",
13
22        ];
13
23
24
25        const listEl = document.getElementById("processorList");
13
26
27
28        // Generate list items dynamically
13
29        processors.forEach((proc, index) => {
```

```
13
8      const li = document.createElement("li");
13
9      li.className = "processor-item";
14
10     li.innerHTML = `
14
11         <div class="item-number">${index + 1}</div>
14
12         <div class="item-name">${proc}</div>
14
13     `;
14
14     listEl.appendChild(li);
14
15     });
14
16 </script>
14
17 </body>
14
18 </html>
```

## Question 3

### Problem Statement

■ Write in lab record

Design a webpage for a software company to accept on-line job applications. The webpage must be designed to use form with elements:

- Position applied for (autofocus), name, nationality, date of birth (selected from an auto picker), address (in a text area), telephone number and email (required).
- Educational history and qualifications.
- Work experience/employment/training in terms of employer history and number of years of experience selected from a slider. Set maximum years of experience to 10 years.
- Personal statement.
- Two referees including names, occupation, relationship, address, telephone.

## Preview

✕ Do not copy. Read for understanding and the viva

### Ethercorps Online Job Application Portal

#### Personal & Contact Information

Position Applied For \*

e.g. Senior Frontend Developer

Full Name \*

Nationality \*

Date of Birth \*

dd/mm/yyyy



Email Address \*

you@company.com

Address

Telephone Number \*

+1 (555) 000-0000

## Education & Qualifications

### Educational History & Qualifications

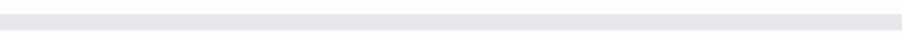
Degrees, certifications, institutions, graduation dates...

## Work Experience & Training

### Employer History & Training

Previous companies, roles, key responsibilities, training programs...

### Years of Experience



0  
yrs

## Personal Statement

### Why are you a strong candidate? \*

Highlight your skills, motivation, and career objectives...

## Referees

### Referee 1

Full Name \*

Occupation \*

Relationship \*

Address \*

Telephone \*

Referee 2

---

Full Name \*

Occupation \*

Relationship \*

Address \*

Telephone \*

Submit Application

## Code

■ Write in lab record

job-portal.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>Apply Now - TechNova Solutions</title>
7     <style>
8       :root {
9         --primary: #2563eb;
10        --primary-hover: #1d4ed8;
11        --bg: #f8fafc;
12        --surface: #ffffff;
13        --text: #0f172a;
14        --text-muted: #475569;
15        --border: #cbd5e1;
16        --focus-ring: rgba(37, 99, 235, 0.3);
17        --error: #ef4444;
18        --success: #10b981;
19        --radius: 10px;
20        --shadow:
21          0 4px 6px -1px rgb(0 0 0 / 0.1),
22          0 2px 4px -2px rgb(0 0 0 / 0.1);
23      }
24
25      * {
26        box-sizing: border-box;
27        margin: 0;
28        padding: 0;
29      }
30
31      body {
32        font-family:
33          system-ui,
34          -apple-system,
35          "Segoe UI",
36          Roboto,
37          sans-serif;
38        background: var(--bg);
39        color: var(--text);
40        line-height: 1.5;
41        padding: 2rem 1rem;
42        min-height: 100vh;
43      }
```

```
44
45     .container {
46         max-width: 900px;
47         margin: 0 auto;
48         background: var(--surface);
49         padding: 2.5rem;
50         border-radius: var(--radius);
51         box-shadow: var(--shadow);
52     }
53
54     header {
55         text-align: center;
56         margin-bottom: 2rem;
57         padding-bottom: 1.5rem;
58         border-bottom: 1px solid var(--border);
59     }
60
61     header h1 {
62         font-size: 1.75rem;
63         font-weight: 700;
64         color: var(--text);
65     }
66
67     header p {
68         color: var(--text-muted);
69         margin-top: 0.25rem;
70     }
71
72     form {
73         display: flex;
74         flex-direction: column;
75         gap: 1.75rem;
76     }
77
78     fieldset {
79         border: 1px solid var(--border);
80         border-radius: var(--radius);
81         padding: 1.5rem;
82         background: #fdfdfd;
83     }
84
85     legend {
86         font-weight: 600;
87         padding: 0 0.5rem;
```

```
88         color: var(--primary);
89         font-size: 1.1rem;
90     }
91
92     .form-grid {
93         display: grid;
94         grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
95         gap: 1.25rem;
96     }
97
98     .form-group {
99         display: flex;
100
101         flex-direction: column;
102
103         gap: 0.4rem;
104     }
105
106     .form-group.full-width {
107
108         grid-column: 1 / -1;
109     }
110
111     label {
112
113         font-size: 0.9rem;
114
115         font-weight: 500;
116
117         color: var(--text);
118     }
119
120     label .req {
121
122         color: var(--error);
```

```
11
6      margin-left: 2px;
11
7    }
11
8
11
9    input,
12
10   textarea,
12
11   select {
12
12     padding: 0.75rem;
12
13     border: 1px solid var(--border);
12
14     border-radius: 8px;
12
15     font-size: 0.95rem;
12
16     font-family: inherit;
12
17     transition:
12
18         border-color 0.2s,
12
19         box-shadow 0.2s;
13
20     background: #fff;
13
21   }
13
22
13
23   input:focus,
13
24   textarea:focus {
13
25     outline: none;
13
26     border-color: var(--primary);
13
27     box-shadow: 0 0 0 3px var(--focus-ring);
```

```
13
8      }
13
9
14
0      input:invalid:not(:placeholder-shown) {
14
1      border-color: var(--error);
14
2      }
14
3
14
4      textarea {
14
5          resize: vertical;
14
6          min-height: 100px;
14
7      }
14
8
14
9      /* Slider Styling */
15
0      .slider-wrapper {
15
1          display: flex;
15
2          align-items: center;
15
3          gap: 1rem;
15
4          margin-top: 0.5rem;
15
5      }
15
6
15
7      input[type="range"] {
15
8          flex: 1;
15
9          cursor: pointer;
```

```
16
0      accent-color: var(--primary);
16
1      }
16
2
16
3      .slider-value {
16
4          min-width: 3ch;
16
5          text-align: center;
16
6          font-weight: 600;
16
7          background: var(--bg);
16
8          padding: 0.3rem 0.6rem;
16
9          border-radius: 6px;
17
0          border: 1px solid var(--border);
17
1      }
17
2
17
3      /* Referees Grid */
17
4      .referees-grid {
17
5          display: grid;
17
6          grid-template-columns: 1fr 1fr;
17
7          gap: 1.5rem;
17
8      }
17
9
18
0      .referee-card {
18
1          background: #fff;
```

```
18
2      border: 1px solid var(--border);
18
3      border-radius: 8px;
18
4      padding: 1.25rem;
18
5  }
18
6
18
7  .referee-card h3 {
18
8      margin-bottom: 1rem;
18
9      font-size: 1rem;
19
0      color: var(--text-muted);
19
1      border-bottom: 1px dashed var(--border);
19
2      padding-bottom: 0.5rem;
19
3  }
19
4
19
5  /* Submit Button */
19
6  button[type="submit"] {
19
7      background: var(--primary);
19
8      color: white;
19
9      border: none;
20
0      padding: 1rem 2rem;
20
1      font-size: 1rem;
20
2      font-weight: 600;
20
3      border-radius: 8px;
```

```
20
  4      cursor: pointer;
20
  5      transition:
20
  6          background 0.2s,
20
  7          transform 0.1s;
20
  8      margin-top: 1rem;
20
  9      align-self: flex-start;
21
  0  }
21
  1
21
  2  button[type="submit"]:hover {
21
  3      background: var(--primary-hover);
21
  4  }
21
  5  button[type="submit"]:active {
21
  6      transform: scale(0.98);
21
  7  }
21
  8  button[type="submit"]:disabled {
21
  9      background: var(--text-muted);
22
  0      cursor: not-allowed;
22
  1      opacity: 0.7;
22
  2  }
22
  3
22
  4  /* Message Toast */
22
  5  .message {
```

```
22
6      padding: 1rem;
22
7      border-radius: 8px;
22
8      margin-top: 1rem;
22
9      display: none;
23
10     font-weight: 500;
23
11     text-align: center;
23
12 }
23
13
23
14
23
15 .message.success {
23
16     background: #dcfce7;
23
17     color: #166534;
23
18     border: 1px solid #bbf7d0;
23
19     display: block;
23
20 }
24
21
24
22 .message.error {
24
23     background: #fee2e2;
24
24     color: #991b1b;
24
25     border: 1px solid #fecaca;
24
26     display: block;
24
27 }
```

```
24
8      @media (max-width: 768px) {
24
9          .container {
25
10              padding: 1.5rem;
25
11          }
25
12          .referees-grid {
25
13              grid-template-columns: 1fr;
25
14          }
25
15          button[type="submit"] {
25
16              width: 100%;
25
17              align-self: center;
25
18          }
25
19      }
26
20  </style>
26
21  </head>
26
22  <body>
26
23      <div class="container">
26
24          <header>
26
25              <h1>Ethercorps Solutions</h1>
26
26              <p>Online Job Application Portal</p>
26
27          </header>
26
28
26
29      <form id="applicationForm" novalidate>
```

```
27
0      <!-- Personal & Contact -->
27
1      <fieldset>
27
2          <legend>Personal & Contact Information</legend>
27
3          <div class="form-grid">
27
4              <div class="form-group full-width">
27
5                  <label for="position"
6                      >Position Applied For
27
7                      <span class="req">*</span></label
27
8                      >
27
9                      <input
28
10                         type="text"
28
11                         id="position"
28
12                         name="position"
28
13                         placeholder="e.g. Senior Frontend Developer"
28
14                         required
28
15                         autofocus
28
16                     />
27
17                 </div>
28
18                 <div class="form-group">
28
19                     <label for="fullName"
29
20                         >Full Name <span class="req">*</span></label
29
21                     >
```

```
29
 2      <input
29
 3          type="text"
29
 4          id="fullName"
29
 5          name="fullName"
29
 6          required
29
 7      />
29
 8  </div>
29
 9  <div class="form-group">
30
10      <label for="nationality"
30
11          >Nationality <span class="req">*</span></lab
30
12      >
30
13      <input
30
14          type="text"
30
15          id="nationality"
30
16          name="nationality"
30
17          required
30
18      />
30
19  </div>
31
20  <div class="form-group">
31
21      <label for="dob"
31
22          >Date of Birth <span class="req">*</span></l
31
23      >
```

```
31
 4      <input type="date" id="dob" name="dob" required
31
 5  </div>
31
 6  <div class="form-group">
31
 7      <label for="email"
31
 8          >Email Address <span class="req">*</span></l
31
 9  >
32
10  <input
32
11      type="email"
32
12      id="email"
32
13      name="email"
32
14      placeholder="you@example.com"
32
15      required
32
16  />
32
17 </div>
32
18 <div class="form-group full-width">
32
19     <label for="address">Address</label>
33
20     <textarea
33
21         id="address"
33
22         name="address"
33
23         rows="2"
33
24         required
33
25     ></textarea>
```

```
33
6      </div>
33
7      <div class="form-group">
33
8          <label for="phone"
33
9              >Telephone Number
34
0          <span class="req">*</span></label
34
1      >
34
2      <input
34
3          type="tel"
34
4          id="phone"
34
5          name="phone"
34
6          placeholder="+1 (555) 000-0000"
34
7          required
34
8      />
34
9      </div>
35
0      </div>
35
1  </fieldset>
35
2
35
3  <!-- Education -->
35
4  <fieldset>
35
5      <legend>Education & Qualifications</legend>
35
6      <div class="form-group">
35
7          <label for="education"
```

```
35         >Educational History & Qualifications</label>
8
35     >
9
36     <textarea
0
36         id="education"
1
36         name="education"
2
36         placeholder="Degrees, certifications, institutio
3
36     ></textarea>
4
36 </div>
5
36 </fieldset>
6
36
7
36
8     <!-- Work Experience -->
36
9     <fieldset>
37
0         <legend>Work Experience & Training</legend>
37
1         <div class="form-group">
37
2             <label for="employmentHistory"
37
3                 >Employer History & Training</label>
37
4             >
37
5             <textarea
37
6                 id="employmentHistory"
37
7                 name="employmentHistory"
37
8                 rows="4"
37
9                 placeholder="Previous companies, roles, key resp
```

```
38
0      ></textarea>
38
1      </div>
38
2      <div class="form-group">
38
3          <label for="yearsExp"
38
4              >Years of Experience (Max 10)</label
38
5              >
38
6          <div class="slider-wrapper">
38
7              <input
38
8                  type="range"
38
9                  id="yearsExp"
39
0                  name="yearsExp"
39
1                  min="0"
39
2                  max="10"
39
3                  step="1"
39
4                  value="0"
39
5              />
39
6              <output id="yearsExpOutput" class="slider-value"
39
7                  >0</output
39
8                  >
39
9              </div>
40
0          </div>
40
1      </fieldset>
```

```
40
2
40
3      <!-- Personal Statement -->
40
4      <fieldset>
40
5          <legend>Personal Statement</legend>
40
6          <div class="form-group">
40
7              <label for="personalStatement"
8                  >Why are you a strong candidate?
40
9                  <span class="req">*</span></label>
41
10             >
41
11             <textarea
41
12                 id="personalStatement"
41
13                 name="personalStatement"
41
14                 rows="4"
41
15                 placeholder="Highlight your skills, motivation,
41
16                 required
41
17             ></textarea>
41
18         </div>
41
19     </fieldset>
42
20
42
1      <!-- Referees -->
42
2      <fieldset>
42
3          <legend>Referees</legend>
```

```
42
  4      <div class="referees-grid">
42
  5          <div class="referee-card">
42
  6              <h3>Referee 1</h3>
42
  7              <div class="form-group">
42
  8                  <label for="ref1Name">Full Name</label>
42
  9                  <input
43
  0                      type="text"
43
  1                      id="ref1Name"
43
  2                      name="ref1Name"
43
  3                      required
43
  4                  />
43
  5              </div>
43
  6              <div class="form-group">
43
  7                  <label for="ref10ccupation">Occupation</label>
43
  8                  <input
43
  9                      type="text"
44
  0                      id="ref10ccupation"
44
  1                      name="ref10ccupation"
44
  2                      required
44
  3                  />
44
  4              </div>
44
  5              <div class="form-group">
```

```
44
6      <label for="ref1Relationship"
44
7      >Relationship</label
44
8      >
44
9      <input
45
10      type="text"
45
11      id="ref1Relationship"
45
12      name="ref1Relationship"
45
13      placeholder="e.g. Former Manager"
45
14      required
45
15      />
45
16 </div>
45
17 <div class="form-group">
45
18     <label for="ref1Address">Address</label>
45
19     <textarea
46
20         id="ref1Address"
46
21         name="ref1Address"
46
22         rows="2"
46
23         required
46
24     ></textarea>
46
25 </div>
46
26 <div class="form-group">
46
27     <label for="ref1Phone">Telephone</label>
```

```
46
8      <input
46
9          type="tel"
47
10         id="ref1Phone"
47
11         name="ref1Phone"
47
12         required
47
13     />
47
14 </div>
47
15 </div>
47
16
47
17 <div class="referee-card">
47
18     <h3>Referee 2</h3>
47
19     <div class="form-group">
48
20         <label for="ref2Name">Full Name</label>
48
21         <input
48
22             type="text"
48
23             id="ref2Name"
48
24             name="ref2Name"
48
25             required
48
26         />
48
27     </div>
48
28     <div class="form-group">
48
29         <label for="ref2Occupation">Occupation</label>
```

```
49
0      <input
49
1      type="text"
49
2      id="ref20ccupation"
49
3      name="ref20ccupation"
49
4      required
49
5      />
49
6  </div>
49
7  <div class="form-group">
49
8      <label for="ref2Relationship"
49
9      >Relationship</label
50
0      >
50
1      <input
50
2      type="text"
50
3      id="ref2Relationship"
50
4      name="ref2Relationship"
50
5      required
50
6      />
50
7  </div>
50
8  <div class="form-group">
50
9      <label for="ref2Address">Address</label>
51
0      <textarea
51
1      id="ref2Address"
```

```
51
 2          name="ref2Address"
51
 3          rows="2"
51
 4          required
51
 5      ></textarea>
51
 6  </div>
51
 7  <div class="form-group">
51
 8      <label for="ref2Phone">Telephone</label>
51
 9      <input
52
10          type="tel"
52
11          id="ref2Phone"
52
12          name="ref2Phone"
52
13          required
52
14      />
52
15  </div>
52
16  </div>
52
17  </div>
52
18  </fieldset>
52
19
53
20  <button type="submit" id="submitBtn">Submit Application</but
53
21  <div id="formMessage" class="message" aria-live="polite"></d
53
22  </form>
53
23  </div>
```

```
53
4
53
5    <script>
53
6        document.addEventListener("DOMContentLoaded", () => {
53
7            const form = document.getElementById("applicationForm");
53
8            const slider = document.getElementById("yearsExp");
53
9            const sliderOutput = document.getElementById("yearsExpOutput
54
0            const submitBtn = document.getElementById("submitBtn");
54
1            const messageDiv = document.getElementById("formMessage");
54
2
54
3            // Update slider output dynamically
54
4            slider.addEventListener("input", () => {
54
5                sliderOutput.textContent = slider.value;
54
6            });
54
7
54
8            // Handle form submission
54
9            form.addEventListener("submit", (e) => {
55
0                e.preventDefault();
55
1                messageDiv.className = "message"; // Reset classes
55
2                messageDiv.style.display = "none";
55
3
55
4                // Trigger native HTML5 validation
55
5                if (!form.checkValidity()) {
```

```
55
6      form.reportValidity();
55
7      return;
55
8  }
55
9
56
0      // Simulate async submission
56
1      submitBtn.disabled = true;
56
2      submitBtn.textContent = "Submitting...";
56
3
56
4      // Collect form data (for demonstration)
56
5      const formData = new FormData(form);
56
6      const data = Object.fromEntries(formData.entries());
56
7      console.log("Application Data:", data);
56
8
56
9      // Mock API delay
57
0      setTimeout(() => {
57
1          messageDiv.textContent =
57
2              "✅ Application submitted successfully! We will
57
3              messageDiv.classList.add("success");
57
4              messageDiv.style.display = "block";
57
5
57
6      // Reset form
57
7      form.reset();
```

```
57
8         sliderOutput.textContent = "0";
57
9         submitBtn.disabled = false;
58
10        submitBtn.textContent = "Submit Application";
58
11
58
12        // Scroll to message
58
13        messageDiv.scrollIntoView({
58
14            behavior: "smooth",
58
15            block: "nearest",
58
16        });
58
17        }, 1500);
58
18    });
58
19    });
59
20    </script>
59
21    </body>
59
22    </html>
```

[✎ Edit this page](#)[< Previous](#)  
**Session 1****Session 3** [Next >](#)