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

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

function-constructor.js

```
const expr = document.getElementById("expression-1").value;
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

```html
<!doctype html>
<html lang="en">
    <head>
        <meta charset="UTF-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1.0
        <title>Vanilla Expression Calculator</title>
        <style>
            .demo-1 {
                color: black;
            }
            /* Replicates: grid grid-cols-2 border p-1 w-1/3 mt-3 gap-1 */
            .grid-container {
                display: grid;
                grid-template-columns: 1fr 1fr;
                border: 1px solid #333;
                color: black;
                padding: 0.25rem;
                width: 33.333%;
                margin-top: 0.75rem;
                gap: 0.25rem;
                box-sizing: border-box;
                font-family:
                    system-ui,
                    -apple-system,
                    sans-serif;
            }

            /* Replicates: border p-1 px-2 font-bold */
            .grid-item,
            .grid-input {
                border: 1px solid #333;
                padding: 0.25rem 0.5rem;
                font-weight: bold;
                box-sizing: border-box;
                margin: 0;
                color: black;
            }

            .grid-input {
                outline: none;
                background: #fff;
            }

```

```
            .grid-input:focus {
                border-color: #2563eb;
                box-shadow: 0 0 0 1px #2563eb;
            }

            .grid-header {
                background-color: #f3f4f6;
            }
        </style>
    </head>
    <body class="demo-1">
        <div class="grid-container">
            <p class="grid-item grid-header">Expression</p>
            <p class="grid-item grid-header">Result</p>

            <input type="text" class="grid-input" id="expression-1" />
            <input type="text" class="grid-input" id="result-1" readonly />

            <input type="text" class="grid-input" id="expression-2" />
            <input type="text" class="grid-input" id="result-2" readonly />
        </div>

        <script>
            // 1. Mirror the $state object
            const inputvalue = {
                1: "1+1",
                2: "1+1*2",
                result1: "2",
                result2: "3",
            };

            // 2. Cache DOM elements for performance
            const els = {
                expr1: document.getElementById("expression-1"),
                res1: document.getElementById("result-1"),
                expr2: document.getElementById("expression-2"),
                res2: document.getElementById("result-2"),
            };

            // 3. Initialize inputs with default state
            els.expr1.value = inputvalue[1];
            els.res1.value = inputvalue.result1;
            els.expr2.value = inputvalue[2];
            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 &
10
 0                      const result = eval(inputvalue[index]);
10
 1
10
 2                      // Sync State → DOM
10
 3                      inputvalue[`result${index}`] = result;
10
 4                      resEl.value = result;
10
 5                  } catch (error) {
10
 6                      inputvalue[`result${index}`] = "Error";
10
 7                      resEl.value = "Error";
10
 8                  }
10
 9              }
11
 0
11
 1              // 5. Attach event listeners
11
 2              // Using 'input' fires on every keystroke (better UX for calcula
11
 3              // Change to 'change' if you only want it to fire on blur/enter.
11
 4              els.expr1.addEventListener("input", () => onInputChange(1));
11
 5              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

```html
<!doctype html>
<html lang="en">
    <head>
        <meta charset="UTF-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1.0
        <title>PC Processors Scroll List</title>
        <style>
            * {
                box-sizing: border-box;
                margin: 0;
                padding: 0;
            }

            body {
                font-family:
                    system-ui,
                    -apple-system,
                    sans-serif;
                background: #0f172a;
                display: flex;
                justify-content: center;
                align-items: center;
                height: 100svh;
                padding: 12px;
            }

            .scroll-container {
                width: 100%;
                max-width: 420px;
                height: 50svh;
                background: #ffffff;
                border-radius: 16px;
                box-shadow: 0 12px 40px rgba(0, 0, 0, 0.35);
                overflow-y: auto;
                scroll-behavior: smooth;
                position: relative;
            }

            .list-header {
                position: sticky;
                top: 0;
                background: #1e293b;
                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
```

```
            .item-name {
                font-size: 1rem;
                color: #0f172a;
                font-weight: 500;
            }

            /* Custom Scrollbar */
            .scroll-container::-webkit-scrollbar {
                width: 8px;
            }
            .scroll-container::-webkit-scrollbar-track {
                background: #f8fafc;

            }

            .scroll-container::-webkit-scrollbar-thumb {

                background: #cbd5e1;

                border-radius: 4px;

                border: 2px solid #f8fafc;

            }

            .scroll-container::-webkit-scrollbar-thumb:hover {

                background: #94a3b8;

            }

        </style>

    </head>

    <body>

        <div class="scroll-container">

            <div class="list-header">PC Processors</div>

            <ul class="processor-list" id="processorList"></ul>

        </div>
```

```
11
 6
11
 7          <script>
11
 8              const processors = [
11
 9                  "Intel i7",
12
 0                  "Motorola 68000",
12
 1                  "AMD Athlon",
12
 2                  "Intel 8088",
12
 3                  "AMD Ryzen",
12
 4                  "Intel Pentium MMX",
12
 5                  "Intel i3",
12
 6                  "Intel Pentium II",
12
 7                  "Intel Pentium III",
12
 8                  "Intel i5",
12
 9                  "Intel Celeron",
13
 0                  "PowerPC G3",
13
 1                  "PowerPC G4",
13
 2              ];
13
 3
13
 4              const listEl = document.getElementById("processorList");
13
 5
13
 6              // Generate list items dynamically
13
 7              processors.forEach((proc, index) => {
```

```
                const li = document.createElement("li");
                li.className = "processor-item";
                li.innerHTML = `
                <div class="item-number">${index + 1}</div>
                <div class="item-name">${proc}</div>
            `;
                listEl.appendChild(li);
            });
        </script>
    </body>
</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 *

e.g. Former Manager

Address *

Telephone *

Referee 2

Full Name *

Occupation *

Relationship *

Address *

Telephone *

Submit Application

Code

Write in lab record

job-portal.html

```html
<!doctype html>
<html lang="en">
    <head>
        <meta charset="UTF-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1.0
        <title>Apply Now - TechNova Solutions</title>
        <style>
            :root {
                --primary: #2563eb;
                --primary-hover: #1d4ed8;
                --bg: #f8fafc;
                --surface: #ffffff;
                --text: #0f172a;
                --text-muted: #475569;
                --border: #cbd5e1;
                --focus-ring: rgba(37, 99, 235, 0.3);
                --error: #ef4444;
                --success: #10b981;
                --radius: 10px;
                --shadow:
                    0 4px 6px -1px rgb(0 0 0 / 0.1),
                    0 2px 4px -2px rgb(0 0 0 / 0.1);
            }

            * {
                box-sizing: border-box;
                margin: 0;
                padding: 0;
            }

            body {
                font-family:
                    system-ui,
                    -apple-system,
                    "Segoe UI",
                    Roboto,
                    sans-serif;
                background: var(--bg);
                color: var(--text);
                line-height: 1.5;
                padding: 2rem 1rem;
                min-height: 100vh;
            }
```

```
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;
10
 0                  flex-direction: column;
10
 1                  gap: 0.4rem;
10
 2              }
10
 3
10
 4              .form-group.full-width {
10
 5                  grid-column: 1 / -1;
10
 6              }
10
 7
10
 8              label {
10
 9                  font-size: 0.9rem;
11
 0                  font-weight: 500;
11
 1                  color: var(--text);
11
 2              }
11
 3
11
 4              label .req {
11
 5                  color: var(--error);
```

```
11
 6                margin-left: 2px;
11
 7            }
11
 8
11
 9            input,
12
 0            textarea,
12
 1            select {
12
 2                padding: 0.75rem;
12
 3                border: 1px solid var(--border);
12
 4                border-radius: 8px;
12
 5                font-size: 0.95rem;
12
 6                font-family: inherit;
12
 7                transition:
12
 8                    border-color 0.2s,
12
 9                    box-shadow 0.2s;
13
 0                background: #fff;
13
 1            }
13
 2
13
 3            input:focus,
13
 4            textarea:focus {
13
 5                outline: none;
13
 6                border-color: var(--primary);
13
 7                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;
```

```
                accent-color: var(--primary);
            }


            .slider-value {
                min-width: 3ch;
                text-align: center;
                font-weight: 600;
                background: var(--bg);
                padding: 0.3rem 0.6rem;
                border-radius: 6px;
                border: 1px solid var(--border);
            }


            /* Referees Grid */
            .referees-grid {
                display: grid;
                grid-template-columns: 1fr 1fr;
                gap: 1.5rem;
            }


            .referee-card {
                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
 0                font-weight: 500;
23
 1                text-align: center;
23
 2            }
23
 3
23
 4            .message.success {
23
 5                background: #dcfce7;
23
 6                color: #166534;
23
 7                border: 1px solid #bbf7d0;
23
 8                display: block;
23
 9            }
24
 0
24
 1            .message.error {
24
 2                background: #fee2e2;
24
 3                color: #991b1b;
24
 4                border: 1px solid #fecaca;
24
 5                display: block;
24
 6            }
24
 7
```

```
24
 8            @media (max-width: 768px) {
24
 9                .container {
25
 0                    padding: 1.5rem;
25
 1                }
25
 2                .referees-grid {
25
 3                    grid-template-columns: 1fr;
25
 4                }
25
 5                button[type="submit"] {
25
 6                    width: 100%;
25
 7                    align-self: center;
25
 8                }
25
 9            }
26
 0        </style>
26
 1    </head>
26
 2    <body>
26
 3        <div class="container">
26
 4            <header>
26
 5                <h1>Ethercorps Solutions</h1>
26
 6                <p>Online Job Application Portal</p>
26
 7            </header>
26
 8
26
 9            <form id="applicationForm" novalidate>
```

```
                <!— Personal & Contact —>
                <fieldset>
                    <legend>Personal & Contact Information</legend>
                    <div class="form-grid">
                        <div class="form-group full-width">
                            <label for="position"
                                >Position Applied For
                                <span class="req">*</span></label
                            >
                            <input
                                type="text"
                                id="position"
                                name="position"
                                placeholder="e.g. Senior Frontend Developer"
                                required
                                autofocus
                            />
                        </div>
                        <div class="form-group">
                            <label for="fullName"
                                >Full Name <span class="req">*</span></label
                            >
```

```
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
 0                                <label for="nationality"
30
 1                                    >Nationality <span class="req">*</span></lab
30
 2                                >
30
 3                                <input
30
 4                                    type="text"
30
 5                                    id="nationality"
30
 6                                    name="nationality"
30
 7                                    required
30
 8                                />
30
 9                            </div>
31
 0                            <div class="form-group">
31
 1                                <label for="dob"
31
 2                                    >Date of Birth <span class="req">*</span></l
31
 3                                >
```

```
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
 0                                <input
32
 1                                    type="email"
32
 2                                    id="email"
32
 3                                    name="email"
32
 4                                    placeholder="you@example.com"
32
 5                                    required
32
 6                                />
32
 7                            </div>
32
 8                            <div class="form-group full-width">
32
 9                                <label for="address">Address</label>
33
 0                                <textarea
33
 1                                    id="address"
33
 2                                    name="address"
33
 3                                    rows="2"
33
 4                                    required
33
 5                                ></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
 8                           >Educational History & Qualifications</label
35
 9                       >
36
 0                       <textarea
36
 1                           id="education"
36
 2                           name="education"
36
 3                           placeholder="Degrees, certifications, institutio
36
 4                       ></textarea>
36
 5                   </div>
36
 6               </fieldset>
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
```

```
                        ></textarea>
                    </div>
                    <div class="form-group">
                        <label for="yearsExp"
                            >Years of Experience (Max 10)</label
                        >
                        <div class="slider-wrapper">
                            <input
                                type="range"
                                id="yearsExp"
                                name="yearsExp"
                                min="0"
                                max="10"
                                step="1"
                                value="0"
                            />
                            <output id="yearsExpOutput" class="slider-value"
                                >0</output
                            >
                        </div>
                    </div>
                </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"
40
 8                            >Why are you a strong candidate?
40
 9                            <span class="req">*</span></label
41
 0                        >
41
 1                        <textarea
41
 2                            id="personalStatement"
41
 3                            name="personalStatement"
41
 4                            rows="4"
41
 5                            placeholder="Highlight your skills, motivation,
41
 6                            required
41
 7                        ></textarea>
41
 8                    </div>
41
 9                </fieldset>
42
 0
42
 1                <!— Referees —>
42
 2                <fieldset>
42
 3                    <legend>Referees</legend>
```

```
                    <div class="referees-grid">
                        <div class="referee-card">
                            <h3>Referee 1</h3>
                            <div class="form-group">
                                <label for="ref1Name">Full Name</label>
                                <input
                                    type="text"
                                    id="ref1Name"
                                    name="ref1Name"
                                    required
                                />
                            </div>
                            <div class="form-group">
                                <label for="ref1Occupation">Occupation</labe
                                <input
                                    type="text"
                                    id="ref1Occupation"
                                    name="ref1Occupation"
                                    required
                                />
                            </div>
                            <div class="form-group">
```

```
44
 6                                <label for="ref1Relationship"
44
 7                                    >Relationship</label
44
 8                                >
44
 9                                <input
45
 0                                    type="text"
45
 1                                    id="ref1Relationship"
45
 2                                    name="ref1Relationship"
45
 3                                    placeholder="e.g. Former Manager"
45
 4                                    required
45
 5                                />
45
 6                            </div>
45
 7                            <div class="form-group">
45
 8                                <label for="ref1Address">Address</label>
45
 9                                <textarea
46
 0                                    id="ref1Address"
46
 1                                    name="ref1Address"
46
 2                                    rows="2"
46
 3                                    required
46
 4                                ></textarea>
46
 5                            </div>
46
 6                            <div class="form-group">
46
 7                                <label for="ref1Phone">Telephone</label>
```

```
                                <input
                                    type="tel"
                                    id="ref1Phone"
                                    name="ref1Phone"
                                    required
                                />
                            </div>
                        </div>


                        <div class="referee-card">
                            <h3>Referee 2</h3>
                            <div class="form-group">
                                <label for="ref2Name">Full Name</label>
                                <input
                                    type="text"
                                    id="ref2Name"
                                    name="ref2Name"
                                    required
                                />
                            </div>
                            <div class="form-group">
                                <label for="ref2Occupation">Occupation</labe
```

```
                                <input
                                    type="text"
                                    id="ref2Occupation"
                                    name="ref2Occupation"
                                    required
                                />
                            </div>
                            <div class="form-group">
                                <label for="ref2Relationship"
                                    >Relationship</label
                                >
                                <input
                                    type="text"
                                    id="ref2Relationship"
                                    name="ref2Relationship"
                                    required
                                />
                            </div>
                            <div class="form-group">
                                <label for="ref2Address">Address</label>
                                <textarea
                                    id="ref2Address"
```

```
                                    name="ref2Address"
                                    rows="2"
                                    required
                                ></textarea>
                            </div>
                            <div class="form-group">
                                <label for="ref2Phone">Telephone</label>
                                <input
                                    type="tel"
                                    id="ref2Phone"
                                    name="ref2Phone"
                                    required
                                />
                            </div>
                        </div>
                    </div>
                </fieldset>

                <button type="submit" id="submitBtn">Submit Application</but
                <div id="formMessage" class="message" aria-live="polite"></d
            </form>
        </div>
```

```
<script>
    document.addEventListener("DOMContentLoaded", () => {
        const form = document.getElementById("applicationForm");
        const slider = document.getElementById("yearsExp");
        const sliderOutput = document.getElementById("yearsExpOutput
        const submitBtn = document.getElementById("submitBtn");
        const messageDiv = document.getElementById("formMessage");


        // Update slider output dynamically
        slider.addEventListener("input", () => {
            sliderOutput.textContent = slider.value;
        });


        // Handle form submission
        form.addEventListener("submit", (e) => {
            e.preventDefault();
            messageDiv.className = "message"; // Reset classes
            messageDiv.style.display = "none";


            // Trigger native HTML5 validation
            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();
```

```
                        sliderOutput.textContent = "0";
                        submitBtn.disabled = false;
                        submitBtn.textContent = "Submit Application";


                        // Scroll to message
                        messageDiv.scrollIntoView({
                            behavior: "smooth",
                            block: "nearest",
                        });
                    }, 1500);
                });
            });
        </script>
    </body>
</html>
```

Edit this page

Previous
Session 1

Next
Session 3