**UX Instructions Exercise 4**

**Part 1: Breaking Down the HTML Form From the Starter**

* This starts the SPA section for "Contact." It is hidden or shown based on the current hash (like #contact).

<div id="contact" class=“section">

* Basic section heading and description text.

<h2>Contact Us</h2>

<p>Please fill out the form below, and we’ll get back to you shortly.</p>

The form element:

* This begins the form.
* id="contact-form" lets us reference it in JavaScript.
* novalidate disables the browser's default validation bubbles so you can customize validation messages if needed.

<form id="contact-form" novalidate>

**Each .form-group includes:**

* label is semantically connected to its input via for="..." and id=“..." and is an accessibility feature. Try clicking the text in label - what happens?
* required is a boolean attribute that makes it a required field.
* placeholder gives hint text inside the input.

<div class="form-group">

<label for="name">Full Name \*</label>

<input type="text" id="name" name="name" required placeholder="Your full name" />

</div>

* Textarea is for a longer message input, also required in this form.

<textarea ... rows="6"></textarea>

* Submits the form to the action attribute’s file in the form tag.

<button type="submit">Send Message</button>

**Part 2: Add CSS**

**Top-Level Form Styles**

|  |  |
| --- | --- |
| **Rule** | **What It Does** |
| max-width: 600px | Prevents the form from getting too wide — makes it centered and readable |
| margin: 0 auto | Horizontally centers the form in the page |
| padding: 1rem | Adds space inside the form boundary |
| display: flex; flex-direction: column | Stacks child elements vertically using Flexbox |
| gap: 1.5rem | Adds spacing between each form element |
| font-family: Arial, sans-serif | Sets a clean, readable font |

#contact-form {

max-width: 600px;

margin: 0 auto;

padding: 1rem;

display: flex;

flex-direction: column;

gap: 1.5rem;

font-family: Arial, sans-serif;

}

**.form-group Structure**

* Each label/input pair is a **group** that stacks its label on top of the input or textarea.
* Helps maintain vertical alignment and spacing.

#contact-form .form-group {

display: flex;

flex-direction: column;

}

**Labels**

* Labels are **bold**, slightly spaced from the field below, and easy to read.
* Improves usability and accessibility.

#contact-form label {

margin-bottom: 0.5rem;

font-weight: bold;

font-size: 1rem;

}

**Inputs and Textareas**

|  |  |
| --- | --- |
| **Style** | **Purpose** |
| padding | Creates inner space for comfortable typing |
| border | Light gray border for visual clarity |
| border-radius: 5px | Rounded corners for modern feel |
| transition | Smooth effect when the field is focused (see next section) |

#contact-form input,

#contact-form textarea {

padding: 0.75rem;

font-size: 1rem;

border: 1px solid #ccc;

border-radius: 5px;

transition: border-color 0.2s ease, box-shadow 0.2s ease;

}

**Focus States**

|  |  |
| --- | --- |
| **Style** | **Purpose** |
| border-color | Changes to indigo on focus, signaling active input |
| box-shadow | Adds a subtle glow for clarity |
| outline: none | Hides the default browser outline (replaced with custom visual cues) |

#contact-form input:focus,

#contact-form textarea:focus {

border-color: #4F46E5;

box-shadow: 0 0 5px rgba(79, 70, 229, 0.3);

outline: none;

}

**Placeholder Styling**

* Sets a **lighter gray color** for placeholder text so it looks soft and non-intrusive.

#contact-form input::placeholder,

#contact-form textarea::placeholder {

color: #999;

}

**Date Input Styling**

* Limits the width of date input so it doesn't stretch the full form width.

#contact-form input[type="date"] {

max-width: 250px;

}

**Submit Button**

|  |  |
| --- | --- |
| **Style** | **Purpose** |
| align-self: start | Aligns the button to the left side (instead of stretching full-width) |
| padding | Gives it a nice click-friendly area |
| background | Indigo base color |
| transition | Smooth hover and focus states |

#contact-form button[type="submit"] {

align-self: start;

padding: 0.75rem 1.5rem;

font-size: 1rem;

background: #4F46E5;

color: white;

border: none;

border-radius: 5px;

cursor: pointer;

transition: background 0.3s ease, box-shadow 0.3s ease;

}

**Hover & Focus States for Button**

* On hover: the button gets darker and a subtle glow.

#contact-form button[type="submit"]:hover {

background: #4338ca;

box-shadow: 0 0 10px rgba(79, 70, 229, 0.5);

}

* On focus (keyboard users): custom accessible outline appears.

#contact-form button[type="submit"]:focus {

outline: 2px solid #a5b4fc;

outline-offset: 2px;

}

**This CSS ensures your form is:**

|  |  |
| --- | --- |
| **Trait** | **Achieved by** |
| Clean & readable layout | Flexbox, spacing, font |
| Visually appealing | Shadows, border radius, color transitions |
| Mobile-friendly | Max-width, vertical layout, responsive spacing |
| Accessible | Focus states, clear labels, usable buttons |

**Part 3: Enhanced Form with Validations and New Fields**

Create an improved form with:

* type="email" with regex pattern
* type="date" for birthdate
* Real-time validation feedback
* minlength, maxlength, and pattern attributes
* Remove the novalidate attribute

<form id="contact-form">

<div class="form-group">

<label for="name">Full Name \*</label>

<input type="text" id="name" name="name" required placeholder="Your full name" minlength="2" maxlength="50" />

</div>

<div class="form-group">

<label for="email">Email Address \*</label>

<input

type="email"

id="email"

name="email"

required

placeholder="you@example.com"

pattern="^[a-zA-Z0-9.\_%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$"

/>

</div>

<div class="form-group">

<label for="subject">Subject \*</label>

<input type="text" id="subject" name="subject" required placeholder="Subject of your message" maxlength="100" />

</div>

<div class="form-group">

<label for="birthdate">Your Birthdate</label>

<input type="date" id="birthdate" name="birthdate" max="2025-12-31" />

</div>

<div class="form-group">

<label for="message">Message \*</label>

<textarea

id="message"

name="message"

rows="6"

required

placeholder="Write your message here..."

minlength="10"

maxlength="500"

></textarea>

</div>

<button type="submit">Send Message</button>

</form>

**Added Features**

|  |  |  |
| --- | --- | --- |
| **Feature** | **Tag used** | **Purpose** |
| minlength, maxlength | input, textarea | Enforce field length |
| type="email" | Validates email format | Built-in + custom pattern supported |
| pattern (regex) | Custom format enforcement | Stronger email or phone validation |
| type="date" | input calendar support | Easy UX for dates |
| novalidate | Form tag | Lets you use JS to customize validation |

Let’s add a dynamic field. When the user selects “Business”, we dynamically add a new input field for the Business Name.

**HTML**

**Add to your form:**

* <select id="user-type"> is the trigger.
* <div id="dynamic-fields"> is the target container where new fields will be inserted.

<select id="user-type" name="user-type">

<option value="">Please select</option>

<option value="individual">Individual</option>

<option value="business">Business</option>

</select>

<div id="dynamic-fields"></div>

**Add JavaScript**

// Get references to the dropdown and placeholder container

const userType = document.getElementById("user-type");

const dynamicFields = document.getElementById("dynamic-fields");

// When the dropdown changes, trigger this function

userType.addEventListener("change", () => {

// Step 1: Clear any previously added fields

dynamicFields.innerHTML = "";

// Step 2: If "business" is selected, add the extra field

if (userType.value === "business") {

// Create a new div to act like a form-group

const wrapper = document.createElement("div");

wrapper.className = "form-group";

// Create the label for the business input

const label = document.createElement("label");

label.setAttribute("for", "business-name");

label.textContent = "Business Name \*";

// Create the actual input field

const input = document.createElement("input");

input.type = "text";

input.id = "business-name";

input.name = "business-name";

input.required = true;

input.placeholder = "Your company name";

// Add label and input into the wrapper

wrapper.appendChild(label);

wrapper.appendChild(input);

// Add the wrapper into the dynamic area of the form

dynamicFields.appendChild(wrapper);

}

});

**Add CSS for the pull-down (select tag)**

#contact-form select {

appearance: none;

-webkit-appearance: none;

-moz-appearance: none;

background-color: white;

padding: 0.75rem;

font-size: 1rem;

border: 1px solid #ccc;

border-radius: 5px;

background-image: url("data:image/svg+xml,%3Csvg viewBox='0 0 10 6' xmlns='http://www.w3.org/2000/svg'%3E%3Cpath d='M0 0l5 6 5-6z' fill='%23999'/%3E%3C/svg%3E");

background-repeat: no-repeat;

background-position: right 0.75rem center;

background-size: 12px 8px;

transition: border-color 0.2s ease, box-shadow 0.2s ease;

}

#contact-form select:focus {

border-color: #4F46E5;

box-shadow: 0 0 5px rgba(79, 70, 229, 0.3);

outline: none;

}