Here's a comprehensive assignment that combines HTML, CSS, and JavaScript for building a simple contact form with validation and styling.

---

## \*\*Assignment: Contact Form with Validation\*\*

### \*\*Objective\*\*

The objective of this assignment is to create a fully functional contact form using HTML, CSS, and JavaScript. You will implement form validation to ensure user inputs are correct and visually enhance the form using CSS.

### \*\*Task Description\*\*

You are required to create a contact form that collects user data, including their name, email, and comments. The form should validate the input before submission and provide user feedback. Additionally, you will apply CSS styles to make the form visually appealing.

### \*\*Requirements\*\*

1. \*\*HTML Structure\*\*:

- Create a new HTML file (e.g., `contact-form.html`) with the following elements:

- A form (`<form>`) that includes:

- An input field for the user's name (`<input type="text">`).

- An input field for the user's email (`<input type="email">`).

- A textarea for user comments or messages (`<textarea>`).

- A submit button (`<button>`).

- A section to display error messages if validation fails.

- A section to display a success message upon successful form submission.

2. \*\*CSS Styling\*\*:

- Create a new CSS file (e.g., `styles.css`) to style your form. Include styles for:

- The form layout (margin, padding, and alignment).

- Input fields, textarea, and button (background color, border, padding, etc.).

- Error messages (color and visibility).

- Success messages (color and visibility).

3. \*\*JavaScript Validation\*\*:

- Write a `<script>` section in your HTML file or link to an external JavaScript file (e.g., `form-validation.js`).

- Implement the following validation requirements:

- Ensure the name field is not empty and display an error message if it is.

- Ensure the email field contains a valid email format and display an error message if it does not.

- Ensure the comments field is not empty and display an error message if it is.

- If all fields are valid, display a success message and allow the form to be submitted.

### \*\*Example HTML Structure\*\*

Here's a sample structure of how your HTML might look:

```html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Contact Form</title>

<link rel="stylesheet" href="styles.css"> <!-- Link to your CSS file -->

</head>

<body>

<div class="form-container">

<h1>Contact Us</h1>

<form id="contact-form">

<label for="name">Name:</label>

<input type="text" id="name" name="name" required>

<span id="name-error" class="error-message"></span> <!-- Error message for name -->

<label for="email">Email:</label>

<input type="email" id="email" name="email" required>

<span id="email-error" class="error-message"></span> <!-- Error message for email -->

<label for="comments">Comments:</label>

<textarea id="comments" name="comments" required></textarea>

<span id="comments-error" class="error-message"></span> <!-- Error message for comments -->

<button type="submit">Submit</button>

</form>

<div id="success-message" class="success-message"></div> <!-- Success message area -->

</div>

<script src="form-validation.js"></script> <!-- Link to your JavaScript file -->

</body>

</html>

```

### \*\*Example CSS Styles\*\*

Here's a sample CSS code snippet to help you style the form:

```css

/\* styles.css \*/

body {

font-family: Arial, sans-serif;

background-color: #f4f4f4;

padding: 20px;

}

.form-container {

background-color: white;

border-radius: 5px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

padding: 20px;

max-width: 400px;

margin: auto;

}

h1 {

text-align: center;

}

label {

display: block;

margin-bottom: 5px;

}

input[type="text"],

input[type="email"],

textarea {

width: 100%;

padding: 10px;

margin-bottom: 10px;

border: 1px solid #ccc;

border-radius: 4px;

}

button {

width: 100%;

padding: 10px;

background-color: #28a745;

color: white;

border: none;

border-radius: 4px;

cursor: pointer;

}

button:hover {

background-color: #218838;

}

.error-message {

color: red;

font-size: 0.9em;

}

.success-message {

color: green;

font-size: 1em;

margin-top: 10px;

text-align: center;

}

```

### \*\*Example JavaScript Code\*\*

Here's a sample JavaScript code snippet to help you with form validation:

```javascript

// form-validation.js

// Function to validate the form

function validateForm(event) {

// Prevent default form submission

event.preventDefault();

// Clear previous error messages

document.getElementById("name-error").innerHTML = "";

document.getElementById("email-error").innerHTML = "";

document.getElementById("comments-error").innerHTML = "";

document.getElementById("success-message").innerHTML = "";

// Get form field values

var name = document.getElementById("name").value.trim();

var email = document.getElementById("email").value.trim();

var comments = document.getElementById("comments").value.trim();

var isValid = true;

// Validate name

if (name === "") {

document.getElementById("name-error").innerHTML = "Name is required.";

isValid = false;

}

// Validate email

var emailPattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/; // Basic email pattern

if (email === "") {

document.getElementById("email-error").innerHTML = "Email is required.";

isValid = false;

} else if (!emailPattern.test(email)) {

document.getElementById("email-error").innerHTML = "Please enter a valid email address.";

isValid = false;

}

// Validate comments

if (comments === "") {

document.getElementById("comments-error").innerHTML = "Comments are required.";

isValid = false;

}

// If the form is valid, show success message and clear the form

if (isValid) {

document.getElementById("success-message").innerHTML = "Form submitted successfully!";

document.getElementById("contact-form").reset(); // Clear form fields

}

}

// Add event listener to the form

document.getElementById("contact-form").addEventListener("submit", validateForm);

```

### \*\*Submission Guidelines\*\*

- Submit your HTML file (`contact-form.html`), your CSS file (`styles.css`), and your JavaScript file (`form-validation.js`).

- Ensure that your JavaScript code functions correctly and provides the expected validation feedback.

- Validate your HTML and CSS for any errors.

### \*\*Evaluation Criteria\*\*

- Correct implementation of form validation techniques.

- Clear and meaningful error messages that guide the user.

- Appropriate styling that enhances the user experience.

- Overall user experience and responsiveness of the form.

### \*\*Additional Tips\*\*

- Use browser developer tools to debug and test your JavaScript code.

- Experiment with additional validation rules, such as minimum character lengths for the name or comments.

---

This assignment combines HTML, CSS, and JavaScript to provide a holistic learning experience. Students will learn how to create a functional form, apply styles, and implement validation effectively.