Simple Contact Form Project

Let's build a **Simple Contact Form Project** with accessibility and responsive design. This form will collect basic information (name, email, subject, and message) and be accessible, structured, and user-friendly.

Project Overview: Simple Contact Form

This contact form will include:
- A form title
- Fields for name, email, subject, and message
- Accessible labels for each field
- A submit button
- Basic CSS for layout and responsiveness

Project Structure

Here's the file structure:
```
contact-form-project/
├── index.html
└── styles.css
```

Step 1: HTML File (index.html)

Create a basic HTML form with labels, inputs, and a submit button. Add placeholder text, required fields, and ARIA attributes to enhance accessibility.

```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">
```

```
</head>
<body>
  <div class="form-container">
    <h1>Contact Us</h1>
    <form action="/submit-form" method="POST">

      <label for="name">Name:</label>
      <input type="text" id="name" name="name" placeholder="Your Name" required>

      <label for="email">Email:</label>
      <input type="email" id="email" name="email" placeholder="Your Email" required>

      <label for="subject">Subject:</label>
      <input type="text" id="subject" name="subject" placeholder="Subject" required>

      <label for="message">Message:</label>
      <textarea id="message" name="message" rows="4" placeholder="Your Message" required></textarea>

      <button type="submit">Send Message</button>
    </form>
  </div>
</body>
</html>
```

Step 2: CSS File (styles.css)

Let's style the form to be clean and responsive:

```css
/* Basic Reset */
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

body {
  font-family: Arial, sans-serif;
  display: flex;
  justify-content: center;
  align-items: center;
```

```css
  height: 100vh;
  background-color: #f9f9f9;
}

.form-container {
  width: 100%;
  max-width: 400px;
  padding: 2rem;
  background-color: #fff;
  box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.1);
  border-radius: 8px;
}

h1 {
  text-align: center;
  color: #333;
  margin-bottom: 1.5rem;
}

form label {
  font-weight: bold;
  margin-top: 1rem;
  display: block;
  color: #555;
}

form input[type="text"],
form input[type="email"],
form textarea {
  width: 100%;
  padding: 0.75rem;
  margin-top: 0.5rem;
  border: 1px solid #ddd;
  border-radius: 4px;
  font-size: 1rem;
}

form textarea {
  resize: vertical;
}

button[type="submit"] {
  width: 100%;
  padding: 0.75rem;
```

```
  font-size: 1rem;
  font-weight: bold;
  color: #fff;
  background-color: #007bff;
  border: none;
  border-radius: 4px;
  cursor: pointer;
  margin-top: 1.5rem;
}

button[type="submit"]:hover {
  background-color: #0056b3;
}
```

Step 3: Enhancing Accessibility

Incorporate these tips to improve accessibility:
- Labels: Ensure each `<input>` and `<textarea>` has a `<label>`.
- ARIA Roles (Optional): Add `aria-label` or `aria-describedby` where needed for additional context.

Step 4: Testing Responsiveness and Accessibility

1. Resize the Browser: Ensure the form adjusts to different screen sizes.
2. Accessibility Check: Use a screen reader tool or `tab` through the form to check that labels and input fields are correctly connected.

Optional Enhancements

1. Form Validation: Use HTML5 `required` attributes and pattern validation.
2. Thank You Page: Redirect to a thank you page upon submission.
3. Form Success Alert: Add a message that shows after submission (you can add this with JavaScript).

This project will give you a solid practice in building accessible forms, structuring HTML, and basic styling. Let me know if you'd like to add interactivity or a backend!