**Prompts for Creating a Credit Card Form in Angular Using Cursor AI**

**Step 1: Create the Component**

**Prompt:**

**Create a new Angular component named `CreditCardFormComponent`. The component should have:**

**- A TypeScript file for logic.**

**- An HTML template for the form structure.**

**- A CSS file for styling the form.**

**Step 2: Define the Form Structure**

**Prompt:**

**Update the `CreditCardFormComponent` to include a reactive form using Angular's `FormBuilder`.**

**The form should have the following fields:**

**1. Cardholder Name (text input)**

**2. Card Number (text input with validation for 16 digits)**

**3. Expiry Date (month/year input)**

**4. CVV (numeric input with validation for 3 digits)**

**5. Submit Button**

**Add proper placeholders and labels for each field.**

**Step 3: Add Form Validation**

**Prompt:**

**Enhance the `CreditCardFormComponent` to include validation rules:**

**1. Cardholder Name: Required.**

**2. Card Number: Required, 16 digits, numeric only.**

**3. Expiry Date: Required, valid future date.**

**4. CVV: Required, 3 digits, numeric only.**

**Display error messages below each field if validation fails.**

**Step 4: Implement Form Styling**

**Prompt:**

**Style the credit card form using CSS to make it visually appealing. Ensure the form:**

**- Is centered on the page.**

**- Has padding and borders for each input field.**

**- Uses a modern and clean font.**

**- Has hover and focus effects for better user experience.**

**Use SCSS if your Angular project supports it.**

**Step 5: Handle Form Submission**

**Prompt:**

**Add a `submitForm` method to the `CreditCardFormComponent` that:**

**1. Matches the submitted form data with a provided value in a JSON file.**

**2. Navigates to a successful page if the data matches.**

**3. Logs an error message and prevents navigation if the data does not match.**

**Ensure:**

**- The form only submits if all validations pass.**

**- A loading spinner is shown when the form is being submitted.**

**- A success message is displayed after successful submission.**

**Provide an example JSON file with mock data for matching.**

**Step 6: Create a Credit Card Directive (Optional)**

**Prompt:**

**Create a custom directive called `CreditCardMaskDirective` that formats the card number input as the user types. The directive should:**

**- Add a space after every 4 digits (e.g., 1234 5678 9012 3456).**

**- Prevent non-numeric characters.**

**- Work seamlessly with the form.**

**Step7:**

**Prompt**

**Style the form as per the attached image**