**Define the rules of a component via cursor rules.**  
# Component Structure

- \*\*PascalCase\*\* for component names (e.g., `CreditCardFormComponent`).

- Place components in `/src/app/components/`.

- Use Angular reactive forms for managing input states.

- Group related files in the same folder (e.g., component, HTML, CSS, spec files).

# Naming Conventions

- \*\*Variables/Functions\*\*: Use camelCase (e.g., `validateCardNumber`).

- \*\*Files\*\*: Use kebab-case (e.g., `credit-card-form.component.ts`).

# Validation Standards

- \*\*Card Number\*\*:

  - Validate using the Luhn algorithm.

  - Provide error messages for invalid formats and length.

- \*\*Expiration Date\*\*:

  - Ensure MM/YY format and check if the date is in the future.

- \*\*CVV\*\*:

  - Validate as exactly 3 digits for most cards or 4 digits for AmEx.

- \*\*Amount\*\*:

  - Ensure it's a valid, positive number, properly formatted (e.g., USD).

# Redirection

- Use Angular Router's `Router.navigate` for navigation.

- Redirect users only after successful validation.

# Type Safety

- Avoid `any` type. Use strict types like `CardDetails`, `ValidationError`.

- Create reusable validation logic in `/src/app/utils/validators.ts`.

# UI/UX

- Provide real-time validation feedback for each field.

- Show clear error messages and success cues.

# Testing

- Write unit tests with Jasmine and Karma.

- Test validation functions (e.g., `validateCardNumber`) and routing.

# Security

- Never store full card details or sensitive information on the client side.

- Transmit all data over HTTPS and validate server-side.

# Tooling & Dependencies

- Avoid adding external libraries for validation unless pre-approved.

- Use lightweight and necessary libraries only.

# Accessibility

- Ensure all form fields have `aria-label` or `aria-describedby` attributes.

- Provide keyboard navigation and focus states for interactive elements.

# Automation & Housekeeping

- Follow semantic versioning.

- Update `CHANGELOG.md` for new features or changes.

# Do's and Don'ts

\*\*Do:\*\*

- Validate form data on both client and server.

- Redirect users after successful validation.

\*\*Don't:\*\*

- Never redirect users prematurely.

- Never store sensitive information in the frontend.

**Prompts**

1. **Generate a credit card component according to the rules and style it as shown in the attached image.**
2. **The next prompts will be based on the requirements.**