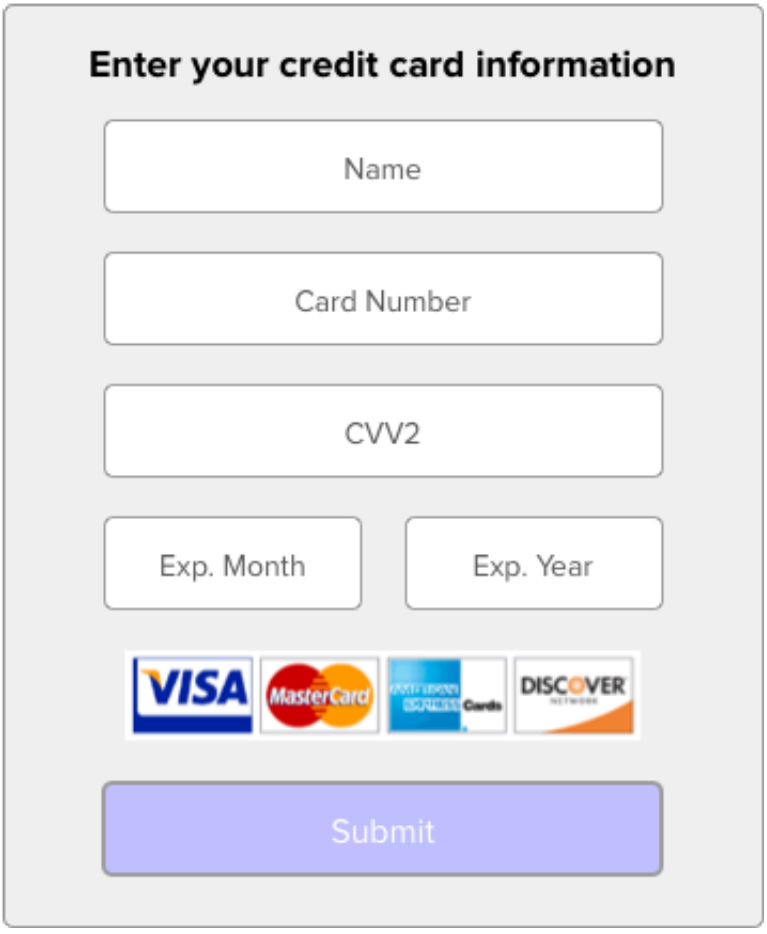
Affirm · Independent Coding Exercise, Front End

Overview

Congratulations on moving on to the next phase of your interview with Affirm. In this challenge, you will design and build a form that allows users to submit credit card information using the modern front-end tools of your choice. We suggest a time box of about 3 hours for this project.

Design Wireframe



Project Spec

Create a project that renders a UI allowing for users to enter their credit card information. Your solution should satisfy following criteria:

1. The solution should be implemented using a modern Javascript framework. At Affirm, our front-end teams use React. That said, you are free to choose the framework you are most comfortable with.
2. The user interface should provide sufficient input validation. If there is a validation error, you should surface it to the user.
3. The user interface should be properly styled. The included wireframe provides a basic idea of the desired look, but you are free to be creative and refine the design. Please **do not use a UI framework, such as Bootstrap, Ant Design, etc. We want to see you style your own components!**
4. Your submission should include basic test coverage of your code.

Beyond these basic criteria, feel free to explore interface features that manipulate the user's inputs to reduce the occurrence of errors or provide easier understanding of input errors. For example, consider how we might give the user early feedback as they enter long strings of numbers.

Inputs

The user should be able to enter the following fields:

- Name
- Credit Card Number
- CVV2
- Expiration Date (Month and Year)

There are a handful of established standards credit card numbers. For simplicity, your solution should only consider Visa and AMEX.

Visa

- Credit Card Numbers
  - 4 groups of 4 numbers
  - 16 characters total
  - Start with "4"
- CVV2
  - 3 characters

AMEX

- Credit Card Numbers
  - 4-6-5 number grouping
  - 15 characters total
  - Start with "34" or "37"
- CVV2
  - 4 characters

Expiration dates are considered valid if they are after the current month and year. For example, in September 2020, October 2020 (10/2020) and onward are valid. Keep these constraints in mind when validating user input.

Deliverables

Your final deliverable should include a project that implements the spec outlined above. Please submit a folder containing the source code to your program. Please ensure that your submission includes clear instructions on how to run the project. If your implementation is using React, we highly recommend using create-react-app.

We want to spend our time evaluating your work, not working to run your project. Please ensure that you make it easy to run your project, and that you include any information needed to do so.

Evaluation

Your project will be evaluated based on the following criteria, in decreasing order of importance:

1. *Correctness*: Can the user enter credit card information, and be shown an error if their input fails to validate?
2. *Clarity*: Is the code well-organized, easy to read and extensible? Is it well-tested?
3. *Usability*: Is the UI styled properly? Is it easy to make mistakes when entering information? Does the interface help correct mistakes?
4. *Extensibility*: Can this component be reused and dropped onto a page? Is it easy to add additional credit card types and their constraints?

Final Notes and Suggestions

- We suggest starting with a simple, straight-forward interface that meets the requirements first, then to iterate on user experience improvements.
- It's not necessary to support multiple browsers. Please target a modern version of Chrome, and note any browser constraints that the engineer evaluating your solution should consider.
- Feel free to use any language (e.g., one that cross compiles to JavaScript), libraries, frameworks, or tools that makes your job easier. We suggest to not go overboard with front end tooling; consider the problem domain and how your choice of tools may affect the portability and reusability of your code.
- Feel free to express your skills and creativity with styling and design choices. We will not penalize solutions for sticking with barebones design. Consider it an opportunity to showcase your front-end skills.