1. How long did you spend working on the problem? How much time did you spend

thinking about the design before writing your code?

It actually didn't take me long time to have a basic idea of the entire layout, and what took me the longest is actually due to attempting to test dynamic content (for example, if one of the input field container error, an error would shows up under the input field). I wasn’t able to find a way to test this properly, and so I decided to simply test whether my error message was set correctly.

2. What are the UI/UX usability features you implemented, or thought about implementing? How do they help validate the user input?

Input field that should consists of integer only should only allow user to enter numeric input. I decided to add logic for preventing user from entering character other than numeric input.

Also, I was planning to use drop down list for expiration month and year rather than using just a simple input field, but have decided to use input fields instead.

3. What would a form submission/API payload of this look like? How would you deal with validation errors that may come from that API response?

{

card: {

name: “John Doe”,

number: 4444444444444444,

cvv: 123,

expMonth: 08,

expYear: 20

}

}

Ideally the card number, cvv, expiration month and date should be encoded first, and then place in a JSON object in order to submit to the API endpoint.

Validation errors should be rendered under each corresponding input field.

4. What are some styling and layout considerations for these types of form inputs?

Styling should take consideration of form dimension to make sure that content within the form will not expand parent height.