FE Technical Assignment

Thank you for applying to our Front-end developer role at Vaimo. For the next step of our application process, we ask that you complete the following technical assignment that we have devised with technical and visual developers in mind. Carefully read what to do and what not to do. Good luck!

Assignment

1. Implement a simple product detail page matching the provided mobile and desktop designs:
    a. Designs:
        i. Desktop: https://marvelapp.com/prototype/619dgg5
        ii. Mobile: https://marvelapp.com/prototype/28jb3a3g
    b. Data for the product must be fetched from our API - https://bit.ly/30KQk2j. No hardcoded HTML markup, unless for elements that are not provided by API.
2. QTY rockers next to product options should function (+ and - buttons increase and decrease QTY) as well as QTY itself must be an input field that accepts valid numbers:
    a. Lowest allowed value is 0. If the value is 0, - button becomes grayed out and non-clickable.
    b. Increment step is 1 (if value is 3 and you press “+”, value changes to “4” and so forth).
    c. If an incorrect value is entered using the input field (e.g. -10), on blur-event that value should change to the last valid value used instead.
    d. Having QTY 1 or more means the option is primed to be added to cart and must appear in the right hand sidebar above shipping info.
3. Right sidebar should contain product options with QTY 1 or more (option summary) and the calculated price for each of them:
    a. If an option with a price of R10 has QTY=2, the price in summary should be R20.
4. **Optional**: API returns info for the shipping time and lead time. Add a tooltip on hover that displays it.
5. **Optional**: have the discount timer count down to the discount end date provided by the API.
    a. Don't worry about the state when the countdown timer reaches the end.
6. **Optional**: Have the QTY rocker be a shareable component that can be configured to have different min value, max value and step increment value.

Areas of judgement

- How well your implementation **matches the provided design**.
- That all listed requirements are fulfilled.
- How elegant the solution is in terms of code structure and readability.
- Extra visual or functional features are a plus.

Technical requirements:

1. **Design must be pixel perfect**. Use Marvel's Handoff feature to get accurate spacings, font sizes, colours, etc. The font used is Roboto. All images and icons can be downloaded from Marvel Handoff by clicking on them and then clicking “download” on the right side.
2. You need to use Javascript. We prefer React, but you can choose what you feel comfortable with.
3. Data must be fetched via the Rest API that we have provided you with. No hardcoded markup for data available on the API (product name, reviews, options with pricing, discount info, shipping info).
4. The design implementation has to be responsive. We provide mobile and desktop designs at 360px and 1320px widths. How it looks between those breakpoints is up to you.
5. Do not use CSS frameworks, like Bootstrap or Tailwindcss. We encourage you to use CSS preprocessors like Less or Sass.
6. Solution must work and look the same on all major browsers within the last 2 versions:

a. You can use Browserlist with query “last 2 major versions, not dead, not ie 11” to determine browsers that need to be supported.
b. **No need to support Internet Explorer**

What are our expectations

Depending on your level of experience we of course expect different solutions. Make choices based on what you think you can do within the given time frame. We encourage you to use frameworks like React or Vue.js, but you can also go the plain Javascript route. Ideally you will have prepared every component in its most practical form.

You can choose where to put your focus on:

- Are you more visual detail oriented, then wow us with your interpretation of hover effects and transition. Think of what happens when buttons get pressed, how QTY changes in QTY rocker or how option summary updates, as well as overall loading state while product data is being fetched - what does the user see before data is ready and how does everything appear once data is available and presented. Be as creative as possible while still having a user-friendly interface.
- Or if you're more of a technical developer, a javascript guru - show it to us. Think of the application structure, how you're communicating with the API, does this page warrant some form of global state, can the components be reused.

We don't expect you to complete all the “extra points” assignment items. Nor do we expect you to focus both on visual eye-candy and technical architecture at the same time. Consider your strengths and weaknesses and approach this assignment in a manner that you know you can complete within the timeframe you have been given. We do however expect that the items completed are pixel-perfect when compared to the design. Attention to detail is key.

We do however expect you to try and get to **as close to pixel-perfect as possible**. We are looking for front-end developers who can deliver our designs to our clients as close as possible.

Submission instructions

1. You have 1 week to complete the assignment (ideally it should take 72 hours total).
2. Put your code on GitHub (or Gitlab, Bitbucket etc.) and send us the link to your repository where we can find the source code. No ZIP files!
3. Include a readme.md file where you have provided the following:
    - Instructions for compiling and running the code.
    - Brief overview of the architecture.
    - Short summary of your choice of technology stack - why did you choose it and if you were to do it again, would you change anything?
4. If you have any questions before or during the assignment, reach out to your contact person and we will get back to you as soon as possible.