Coding and architecture exercise

Practical exercise asked to applicants to a Doctena IT Developer role - Ux / UI developer

Requirements specification

Technical requirements

You are asked to develop a simple application that meets the following high-level requirements:

<u>MUST HAVE</u>

1. Developed using ECMA Script 6, and industry standard frameworks like ReactJS, Bootstrap, etc. You may chose! But please explain your choice.
2. Mobile-first, responsive design. Your prototype must be optimized to work on a smartphone, and equally well on a large desktop interface.
3. Optimized for performance - think of any specific techniques to reduce page load times, and explain.
4. Must be easily testable by us (i.e. everything must be provided)
5. Code managed on GitHub[1]

Functional requirements[2]

The front-end application is a "notes" (sticky notes, post-it) application that allows the user to input notes (title, date, text), and to see a list of his notes that can be re-arranged in order of priority (by drag&drop, etc). The usual interface elements for adding, modifying, deleting etc. must be present. The notes author must also have the option to add a "reminder" to a note, which must be visible in the interface (i.e. Remind me of this note on the 20th of January at 10h00).

This feature would need to be integrated in an existing web application as one or several "widgets" so that it behaves like a universal note-taking "service" that can be used anywhere and added to any existing website or web application.

You must assume that the backend of this application will be a webservice (API). You are NOT required to implement this backend API (but you can if you want to). It is however

---

[1] You might want to give us early access to your repo so that we can follow your progress!
[2] The functional description is not more precise on purpose - you are free to ask more details if required, and you are free to improvise!

required that you document which calls you would need to make towards the API so that these requirements can be communicated to a back-end developer.

Finally, if you want to go the extra mile, you are free to add any interesting or "cool" features - surprise us!

Expected deliverables

1. Installation / test procedures so that we can "use" your development
2. Any design and architecture documents, schemas, drawings, etc.
3. The source code
4. The documentation, if required, on how to deploy / launch the application
5. Any additional documentation and comments that you judge meaningful
6. Through this exercise, a show-case of your IT developer talent and experience!

What we will judge

1. The look&feel and usability of your implemented user interface
2. Compatibility with browsers
3. Speed / page load and rendering times
4. The overall functional state of your submission
5. The quality of code and documentation
6. The choice and implementation of your chosen architecture
7. Any "cool - but meaningful! - extras" that you would decide to add!

What we will NOT judge

1. The "cleanliness" of the provided diagrams or schemas (can be hand-drawn)
2. Any non-critical functional features that a "professional" product would require ; this remains an exercise / POC
3. The time required to perform this exercise; we prefer quality over speed in this case, while staying within a reasonable period

What's next?

After your submission, we will evaluate your application internally, and next we will organize a remote or local meeting with you to discuss your submission and your choices.