Frontend test

Part 1 (up to 75 minutes)

Please prepare an **auto-complete component** in React.

1. You cannot use any 3rd party libraries - only pure React and internal DOM functions.
2. The function to filter the data should be *asynchronous*. You can use mock data (such as a JSON array), but the function which uses it should be asynchronous (similar to a real REST call).
3. It should have basic working CSS. No need for anything fancy (such as drop-shadows etc), but should look decent.
4. You need to handle all non-standard/edge use-cases - it should have a perfect user-experience.
5. Bonus points if you highlight the matching part of the text and not only showing it.
6. Prefer using plain JS, not TS
7. No external state management libraries (refer to #1 as well), only native React method.
8. Use only class components, feel free to use life-cycle methods if you need.
9. Shortcuts and hacks are perfectly ok - BUT you have to *add comments* on what are you doing there and why. You should either write production ready code or include comments on what needs to be changed for production.
10. Add a README.md file explaining how to run the project.
11. Bonus #1: now implement *also* as functional component with hooks.
12. Bonus #2: load data using a read API call to some resource.

Results need to be in the form of a GitHub repo link.

Part 2 (up to 45 minutes)

Please answer the following questions to the best of your knowledge, in clear English. Elaborate and try to demonstrate the React knowledge you have. Feel free to give examples and use cases.
DO NOT USE ANY WEB OR OTHER RESOURCE.

1. What is the difference between Component and PureComponent? give an example where it might break my app.
2. Context + ShouldComponentUpdate might be dangerous. Can think of why is that?
3. Describe 3 ways to pass *information* from a component to its PARENT.
4. Give 2 ways to prevent components from re-rendering.
5. What is a fragment and why do we need it? Give an example where it might break my app.
6. Give 3 examples of the HOC pattern.
7. what's the difference in handling exceptions in promises, callbacks and async...await.
8. How many arguments does setState take and why is it async.
9. List the steps needed to migrate a Class to Function Component.
10. List a few ways styles can be used with components.
11. How to render an HTML string coming from the server.

Results need to be submitted in a questions.md file, *in the same GitHub repo* as part 1.

Good luck!