realworld one Frontend Developer Test

Prerequisites

Choose any JSON dummy data online generator. This will be your data source that will be used within your application (so called Test Data). The Test Data size should be at least 500 records.

e.g. https://jsonplaceholder.typicode.com/comments

Task

1. Create a React Single Page App with 3 “pages”.
2. Each page should be reachable by a unique route.
3. First page is a landing page which contains links to the other pages.
4. The second page should contain a representation of the Test Data (e.g. list of components for each record).
5. The third page should summarize the rendering performance of the Test Data.

Consider recording measurements of the rendering process and provide retrospective access to this data. There are no strict requirements on what exactly should be tracked as a part of the performance data, however, you could consider time that is spent for rendering of the whole page and its components, and time spent for obtaining the Test Data.

Notes

- We don't have any preference on the way the Test Data is rendered. It could be table, list, graph, etc.
- We don't have any preference on the way the performance data is displayed.
- You might consider not automatically rendering the Test Data but having a ‘Start’ button that will start the process of fetching the Test Data and the performance measurement.
- The visual quality of the UI will not be judged, but some form of CSS should be integrated.

Extension

The task is purposely straightforward. It could make sense to add some features which you think make the App more useful. You might consider:

- Multiple alternative views of the test data (e.g. table, list, grid, graph, etc.)
- Multiple alternative views of the performance data (e.g. table, list, grid, graph, etc.)
- Storing multiple measurements, along with functionality to select, remove, aggregate, etc., said measurements.
- Persisting measurements for future visits to the app (i.e. measurements are not lost on refresh).
- Adding a measurement detail page with routing to each measurement.
- Adding unit tests to ensure the validity of your code.
- Making the app responsive such that it functions correctly and intuitively for various screen sizes
- Integrating a CSS framework for consistent design

Important

Please include a readme file which contains some brief information about your decision-making process while undertaking the test.