micadoCX

Technical Interview Challenge
Full-Stack Developer

Problem Definition

You are required to put together a modern end-to-end React app where its components run in Docker containers. We are looking for an application that makes use of calls through an API to the database to visualise, filter and search the result set.

Data

You are given a sample dataset from Stats NZ website in a CSV format. The data is on daily COVID-19 case (active, recovered and deceased) and test numbers (daily and cumulative) for New Zealand. You might need to transform the data.

Requirements

Make sure to fulfil as many requirements as you can. We are trying to understand your way of thinking, how you approach a particular problem, what tools you use to produce the outcome.

Git Repository

- Feel free to use any git provider you feel comfortable with, i.e., GitLab, GitHub, etc.
- You should have a clear commit history of your steps. Bonus points if you can provide a history that follows “git-flow”-like frameworks.

Technology Stack

Following list provides the primary technology stack we are currently using at Micado.

- React
- NodeJS
- PostgreSQL
- Docker + Docker Compose

App Requirements

You have the freedom to design the look and feel of your app. Feel free to use any level of CSS and/or any framework to pretty things up.

Front-end Components

1. Your front-end should provide a way to display and filter the data.
2. You should have
    - summary metrics at the top of the layout,
    - followed with a visualisation that shows how the cases and tests results are evolving over time,
    - and a table view to provide a more detailed breakdown of the sample data.
3. As a global filter, you should provide a date picker to allow slicing the data in a date range.
4. Your app should provide a drop-down filter for the visualisation and table to switch between cases and number of tests in New Zealand.

Bonus: Selecting a region on your timeseries visualisation would cross-filter summery metrics and the table view. You are welcome to use any open-source frameworks to achieve this.

Infrastructure Components

1. Your app components (front-end, API layer and PostgreSQL) ideally should run in separate Docker containers.
2. We should be able to pull, build and start the application with clear instructions. Please provide a README document to outline necessary steps to build and start the application.