

# Insights Code Challenge

## Data

You will be provided a small JSON payload that will act as data for a small webapp. To summarize the data, it is 1) an array of Classes 2) which all contain an array of Students 3) which all contain an array of Tests. Classes, Students and Tests all contain a small bit of metadata (ex: a Class has a name (Mathematics), a Student has a name (Darth Vader), and a Test has a name (Week 2 exam) and a score (98)).

```
type Classes = {
  className: String;
  students: Array<Students>;
};

type Students = {
  studentName: String;
  tests: Array<Tests>;
};

type Tests = {
  testName: String;
  testScore: Number;
};
```

## Functionality of the Application

Build a React application that, when using the provided JSON as input data, displays the highest level (Classes) of the JSON's data. For example, display all of the Classes in a list of HTML elements.

Provide a way for the user to select a class they are interested in. After the user selects the class, remove the list of Classes from the page and display the list of Students associated with that class. Repeat this functionality so that a user can "drill down" all the way to a list of Tests for a given Student in a given Class.

Example: The page initially shows a list of class names. After clicking on a class name, the page will show a list of students in that class. After clicking on a student, the page will show a list of tests and scores for that test that that given student has completed in the selected class.

## Some things we're looking for and may ask about

- Did you structure the application using semantic HTML where applicable?
- Did you write comments to document your thought process?
- Did you organize your code and components in a sensible way?

## Requirements

- Write the application in React
- If you have experience with it, Typescript is preferred
- You may supply this JSON to your components however you please (hardcoded as a const, delivered by an Express server that runs alongside it, etc)
- You are not being assessed on your knowledge of Javascript build tools. You are free to use build tools (CRA, NextJS, your own personal boilerplate from a previous project, etc) to save time.
- You are not being assessed on your ability to design the UI/UX (we have designers for that!). That being said, feel free to style the application if you'd like to!
- Document your thought process in comments. The more you write, the more we understand what you're doing.
- Be sure to tell us how to build and run your code. Assume we've never seen it before ;)
- If you import any third party libraries outside the standard build tools, please explain their usage and why you included them. This project should be doable without any uncommon libraries.
- This challenge is intended to be respectful of your time. Therefore, unit testing and styling of the application is not a requirement. You are free to implement these if desired, and we may ask about these concepts during the interview.

Extra credit:

Implement "Breadcrumbs", which show the user where they are in the drill-in layers (ex: Classes > Students > Tests). Would you be able to allow the user to traverse back up the layers if they clicked on a breadcrumb?

## Submission

To submit the completed challenge, publish it in a public repository (github, stash, bitbucket, etc) and send your HR representative the link, or you can zip your project directory (**please delete node\_modules**) and send it to your HR representative.

*An experienced React Developer with plenty of coffee completed the basic requirements in approximately 90 minutes.*

## JSON

*This should be supplied to you in the provided .zip as data.json*

```
[
  {
    "className": "Mathematics",
    "students": [
      {
        "studentName": "Darth Vader",
        "tests": [
          {
            "testName": "Week 1 pre-exam",
            "testScore": 93
          },
          {
            "testName": "Week 2 exam",
            "testScore": 80
          },
          {
            "testName": "Week 3 exam",
            "testScore": 70
          },
          {
            "testName": "Week 4 exam",
            "testScore": 90
          }
        ]
      },
      {
        "studentName": "Obi-Wan Kenobi",
        "tests": [
          {
            "testName": "Week 1 pre-exam",
            "testScore": 47
          },
          {
            "testName": "Week 2 exam",
```

```
        "testScore": 81
      },
      {
        "testName": "Week 3 exam",
        "testScore": 99
      },
      {
        "testName": "Week 4 exam",
        "testScore": 48
      }
    ]
  },
  {
    "studentName": "Death Star Engineer",
    "tests": [
      {
        "testName": "Week 1 pre-exam",
        "testScore": 89
      },
      {
        "testName": "Week 2 exam",
        "testScore": 98
      },
      {
        "testName": "Week 3 exam",
        "testScore": 87
      },
      {
        "testName": "Week 4 exam",
        "testScore": 92
      }
    ]
  }
]
},
{
  "className": "Science",
  "students": [
    {
      "studentName": "Darth Vader",
      "tests": [
        {
          "testName": "Week 1 pre-exam",
          "testScore": 83
        },
        {
          "testName": "Week 2 exam",
          "testScore": 40
        },
        {
```

```
        "testName": "Week 3 exam",
        "testScore": 83
    },
    {
        "testName": "Week 4 exam",
        "testScore": 71
    }
]
},
{
    "studentName": "Obi-Wan Kenobi",
    "tests": [
        {
            "testName": "Week 1 pre-exam",
            "testScore": 98
        },
        {
            "testName": "Week 2 exam",
            "testScore": 41
        },
        {
            "testName": "Week 3 exam",
            "testScore": 79
        },
        {
            "testName": "Week 4 exam",
            "testScore": 58
        }
    ]
},
{
    "studentName": "Death Star Engineer",
    "tests": [
        {
            "testName": "Week 1 pre-exam",
            "testScore": 99
        },
        {
            "testName": "Week 2 exam",
            "testScore": 97
        },
        {
            "testName": "Week 3 exam",
            "testScore": 92
        },
        {
            "testName": "Week 4 exam",
            "testScore": 94
        }
    ]
}
```

```
]
  }
    ]
      }
```