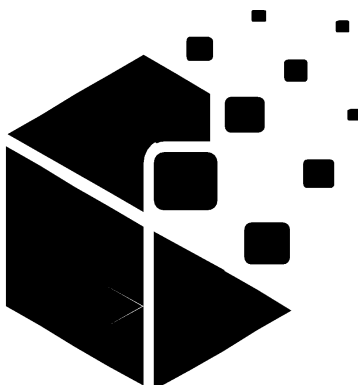


Contilio Code Challenge

Front-end developer (React)

Estimated completion time: 2-3 hours

Last updated: 01/08/2022



Simple Web Application

As a simple demonstration of your coding skills, please build a basic, data-driven 2-page web app using [Create React App](#) that has the following features:

1. Login page

- This page should contain a simple form, centred on the page both horizontally and vertically, consisting of
 - i. Title
 - ii. Simple instruction or welcome message
 - iii. Username input
 - iv. Password input
 - v. Submit button
- Title and instructions can be [lorem ipsum](#) (or just make something up)
- All form elements should be the same width and height, with an equally sized gap between each
- The username and password inputs should both have a *placeholder*
- The form does not need to connect to any web services, nor accept a specific username and password, but the submit button should only be enabled if 1 or more characters have been entered into both the username *and* password input fields
- When enabled, clicking the Submit button should take you to the...

2. Dashboard page

- Save the [JSON data at the end of this document](#) into a file named *data.json* and, using *fetch*, load the data and
 - i. Display the title of the first item at the top of the page; *then below the title*
 - ii. Display each of the *attributes* (name/value pairs) of the first item in a table to the left of the page with column headings "Name" and "Value"; *and next to that horizontally*
 - iii. Display the attributes of the first item as a simple bar chart to the right; *and finally*
 - iv. Display a [horizontal range input](#) below the rest of the content at the bottom of the page that, when dragged by the user, changes the current item and renders the title and attributes in the table and chart on the page above it
- The content of this page should be centred horizontally, with a maximum width of 1000px (or 100% of the browser width, if smaller)
- The width of the table and chart components on this page should always be the same, even if the user resizes their browser
- This page should be reactive, with a breakpoint at 500px that switches the layout of the table and chart from horizontal to vertical on small screens

Notes

- Please use class based React components, *not* function components
- Please use *import*-ed CSS files for styling, *not* inline or JS/JSX
- Please load JSON using *fetch*, do *not* embed or import it
- Please demonstrate ES6 concepts such as destructuring assignment, spreaders and arrow functions, where appropriate
- Please feel free to use third-party [NPM packages](#); you don't need to build everything from scratch
- Please structure your source code into folders in the same way you would if this were a much larger project, rather than leaving everything in the same folder
- Please include a simple JSDoc to explain your coding decisions
- Please make the source code available using a GitHub or GitLab repository

Extras

Consider implementing some or all of the following:

- Styling the web app to approximate <https://contilio.com/> (the font is [Montserrat](#))
- Using SCSS instead of CSS
- Using TypeScript instead of JavaScript
- Including a *Dockerfile* to enable the web app to be deployed in a container

Appendix 1: JSON data file

data.json

```
[
  {
    "title": "Lorem ipsum",
    "attributes": [
      { "name": "length", "value": 10, "unit": "m" },
      { "name": "width", "value": 7, "unit": "m" },
      { "name": "height", "value": 3, "unit": "m" },
      { "name": "weight", "value": 4, "unit": "kg" }
    ]
  },
  {
    "title": "Donec convallis",
    "attributes": [
      { "name": "length", "value": 4, "unit": "m" },
      { "name": "width", "value": 3, "unit": "m" },
      { "name": "height", "value": 2, "unit": "m" },
      { "name": "weight", "value": 1, "unit": "kg" }
    ]
  },
  {
    "title": "Vivamus sed",
    "attributes": [
      { "name": "length", "value": 7, "unit": "m" },
      { "name": "width", "value": 10, "unit": "m" },
      { "name": "height", "value": 2, "unit": "m" },
      { "name": "weight", "value": 8, "unit": "kg" }
    ]
  },
  {
    "title": "Ut tempor",
    "attributes": [
      { "name": "length", "value": 9, "unit": "m" },
      { "name": "width", "value": 9, "unit": "m" },
      { "name": "height", "value": 9, "unit": "m" },
      { "name": "weight", "value": 9, "unit": "kg" }
    ]
  }
]
```