

COMP 4513

Assignment #1: React (*version 2*)

Due Monday February 26, 2018 at midnightish

Changes in **yellow** or **green**

Overview

This assignment provides an opportunity for you to demonstrate your ability to generate a mobile-first, single-page application using React.

Beginning

Use the same techniques covered in the second React lab for this assignment. Use as well the Bulma CSS framework, which will give you the responsive layout to look good at mobile sizes. I will mark the assignment in a browser size similar to a portrait-sized mobile phone.

The GitHub repo for assignment: <https://github.com/rconnolly/comp4513-w2018-assign1.git>.

Submitting

For me to mark your assignment, you will need to share your workspace with me once it is complete. To do so, click the Share button in the upper-right part of the Cloud9 workspace. Invite me via my email (rconnolly@mtroyal.ca) and be sure to give me RW access (otherwise I can't run anything).

Grading

The grade for this assignment will be broken down as follows:

| | |
|--------------------------------------|-----|
| Visual Design and Styling | 20% |
| Programming Design | 15% |
| Functionality (follows requirements) | 65% |

Data Files

You have been provided with two JSON files, `stocks.json` and `userStockPortfolio.json`. The first represents a list of about 70 stocks. The second is a list of stocks that users own (that is, portfolios).

User information will be retrieved via the <https://jsonplaceholder.typicode.com/users> web service.

For the company logos, **place the logos folder within the public folder**. You can then reference them using an absolute reference (e.g., `src='/logos/ABT.svg'`). Some of these logos are quite large; because they are `svg` files they scale to any size, but this will require custom CSS (which you can place inside `App.css`), to specify suitable `width` and `max-width` properties.

You can load a JSON file quite simply in React via a single `import` statement (e.g. `import variable from 'path'`). Once imported, this `variable` will then be a JS object containing the JS object represented by the JSON file, and can be used like any other variable. Variables defined outside the class using the `import` statement are in scope within the class. **Be sure to place JSON files somewhere within the `src` folder (for instance in a data subfolder)**.

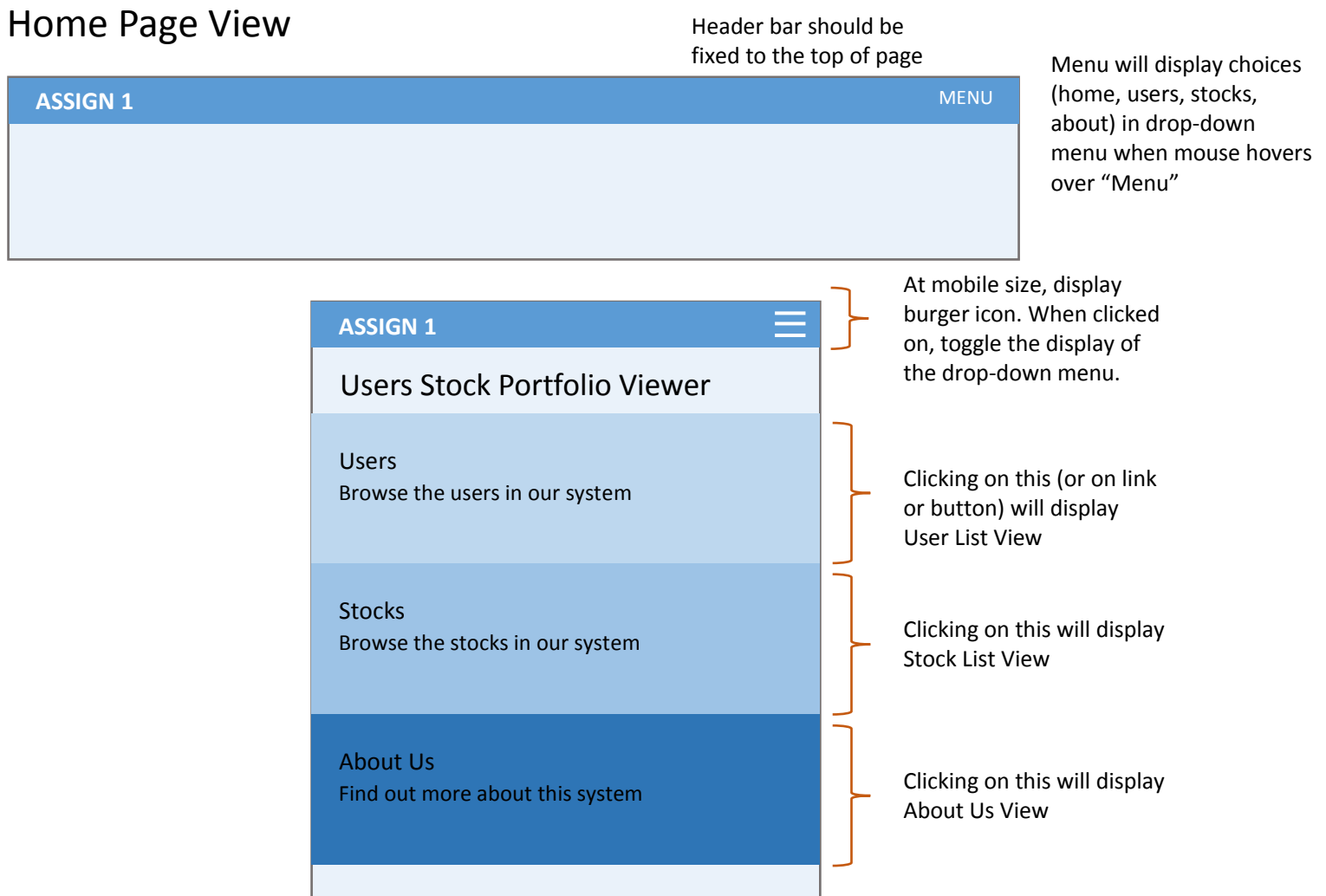
Requirements

1. You must make use of the `create-react-app` starting files and structure.
2. You must use the Bulma CSS Framework. You'll likely need a small amount of additional CSS. I would recommend defining them inside the `App.css` file in the `src` folder.
3. This assignment consists of six views (remember this is a single-page application, so it is best to think of the assignment consisting of different views). In the remainder of this assignment, I have provided a sketch of the basic layout of each view. **They do not show the precise styling; rather they show functionality and content. I will be expecting your pages to look significantly more polished and attractive than these sketches.**
4. You are welcome to use any colors you'd like. I'd recommend, for simplicity sake, sticking with the color modifiers defined by Bulma.
5. If you've used code or CSS that you've found online, I expect you to indicate that in your documentation. Be sure to provide URL of where you found it in the documentation. Failure to do so might result in a zero mark, so give credit for other people's work!

6. When your site loads, it should display the Home Page View. When the user selects a Home links (from breadcrumb or drop-down menu in upper-right corner), it should display this Home Page View. Again, I would expect your version to be substantially more attractive than this sketch, perhaps adding images and icons.

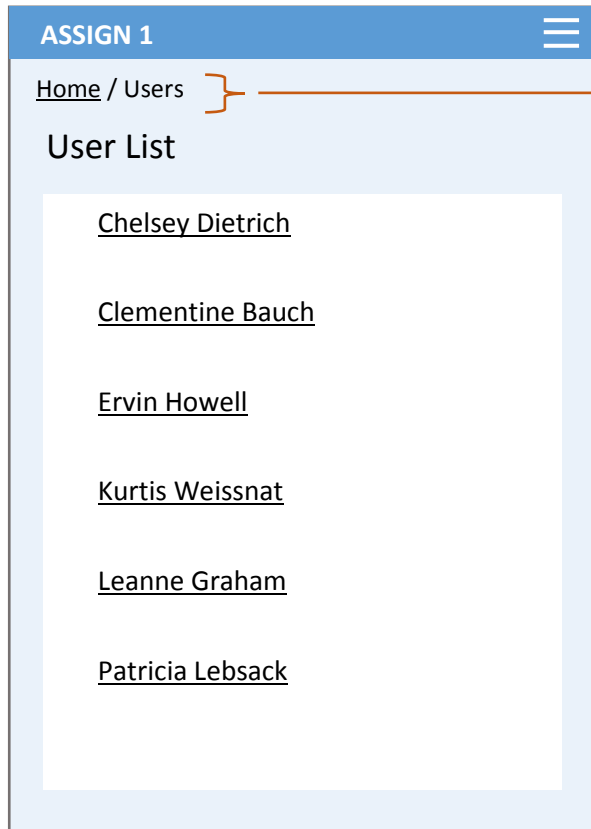
If you use the Bulma navbar element, the so-called burger icon will automatically be displayed (if you've added the markup for it) when browser is resized to mobile dimensions, as well as automatically display the Menu when browser resized back to larger dimensions. What won't happen automatically is the display of the drop-down menu when the user clicks on the burger icon. You will have to add a click handler in React and simply toggle the "is-active" class for the <div> wrapping the drop-down menu.

Home Page View



7. When the user selects a Users links (from home page, breadcrumb, or drop-down menu in upper-right corner), it should display the User View. The data comes from the web service. You will have to sort it after you receive the data. To sort an array or objects by a property value, you will use the JS sort function.

User List View



Breadcrumb navigation (see Bulma documentation). Each underlined word indicates a link to the appropriate view

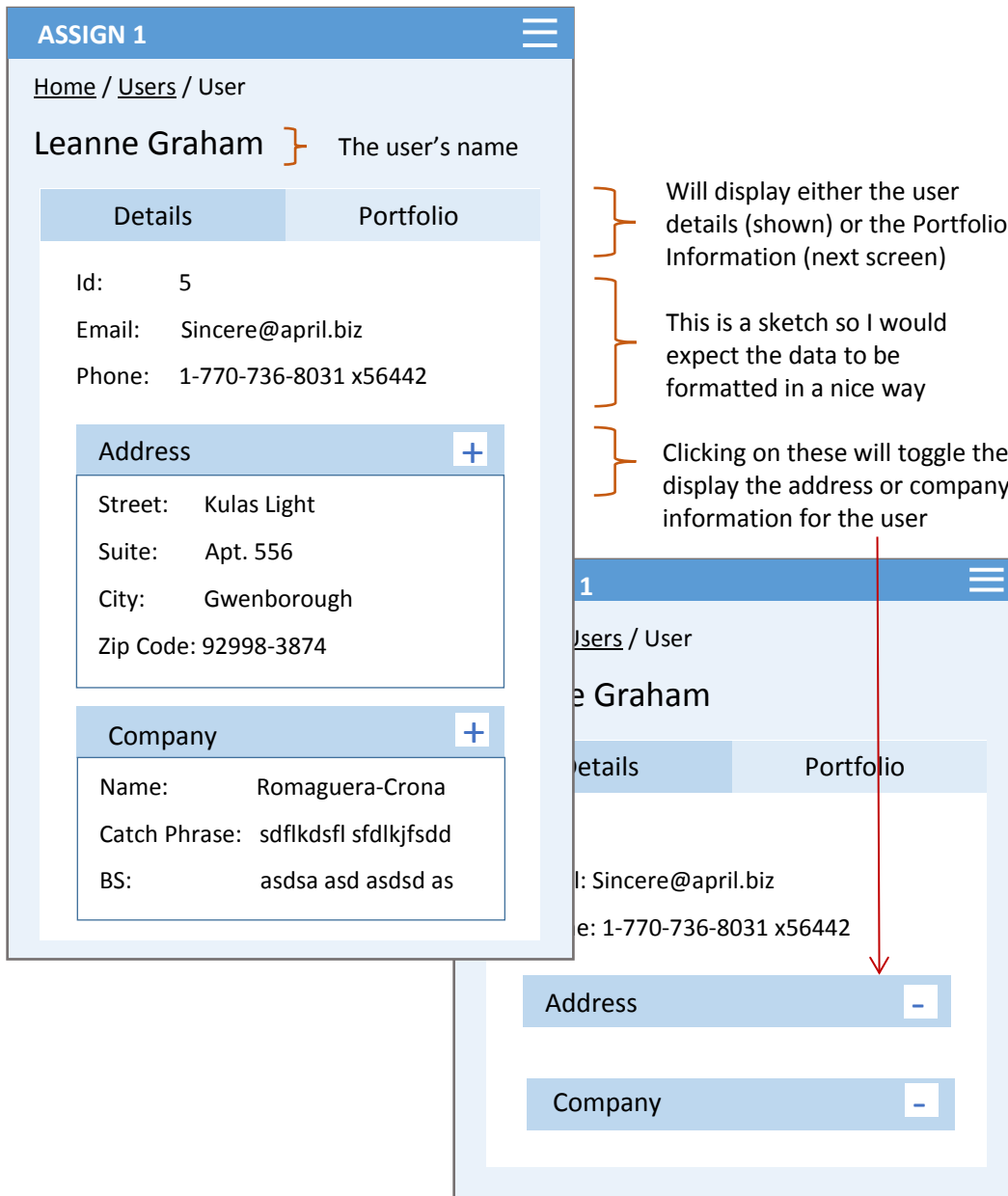
Display list of users sorted by name. Each should be a link to Single User view

8. When the user selects a single user, it should initially display the Single User View. This view has two subviews: Details View (shown here) and Portfolio View (shown next page), switchable via the Details/Portfolio tabs. Data for the detail view comes from the users web service.

The easiest way to manage the switch from details view to portfolio view, is to implement each subview as a separate component, and then use conditional rendering, for example:

```
let toDisplay = <ComponentOne attribute={variable} />
if (someCondition) toDisplay = <ComponentTwo attribute={variable} />
return (
  <div>
    ...
    { toDisplay }
  </div>
);
```

Single User – Detail View



9. Within the single user view, if the user clicks the Portfolio tab, it should display the user's stock portfolio. Data for this comes from two files: `stocks.json` and `userStockPortfolio.json` file. In SQL you would make use of an `INNER JOIN` and a `WHERE` to bring together information from two separate tables with a criteria filter. You are going to have to do the equivalent however in Javascript. You are going to need to implement this algorithm:
- use the array `filter()` function to reduce the array from `userStockPortfolio.json` to just those whose `userId` field matches the current user id.
 - create a new empty array
 - loop through the reduced array, and for each element do the following:
 - search the array from `stocks.json` for the matching stock
 - create a new object that contains the `symbol`, `amount`, and the stock name
 - push this object onto the empty array
 - sort this array
 - display this array

Single User – Portfolio View

ASSIGN 1

Home / Users / User

Leanne Graham

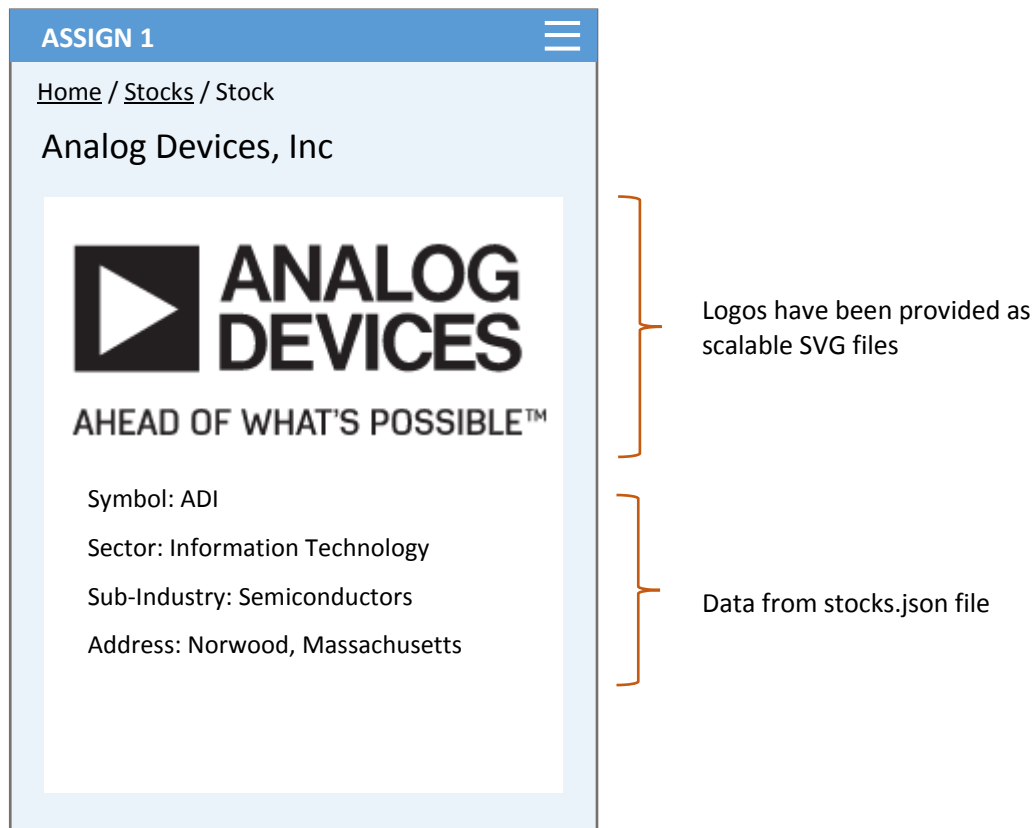
| Details | | Portfolio |
|----------------------|-------------------------------------|-----------|
| Symbol | Name | Amount |
| AMZN | Amazon.com Inc | 1000 |
| ADI | Analog Devices, Inc | 700 |
| AMAT | AMAT | 650 |
| CTXS | Citrix Systems | 500 |
| CB | Chubb Limited | 300 |

Clicking on these headings will change the sort order to the column clicked on. Clicking repeatedly on the column name will toggle between ascending and descending. Be sure to change the CSS cursor property of these to pointer.

Display list of stocks for that user, with the symbol and name being a link/route to single stock view (showing the appropriate stock). Some users have multiple entries for the same stock.

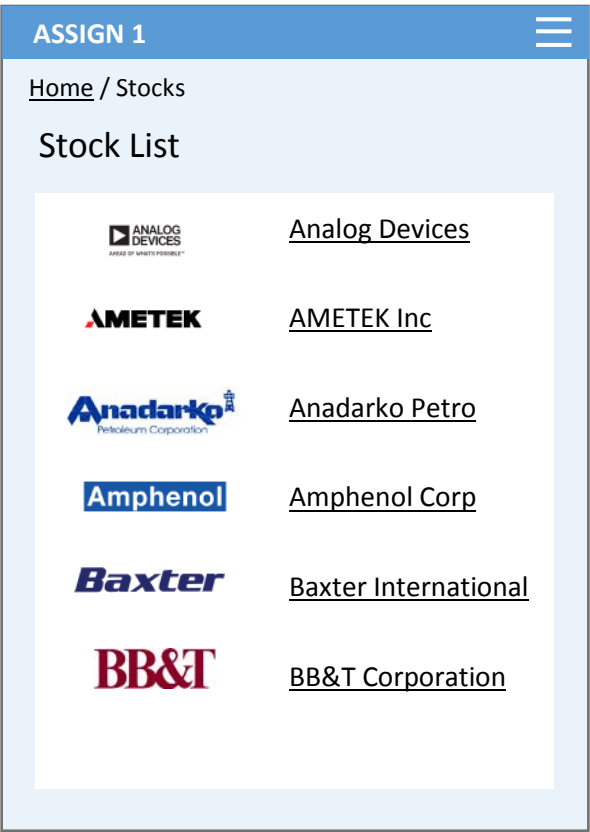
10. When the user clicks on a stock, it should display the Single Stock View. You will need to set the width and max-width properties via CSS for the logos.

Single Stock View



11. When the user selects a Stocks links (from home page, breadcrumb, or drop-down menu in upper-right corner), it should display the Stock List View.

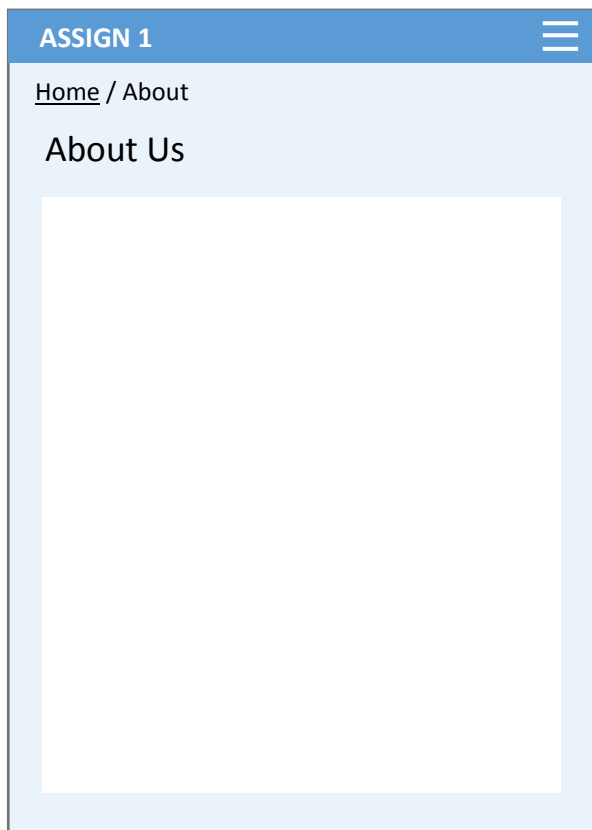
Stock List View



Display list of stocks sorted by name. The stock name should be link to Single Stock view.

12. When the user selects an About link, it should display the About Us View.

About Us View



Tell us who you are, what technologies were used and what tools and packages you have used. If used other sites for help, list them here with links.