

React Technical Challenge

For this technical challenge you will be answering some technical background questions to gauge your ability to communicate technical issues, and also developing a small application with React components to gauge code style, library use and experience with React and JavaScript in general.

Please put your source code in a public repo and provide us with the link to review. Create commits as appropriate as you work, avoiding a single commit when all work is completed. Answers to the background questions should be included as ANSWERS.txt in the root directory.

You are encouraged to use modules and libraries as appropriate. We are looking for simple, clean, elegant code, and appropriate testing.

Background Questions

Please explain what is wrong with this code and what the observed behaviour might be in a component that included it.

```
this.setState({count: this.state.count + 1})
```

Can you please explain how Redux works, assuming you were talking to a non-technical audience.

Technical Challenge

Please develop a single-screen application that allows a user to interactively filter a data set of users based on certain criteria, to narrow down the number of users. The app should present a set of interactive filters, a graph of matching users and a table of matching users. Changes to the filters should modify the contents of the graph and table in realtime.

Assume that your app would receive data in the following format:

```
export const Users = [  
  { id: '1', birthday: 3, spend: 1000, region: 'United States', gender: 'Female'},  
  { id: '2', birthday: 12, spend: 2000, region: 'APAC', gender: 'Female'},  
  { id: '3', birthday: 10, spend: 4000, region: 'Asia', gender: 'Female'},  
  { id: '4', birthday: 9, spend: 2109, region: 'Latin America', gender: 'Female'},  
  ...  
];
```

You should generate a test file that has at least 1000 rows with users in this format and include this as instance data on your page.

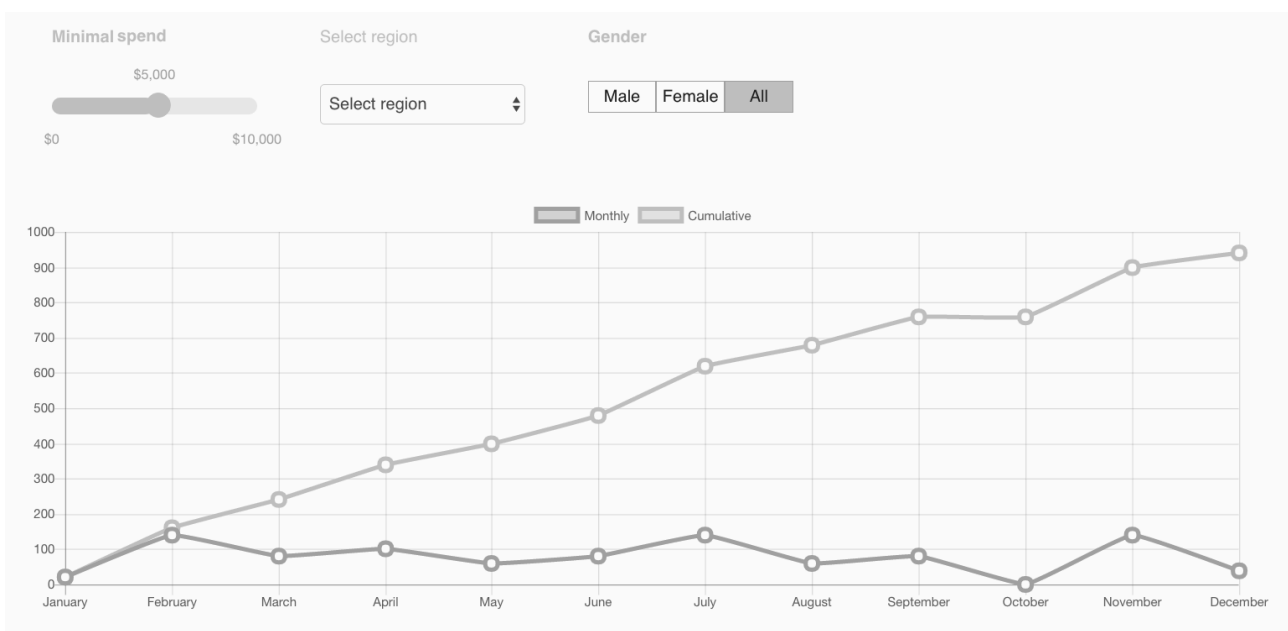
- id is unique and numeric sequential
- birthday is a number between 1 and 12 (indicating month of birth)
- spend is a number between 0 and 5000, indicating how much a customer has spent with the company in dollars
- region is an enum of 'United States', 'Europe', 'APAC', or 'Latin America'
- gender is 'Male', or 'Female'

The intention of this mini-application is to calculate the costs involved in giving a \$5 coupon to each of these customers over the course of a year, if they received it on their birthday.

Requirements

Please use the UI library Ant.design (“antd”) for the tables (<https://ant.design/components/table/>). Your choice of graphing library is up to you, but we can recommend chart.js as an option if you don't have a preference.

The below mockup shows an example of how this could look, with three interactive filters above (minimum spend, region and gender), and a graph below. A table is not shown but should be rendered below the graph. This visualisation is not prescriptive, so yours can look different as long as the logic is correct.



The graph should redraw itself dynamically based on the UI state, so that users with spend less than minimum spend are filtered from the graph and table as the slider is interacted with. Similarly a selection in the region dropdown should filter the data set to just those users. Gender can be filtered by either male, female or both (i.e., no filter applied).

The graph should plot the count of users that remain after the filters are applied, bucketed by their month of birth (the “monthly” line). A second line (labelled “cumulative” in the diagram above) should show the accumulated cost over the course of the year if every one

of those users was given (and redeemed) a \$5 discount. So if the “monthly” line was plotted at 10 and 12 for January and February, for example, the “cumulative” line would be plotted at \$50 ($10 \times \5) and \$110 ($22 \times \5).

Please include documentation on how to run this project from scratch and view in the browser, assuming that the target machine has *git* and *yarn* installed. You can build the react app statically or serve it from a local server as preferred.