SEE HOW EASY IT IS TO BUILD A QUICK AND USEFUL REACT APP

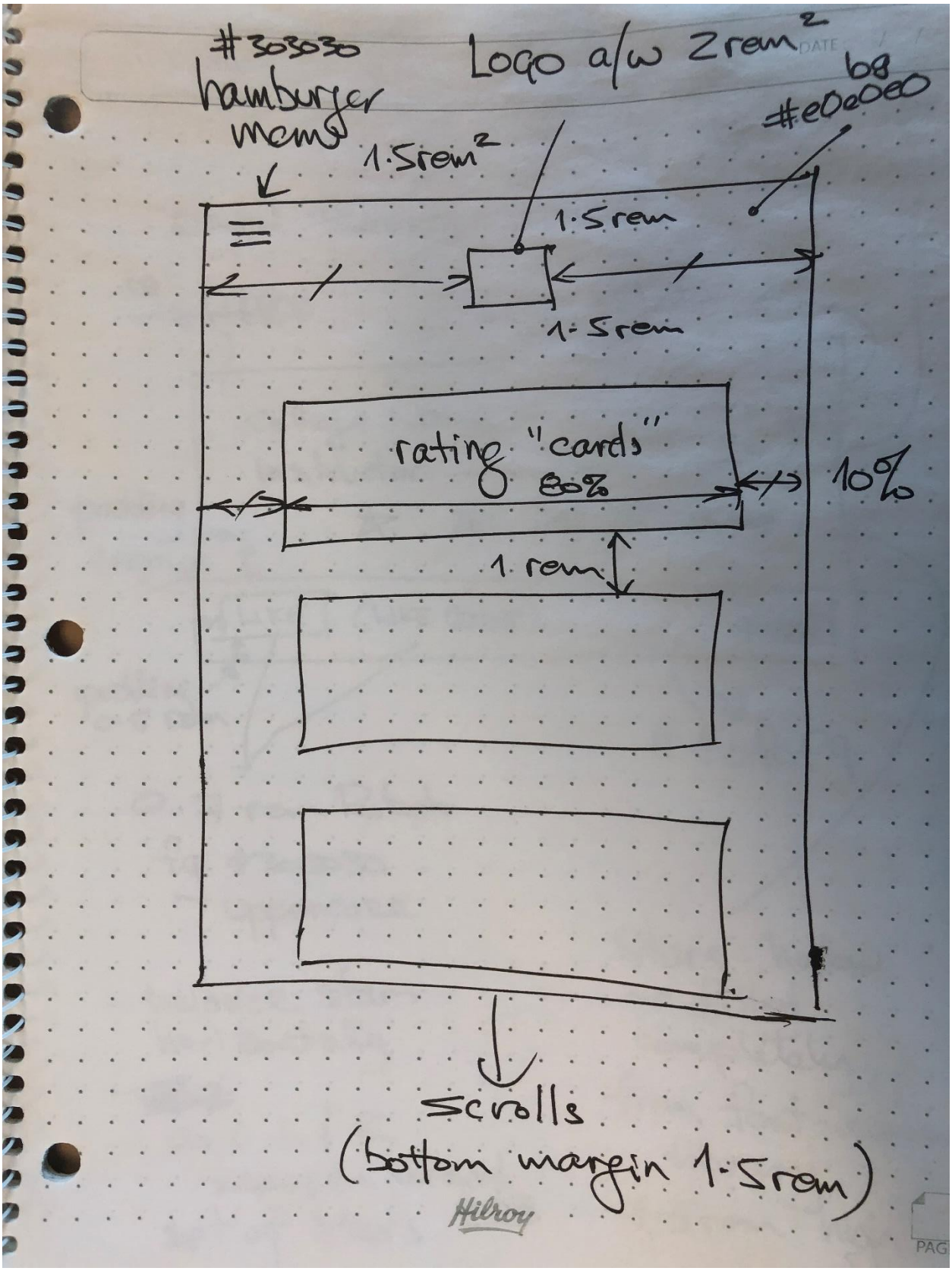
Worksheet: Build The Raterator (1)

In this worksheet, you will put together a simple React web application that allows you to rate instructors (hey, we all try to get five stars). We'll call this app the **Raterator**, because things that end in "ator" are usually either **interesting || scary** and sometimes **interesting && scary**. With this app you can view ratings, create new ratings, and (if you're an administrator) edit and delete ratings. Thus we are setting out to build a straightforward CRUD app. Our customer has indicated that they think displaying ratings is a higher priority than creating or updating ratings, so we'll just implement that functionality in this worksheet. We'll add additional capability later.

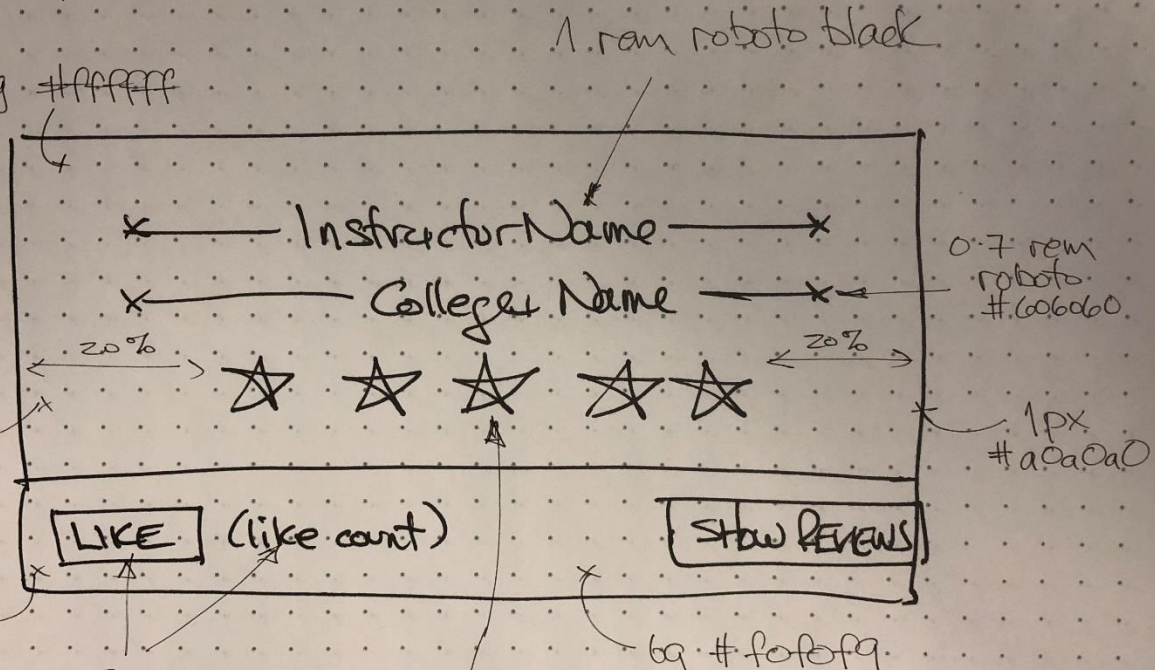
This app draws on your experience with JavaScript, React, and of course HTML/CSS. You should not use any CSS frameworks or jQuery in your solution. You are working against a mockup prepared by a designer. This means that most design decisions have been made, and your implementation should match them faithfully.

Your app should display a single "card" for each instructor. Each card should display the instructor's name, the college's name, and the arithmetic mean of all the ratings (i.e. the aggregate rating for that instructor) using the stars, rounded to the nearest whole star. Your app should also display the total number of "likes" received for the instructor (likes are independent from the ratings). A "like" button increments the number of likes for the instructor when clicked, but for now that information is only saved in the app's model.

A future iteration of this app will let users add new ratings, so you should consider that when designing your state model.


#303030
hamburger menu
Logo a/w 2rem²
1.5rem²
#e0e0e0
1.5rem
1.5rem
rating "cards"
80%
10%
1 rem
scrolls
(bottom margin 1.5rem)

RATING 'CARD'


1.rem roboto black
bg #ffffff
InstructorName
CollegeName
0.7 rem roboto #606060
20%
20%
1rem padding
1px #a0a0a0
LIKE
(like count)
SHOW REVIEWS
0.5rem padding
0.7 rem roboto uppercase #303030
bg #f0f0f9
Stars:
font-awesome library
1.5 rem
rgb = 152, 0, 0
centered

SAMPLE DATA

{"RateratorSchemaVersion":1,"Schools":[{"Name":"Calgary College Of The Arts","SchoolSid":"CCARTS"},{"Name":"Bow Ridge College","SchoolSid":"BOWRID"},{"Name":"Calgary Coding Bootcamp","SchoolSid":"CCBOOT"},{"Name":"Alberta Machine Learning Institute","SchoolSid":"AMLINS"},{"Name":"JavaScript Judo Academy","SchoolSid":"JJACAD"}],"Instructors":[{"Instructor":"Sidney Thompson","InstructorSid":"SIDTHO"},{"Instructor":"Stephanie Chan","InstructorSid":"STECHA"},{"Instructor":"Emma Jones","InstructorSid":"EMMJON"},{"Instructor":"Peter Van Gogh","InstructorSid":"PETVAN"},{"Instructor":"T.G. Best","InstructorSid":"TGBEST"},{"Instructor":"Sean Harrison","InstructorSid":"SEAHAR"},{"Instructor":"Riley Robertson","InstructorSid":"RILROB"}],"Likes":[{"InstructorSid":"SIDTHO","Likes":45},{"InstructorSid":"STECHA","Likes":90},{"InstructorSid":"EMMJON","Likes":102},{"InstructorSid":"PETVAN","Likes":85},{"InstructorSid":"TGBEST","Likes":385},{"InstructorSid":"SEAHAR","Likes":35},{"InstructorSid":"RILROB","Likes":225}],"Ratings":[{"InstructorSid":"SIDTHO","SchoolSid":"CCARTS","Rating":3.5,"Username":"codeninja","SubmittedOnUtc":"2016-03-01T07:00:00.000Z"},{"InstructorSid":"SIDTHO","SchoolSid":"CCARTS","Rating":2.5,"Username":"aiaiai","SubmittedOnUtc":"2018-02-14T07:00:00.000Z"},{"InstructorSid":"EMMJON","SchoolSid":"BOWRID","Rating":3.95,"Username":"igetwhatclosuresare","SubmittedOnUtc":"2018-06-12T06:00:00.000Z"},{"InstructorSid":"SIDTHO","SchoolSid":"CCARTS","Rating":2.1,"Username":"superdev","SubmittedOnUtc":"2018-05-16T06:00:00.000Z"},{"InstructorSid":"EMMJON","SchoolSid":"BOWRID","Rating":4.25,"Username":"superdev","SubmittedOnUtc":"2018-07-15T06:00:00.000Z"},{"InstructorSid":"PETVAN","SchoolSid":"CCARTS","Rating":3.1,"Username":"iotrocker","SubmittedOnUtc":"2018-05-02T06:00:00.000Z"},{"InstructorSid":"RILROB","SchoolSid":"BOWRID","Rating":5,"Username":"sharpc","SubmittedOnUtc":"2018-12-02T07:00:00.000Z"},{"InstructorSid":"PETVAN","SchoolSid":"CCARTS","Rating":3.8,"Username":"dbjedi","SubmittedOnUtc":"2018-11-12T07:00:00.000Z"},{"InstructorSid":"RILROB","SchoolSid":"BOWRID","Rating":5,"Username":"dreamsincode","SubmittedOnUtc":"2017-09-24T06:00:00.000Z"}]}

WHAT TO DO

Here's a checklist to help you complete this worksheet.

Study the designer's artwork. Most of the branding and style decisions have already been made. If something is unclear or seems missing, bring it forward. The app should render correctly and legibly on desktop landscape and mobile portrait down to 500px wide. The logo artwork will be added later, so just draw a placeholder box for now.

Study the sample data. Some sample instructor ratings are provided in a .json file. Your code will need to interpret this file and summarize it for presentation. The data contains a number of reference entities (schools, instructors) and instructor ratings and likes. Each rating represents a contribution by a single user. Each rating is made up from a college, an instructor, and an integer rating from 1 to 5. In addition, each instructor can have an integer "like" count, which is independent of ratings. You should not assume this data set has referential integrity.

Block out the components. Identify the major components that will make up this app and figure out what components are reusable.

Design a model. Figure out what "state" needs to be managed in the app and where it will be stored.

Build & test it! Implement the app using React. You won't need any libraries outside of Node and whatever CSS imports you feel are helpful. Don't use any CSS frameworks.

Submit it! When your app is completed, please zip up the project folder **excluding node_modules** and submit it using the answer sheet form as below. After completing the answer sheet, you'll get a zip file back containing your proof of submission.

Update D2L. Post the proof of submission zip file to **Worksheet Dropbox: Build The Raterator (1)**.

Worksheet: Build The Raterator (1)
Worksheet Code: BTRATR
Collaborators: Blank (this is an individual work effort)

Q1. Please upload the zip file containing your app's project folder.

Q2. Please explain the rationale behind your choice of how to manage state in this application.

Q3. Please identify any issues preventing this application from scaling up to a large number of users.

HOW TO GET TOP MARKS

A clean, well-reasoned set of React components is the design objective. You'll also need to demonstrate reasonable design decisions regarding how to manage state. And of course, you need to match the mockups and write code that is easy to follow and well documented.

ASSESSMENT CHECKLIST

Here is the grid we will use to assess your work.

App uses a set of React components that demonstrate reuse and low coupling	/30
App has a model (state) with appropriate structure	/30
App matches designer's mockup accurately and completely	/20
Code is easy to follow and well documented (hygiene)	/20
Total	/100
Plagiarized or unoriginal work (0% plus **academic sanction**)	
Total adjusted	/100