CS4HC3 / SE4HC3 – Human Computer Interfaces

Assignment #4 – React SPA with Websockets

Weight

5% of total course grade

Due date

Sunday November 1st at 11:59pm

Primary Learning Objectives

- Create a single page application with React and React Router.
- Use Websockets to build a real-time web application.

Requirements

Social media posts such as Tweets on Twitter can be analyzed during disasters (e.g. hurricanes, earthquakes) to better understand and inform disaster responses (e.g. see Rapid assessment of disaster damage using social media activity, along with many other Google Scholar search results). Machine learning can be used to analyze the content of social media posts in order to better assess damage (an example). In this assignment, we will create a dashboard for viewing and analyzing social media posts resulting from a disaster that have been categorized by an ML algorithm.

See the starter code on Avenue to Learn for this assignment. The server code can be run by navigating to the unzipped /server folder and running **npm install** followed by **node server.js** from the command-line. This will start a server that simulates a machine learning algorithm's categorization of social media posts following a disaster. The server will randomly generate social media posts, including a name, photo, and content, as well as a categorization of the problem (fire, flood, power outage, medical) and the priority (low, medium, high, critical). The server will send the data for each simulated social media post to the front-end as a JavaScript object via Websockets.

The starter code also includes a basic frontend that dumps the social media post data to the webpage. You can run the front end by navigating to the unzipped /frontend folder and running **npm install** followed by **npm start**. When both the front-end and server are running, the front-end should receive and display the social media post data. You should not modify the server code at all, but you should use the /frontend folder code as the starting point for your application.

The dashboard for viewing and analyzing the social media posts should include the following:

- It should have three pages: Home, Live Feed, Analytics
    - The user should be able to navigate between these pages using links in a navigation bar.
    - The navigation link of the currently active page should be highlighted.
    - Use React Router to implement this functionality.
- The dashboard should continuously receive social media posts from the server, and update accordingly.
- The Home page should contain some simple text explaining the dashboard and the two other pages, and that's all.
- The Live Feed page should contain a "live feed" of the social media posts.
    - The live feed should update automatically as more social media posts are received and include these posts in the live feed.
    - The live feed should either be a stylized list or table containing the social media posts, similar to a "News feed" on Facebook or a list of Tweets on Twitter.
    - The live feed page should have checkboxes that allow users to include and exclude social media posts from the feed based on the categorization of the problem and the priority level. There should be one checkbox for each problem type and one for each priority level, that when checked, will include social media posts of that problem type or priority level in the feed that is displayed, and when unchecked will exclude social media posts of that problem type or priority level in the feed that is displayed. The page should initially load with all 8 checkboxes checked and all social media posts displaying. If from this point a user were to uncheck say the "low" checkbox, than social media posts with the "low" priority level should not be displayed in the feed.
- The Analytics page should contain a table summarizing the total number of each problem type and priority level, for example:

	Fire	Flood	Power	Medical	Total
Low	2	1	4	2	9
Medium	3	2	3	1	9
High	4	5	1	5	15
Critical	2	2	3	3	10
Total	11	10	11	11	43

    - The table should contain the relevant totals for each cell... e.g. the total fire problems of low priority would be in the top-left corner, etc.
    - The table numbers should update live as more social media posts are received.

In the case of both the live feed page and the analytics page, you only need to include the data for the social media posts since the last time the page was loaded in the browser (like the starter code).

Style the application nicely, using either your own styles or external styles (e.g. Bootstrap, w3Schools, etc.). The navigation, feed of social media posts, and the analytics table should all look professional. It doesn't need to look nice on a mobile screen size, you can assume a desktop screen size.

Help

The .filter() method can definitely be useful for this assignment: https://www.w3schools.com/jsref/jsref_filter.asp.

Depending on how you solve the problem, you may need to pass the components rendered by React Router values via props. This is one way of doing so:

```
<Route path="/livefeed" exact render={(props) => <LiveFeed {...props}
    posts={this.state.posts} /> } />
```

And this is considered "the right way" of doing this (e.g. article). You could also use more advanced ways of managing state across components (e.g. Redux), but that's not expected.

Submission

Before submitting your work, delete the node_modules folder from your /frontend folder (this decreases the size of your upload to the dropbox, and helps the marker to mark quicker). Then zip your /frontend folder and submit to the dropbox on Avenue. The marker will mark your assignment by starting the server, and then running **npm install** and **npm start** in your unzipped /frontend folder.

Your submission must include the ***uncompiled*** source code in the /src folder. Use create-react-app for this application. You can install additional modules if you like (and you'll need to install React Router).

Marking scheme

Component	Description	Marks
Navigation	Does the navigation work as required? Is navigational highlighting used? Is React Router used to implement this feature?	10
Live feed page	Is the live feed displayed? Do the filtering options work as required?	40
Analytics page	Are the analytics displayed? Are they accurate?	30
Style and design	Is the application, in particular the navigation, live feed and analytics table, professional in appearance? Are these aspects well-designed?	20
	Total:	100