Stack Overflow Web App

Design:

For my implementation I used the React framework with JavaScript. Since it is a web application, I thought using JavaScript would be the easiest approach and I am already familiar with React, so it made sense to use it. Since there is only one page, I didn't use any routing and all the logic and layout is in App.js. There is however a separate file called getPostInfo.js that contains the two API calls I needed.

The first one is "questions", that I call twice for the top 10 newest and top 10 most voted, just with different parameters. The request is done using a library called axios. I don't know if axios is decompressing the response or if it was the browser I'm using but the response was decompressed. Just in case I used another library called decompress-response which I think is a React library, that will decompress a response in either GZIP or DEFLATE or just pass it through if it's already decompressed. The second API call is "questions{ids}" which just lets me get the answers with their body and comments using a certain type of filter which lets that through.

Both calls will trigger an alert to retry if they fail or backoff if the API is unavailable. Both have their own filter which I set to be the minimum data needed for the app. Both have a key stored in a .env file that lets the app make more requests per day. To get the key I registered my app with Stack Apps.

For the layout, when a tag is searched for it will show up to 20 responses in collapsibles. Clicking on one will show a number a divs. First is the question, followed by the comments underneath which are underlined and have their creation date and votes in bold. The next is the answers which are all contained in a border. The answers will also show their comments at the bottom if applicable.

Finally, response time will be shown at the bottom at the bottom of the screen. There is two response times because I don't fetch all question, answer and comment data when a tag is searched. Question Response Time shows the response time to load all questions and their related titles, bodies, creation dates, votes, and comments, which will be updated when a tag is searched for. Answer Response Time shows the response time to load all answers and their related titles, bodies, creation dates, votes, and comments, which will be updated when a question collapsible is clicked.

Docker:

To create my docker image I needed to add a Dockerfile and a .dockerignore to my application directory. From there I created an image using **docker build . -t so-questions-app**.

Link to docker hub image: https://hub.docker.com/r/harveyjames/so-questions-app

Tag version: v2

GitHub:

Username: Le-DarkOverlord

Repository won't have the .env file so it won't have the app key.

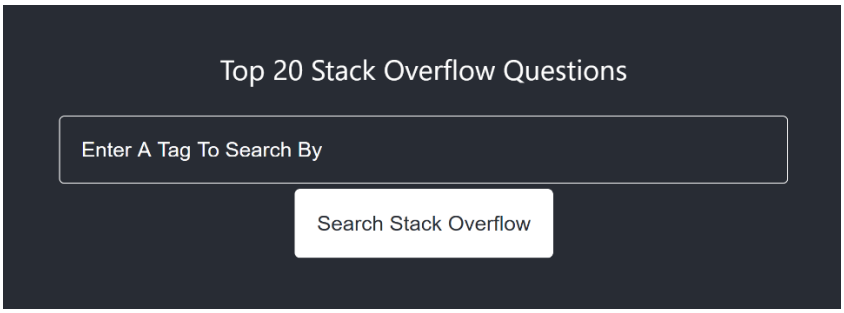
Link to GitHub repository: https://github.com/Le-DarkOverlord/SOquestionsApp

How to run:

To run, pull the image from docker using **docker pull harveyjames/so-questions-app:v2**. Then run the container using **docker run -dit -p 3000:80 harveyjames/so-questions-app:v2**.

To see running container navigate to **http://localhost:3000/.**

You will see a search box and a button. Enter in a single tag (ex. Java) and you will get back a list of 20 questions from the past week sorted by descending creation date. You can optionally enter multiple tag separated by a semicolon (ex. python;flask) which will also work but has not been extensively tested.



Click on the any of the collapsibles to see the question body, answers, and comments.

Answer #1
Tue Mar 09 2021 0:20
Votes: 1

Finally, view response time of web application at bottom of screen.