

Task 2: You are working on a dashboard using ReactJS and Tailwind CSS. The dashboard displays data charts which are fetched from a REST API. However, the charts are not updating in real-time as new data arrives from the API.

Question: How would you troubleshoot and resolve this issue ensuring the charts update in real-time as the data changes? Explain your approach, potential challenges and the overall thought process.

Solution: I would follow the following approach to troubleshoot and resolve the issue of real-time updates failure for the ReactJS and Tailwind CSS dashboard:

- 1) I would start with understanding how the data is fetched, and how it is stored till it is rendered.
- 2) Then I would check the current codebase and look for any faulty lifecycle/hooks implementation.
- 3) I would check how the data is fetched in realtime and consider using websocket or socket.io if there are not any existing mechanism in place. Furthermore, it can also be a periodical fetch call eg. `setInterval()`.
- 4) Then I would ensure the data is in sync by updating the states in our app. I would check browsers network tab and see how the states update compared to websocket messages.
- 5) Furthermore, I would also make sure the connections are properly closed upon unmount.

Example:

```
home > mosi > Repo > interview > react-task > src > App.jsx > default
1  import React, { useEffect, useState } from 'react';
2  import io from 'socket.io-client';
3
4  const Dashboard = () => {
5    const [data, setData] = useState([]);
6
7    useEffect(() => {
8      const socket = io('http://localhost:5000');
9      socket.on('new_data', newData => {
10        setData(currentData => [...currentData, newData]);
11      });
12      return () => socket.disconnect();
13    }, []);
14
15    return (
16      <div>
17        {data.map((datum, index) => (
18          <div key={index}>{datum.value}</div>
19        ))}
20      </div>
21    );
22  };
23
```