* Add api and get first 5 records from api data using using class based component and display data in a function based component.

App.js

import React, { Component } from "react";

import axios from "axios";

import ShowData from "./ShowData";

class App extends Component {

constructor() {

super();

this.state = {

post: [],

};

}

componentDidMount = () => {

console.log("componenty did mount");

axios

.get("https://jsonplaceholder.typicode.com/users/")

.then((res) => {

console.log(res.data);

this.setState({

post: res.data,

});

})

.catch((err) => {

console.log(err);

});

};

render() {

return (

<div>

<p>Question 1</p>

<ShowData post={this.state.post} />

</div>

);

}

}

export default App;

ShowData.js

import React from "react";

import ClipLoader from "react-spinners/ClipLoader";

const ShowData = (props) => {

const { post } = props;

const userList = post.length ? (

post.slice(0, 5).map((info) => {

return (

<React.Fragment key={info.id}>

<p>Name: {info.name}</p>

<p>Email: {info.email}</p>

<p>Phone number:{info.phone}</p>

</React.Fragment>

);

})

) : (

<p>

<ClipLoader

color={"purple"}

loading={true}

size={150}

aria-label="Loading Spinner"

data-testid="loader"

/>

</p>

);

return <>{userList}</>;

};

export default ShowData;

* Create a state (score). Take input from user of score. If it is greater than prev state then, update the state, otherwise not. (Show messages in pop-up/modal like points update). - (I need help with this question)

App.js

class App extends Component {

constructor() {

super();

this.state = {

post: [],

score: "",

};

}

handleChange = (e) => {

console.log(e);

console.log(e.target.value);

console.log(e.target.id);

if (e.target.id == "score") {

this.validateScore(e.target.value);

}

};

validateScore = (score) => {

this.setState({

score,

});

};

changepoints = (e) => {

e.preventDefault();

this.setState(this.state.score);

};

shouldComponentUpdate = () => {

return true;

};

getSnapshotBeforeUpdate(prevProps, prevState) {

if (this.state.score > prevState.score) {

document.getElementById("p1").innerHTML =

"Before the update, the score was " + prevState.score;

}

return null;

}

componentDidUpdate() {

document.getElementById("p2").innerHTML =

"The updated score is " + this.state.score;

}

render() {

return (

<div>

<p>Question 1</p>

<ShowData post={this.state.post} />

<br></br>

<p>Question 2</p>

<p id="p1"></p>

<p id="p2"></p>

<form>

<label>Score:</label>

<input

type="number"

id="score"

onChange={this.handleChange}

value={this.state.score}

/>

<button type="button" onClick={this.changepoints}>

Update

</button>

</form>

</div>

);

}

}

* Read more about: https://blog.logrocket.com/react-lifecycle-methods-tutorial-examples/#whatarereactlifecyclemethods