

How I made This?

Step1:How to Create a component that will fetch the given API? Here how it goes...

```
import React, { useState, useEffect } from "react";

export default function UserList({ visible }) {
  const [users, setUsers] = useState([]);
  const [loading, setLoading] = useState(true);

  useEffect(() => {
    fetch("https://reqres.in/api/users?page=1")
      .then((response) => response.json())
      .then((data) => {
        console.log(data);
        setUsers(data.data);
        setLoading(false);
      })
      .catch((error) => console.error(error));
  }, []);

  if (loading) {
    return (
      <div
        className="d-flex justify-content-center align-items-center"
        style={{ height: "100vh" }}
      >
        <div className="spinner-border text-bg-primary" role="status">
          Loading...
        </div>
      </div>
    );
  }

  return (
    <div className="container" style={{ display: visible ? "block" : "none" }}>
```

```

<h1>Users</h1>
<div className="row">
  {users.map((user) => (
    <div key={user.id} className="col-md-4">
      <div className="card mb-4 shadow-sm">
        <img
          src={user.avatar}
          alt={user.first_name + " " + user.last_name}
          className="card-img-top"
        />
        <div className="card-body">
          <h5 className="card-title">
            {user.first_name + " " + user.last_name}
          </h5>
          <p className="card-text">{user.email}</p>
        </div>
      </div>
    </div>
  ))}
</div>
</div>
);
}

```

Step1: creating a component that displays the API data

First I have to create a component by using React hooks to manage state and perform a side-effect.

The 'useState' hook is used to initialize state variables for 'users' and 'loading' is a Boolean variable

that indicates whether the API request is still being processed or not.

The 'useEffect' hook is used to initialize state variables for 'users' and 'loading' is a Boolean variables that indicates whether the API request is still being processed or not.

The 'useEffect' hook is used to make an API request to fetch user data from "https://reqres.in/api/users?page=1" when the component mounts. The 'useEffect' hook has a dependency array that is empty, which means that the Api response is then converted to JSON format, and the data is

extracted and stored in the 'users' state variable using the 'setUsers' function. The 'setLoading' function is also called to change the 'loading' state variable to 'false' once the API request is complete.

If 'loading' is 'true' then the component displays the message "Loading...." Otherwise, the 'users' array is iterated using the

Step1: Creating navbar:

```

import React from "react";

export default function Navbar({ toggleVisible }) {
  return (
    <nav className="navbar navbar-expand-lg navbar-dark bg-primary">
      <a
        class="navbar-brand text-black p-1 m-lg-1"
        href="https://biztyz.advertyzement.com/"
      >
        Advertyzment
      </a>
      <button
        type="button"
        className="navbar btn btn-primary bg-body-tertiary rounded-1"
        onClick={toggleVisible}
      >
        Users
      </button>
    </nav>
  );
}

```

Import React from “**react**”;

Export default function Navbar({toggleVisible}){

This line imports the React library and declares a functional component called Navbar.it also receives an object destructured ‘props’ with a single property ‘toggleVisible’.

Step 2:writing written Statement

```
<nav className="navbar navbar-expand-lg navbar-dark bg-primary">  
<a class="navbar-brand text-black p-1  
m-lg-1" href=https://biztyz.advertyzement.com/>
```

```
AdverTymment</a>
```

```
<button type="button"  
className="navbar btn btn-primary  
bg-body-teritary rounded-1  
onClick={toggleVisible}>Users</button></nav>);
```

Step 3: here I defined **JSX** elements to be rendered as a
'Navbar' component. it creates a 'nav' element with a bootstrap class

`'navbar navbar-expand-lg navbar-dark bg-primary'`.

It also includes an anchor `'a'` tag with a class `'navbar-brand text-black p-1 m-lg-1'` and href to

`'https://biztyz.advertyzment.com/'` displaying the text `'Advertyzment'`.

The `'Button'` element with class `'navbar btn btn-primary bg-body-teritary rounded-1'` displaying the text `'Advertyzment'`.

The `'button'` element with class `'navbar btn btn-primary bg-body-teritary rounded-1'` displays the text `'Users'` and has an `'onClick'` event that executes the `'toggleVisible'` function passed as a prop.

The 'useEffect' hook is used to make an API request to fetch user data from <https://reqres.in/api/users?page=1> when the component mounts. The 'useEffect' hook has a dependency array that is empty , which means the API request only made once when the component mounts.(because '[]' is passed as the dependency array, indicating the effect should only run once). The fetched data is logged to the console for debugging purposes, then the 'users' state variable is set to the 'data' property of the response object, and 'loading is set to 'false'


```
if (loading) {  
  return <div>Loading....</div>  
}
```

gave bootstrap animation and style called spinner. This checks if 'loading' is set to 'true'. If it is, then it displays a "Loading "...message while the data is being fetched. Finally, if 'loading' is 'false', it displays the fetched data as a list of cards. The 'visible' prop is used to conditionally show or hide the component based on the value of "Visible".

STAGE 2: Creating App.js the Parent Component and importing the child components.

```

import React, { useState } from "react";
import Navbar from "../components/Navbar";
import UserList from "../components/userCard";

function App() {
  const [visible, setVisible] = useState(false);

  const toggleVisible = () => {
    setVisible(!visible);
  };

  return (
    <div className="App container-fluid">
      <Navbar toggleVisible={toggleVisible} />
      <UserList visible={visible} />
    </div>
  );
}

export default App;

```

generally ‘App.js’ is a file that defines the main component of the React application. Think of it as the “boss” of the application. It controls what is displayed on the screen and how the application behaves.

2. In the above code, we’re defining a component called ‘App’. This component has two “children components” called ‘Navbar’ and

“UserList”.The “Navbar” is a component that displays a button labelled “Users”,and the “userList” is a component that displays a list of users.

3.Now comes the main phase of the code...to control when the ‘userList’ is a component that displayed on the screen, we need to keep track of whether it should be visible or not.That where the ‘useState’ hook comes in.It allows us to define a variable called ‘visible’, which means the ‘Userlist’ component wont be visible.

4.To make the 'UserList' component visible when the "Users" button is clicked, we need to define a function that changes the value of 'visible' state variable from 'false' to 'true', vice-versa. This function is called 'toggle-Visible'.

5.We pass the 'toggleVisible' function down to the 'Navbar' component as a 'prop', which is just a way to pass data from parent components to child component. The "Navbar" component can then call the 'toggleVisible' function when the "Users" button is clicked, and this will toggle the value of the 'visible' state variable.

Finally, we pass the 'visible' state variable down to the 'userList' component as another prop. This allows the 'UserList' component to check whether it should be visible or not, and to display itself accordingly.

I hope that helps! Let me know if you have any more questions.

Contact: gorrepatisameer@gmail.com