

Episode 6 : EXPLORE THE WORLD

part 3:

Now we have used the live swiggy api now we can remove the data.js (mock data file).

Part 4:

so now we have added the api and we are using the data received from the api to render the cards(when use effect is rendered)

load->render->api->render

Generally when page loads the api takes some time to fetch data meantime the data is not rendered in the app.

So we can add loader ,if api data is not fetched we show loader but if data is fetched the page will be rendered.

But using loader is not a good idea , instead we can use shimmer UI

What is Shimmer UI?

If you have ever used a web or mobile app that takes some time to load data from a server, you might have seen a loading spinner or a progress bar that indicates that something is happening. While these are common ways to show loading states, they are not very engaging or informative for the user. They don't tell the user what kind of content is being loaded, how long it will take, or what to expect next.

A better way to show loading states is to use a shimmer UI. A shimmer UI is a version of the UI that doesn't contain actual content, but instead mimics the layout and shapes of the content that will eventually appear. It uses a gradient animation that creates a shimmering effect over the placeholders, giving the user a sense of motion and progress.

Part 5: (Login and Logout button)

In this episode we are going to add the feature login logout ,if we click on login the button changes to logout ,if you click on logout it becomes login.

We use state variable (useState)to add this feature

Part 6 : (Search functionality)

Whenever you type text in input field of search bar the component rerenders,it is because the input field is tied with the state variable ,even it is re rendering the whole component it is just updating the input box in the component as it just updates the differences(with the help of reconciliation)

Part 07 : Making search functionality to work for all the data