

# Let's Explore the World

## Monolith and Microservice Architecture :-

### Monolith :-

A monolithic Architecture is a traditional software development model that uses one code base to perform multiple business functions. All the software components in a monolithic system are independent due to the data exchange mechanism within the system. It's restrictive and time consuming to modify monolithic architecture as small changes impact large areas of the code base. ~~In contrast to microservices~~

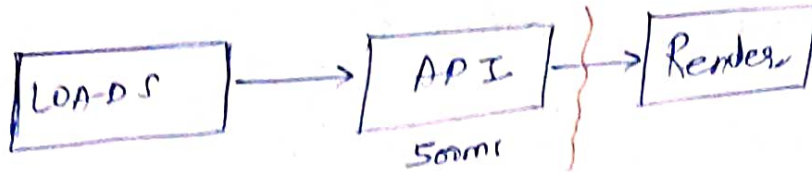
### Microservices :-

It's an architectural approach that composes software into small independent components or services. Each service performs a single function and communicates with other services through a well-defined interface. Because they run independently, you can update, modify, deploy or scale each service as required.

Diagram

## How web API connects with Backend Data :-

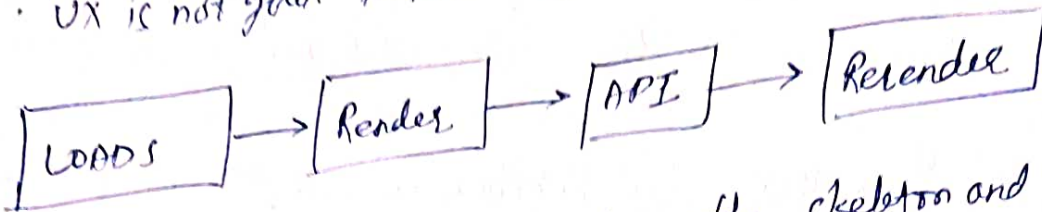
①



As soon as page loads we make the API call then renders all the stuff.

• UX is not good. Website is frozen till 500ms.

②



As soon as page loads, we render the skeleton and then made a API call then re-renders all the stuff acc. to API.

loads with shimmer UI

\* It has better UX User Experience

\* In React we always follow 2nd Approach.

\* use Effect  $(() \Rightarrow \{ \}, [])$   
                  ↑                   ↓  
          call back fn   dependancy Array.

Use Effect function called After our component renders.

\* First body will renders then our use Effect will be called.

\* After making API Call we will update the list of restaurants to dynamic data. So that the fetched data renders.

\* Initially list of restaurant should be an empty array.

const [listOfRestaurants, setListOfRestaurants] = useState([]);  
↑  
empty

## Shimmer UI

\* We show fake cards, until we get the data.

\* Shimmer loading is a cool animation trick often used when a web page or app is loading. It holds a part of the data in place with a moving animation until everything is loaded.

Advantages:

\* Enhances user Experience

\* Boost Performance

\* Most Importantly: When users see a loading screen with shimmer, they probably think it's their internet being slow not the product.

"There is no need to load the entire page all at once".

We will use this when our api is not called.

```
if (listOfRestaurants.length === 0) {  
  return <Shimmer/>;  
}
```

→ It is condition Rendering.



{Core's plugin to bypass the core issue}

Whenever state variable changes, React re-renders Component.

Created Toggle functionality using Ternary Operator.

```
const [btnNameReact, setBtnNameReact] = useState("login");
```

```
<button className = "login-logout-btn"
```

~~onClick~~

```
onClick = { () => {
```

```
  btnNameReact === "login"
```

```
  ? setBtnNameReact("logout")
```

```
  : setBtnNameReact("login");
```

```
}}>
```

Whenever state variable update, react triggers reconciliation cycle (re-renders the component).

Created A Search box using Input and bind to local state variable but then it is not taking input after binding it to local state variable then we have added "onChange" event handler.

Now after that applied filtered ~~to~~ condition or event handler to button.

Now solved One bug that Search text is case sensitive by using .toLowerCase() in both input getting and search text.

After that we have created of ~~API~~ local state variable list of restaurants  
to ~~use~~ protect original because it is modifying original data  
when we are searching.

e.g. we search burger  $\Rightarrow$  2 results come  
then again we search cafe  $\Rightarrow$  No results come  
because it is now searching from ~~the~~ these two.

### Cores plugin Issue Resolved

paste the url of corsproxy.io before Api call.

<https://corsproxy.io/>