



Episode 4 - Talk is Cheap, Show me the code

Project → Food Ordering App

Planning

- Exactly know what to build
- User Interface

Name of your Project → Romato

Low Level Design

/*

Header

- Logo
- Nav Items

Body

- Search
- Restaurant Container
 - Restaurant Card

Footer

- Copyright
- Route Links
- Social Links
- Address

(Add more components as you like)

*/

Props

Short form for Properties
They are used to dynamically send data to a component

Passing a prop to a function is like passing an argument to a function

Eg:-

```
<RestaurantCard  
  Props { resName = "Meghana Foods"  
         cuisine = "Biryani, North Indian"  
    } />
```

Passing props to component

Config Driven UI

It is a user interface that is built and configured using a declarative configuration file or data structure, rather than being hardcoded.

Config is the data coming from the api which keeps on changing according to different factors like user, location etc.

To add something after each element of an array

Eg If you want to add (,) after every element

```
resData.data.cuisines.join(",")
```

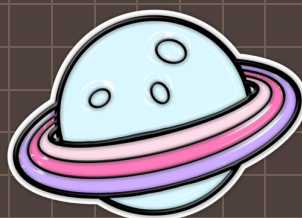
Good Practices

1) Destructure Props:
↳ Optional Chaining

Eg:

```
const { name, avgRating } = resData?.data
```

destructuring optional chaining



2) Repeating Ourselves (using a component again & again)

Instead use Dynamic Component Listing using JS map() function

3) Unique Key id while using map

Each item in the list must be uniquely identified

Why?

When we have components at same level and if new component comes on the first place (without id of any component)

DOM is going to re-render all the components again. As DOM can't identify where to put it

But if we give each of them a unique id then react knows where to put that component according to id

It is a good optimization & performance thing

** Never use index as keys in map. It is not recommended.

