

Ep-04 Talk is cheap, show me the code

apsara
Date: _____

★ Building Real world project
Food Ordering App.

→ Mock Designing

I Planning - "If your planning is good, code is very easy to write".

To build an app we need good plan.

Important :

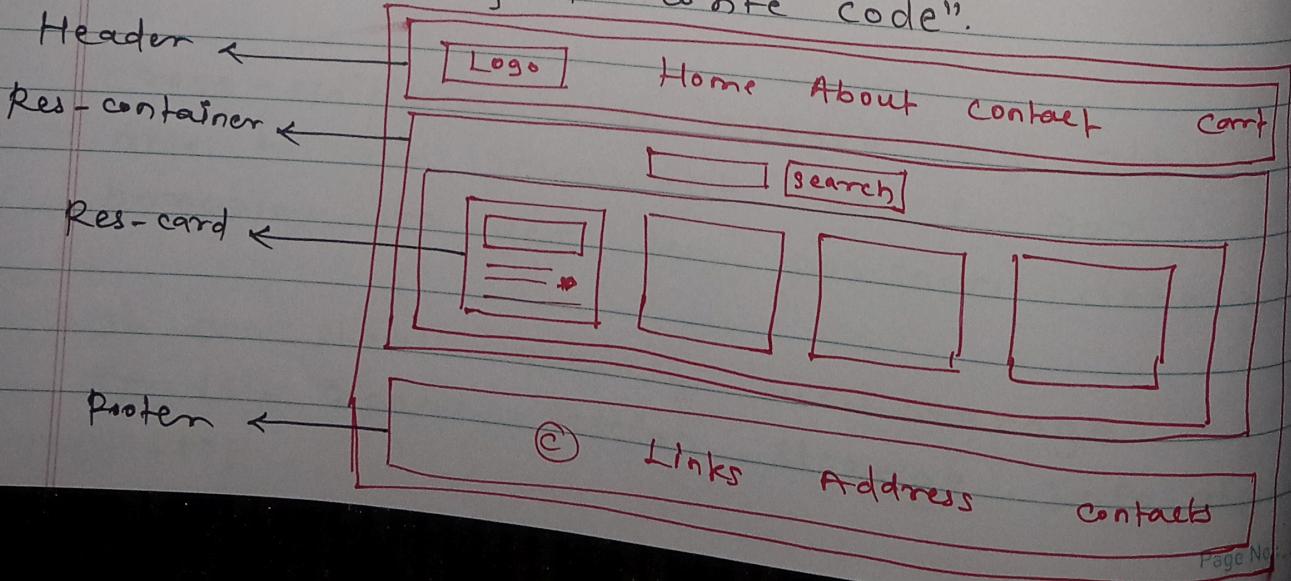
You should know exactly what to build and process to build it

2] Build UI mock (visual representation of our app)

e.g :

Name of app : Namaste Food.

Mock UI : "Once we have a visual picture in our head, that where things will come up it will be very easy to write code".



3] Low Level Planning -

e.g components we need to create, basic structure of app

e.g: 1) Header

- logo
- nav items

2) Body

- Search
- ResContainer
- ResCard

3) Footer

- copyright
- Links
- Address
- contacts

Build App.

1] App Layout Component

- Where all components of an app reside.

e.g const AppLayout = () => {

return (

 <div className = "App">

 <Header/>

 <Body/>

 <Footer/>

 </div>

)};

2] Header Component

```
Const Header = () => {
  return (
    <div className="header">
      <div className="logo-container">
        <img src="" className="logo" />
      </div>
      <div className="Nav-Items">
        <ul>
          <li> Home </li>
          <li> About </li>
          <li> Cart </li>
        </ul>
      </div> </div>
    );
};
```

3] Body Component

```
Const Body = () => {
  return (
    <div className="body">
      <div className="search">
        <input placeholder="Search" />
        <button> Search </button>
      </div>
      <div className="res-container">
        <RestaurantCard />
      </div>
    );
};
```

4) RestaurantCard Component

"There is multiple time one component need to render , make it as separate components".

It is Reusable component.

- Inline and Internal CSS in JSX
- style attribute in HTML and JSX is different
- In JSX, we can write inline CSS to tags using style attribute
- But this style attribute takes JS object

e.g

```
const styleCard = {  
  backgroundColor: "red"  
}
```

e.g

```
<div className=  
      "card"  
      style={styleCard}  
      >
```

```
  </div>
```

```
<div className="card"  
      style={styleCard}>  
  </div>
```

Internal CSS

Inline CSS

Date: _____

```

const RestaurantCard = () => {
  return (
    <div className="res-card">
      
      <h3> Meghana Food </h3>
      <h4> Biryani, North Indian </h4>
      <h4> 4.5 Stars </h4>
      <h4> 20 min </h4>
    </div>
  );
};

```

• Making RestaurantCard dynamic

1] props -

- short form of properties
- props pass to the components.
- If we want to dynamically pass data to components, pass it as props.
- <RestaurantCard /> → React functional component → JS function
 props → Normal argument to function

e.g <RestaurentCard restName = "KFC" cuisine = "Burger, Fast Food" />

<RestaurentCard restName = "Meghana Food" cuisine = "Biryani" />

Here restName & Cuisine are props pass to the <RestaurentCard /> component.

- Now react will take all this properties, and it will wrap it inside object and pass to the functional component as argument named as props.

```
const RestaurentCard = (props) => {
  return (
    <div className = "res-card" >
      <img src = "" />
      <h3> </h3>
      <h4> </h4>
    )
};
```

Argument
named as
props

If console.log(props); // print object

- Make card dynamic using props / Read props.

props are JS object

To use it in JSX, we write JS in {}.

Date: _____
apsara

```
const RestaurantCard = (props) => {
  e.g <div className = "res-card"
    <img />
    <h3> { props.restaurant } </h3>
    <h4> { props.cuisine } </h4>
  </div>
};
```

- When we have to pass dynamically passing some data to a component you pass in as props.

- Destructuring of props

1] const ResCard = ({ resName, cuisine })
 => {
 return (
);

destructuring of an object

2] const ResCard = (props) => {

const { resName, cuisine } = props;

destructuring of props

```
return (  
);
```

→ Another way of passing data dynamically

- From backend data will come in form of JSON i.e api

→ Config driven UI (Imp concept)

- website driven by data (config)
- we have written the UI once and now data is coming from backend my ^{UI} data is changed.
- UI Layer & data layer both are building frontend application.

"A good frontend engineer is good UI as well as data layer engineer".

eg const ResObj = {

ResName: "KFC",
cuisines: ["Fast Food", "Burger"],
avgRating: 4.5,

};

<div className = "res-container">

<RestaurantCard resData = {resObj} />
passing props resObj
as props

const RestaurantCard = (props) => {

const {resData} = props; → destructuring

return (

<div className = "res-card">

```

<img />
<h3> {resData.info.name } </h3>
</div>
);

```

- In real world, the data is not a single object.

The Data will in the form of array of an object.

and each object is an restaurant

- "when you build production ready app, you have to write reusable component"

→ optimising code

We should know best practices to write code

③ By destructuring

```

eg const ResCard = (props) => {
  return const { resData } = props;
  const { name, avgRating, cuisines,
  costForTwo } = resData?.Data;
  return (
    <div>
      <h3> {name} </h3>
      <h4> {avgRating} </h4>
      <h4> {cuisines} </h4>
    </div> );

```

2] Using map function

- Render one component multiple time we use map function.

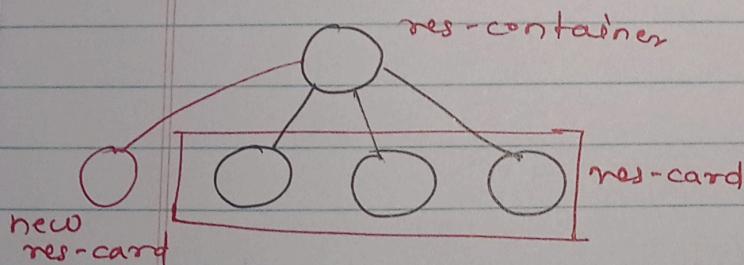
e.g ↴ `resList.map((restaurant) => (`
`<RestaurantCard resData = {restaurant} >)`
`)`

↪ Warning: Each child in a list should have unique 'key' prop.

- whenever we are looping an any list, we have to give a key for every list children.

`<Restaurant Card key = {restaurant.info.id} >`
`/>`

HTML DOM



- 'res-card' is same level component.
- They need to have unique id
- They should be uniquely represented

- suppose there are no id to 'res-card' and 'new res-card' came in.
- React will rerender all this 'res-card' Because react doesn't know which res-card is new.

- React don't know on which it should need to place.
- If we give to them unique id, react will render only that one new 'res-card' instead of 're-rending' all of 'res-cards'.

Note - we can use 'index' as key also.
But React itself says "never use index as an key".