

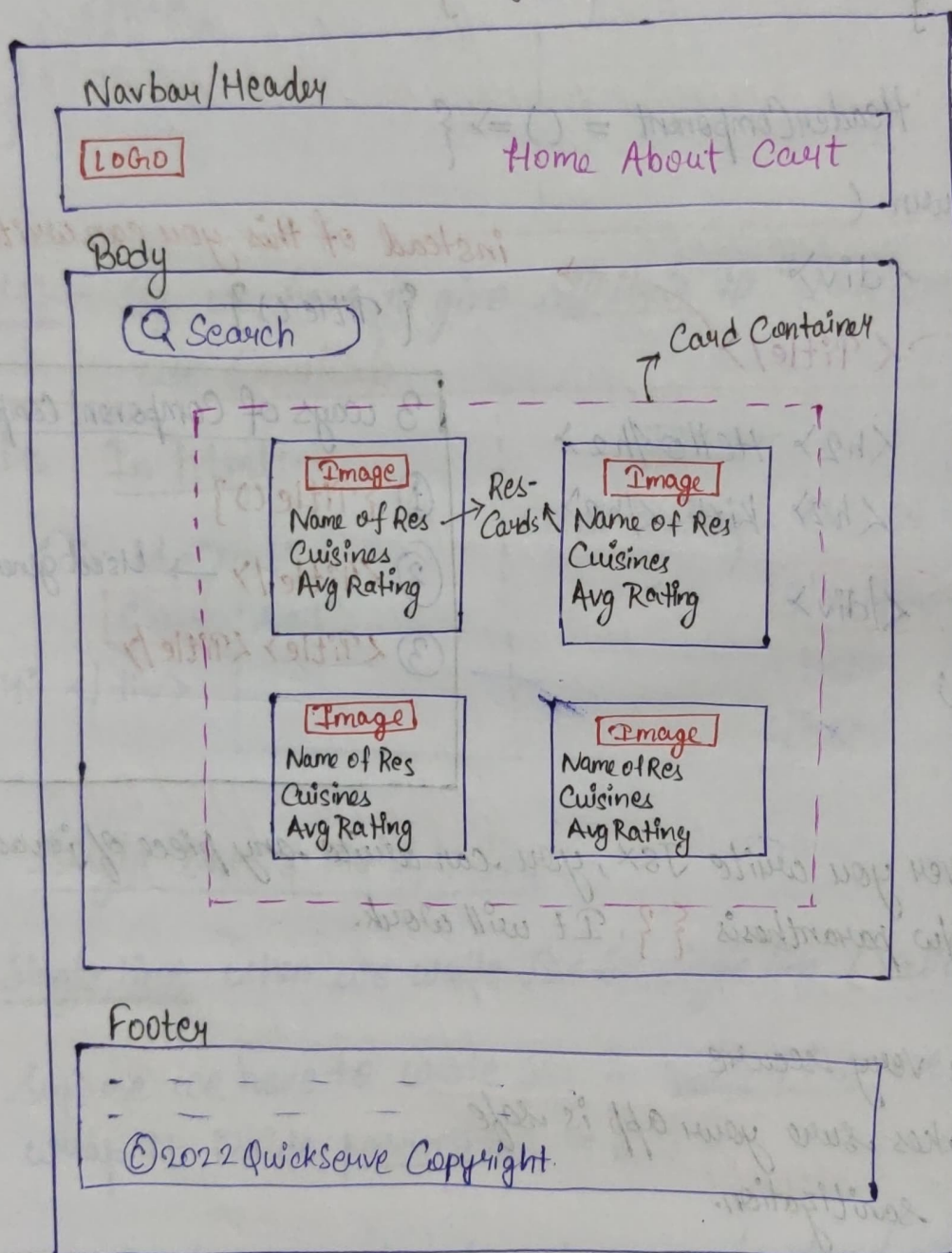
EPISODE-4 : TALK IS CHEAP, SHOW ME THE CODE

"Building a food ordering App"

Name: Quick Serve

→ Whenever you are writing code, do planning "our app look like this" ⇒

QuickServe



Step 2 - Think About Components that App will have.

- ↳ ① Header
- logo
 - Nav items

- ↳ ② Body
- Search Box
 - Restaurant Container
 - Restaurant Cards
 - Image
 - Name of Res
 - Star Rating
 - Cusines

- ↳ ③ Footer
- Copyright
 - Links
 - Address
 - ~~Contact~~ Contact

These all components comes under

```
const AppLayout = () => {  
  return(  
    - Header  
    - Body  
    - Footer  
  )  
}
```

→ Giving Style inside React

To give inline styles in react, do:-

- first method ⇒

```
const styleObj = {  
  backgroundColor: "red",  
};
```

This is normal JS object

* Const jsx = (

<div style = { styleObj } >

<h1> Hello World </h1>

</div>

);

Inside this parenthesis,
you can write any piece
of JS code.

→ In react, style is given using Javascript objects.

Alternative way :-

* Const jsx = (

<div style = { {

backgroundColor: "yellow";

}} >

<h1> Hello World </h1>

);

• SECOND METHOD ⇒

(or whatever tag)

Give class name to the div, and write CSS inside CSS file.

Const jsx = (

<div className = "jsx" >

<h1> Hello World </h1>

</div>);

* CSS file —

jsx {

backgroundColor: "blue";

}

• THIRD METHOD ⇒

Using external library like 'Tailwind css', 'Bootstrap', 'MATERIAL UI', etc.

Que:- Can I use a 'React Fragment' inside my 'React Fragment'?

Ans: Yes, you can nest 'React Fragment' components inside other 'React Fragment' components.

E.g:- import React from 'react';

```
const jsx = (
```

```
<>
```

```
<Child A />
```

```
<>
```

```
<Child B />
```

```
<Child C />
```

```
</>
```

```
<Child D />
```

```
</>
```

```
);
```

2

Que: What is React Fragment?

→ It is a component which is exported by 'React'.

→ Grouped a list of children without adding extra nodes to the DOM.

E.g:-

```
const AppLayout = () => {  
  return (  
    <React.Fragment>  
      <Header />  
      <Footer />  
    </React.Fragment>  
  );  
}
```

* Shorthand syntax is

```
<> </>
```

for <React.Fragment>
</React.Fragment>

Body Component

While building restaurant cards, we need some data for this card. Ways:—

- Using hard coded data
- Integrate with Api.

→ But in real world we can not pass "Hardcoded data" because:

- Data does not change i.e. Static & doesn't change automatically
- Difficult to scale & maintain
- Requires manual updates, increasing error risk

Dynamic data.

- Adapts in real-time with external sources like APIs or DB.
- Enables personalized, user-specific content.
- Easier to maintain & more flexible for large-scale applⁿ.