4. Talk is cheap , show me the code !

Components	**classNames**
Header	Header
Logo	logo-container
Nav items (menu items)	nav-items
Body	
Search	
Restaurant Container	
RestaurantCard	
Img	
Name of the res	
Star rating	
Cuisine	
Delivery time	
Footer	
Copyright	
Links	
Address	
Contact	

Attached External css file <link rel="stylesheet" href="index.css">

AppLayout - BigComponent - all component inside that

```
const AppLayout = () =>
{
  return(
        <div className="app">

          // Header
          // Body
          // Footer

        </div>
  )
}
```

Basic structure

Header

App .js

```
<div className="header">
    <div className="logo-container">

        <img className="logo"
        src="https://www.logodesign.net/logo/smoking-burger-with-lettuce-3624ld.png?nwm=1&nws=1&indust
    </div>

    <div className="nav-items">


        <ul>
            <li>Home</li>
            <li>About Us</li>
            <li>Contact US</li>
            <li>Cart</li>
        </ul>

    </div>

</div>
```

Index.css

```
.header
{
    display: flex;
    justify-content: space-between;
}
.logo
{
    width: 200px;
}

.nav-items  ul{

    display: flex;
     margin: 20px;
}

.nav-items ul li{
    padding: 20px;
    margin: 20px;
    list-style-type: none;
}
```

Content display in vertical - **Display : flex**
Move menu to right side **- justify-content : space-between**

Body

Body.js

```
const Body =() =>
{
    return(
            <div className="Body">
                <div className="search">Search</div>
                <div className="res-container"></div>
            </div>

    )
}
```

RestaurantCard (component)

For RestaurantCard we will create new component because we will reuse it
That data we will use again and again always create separate component for that

Inline CSS

```
const RestaurantCard = () =>
{
  return(
          <div className="res-card" style={styleCard} >
            <h3>Meghana foods</h3>

          </div>
  )
}
```

```
const styleCard =
{
        backgroundColor:"#f0f0f0",
};
```

Here styleCard is Javascript Object
Instead of styleCard we can direct write css code in style= {{}}
Here first {} - telling JS code

Second {}- Js Object

```
<div className="res-card" style={{ backgroundColor:"#f0f0f0"}} >
  <h3>Meghana foods</h3>

</div>
```

Creating 1 Restaurant Card

```
const RestaurantCard =() =>
{
    return (

        <div className="res-card">
            <img className="res-logo" src="https://media-assets.swiggy.com/swiggy/image/upload/fl_loss
            <h3>Silly Chilly</h3>
            <h4>Chinese, Thane , Maharashtra </h4>
            <h4>4.3 Stars</h4>
            <h4>33 minutes</h4>
        </div>

    )
}
```

Calling inside body

```
const Body =() =>
{
    return(
            <div className="Body">
                <div className="search">Search</div>
                <div className="res-container">
                <RestaurantCard/>
                </div>
            </div>

    )
}
```

Index .css

```css
.search
{
    padding:10px;

}
.res-card
{
    width: 200px;
    height: 350px;
    padding: 5px;
    background-color: #f0f0f0;
}

.res-card:hover
{
    border: 1px solid black;
    cursor: pointer;
}
.res-logo{
    width: 100%;
   height: 170px;
```

Creating multiple cards

App.js

```jsx
const Body =() =>
{
    return(
            <div className="Body">
                <div className="search">Search</div>
                <div className="res-container">
                <RestaurantCard/>
                <RestaurantCard/>
                <RestaurantCard/>
                <RestaurantCard/>
                <RestaurantCard/>
                <RestaurantCard/>
                <RestaurantCard/>
                <RestaurantCard/>
                <RestaurantCard/>
                <RestaurantCard/>
                </div>
            </div>
```

Index.css

```css
.res-container
{
    display: flex;
    flex-wrap: wrap;
}
.res-card
{
    width: 200px;
    height: 350px;
    padding: 5px;
    background-color: #f0f0f0;
    margin: 5px;
}
```

How to make this cards dynamic

Props - properties

Property is something which we can pass to component

Suppose we want to dynamically pass data to some component so we can pass through props

Like
Component = normal js function similarly
Props argument to js function

When we have to dynamically passing some data to component we pass

```jsx
const Body =() =>
{
    return(
            <div className="Body">
                <div className="search">Search</div>
                <div className="res-container">
                <RestaurantCard  resName="Silly Chilly" cuisine="Chinese, Thane , Maharashtra" />
                <RestaurantCard resName="Punjab Sind Premium Dairy" cuisine="Sweets, Desserts Thane"/

                </div>
            </div>

    )
}
```

```
const RestaurantCard =(props) =>
{
    return (

        <div className="res-card">
            <img className="res-logo" src="https://media-assets.swiggy.com/swiggy/image/
            <h3>{props.resName}</h3>
            <h4>{props.cuisine} </h4>
            <h4>4.3 Stars</h4>
            <h4>33 minutes</h4>
        </div>

    )
}
```

Instead of this we can use like this props

Destructuring ::

```
const RestaurantCard =({resName, cuisine}) =>
{
    return (

        <div className="res-card">
            <img className="res-logo" src="https://media-assets.swiggy.com/swiggy/image/uplo
            <h3>{resName}</h3>
            <h4>{cuisine} </h4>
            <h4>4.3 Stars</h4>
            <h4>33 minutes</h4>
        </div>

    )
}
```

We can write like this also this is also same thing

```
const RestaurantCard =(props) =>
{
    const { resName, cuisine} = props;
    return (

        <div className="res-card">
            <img className="res-logo" src="https://media-assets.swiggy.com/swiggy/image/upload/fl_lossy,f
            <h3>{resName}</h3>
            <h4>{cuisine} </h4>
            <h4>4.3 Stars</h4>
            <h4>33 minutes</h4>
        </div>

    )
}
```

How to data come from backend to us
It will come form of JSON

Swiggy api -confect driven UI

config driven UI
Controlling our UI , how the UI looks like using data , using config
Api data is config
Data can be diff for delhi
Data can be diff for mumbai
Data can be diff for pune

We have written the UI once and now according to data which is coming from backend our UI is getting change this is known as config driven UI

Any React application = UI Layer = 1. UI
2. Data

Eg . Suppose if we want to show red color background in delhi and blue color background in mumbai and green color background in kolkata we can do that too we just have to send color as config and config as drive our UI

Taking one restaurant card of Live data of Swiggy

Copy paste one res card data in

```
const resObj= ···
  }
```

Here cuisines are array
How to join **, separated** values

Before

McDonald's

BurgersBeveragesCafeDesserts

4.3 Stars

₹400 for two

22 Minutes

After

McDonald's

Burgers, Beverages, Cafe, Desserts

4.3 Stars

₹400 for two

22 Minutes

```
const RestaurantCard =(props) =>
{
    const {resData} = props;
    return (

        <div className="res-card">
            <img className="res-logo" src="https://media-assets.swigg
            <h3>{resData.info.name}</h3>
            <h4>{resData.info.cuisines.join(", ")} </h4>
            <h4>{resData.info.avgRating} Stars</h4>
            <h4>{resData.info.sla.deliveryTime} Minutes</h4>
        </div>

    )
}
```

Destructuring for one card of live API

```
const RestaurantCard =(props) =>
{
    const {resData} = props;
    return (

        <div className="res-card">
            <img className="res-logo" src="https://media-assets.swiggy.com/
            <h3>{resData.info.name}</h3>
            <h4>{resData.info.cuisines.join(", ")} </h4>
            <h4>{resData.info.avgRating} Stars</h4>
            <h4>{resData.info.sla.deliveryTime} Minutes</h4>
        </div>

    )
}
```

```
const Body =() =>
{
    return(
            <div className="Body">
                <div className="search">Search</div>
                <div className="res-container">
                <RestaurantCard  resData={resObj} />


                </div>
            </div>

    )
}
```

Making img also dynamic

1.go to swiggy's website
2. Open any img in new tab
3. Copy URL and change img id of that img (made changes last part of URL)
4. Observe img changes to another img

```
const {resData} = props;
return (

    <div className="res-card">
        <img className="res-logo" alt="" src={
          "https://media-assets.swiggy.com/swiggy/image/upload/fl_lossy,f_auto,q_auto,w_660/"
          + resData.info.cloudinaryImageId }  />
        <h3>{resData.info.name}</h3>
        <h4>{resData.info.cuisines.join(", ")} </h4>
        <h4>{resData.info.avgRating} Stars</h4>
        <h4>{resData.info.sla.deliveryTime} Minutes</h4>
    </div>
```

We are creating one card

So copy multiple cards data (from json file)and paste inside the resObj

* **Just cleaning mess** (Best Practice to write code)

problem

```
<div className="res-card">
    <img className="res-logo" alt="" src={
      "https://media-assets.swiggy.com/swiggy/image/upload/fl_lossy,f_auto,q_auto,w_660/"
      + resData.info.cloudinaryImageId }  />
    <h3>{resData.info.name}</h3>
    <h4>{resData.info.cuisines.join(", ")} </h4>
    <h4>{resData.info.avgRating} Stars</h4>
    <h4>{resData.info.sla.deliveryTime} Minutes</h4>
</div>
```

Destructing this content so not need to write again and again

Solution

```jsx
const RestaurantCard =(props) =>
{
    const {resData} = props;
    const {cloudinaryImageId, name, cuisines, avgRating }= resData?.info;
    const {deliveryTime}=resData?.info?.sla;
    return (

        <div className="res-card">
            <img className="res-logo" alt="" src={
              "https://media-assets.swiggy.com/swiggy/image/upload/fl_lossy,f_auto,q_auto,w_660/"
              + cloudinaryImageId }  />
            <h3>{name}</h3>
            <h4>{cuisines.join(", ")} </h4>
            <h4>{avgRating} Stars</h4>
            <h4>{deliveryTime} Minutes</h4>
        </div>

    )
}
```

Problem

```jsx
const Body =() =>
{
    return(
            <div className="body">
                <div className="search">Search</div>
                <div className="res-container">
                <RestaurantCard  resData={resList[0]} />
                <RestaurantCard  resData={resList[1]} />
                <RestaurantCard  resData={resList[2]} />
                <RestaurantCard  resData={resList[3]} />
                <RestaurantCard  resData={resList[4]} />
                <RestaurantCard  resData={resList[5]} />
                <RestaurantCard  resData={resList[6]} />
                <RestaurantCard  resData={resList[7]} />
                <RestaurantCard  resData={resList[8]} />
                <RestaurantCard  resData={resList[9]} />
                <RestaurantCard  resData={resList[10]} />
                <RestaurantCard  resData={resList[11]} />
                <RestaurantCard  resData={resList[12]} />
```

This is not good way need to do this cards dynamically

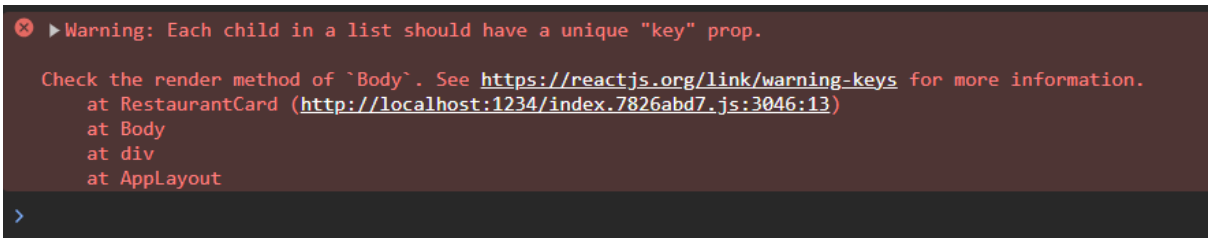
Solution

```jsx
const Body =() =>
{
    return(
            <div className="body">
                <div className="search">Search</div>
                <div className="res-container">

                {resList.map((restaurant)=>
                <RestaurantCard  resData={restaurant} />
                )
                }
```

Suppose we add or remove some restaurants then it will show less or more we don't worry about that

Problem



Solution

```
{resList.map((restaurant)=>
<RestaurantCard key={restaurant.info.id} resData={restaurant} />
)
```

When we use map, filter and reduce we need to give key
Or whenever we looping something need to use key

Without key

React does not identify element uniquely

It’s ok to use index as key but not recommended

```
// not using key (not acceptable ) <<< index as key <<<<<<<<<< Unique id (Best Practice)
```

Index is key better than not key
Not key is not acceptable