Episode 5 - Jets Get Hooked

Best Practices

1) Keep code in seperate files
2) Don't write all the code in one file
3) All the code of a project is kept in src folder.
4) Whenever you have a hardcoded data don't keep it in the same file as a component
5) Don't keep the image src link in the component file. Create a seperate file in utils/common folder

```
const CDN_URL = "https:/...."
const LOGO_URL = "https://...."
```

Snake Case

Types of Exports

1) Default Export → In a file it can be written just once to export one main function

   Eg :- export default Button;

2) Named Export → For multiple exports Named export is used

   Eg :- export { PrimaryButton, SecondaryButton };

More Example

Default Export/Import :

```
export default component;
import Component from "path";
```

Named Export / Import :

```
export const Component
import {Component} from "path";
```

React Hooks

React Hooks is a normal javascript function which is given to us by react which comes with some superpower

useState() → Superpowerful state variable

```
const [list, setList] = useState([])
```

↑ state variable ↑ state function

Filter on Button Click

Data from api or Hardcoded Array of objects

```
const [listOfRestaurant, setListOfRestaurant] = useState([{...}, {....}])

<button onClick={() => {
                    const filteredList = listOfRestaurants.filter(
                        (res) => res.data.avgRating > 4
                    );
                    setListOfRestaurant(filteredList);
                }}
```

A normal JS variable will not update the UI but a JS state variable does. It will keep the UI in sync with the variable. It will re-render the dom very fast with that component

React Algorithm

Reconciliation Algorithm → It will be used whenever there is some change in UI

Whenever you have something in UI react create a Virtual UI for it.

Virtual DOM is a representation of actual DOM in the form of a JS Object

The overall React Element gets converted into an Obj which is the Virtual DOM

Diff Algorithm → It finds out the difference between 2 Virtual doms - The updated Virtual DOM and the previous Virtual Dom Then re-renders it & update the actual dom

Incremental Rendering
The ability to split rendering work into chunks & spread it out over multiple frames

Why React is fast?
Because it is doing alot of efficient dom manipulation because it has a virtual dom. It can efficiently find the difference between 2 virtual dom and update only the changed portion in the actual dom using Diff and Reconciliation Algorithm.

Other ways of Writing useState Hook

```
Const arr = useState(reslist)

Const [listOfRestaurants, setListOfRestaurant] =arr

// This is array destructuring
```

Or

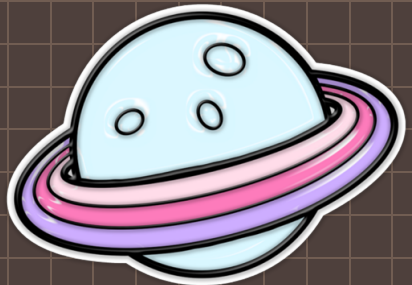
```
Const arr = usestate(reslist);

Const listOfRestaurants = arr[0];
Const setlistOfRestaurants = arr[1];
```

One way data binding
The data can come from or update from one place but not from the other way

Eg Can send a value to input from code but can't update the input from UI when using a Js variable instead of state variable

What is a hook?
Hook is a normal function. Like useState hooks returns an array first element of array is state variable and second element is a state function



Search logic

// Code

```
function filterData( searchText, restaurants) {
    const filterData = restaurants.filter ((rest) =>
                        rest.data.name.includes (searchText)
                      );
    return filterData;
}

filterData(searchText, RestaurantList);
```

Shimmer UI

// Code

```
{ Array (20).fill().map ((s, index) => {
        return (
            <div>
            </div>
        )
    }
    )
}
```

Style the Div accordingly