

## → React File Structure (Industry practices)

1) All main code in react library we keep in 'src' folder.

'src' folder have source code

All source code files put in 'src' folder.

2) Make a separate file for every component

Create 'component' folder for components source code files.

keep basic structure at start, If our project have large files we keep restructuring it.

Header.js or Header.jsx (extension depends on developer)

## → Import & Export

- In app.js we are using <Header> component.

- In this component App.js file <Header> component no longer exists. It does not recognize it and 'throw an error' Header is not defined'

- So, we have to import 'Header' component in app.js before importing it needs to export from Header.js file.

e.g. export default Header;

e.g App.js

import Header from "./components/Header.js" or  
Header";

### → config.js file

- whenever you have hard coded data or mockData never keep it in Component file

e.g mockData / staticData, Img CDN links,  
logo Png Link, Hard Coded string.

Create separate file for it

e.g config.js / constant.js

### 'utils' folder →

create for common utilities used  
~~at~~ all across the app

- constant.js

```
const CDN_URL = " ";  
const LOGO_URL = " ";
```

- Good practice to capital all constants

- mockData.js

```
const resList = [{} , {} , {}]
```

- List of all restaurants.

## → Types of Importing & exporting

- 1) Exporting Default Export & Import
- 2) Named Export & Import

### 3) Default Export & Import

e.g export default resList;

There is only one default export

e.g import resList from "../utils/mockData";

By default a file can only export one thing.

## 2) Named Export & Import

- In a single file , you have to export multiple things .
- e.g  
export const LOGO\_URL = " " ;  
export const CDN\_URL = " " ;  
import { LOGO\_URL } from "../utils/constant";  
import { CDN\_URL } from "../utils/constant";
- Write export keyword in front of variable to export
- for named import we always use { } .

## → Building Feature 'Top Rated Restaurant'

- Show only Restaurants , which have greater than 4 Rating .
- Brent Handlen

### I) onclick Event

- pass as attribute onClick = { }
- It takes callback function ( function which will be called onClick ) .
- this is arrow function .

e.g <button onClick={() => {  
  console.log("Button Clicked");  
}}> Top Rated Restaurants </button>

### • Logic

- 1] Restaurant Card data is coming from resList
- 2] If we change resList, UI will change  
Because it is config driven UI.
- 3] If anything get changed in resList, UI  
should change that is dynamism in react

### • Imp to remember remember

- There is 2 layer UI & data layer . UI layer will display on the basis of data i.e send by data layer
- Build logic for data layer

e.g <button onClick={() => {  
  listOfRes = listOfRes.filter((res) =>  
    res.data.avgRating > 4)  
}}> Top Rated Restaurant </button>

- This will filter out top rated restaurant.  
But change does not reflect on UI
- To update UI , we need React Variable

- 4] React is efficient in DOM manipulation
- 3) In React , UI layer & Data layer works in sync.

- To do this updation of UI with data layer , we use superpower React variable / state variable
- state variable or react variable maintains the state of component

## ↳ Hooks

- To create state variable we use react hooks (`useState()`)
- A react hook is a normal Javascript function.
- prebuilt / utility function.
- There are multiple react hooks  
2 most imp hooks
  - 1] `useState()`
  - 2] `useEffect()`

## ↳ `useState()` Hook

- This is used to generate superpowerful state variable in react

- Import from 'react'  
Import {useState} from 'react';
- state variable  
Maintains the state of component
- Local State Variable  
Scope of it inside that component only
- const [listOfRestaurant] = useState(initial value);  
state variable
- useState(): when we call it, it gives us a variable and we will receive it e.g listOfRestaurant.
- Normal JS Var  
let listOfRestaurant ;  
state React Var  
const [listOfRestaurants] = useState();
- const [listOfRes] = useState([{}, {}, {}]);  
Creation of Variable
- To modify state variable, we use a function.  
e.g const [listOfRestaurant, setListOfRestaurant] = useState([]);  
A function to modify variable

A function is used to update the list

- To update state variable, call function and push new data inside it
  - we can't modify state variable directly like JS variable
  - superpowerful state variable keeps UI & data layer in ~~sync~~ sync

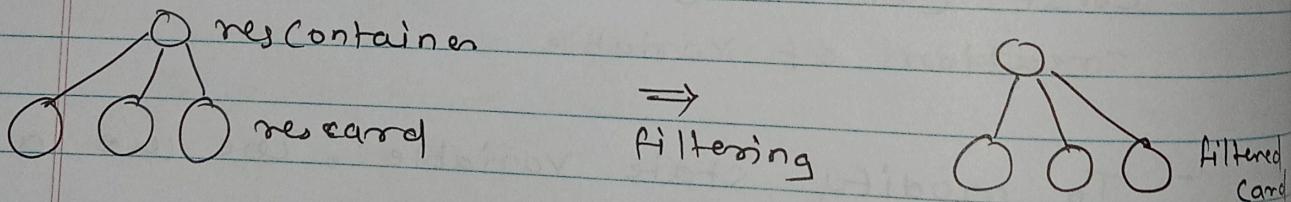
Note :

Note : whenever state variable update , react re-renders the component

→ Diff & Reconciliation Algorithm → React Fiber.

1) React uses 'reconciliation' algorithm  
Also known as 'react fiber'.

## 2) DOM (real DOM)



3) How React Do fast rendering?

- React creates virtual DOM of actual DOM
  - Virtual DOM : It is representation of an actual DOM.

Actual DOM

```
<div>
  <img>
  <h1>
</div>
```

Virtual DOM

```
{ }
```

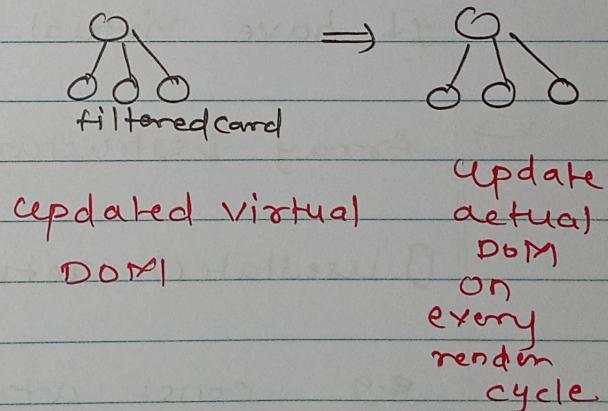
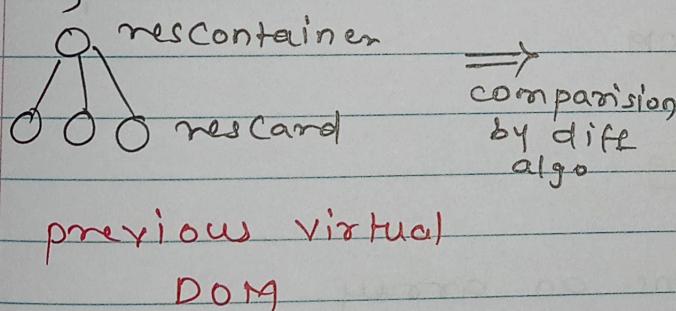
- To see `console.log(<Body>)`
- Returns an object
- object is virtual DOM

- Virtual DOM is basically react elements
- React creates object from JSX. Virtual DOM is normal Javascript Object

### → "diff" algorithm

- Finds out the difference between two Virtual DOM.  
ie updated Virtual DOM & previous Virtual DOM.

eg



### → Important to know

- ⇒ In react 16, new algo to update the DOM came out the algo is known as react fiber

- 2] New way of finding 'diff' & updating the DOM.
- 3] 'diff' algo finds out difference between OLD object & NEW object & update the DOM)
- 4] "Virtual DOM is like object representation of actual DOM".
- 5] It is easy to find out difference between two objects fast instead of two HTML node

### → Interview Question

Why React is fast?

Because react do efficient DOM manipulation  
It have virtual DOM.

### → Array Destructuring

- 1] useState() returns an array.

e.g. `const arr = useState ([resList]);  
const [listOfRes, setListOfRes] = arr;`  
OR

`const listofRes = arr[0];  
const setoflistRes = arr[1];`