

## 11 - Data is the New Oil

18 July 2023 21:35

### Higher Order Components:

- It's a function that takes a  $f()$  & return a  $f()$ .
- It's take the component as input & enhances the component if add extra features to the component & return it back.

### let's implement it

for e.g. In swiggy banglore they are 1807 restaurant

& it has few promoted restaurant.

The only diff b/w normal & promoted restaurant is it has promoted label at the top.

### let's create the feature:

```
costForTwo: 30000
costForTwoString: "*300 FOR TWO"
cuisines: () ['North Indian', 'Kebabs', 'Biryani']
deliveryTime: 37
favorite: false
feeDetails: {fees: Array(3), totalFees: 3700, message: '', title: 'Delivery Charge', amount: '3700', ...}
hasSurge: false
id: "356251"
lastMileTravel: 4.5
lastMileTravelString: "4.5 kms"
locality: ""
longDistanceEnabled: 0
maxDeliveryTime: 37
minDeliveryTime: 37
name: "Hotel Empire"
new: false
partnerId: 425
promoted: true
rainMode: "none"
ribbon: [{}]
select: false
sla: {restaurantId: '356251', deliveryTime: 37, minDeliveryTime: 37, maxDeliveryTime: 37, lastMileTravel: 4.5, ...}
```

in console log you can see promoted is true.

```

JS Body.js M JS RestaurantCard.js M
src > components > JS Body.js > [e] Body > filteredRestaurant.map() callback
  ...
  </div>
  <div className="flex flex-wrap">
    {filteredRestaurant.map((restaurant) => (
      <Link
        key={restaurant.data.id}
        to={`/restaurants/${restaurant.data.id}`}
        /* if the restaurant is promoted then add a promoted label to it */
        <RestaurantCard resData={restaurant} /> You, 3 minutes ago + Uncommitted changes
      </Link>
    ))
  </div>
</div>
);
};

export default Body;

```

```

JS Body.js M JS RestaurantCard.js M
src > components > JS RestaurantCard.js > [e] withPromotedLabel
  You, 1 second ago | 1 author (You)
  import { CDN_URL } from "../utils/constants";
  ...
  const RestaurantCard = (props) => {
    const { resData } = props;
    const { cloudinaryImageId, name, cuisines, avgRating, deliveryTime } =
      resData?.data;
    return (...);
  };
  ...
  // Higher Order Component
  // input -> restaurant card
  // output -> restaurantCardPromoted
  ...
const withPromotedLabel = (RestaurantCard) => {
  // as it will return another component inside it
  return () => {
    You, 1 second ago + Uncommitted changes
  }
};
export default RestaurantCard;

```

→ as it return another component inside it

```

  ...
  export const withPromotedLabel = (RestaurantCard) => {
    // as it will return another component inside it
    return (props) => {
      <div>
        <label>Promoted</label>
        <RestaurantCard {...props} /> → it will fetch all the props from RestaurantCard.
      </div>
    };
  };
  ...
  export default RestaurantCard;

```

→ it will fetch all the props from RestaurantCard.

- Now let's display all the categories from the Restaurant Menu.

```

  ...
  <ITEMARDS: (...) [1] > title: "Recommended"
    & type: "CATEGORY_TYPE_RECOMMENDED"
    & > [[Prototype]]: Object
    & > [[Prototype]]: Object
    & > [[Prototype]]: Object
  & 2:
    & card:
      & card:
        & type: "type.googleapis.com/swiggy.presentation.food.v2.ItemCategory"
        & itemards: (5) [{} , {} , {} , {} , {}]
          & title: "Beverages"
          & > [[Prototype]]: Object
          & > [[Prototype]]: Object
          & > [[Prototype]]: Object
  & 3:
    & card:
      & card:
        & type: "type.googleapis.com/swiggy.presentation.food.v2.ItemCategory"
        & itemards: (4) [{} , {} , {} , {}]
          & title: "Juice & Milk Shake"
          & > [[Prototype]]: Object

```

- in all this cards we have "@type" and it has ItemCategory  
 so we will filter out the ItemCategory as we need only  
 itemcategory. (Not Veg Only or Address)

`[ " @ type " ]`  
 ↳ as `@` is not a valid js so we use `[ " " ]`.



```

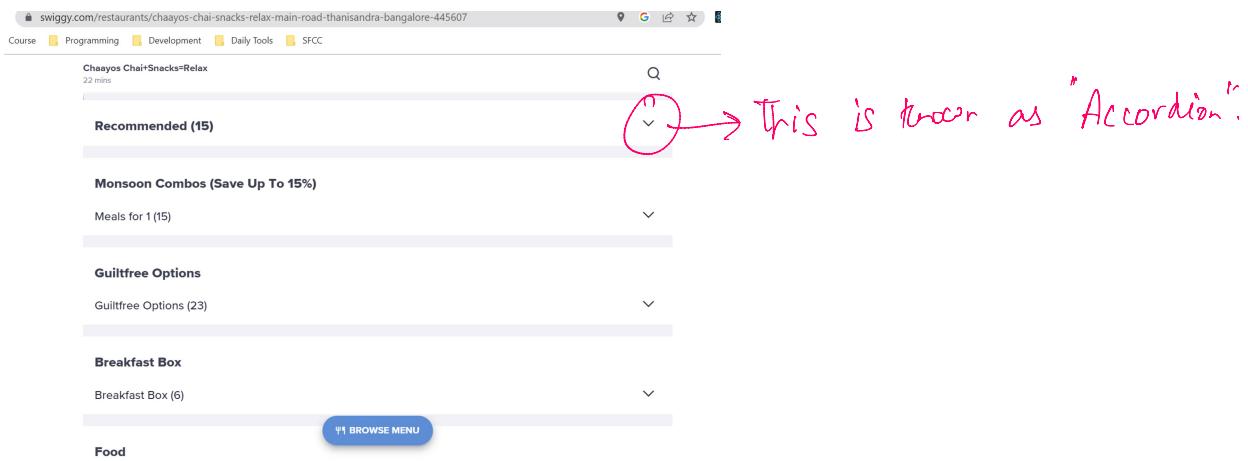
JS RestaurantMenu.js M X
src > components > JS RestaurantMenu.js > [e] RestaurantMenu
  ...
13
14   const { itemCards } =
15     resInfo?.cards[2]?.groupedCard?.cardGroupMap?.REGULAR?.cards[1]?.card?.card;
16   console.log(resInfo?.cards[2]?.groupedCard?.cardGroupMap?.REGULAR?.cards);
17
18   const categories =
19     resInfo?.cards[2]?.groupedCard?.cardGroupMap?.REGULAR?.cards.filter(
20     (c) =>
21       c.card?.card?.["@type"] ===
22         "type.googleapis.com/swiggy.presentation.food.v2.ItemCategory"
23   );
24   console.log(categories);      You, 1 second ago · Uncommitted changes

```

`(25) [{}]`  
`(20) [{}]`
RestaurantMenu.js:16  
RestaurantMenu.js:24

`↳ filtered out`

Let's build this type of UI



In Accordion we have two things

- Accordion Header
- Accordion Data

```

JS RestaurantMenu.js M × JS RestaurantCategory.js U ●
src > components > JS RestaurantMenu.js > ...
19  const categories =
20    resInfo?.cards[2]?.groupedCard?.cardGroupMap?.REGULAR?.cards.filter(
21      (c) =>
22        c.card?.card?.["@type"] ===
23        "type.googleapis.com/swiggy.presentation.food.v2.ItemCategory"
24    );
25    // console.log(categories);
26    return (
27      <div className="bg-gray-300 m-5 p-5 rounded-lg shadow-lg text-center">
28        <h1 className="text-2xl font-bold mb-2 ">{name}</h1>
29        <p className="text-gray-700 mb-4">
30          {cuisines ? cuisines.join(", ") : ""} - {costForTwoMessage}
31        </p>
32        /* Categories Accordions */
33        {categories.map((category)=> (
34          <RestaurantCategory data={category?.card?.card}/> → Passing maps (data) here
35        )))
36      </div>
37    );
38  };
39
40 export default RestaurantMenu;
41

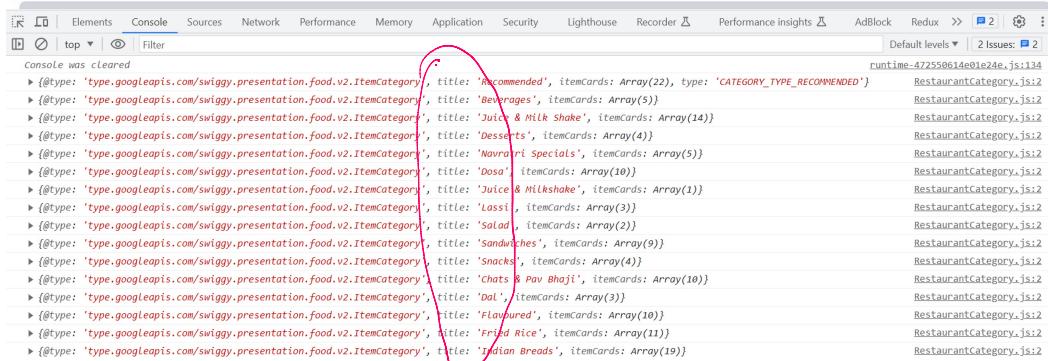
```

→ Passing maps (data) here  
↙ you will receive in Restaurant category

```

JS RestaurantMenu.js M × JS RestaurantCategory.js U ×
src > components > JS RestaurantCategory.js > [o] default
1  const RestaurantCategory = ({data})=> {
2    console.log(data)
3    return (<div>
4      /* Accordion Header */
5      RestaurantCategory
6      /* Accordion Data */
7    </div>
8  );
9
10 export default RestaurantCategory;

```



```

src > components > JS RestaurantCategory.js ...
1  const RestaurantCategory = ({ data }) => {
2    console.log(data);
3    return (
4      <div className="border border-gray-300 rounded-lg mb-4">
5        /* Accordion Header */
6        <div className="flex justify-between items-center px-4 py-2 bg-gray-200">
7          <span className="font-semibold">[data.title]</span>
8          <span className="text-2xl">↓</span>
9        </div>
10       /* Accordion Data */
11     </div>
12   );
13
14   export default RestaurantCategory;

```

Udupi Gardenia (jayanthinagar)  
South Indian, North Indian - ₹250 for two

- Recommended
- Beverages
- Juice & Milk Shake
- Desserts
- Navratri Specials
- Dosa

Let's make Hide & Show the Item lists!

If my showItems is true then only show the itemlist.

```

JS ItemList.js U JS RestaurantCategory.js U X JS RestaurantMenus M
src > components > JS RestaurantCategory.js > (1) RestaurantCategory
1 import { useState } from "react";
2 import ItemList from "./ItemList";
3
4 const RestaurantCategory = ({ data }) => {
5   const [showItems, setShowItems] = useState(false);
6
7   const handleClick = () => {
8     console.log("Clicked")
9   }
10
11   return (
12     <div className="border border-gray-300 rounded-lg mb-4">
13       <div className="flex justify-between items-center px-4 py-2 bg-white shadow-md cursor-pointer" onClick={handleClick}>
14         <span className="font-semibold text-center">{data?.title} ({data?.itemCards?.length})</span>
15         <span className="text-2xl">↓</span>
16       </div>
17     /* Accordion Body we will call it as item list */
18     {showItems ? <ItemList items={data.itemCards} />}
19   </div>
20 );
21
22 };
23
24 export default RestaurantCategory;
25

```

Now all the accordions is collapsed.

Recommended (19)

Sweets (5)

South Indian Snacks (11)

Dosa (16)

Beverages (8)

Chats (10)

Now onclick on it, it should show;

→ write a logic to turn it as true.

```

JS RestaurantCategory.js U X JS RestaurantMenus M
src > components > JS RestaurantCategory.js > [x] RestaurantCategory > [x] handleClick
1 √ import { useState } from "react";
2 import ItemList from "./ItemList";
3
4 √ const RestaurantCategory = ({ data }) => {
5
6   const [showItems, setShowItems] = useState(false);
7
8 √   const handleClick = () => {
9     | setShowItems(true);
10

```

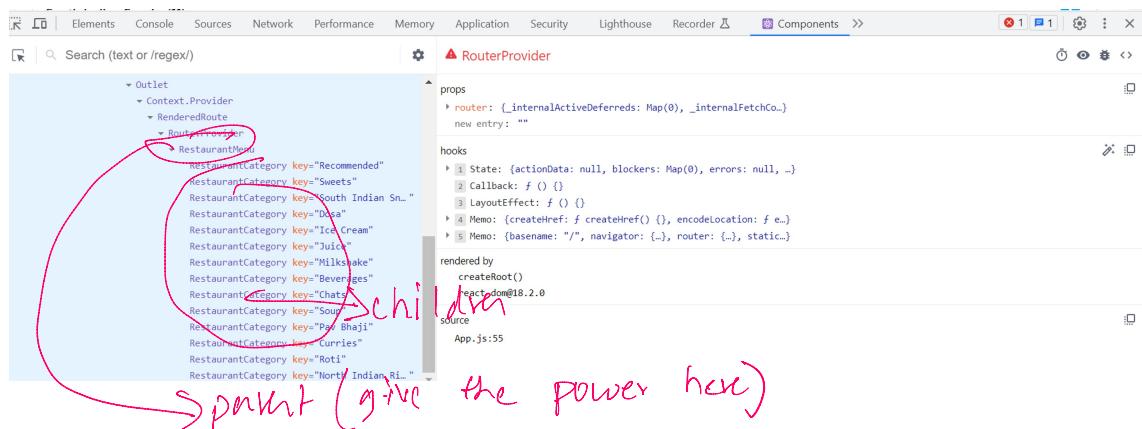
Now after expanding again on click it should hide;  
Basically a toggle.

```

JS RestaurantCategory.js U X JS RestaurantMenus M
src > components > JS RestaurantCategory.js > [x] RestaurantCategory > [x] handleClick
1 √ import { useState } from "react";
2 import ItemList from "./ItemList";
3
4 √ const RestaurantCategory = ({ data }) => {
5
6   const [showItems, setShowItems] = useState(false);
7
8 √   const handleClick = () => {
9     | setShowItems(!showItems); → Toggle.
10

```

Now when Recommended is shown & now you click on newly added Recommended should hide & newly added should be shown.



You can give power of show/hide to the parent.  
instead of category give it to the Restaurant Menu.  
it will control.

- give the showItem in the Menu & pass the props to the category.

```
JS RestaurantCategory.js U JS RestaurantMenu.js M
src > components > JS RestaurantMenu.js > ...
1 return (
2   <div className="bg-white m-5 p-5 rounded-lg shadow-lg">
3     <h1 className="text-2xl font-bold mb-2">{name}</h1>
4     <p className="text-gray-700 mb-4 font-semibold">
5       {cuisines ? cuisines.join(", ") : ""} - {costForTwoMessage}
6     </p>
7     {/* Categories Accordion */}
8     {categories.map((category) => (
9       <RestaurantCategory key={category?.card?.card?.title} data={category?.card?.card}>
10      {showItems = {false}}
11    </RestaurantCategory>
12  ))}
13 </div>
14 );
15 
```

```
JS RestaurantCategory.js U JS RestaurantMenu.js M
src > components > JS RestaurantCategory.js > RestaurantCategory > showItems
1 import { useState } from "react";
2 import ItemList from "./ItemList";
3
4 const RestaurantCategory = ({ data, showItems }) => {
5   // const [showItems, setShowItems] = useState(false);
6
7   // const handleClick = () => {
8   //   setShowItems(!showItems);
9   // }
10
11   return (
12     <div className="border border-gray-300 rounded-lg mb-4">
13       <div className="flex justify-between items-center px-4 py-2">
14         <span className="font-semibold text-center">{data?.title}</span>
15         <span className="text-2xl">{</span>
16       </div>
17       {/* Accordion Body we will call it as item list */}
18       {showItems && <ItemList items={data.itemCards} />}
19     </div>
20   );
21 }
22 
```

→ Controlled Component

```
JS RestaurantCategory.js U JS RestaurantMenu.js M
src > components > JS RestaurantCategory.js > RestaurantCategory
1 import { useState } from "react";
2 import ItemList from "./ItemList";
3
4 const RestaurantCategory = ({ data }) => {
5
6   const [showItems, setShowItems] = useState(false);
7
8   const handleClick = () => {
9     setShowItems(!showItems);
10   }
11
12   return (
13     <div className="border border-gray-300 rounded-lg mb-4">
14       <div className="flex justify-between items-center px-4 py-2">
15         <span className="font-semibold text-center">{data?.title}</span>
16         <span className="text-2xl">{</span>
17       </div>
18       {/* Accordion Body we will call it as item list */}
19       {showItems && <ItemList items={data.itemCards} />}
20     </div>
21   );
22 }
23 
```

→ Uncontrolled Component

We need to control the component if has props and didn't have main state.

```
JS RestaurantCategory.js U JS RestaurantMenu.js M
src > components > JS RestaurantMenu.js > RestaurantMenu > categories.map() callback
1 You, 2 minutes ago | 1 author (You)
import { useParams } from "react-router-dom";
2 import Shimmer from "./Shimmer";
3 import useRestaurantMenu from "../../../../utils/useRestaurantMenu";
4 import RestaurantCategory from "./RestaurantCategory";
5 import { useState } from "react";
6
7 const RestaurantMenu = () => {
8   const { resId } = useParams();
9   const resInfo = useRestaurantMenu(resId);
10
11   // by default 0th index is open
12   const [showIndex, setShowIndex] = useState(0);
13 }
14 
```

```
JS RestaurantCategory.js U JS RestaurantMenu.js M
src > components > JS RestaurantMenu.js > RestaurantMenu > categories.map() callback
33
34   <p>{cuisines ? cuisines.join(", ") : ""} - {costForTwoMessage}</p>
35   {/* Categories Accordion */}
36   {categories.map((category, index) => (
37     // Controller Component
38     <RestaurantCategory key={category?.card?.card?.title} data={category?.card?.card}>
39       {showItems = {index === showIndex ? true : false}}
40       setShowIndex= {() => setShowIndex(index)} → it's different all the time now
41     </RestaurantCategory>
42   ))}
43 
```

the feature works

```
JS RestaurantCategory.js U JS RestaurantMenu.js M
src > components > JS RestaurantCategory.js > ...
1 import { useState } from "react";
2 import ItemList from "./ItemList";
3
4 const RestaurantCategory = ({ data, showItems, setShowIndex }) => {
5
6   const handleClick = () => {
7     setShowIndex();
8   }
9
10   return (
11     <div>
12       <h1>{data?.title}</h1>
13       <div>
14         <span>{cuisines ? cuisines.join(", ") : ""}</span>
15         <span>${costForTwoMessage}</span>
16       </div>
17       <div>
18         <span>{data?.itemCards.length}</span>
19         <span>Items</span>
20       </div>
21       <div>
22         <span>{data?.rating}</span>
23         <span>Rating</span>
24       </div>
25       <div>
26         <span>{data?.cost}</span>
27         <span>Cost</span>
28       </div>
29       <div>
30         <span>{data?.cuisines}</span>
31         <span>Cuisines</span>
32       </div>
33       <div>
34         <span>{data?.address}</span>
35         <span>Address</span>
36       </div>
37       <div>
38         <span>{data?.phone}</span>
39         <span>Phone</span>
40       </div>
41       <div>
42         <span>{data?.id}</span>
43         <span>ID</span>
44       </div>
45       <div>
46         <span>{data?.lat}</span>
47         <span>Lat</span>
48       </div>
49       <div>
50         <span>{data?.lon}</span>
51         <span>Lon</span>
52       </div>
53       <div>
54         <span>{data?.cost}</span>
55         <span>Cost</span>
56       </div>
57       <div>
58         <span>{data?.cuisines}</span>
59         <span>Cuisines</span>
60       </div>
61       <div>
62         <span>{data?.address}</span>
63         <span>Address</span>
64       </div>
65       <div>
66         <span>{data?.phone}</span>
67         <span>Phone</span>
68       </div>
69       <div>
70         <span>{data?.id}</span>
71         <span>ID</span>
72       </div>
73       <div>
74         <span>{data?.lat}</span>
75         <span>Lat</span>
76       </div>
77       <div>
78         <span>{data?.lon}</span>
79         <span>Lon</span>
80       </div>
81       <div>
82         <span>{data?.cost}</span>
83         <span>Cost</span>
84       </div>
85       <div>
86         <span>{data?.cuisines}</span>
87         <span>Cuisines</span>
88       </div>
89       <div>
90         <span>{data?.address}</span>
91         <span>Address</span>
92       </div>
93       <div>
94         <span>{data?.phone}</span>
95         <span>Phone</span>
96       </div>
97       <div>
98         <span>{data?.id}</span>
99         <span>ID</span>
100      </div>
101      <div>
102        <span>{data?.lat}</span>
103        <span>Lat</span>
104      </div>
105      <div>
106        <span>{data?.lon}</span>
107        <span>Lon</span>
108      </div>
109      <div>
110        <span>{data?.cost}</span>
111        <span>Cost</span>
112      </div>
113      <div>
114        <span>{data?.cuisines}</span>
115        <span>Cuisines</span>
116      </div>
117      <div>
118        <span>{data?.address}</span>
119        <span>Address</span>
120      </div>
121      <div>
122        <span>{data?.phone}</span>
123        <span>Phone</span>
124      </div>
125      <div>
126        <span>{data?.id}</span>
127        <span>ID</span>
128      </div>
129      <div>
130        <span>{data?.lat}</span>
131        <span>Lat</span>
132      </div>
133      <div>
134        <span>{data?.lon}</span>
135        <span>Lon</span>
136      </div>
137      <div>
138        <span>{data?.cost}</span>
139        <span>Cost</span>
140      </div>
141      <div>
142        <span>{data?.cuisines}</span>
143        <span>Cuisines</span>
144      </div>
145      <div>
146        <span>{data?.address}</span>
147        <span>Address</span>
148      </div>
149      <div>
150        <span>{data?.phone}</span>
151        <span>Phone</span>
152      </div>
153      <div>
154        <span>{data?.id}</span>
155        <span>ID</span>
156      </div>
157      <div>
158        <span>{data?.lat}</span>
159        <span>Lat</span>
160      </div>
161      <div>
162        <span>{data?.lon}</span>
163        <span>Lon</span>
164      </div>
165      <div>
166        <span>{data?.cost}</span>
167        <span>Cost</span>
168      </div>
169      <div>
170        <span>{data?.cuisines}</span>
171        <span>Cuisines</span>
172      </div>
173      <div>
174        <span>{data?.address}</span>
175        <span>Address</span>
176      </div>
177      <div>
178        <span>{data?.phone}</span>
179        <span>Phone</span>
180      </div>
181      <div>
182        <span>{data?.id}</span>
183        <span>ID</span>
184      </div>
185      <div>
186        <span>{data?.lat}</span>
187        <span>Lat</span>
188      </div>
189      <div>
190        <span>{data?.lon}</span>
191        <span>Lon</span>
192      </div>
193      <div>
194        <span>{data?.cost}</span>
195        <span>Cost</span>
196      </div>
197      <div>
198        <span>{data?.cuisines}</span>
199        <span>Cuisines</span>
200      </div>
201      <div>
202        <span>{data?.address}</span>
203        <span>Address</span>
204      </div>
205      <div>
206        <span>{data?.phone}</span>
207        <span>Phone</span>
208      </div>
209      <div>
210        <span>{data?.id}</span>
211        <span>ID</span>
212      </div>
213      <div>
214        <span>{data?.lat}</span>
215        <span>Lat</span>
216      </div>
217      <div>
218        <span>{data?.lon}</span>
219        <span>Lon</span>
220      </div>
221      <div>
222        <span>{data?.cost}</span>
223        <span>Cost</span>
224      </div>
225      <div>
226        <span>{data?.cuisines}</span>
227        <span>Cuisines</span>
228      </div>
229      <div>
230        <span>{data?.address}</span>
231        <span>Address</span>
232      </div>
233      <div>
234        <span>{data?.phone}</span>
235        <span>Phone</span>
236      </div>
237      <div>
238        <span>{data?.id}</span>
239        <span>ID</span>
240      </div>
241      <div>
242        <span>{data?.lat}</span>
243        <span>Lat</span>
244      </div>
245      <div>
246        <span>{data?.lon}</span>
247        <span>Lon</span>
248      </div>
249      <div>
250        <span>{data?.cost}</span>
251        <span>Cost</span>
252      </div>
253      <div>
254        <span>{data?.cuisines}</span>
255        <span>Cuisines</span>
256      </div>
257      <div>
258        <span>{data?.address}</span>
259        <span>Address</span>
260      </div>
261      <div>
262        <span>{data?.phone}</span>
263        <span>Phone</span>
264      </div>
265      <div>
266        <span>{data?.id}</span>
267        <span>ID</span>
268      </div>
269      <div>
270        <span>{data?.lat}</span>
271        <span>Lat</span>
272      </div>
273      <div>
274        <span>{data?.lon}</span>
275        <span>Lon</span>
276      </div>
277      <div>
278        <span>{data?.cost}</span>
279        <span>Cost</span>
280      </div>
281      <div>
282        <span>{data?.cuisines}</span>
283        <span>Cuisines</span>
284      </div>
285      <div>
286        <span>{data?.address}</span>
287        <span>Address</span>
288      </div>
289      <div>
290        <span>{data?.phone}</span>
291        <span>Phone</span>
292      </div>
293      <div>
294        <span>{data?.id}</span>
295        <span>ID</span>
296      </div>
297      <div>
298        <span>{data?.lat}</span>
299        <span>Lat</span>
300      </div>
301      <div>
302        <span>{data?.lon}</span>
303        <span>Lon</span>
304      </div>
305      <div>
306        <span>{data?.cost}</span>
307        <span>Cost</span>
308      </div>
309      <div>
310        <span>{data?.cuisines}</span>
311        <span>Cuisines</span>
312      </div>
313      <div>
314        <span>{data?.address}</span>
315        <span>Address</span>
316      </div>
317      <div>
318        <span>{data?.phone}</span>
319        <span>Phone</span>
320      </div>
321      <div>
322        <span>{data?.id}</span>
323        <span>ID</span>
324      </div>
325      <div>
326        <span>{data?.lat}</span>
327        <span>Lat</span>
328      </div>
329      <div>
330        <span>{data?.lon}</span>
331        <span>Lon</span>
332      </div>
333      <div>
334        <span>{data?.cost}</span>
335        <span>Cost</span>
336      </div>
337      <div>
338        <span>{data?.cuisines}</span>
339        <span>Cuisines</span>
340      </div>
341      <div>
342        <span>{data?.address}</span>
343        <span>Address</span>
344      </div>
345      <div>
346        <span>{data?.phone}</span>
347        <span>Phone</span>
348      </div>
349      <div>
350        <span>{data?.id}</span>
351        <span>ID</span>
352      </div>
353      <div>
354        <span>{data?.lat}</span>
355        <span>Lat</span>
356      </div>
357      <div>
358        <span>{data?.lon}</span>
359        <span>Lon</span>
360      </div>
361      <div>
362        <span>{data?.cost}</span>
363        <span>Cost</span>
364      </div>
365      <div>
366        <span>{data?.cuisines}</span>
367        <span>Cuisines</span>
368      </div>
369      <div>
370        <span>{data?.address}</span>
371        <span>Address</span>
372      </div>
373      <div>
374        <span>{data?.phone}</span>
375        <span>Phone</span>
376      </div>
377      <div>
378        <span>{data?.id}</span>
379        <span>ID</span>
380      </div>
381      <div>
382        <span>{data?.lat}</span>
383        <span>Lat</span>
384      </div>
385      <div>
386        <span>{data?.lon}</span>
387        <span>Lon</span>
388      </div>
389      <div>
390        <span>{data?.cost}</span>
391        <span>Cost</span>
392      </div>
393      <div>
394        <span>{data?.cuisines}</span>
395        <span>Cuisines</span>
396      </div>
397      <div>
398        <span>{data?.address}</span>
399        <span>Address</span>
400      </div>
401      <div>
402        <span>{data?.phone}</span>
403        <span>Phone</span>
404      </div>
405      <div>
406        <span>{data?.id}</span>
407        <span>ID</span>
408      </div>
409      <div>
410        <span>{data?.lat}</span>
411        <span>Lat</span>
412      </div>
413      <div>
414        <span>{data?.lon}</span>
415        <span>Lon</span>
416      </div>
417      <div>
418        <span>{data?.cost}</span>
419        <span>Cost</span>
420      </div>
421      <div>
422        <span>{data?.cuisines}</span>
423        <span>Cuisines</span>
424      </div>
425      <div>
426        <span>{data?.address}</span>
427        <span>Address</span>
428      </div>
429      <div>
430        <span>{data?.phone}</span>
431        <span>Phone</span>
432      </div>
433      <div>
434        <span>{data?.id}</span>
435        <span>ID</span>
436      </div>
437      <div>
438        <span>{data?.lat}</span>
439        <span>Lat</span>
440      </div>
441      <div>
442        <span>{data?.lon}</span>
443        <span>Lon</span>
444      </div>
445      <div>
446        <span>{data?.cost}</span>
447        <span>Cost</span>
448      </div>
449      <div>
450        <span>{data?.cuisines}</span>
451        <span>Cuisines</span>
452      </div>
453      <div>
454        <span>{data?.address}</span>
455        <span>Address</span>
456      </div>
457      <div>
458        <span>{data?.phone}</span>
459        <span>Phone</span>
460      </div>
461      <div>
462        <span>{data?.id}</span>
463        <span>ID</span>
464      </div>
465      <div>
466        <span>{data?.lat}</span>
467        <span>Lat</span>
468      </div>
469      <div>
470        <span>{data?.lon}</span>
471        <span>Lon</span>
472      </div>
473      <div>
474        <span>{data?.cost}</span>
475        <span>Cost</span>
476      </div>
477      <div>
478        <span>{data?.cuisines}</span>
479        <span>Cuisines</span>
480      </div>
481      <div>
482        <span>{data?.address}</span>
483        <span>Address</span>
484      </div>
485      <div>
486        <span>{data?.phone}</span>
487        <span>Phone</span>
488      </div>
489      <div>
490        <span>{data?.id}</span>
491        <span>ID</span>
492      </div>
493      <div>
494        <span>{data?.lat}</span>
495        <span>Lat</span>
496      </div>
497      <div>
498        <span>{data?.lon}</span>
499        <span>Lon</span>
500      </div>
501      <div>
502        <span>{data?.cost}</span>
503        <span>Cost</span>
504      </div>
505      <div>
506        <span>{data?.cuisines}</span>
507        <span>Cuisines</span>
508      </div>
509      <div>
510        <span>{data?.address}</span>
511        <span>Address</span>
512      </div>
513      <div>
514        <span>{data?.phone}</span>
515        <span>Phone</span>
516      </div>
517      <div>
518        <span>{data?.id}</span>
519        <span>ID</span>
520      </div>
521      <div>
522        <span>{data?.lat}</span>
523        <span>Lat</span>
524      </div>
525      <div>
526        <span>{data?.lon}</span>
527        <span>Lon</span>
528      </div>
529      <div>
530        <span>{data?.cost}</span>
531        <span>Cost</span>
532      </div>
533      <div>
534        <span>{data?.cuisines}</span>
535        <span>Cuisines</span>
536      </div>
537      <div>
538        <span>{data?.address}</span>
539        <span>Address</span>
540      </div>
541      <div>
542        <span>{data?.phone}</span>
543        <span>Phone</span>
544      </div>
545      <div>
546        <span>{data?.id}</span>
547        <span>ID</span>
548      </div>
549      <div>
550        <span>{data?.lat}</span>
551        <span>Lat</span>
552      </div>
553      <div>
554        <span>{data?.lon}</span>
555        <span>Lon</span>
556      </div>
557      <div>
558        <span>{data?.cost}</span>
559        <span>Cost</span>
560      </div>
561      <div>
562        <span>{data?.cuisines}</span>
563        <span>Cuisines</span>
564      </div>
565      <div>
566        <span>{data?.address}</span>
567        <span>Address</span>
568      </div>
569      <div>
570        <span>{data?.phone}</span>
571        <span>Phone</span>
572      </div>
573      <div>
574        <span>{data?.id}</span>
575        <span>ID</span>
576      </div>
577      <div>
578        <span>{data?.lat}</span>
579        <span>Lat</span>
580      </div>
581      <div>
582        <span>{data?.lon}</span>
583        <span>Lon</span>
584      </div>
585      <div>
586        <span>{data?.cost}</span>
587        <span>Cost</span>
588      </div>
589      <div>
590        <span>{data?.cuisines}</span>
591        <span>Cuisines</span>
592      </div>
593      <div>
594        <span>{data?.address}</span>
595        <span>Address</span>
596      </div>
597      <div>
598        <span>{data?.phone}</span>
599        <span>Phone</span>
600      </div>
601      <div>
602        <span>{data?.id}</span>
603        <span>ID</span>
604      </div>
605      <div>
606        <span>{data?.lat}</span>
607        <span>Lat</span>
608      </div>
609      <div>
610        <span>{data?.lon}</span>
611        <span>Lon</span>
612      </div>
613      <div>
614        <span>{data?.cost}</span>
615        <span>Cost</span>
616      </div>
617      <div>
618        <span>{data?.cuisines}</span>
619        <span>Cuisines</span>
620      </div>
621      <div>
622        <span>{data?.address}</span>
623        <span>Address</span>
624      </div>
625      <div>
626        <span>{data?.phone}</span>
627        <span>Phone</span>
628      </div>
629      <div>
630        <span>{data?.id}</span>
631        <span>ID</span>
632      </div>
633      <div>
634        <span>{data?.lat}</span>
635        <span>Lat</span>
636      </div>
637      <div>
638        <span>{data?.lon}</span>
639        <span>Lon</span>
640      </div>
641      <div>
642        <span>{data?.cost}</span>
643        <span>Cost</span>
644      </div>
645      <div>
646        <span>{data?.cuisines}</span>
647        <span>Cuisines</span>
648      </div>
649      <div>
650        <span>{data?.address}</span>
651        <span>Address</span>
652      </div>
653      <div>
654        <span>{data?.phone}</span>
655        <span>Phone</span>
656      </div>
657      <div>
658        <span>{data?.id}</span>
659        <span>ID</span>
660      </div>
661      <div>
662        <span>{data?.lat}</span>
663        <span>Lat</span>
664      </div>
665      <div>
666        <span>{data?.lon}</span>
667        <span>Lon</span>
668      </div>
669      <div>
670        <span>{data?.cost}</span>
671        <span>Cost</span>
672      </div>
673      <div>
674        <span>{data?.cuisines}</span>
675        <span>Cuisines</span>
676      </div>
677      <div>
678        <span>{data?.address}</span>
679        <span>Address</span>
680      </div>
681      <div>
682        <span>{data?.phone}</span>
683        <span>Phone</span>
684      </div>
685      <div>
686        <span>{data?.id}</span>
687        <span>ID</span>
688      </div>
689      <div>
690        <span>{data?.lat}</span>

```

Read about lifting the state up

There is small bug in the accordion, when you select on any category (Recommend) it get expanded but again when you click on same category it doesn't hide.

To fix that

A screenshot of a code editor showing the `RestaurantCategory.js` file. The code defines a component that maps over categories and renders them as `RestaurantCategory` components. Each category is passed specific props: `key` (using `category?.card?.card?.title`), `data` (using `category?.card?.card`), `showItems` (using `index === showIndex`), and a `setShowIndex` function (using `() => setShowIndex(index === showIndex ? null : index)`).

```
JS RestaurantCategory.js U JS RestaurantMenu.js M
src > components > JS RestaurantCategory.js > [e] RestaurantCategory
1 <RestaurantCategory
2   key={category?.card?.card?.title}
3   data={category?.card?.card}
4   showItems={index === showIndex}
5   setShowIndex={() => setShowIndex(index === showIndex ? null : index)}
6 >/>
7 > >
```

A screenshot of a code editor showing the `RestaurantCategory.js` file. The code defines a `handleClick` function that takes no arguments and returns a function. This inner function takes a `prevIndex` parameter and returns either `null` or `showItems` based on whether they are equal.

```
JS RestaurantCategory.js U JS RestaurantMenu.js M
src > components > JS RestaurantCategory.js > [e] RestaurantCategory
1 import { useState } from "react";
2 import ItemList from "./ItemList";
3
4 const RestaurantCategory = ({ data, showItems, setShowIndex }) => {
5   const handleClick = () => {
6     setShowIndex((prevIndex) => (prevIndex === showItems ? null : showItems));
7   };
8 }
```

In the Restaurant Menu Component

- We added a 'showIndex' state variable to keep the track of the currently expanded category index. It's initialized with 'null' since no category is open by default.
- When mapping through the 'categories', we use the 'index' to determine whether the category should be expanded or collapsed based on whether its index matches the 'showIndex'.
- We pass the 'showIndex' state as a function 'setShowIndex' in the 'RestaurantCategory' component.

- we pass the 'showIndex' state as a function 'setShowIndex' to the 'RestaurantCategory' component.
  - The 'setShowIndex' f() is used to update the 'showIndex' state when a category is clicked. If the category index matches the 'showIndex', we set it to 'null' to collapse it.

## Props Drilling:

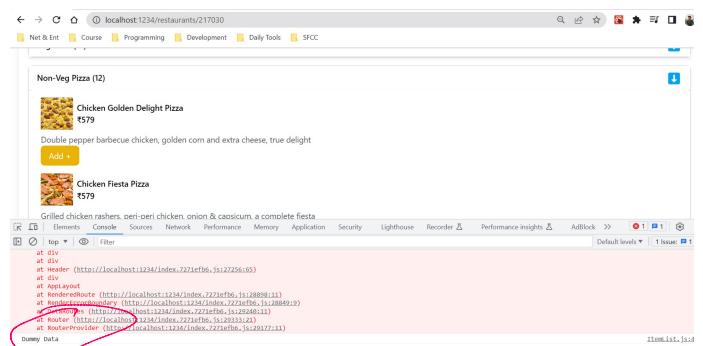
- you want to pass Dummy Data in ItemList but you can't pass it directly.
  - You need to pass it via its intermediate parent.

src > components > **JS RestaurantMenus.js** U    JS ItemList.js U    JS RestaurantMenu.js M X

```
You, 1 minute ago | 1 author (You)
1 import { useParams } from "react-router-dom";
2 import Shimmer from "./Shimmer";
3 import useRestaurantMenu from "../../utils/useRestaurantMenu";
4 import RestaurantCategory from "./RestaurantCategory";
5 import { useState } from "react";
6
7 const RestaurantMenu = () => {
8   const { resId } = useParams();
9
10   const dummy = "Dummy Data";
11   const resInfo = useRestaurantMenu(resId);
12
13   // CONSTRUCT Components
14   <RestaurantCategory
15     key={category?.card?.card?.title}
16     data={category?.card?.card}
17     showItems={index === showIndex}
18     setShowIndex={() => setShowIndex(index === showIndex ? null : index)}
19     dummy={dummy} />
20   You, 1 minute ago • Uncommitted changes
```

```
src > components > JS RestaurantCategory.js U X JS ItemList.js U JS RestaurantMenu.js M
1 import { useState } from "react";
2 import ItemList from "./ItemList";
3
4 const RestaurantCategory = ({ data, showItems, setShowIndex, dummy }) => {
5   const handleClick = () => {
6     setShowIndex((prevIndex) => (prevIndex === showItems ? null : showItems));
7   };
8
9   return (
10     <div className="border border-gray-300 rounded-lg mb-4">
11       <div
12         className="flex justify-between items-center px-4 py-2 bg-white shadow"
13         onClick={handleClick}
14       >
15         <span className="font-semibold text-center">
16           | {data?.title} ({data?.itemCards?.length})
17         </span>
18         <span className="text-2xl">↓</span>
19       </div>
20       {/* Accordion Body we will call it as item list */}
21       {showItems && <ItemList items={data.itemCards} dummy={dummy}>/}
22     </div>
23   );
24 }
25
```

```
JS RestaurantCategory.js U JS ItemList.js U X JS RestaurantMenu.js M
src > components > JS ItemList.js > (e) ItemList > (e) dummy
  1 import { CDN_URL } from "../../utils/constants";
  2 
  3 const ItemList = ({ items, dummy }) => {
  4   console.log(dummy);
  5   return (
  6     <div className="py-4 px-6 bg-white rounded-lg shadow-md">
  7       {items.map((item) => (
  8         <div key={item.card?.info?.id} className="mb-4">
  9           <div className="flex items-center mb-2">
 10             <img
 11               alt={CDN_URL + item.card.info.imageUrl}
 12             style={{"width": "40px", "height": "40px", "border": "1px solid #ccc", "border-radius": "50%", "object-fit": "cover"}}
 13             />
```



Now suppose if there are so many deep level components you want to pass from parent  $\rightarrow$  children the above

Now suppose if you -

You want to pass from parent → children the above way is NOT good way it's known as prop drilling.

### The problem with passing props

Passing props is a great way to explicitly pipe data through your UI tree to the components that use it.

But passing props can become verbose and inconvenient when you need to pass some prop deeply through the tree, or if many components need the same prop. The nearest common ancestor could be far removed from the components that need data, and lifting state up that high can lead to a situation called "prop drilling".

To avoid this we have something known as React Context.

React Context:

- It's like global space where the data is kept and anybody can access it.

For e.g.

- A piece of data like the data should be accessible anywhere from the app. (LoggedInUserInfo, Theme Dark/light)
- Context you can keep inside utils.

```
src > utils > UserContext.js
1 import {createContext} from "react";
2
3 const UserContext = createContext({
4   // pieces of information
5   loggedInUser: "Default User",
6 });
7
8 export default UserContext;
```

Now you want to access this in header component:

React gives us a hook to access it → useContext.

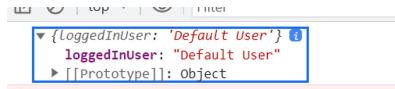
hook

const data = useContext(UserContext)

02-25

→ from the file.

```
src > components > Header.js > ...
1 import { LOGO_URL } from "../utils/constants";
2 import { useState, useContext } from "react";
3 import { Link, NavLink } from "react-router-dom";
4 import useOnlineStatus from "./utils/useOnlineStatus";
5 import UserContext from "../utils/UserContext";
6
7 const Header = () => {
8   const [btnNameReact, setBtnNameReact] = useState("Login");
9
10  const onlineStatus = useOnlineStatus();
11
12  const data = useContext(UserContext);
13  console.log(data);
14
```



To extract the loggedInUser:

```
src > components > Header.js > ...
11
12  const {loggedInUser} = useContext(UserContext);
13  console.log(loggedInUser);
14
```

```
src > components > Header.js > ...
83
84
85
86
87  <li className="px-4 font-semibold text-white">{loggedInUser}</li>
88
89
```

Nandhana Palace  
Andhra - ₹500 for two

Recommended (153)

Legendary Andhra Meals (3)

Breads (16)

Accompaniments (8)

Dessert (5)

Console

Default User

Console has cleared

Default User

runtime=472550614e01e24e.js:134

Header.js:13

To use useContext in ClassBased Component:

Class based component doesn't have hooks, so we do this way

<UserContext.Consumer>

{(data) => console.log(data)}

inside JSX → callback f() →

it has access to data & the

</UserContext.Consumer>

the data is context data.

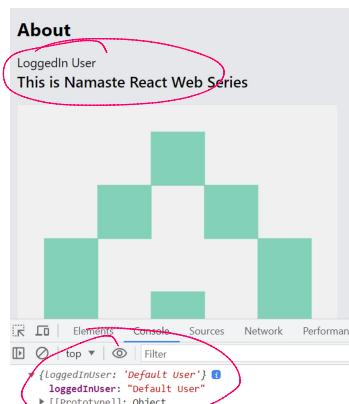
```

src > components > About.js > render
    You, 1 minute ago | 1 author (You)
1   import UserClass from "./UserClass";
2   import { Component } from "react";
3   import UserContext from "../utils/UserContext";
4
5   class About extends Component {
6       constructor(props) {
7           super(props);
8       }
9
10      componentDidMount() {
11      }
12
13      render() {
14          return (
15              <div className="bg-gray-200 p-4">
16                  <h1 className="text-2xl font-bold mb-4">About</h1>
17                  <div>
18                      LoggedIn User
19                      <UserContext.Consumer>
20                          {(data) => {
21                              console.log(data)
22                          }}
23                      </UserContext.Consumer>
24                  </div>
25          )
26      }
27
28  }

```

You, 1 minute ago | 1 author (You)

This is Namaste React Web Series



```

1   <div className="text-2xl font-bold mb-4">About</div>
2   <div>
3       LoggedIn User
4       <UserContext.Consumer>
5           {(LoggedInUser) => {
6               You, 19 seconds ago + Uncommitted changes
7                   <h1 className="text-2xl font-bold">{LoggedInUser}</h1>
8           }
9       </UserContext.Consumer>
10  </div>

```

What if we need to change the value of the User?

for e.g. In authentication user are different

So in root level:

```

src > App.js > AppLayout
import Body from "./components/Body";
import About from "./components/About";
import Contact from "./components/Contact";
import Error from "./components/Error";
import { createBrowserRouter, RouterProvider, Outlet } from "react-router-dom";
import RestaurantMenu from "./components/RestaurantMenu";
const Grocery = lazy(() => import("./components/Grocery"));

const AppLayout = () => {
    const [userInfo, setUserInfo] = useState();
    // authentication
    useEffect(() => {
        // Make an API call and send username and password
        // dummy data
        const data = {
            name: "Rizon Kumar Rahi"
        }
        // update the userInfo
        setUserInfo(data.name);
    }, [])
}

```

You, now + Uncommitted changes

now by default the default user is called, now if

I have to update it now we need to pass context

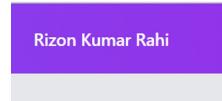
to APP.

We will use context provider.

↳ you need to wrap in the whole app.

↳ you need to wrap in the whole app.

```
src > App.js > AppLayout.js
1 import { createBrowserRouter, RouterProvider, Outlet } from "react-router-dom";
2 import RestaurantMenu from "./components/RestaurantMenu";
3 import UserContext from "./utils/UserContext";
4
5 const Grocery = lazy(() => import("./components/Grocery"));
6
7
8 const AppLayout = () => {
9   const [userName, setUserName] = useState();
10  // You, 1 second ago + Uncommitted changes
11  // authentication
12  // Make an API call and send username and password
13  // dummy data
14  const data = {
15    name: "Rizon Kumar Rahi"
16  }
17  // update the userInfo
18  setUserName(data.name);
19  []
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
```



Can you have nested provider and change the value.

```
31
32
33
34
35
36
37
38
39
40
41
```

Yes you can.

The header will be "Elon Mask"

Above "Top Restaurant" create a ip box and modify the  
userName with ip box and it should update it.

(watch the video (2:50 - ))