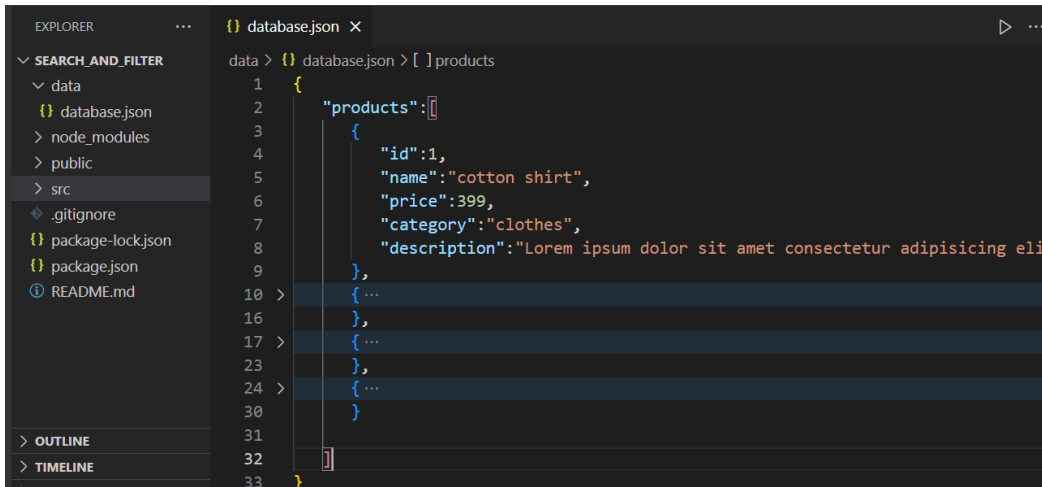
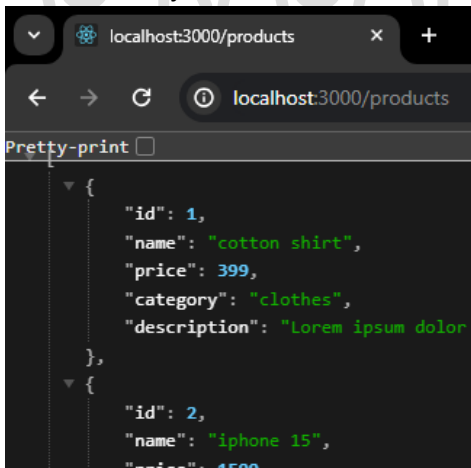


1. >npx create-react-app search_and_filter
2. Working with json server
 - a. Create data folder (at project level)
 - b. Create database.json file in it and add 5-8 products with category – clothes, mobile and tv



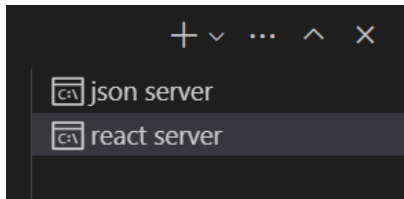
```
data > {} database.json > [ ] products
1  {
2    "products": [
3      {
4        "id": 1,
5        "name": "cotton shirt",
6        "price": 399,
7        "category": "clothes",
8        "description": "Lorem ipsum dolor sit amet consectetur adipisicing eli
9      },
10     { ...
16   },
17   { ...
23   },
24   { ...
30   }
31
32
33 }
```

- c. Install json server
 - i. >npm i -g json-server
- d. Start json server
 - i. Open cmd at project location
 - ii. >json-server -w ./data/database.json
- e. Check for our json file in the browser

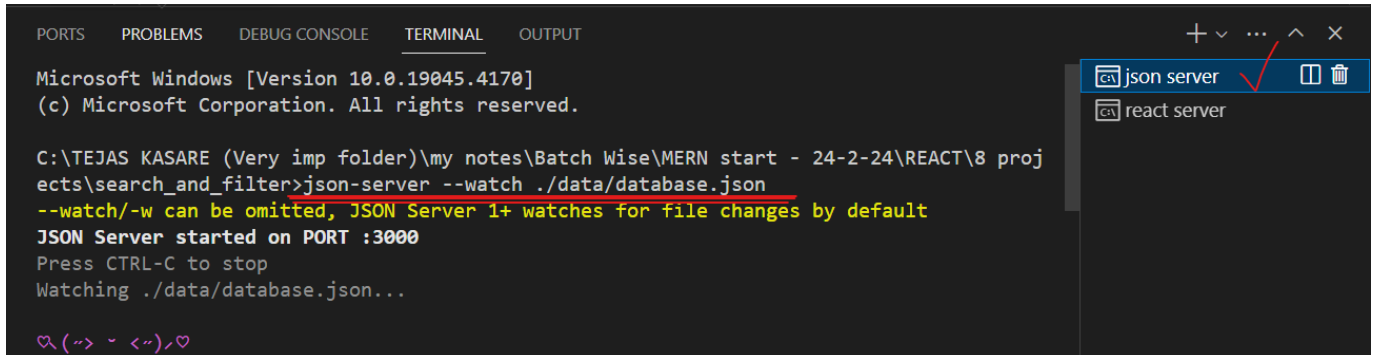


```
localhost:3000/products
localhost:3000/products
Pretty-print
{
  "id": 1,
  "name": "cotton shirt",
  "price": 399,
  "category": "clothes",
  "description": "Lorem ipsum dolor
},
{
  "id": 2,
  "name": "iphone 15",
  "price": 1599
```

3. Creating 2 terminals

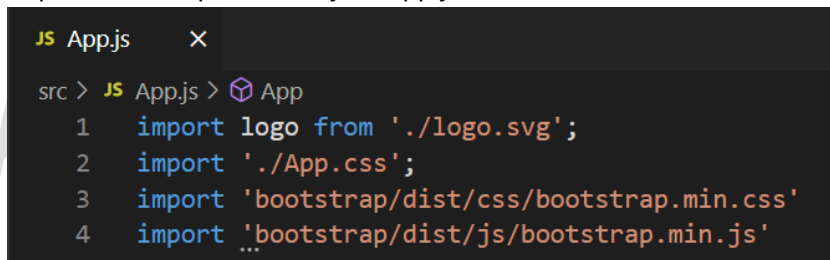


Start json server in json-server terminal



4. Adding bootstrap to project

- Open cmd at project location
- >npm i [bootstrap@5.3.3](#)
- Import bootstrap's css and js in App.js



5. Displaying data into component

Let our project structure will be

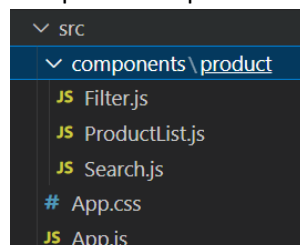
<App/>

<Filter/>

<Search/>

<ProductList/>

- Create components folder in project level
 - components > product > Filter.js, ProductList.js, Search.js



b. initial code for above files :

```
JS Search.js ×
src > components > product > JS Search.js > ...
1 import React from 'react'
2
3 export default function Search() {
4   return (
5     <>
6     | <h1>Search</h1>
7     </>
8   )
9 }
10 |

JS Filter.js ×
src > components > product > JS Filter.js > ...
1 import React from 'react'
2
3 export default function Filter() {
4   return (
5     <>
6     | <h1>Filter</h1>
7     </>
8   )
9 }

JS ProductList.js ×
src > components > product > JS ProductList.js > ...
1 import React from 'react'
2 export default function ProductList() {
3   return (
4     <>
5     | <h1>ProductList</h1>
6     </>
7   )
8 }
```

```
import './App.css';
import 'bootstrap/dist/css/bootstrap.min.css'
import 'bootstrap/dist/js/bootstrap.min.js'
import Filter from './components/product/Filter';
import Search from './components/product/Search';
import ProductList from './components/product/ProductList';

function App() {

  return (
    <>
      <div className='container'>
        <div className="row">
          <div className="col-lg-6 p-2 text-center ">
            <Filter/>
          </div>
          <div className="col-lg-6 ">
            <Search/>
          </div>
        </div>
        <div className="row mt-3">
          <div className="col">
            <ProductList/>
          </div>
        </div>
      </div>
    </>
  );
}
export default App;
```

OUTPUT :

>npm start



Filter

Search

ProductList

- c. fetching data from api and displaying
 - i. stop react server (bcoz, it is using 3000 port which is required for json server)
 - ii. start json server

1. `package.json` file

```

{
  "scripts": {
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject",
    "json:start": "json-server --watch ./data/database.json"
  },
  "eslintConfig": {

```

2. `"json:start": "json-server --watch ./data/database.json"`

3. `json server` and `react server` in the taskbar

4. `npm run json:start` in the terminal

Terminal output:

```

Microsoft Windows [Version 10.0.19045.4170]
(c) Microsoft Corporation. All rights reserved.

C:\TEJAS KASARE (Very imp folder)\my notes\Batch Wise\MERN start - 24-2-24\REACT\8 projects\search_and_filter>npm run json:start

> search_and_filter@0.1.0 json:start
> json-server --watch ./data/database.json

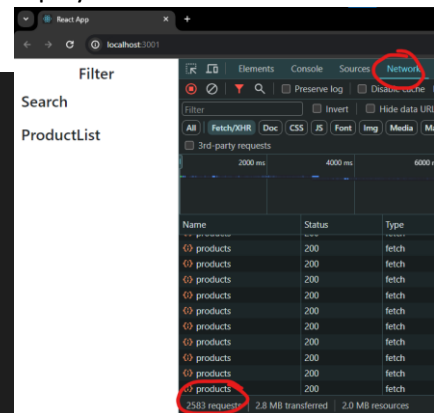
```

- iii. start react server `> npm start`
- iv. fetch data into `App.js` and pass it to `ProductList.js` to display

```

function App() {
  let [products, setProducts]=useState([])
  console.log(products);
  fetch("http://localhost:3000/products")
    .then(res => res.json())
    // .then(data=>console.log(data))
    .then(data=>setProducts(data))
    .catch()
  return (

```



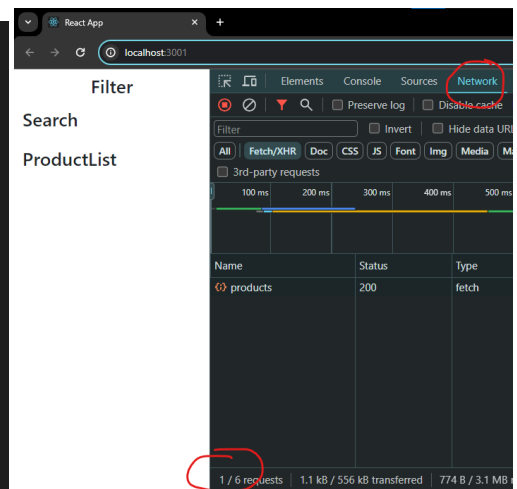
This will cause infinite API call because of re rendering
 Solution : `useEffect()` to handle side effect

```

function App() {
  let [products, setProducts]=useState([])
  console.log(products);

  useEffect(()=>{
    fetch("http://localhost:3000/products")
      .then(res => res.json())
      // .then(data=>console.log(data))
      .then(data=>setProducts(data))
      .catch()
    },[])
  return (

```



- v. pass this fetched products to ProductList.js and display

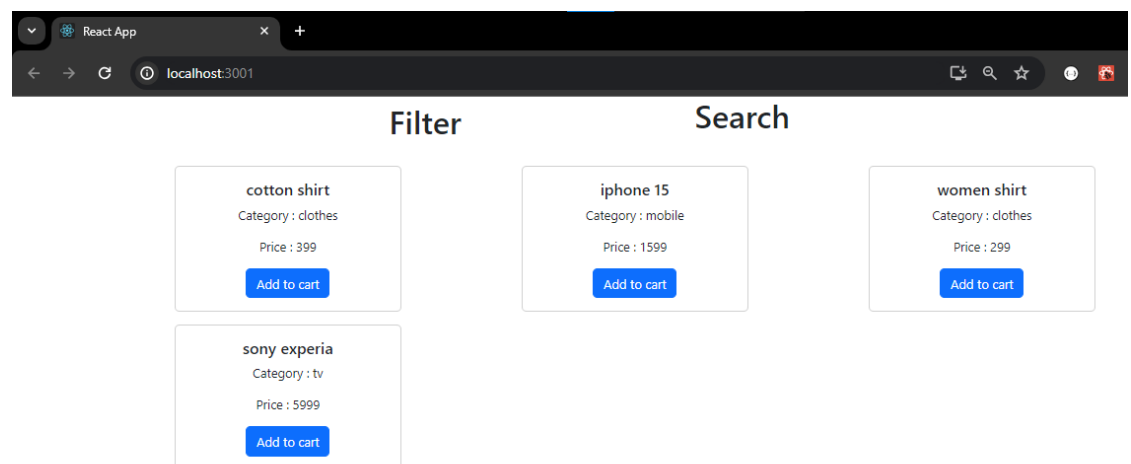
```
JS App.js x
src > JS App.js > ...
33     </div>
34     <div className="row mt-3">
35       <div className="col">
36         <ProductList products={products}/>
37       </div>
38     </div>
39   </div>
40 </>
```

Code for ProductList.js

```
import React from 'react'

export default function ProductList(props) {
  return (
    <>
      <div className="row">
        {props.products.map(product =>
          <div className="col-lg-4 mb-3">
            <div className="card text-center" style={{width: "18rem"}}>
              <div className="card-body">
                <h5 className="card-title">{product.name}</h5>
                <p className="card-text">Category : {product.category}</p>
                <p className="card-text">Price : {product.price}</p>
                <a href="#" className="btn btn-primary">Add to cart</a>
              </div>
            </div>
          </div>
        )}
      </div>
    </>
  )
}
```

Output :



6. Filtration Logic

a. Update App.js as :

```
import './App.css';
import 'bootstrap/dist/css/bootstrap.min.css'
import 'bootstrap/dist/js/bootstrap.min.js'
import Filter from './components/product/Filter';
import Search from './components/product/Search';
import ProductList from './components/product/ProductList';
import { useEffect, useState } from 'react';

function App() {
  let [products, setProducts]=useState([])
  let [url, setsetUrl]=useState("http://localhost:3000/products")
  console.log(products);

  useEffect(()=>{
    fetch(url)
      .then(res => res.json())
      // .then(data=>console.log(data))
      .then(data=>setProducts(data))
      .catch()
  },[url])

  const changeUrl =(category) =>{
    setsetUrl("http://localhost:3000/products?category="+category)
  }

  return (
    <>
      <div className='container'>
        <div className="row">
          <div className="col-lg-6 p-2 text-center">
            <Filter onChangeUrl={changeUrl}/>
          </div>
          <div className="col-lg-6 p-2">
            <Search/>
          </div>
        </div>
        <div className="row mt-3">
          <div className="col">
            <ProductList products={products}/>
          </div>
        </div>
      </>
    );
  }

  export default App;
```

b. Add radio group buttons in Filter.js and add following code

```
import React from 'react'

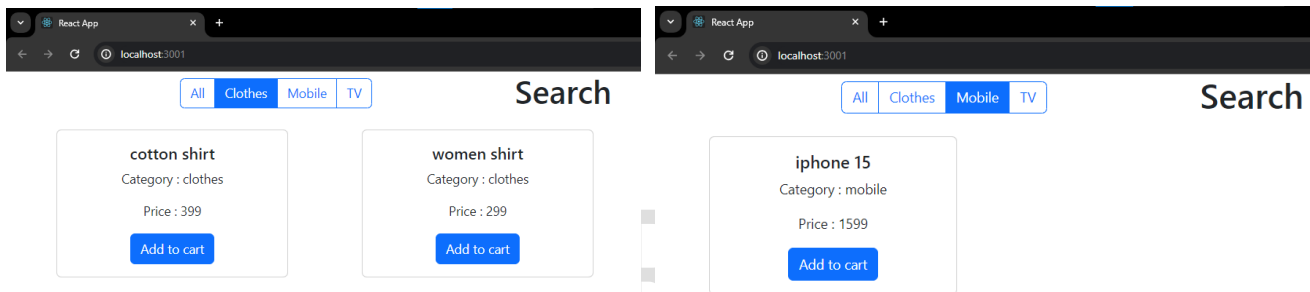
export default function Filter(props) {
  return (
    <div>
      <div className="btn-group" role="group" aria-label="Basic radio toggle button group">
        <input type="radio" className="btn-check" name="btnradio" id="btnradio1" />
        <label className="btn btn-outline-primary" htmlFor="btnradio1" onClick={()=>{props.onChageUrl("")}}>All</label>

        <input type="radio" className="btn-check" name="btnradio" id="btnradio2" />
        <label className="btn btn-outline-primary" htmlFor="btnradio2" onClick={()=>{props.onChageUrl("clothes")}}>Clothes</label>

        <input type="radio" className="btn-check" name="btnradio" id="btnradio3" />
        <label className="btn btn-outline-primary" htmlFor="btnradio3" onClick={()=>{props.onChageUrl("mobile")}}>Mobile</label>

        <input type="radio" className="btn-check" name="btnradio" id="btnradio4" />
        <label className="btn btn-outline-primary" htmlFor="btnradio4" onClick={()=>{props.onChageUrl("tv")}}>TV</label>
      </div>
    </div>
  )
}
```

Output :

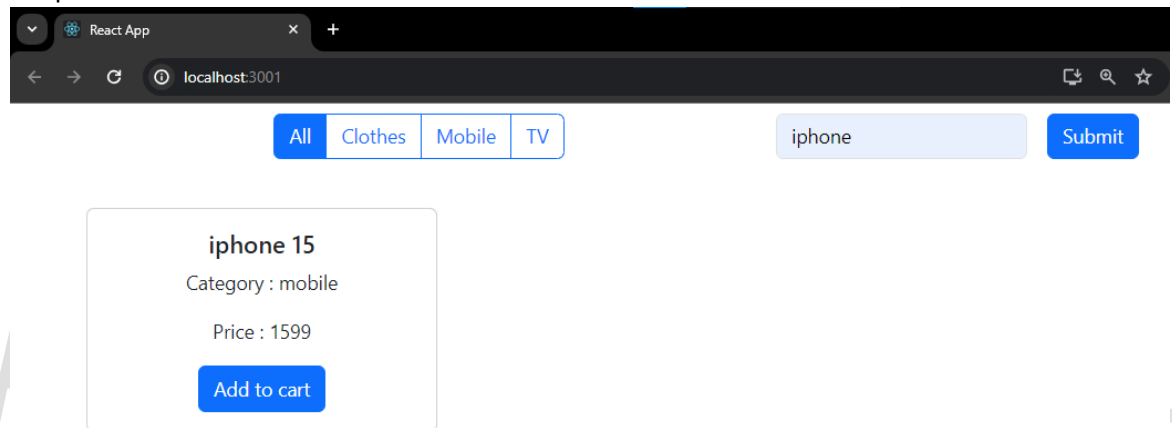



```

<>
  <form className="row g-3" onSubmit={submitForm}>
    <div className="col-auto">
      <input type="text" className="form-control" id="product_name" placeholder="Product
name"
      onChange={changeProductName}/>
    </div>
    <div className="col-auto">
      <input type="submit" className="btn btn-primary mb-3"/>
    </div>
  </form>
</>
)
}

```

d. Output



8. Fetching data using async await pattern

```

JS App.js
src > JS App.js > App > changeUrl
15 console.log(products);
16
17 //collecting data from api by using fetch() function
18 // useEffect(()=>{
19 //   fetch(url)
20 //   .then(res => res.json())
21 //   // .then(data=>console.log(data))
22 //   .then(data=> {setProducts(data)})
23 //   .catch()
24 //   .catch()
25 // },[url])
26
27 //collecting data from api by using async await pattern
28 const fetchProducts = async() =>
29 {
30   let response= await fetch(url)
31   let data=await response.json()
32   setProducts(data)
33 }
34 useEffect(()=>
35 {
36   fetchProducts()
37 },[url])
38
39 const changeUrl =(category) =>{
40   setsetUrl("http://localhost:3000/products?category="+category)
41 }
42

```

9. Loading functionality

a. Manage state

```
JS App.js x
src > JS App.js > App > fetchProducts
9
10 function App() {
11   let [products, setProducts]=useState([])
12   let [loading, setLoading]=useState(true)
13   let all_products;
```

b. Setting loading to false after receiving data

```
JS App.js x
src > JS App.js > App > fetchProducts
28 //collecting data from api by using
29 const fetchProducts = async() =>
30 {
31   let response= await fetch(url)
32   let data=await response.json()
33   setLoading(false)
34   setProducts(data)
35 }
36 }
```

c. Showing loading...

```
JS App.js x
src > JS App.js > App > fetchProducts
68 </div>
69 {loading && "loading..."} ✓
70 <div className="row mt-3">
71   <div className="col">
72     <ProductList products={products} />
73   </div>
74 </div>
```

10. Creating custom hook

- Create hooks folder in src
- Create useFetchApi.js in hooks folder
- Code for useFetchApi.js

```
import React, { useEffect, useState } from 'react'

export default function useFetchApi(url) {
  let [data, setData]=useState(null)
  let [loading, setLoading]=useState(true)
  let [error, setError]=useState(null)

  const fetchProducts = async() =>
  {
    try
    {
      let response= await fetch(url)
      let result=await response.json()
      setLoading(false)
      setData(result)
    }
    catch(er)
    {
      setError(er)
      setLoading(false)
    }
  }

  useEffect(()=>
  {
    fetchProducts()
  },[url])

  return {data,loading,setData,error}
}
```

- Update following code in App.js

```
import './App.css';
import 'bootstrap/dist/css/bootstrap.min.css'
import 'bootstrap/dist/js/bootstrap.min.js'
import Filter from './components/product/Filter';
import Search from './components/product/Search';
import ProductList from './components/product/ProductList';
import { useEffect, useState } from 'react';
import useFetchApi from './hooks/useFetchApi';

function App() {
  let [url, setUrl]=useState("http://localhost:3000/products")
  let {data:products,loading,setData,error}=useFetchApi(url)

  const changeUrl =(category) =>{
    setUrl("http://localhost:3000/products?category="+category)
  }

  //searching logic
  const searchProduct = (productName) =>
  {
    let searchedProducts = products.filter(product =>
      product.name.toLowerCase()
      .includes(productName.toLowerCase())
    )
  }
}
```

```

    // setProducts(searchedProducts)
    setData(searchedProducts)
  }

  return (
    <>
      <div className='container'>
        <div className="row">
          <div className="col-lg-6 p-2 text-center">
            <Filter onChangeUrl={changeUrl}/>
          </div>
          <div className="col-lg-6 p-2">
            <Search onSearch={searchProduct}/>
          </div>
        </div>
        <div className="row mt-3">
          <div className="col text-center">
            {loading &&  }
            {error &&  }
          </div>
        </div>

        <div className="row mt-3">
          <div className="col">
            <ProductList products={products} />
          </div>
        </div>
      </>
    );
  }
}

export default App;

```

e. Update following in ProductList.js

```

JS ProductList.js X
src > components > product > JS ProductList.js > ProductList > props.products.map() call
5      <>
6      <div className="row">
7      {props.products && props.products.map(product =>
8      <div className="col-lg-4 mb-3" key={product.id}>

```

You will see above images in 2 cases :

- Passing wrong API url – to see something went wrong
- Make network slow 3G – to see loading

11. Cancelling api call

TEJAS KASARE 8459849415