React Resources

what is there in that above link that only pasted below okay ...

React Resources

Note: This resource section only makes sense if you have watched the course lectures otherwise you will see random points and multiple links.

Includes: Reference links, reading material, cheat-sheet PDFs, exercises and more.

Documentations:

- https://reactjs.org/
- https://beta.reactjs.org/
- https://react.dev/
- CRA: https://reactjs.org/docs/create-a-new-react-app.html StackBlitz: https://stackblitz.com/fork/react
- Extensions:
- ES7 React Snippet, Auto Import, Material Icon Theme, Simple React Snippets
- Theme (File > Preferences > Color Theme)
- Settings ((Ctrl) + ,) Search "emmet include languages", item: **javascript**, value: **javascriptreact** (and then hit ok)
- Settings ((Ctrl) + ,) Search “Sticky Scroll” > Check Enable
- Common Doubts:
- How To Use Specific React Version With CRA: https://stackoverflow.com/a/72685421/7846238
- Axios vs Fetch https://stackoverflow.com/a/50326744/7846238

Code Files,

- Lecture code files and resources: https://github.com/ShubhamSarda/react-ul

Project Demo:

- Taskmate: https://taskmate-ul.netlify.app | https://github.com/ShubhamSarda/taskmate-react
- Cinemate: https://cinemate-ul.netlify.app | https://github.com/ShubhamSarda/cinemate
- Shopmate (Context Reducer): https://shopmatecr-ul.netlify.app |
- Shopmate (Redux): https://shopmate-redux.netlify.app | https://github.com/ShubhamSarda/shopmate-redux
- Codebook: https://codebook-ul.netlify.app | https://github.com/ShubhamSarda/codebook
- Writenode: https://writenode-ul.netlify.app | https://github.com/ShubhamSarda/writenode

Assignment:

- Shopmate: https://shopmate-ul.netlify.app | https://github.com/ShubhamSarda/shopmate
- Word Counter: https://wordcounter-ul.netlify.app | https://github.com/ShubhamSarda/wordcount-ul

Conceptual Project:

- Taskmate: https://github.com/ShubhamSarda/react-ul/tree/main/030
- Routemate: https://github.com/ShubhamSarda/react-ul/tree/main/060

Tailwind Demo

- Demo: https://tailwind-ul.netlify.app/

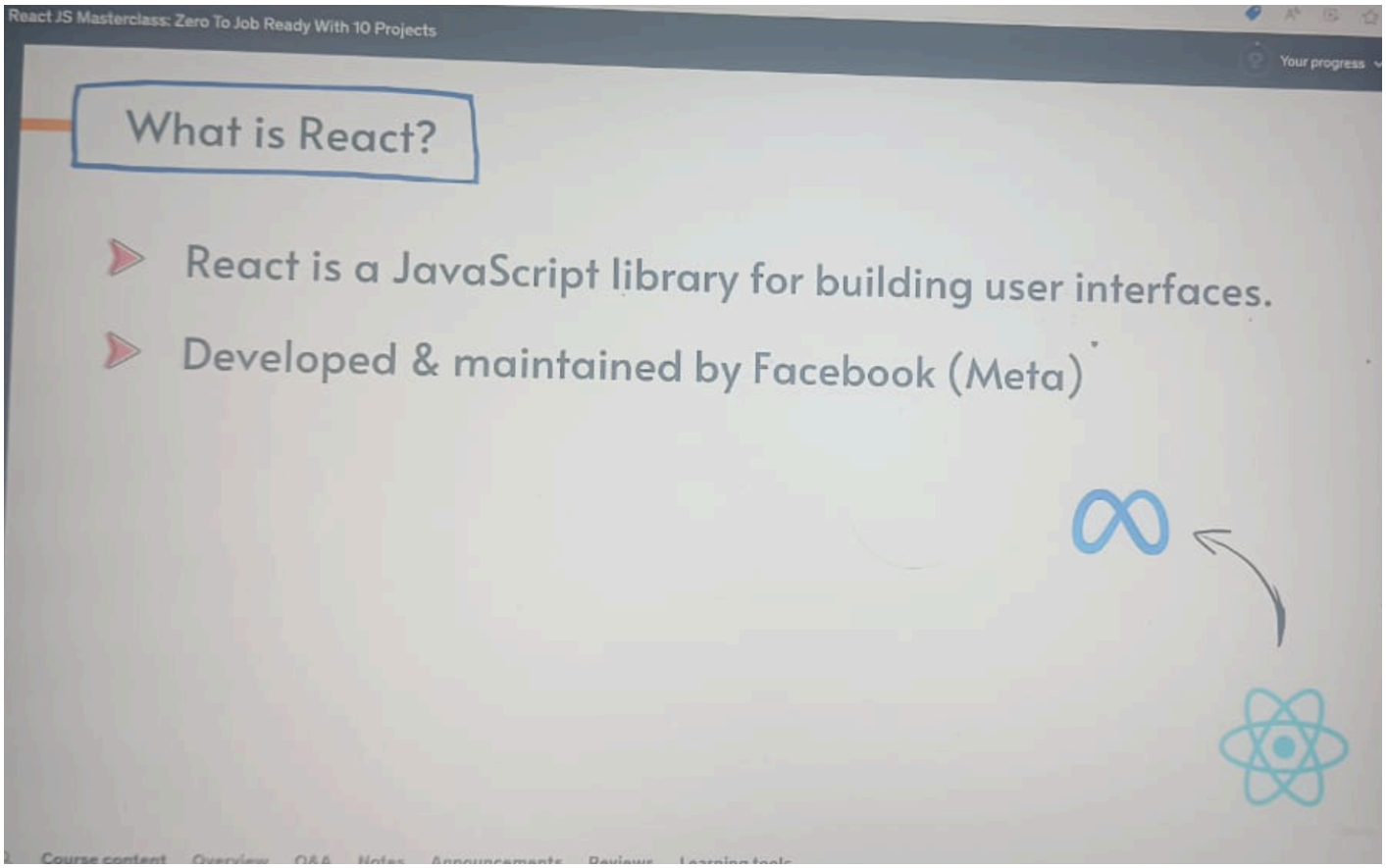
Guide:

- React Deployment (Using Git & Github): React: How To Deploy
- Tailwind CSS: Tailwind CSS - Crash Course

- Context & Reducers: Context and Reducers
- Redux Toolkit: Redux Toolkit
- React Testing: React Testing

in GitHub projects we can do **github.dev** or **github1s.com** to access the code online i had already forked the code into my github okay by creating a fork okay ..so these new things i am finding it okay ..

here in React we do SPA means single page application here only single page index will load and in that page based on selection or click of button different components are loaded here .



What is Component?

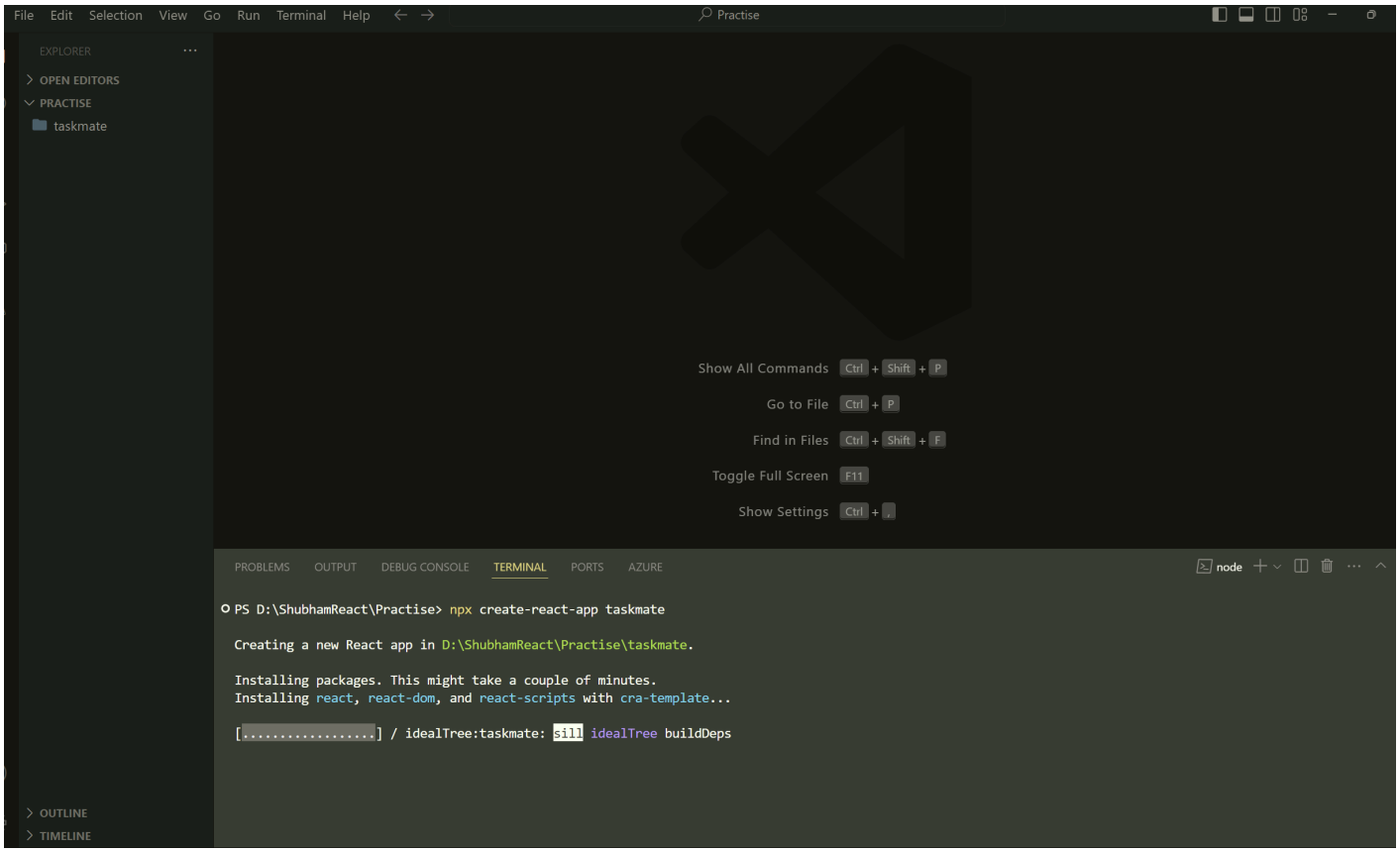
- A component is a small, reusable piece of code that defines a part of a user interface.

What is Component?

- A component is a small, reusable piece of code that defines a part of a user interface.
- They can be reused throughout an application, making it easier to build and maintain large, modular applications.
- Components can be nested inside other components to create complex user interfaces.

lets now create an app

open the terminal and do this command



suppose the website which u now created u want to run online we can go for stackblitz.com
https://stackblitz.com/ but right now as a beginner the author is not recommending it so we will use it later once we get more familiar with react

now as the app is created now first delete src folder and do this things now okay when u run it will first look for index.js file so let us delete the src folder and follow below steps to build the things from scratch okay

so in the project first create src folder now then as when running it is asking me index.js file so first add index.js file,App.js and also index.css file okay .

and respective code goes like this

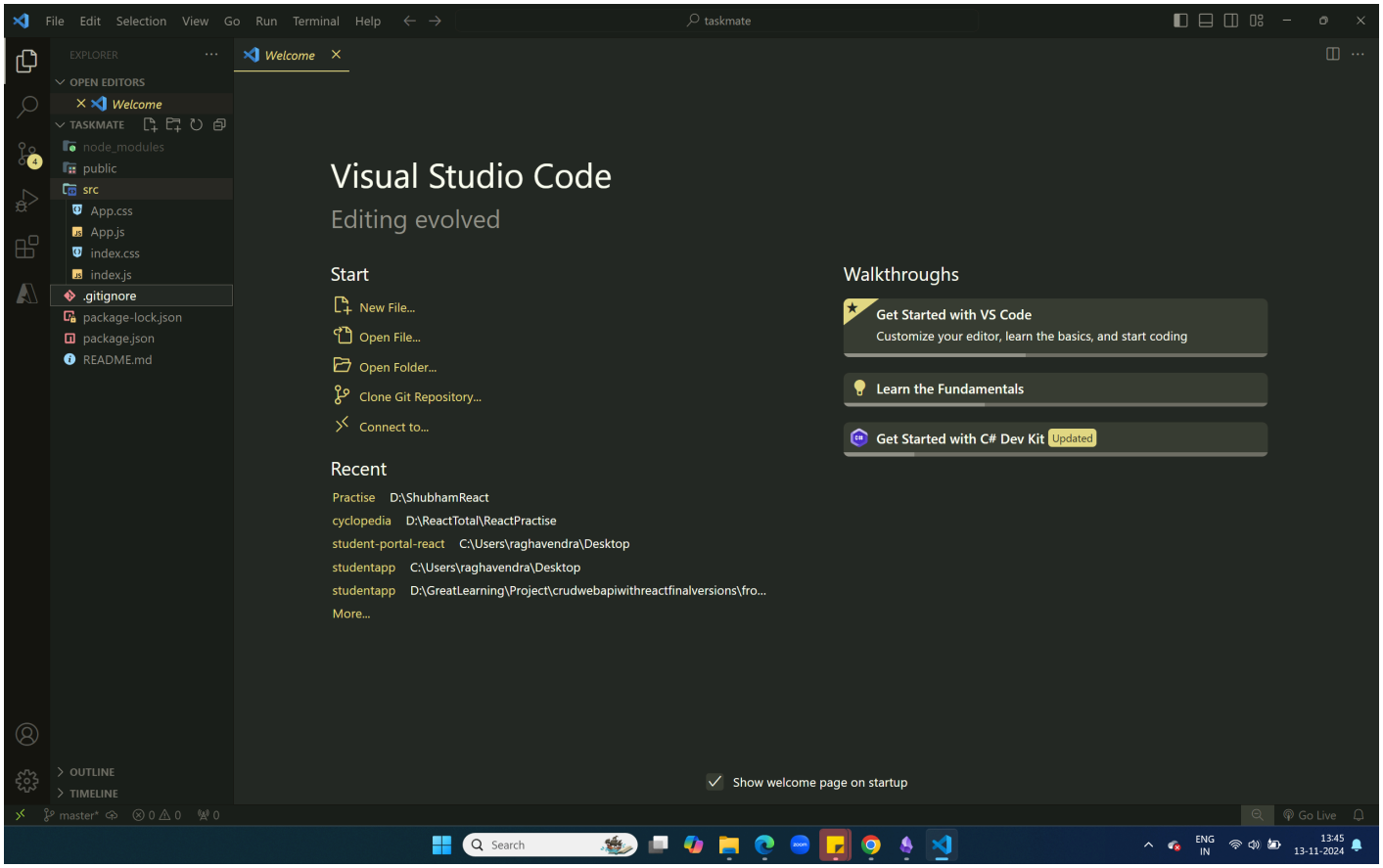
```
index.css
------------
h1{
    font-size: 100px;
}

App.js
-------
const App = () => {
    return (
        <h1>Hello!</h1>
    );
}

export default App;

index.js
---------
import React from "react";
import ReactDOM from "react-dom/client";

import App from "./App";
import "./index.css";

const root = ReactDOM.createRoot(document.getElementById("root"));
root.render(
    <React.StrictMode>
        <App />
    </React.StrictMode>
```

```
);
```

so i will get the output here as hello okay so in video it is explained how code flow is happening okay .

here App.js is a component the design code i had taken into App.js and that is getting displayed okay .

now delete the project again fully and create again new app okay now in src folder i will keep App and index files remaining i will delete



so now to avoid errors index.js and App.js file will look like this now after removing some elements from it

```
index.js
---------
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);

App.js
-------

import './App.css';

function App() {
  return (
    <div className="App">

    </div>
  );
}

export default App;
```

so you can see above How I had used simple functions when return comes in that design goes and inside the function app after the flower bracket you can declare variables also those variables can be substituted in the design also

we have short cut commands to write function code like rfc, rfce ,rafc etc. so after writing this things u can check what automatic code it will form okay

But here I want to discuss class based components so I'm using RCC so let us see how it looks

```
import './App.css';
import React, { Component } from 'react'

export default class App extends Component {
  render() {
    return (
      <div>App</div>
    )
  }
}
```

But i am not using it i am going with regular rfc only here

```
import './App.css';

export default function App() {
  return (
    <div>App</div>
  )
}
```

so we can see App as message as if now and again now remove the code in index.css and App.css so that in next lecture we can add some things over there ..

Now let us understand what is jsx here okay ..

so here whatever i am returning here is not html but it is a jsx tag .

```
import './App.css';

export default function App() {
  return (
    <div>

      <h1>Raghavendra</h1>
      <p>Lorem ipsum dolor sit amet consectetur adipisicing elit.
        Saepe dicta odio facere similique nemo perferendis dolorem, fugiat unde dolor cum illum sint numquam.
        Magni maxime voluptas, qui exercitationem odio nobis?</p>


    </div>
  )
}
```

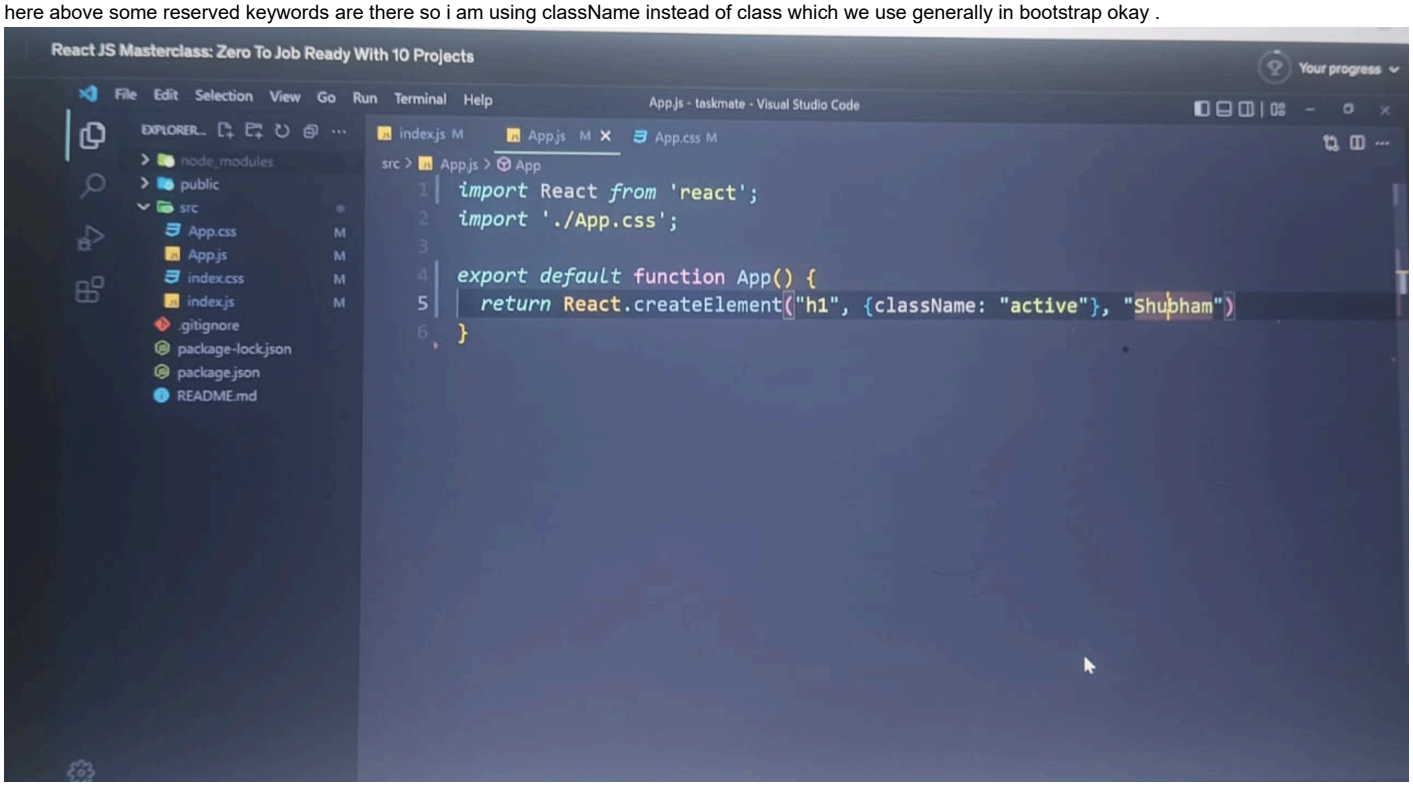
here div tag is not needed it i remove main div root tag i will get error why it is like this it is because i am using jsx here we have to follow some jsx rules here okay

here every tag need to closed that is another rule for jsx and always use **camel Notations** in programming of jsx file okay so here even though i write here js it is jsx only okay as return some code is coming here okay
change in the code of App.css and App.js is like this

```
// import React from 'react';
import './App.css';

export default function App() {
  const username = "Shubham";

  return (
    <div>
      <h1 className='active'>{username}</h1>
      <p>Lorem ipsum dolor, placeat.</p>
    </div>
```

```
  )

  // return React.createElement("h1", {className: "active"}, "Shubham")
}
```

here above some reserved keywords are there so i am using className instead of class which we use generally in bootstrap okay .



so i can write like this jsx code same output i will get so internally it does like that i am saying it okay ..

refer the below link for more information about jsx

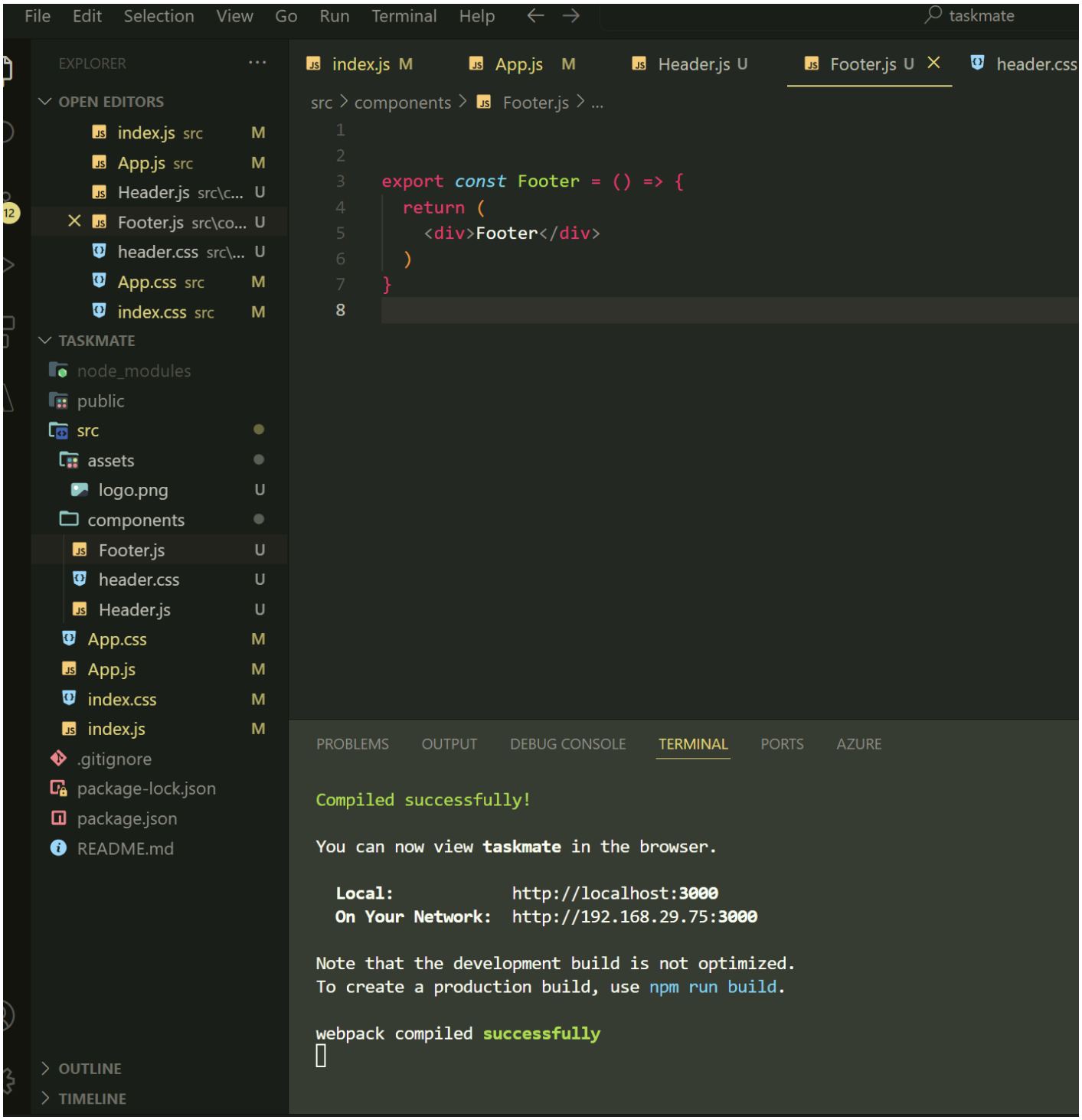
[Writing Markup with JSX – React]

Anything which I want to use in index dot html like images and all I can put that in public folder but something I want to use in my project I'll put it in src folder in assets folder I will keep images logos etc. things so in src folder I will create one folder assets and put all those things so generally this is a convention to follow by creating a folder assets

create one folder assets and folder components in src folder and in assets folder add one logo.png and in components folder add Footer,Header,header.css files and refer that in App.js so complete code of all is provided below

so i had used rafc here react arrow functional component here it means and from header and footer i am removing react namespace because they are used as sub components so there that namespace is used which is **import React from 'react'**

so folder structure



so above i can see footer code next codes is shown below and logo u can download from github of video tutorials okay

header.css

```
.logo {
    height: 80px;
}
```

Header.js

```
import Logo from "../assets/logo.png"
import "./header.css";

export const Header = () => {
  return (
    <div>
```

```
        <img className="logo" src={Logo} alt="" />
    </div>
  )
}

App.js
------
import { Header } from "./components/Header";
import { Footer } from "./components/Footer";
import './App.css';

export default function App() {
  return (
    <>
      <Header />
      <div>
        <h1>Shubham</h1>
      </div>
      <Footer />
    </>
  )
}


index.js
-----------
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```

so this is the end of section 2 and in github till 008 it is done okay ..

Section 3 (state and useState)

now again i deleted all the files and again created new app with the same name and in that new app i deleted earlier like unwanted files and kept only app and index js and CSS files into it and configured okay
but here for good look and feel in design i had added some CSS as well okay .

so complete code is like this till now

```
App.css
------------
.App{
  text-align: center;
}

.box{
  max-width: 200px;
  margin: 50px auto;
  border-radius: 5px;
  box-shadow: rgba(0, 0, 0, 0.16) 0px 1px 4px;
  padding: 10px;
}

p{
  margin-top: 20px;
  font-size: 24px;
  text-align: center;
}

button{
  padding: 10px;
  margin: 20px 10px;
```

```
  font-size: 16px;
  color: #FFFFFF;
  border: 0px;
  border-radius: 5px;
  cursor: pointer;
}

button.add{
  background-color: #18978F;
}

button.sub{
  background-color: #EB4747;
}

button.reset{
  background-color: #0F3460;
}
```

App.js

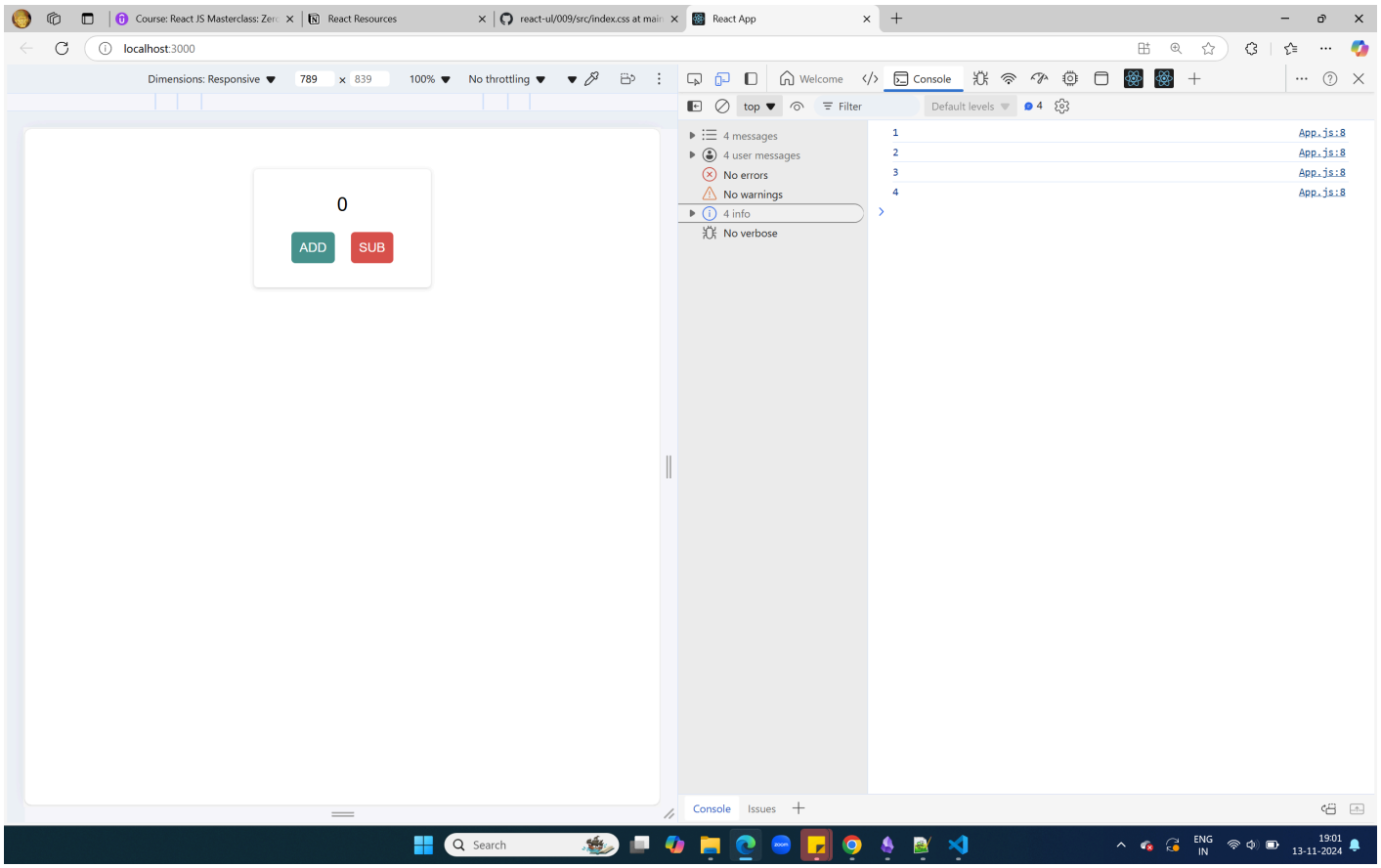
```
import './App.css';

function App() {
  let count = 0;

  function handleAdd(){
    count+=1;
    console.log(count);
  }

  return (
    <div className="App">
      <div className="box">
        <p>{count}</p>
        <button onClick={handleAdd} className='add'>ADD</button>
        <button className='sub'>SUB</button>
      </div>
    </div>
  );
}

export default App;
```

index.css

```
*{
  font-family: Arial, Helvetica, sans-serif;
  margin: 0px;
  padding: 0px;
}
```

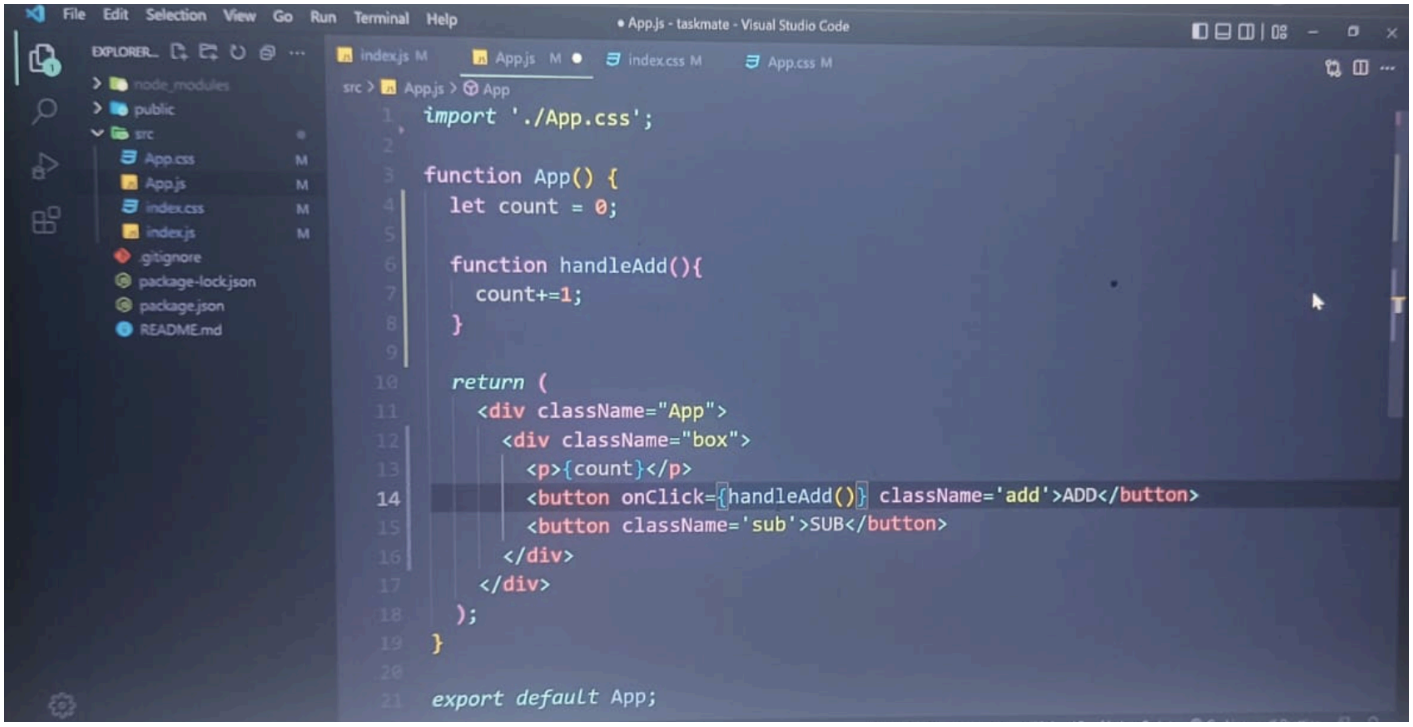
index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```

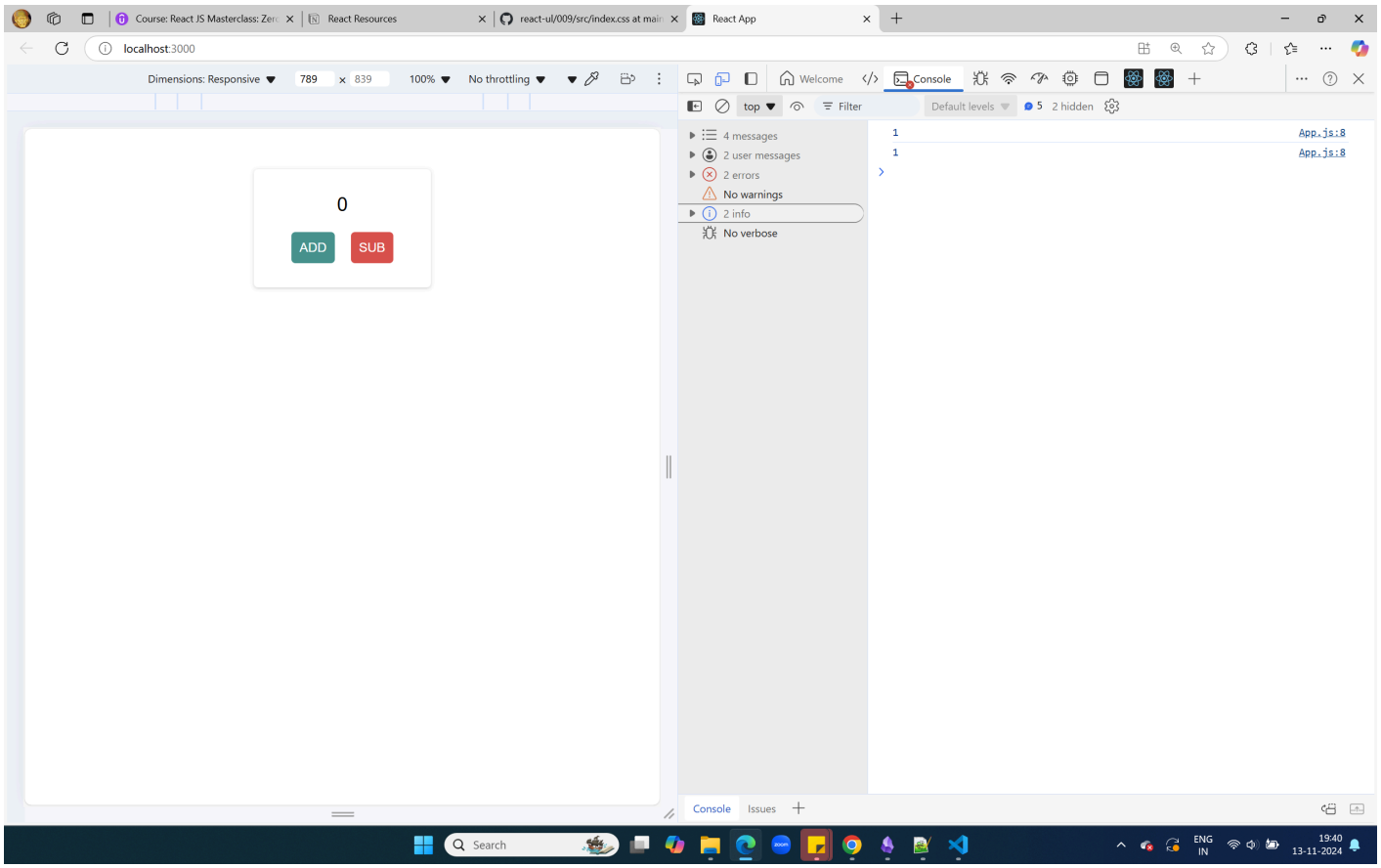


getting output like this so in console it is incrementing but in page it is not updating also not rendering

so before that i want to tell that some different ways of using App.js event handler code



when i try this code the code will execute when i load the page or when it renders means in console log only 1 and 1 is getting printed so this coding wont work in console log also



next if i use inline function then that also will show me the same correct output as first one in console log

```
import './App.css';

function App() {
  let count = 0;

  // function handleAdd(){
  //   count+=1;
  //   console.log(count);
  // }

  return (
    <div className="App">
      <div className="box">
        <p>{count}</p>
        <button onClick={() => { count += 1; console.log(count); }} className='add'>ADD</button>
        <button className='sub'>SUB</button>
      </div>
    </div>
  );
}

export default App;
```

so it is working fine the above code in console.log okay

Now as it is updating only in log not in screen for that we have to use useState so for that i am jumping to 14th video and as per code 010 i have to check so now hyperlink for documentation is provided below okay ..

useState – React

so now i am changing App.js code completely like this

```
import { useState } from 'react';
import './App.css';

function App() {
  const [count, setCount] = useState(0);
```

```
  function handleAdd(){
    setCount(count + 1);
  }

  function handleSub(){
    setCount(count - 1);
  }

  function handleReset(){
    setCount(0);
  }

  return (
    <div className="App">
      <div className="box">
        <p>{count}</p>
        <button onClick={handleAdd} className='add'>ADD</button>
        <button onClick={handleSub} className='sub'>SUB</button>
        <button onClick={handleReset} className='reset'>RESET</button>
      </div>
    </div>
  );
}

export default App;
```

now run the code and see you can check the explanation in video 14 okay ..

so i checked the program is working properly next is

now i am moving to video 15th in this what i am doing is in brughumen video when the issue of multiple change in the same function was there i used PrevState thing and now without using PrevState how to do let us see and also use PrevState in the below example

now taking the above code ...

now suppose my code is changed like this

```
import { useState } from 'react';
import './App.css';

function App() {
  const [count, setCount] = useState(0);

  function handleAdd(){
    setCount(count + 1);
    setCount(count + 1);
     setCount(count + 1);
  }

  function handleSub(){
    setCount(count - 1);
  }

  function handleReset(){
    setCount(0);
  }

  return (
    <div className="App">
      <div className="box">
        <p>{count}</p>
        <button onClick={handleAdd} className='add'>ADD</button>
        <button onClick={handleSub} className='sub'>SUB</button>
        <button onClick={handleReset} className='reset'>RESET</button>
      </div>
    </div>
  );
}

export default App;
```

here i am incrementing 3 times so when i click the button add it should show me 3 in one go

so when i checked it is not happening okay now

i will change the code like this as it is every time getting reinitialized to zero okay .means i will write an anonymous function okay

```
import { useState } from 'react';
import './App.css';

function App() {
  const [count, setCount] = useState(0);

  function handleAdd(){
    setCount(count=>count + 1);
    setCount(count=>count + 1);
     setCount(count=>count + 1);
  }

  function handleSub(){
    setCount(count - 1);
  }

  function handleReset(){
    setCount(0);
  }

  return (
    <div className="App">
      <div className="box">
        <p>{count}</p>
        <button onClick={handleAdd} className='add'>ADD</button>
        <button onClick={handleSub} className='sub'>SUB</button>
        <button onClick={handleReset} className='reset'>RESET</button>
      </div>
    </div>
  );
}

export default App;
```

now it is happening now let us do using prevState as well another way of doing it which we have done in Brughumen Patel video okay

```
import { useState } from 'react';
import './App.css';

function App() {
  const [count, setCount] = useState(0);

  function handleAdd(){
    setCount((prevCount) => prevCount + 1);
    setCount((prevCount) => prevCount + 1);
     setCount((prevCount) => prevCount + 1);
  }

  function handleSub(){
    setCount((prevCount)=>prevCount - 1);
  }

  function handleReset(){
    setCount(0);
  }

  return (
    <div className="App">
      <div className="box">
        <p>{count}</p>
        <button onClick={handleAdd} className='add'>ADD</button>
        <button onClick={handleSub} className='sub'>SUB</button>
        <button onClick={handleReset} className='reset'>RESET</button>
      </div>
    </div>
  );
}

export default App;
```

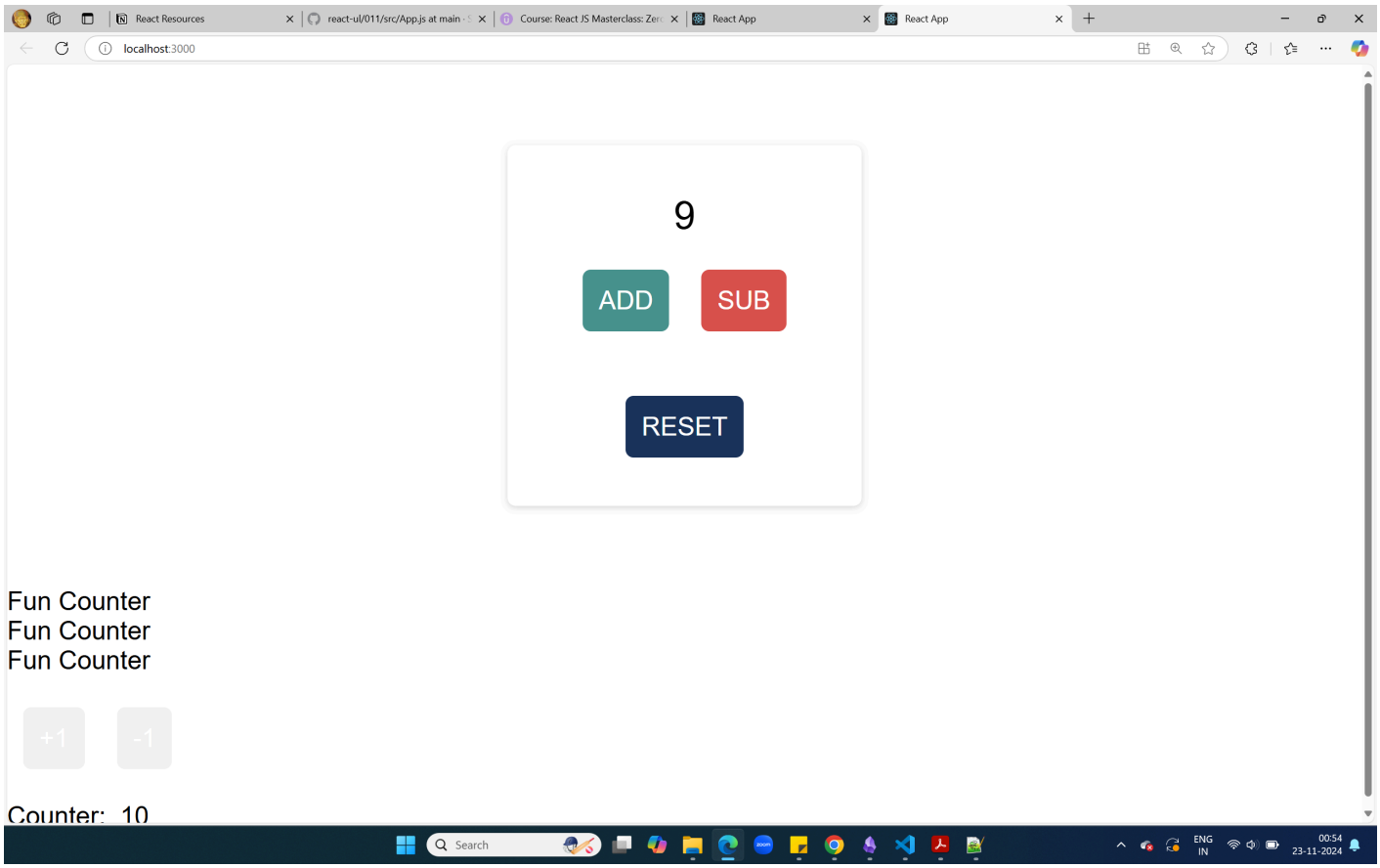
so camel notation prevCount is used in bracket which is working perfectly fine okay

now here count value is constant means it not stored in object like a name value value pair so in this case I am writing another code in App2.js file like this means i am creating a new file with the name App2.js and will call that file in index.js okay like this

in src folder only i will create this file App2.js okay

```
App2.js
----------
import { useState } from "react";

const Counter = () =>
{
    const [counterState, setCounterState] = useState(() => { return { counter: 10,title2:"Fun" }; });

    const[titleState,setTitleState]=useState("Fun")// not much used
     const[titleState1,setTitleState1]=useState(() => { return { title1: "Fun" }; })
    function incrementCounter()
    {
        setCounterState((prevState) => { return { counter:prevState.counter + 1 }; });
    }

    function decrementCounter()
    {
        setCounterState((prevState) => { return { counter: prevState.counter - 1 } });
        setCounterState((prevState)=>{ return {counter:prevState.counter -1} });
    }

        return (
        <div className="col-12 col-md-4 offset-md-4 border text-white">

                <span className="h2 pt-4 m-2 text-white-50">{titleState} Counter </span><br/>
                <span className="h2 pt-4 m-2 text-white-50">{titleState1.title1} Counter </span><br/>
                   <span className="h2 pt-4 m-2 text-white-50">{counterState.title2} Counter </span>
        <br />
        <button className="btn btn-success m-1" onClick={incrementCounter}>+1</button>
        <button className="btn btn-danger m-1" onClick={decrementCounter}>-1</button>
        <br />
        <span className="h4">
        Counter:  
                    <span className="text-success">{counterState.counter }</span>
        </span>
        </div>
            );

}

export default Counter;

index.js
-------------
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import App2 from './App2';
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>

    <App />
      <App2 />

  </React.StrictMode>
);
```

and here concentrate on usage of object and this coding i had taken from Brughumen Patel video just for extra information how fun is used here so the output is working fine



now just see how Fun is used differently here now let us move on to video 16 in video 15 we have seen multiple updates at a time okay .and in 15th video check how destructing concept is used .

now let us work on arrays getting initialize okay

Now add a new file App3.js in src folder and modify the code like this in it and in index.js only display App3.js like this okay and i have used App3.css now here index is needed to tell that this is unique value so along with index if u don't want to use index and u want to use id only for that also code is given down so just analyze the code i had written both version and for each task i had applied and used jsx and span tags used try to analyze it carefully

```
App3.js
----------
import { useState } from 'react';
import './App3.css';

function App() {
  const [tasks, setTasks] = useState([
    {id: 5271, name: "Record React Lectures", completed: true},
    {id: 7825, name: "Edit React Lectures", completed: false},
    {id: 8391, name: "Watch Lectures", completed: false}
  ]);

  return (
    <div className="App">
      <h1>Task List</h1>
      <ul>
        { tasks.map((task,index) => (
          <li key={index}>
            <span>{task.id} - {task.name}</span>
            <button className='delete'>Delete</button>
          </li>
        )) }
      </ul>
    </div>
  );
}

export default App;

without index instead of index using key as task id as it will be unique only okay
```

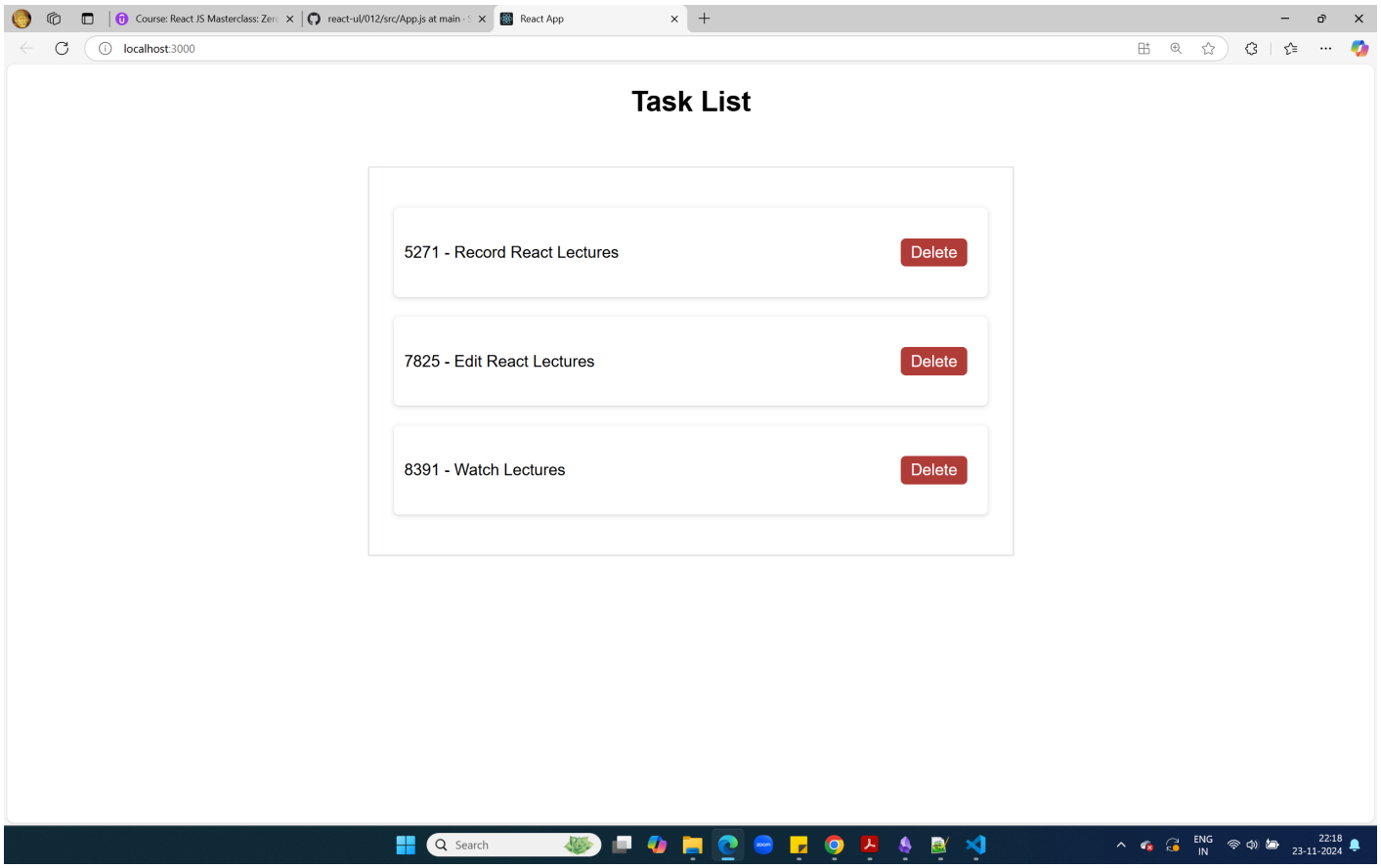
```jsx
import { useState } from 'react';
import './App3.css';

function App() {
  const [tasks, setTasks] = useState([
    {id: 5271, name: "Record React Lectures", completed: true},
    {id: 7825, name: "Edit React Lectures", completed: false},
    {id: 8391, name: "Watch Lectures", completed: false}
  ]);

  return (
    <div className="App">
      <h1>Task List</h1>
      <ul>
        { tasks.map((task) => (
          <li key={task.id}>
            <span>{task.id} - {task.name}</span>
            <button className='delete'>Delete</button>
          </li>
        )) }
      </ul>
    </div>
  );
}

export default App;
```

App3.css

```css
.App {
    text-align: center;
}

a {
    text-decoration: none;
    color: #000000;
}

h1 {
    font-size: 28px;
    text-align: center;
    margin: 20px;
}

header {
    display: flex;
    justify-content: space-between;
    align-items: center;
    height: 50px;
    align-items: center;
    max-width: 1000px;
    margin: auto;
    border-bottom: 1px solid #d7d7d7;
    font-size: 18px;
    padding: 0px 10px;
}

img {
    max-width: 40px;
}

ul {
    max-width: 600px;
    margin: 50px auto;
    padding: 20px;
    box-shadow: rgba(0, 0, 0, 0.02) 0px 1px 3px 0px, rgba(27, 31, 35, 0.15) 0px 0px 0px 1px;
}

li {
    font: 16px;
    list-style: none;
    margin: 20px 5px;
```

```
    box-shadow: rgba(0, 0, 0, 0.16) 0px 1px 4px;
    border-radius: 5px;
    padding: 10px;
    display: flex;
    justify-content: space-between;
    align-items: center;
}

li.completed {
    box-shadow: rgb(62, 150, 0) 0px 1px 4px;
}

li.incomplete {
    box-shadow: rgba(135, 20, 0, 0.689) 0px 1px 4px;
}

button.delete {
    border: 0px;
    border-radius: 5px;
    background-color: #be3434;
    color: #FFFFFF;
    padding: 5px 10px;
    cursor: pointer;
}

button.trigger {
    border: 0px;
    border-radius: 5px;
    background-color: #0F3460;
    color: #FFFFFF;
    padding: 5px 10px;
    cursor: pointer;
}

.box {
    width: 400px;
    margin: 20px auto;
    box-shadow: rgba(0, 0, 0, 0.16) 0px 1px 4px;
    padding: 20px 5px;
    border: 0px;
    border-radius: 5px;
}

.box.success {
    background-color: #baffbf;
}

.box.alert {
    background-color: #ffb4b4;
}

.box.warning {
    background-color: #FFDEB4;
}

.box p {
    margin: 10px;
}

.box p.title {
    font-size: 20px;
}

.box p.description {
    font-size: 14px;
}

.hidden {
    display: none;
}
index.js
---------
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
```

```
import App from './App';
import App2 from './App2';
import App3 from './App3';
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>

    <App3 />

    {/* <App />
       <App2 />
     */}
  </React.StrictMode>
);
```



so this is 16th video completion okay next

now in 17th video i am using delete function and i am using filter function which tells when the condition is true delete it and in the new array while rendering you have to display unmatched values okay so further code of App3.js is given below just check it

App3.js

```
import { useState } from 'react';
import './App3.css';

function App() {
  const [tasks, setTasks] = useState([
    {id: 5271, name: "Record React Lectures", completed: true},
    {id: 7825, name: "Edit React Lectures", completed: false},
    {id: 8391, name: "Watch Lectures", completed: false}
  ]);

  function handleDelete(id){
    setTasks(tasks.filter(task => task.id !== id));
  }

  return (
    <div className="App">
      <h1>Task List</h1>
```

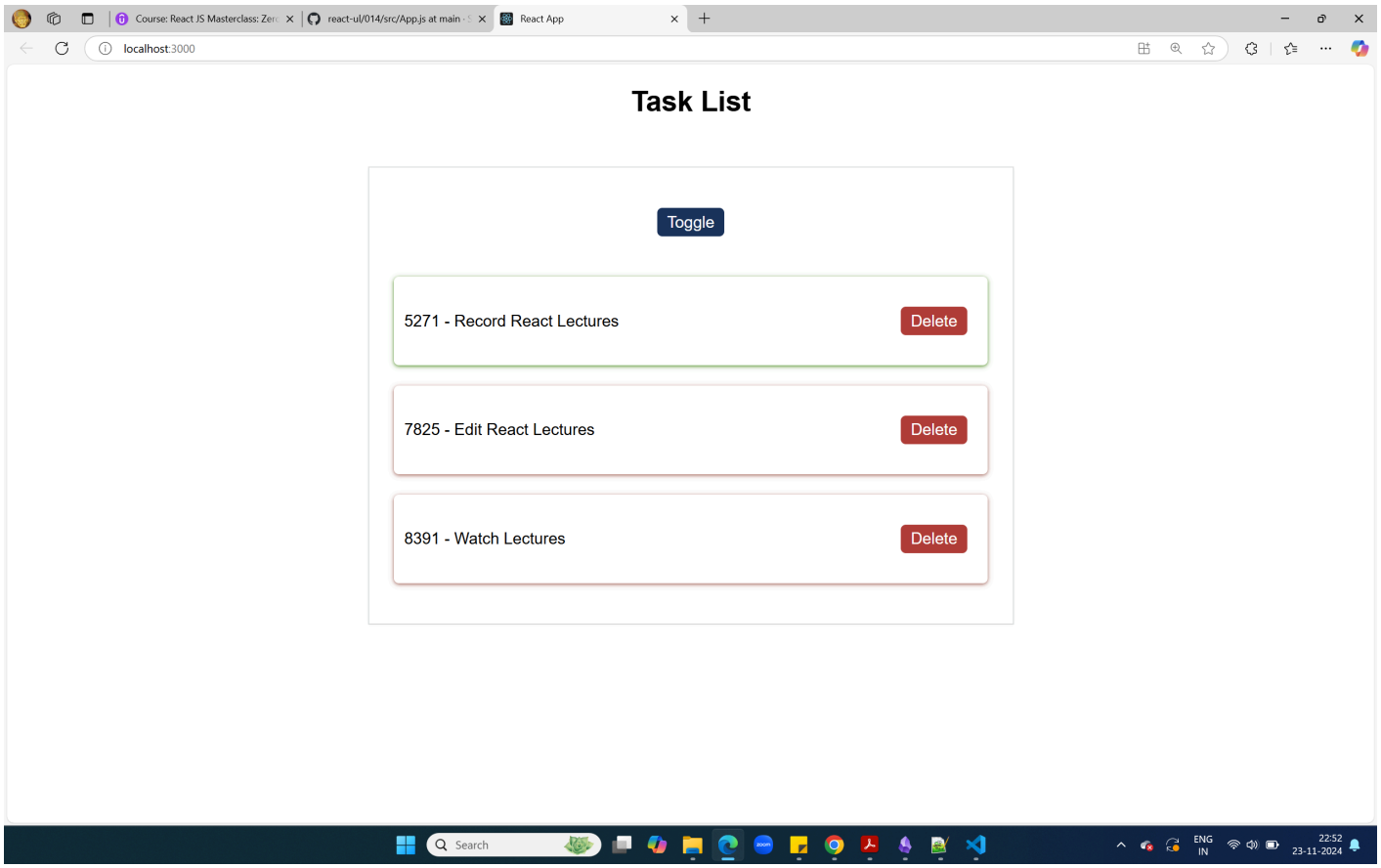
```
      <ul>
        { tasks.map((task) => (
          <li key={task.id}>
            <span>{task.id} - {task.name}</span>
            <button onClick={() => handleDelete(task.id)} className='delete'>Delete</button>
          </li>
        )) }
      </ul>
    </div>
  );
}

export default App;
```

so the coding is working fine u can see in on Click method i am using anonymous function here okay so u can see the correct usage of setTasks method which i have not used earlier okay .

so lets move to next video here now in this conditional display i will use so for this new state Show i will use okay will use button to display based on completed and not completed task okay ..I am using ternary operator also to display in green and red

App3.js

```
--------
import { useState } from 'react';
import './App3.css';

function App() {
  const [tasks, setTasks] = useState([
    {id: 5271, name: "Record React Lectures", completed: true},
    {id: 7825, name: "Edit React Lectures", completed: false},
    {id: 8391, name: "Watch Lectures", completed: false}
  ]);
  const [show, setShow] = useState(true);

  function handleDelete(id){
    setTasks(tasks.filter(task => task.id !== id));
  }

  return (
    <div className="App">
      <h1>Task List</h1>
      <ul>
        <button className='trigger' onClick={() => setShow(!show)}>Toggle</button>
        { show && tasks.map((task) => (
          <li key={task.id} className={task.completed ? "completed" : "incomplete"}>
            <span>{task.id} - {task.name}</span>
            <button onClick={() => handleDelete(task.id)} className='delete'>Delete</button>
          </li>
        )) }
      </ul>
    </div>
  );
}

export default App;
```

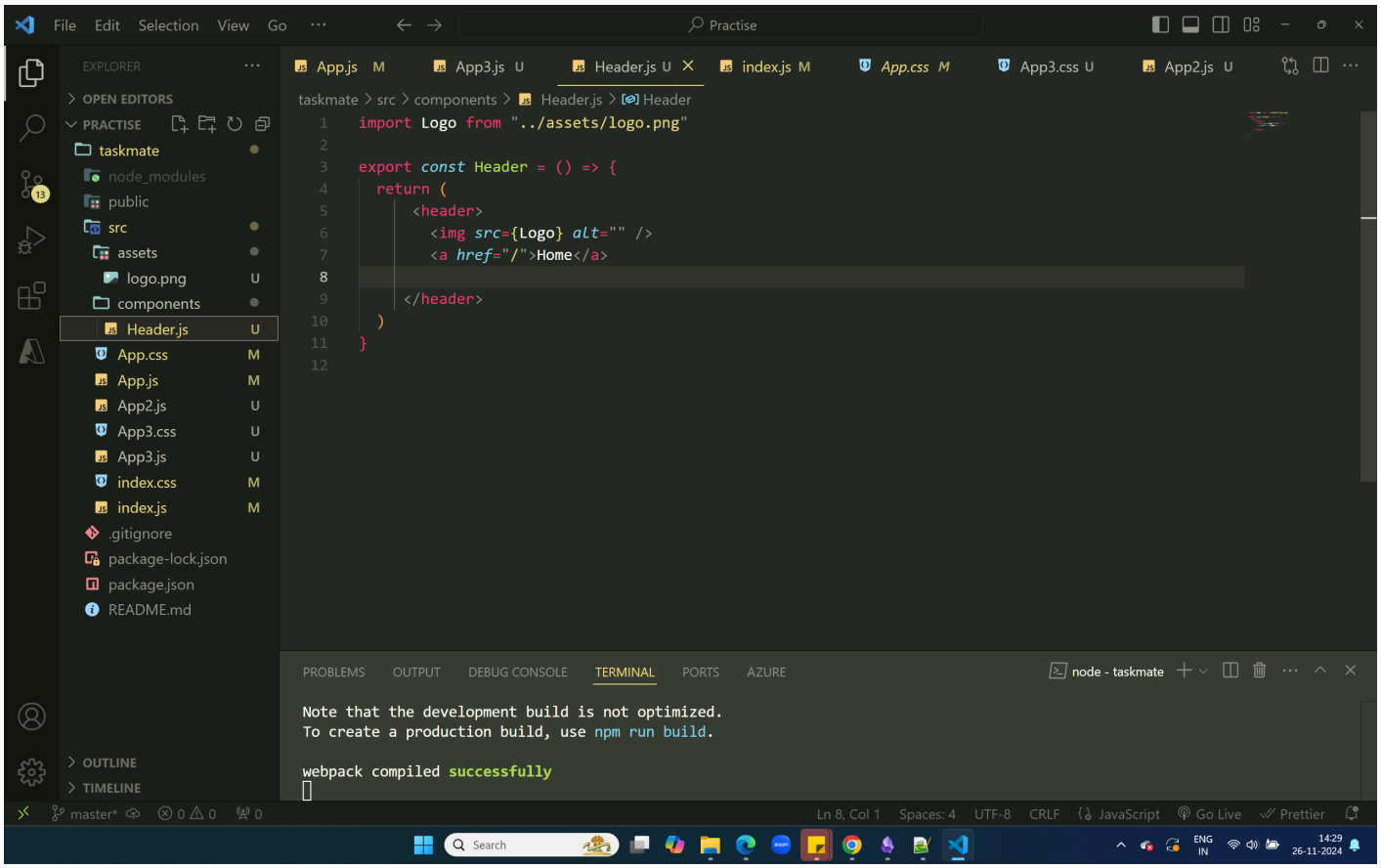


Section 4(Components and props)

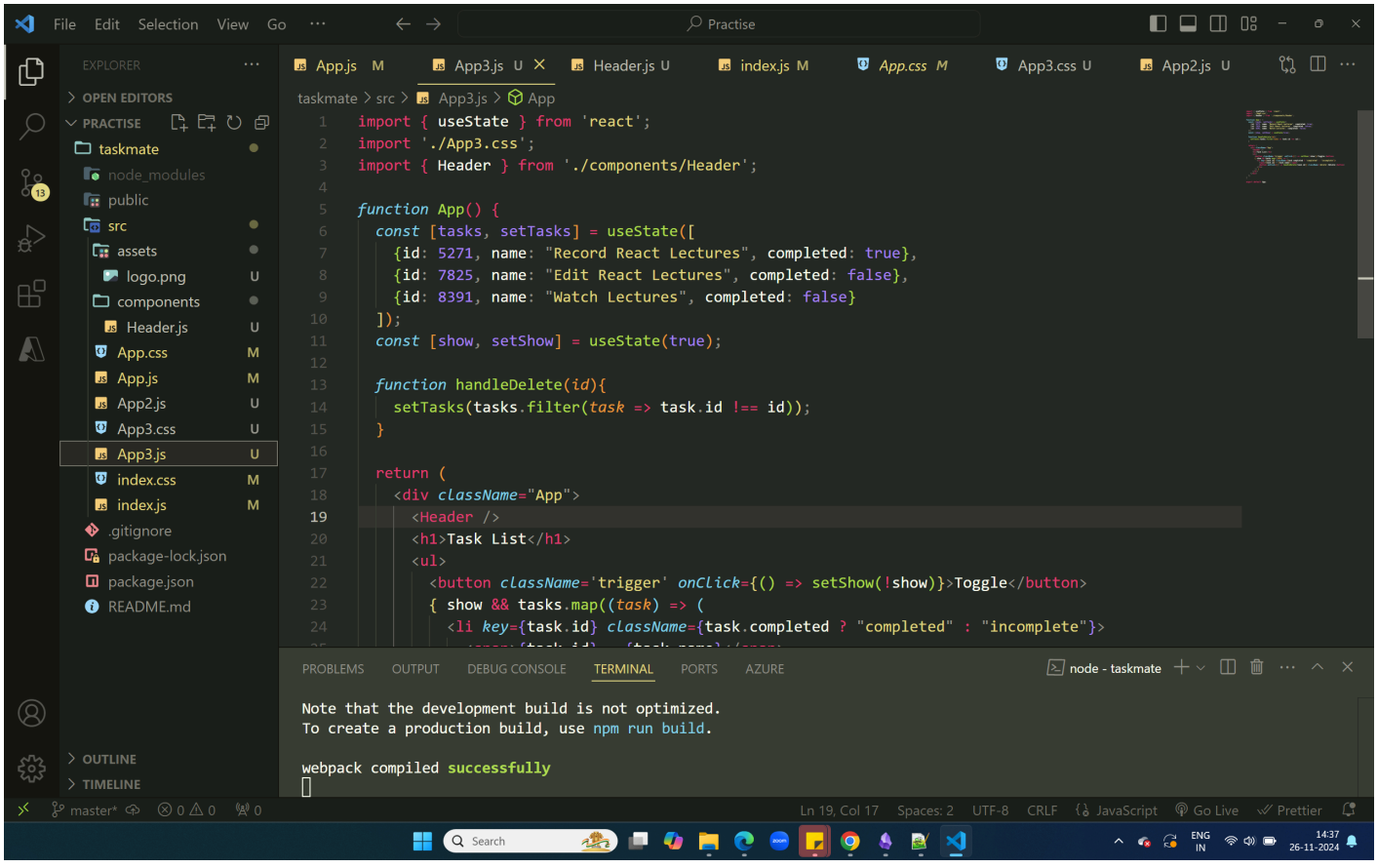
now i am having App3.js and it has lot of code so i want to add header and footer and further want to divide the App3 .js into smaller components okay ..

Here i am going to discuss about props also means from the collection i will pass one property and all sub elements will take them so all this thing we will see here

we will now create a folder in src folder components and in that Header.js file we will add
then write there rafc and remove import statement that is not needed and then in src folder only created one folder assets and in that one logo added and the complete code is here below for Header.js



Then in App3.js i am i am including that Header okay above h1 tag i had added that u can see below image



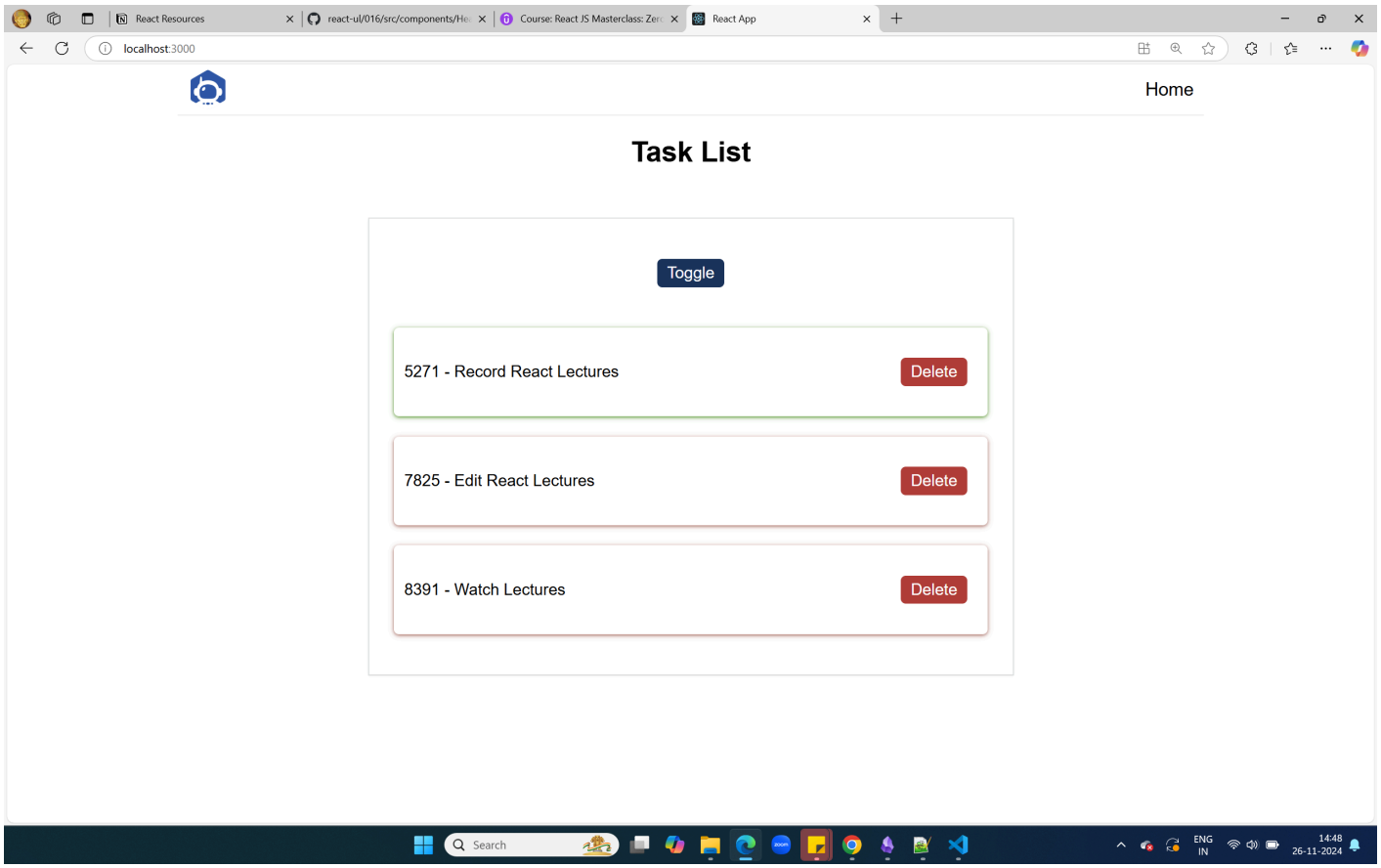
so the code till now which i discussed is

Header.js

```
import Logo from "../assets/logo.png"

export const Header = () => {
  return (
      <header>
        <img src={Logo} alt="" />
        <a href="/">Home</a>

     </header>
  )
}
```

App3.js

```
import { useState } from 'react';
import './App3.css';
import { Header } from './components/Header';

function App() {
  const [tasks, setTasks] = useState([
    {id: 5271, name: "Record React Lectures", completed: true},
    {id: 7825, name: "Edit React Lectures", completed: false},
    {id: 8391, name: "Watch Lectures", completed: false}
  ]);
  const [show, setShow] = useState(true);

  function handleDelete(id){
    setTasks(tasks.filter(task => task.id !== id));
  }

  return (
    <div className="App">
      <Header />
      <h1>Task List</h1>
      <ul>
        <button className='trigger' onClick={() => setShow(!show)}>Toggle</button>
        { show && tasks.map((task) => (
          <li key={task.id} className={task.completed ? "completed" : "incomplete"}>
            <span>{task.id} - {task.name}</span>
            <button onClick={() => handleDelete(task.id)} className='delete'>Delete</button>
          </li>
        )) }
      </ul>
    </div>
  );
}

export default App;
```



so you can see the header there on the top next

now create a TaskList.js component and in that pass all coding of App3.js which is needed mainly the collection code into it okay type there again rafc and remove import and in return paste the code from App3.js which is provided below and again i should not see this code of below in App3.js and in TaskList.js put it in root tag to avoid errors and do necessary imports okay in App3.js

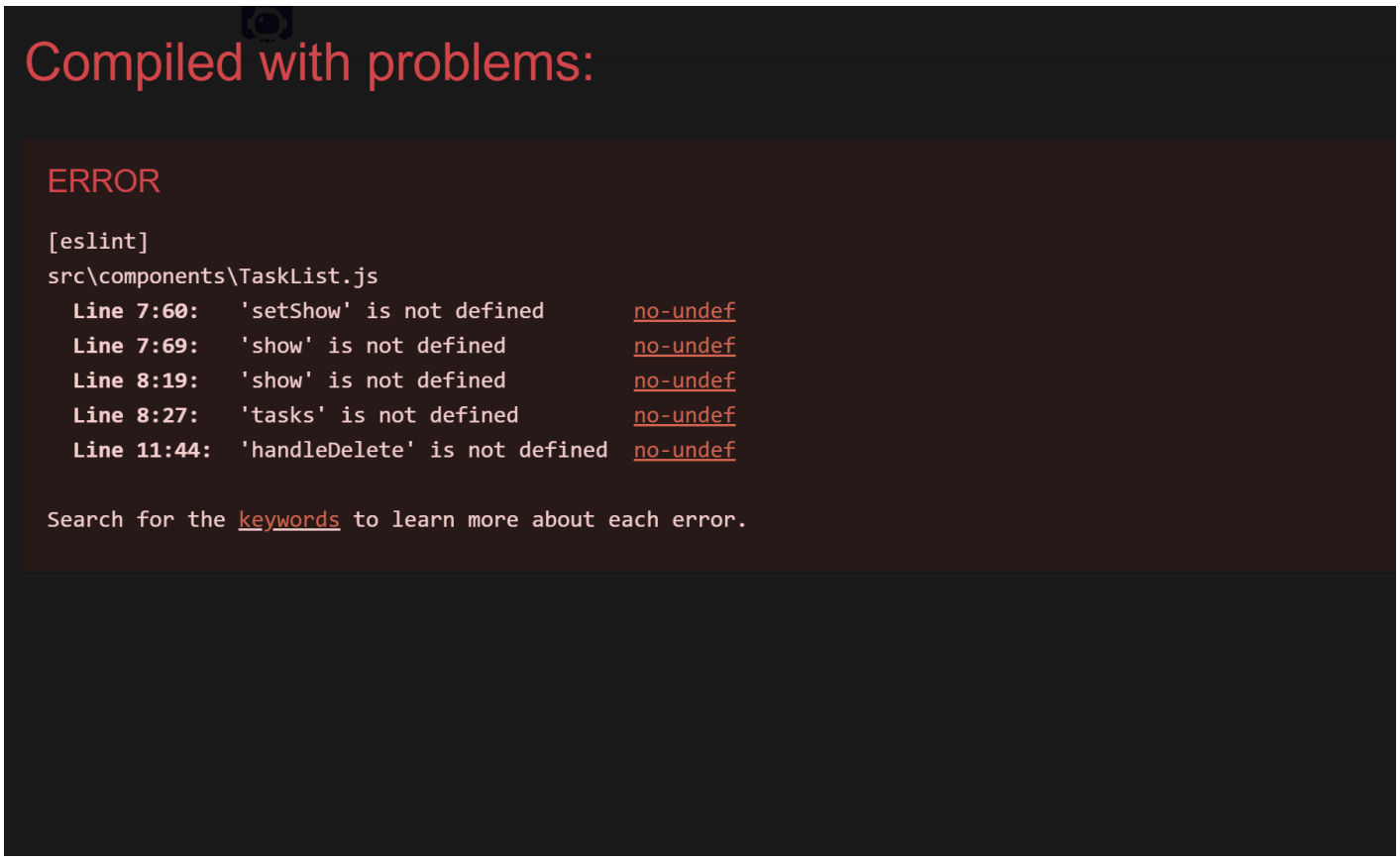
```
<h1>Task List</h1>
      <ul>
        <button className='trigger' onClick={() => setShow(!show)}>Toggle</button>
        { show && tasks.map((task) => (
          <li key={task.id} className={task.completed ? "completed" : "incomplete"}>
            <span>{task.id} - {task.name}</span>
            <button onClick={() => handleDelete(task.id)} className='delete'>Delete</button>
          </li>
        )) }
      </ul>
```

so now complete coding looks like this after doing changes of what we have told okay above okay

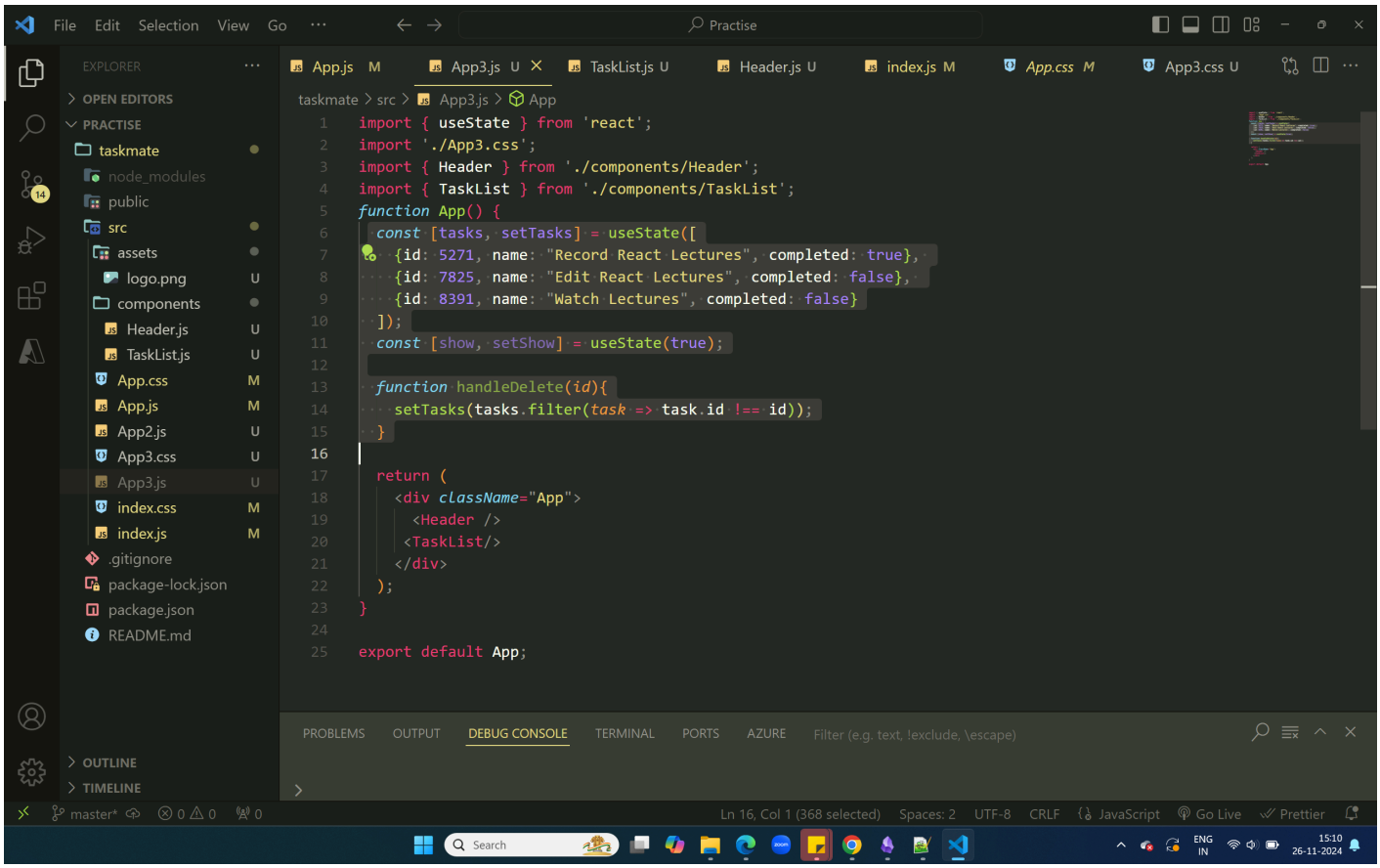
```
App3.js
-----------

import { useState } from 'react';
import './App3.css';
import { Header } from './components/Header';
import { TaskList } from './components/TaskList';
function App() {
  const [tasks, setTasks] = useState([
    {id: 5271, name: "Record React Lectures", completed: true},
    {id: 7825, name: "Edit React Lectures", completed: false},
    {id: 8391, name: "Watch Lectures", completed: false}
  ]);
  const [show, setShow] = useState(true);

  function handleDelete(id){
    setTasks(tasks.filter(task => task.id !== id));
  }
```

```
  return (
    <div className="App">
      <Header />
     <TaskList/>
    </div>
  );
}

export default App;

TaskList.js
----------

export const TaskList = () => {
  return (
      <>
            <h1>Task List</h1>
            <ul>
                <button className='trigger' onClick={() => setShow(!show)}>Toggle</button>
                { show && tasks.map((task) => (
                <li key={task.id} className={task.completed ? "completed" : "incomplete"}>
                    <span>{task.id} - {task.name}</span>
                    <button onClick={() => handleDelete(task.id)} className='delete'>Delete</button>
                </li>
                )) }
            </ul>
      </>
  )
}
```

so still i am getting some error telling me



so some more code i need to paste from App3 to TaskList.js now okay



so take the highlighted code and paste it again in Task List now okay

updated codes are like this now

TaskList.js

```
import { useState } from "react";

export const TaskList = () => {

        const [tasks, setTasks] = useState([
                {id: 5271, name: "Record React Lectures", completed: true},
                {id: 7825, name: "Edit React Lectures", completed: false},
                {id: 8391, name: "Watch Lectures", completed: false}
                                        ]);
        const [show, setShow] = useState(true);

        function handleDelete(id){
            setTasks(tasks.filter(task => task.id !== id));
        }
  return (
      <>
            <h1>Task List</h1>
            <ul>
                <button className='trigger' onClick={() => setShow(!show)}>Toggle</button>
                { show && tasks.map((task) => (
                <li key={task.id} className={task.completed ? "completed" : "incomplete"}>
                    <span>{task.id} - {task.name}</span>
                    <button onClick={() => handleDelete(task.id)} className='delete'>Delete</button>
                </li>
                )) }
            </ul>
      </>
  )
}
```

App3.js

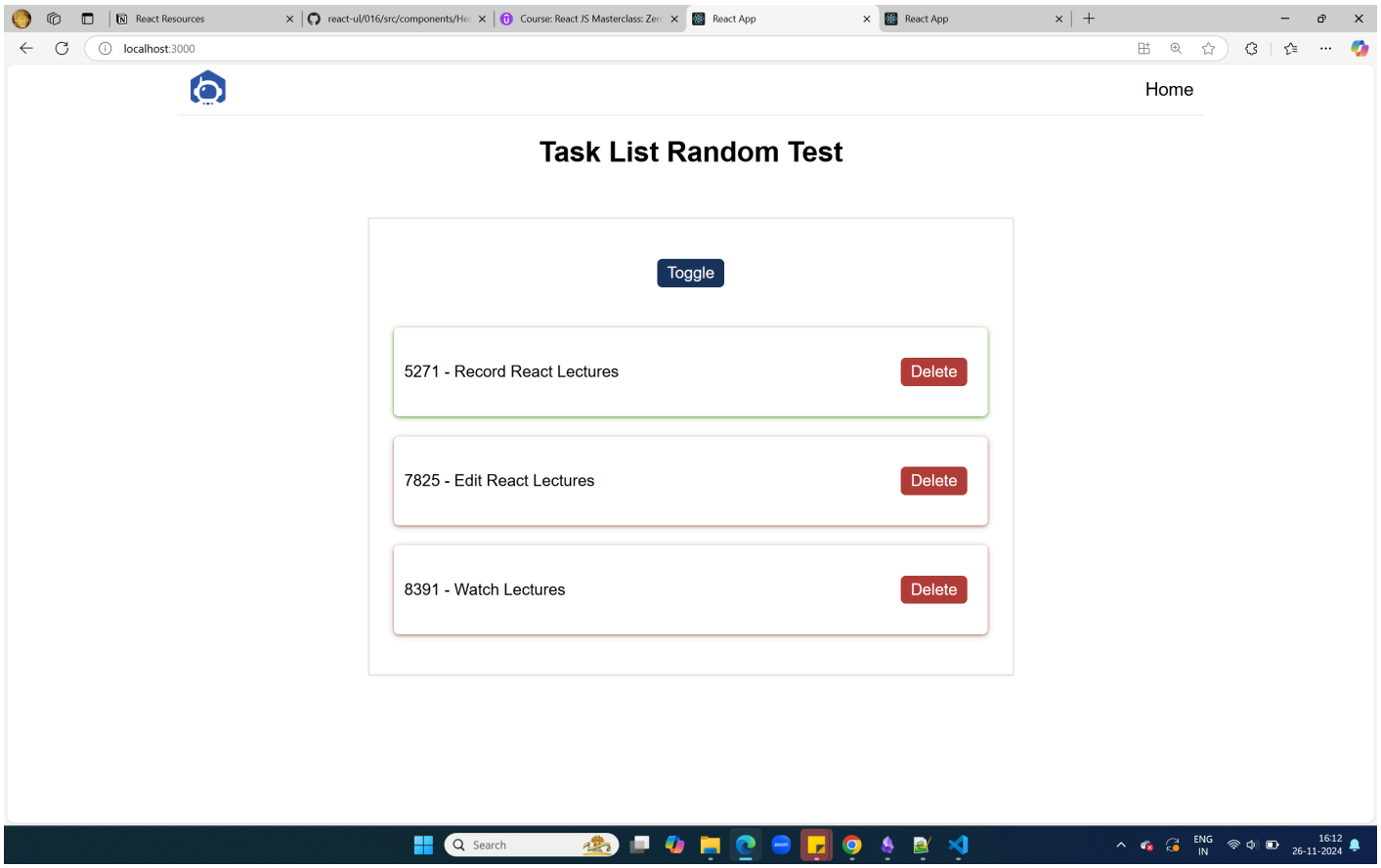
```
import './App3.css';
```

```
import { Header } from './components/Header';
import { TaskList } from './components/TaskList';
function App() {

  return (
    <div className="App">
      <Header />
     <TaskList/>
    </div>
  );
}

export default App;
```

so code is fine and it is working and remaining all functionalities which was working like delete and after deleting refreshing all are working fine till now next

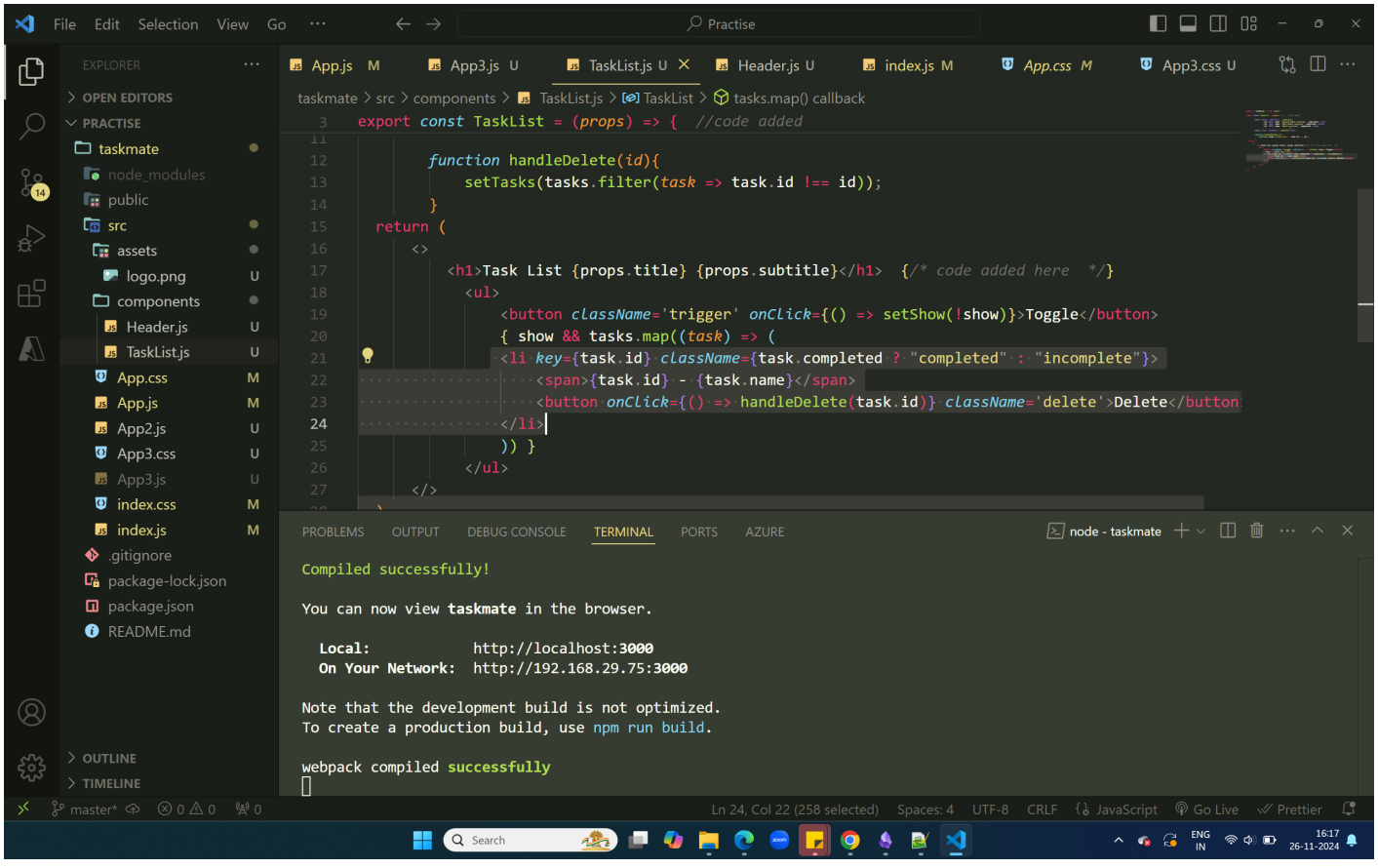
now i want to pass properties from one component to another here what i will do is like this so u can see here i had added code added here so here from App3 two properties i had added to TaskList.js and then going there accessing using props and where u want to show it is in h1 tag okay .

```
App3.js
----------

import './App3.css';
import { Header } from './components/Header';
import { TaskList } from './components/TaskList';
function App() {

  return (
    <div className="App">
      <Header />
     <TaskList title="Random" subtitle="Test" />  {/* code added here */}
    </div>
  );
}

export default App;

TaskList.js
--------------
import { useState } from "react";

export const TaskList = (props) => {  //code added

        const [tasks, setTasks] = useState([
                {id: 5271, name: "Record React Lectures", completed: true},
                {id: 7825, name: "Edit React Lectures", completed: false},
                {id: 8391, name: "Watch Lectures", completed: false}
                                        ]);
        const [show, setShow] = useState(true);

        function handleDelete(id){
            setTasks(tasks.filter(task => task.id !== id));
        }
  return (
      <>
       <h1>Task List {props.title} {props.subtitle}</h1>  {/* code added here */}
            <ul>
                <button className='trigger' onClick={() => setShow(!show)}>Toggle</button>
                { show && tasks.map((task) => (
                <li key={task.id} className={task.completed ? "completed" : "incomplete"}>
                    <span>{task.id} - {task.name}</span>
                    <button onClick={() => handleDelete(task.id)} className='delete'>Delete</button>
                </li>
                )) }
            </ul>
      </>
  )
}
```



so here u can see in heading two properties Random and Test is getting displayed more explanation is not needed i am deleting complete code and going to write the fresh code this is just for your understanding okay

Now what i will do is each individual task is represented by a card okay .

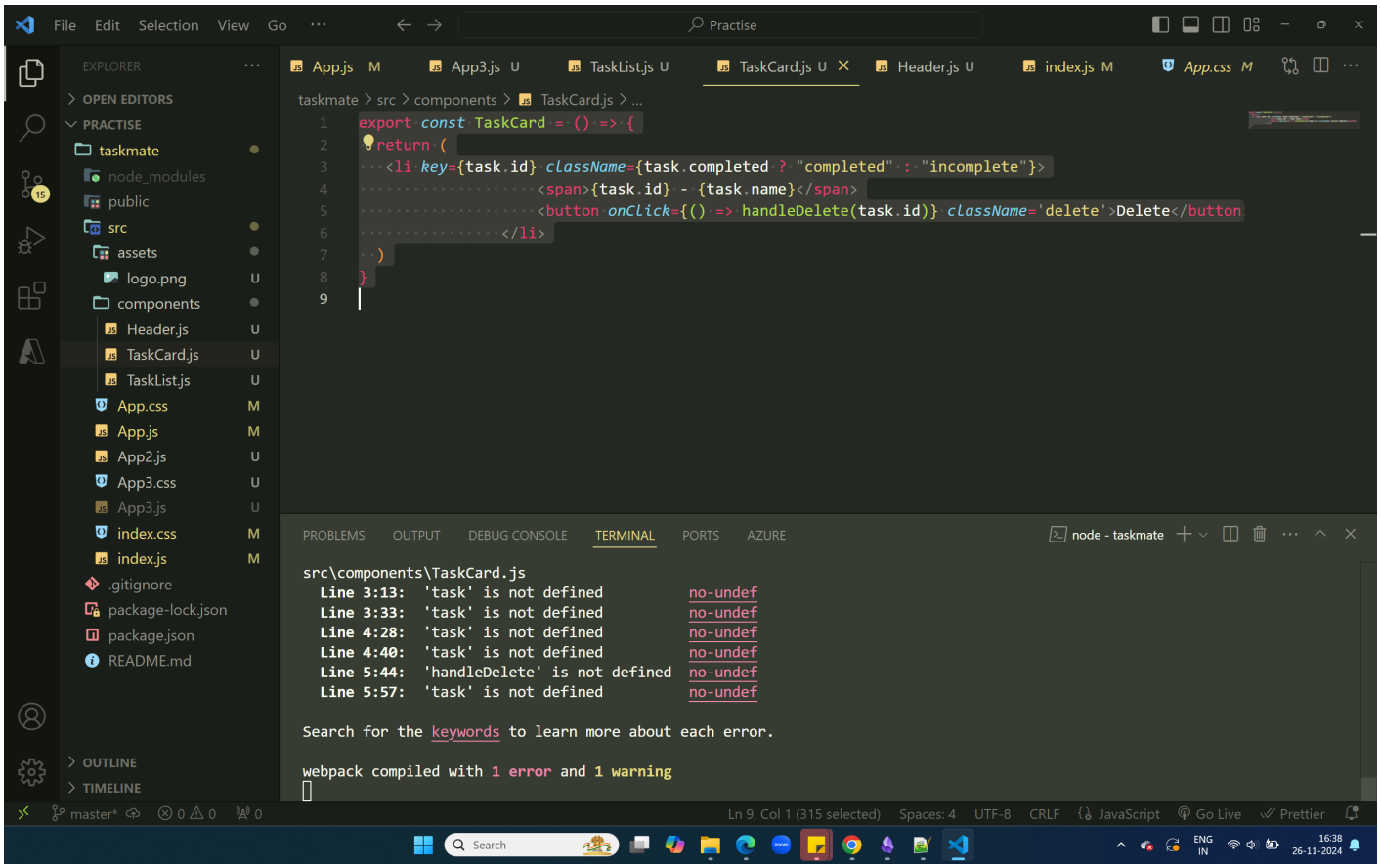


so i am talking about high lighted thing in the image okay .

now add a new file TaskCard.js and do rafc and remove imported statement and paste the highlighted code into TaskCard.js

```
TaskList.js
---------------

import { useState } from "react";
import { TaskCard } from "./TaskCard";
export const TaskList = (props) => {  //code added

        const [tasks, setTasks] = useState([
                {id: 5271, name: "Record React Lectures", completed: true},
                {id: 7825, name: "Edit React Lectures", completed: false},
                {id: 8391, name: "Watch Lectures", completed: false}
                                        ]);
        const [show, setShow] = useState(true);

        function handleDelete(id){
            setTasks(tasks.filter(task => task.id !== id));
        }
  return (
      <>
          <h1>Task List {props.title} {props.subtitle}</h1>  {/* code added here  */}
            <ul>
                <button className='trigger' onClick={() => setShow(!show)}>Toggle</button>
                { show && tasks.map((task) => (
                <TaskCard />
                )) }
            </ul>
      </>
  )
}

TaskCard.js
---------
export const TaskCard = () => {
  return (
   <li key={task.id} className={task.completed ? "completed" : "incomplete"}>
                    <span>{task.id} - {task.name}</span>
                    <button onClick={() => handleDelete(task.id)} className='delete'>Delete</button>
                </li>
  )
}
```

I am getting some errors like this



so what to do now so now again is that will use props like earlier but now again here this props how i am sending it from parent to child component now i have to send from TaskList.js to TaskCard.js okay .

See here now i am sending the object you know there task object is going same is send as props here which will help me to take out properties from the object and display it at respective locations so complete code is here .

so here again delete code is left out for that in props i can pass function also so here task was dynamic value and i can pass function value and i can pass hardcoded values like in earlier video i had done okay .

so now all code i am putting below for your reference okay ..

TaskList.js

```
---------------

import { useState } from "react";
import { TaskCard } from "./TaskCard";
export const TaskList = (props) => {

        const [tasks, setTasks] = useState([
                {id: 5271, name: "Record React Lectures", completed: true},
                {id: 7825, name: "Edit React Lectures", completed: false},
                {id: 8391, name: "Watch Lectures", completed: false}
                                        ]);
        const [show, setShow] = useState(true);

        function handleDelete(id){
            setTasks(tasks.filter(task => task.id !== id));
        }
  return (
      <>
          <h1>Task List {props.title} {props.subtitle}</h1>
            <ul>
                <button className='trigger' onClick={() => setShow(!show)}>Toggle</button>
                { show && tasks.map((task) => (
                    <TaskCard task={task} handleDelete={handleDelete} />
                )) }
            </ul>
      </>
  )
```

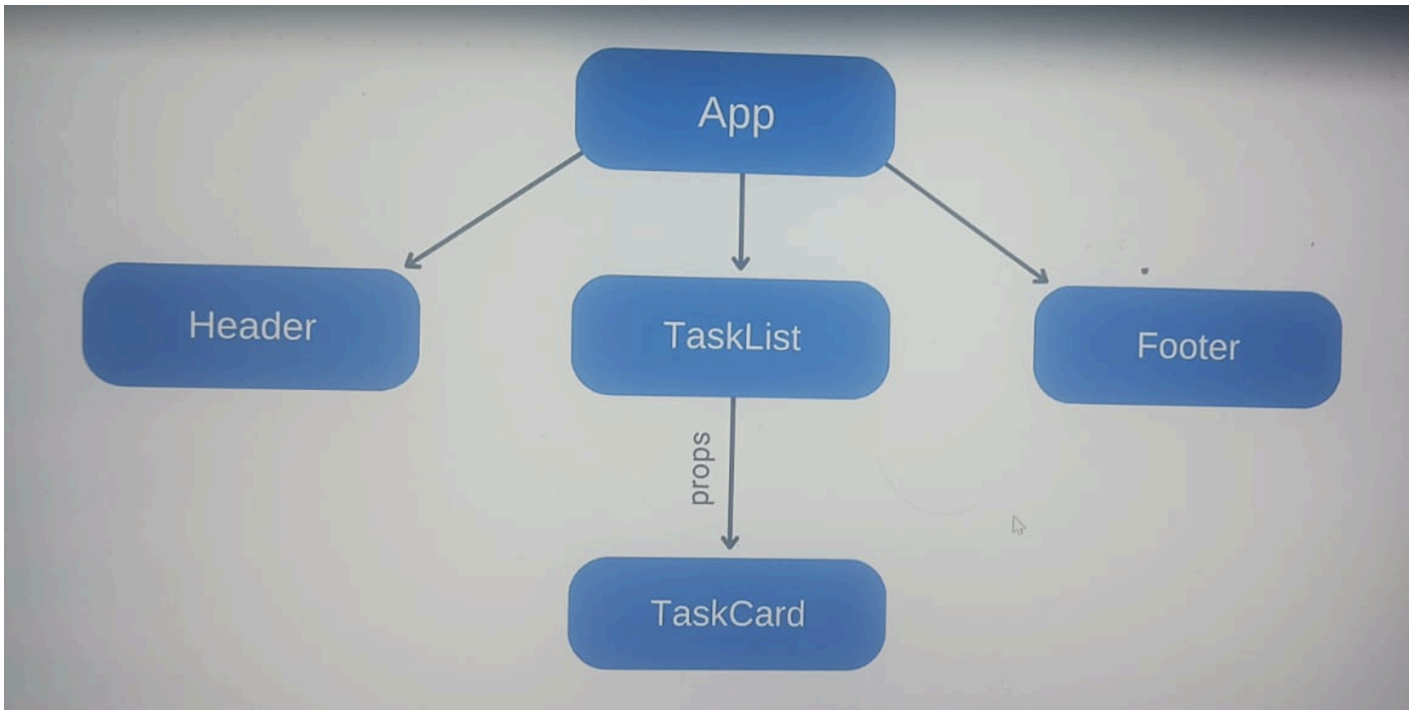
```
}


TaskCard.js
---------
export const TaskCard = (props) => {
  return (
        <li key={props.task.id} className={props.task.completed ? "completed" : "incomplete"}>
                    <span>{props.task.id} - {props.task.name}</span>
                    <button onClick={() =>props.handleDelete(props.task.id)} className='delete'>Delete</button>
        </li>
  )
}
```

so for above coding all the things are working fine .so inside the props we have this value of object task .

so this using of props again and again if u don't want to do it then you can do DE structure of object now and key i had removed from TaskCard.js and kept it in TaskList.js so both coding are provided below which is working perfectly fine ..

```
TaskList.js
---------------
import { useState } from "react";
import { TaskCard } from "./TaskCard";
export const TaskList = (props) => {  //code added

        const [tasks, setTasks] = useState([
                {id: 5271, name: "Record React Lectures", completed: true},
                {id: 7825, name: "Edit React Lectures", completed: false},
                {id: 8391, name: "Watch Lectures", completed: false}
                                        ]);
        const [show, setShow] = useState(true);

        function handleDelete(id){
            setTasks(tasks.filter(task => task.id !== id));
        }
  return (
      <>
          <h1>Task List {props.title} {props.subtitle}</h1>  {/* code added here  */}
            <ul>
                <button className='trigger' onClick={() => setShow(!show)}>Toggle</button>
                { show && tasks.map((task) => (
                    <TaskCard  key={task.id} task={task} handleDelete={handleDelete} />
                )) }
            </ul>
      </>
  )
}



TaskCard.js
---------
export const TaskCard = ({ task, handleDelete }) => {
  return (
        <li className={task.completed ? "completed" : "incomplete"}>
                    <span>{task.id} - {task.name}</span>
                    <button onClick={() =>handleDelete(task.id)} className='delete'>Delete</button>
        </li>
  )
}
```

Here anything you can convert into reusable component so right now the tree structure is like this



so in this footer coding and design is pending which we will do it later .

so here after the ul tag

```
TaskList.js
---------------
import { useState } from "react";
import { TaskCard } from "./TaskCard";
export const TaskList = (props) => {  //code added

        const [tasks, setTasks] = useState([
                {id: 5271, name: "Record React Lectures", completed: true},
                {id: 7825, name: "Edit React Lectures", completed: false},
                {id: 8391, name: "Watch Lectures", completed: false}
                                        ]);
        const [show, setShow] = useState(true);

        function handleDelete(id){
            setTasks(tasks.filter(task => task.id !== id));
        }
  return (
      <>
          <h1>Task List {props.title} {props.subtitle}</h1>  {/* code added here  */}
            <ul>
                <button className='trigger' onClick={() => setShow(!show)}>Toggle</button>
                { show && tasks.map((task) => (
                    <TaskCard  key={task.id} task={task} handleDelete={handleDelete} />
                )) }
            </ul>
            <div className="box success">
                <p className="title">Lorem ipsum dolor sit amet.</p>
                <p className="description">Lorem ipsum dolor sit amet consectetur adipisicing elit. Repudiandae porro optio illum
doloremque distinctio officiis ab maxime veniam suscipit perspiciatis!</p>
            </div>
             <div className="box warning">
                <p className="title">Lorem ipsum dolor sit amet.</p>
                <p className="description">Lorem ipsum dolor sit amet consectetur adipisicing elit. Repudiandae porro optio illum
doloremque distinctio officiis ab maxime veniam suscipit perspiciatis!</p>
            </div>
             <div className="box alert">
                <p className="title">Lorem ipsum dolor sit amet.</p>
                <p className="description">Lorem ipsum dolor sit, amet consectetur adipisicing elit. Aut quaerat, velit officia
exercitationem ut, facere ducimus, consequuntur quo libero enim repudiandae reprehenderit maiores ipsam voluptatum laudantium illo ab
sapiente aperiam voluptatem pariatur et debitis ipsa unde. Maiores assumenda praesentium tempora?</p>
            </div>
      </>
```

```
  )
}
```

so for this highlighted code or the code after ul means talking about 3 div tags in the end i am doing like this

div.box and press enter

and p.title and press enter

and p.description and press enter

and providing different class names so

so now this whole thing i want to convert into one components and want to use parent and child relationsip okay

and then created one file BoxCard.js and rafc and remove import and then modified the code like this then

```
export const BoxCard = () => {
  return (
      <div className="box">


    </div>
  )
}
```

then again import it like this and use this BoxCard like this you can see how i had used for first card which is same as success so you can do the same thing for other also for your understanding i had kept like this okay

```
TaskList.js
---------------
import { useState } from "react";
import { TaskCard } from "./TaskCard";
import { BoxCard } from "./BoxCard";
export const TaskList = (props) => {  //code added

        const [tasks, setTasks] = useState([
                {id: 5271, name: "Record React Lectures", completed: true},
                {id: 7825, name: "Edit React Lectures", completed: false},
                {id: 8391, name: "Watch Lectures", completed: false}
                                        ]);
        const [show, setShow] = useState(true);

        function handleDelete(id){
            setTasks(tasks.filter(task => task.id !== id));
        }
  return (
      <>
          <h1>Task List {props.title} {props.subtitle}</h1>  {/* code added here  */}
            <ul>
                <button className='trigger' onClick={() => setShow(!show)}>Toggle</button>
                { show && tasks.map((task) => (
                    <TaskCard  key={task.id} task={task} handleDelete={handleDelete} />
                )) }
            </ul>
            <BoxCard Result="success">
                 <p className="title">Lorem ipsum dolor sit amet.</p>
                <p className="description">Lorem ipsum dolor sit amet consectetur adipisicing elit. Repudiandae porro optio illum
doloremque distinctio officiis ab maxime veniam suscipit perspiciatis!</p>
            </BoxCard>
            <div className="box success">
                <p className="title">Lorem ipsum dolor sit amet.</p>
                <p className="description">Lorem ipsum dolor sit amet consectetur adipisicing elit. Repudiandae porro optio illum
doloremque distinctio officiis ab maxime veniam suscipit perspiciatis!</p>
            </div>
             <div className="box warning">
                <p className="title">Lorem ipsum dolor sit amet.</p>
                <p className="description">Lorem ipsum dolor sit amet consectetur adipisicing elit. Repudiandae porro optio illum
doloremque distinctio officiis ab maxime veniam suscipit perspiciatis!</p>
            </div>
             <div className="box alert">
                <p className="title">Lorem ipsum dolor sit amet.</p>
                <p className="description">Lorem ipsum dolor sit, amet consectetur adipisicing elit. Aut quaerat, velit officia
```

```
exercitationem ut, facere ducimus, consequuntur quo libero enim repudiandae reprehenderit maiores ipsam voluptatum laudantium illo ab
sapiente aperiam voluptatem pariatur et debitis ipsa unde. Maiores assumenda praesentium tempora?</p>
          </div>
      </>
  )
}

Boxcard Code
----------------
export const BoxCard = ({Result,children}) => {
  return (
      <div className={`box ${Result}`}>

        {children}

    </div>
  )
}
```

so i am passing the parameter Result and children is understood okay

so like this for all remaining elements i can do like this okay

now again changing the code like this make it applied to all here and removing from second means changed it like this and still the code is working

```
TaskList.js
---------------
import { useState } from "react";
import { TaskCard } from "./TaskCard";
import { BoxCard } from "./BoxCard";
export const TaskList = (props) => {  //code added

        const [tasks, setTasks] = useState([
                {id: 5271, name: "Record React Lectures", completed: true},
                {id: 7825, name: "Edit React Lectures", completed: false},
                {id: 8391, name: "Watch Lectures", completed: false}
                                        ]);
        const [show, setShow] = useState(true);

        function handleDelete(id){
            setTasks(tasks.filter(task => task.id !== id));
        }
  return (
      <>
          <h1>Task List {props.title} {props.subtitle}</h1>  {/* code added here  */}
            <ul>
                <button className='trigger' onClick={() => setShow(!show)}>Toggle</button>
                { show && tasks.map((task) => (
                    <TaskCard  key={task.id} task={task} handleDelete={handleDelete} />
                )) }
          </ul>
          <BoxCard Result="success">
               <p className="title">Lorem ipsum dolor sit amet.</p>
              <p className="description">Lorem ipsum dolor sit amet consectetur adipisicing elit. Repudiandae porro optio illum
doloremque distinctio officiis ab maxime veniam suscipit perspiciatis!</p>
          </BoxCard>

                <BoxCard Result="warning">
               <p className="title">Lorem ipsum dolor sit amet.</p>
              <p className="description">Lorem ipsum dolor sit amet consectetur adipisicing elit. Repudiandae porro optio illum
doloremque distinctio officiis ab maxime veniam suscipit perspiciatis!</p>
          </BoxCard>

                <BoxCard Result="alert">
               <p className="title">Lorem ipsum dolor sit amet.</p>
              <p className="description">Lorem ipsum dolor sit amet consectetur adipisicing elit. Repudiandae porro optio illum
doloremque distinctio officiis ab maxime veniam suscipit perspiciatis!</p>
          </BoxCard>

      </>
  )
}
```

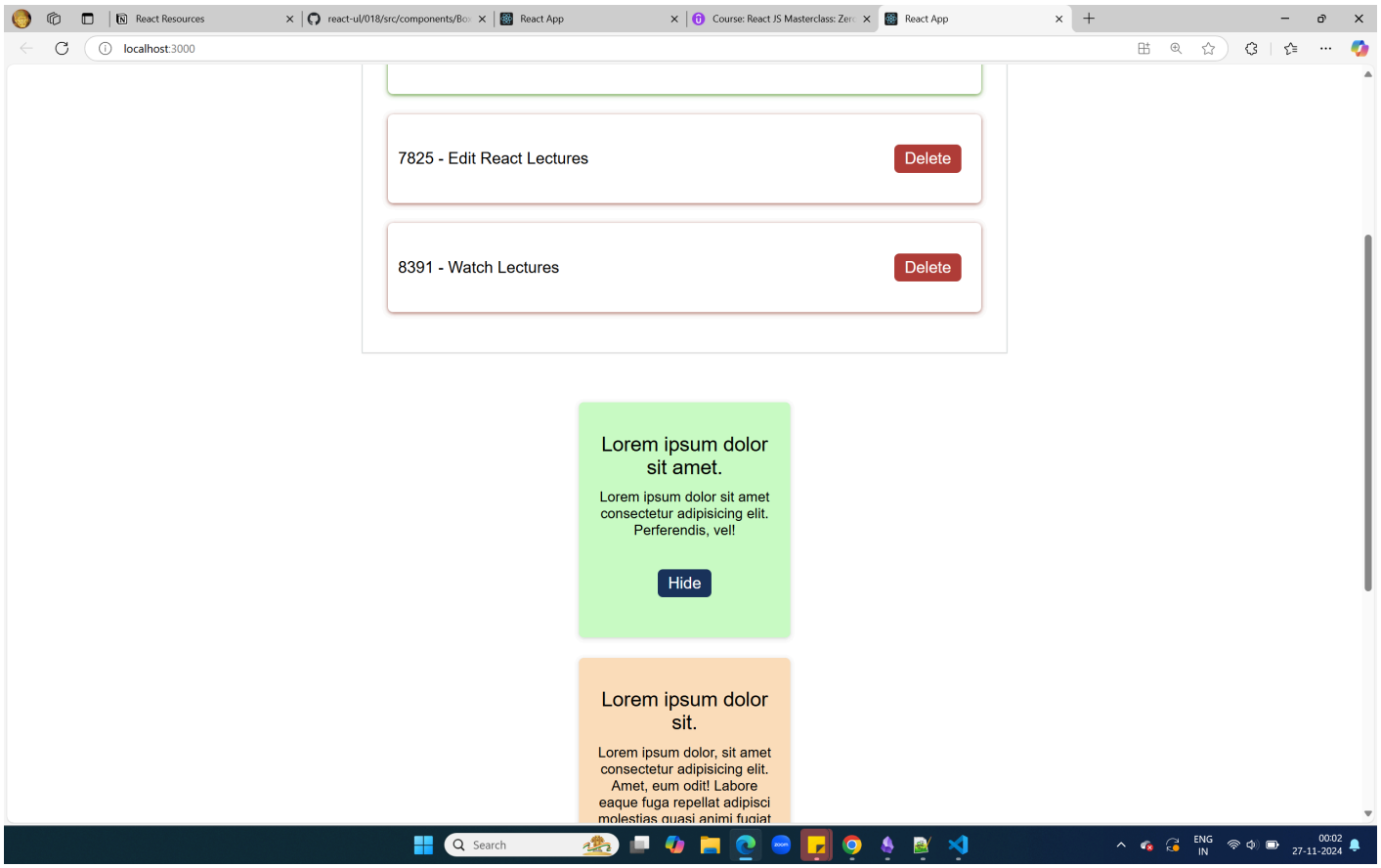
and now finally i am adding some button to hide and show the cards so the final code for that is provided below

TaskList.js

```
import { useState } from 'react';
import { TaskCard } from './TaskCard';
import { BoxCard } from './BoxCard';

export const TaskList = () => {
    const [tasks, setTasks] = useState([
        {id: 5271, name: "Record React Lectures", completed: true},
        {id: 7825, name: "Edit React Lectures", completed: false},
        {id: 8391, name: "Watch Lectures", completed: false}
    ]);
    const [show, setShow] = useState(true);

    function handleDelete(id){
        setTasks(tasks.filter(task => task.id !== id));
    }

  return (
    <>
        <h1>Task List</h1>
        <ul>
            <button className='trigger' onClick={() => setShow(!show)}>Toggle</button>
            { show && tasks.map((task) => (
                <TaskCard key={task.id} task={task} handleDelete={handleDelete} />
            )) }
        </ul>
        <BoxCard result="success">
            <p className="title">Lorem ipsum dolor sit amet.</p>
            <p className="description">Lorem ipsum dolor sit amet consectetur adipisicing elit. Perferendis, vel!</p>
        </BoxCard>

        <BoxCard result="warning">
            <p className="title">Lorem ipsum dolor sit.</p>
            <p className="description">Lorem ipsum dolor, sit amet consectetur adipisicing elit. Amet, eum odit! Labore eaque fuga repellat adipisci molestias quasi animi fugiat necessitatibus sunt vel, optio rem non quidem! Blanditiis cupiditate iusto omnis reprehenderit assumenda maxime, nam perferendis impedit libero odit eius eum aut cum ad, excepturi officiis repudiandae. Molestiae, eum cumque?</p>
            <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Accusamus, nulla?</p>
        </BoxCard>
    </>
  )
}
```

BoxCard.js

```
import { useState } from "react"

export const BoxCard = ({result, children}) => {
    const [show, setShow] = useState(true);

    return (
        <div className={show ? "" : "hidden"}>
            <div className={`box ${result}`}>
                {children}
                <button onClick={() => setShow(!show)} className="trigger">Hide</button>
            </div>
        </div>
    )
}
```



so the code is looking like this and whole code is working fine and so u can refer video or the code is self explanatory how it is done and all okay .

There is something called prop drilling means sending props to sub element and then from sub element to its sub element example is there and complete code is below generally we don't do like this we will do ...state management and centrally try to access okay

App3.js

```
--------
import { Header } from "./components/Header";
import { TaskList } from "./components/TaskList";
import './App.css';

function App() {
  const concept = "Prop Drilling";

  return (
    <div className="App">
      <Header />
      <TaskList concept={concept} />
    </div>
  );
}

export default App;
```

TaskList.js

```
--------------
import { useState } from 'react';
import { TaskCard } from './TaskCard';
import { BoxCard } from './BoxCard';

export const TaskList = ({concept}) => {
    const [tasks, setTasks] = useState([
        {id: 5271, name: "Record React Lectures", completed: true},
        {id: 7825, name: "Edit React Lectures", completed: false},
        {id: 8391, name: "Watch Lectures", completed: false}
    ]);
    const [show, setShow] = useState(true);

    function handleDelete(id){
```

```
        setTasks(tasks.filter(task => task.id !== id));
    }

  return (
    <>
        <h1>Task List</h1>
        <ul>
            <button className='trigger' onClick={() => setShow(!show)}>Toggle</button>
            { show && tasks.map((task) => (
                <TaskCard key={task.id} concept={concept} task={task} handleDelete={handleDelete} />
            )) }
        </ul>
        <BoxCard result="success">
            <p className="title">Lorem ipsum dolor sit amet.</p>
            <p className="description">Lorem ipsum dolor sit amet consectetur adipisicing elit. Perferendis, vel!</p>
        </BoxCard>

        <BoxCard result="warning">
            <p className="title">Lorem ipsum dolor sit.</p>
            <p className="description">Lorem ipsum dolor, sit amet consectetur adipisicing elit. Amet, eum odit! Labore eaque fuga
repellat adipisci molestias quasi animi fugiat necessitatibus sunt vel, optio rem non quidem! Blanditiis cupiditate iusto omnis
reprehenderit assumenda maxime, nam perferendis impedit libero odit eius eum aut cum ad, excepturi officiis repudiandae. Molestiae,
eum cumque?</p>
            <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Accusamus, nulla?</p>
        </BoxCard>
    </>
  )
}

Taskcard.js
---------------
export const TaskCard = ({ task, handleDelete, concept }) => {
  return (
    <li className={task.completed ? "completed" : "incomplete"}>
        <span>{task.id} - {task.name} - {concept}</span>
        <button onClick={() => handleDelete(task.id)} className='delete'>Delete</button>
    </li>
  )
}
```

so this is the end of section 4 and u can see how concept is passed from one component to other okay .

Section 5 (Styling React Application)

in this section we will discuss about CSS okay here if we see we have different kinds of CSS Like app.css and index.css now app is one component and for that component styles have been defined in app.css and index.css is global CSS file now what ever elements of CSS of local or component level or global level which ever comes first will be applied if they have same type in each of the file .

Generally here global means it is available to all the components the properties of the style sheets of global file index.css

If we see the web pages of our website some elements which i had defined some will be using CSS of component and some will be using CSS of index so it depends and the latest one overrides the other if they are acting on same elements with same properties .

Five points with respect to global CSS

1. common fonts
2. common icons we need to import mainly from bootstrap
3. variables we will use which are called as CSS variables which we will see it practically later for default theme or some particular button should be like this through out the application so for all this purposes we declare CSS variables here
4. some common presetting's we want to apply like presetting's for my font family or presetting's for my margin and paddings
5. some element level settings etc. like my default button should be like this etc. so all this we will check one by one

so lets start with 1 and 2 my font and icon imports

```
@import url('https://fonts.googleapis.com/css2?
family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&f
amily=Roboto&display=swap');

@import url("https://cdn.jsdelivr.net/npm/bootstrap-icons@1.9.1/font/bootstrap-icons.css");
```

added above two for fonts and icons in index.css file

next 3rd point about variables

here variables are common values which you are going to utilize it again and again .so it can be a number ,color value etc.

```
:root{
  --theme-border: #d7d7d7;
  --theme-button: #0F3460;
}
```

so now this is defined in index.css again and then i am going to App3.css and there i am changing this like this

```
header {

    display: flex;

    justify-content: space-between;

    align-items: center;

    height: 50px;

    align-items: center;

    max-width: 1000px;

    margin: auto;

    border-bottom: 1px solid #d7d7d7;

    font-size: 18px;

    padding: 0px 10px;

}
```

and now border-bottom can be changed like this

```
header {

    display: flex;

    justify-content: space-between;

    align-items: center;

    height: 50px;

    align-items: center;

    max-width: 1000px;

    margin: auto;

    border-bottom: 1px solid var(--theme-border);

    font-size: 18px;

    padding: 0px 10px;

}
```

and in the same manner i am using button theme as well you can see what has changed
so in the same manner i am using the button variable also here in app3.css okay

```
button.trigger {
    border: 0px;
    border-radius: 5px;
    background-color: #0F3460;
    color: #FFFFFF;
    padding: 5px 10px;
    cursor: pointer;
```

```
}

button.trigger {
    border: 0px;
    border-radius: 5px;
    background-color:var(--theme-button)
    color: #FFFFFF;
    padding: 5px 10px;
    cursor: pointer;
}
```

so after doing the change also i cannot see the change in design so it is working fine the variables which i had substituted okay .

Next is some common settings so i am just copy pasting the code here into index.css which can be used in App3.css later

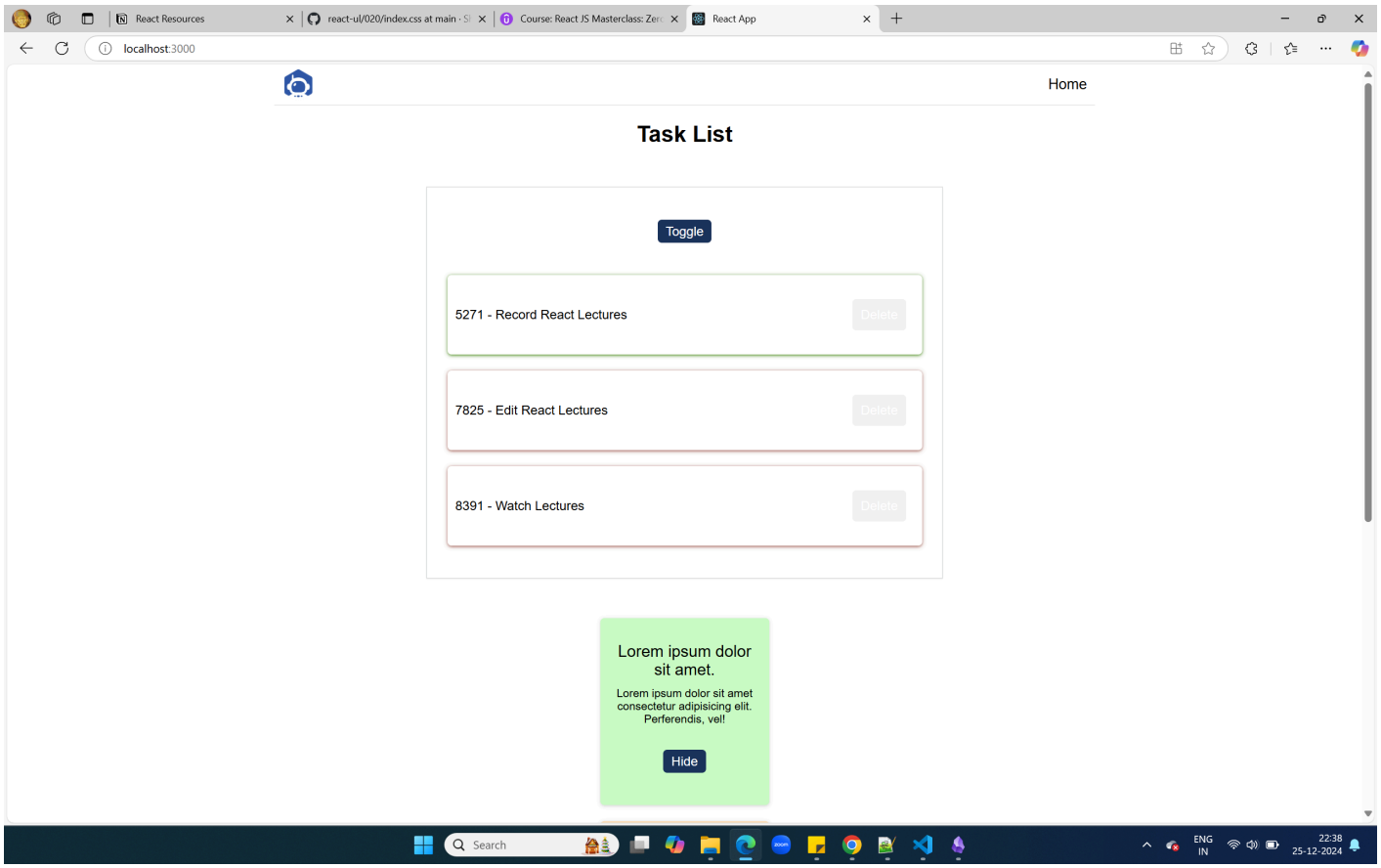
so anything which is * is for common settings applicable to all elements

```
* {
  margin: 0px;
  padding: 0px;
  box-sizing: border-box;
  font-family: 'Poppins', sans-serif;
}
```

so now 4th point is covered and now we go with element level settings okay
now suppose in TaskCard.js code below if i remove the class of delete button then it will not have any color so if i don't apply also i want some settings to be implemented from index.css then these element level settings globally will be implemented but will be overridden if any class name is given there okay

```
export const TaskCard = ({ task, handleDelete }) => {
  return (
        <li className={task.completed ? "completed" : "incomplete"}>
                    <span>{task.id} - {task.name}</span>
                    <button onClick={() =>handleDelete(task.id)} className=''>Delete</button>
        </li>
  )
}
```

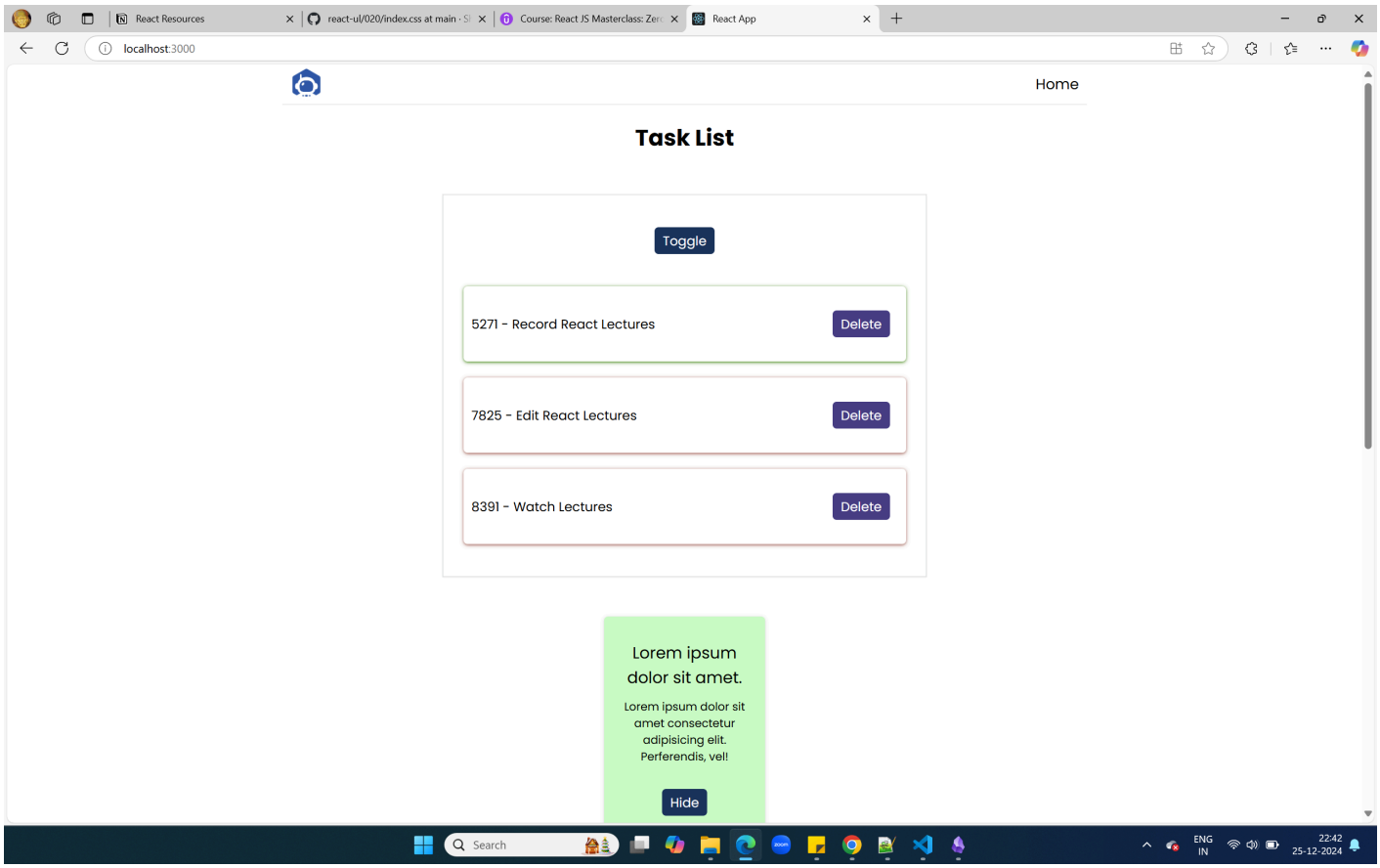
so now output will be blurred okay



I am not able to see the delete button as well so add this in index.css okay

```
button{
  padding: 5px 10px;
  border: 0px;
  border-radius: 5px;
  cursor: pointer;
  background-color: darkslateblue;
  color: #FFFFFF;
}
```

so now it will look like this



so again coming back to original color red so some more element settings for li and a tag also added and final index.css file will be like this below

```
@import url('https://fonts.googleapis.com/css2?
family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&f
amily=Roboto&display=swap');
@import url("https://cdn.jsdelivr.net/npm/bootstrap-icons@1.9.1/font/bootstrap-icons.css");

:root {
  --theme-border: #d7d7d7;
  --theme-button: #0F3460;
}

* {
  margin: 0px;
  padding: 0px;
  box-sizing: border-box;
  font-family: 'Poppins', sans-serif;
}


a {
  text-decoration: none;
  color: #000000;
}

li {
  list-style: none;
}

button {
  padding: 5px 10px;
  border: 0px;
  border-radius: 5px;
  cursor: pointer;
  background-color: darkslateblue;
  color: #FFFFFF;
}
```

and these features are used by App3.css file as well okay .

Now let us now move to component level styling here okay

now here what i am doing is that from App3.css i will take out all styles and group them or keep them in a separate CSS file now now if u observe in that separated component CSS files you will see the main div tag classname followed by the properties here section is also like container for some set of divisions so this section is html 5 element only in built one so we have used in TaskList.js so for main division or for a section one class will be there that class u will find it in its corresponding CSS file and remember if i don't give class name and define something in CSS files it will again will be global only any body can apply it so now i am pasting the complete code here for js and CSS files okay just implement the code and see the difference okay ,

so in components folder add TaskCard.css,Footer.js and Footer.css,Header.css,TaskList.css and BoxBoard.css and to the same file with js available import the corresponding CSS files and finally update the index.css file as well

```
TaskCard.js
-------------
import "./TaskCard.css";

export const TaskCard = ({ task, handleDelete }) => {
  return (
    <div className="taskcard">
      <li className={ task.completed ? "completed" : "incomplete" }>
          <span>{task.id} - {task.name}</span>
          <button onClick={() => handleDelete(task.id)} className='delete'>Delete</button>
      </li>
    </div>
  )
}
```

```
TaskCard.css
-------------
.taskcard li{
    font: 16px;
    list-style: none;
    margin: 20px 5px;
    box-shadow: rgba(0, 0, 0, 0.16) 0px 1px 4px;
    border-radius: 5px;
    padding: 10px;
    display: flex;
    justify-content: space-between;
    align-items: center;
}

.taskcard li.completed{
    box-shadow: rgb(48, 114, 0) 0px 1px 4px;
}

.taskcard li.incomplete{
    box-shadow: rgba(135, 20, 0, 0.689) 0px 1px 4px;
}

.taskcard button.delete{
    border: 0px;
    border-radius: 5px;
    background-color: #be3434;
    color: #FFFFFF;
    padding: 5px 10px;
    cursor: pointer;
}
```

```
Header.js
----------
import Logo from "../assets/logo.png"
import "./Header.css";

export const Header = () => {
  return (
    <header>
        <img src={Logo} alt="" />
        <a href="/">Home</a>
    </header>
  )
}
```

Header.css
```
------------

header {
    display: flex;
    justify-content: space-between;
    align-items: center;
    height: 50px;
    align-items: center;
    max-width: 1000px;
    margin: auto;
    border-bottom: 1px solid var(--theme-border);
    font-size: 18px;
    padding: 0px 10px;
}

header img{
    max-width: 40px;
}
```

Footer.js
```
----------
import "./Footer.css";

export const Footer = () => {
  return (
    <footer>
        <p>2030 - TaskMate</p>
    </footer>
  )
}
```

Footer.css
```
----------
footer{
    max-width: 1000px;
    margin: auto;
    text-align: center;
    padding: 20px;
    border-top: 1px solid var(--theme-border);
  }
```

TaskList.js
```
-------------
import { useState } from 'react';
import { TaskCard } from './TaskCard';
import { BoxCard } from './BoxCard';
import "./TaskList.css";

export const TaskList = () => {
    const [tasks, setTasks] = useState([
        {id: 5271, name: "Record React Lectures", completed: true},
        {id: 7825, name: "Edit React Lectures", completed: false},
        {id: 8391, name: "Watch Lectures", completed: false}
    ]);
    const [show, setShow] = useState(true);

    function handleDelete(id){
        setTasks(tasks.filter(task => task.id !== id));
    }

  return (
    <section className='tasklist'>
        <h1>Task List</h1>
        <ul>
            <button className='trigger' onClick={() => setShow(!show)}>Toggle</button>
            { show && tasks.map((task) => (
                <TaskCard key={task.id} task={task} handleDelete={handleDelete} />
            )) }
        </ul>
        <BoxCard result="success">
            <p className="title">Offer Notification</p>
```

```
            <p className="description">Lorem ipsum dolor sit amet consectetur adipisicing elit. Perferendis, vel!</p>
        </BoxCard>

        <BoxCard result="warning">
            <p className="title">Cookie Notification</p>
            <p className="description">Lorem ipsum dolor sit amet consectetur adipisicing elit. Eligendi, suscipit.</p>
        </BoxCard>
    </section>
  )
}
```

TaskList.css

```
.tasklist h1{
    font-size: 28px;
    text-align: center;
    margin: 20px;
}

.tasklist ul{
    max-width: 600px;
    margin: 50px auto;
    padding: 20px;
    box-shadow: rgba(0, 0, 0, 0.02) 0px 1px 3px 0px, rgba(27, 31, 35, 0.15) 0px 0px 0px 1px;
}

.tasklist button.trigger{
    border: 0px;
    border-radius: 5px;
    background-color: var(--theme-button);
    color: #FFFFFF;
    padding: 5px 10px;
    cursor: pointer;
}
```

BoxCard.js

```
import { useState } from "react"
import "./BoxCard.css";

export const BoxCard = ({result, children}) => {
    const [show, setShow] = useState(true);

    return (
        <div className={show ? "" : "hidden"}>
            <div className={`box ${result}`}>
                {children}
                <button onClick={() => setShow(!show)} className="trigger">Hide</button>
            </div>
        </div>
    )
}
```

BoxCard.css

```
.box {
    max-width: 600px;
    margin: 20px auto;
    box-shadow: rgba(0, 0, 0, 0.16) 0px 1px 4px;
    padding: 20px 5px;
    border: 0px;
    border-radius: 5px;
}

.box.success {
    background-color: #baffbf;
}

.box.alert {
    background-color: #ffb4b4;
}
```

```
.box.warning {
    background-color: #FFDEB4;
}

.box p {
    margin: 10px;
}

.box p.title {
    font-size: 20px;
}

.box p.description {
    font-size: 14px;
}

.box button{
    width: 300px;
    margin-top: 20px;
}
```

App3.js

```
import { Header } from "./components/Header";
import { Footer } from "./components/Footer";
import { TaskList } from "./components/TaskList";
import './App3.css';

function App() {

  return (
    <div className="App">
      <Header />
      <TaskList />
      <Footer />
    </div>
  );
}

export default App;
```

App3.css

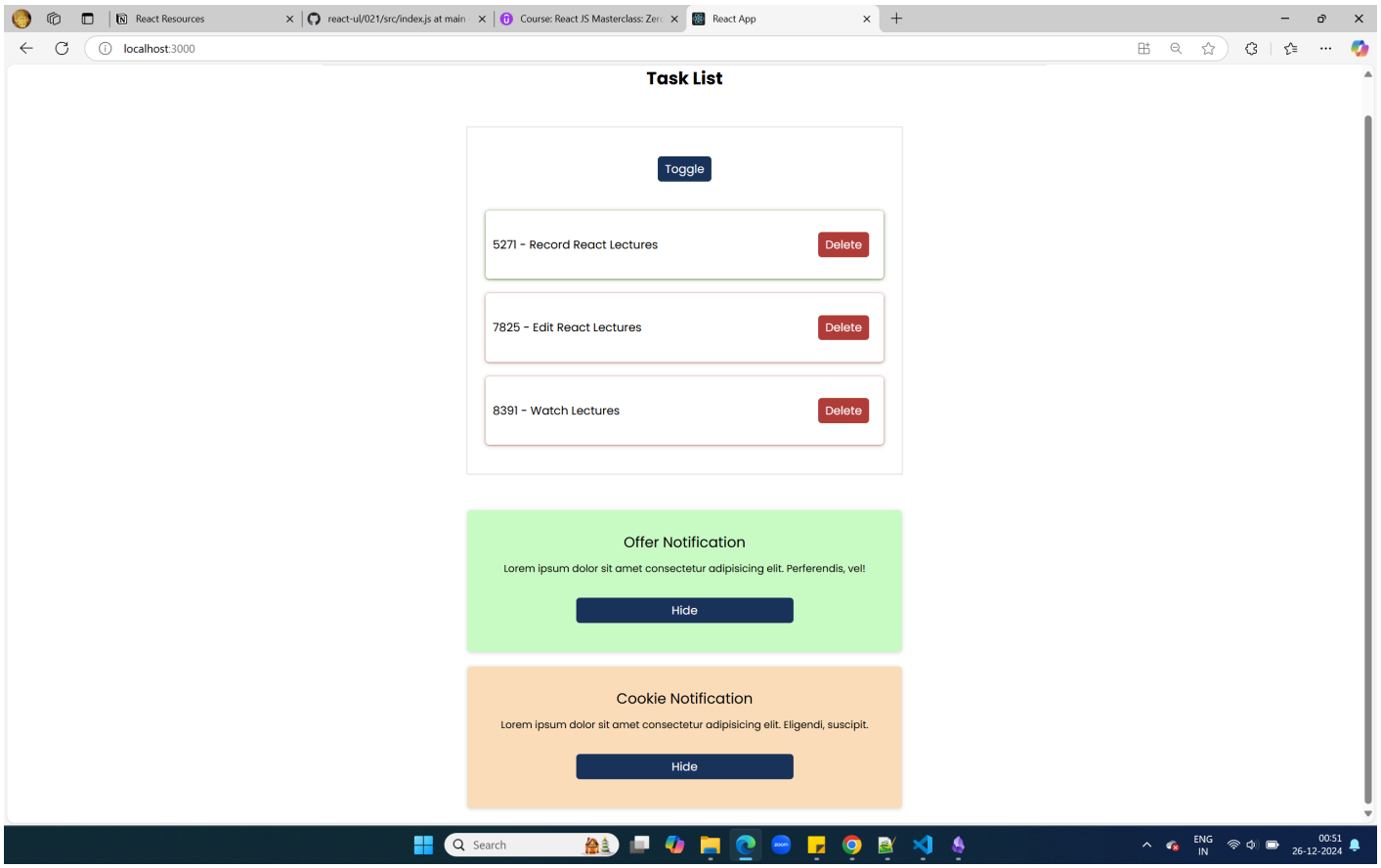
```
.App{
  text-align: center;
}

.hidden{
  display: none;
}
```

index.css

```
@import url('https://fonts.googleapis.com/css2?
family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&f
amily=Roboto&display=swap');
@import url("https://cdn.jsdelivr.net/npm/bootstrap-icons@1.9.1/font/bootstrap-icons.css");

:root{
  --theme-border: #d7d7d7;
  --theme-button: #0F3460;
}

* {
  margin: 0px;
  padding: 0px;
  box-sizing: border-box;
  font-family: 'Poppins', sans-serif;
}

main {
  min-height: 90vh;
```

```
  max-width: 1200px;
  margin: auto;
  padding: 10px;
}

a{
  text-decoration: none;
  color: #000000;
}

li{
  list-style: none;
}

button{
  padding: 5px 10px;
  border: 0px;
  border-radius: 5px;
  cursor: pointer;
  background-color: darkslateblue;
  color: #FFFFFF;
}
```

so accordingly after changing i am getting the same output but little bit modified

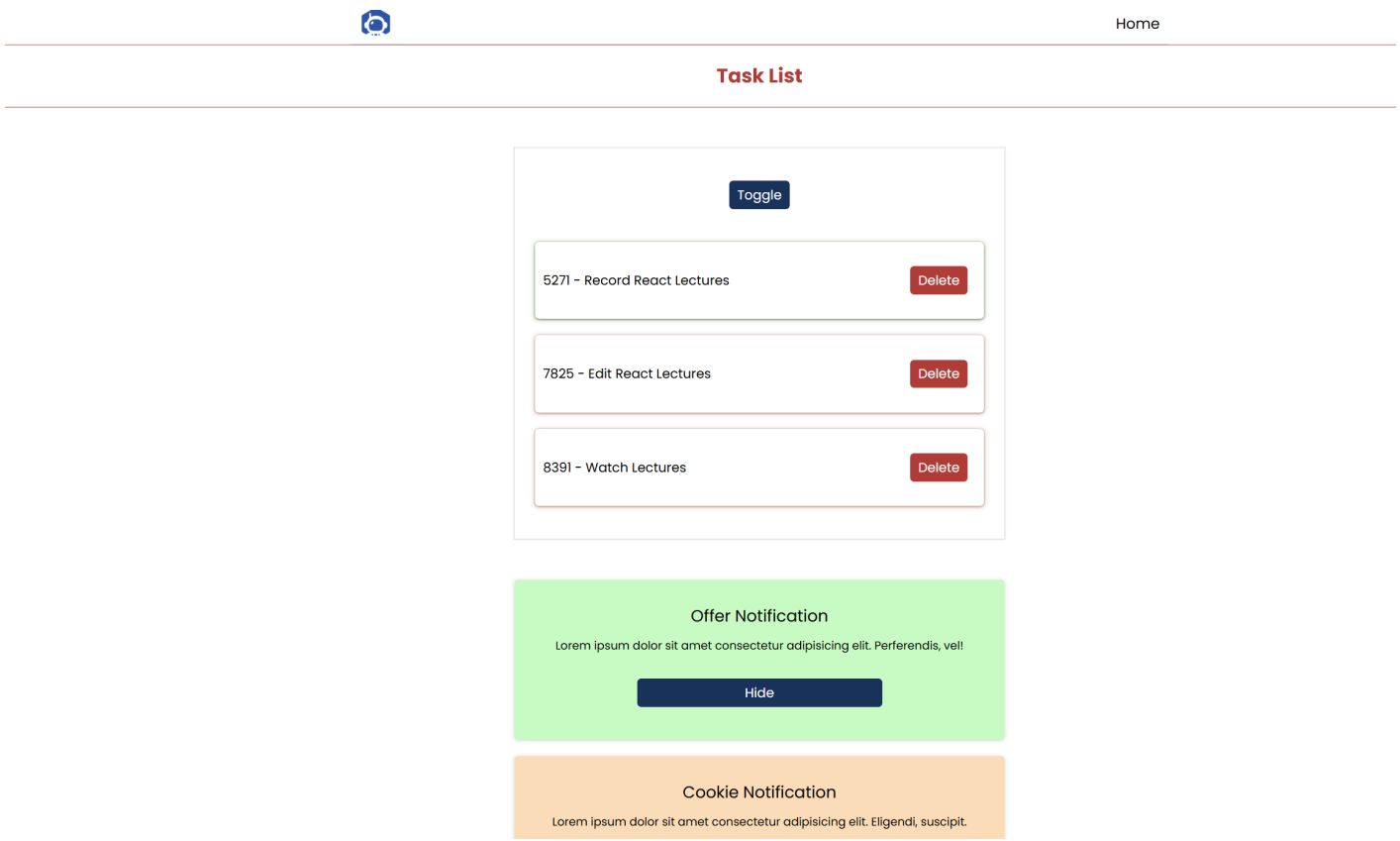


Now let us move to inline styles

so i had taken just TaskList.js file and added inline style which is overriding the CSS of it

```
import { useState } from 'react';
import { TaskCard } from './TaskCard';
import { BoxCard } from './BoxCard';
import "./TaskList.css";

export const TaskList = () => {
    const [tasks, setTasks] = useState([
        {id: 5271, name: "Record React Lectures", completed: true},
        {id: 7825, name: "Edit React Lectures", completed: false},
        {id: 8391, name: "Watch Lectures", completed: false}
```

```
    ]);
    const [show, setShow] = useState(true);
    const styles = {
        color: "#be3434",
        border: "1px solid #be3434",
        borderRadius: "5px",
        padding: "20px"
    }

    function handleDelete(id){
        setTasks(tasks.filter(task => task.id !== id));
    }

  return (
    <section className='tasklist'>
        <h1 style={styles}>Task List</h1>
        <ul>
            <button className='trigger' onClick={() => setShow(!show)}>Toggle</button>
            { show && tasks.map((task) => (
                <TaskCard key={task.id} task={task} handleDelete={handleDelete} />
            )) }
        </ul>
        <BoxCard result="success">
            <p className="title">Offer Notification</p>
            <p className="description">Lorem ipsum dolor sit amet consectetur adipisicing elit. Perferendis, vel!</p>
        </BoxCard>

        <BoxCard result="warning">
            <p className="title">Cookie Notification</p>
            <p className="description">Lorem ipsum dolor sit amet consectetur adipisicing elit. Eligendi, suscipit.</p>
        </BoxCard>
    </section>
  )
}
```

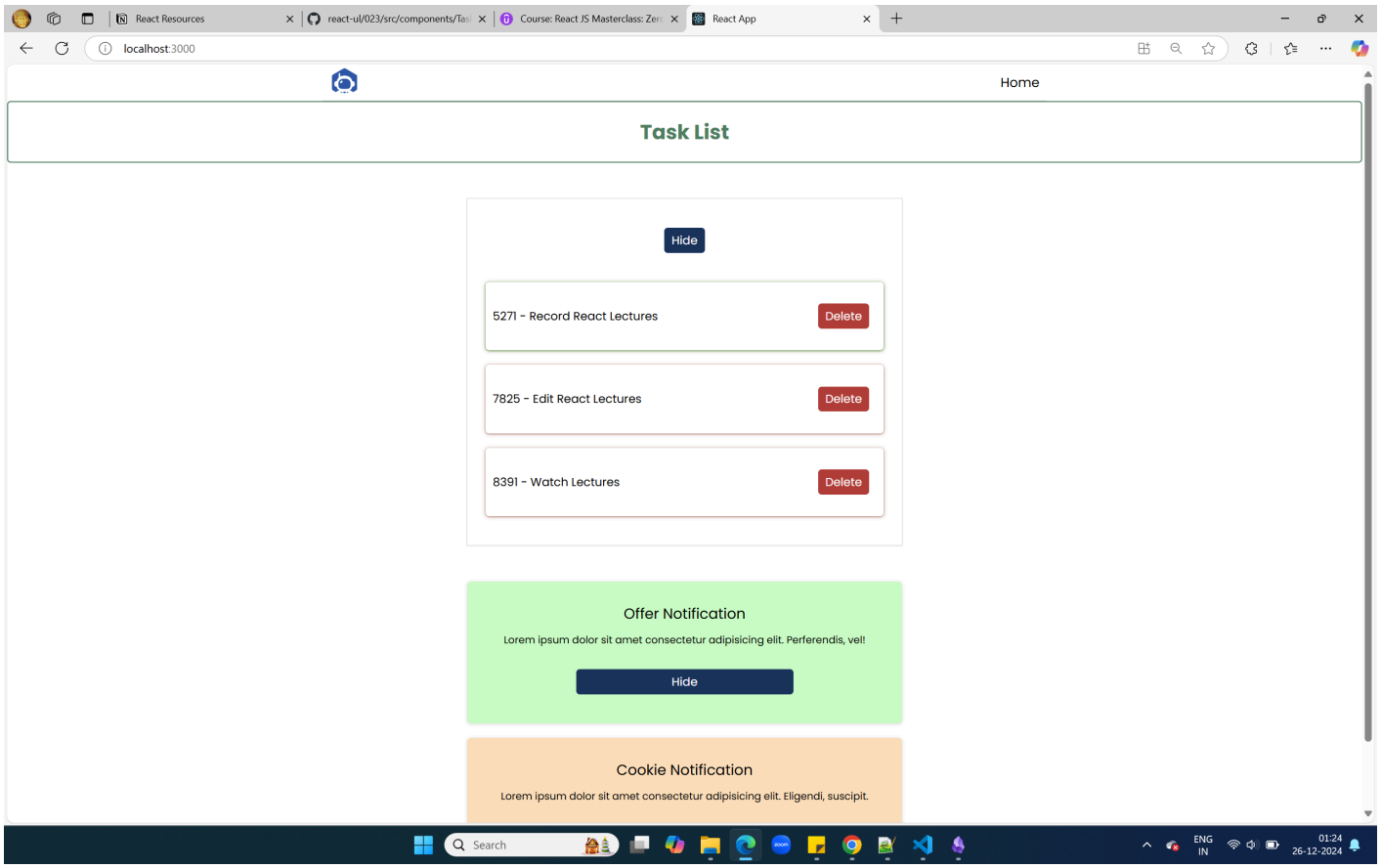


so i will get a border like this which u can see in image so here above the styles i can directly put there no need to use object means using {{ }} hope u understand

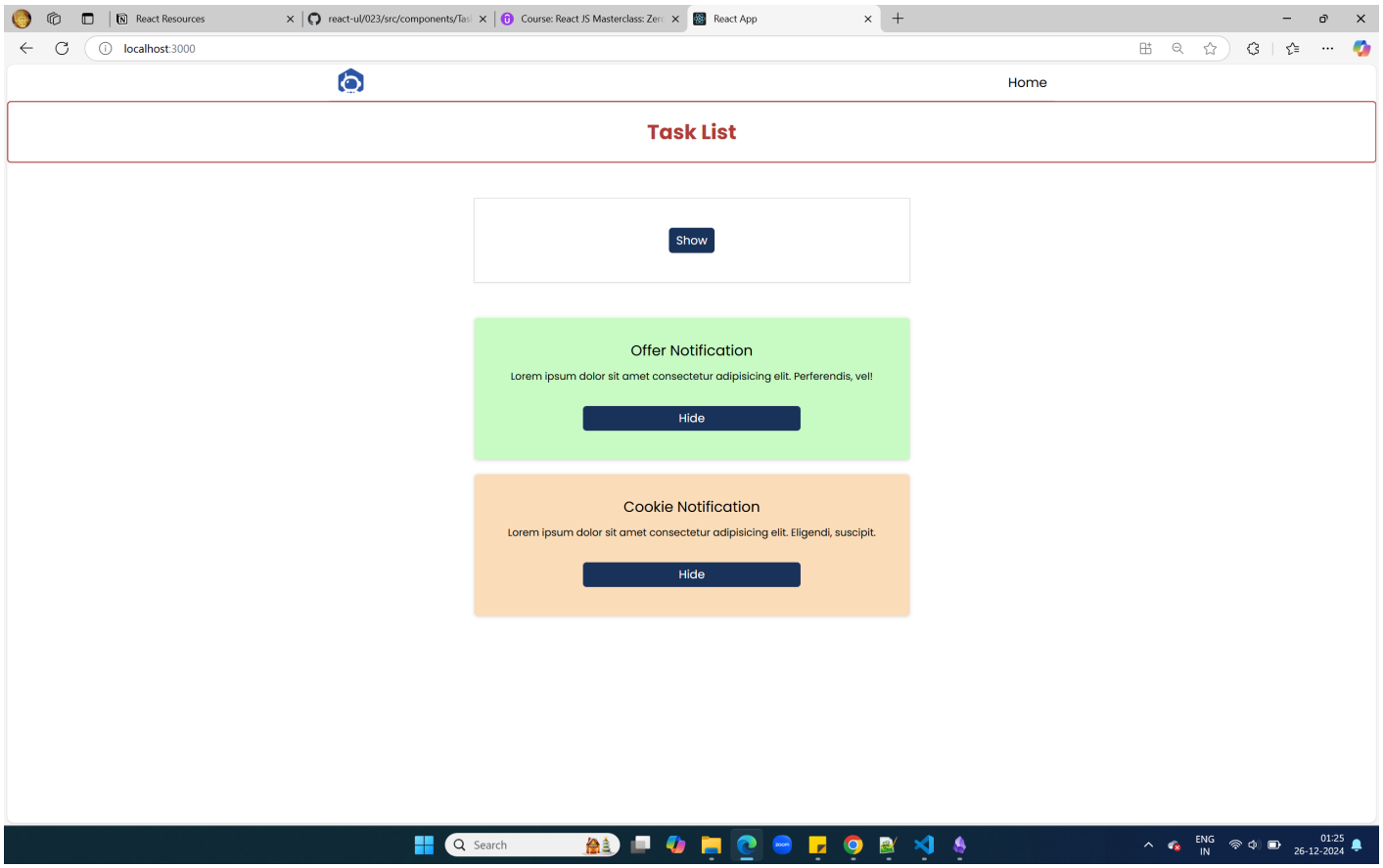
Next i want to implement dynamic in line CSS here okay so same file now that border will show blue one time and another time red based on show and here toggle button again there i will put hide and show instead of just message there of toggle okay

```jsx
import { useState } from 'react';
import { TaskCard } from './TaskCard';
import { BoxCard } from './BoxCard';
import "./TaskList.css";

export const TaskList = () => {
    const [tasks, setTasks] = useState([
        {id: 5271, name: "Record React Lectures", completed: true},
        {id: 7825, name: "Edit React Lectures", completed: false},
        {id: 8391, name: "Watch Lectures", completed: false}
    ]);
    const [show, setShow] = useState(true);
    const styles = {
        color: show ? "#3D8361" : "#be3434",
        border: "2px solid",
        borderColor: show ? "#3D8361" : "#be3434" ,
        borderRadius: "5px",
        fontSize: "28px",
        padding: "20px"
    }

    function handleDelete(id){
        setTasks(tasks.filter(task => task.id !== id));
    }

  return (
    <section className='tasklist'>
        <h1 style={styles}>Task List</h1>
        <ul>
            <button className='trigger' onClick={() => setShow(!show)}>{ show ? "Hide" : "Show"}</button>
            { show && tasks.map((task) => (
                <TaskCard key={task.id} task={task} handleDelete={handleDelete} />
            )) }
        </ul>
        <BoxCard result="success">
            <p className="title">Offer Notification</p>
            <p className="description">Lorem ipsum dolor sit amet consectetur adipisicing elit. Perferendis, vel!</p>
        </BoxCard>

        <BoxCard result="warning">
            <p className="title">Cookie Notification</p>
            <p className="description">Lorem ipsum dolor sit amet consectetur adipisicing elit. Eligendi, suscipit.</p>
        </BoxCard>
    </section>
  )
}
```

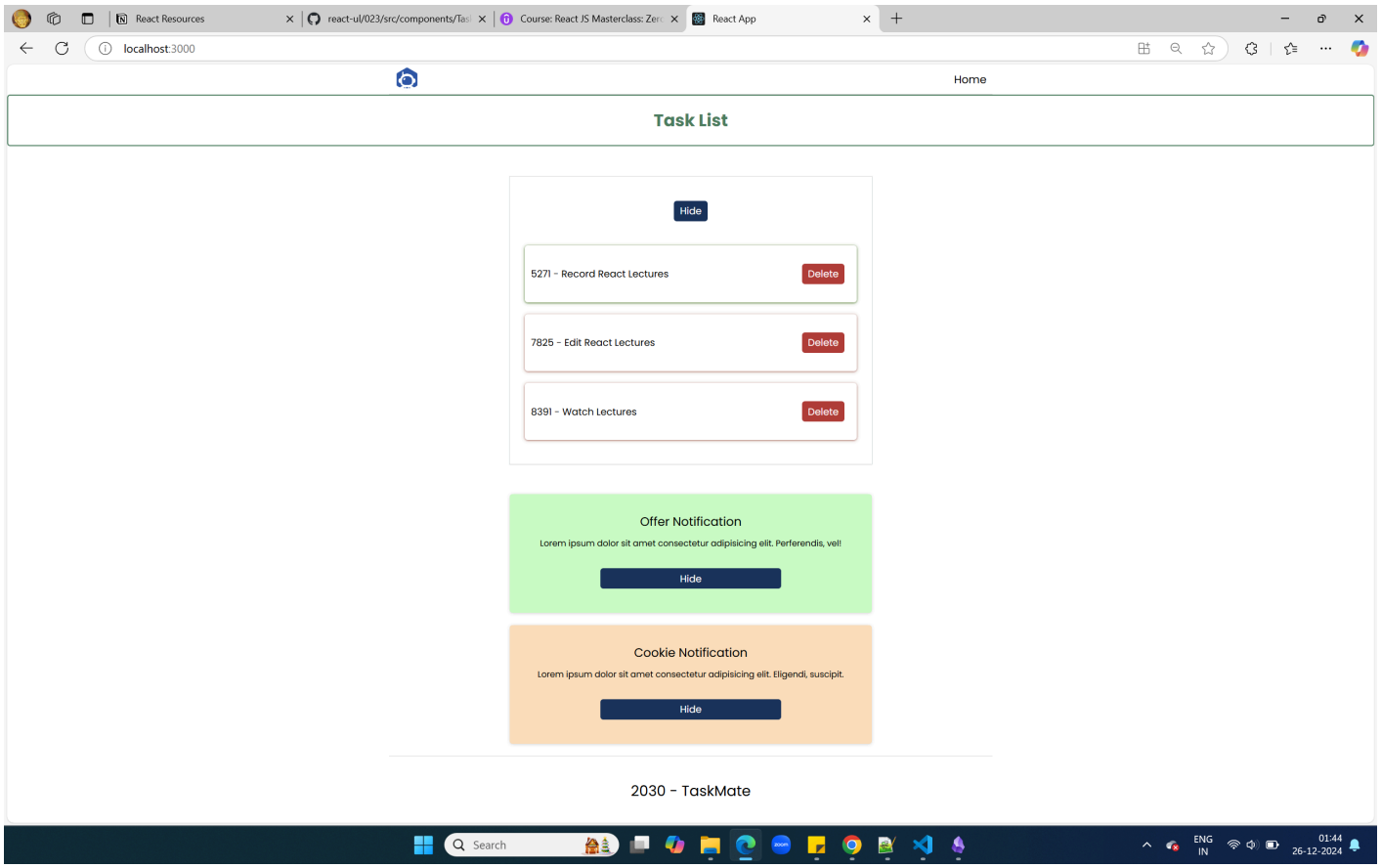
output when show



and when hide is clicked



so this is all about dynamic inline style this thing we have done earlier in TaskCard using ternary operator same thing again i had implemented okay .Final image will look like this



Now let us see scope related to CSS means module level styling let us implement it okay .

This one i am not maintaining as notes as not important so now move to next section

Section 6 (User Input ,Events and useRef)

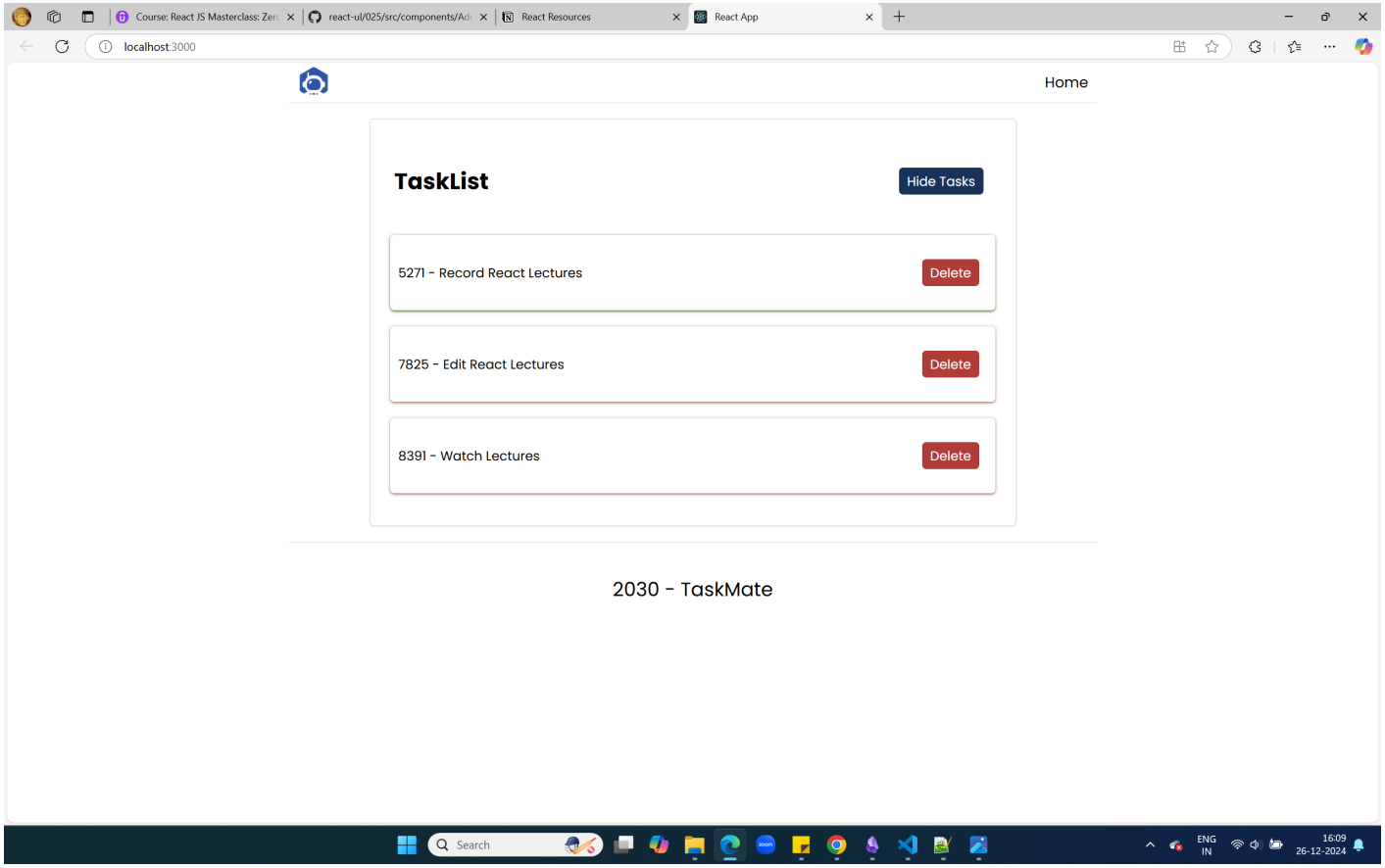
Now what i am doing is that i have added new code to the file means done some slight changes in TaskList.js file and added two CSS files to it for which also coding is provided below .

```
TaskList.js
--------------
import { useState } from 'react';
import { TaskCard } from './TaskCard';
import "./TaskList.css";
import "./AddTask.css";

export const TaskList = () => {
    const [tasks, setTasks] = useState([
        {id: 5271, name: "Record React Lectures", completed: true},
        {id: 7825, name: "Edit React Lectures", completed: false},
        {id: 8391, name: "Watch Lectures", completed: false}
    ]);
    const [show, setShow] = useState(true);

    function handleDelete(id){
        setTasks(tasks.filter(task => task.id !== id));
    }

  return (
    <section className='tasklist'>
        <ul>
            <div className='header'>
                <h1>TaskList</h1>
                <button className='trigger' onClick={() => setShow(!show)}>{ show ? "Hide Tasks" : "Show Tasks"}</button>
            </div>
            { show && tasks.map((task) => (
                <TaskCard key={task.id} task={task} handleDelete={handleDelete} />
            )) }
        </ul>
```

```
    </section>
  )
}


TaskList.css
-------------
.tasklist ul{
    max-width: 800px;
    margin: 20px auto;
    padding: 20px;
    box-shadow: rgba(0, 0, 0, 0.02) 0px 1px 3px 0px, rgba(27, 31, 35, 0.15) 0px 0px 0px 1px;
    border-radius: 5px;
}

.tasklist .header{
    display: flex;
    justify-content: space-between;
    align-items: center;
    margin: 20px 10px;
    margin-bottom: 30px;
}

.tasklist h1{
    font-size: 28px;
    text-align: center;
}

.tasklist button.trigger{
    border: 0px;
    border-radius: 5px;
    background-color: var(--theme-button);
    color: #FFFFFF;
    padding: 5px 10px;
    cursor: pointer;
    font-size: 16px;
}
AddTask.css
------------
.addtask{
    max-width: 800px;
    margin: 50px auto;
    padding: 20px;
    box-shadow: rgba(0, 0, 0, 0.02) 0px 1px 3px 0px, rgba(27, 31, 35, 0.15) 0px 0px 0px 1px;
    border-radius: 5px;
}

.addtask form{
    display: flex;
    justify-content: space-around;
    align-items: center;
    flex-wrap: wrap;
}

.addtask label{
    font-size: 20px;
    margin-right: 5px;
}

.addtask input{
    font-size: 18px;
    padding: 7px;
    border: 1px solid #878787;
    border-radius: 5px;
    flex-grow: 1;
    margin: 10px;
}

.addtask input:focus{
    outline: none;
}

.addtask select{
    font-size: 18px;
    padding: 7px;
```

```
    border-radius: 5px;
    margin: 10px;
    cursor: pointer;
}

.addtask button{
    font-size: 18px;
    padding: 7px 10px;
    margin: 10px;
}
```

so after above code update it looks like this



now i had added Addtask.js code in that i had added a form and in that form text box and button and then on change event i had controlled the text box here so the file is added in App3.js as well so complete code is there below you can understand the changes done in comments section okay
so i had used finally anonymous function .

```
AddTask.js
-------------
import { useState } from "react";
import "./AddTask.css";

export const AddTask = () => {
    const [taskValue, setTaskValue] = useState("");

    // const handleChange = (event) => {
    //     setTaskValue(event.target.value);
        // console.log(event.target.value) // here intially i kept "-" how many times it is appearing that i am checking and by default event was not passed it was there later
        // based on code change the code got modifed and event is passed so we can use small e also it i just parameter which we are passing okay .
    // }

  return (
    <section className="addtask">
        <form>
            <input onChange={(e) => setTaskValue(e.target.value)} type="text" name="task" id="task" placeholder="Task Name" autoComplete="off" />
                        {/* <input onChange={handleChange} type="text" name="task" id="task" placeholder="Task Name"
```

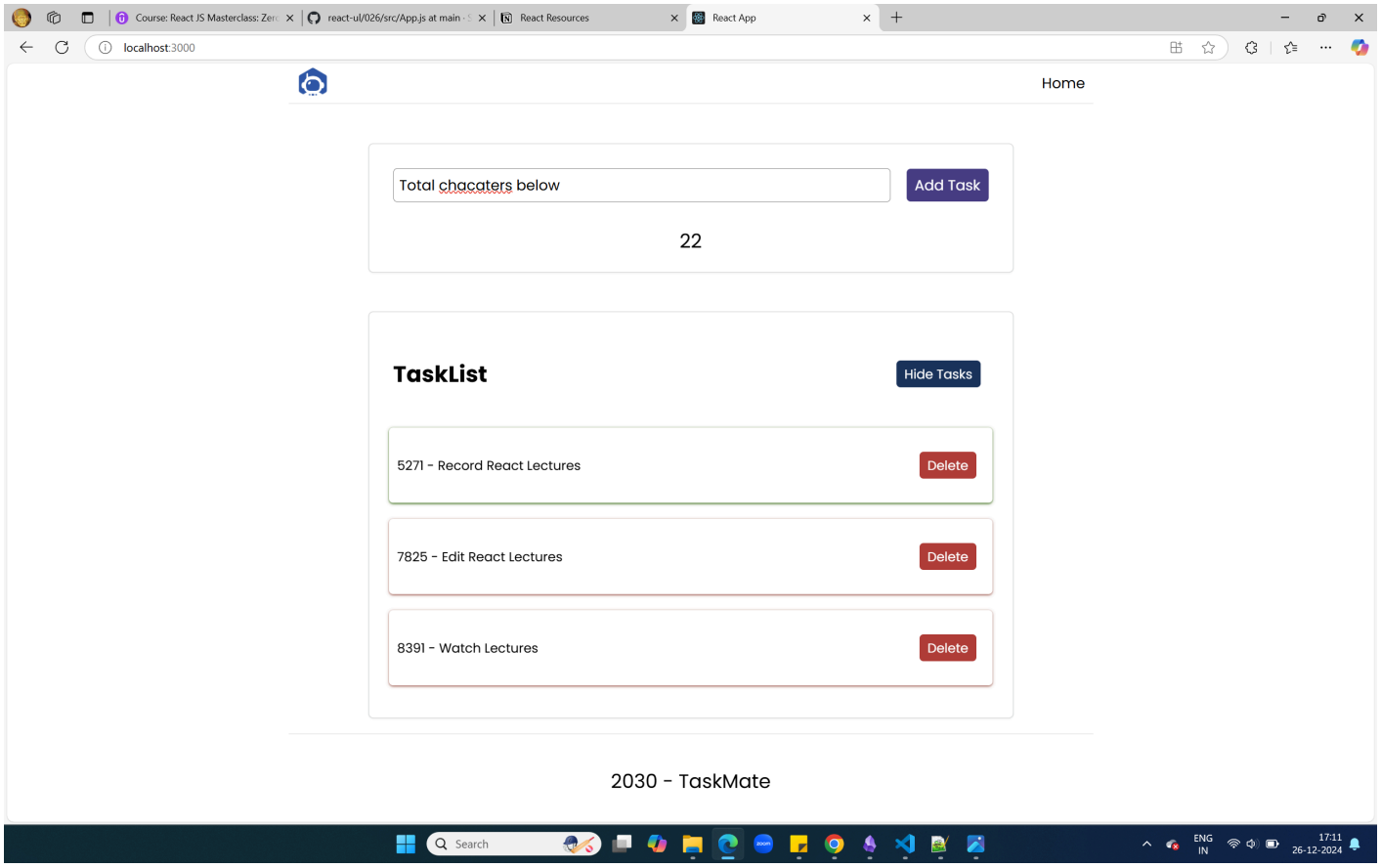
```
autoComplete="off" /> */}
            <button type="submit">Add Task</button>
          </form>
          <p>{taskValue.length}</p>
        {/*  earleir {taskValue} only written later got intersted in finding length so change the code like this  */}
    </section>
  )
}
```

App3.js

```
import './App3.css';
import { Footer } from './components/Footer';
import { Header } from './components/Header';
import { TaskList } from './components/TaskList';
import { AddTask } from './components/AddTask';
function App() {

  return (
    <div className="App">
      <Header />
      <AddTask />
      <TaskList title="Random" subtitle="Test" />  {/* code added here */}
      <Footer/>
    </div>
  );
}

export default App;
```

so here TaskList.js will show me the list and AddTask.js will allow me to add new task into the list

so final output till now will look like this



Now again further changes adding on click method reset functionality here in AddTask.js file

AddTask.js

```
import { useState } from "react";
import "./AddTask.css";

export const AddTask = () => {
    const [taskValue, setTaskValue] = useState("");

    const handleChange = (event) => {
        setTaskValue(event.target.value);
    }

    const handleReset = () => {
        setTaskValue("");
    }

  return (
    <section className="addtask">
        <form>
            <input onChange={handleChange} type="text" name="task" id="task" placeholder="Task Name" autoComplete="off" value=
{taskValue} />
            <button type="submit">Add Task</button>
            <span onClick={handleReset} className="reset">Reset</span>
        </form>
        <p>{taskValue}</p>
    </section>
  )
}
```

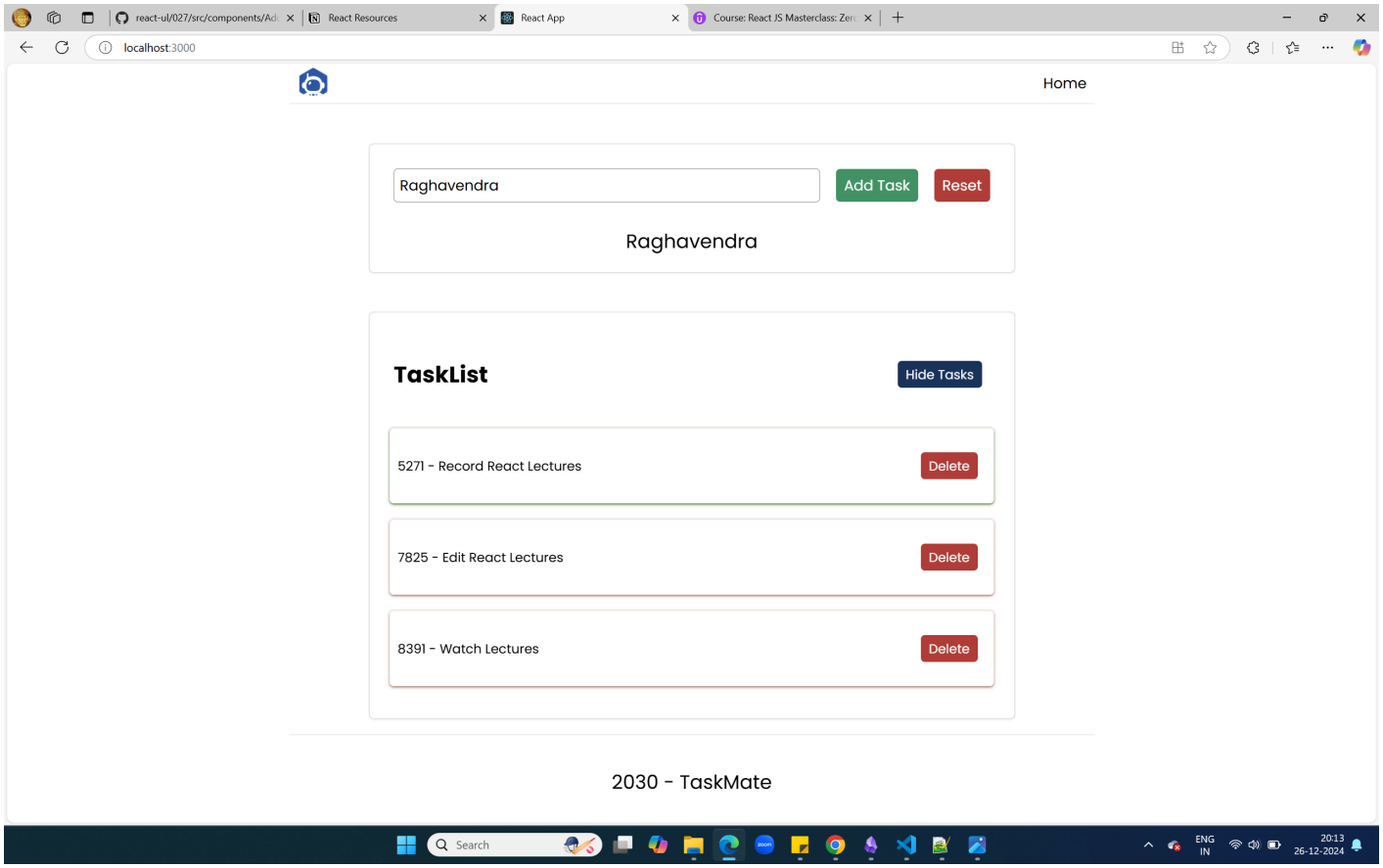
here i had made the code simple where i had put reset functionality here okay also in AddTask.css reset styles are also added

AddTask.css

```
.addtask {
    max-width: 800px;
    margin: 50px auto;
    padding: 20px;
    box-shadow: rgba(0, 0, 0, 0.02) 0px 1px 3px 0px, rgba(27, 31, 35, 0.15) 0px 0px 0px 1px;
    border-radius: 5px;
}

.addtask form {
    display: flex;
    justify-content: space-around;
    align-items: center;
    flex-wrap: wrap;
}

.addtask label {
    font-size: 20px;
    margin-right: 5px;
}

.addtask input {
    font-size: 18px;
    padding: 7px;
    border: 1px solid #878787;
    border-radius: 5px;
    flex-grow: 1;
    margin: 10px;
}

.addtask input:focus {
    outline: none;
}

.addtask select {
    font-size: 18px;
    padding: 7px;
    border-radius: 5px;
    margin: 10px;
    cursor: pointer;
```

```
}

.addtask button {
    font-size: 18px;
    padding: 7px 10px;
    margin: 10px;
    background-color: #129762;
}

.addtask .reset {
    font-size: 18px;
    padding: 7px 10px;
    margin: 10px;
    background-color: #be3434;
    color: #FFFFFF;
    border-radius: 5px;
    cursor: pointer;
}
```

and the output i am getting in this manner



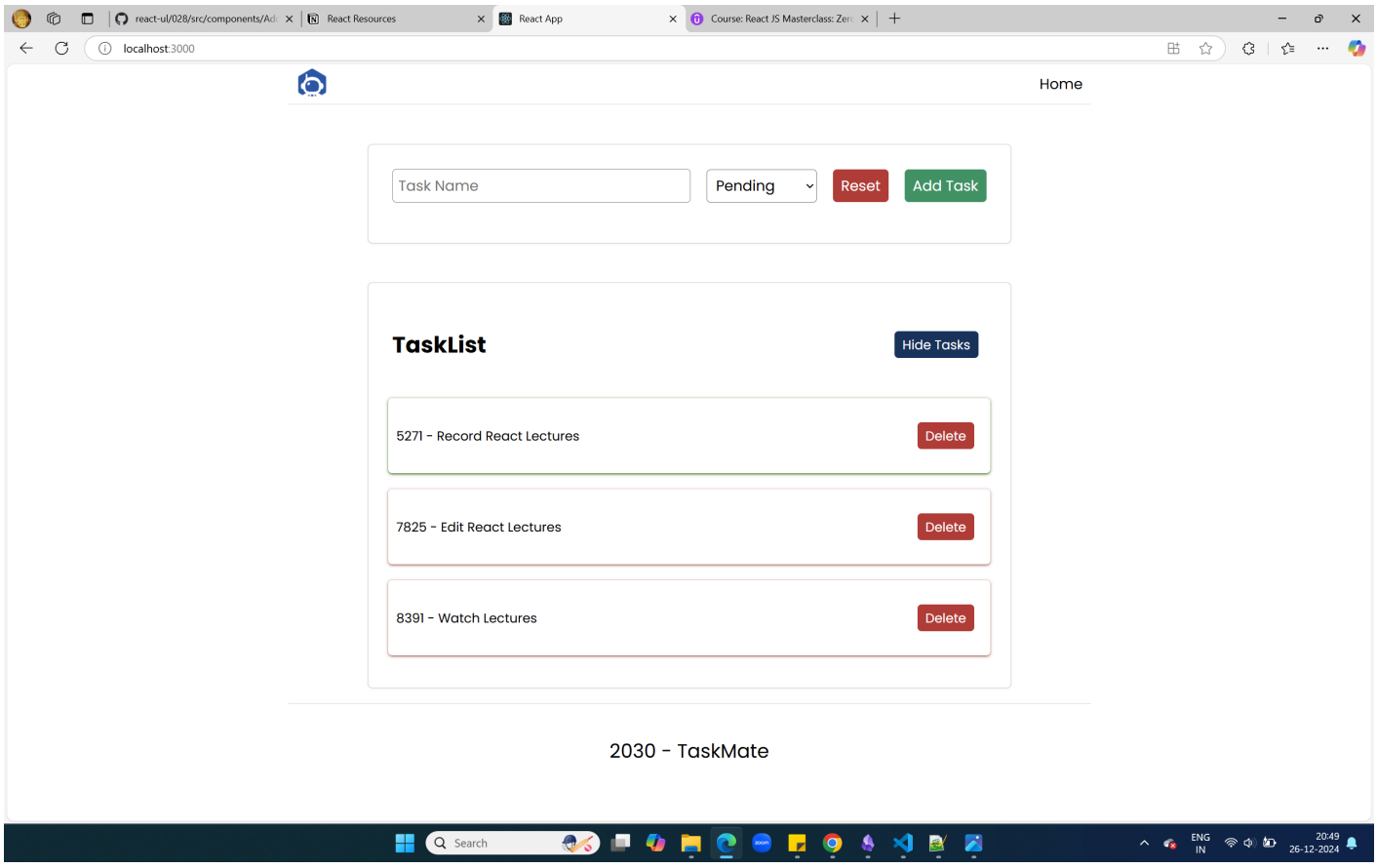
Now i will work on on submit method of form so i am calling a method over there prevent defaults to avoid page refresh and also to remove data after clicking submit button i am doing handle reset method and finally i am creating a drop and from there doing selection and data type which i am getting there is string that i am converting into Boolean and what is value of progress that i am using it and supplying it value so just analyze the code you will understand it clearly .

AddTask.js

```
import { useState } from "react";
import "./AddTask.css";

export const AddTask = () => {
    const [taskValue, setTaskValue] = useState("");
    const [progress, setProgress] = useState(false);

    const handleChange = (event) => {
        setTaskValue(event.target.value);
    }

    const handleReset = () => {
```

```
        setTaskValue("");
        setProgress(false);
    }

    const handleSubmit = (event) => {
        event.preventDefault();// handling resfresh of page
        const task = {
            id: Math.floor(Math.random() * 10000),
            name: taskValue,
            completed: Boolean(progress)
        }
        //      console.log(task);
        handleReset();// clearing the text box after clicking submit button
    }

  return (
    <section className="addtask">
        <form onSubmit={handleSubmit}>
            <input onChange={handleChange} type="text" name="task" id="task" placeholder="Task Name" autoComplete="off" value=
{taskValue} />
            <select onChange={(event) => setProgress(event.target.value)} value={progress}>
                <option value="false">Pending</option>
                <option value="true">Completed</option>
            </select>
            <span onClick={handleReset} className="reset">Reset</span>
            <button type="submit">Add Task</button>
        </form>
        <p>{taskValue}</p>
    </section>
  )
}
```

and the output will be like this



Now i need to add the task object into the current 3 array elements as 4th element okay

i have to add task and using add task component i have to add it TaskList.js so but they both are siblings so not possible to do that so in App3.js file only i will take the collection of arrays and will work like this for which the code is provided okay .

```
// here removed the array and kept in App3.js and as i am passing props so here i had added props also here tasks and setTasks

TaskList.js
-------------
import { useState } from 'react';
import { TaskCard } from './TaskCard';
import "./TaskList.css";
import "./AddTask.css";

export const TaskList = ({tasks,setTasks}) => {

    const [show, setShow] = useState(true);

    function handleDelete(id){
        setTasks(tasks.filter(task => task.id !== id));
    }

  return (
    <section className='tasklist'>
        <ul>
            <div className='header'>
                <h1>TaskList</h1>
                <button className='trigger' onClick={() => setShow(!show)}>{ show ? "Hide Tasks" : "Show Tasks"}</button>
            </div>
            { show && tasks.map((task) => (
                <TaskCard key={task.id} task={task} handleDelete={handleDelete} />
            )) }
        </ul>
    </section>
  )
}


App3.js
--------
import { useState } from 'react';
import './App3.css';
import { Footer } from './components/Footer';
import { Header } from './components/Header';
import { TaskList } from './components/TaskList';
import { AddTask } from './components/AddTask';

function App() {
  const [tasks, setTasks] = useState([
         {id: 5271, name: "Record React Lectures", completed: true},
         {id: 7825, name: "Edit React Lectures", completed: false},
         {id: 8391, name: "Watch Lectures", completed: false}
     ]);
  return (
    <div className="App">
      <Header />
      <AddTask />
      <TaskList tasks={tasks} setTasks={setTasks}  />
      <Footer/>
    </div>
  );
}

export default App;
```

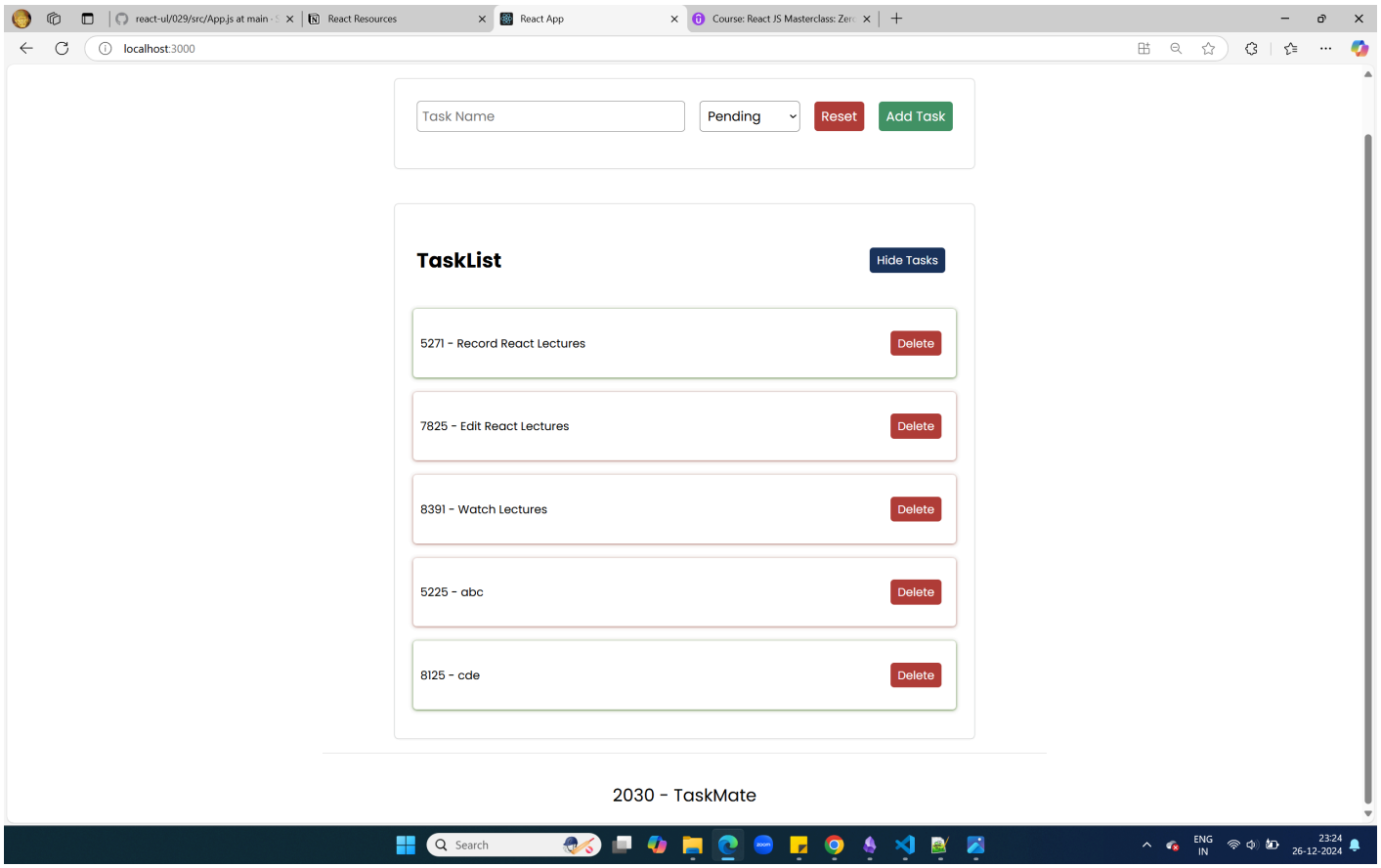
And now also there is no change in output now and then ...since i cannot pass information to siblings so from parent i am passing in the form of props to child and i have to pass this information for AddTask.js as well from App3.js

```
// here added props to AddTasks.js and gone to AddTask.js and

App3.js
--------
import { useState } from 'react';
import './App3.css';
import { Footer } from './components/Footer';
import { Header } from './components/Header';
import { TaskList } from './components/TaskList';
import { AddTask } from './components/AddTask';
```

```
function App() {
  const [tasks, setTasks] = useState([
         {id: 5271, name: "Record React Lectures", completed: true},
         {id: 7825, name: "Edit React Lectures", completed: false},
         {id: 8391, name: "Watch Lectures", completed: false}
     ]);
  return (
    <div className="App">
      <Header />
      <AddTask tasks={tasks} setTasks={setTasks} />
      <TaskList tasks={tasks} setTasks={setTasks}  />
      <Footer/>
    </div>
  );
}

export default App;
```

AddTask.js

```
import { useState } from "react";
import "./AddTask.css";

export const AddTask = ({tasks,setTasks}) => {
    const [taskValue, setTaskValue] = useState("");
    const [progress, setProgress] = useState(false);

    const handleChange = (event) => {
        setTaskValue(event.target.value);
    }

    const handleReset = () => {
        setTaskValue("");
        setProgress(false);
    }

    const handleSubmit = (event) => {
        event.preventDefault();// handling resfresh of page
        const task = {
            id: Math.floor(Math.random() * 10000),
            name: taskValue,
            completed: Boolean(progress)
        }
        //      console.log(task);
        //setTasks(task) // this will give error as it is list so chnaged like this below
        //setTasks([task])// this will override earlier 3 tasks so finally i will write this we have to follow this rule
        setTasks([ ...tasks,task])
        handleReset();// clearing the text box after clicking submit button
    }

  return (
    <section className="addtask">
        <form onSubmit={handleSubmit}>
            <input onChange={handleChange} type="text" name="task" id="task" placeholder="Task Name" autoComplete="off" value=
{taskValue} />
            <select onChange={(event) => setProgress(event.target.value)} value={progress}>
                <option value="false">Pending</option>
                <option value="true">Completed</option>
            </select>
            <span onClick={handleReset} className="reset">Reset</span>
            <button type="submit">Add Task</button>
        </form>
        <p>{taskValue}</p>
    </section>
  )
}
```

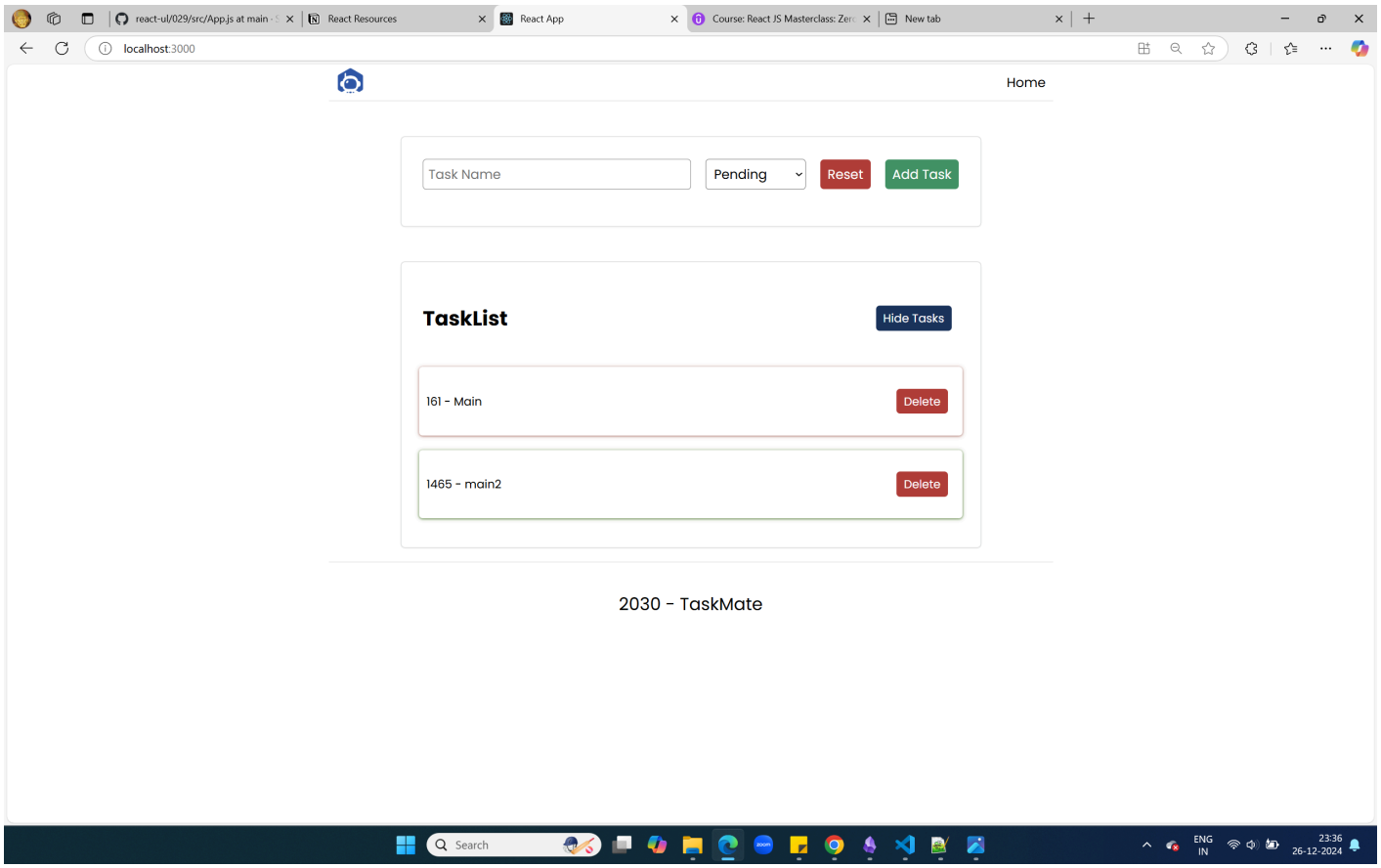


so i am able to add abc and cde tasks here above one not completed another completed completed is green border so now till now the things are working fine .so i am able to add tasks to existing three collection elements okay

Now i want to clear manual coded 3 elements in array and want to make it empty

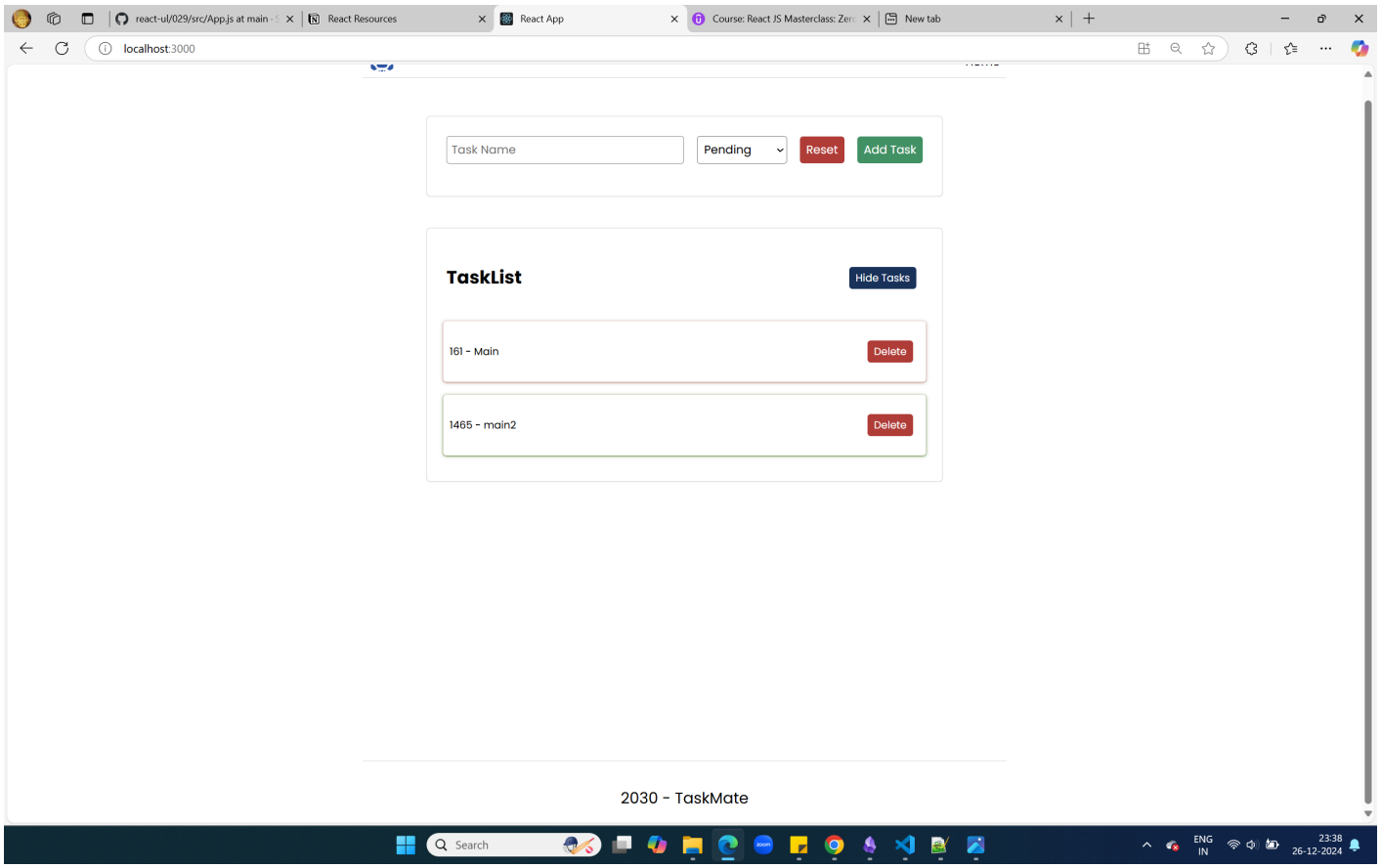
```
App3.js
---------
import { useState } from 'react';
import './App3.css';
import { Footer } from './components/Footer';
import { Header } from './components/Header';
import { TaskList } from './components/TaskList';
import { AddTask } from './components/AddTask';

function App() {
  const [tasks, setTasks] = useState([]);

  return (
    <div className="App">
      <Header />
      <AddTask tasks={tasks} setTasks={setTasks} />
      <TaskList tasks={tasks} setTasks={setTasks}  />
      <Footer/>
    </div>
  );
}

export default App;
```

now if u run you can add all new tasks only okay ..which u can try by yourself and it is working fine also

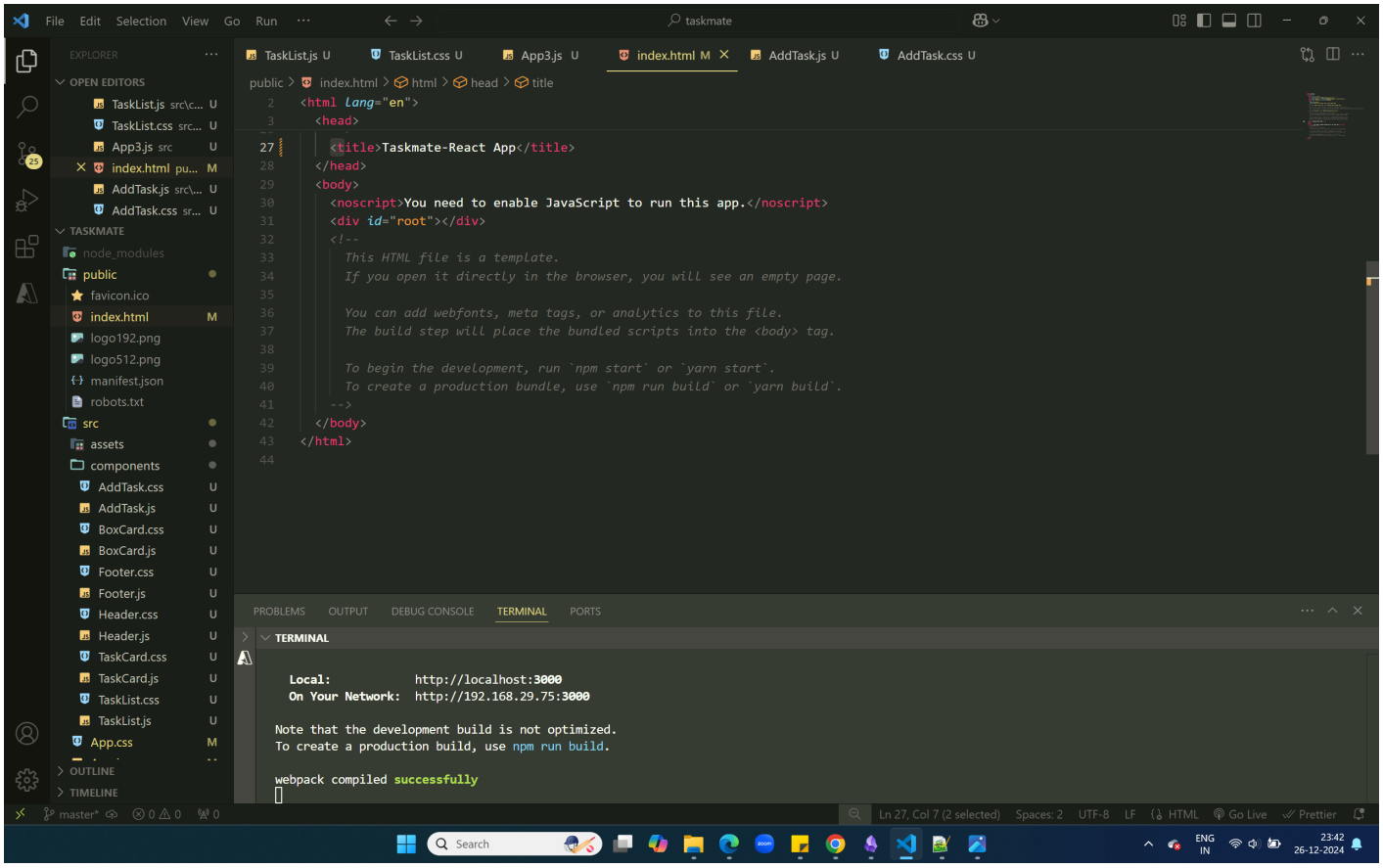


here what i am seeing is that the footer should be down only but is hanging in between for that what to do let us see
in App3.js i had added main like this and u can see CSS of main

```
import { useState } from 'react';
import './App3.css';
import { Footer } from './components/Footer';
import { Header } from './components/Header';
import { TaskList } from './components/TaskList';
import { AddTask } from './components/AddTask';

function App() {
  const [tasks, setTasks] = useState([]);

  return (
    <div className="App">
      <Header />
      <main>
      <AddTask tasks={tasks} setTasks={setTasks} />
        <TaskList tasks={tasks} setTasks={setTasks} />
        </main>
      <Footer/>
    </div>
  );
}

export default App;
```



i had changed the title also by going into index.html file okay



so now let us look into useref a new hook till now we have seen useState hook okay

here we can refer to one particular control like textbox but we cannot display it using useref hook okay let us see that

so what task i was doing for add task i can do it using useref hook also just analyze the code you will understand here i am attaching it to the control so

```
AddTask.js
----------
import { useState, useRef } from "react";
import "./AddTask.css";

export const AddTask = ({tasks, setTasks}) => {
    // const [taskValue, setTaskValue] = useState("");
    const [progress, setProgress] = useState(false);
    const taskRef = useRef("");

    // const handleChange = (event) => {
    //     console.log(taskRef.current.value)
    // }

    const handleReset = () => {
        // setTaskValue("");
        taskRef.current.value = "";
        setProgress(false);
    }

    const handleSubmit = (event) => {
        event.preventDefault();
        const task = {
            id: Math.floor(Math.random() * 10000),
            name: taskRef.current.value,
            completed: Boolean(progress)
        }
        setTasks([...tasks, task]);
        handleReset();
    }

  return (
    <section className="addtask">
        <form onSubmit={handleSubmit}>
            <input type="text" name="task" id="task" placeholder="Task Name" autoComplete="off" ref={taskRef} />
            <select onChange={(event) => setProgress(event.target.value)} value={progress}>
                <option value="false">Pending</option>
                <option value="true">Completed</option>
            </select>
            <span onClick={handleReset} className="reset">Reset</span>
            <button type="submit">Add Task</button>
        </form>
        <p>{taskRef.current.value}</p>
    </section>
  )
}
```

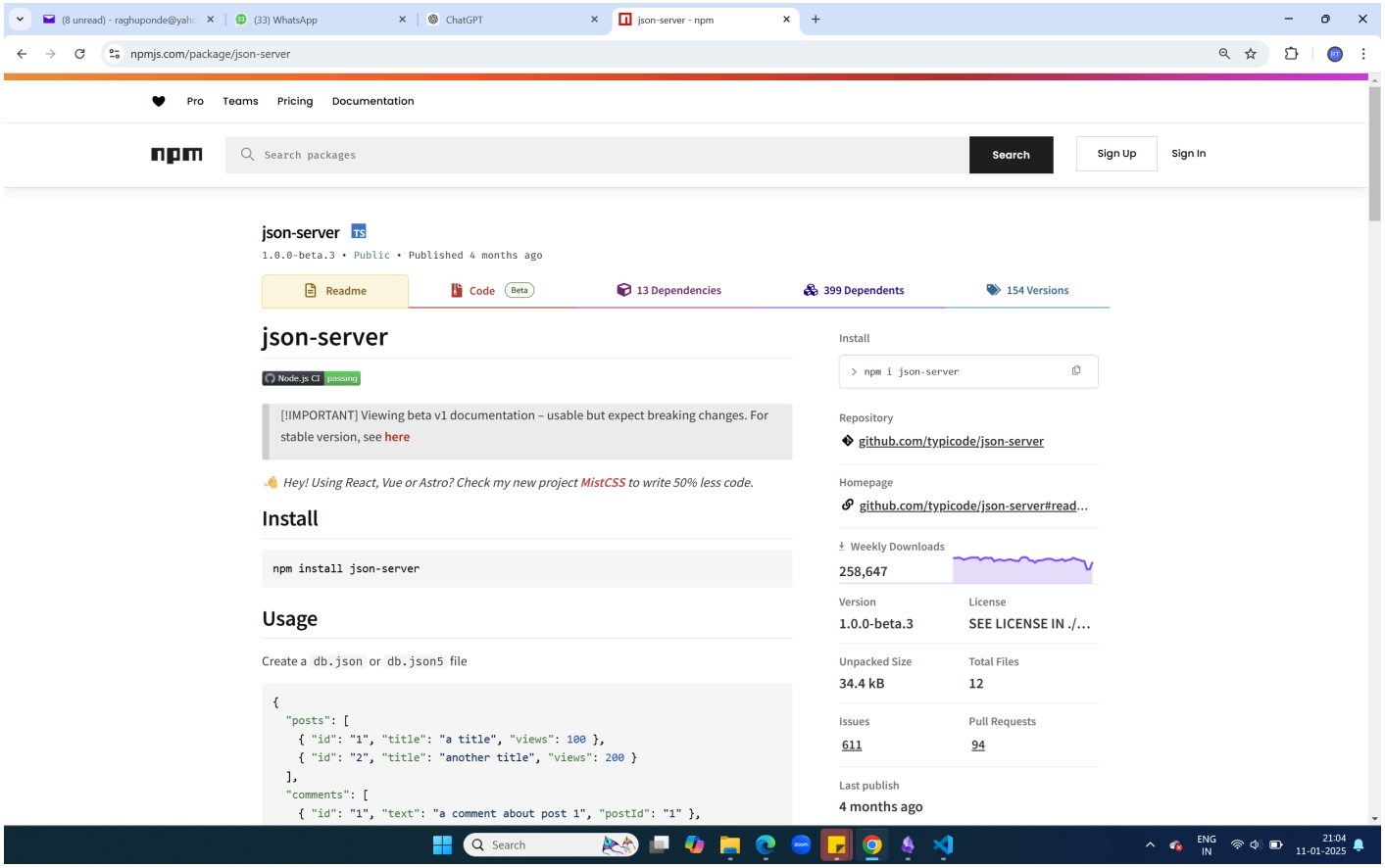
so the code is working fine now we are moving to new module and new section

Section 7 (UseEffects,useCallback,CustomHook)

Here now i will be creating an api and api is generally created at back end now i don't have to learn back end technology like asp.net core i will use fake server means some dummy kind of server here which is json server okay

https://www.npmjs.com/package/json-server

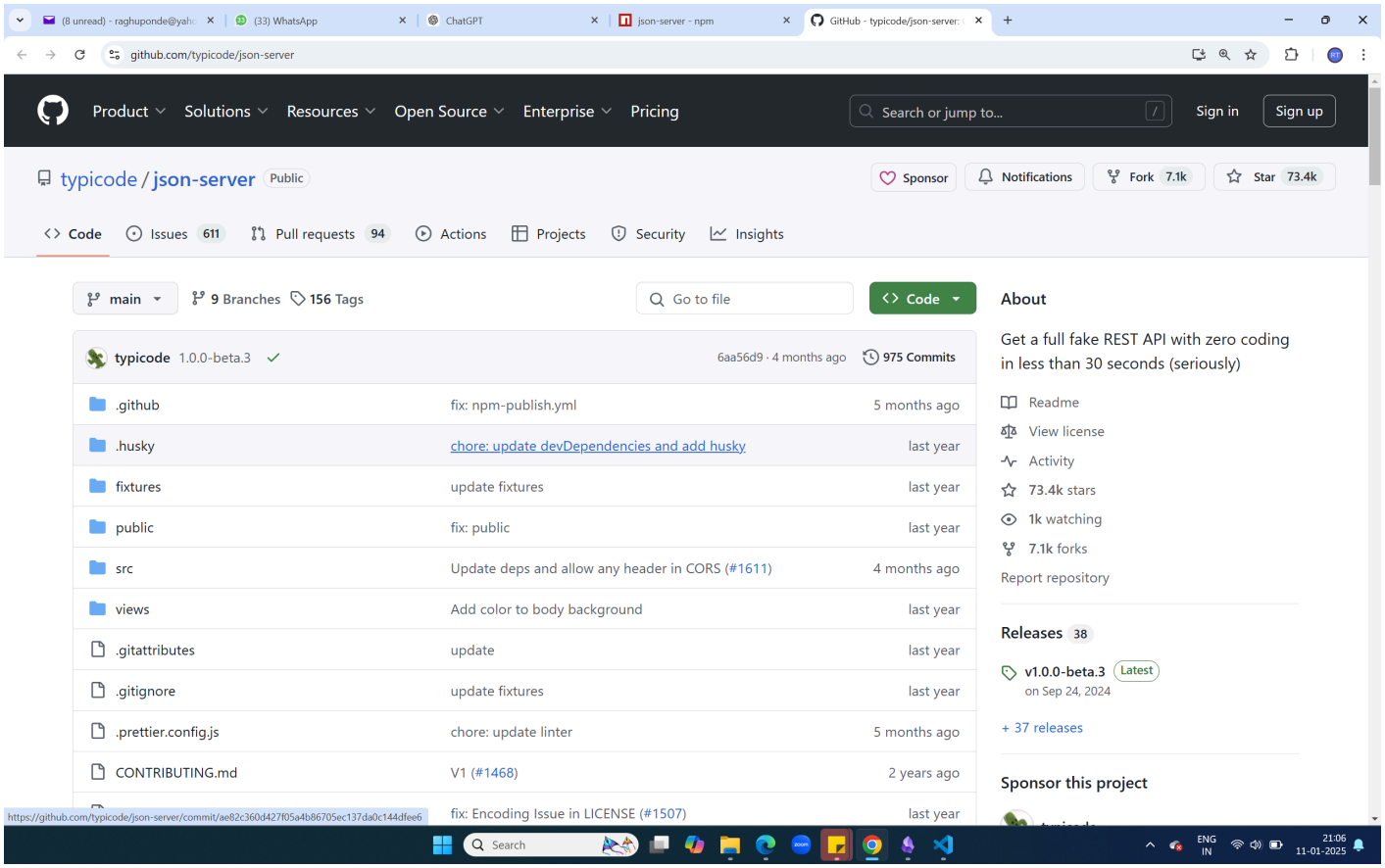
right now it is looking like this ..later on it can change



click the github link in the image you will get lot of information on the code source

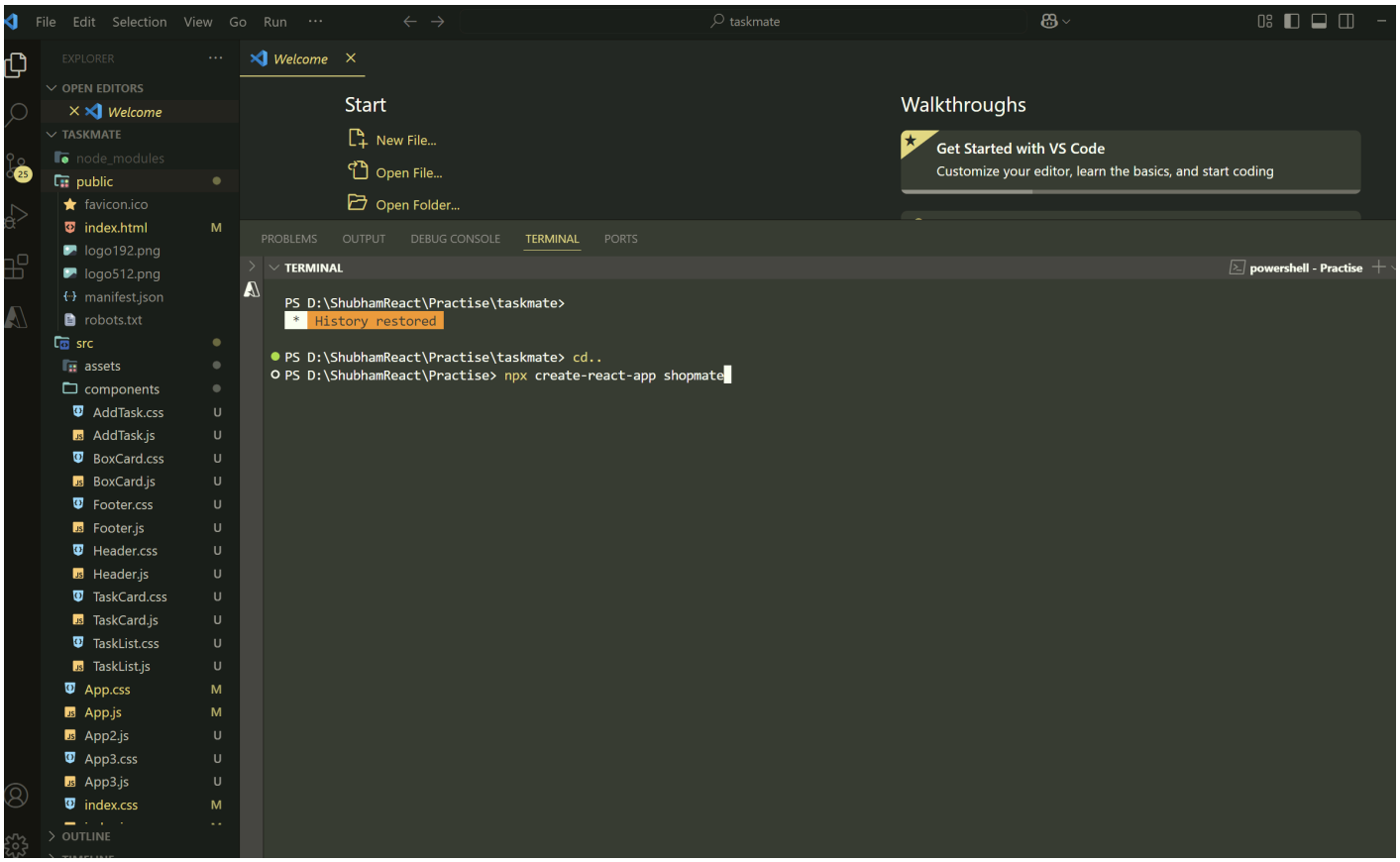
https://github.com/typicode/json-server

right now looking like this later on it can change



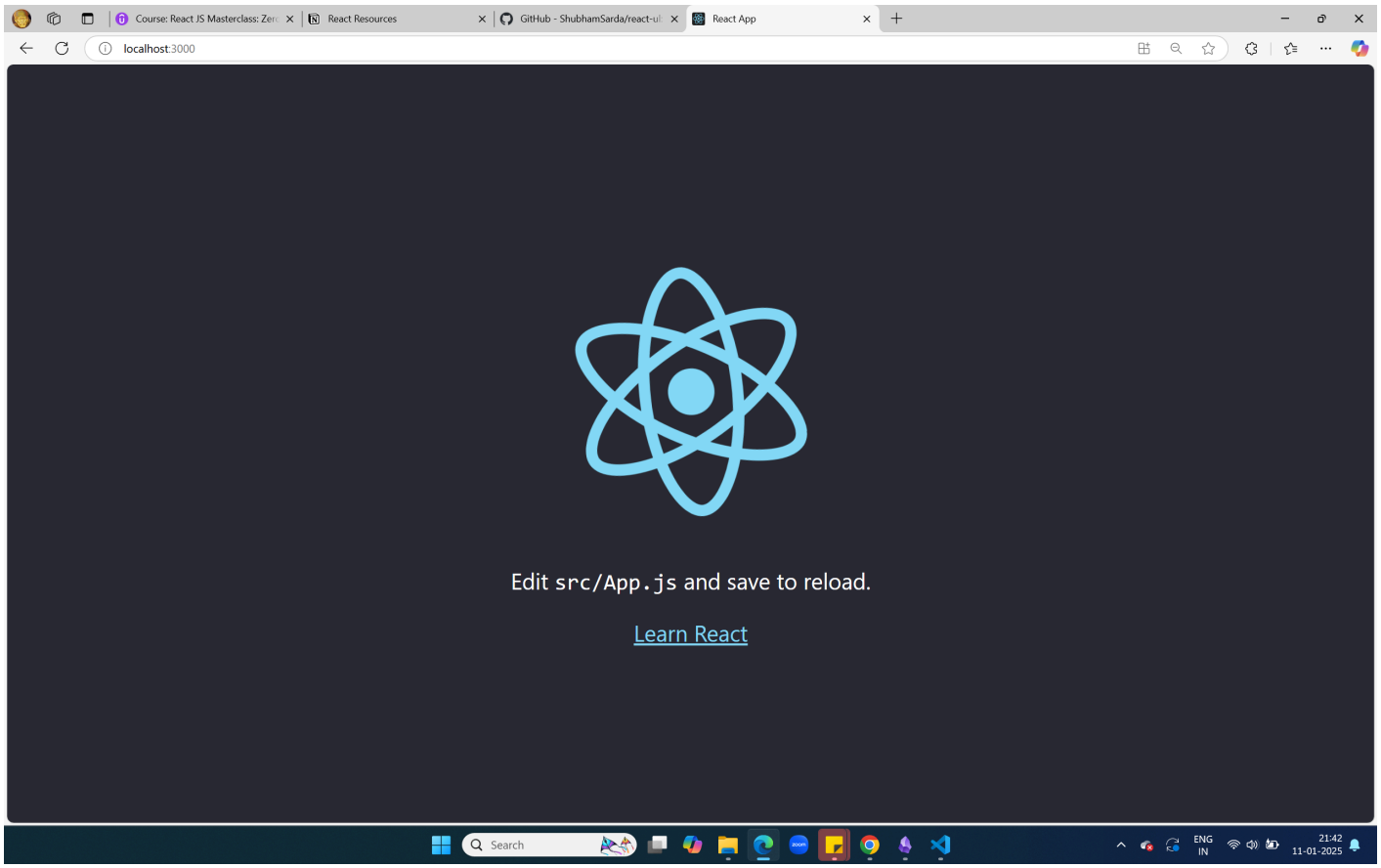
so if u go down all the steps are given how to install and move forward so

so before that lets us create an new app now so in practice folder only create a new app

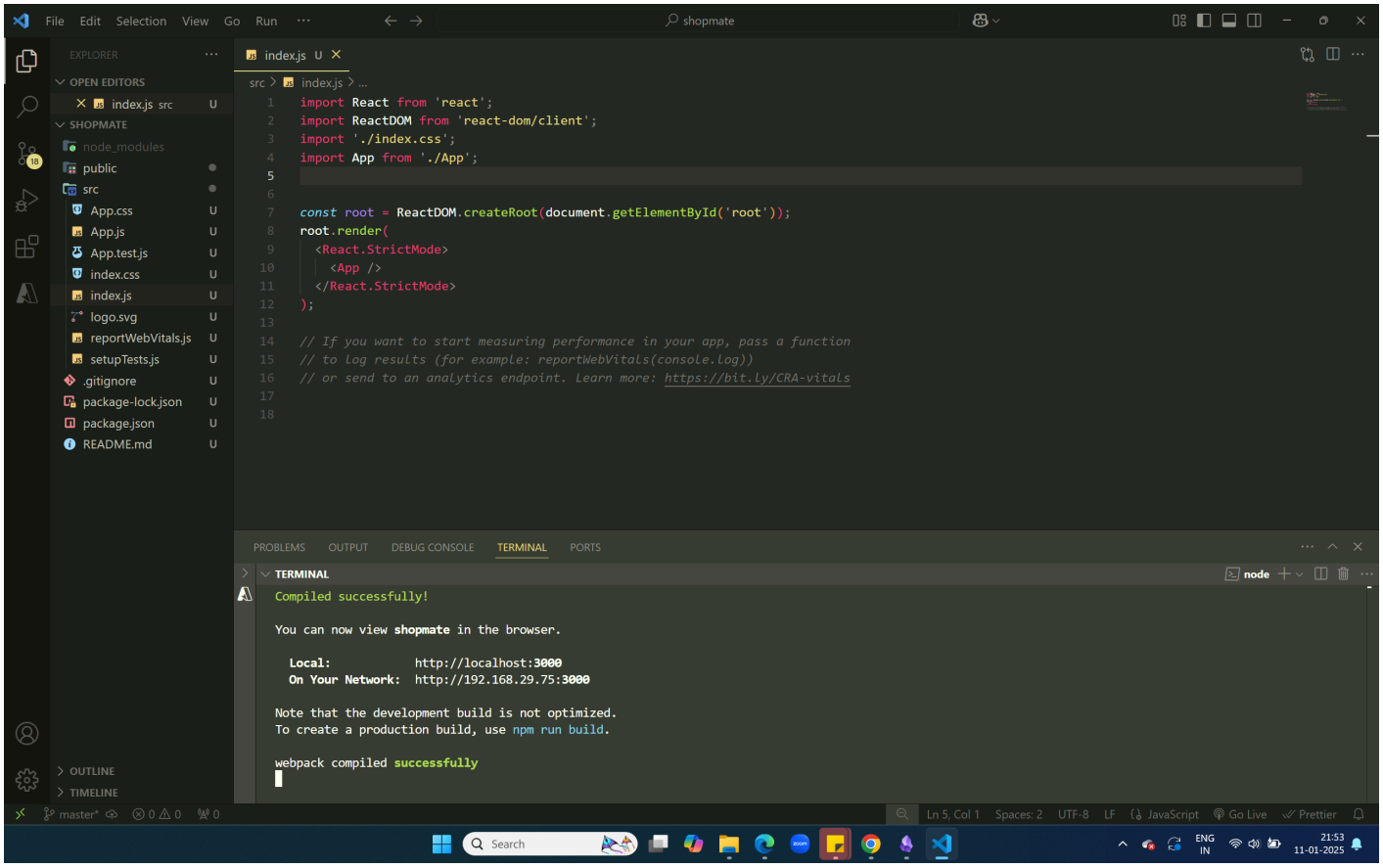


so after app is created opening the app using file new folder and you have to open the app which you have created which is shopmate.

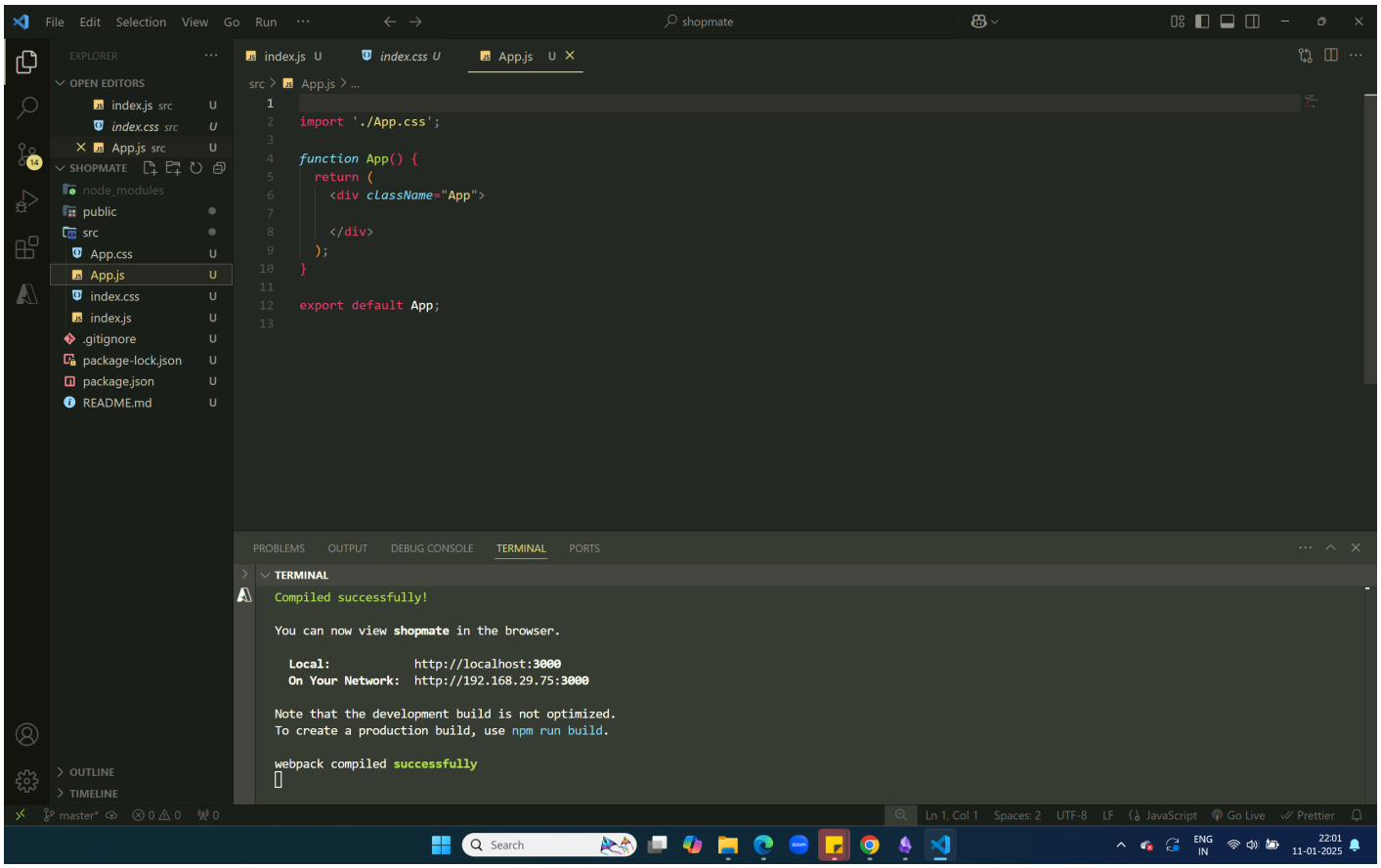
i was getting some error on web vitals so i had removed it where ever reference is there and now my application is running fine



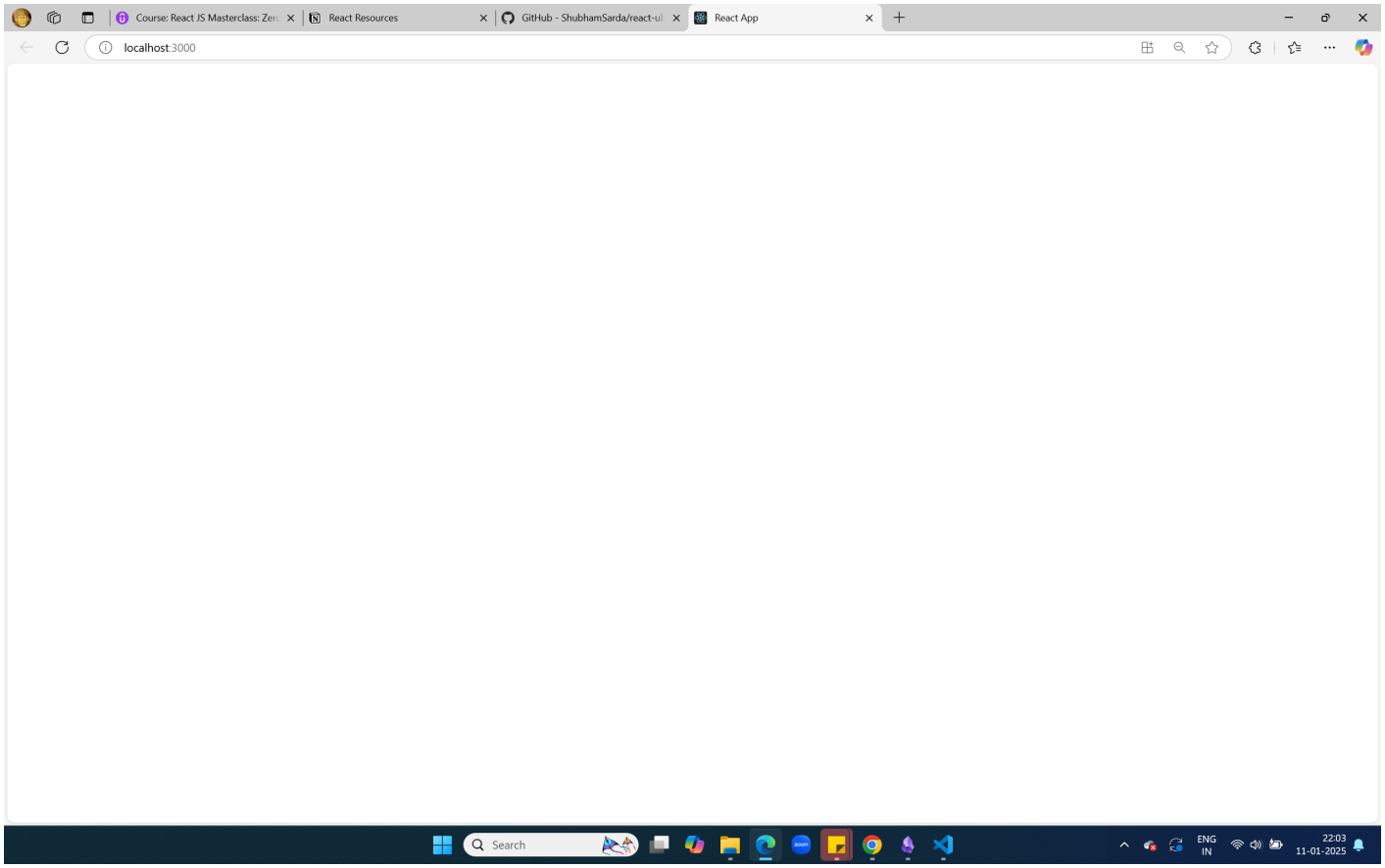
so above .gitignore delete 3 files



logo ,reportwebvitals and setuptest.js files these 3 files and also Apptest.js so i am doing some clean up of files here remove the entire header from App.js file and remove unwanted code in index.js as well like web vitals etc. and okay .



so after removing all and doing set up it looks like this next



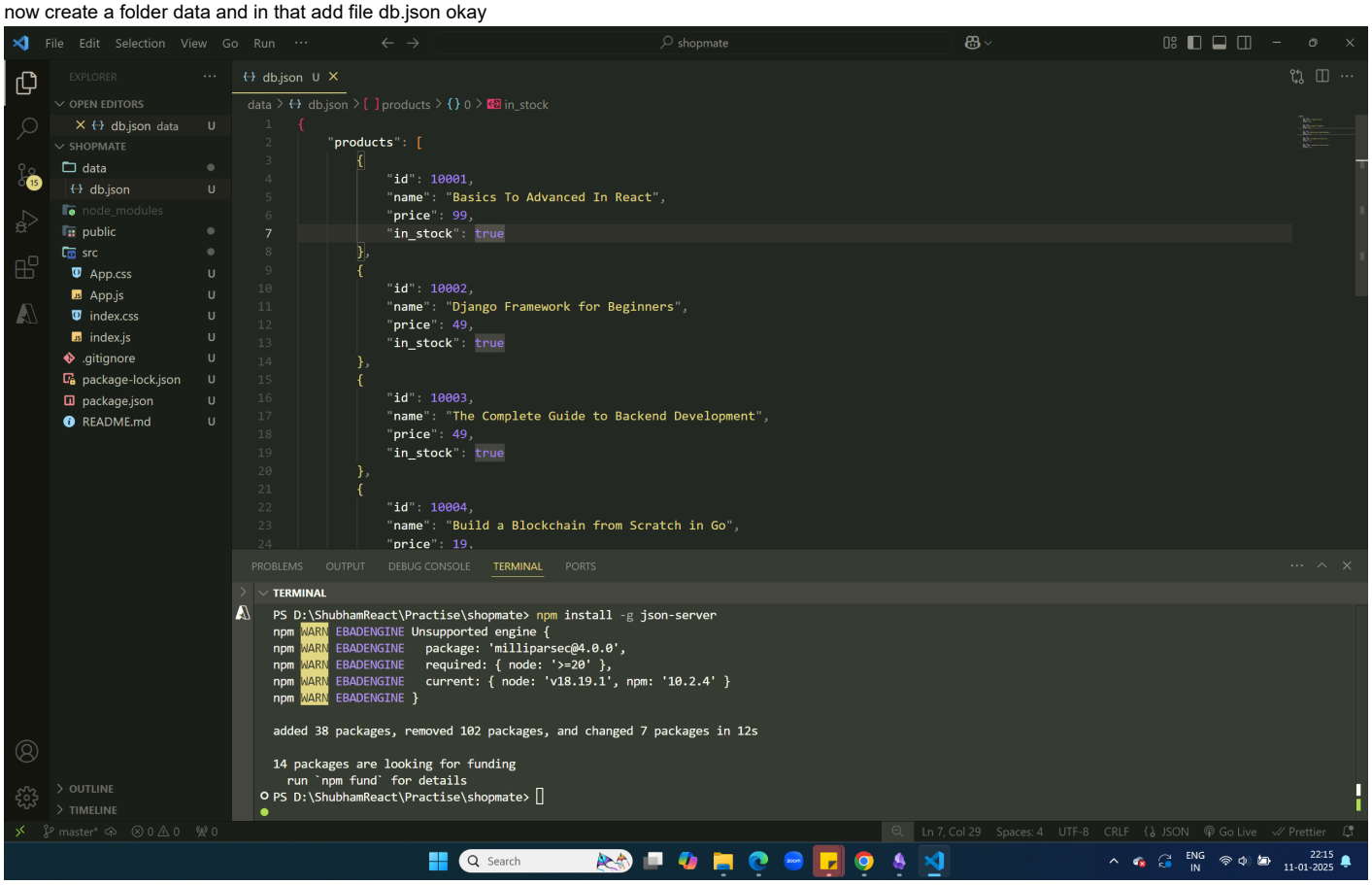
will get empty as nothing is there now written then now

now let us install json server now

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

TERMINAL
PS D:\ShubhamReact\Practise\shopmate> npm install -g json-server
npm WARN EBADENGINE Unsupported engine {
npm WARN EBADENGINE   package: 'milliparsec@4.0.0',
npm WARN EBADENGINE   required: { node: '>=20' },
npm WARN EBADENGINE   current: { node: 'v18.19.1', npm: '10.2.4' }
npm WARN EBADENGINE }

added 38 packages, removed 102 packages, and changed 7 packages in 12s

14 packages are looking for funding
  run `npm fund` for details
PS D:\ShubhamReact\Practise\shopmate>
```

if u see in package.json i cannot see the json server as i have installed globally but i can access it .

now create a folder data and in that add file db.json okay



and the data which is there in db.json file is like this

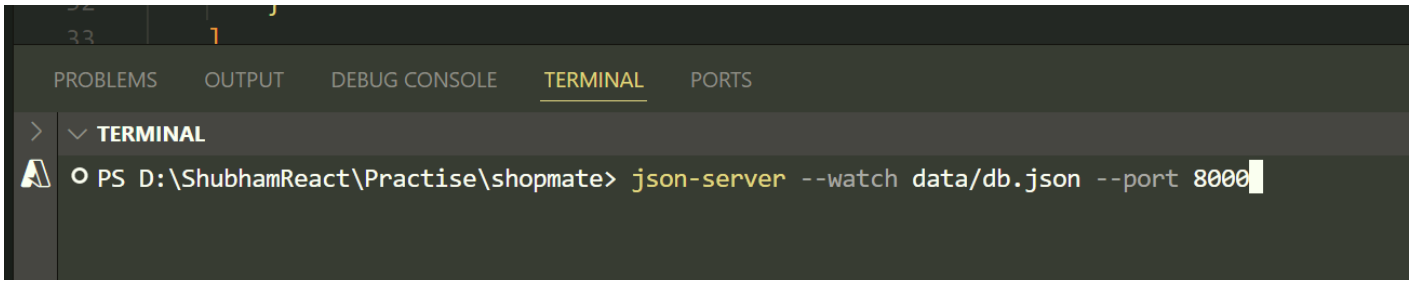
```
{
    "products": [
        {
            "id": 10001,
            "name": "Basics To Advanced In React",
            "price": 99,
            "in_stock": true
        },
        {
            "id": 10002,
            "name": "Django Framework for Beginners",
            "price": 49,
            "in_stock": true
```

```
        },
        {
            "id": 10003,
            "name": "The Complete Guide to Backend Development",
            "price": 49,
            "in_stock": true
        },
        {
            "id": 10004,
            "name": "Build a Blockchain from Scratch in Go",
            "price": 19,
            "in_stock": false
        },
        {
            "id": 10005,
            "name": "Build a Blockchain from Scratch in React",
            "price": 199,
            "in_stock": false
        }
    ]
}
```

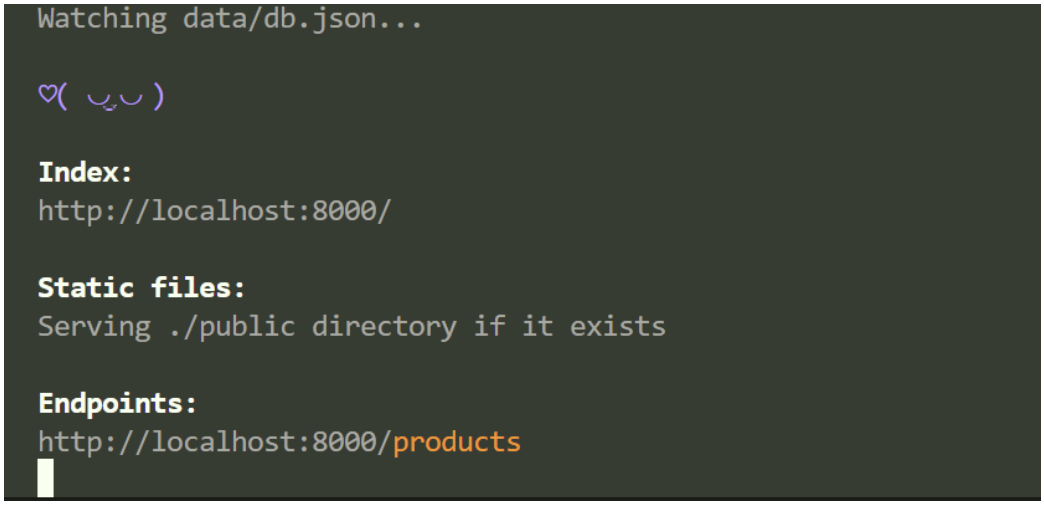

so you can say this as an end point through which our front end interacts using react okay

now i have to run json server so another terminal i will open now

and will type the following command there

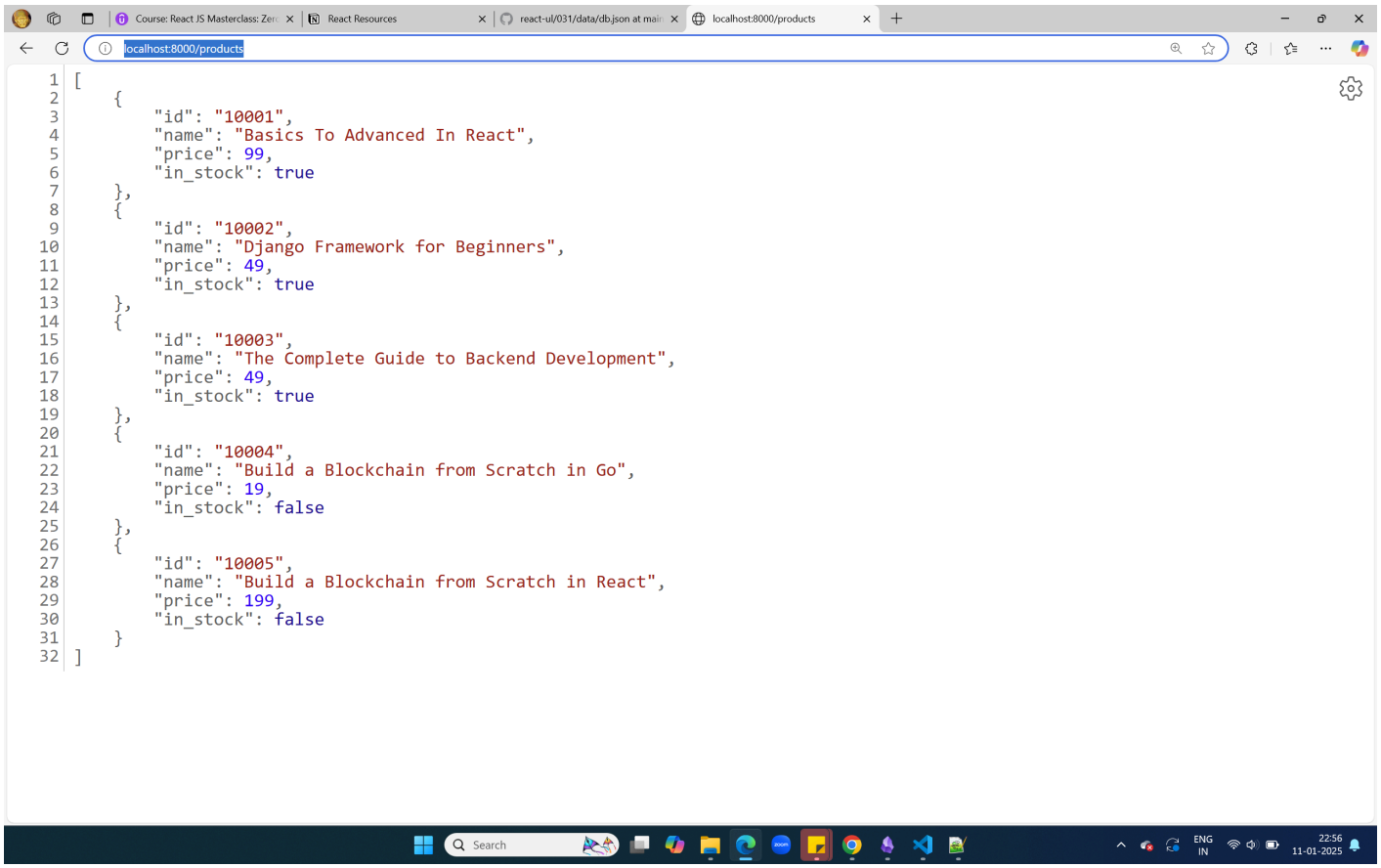


json-server --watch data/db.json --port 8000

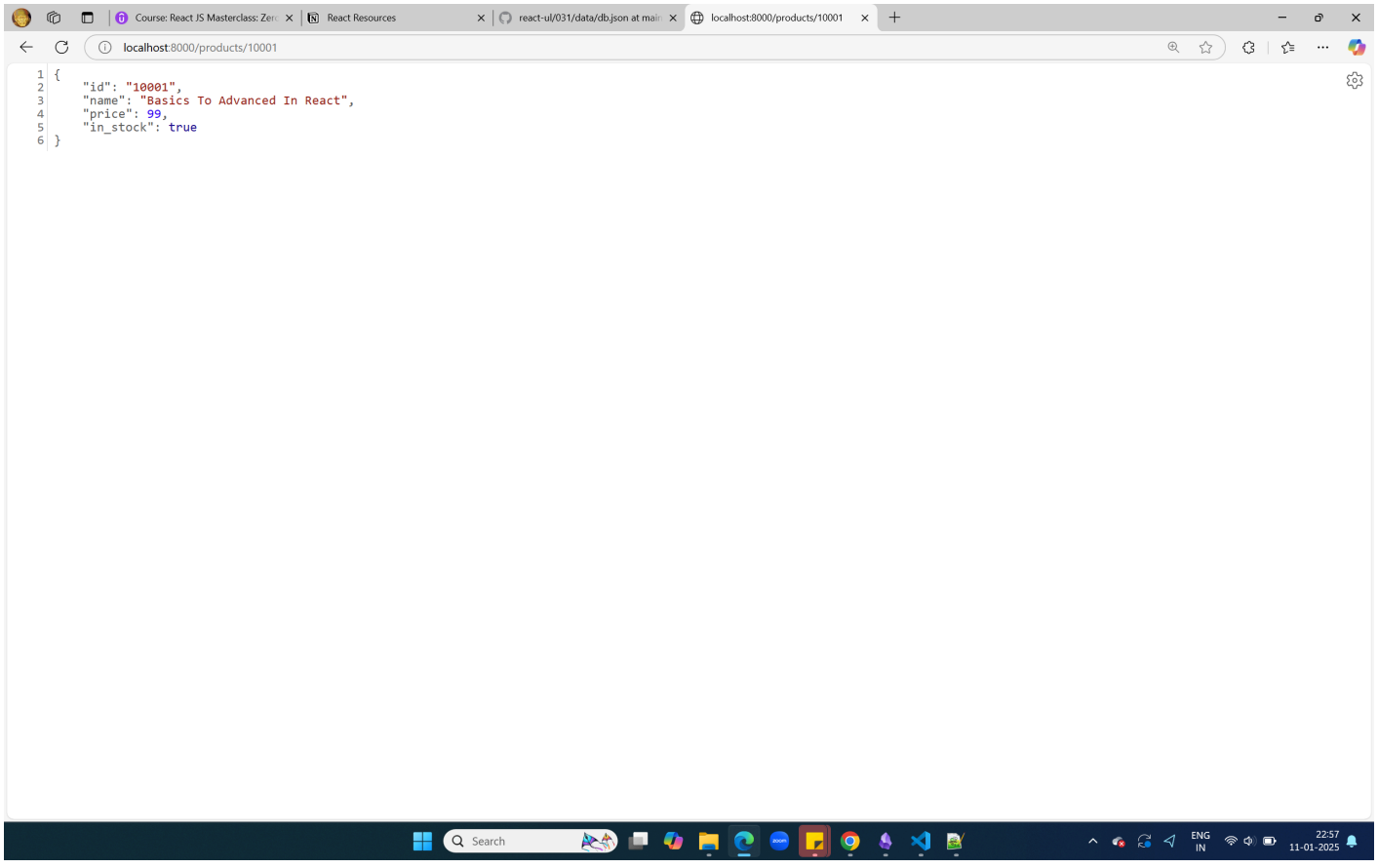


so take the endpoint and paste it and see you can see the data

localhost:8000/products



for single product

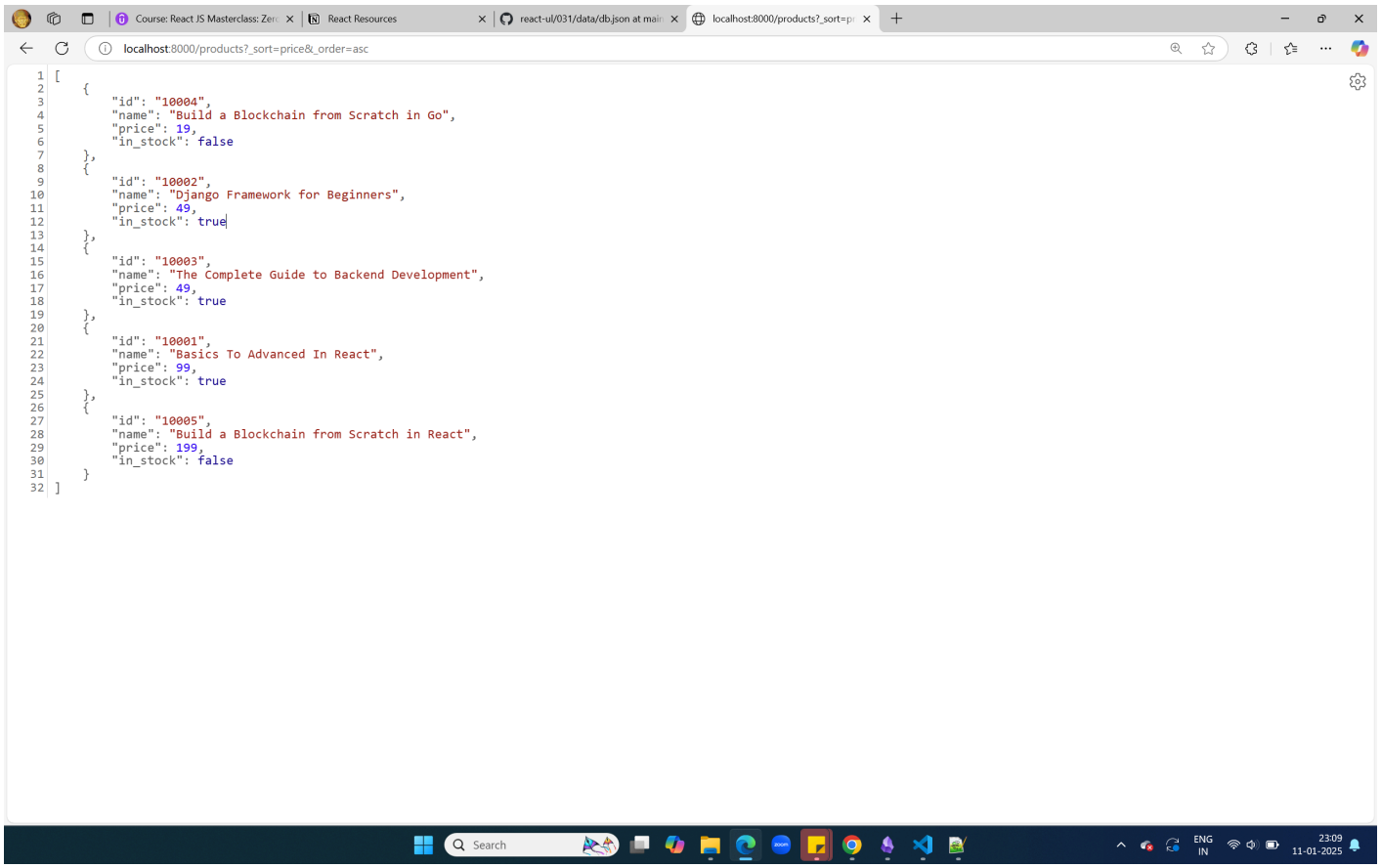


so things will go like this

and still i can filter the result like this

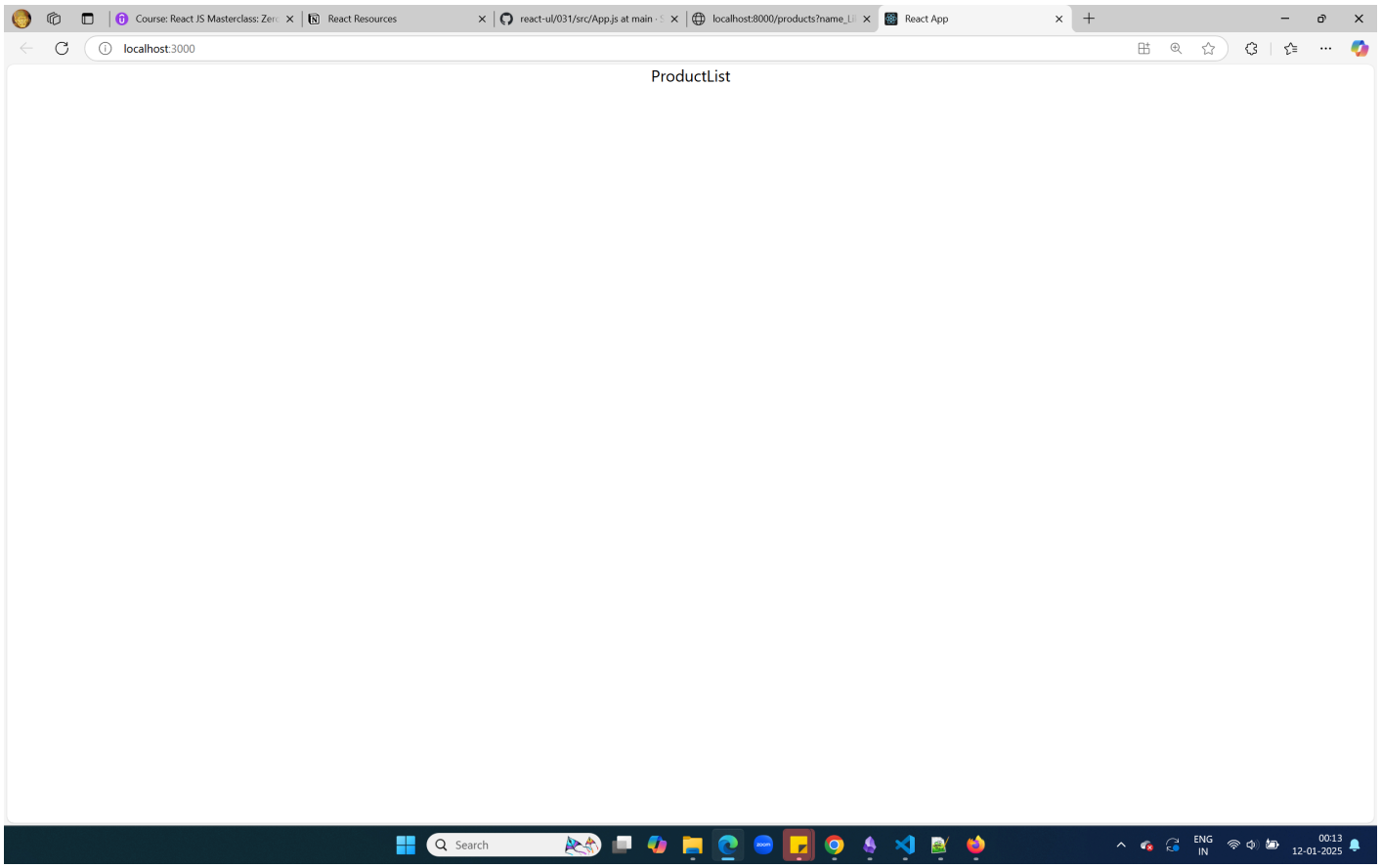
localhost:8000/products?_sort=price&_order=asc

and output will be like this



now create a folder components and in that ProductList.js file and do rafc there and add this ProductList.js into the App.js file so in src folder add components folder so now go ahead

so after doing all this i can see the ProductList.js

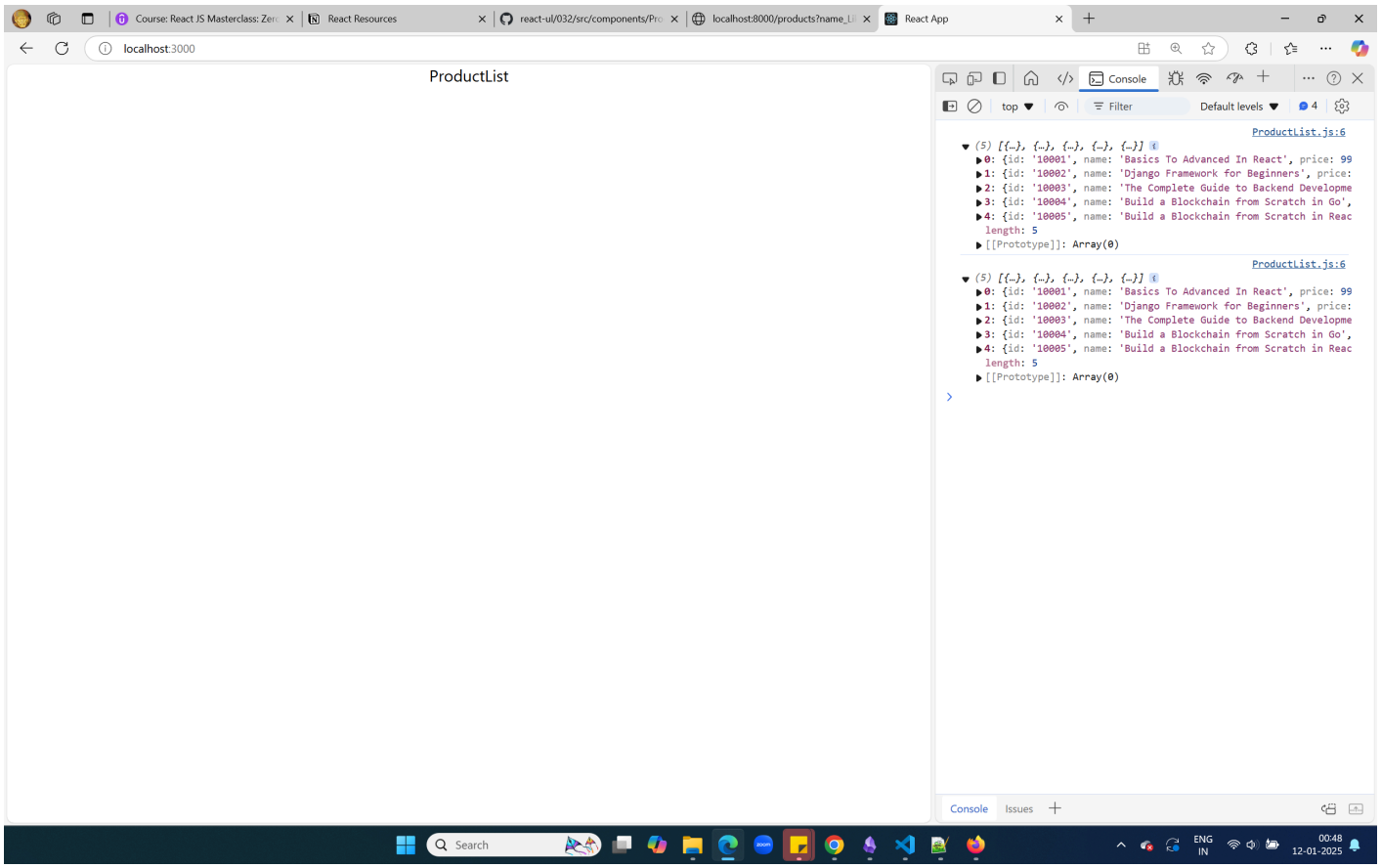


Now ProductList.js code is like this

```
import { useState } from "react"

export const ProductList = () => {
    const [products, setProducts] = useState([]);

    fetch("http://localhost:8000/products")
        .then(response => response.json())
        .then(data => console.log(data));

    return (
        <div>ProductList</div>
    )
}
```

```
App.js code
------------
import './App.css';
import { ProductList } from './components/ProductList';
function App() {
  return (
    <div className="App">
     <ProductList />
    </div>
  );
}

export default App;
```

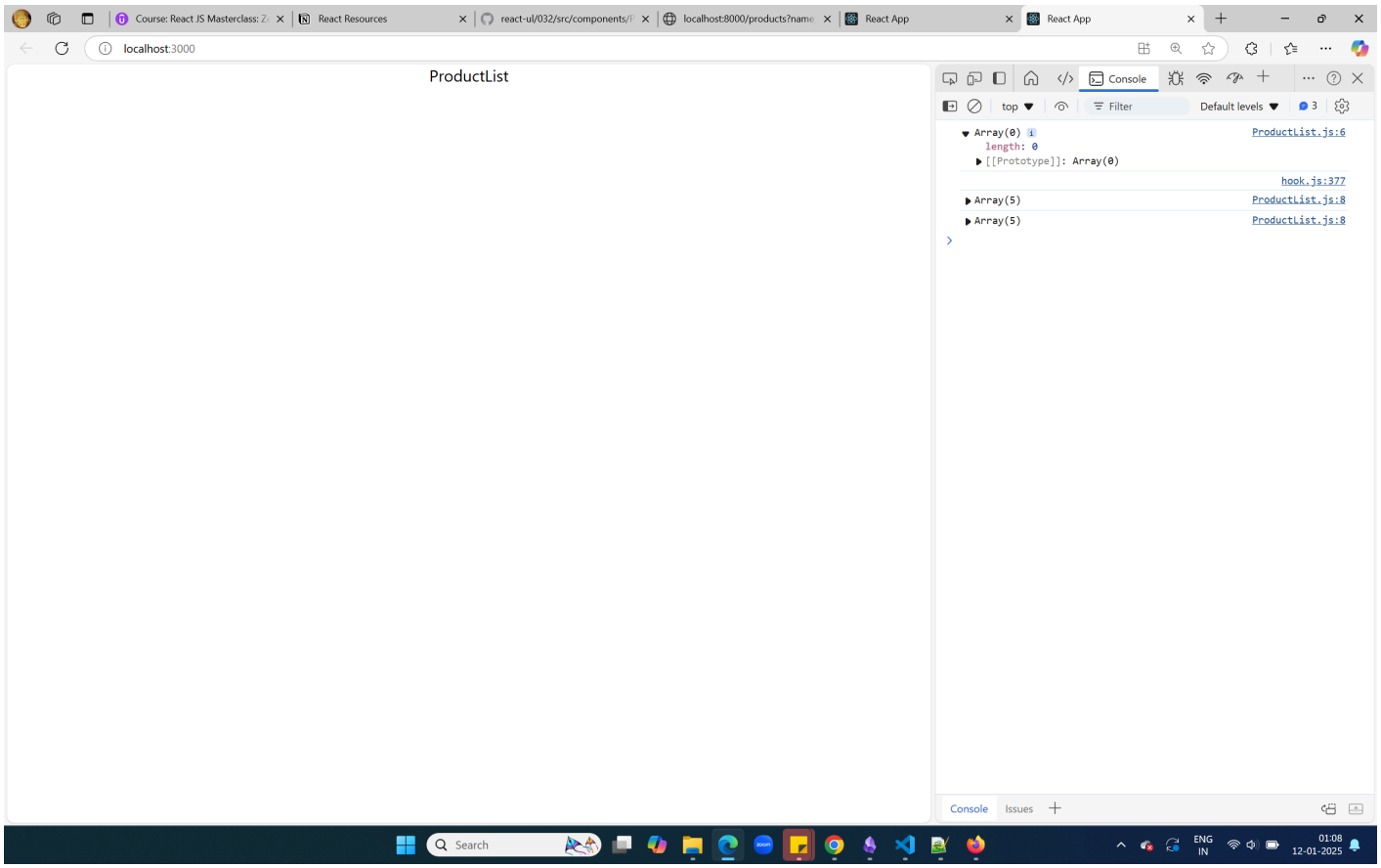
so i can see the elements in console log



Now for this code below

```
import { useState } from "react"

export const ProductList = () => {
    const [products, setProducts] = useState([]);

    console.log(products);

    fetch("http://localhost:8000/products")
        .then(response => response.json())
        .then(data => console.log(data));

    return (
        <div>ProductList</div>
    )
}
```

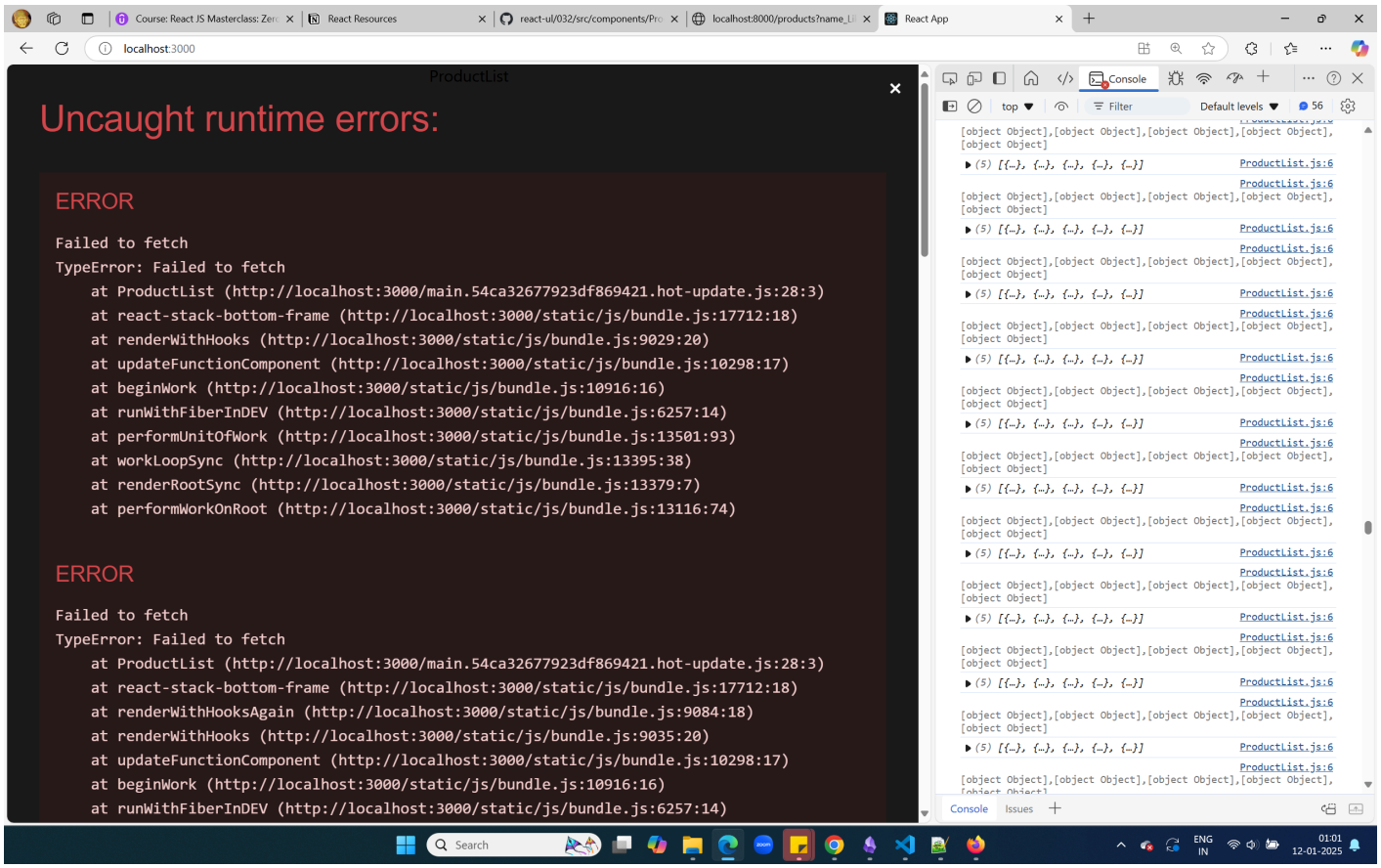
see for the above code i am first getting empty array as i had not fetched yet now



so now inside products i want to store some information and that i can update it using setProducts for that the code is below and that products i want to display so when i do below coding

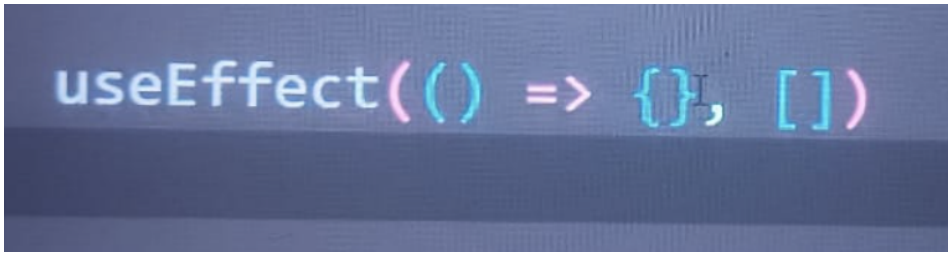
```
import { useState } from "react"

export const ProductList = () => {
    const [products, setProducts] = useState([]);

    console.log(products);

    fetch("http://localhost:8000/products")
        .then(response => response.json())
        .then(data => setProducts(data));

    return (
        <div>ProductList</div>
    )
}
```

for the above change of code infinite loop is running here every time i am updating the product it is again rendering it and going into infinite loop



so it was in infinite loop when i stopped i can see the above image okay .

so here it is going for infinite loop whenever i update any state it will reevaluate everything means entire component and re-render entire component okay and goes in infinite loop okay
call the api again ,set the product again and display again
since it is setting the product again it will reevaluate again okay



so to overcome that we use useEffect here it takes two parameters one is function and another is when we need to run this function whether we need to run it once or we need to run this multiple times or we need to take care of any other dependencies so we need to pass that dependency list into the square bracket means for that only i will work or any change happens there then only i will work okay

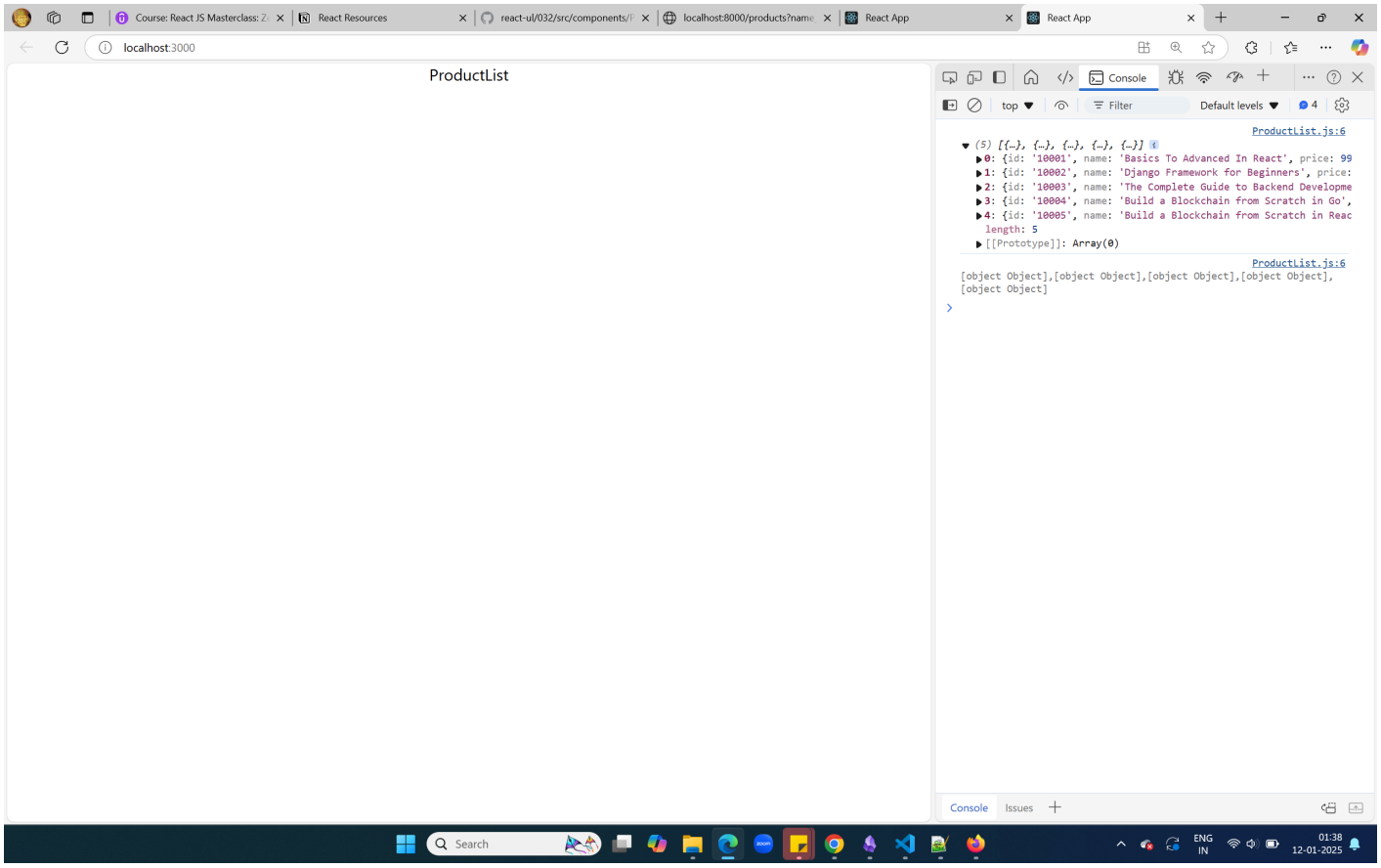
so inside the flower bracket i can write any action okay

when the array is empty the use effect will be called only once

```
import { useEffect, useState } from "react";

export const ProductList = () => {
    const [products, setProducts] = useState([]);

    console.log(products)

    useEffect(() => {
        fetch("http://localhost:8000/products")
            .then(response => response.json())
            .then(data => setProducts(data));
    }, []);

    return (
```

```
        <div>ProductList</div>
    );
};
```

Now for the above code it is not going into infinite loop okay



Now i am using the products to display in page

```
ProductList.js
------------------
import { useEffect, useState } from "react"

export const ProductList = () => {
    const [products, setProducts] = useState([]);

    useEffect(() => {
        fetch("http://localhost:8000/products")
        .then(response => response.json())
        .then(data => setProducts(data));
    }, []);

  return (
    <section>
        { products.map((product) => (
            <div className="card" key={product.id}>
                <p className="id">{product.id}</p>
                <p className="name">{product.name}</p>
                <p className="info">
                    <span>${product.price}</span>
                    <span className={product.in_stock ? "instock" : "unavailable"}>{product.in_stock ? "In Stock" : "Unavailable"}
</span>
                </p>
            </div>
        )) }
    </section>
  )
}
```

index.css

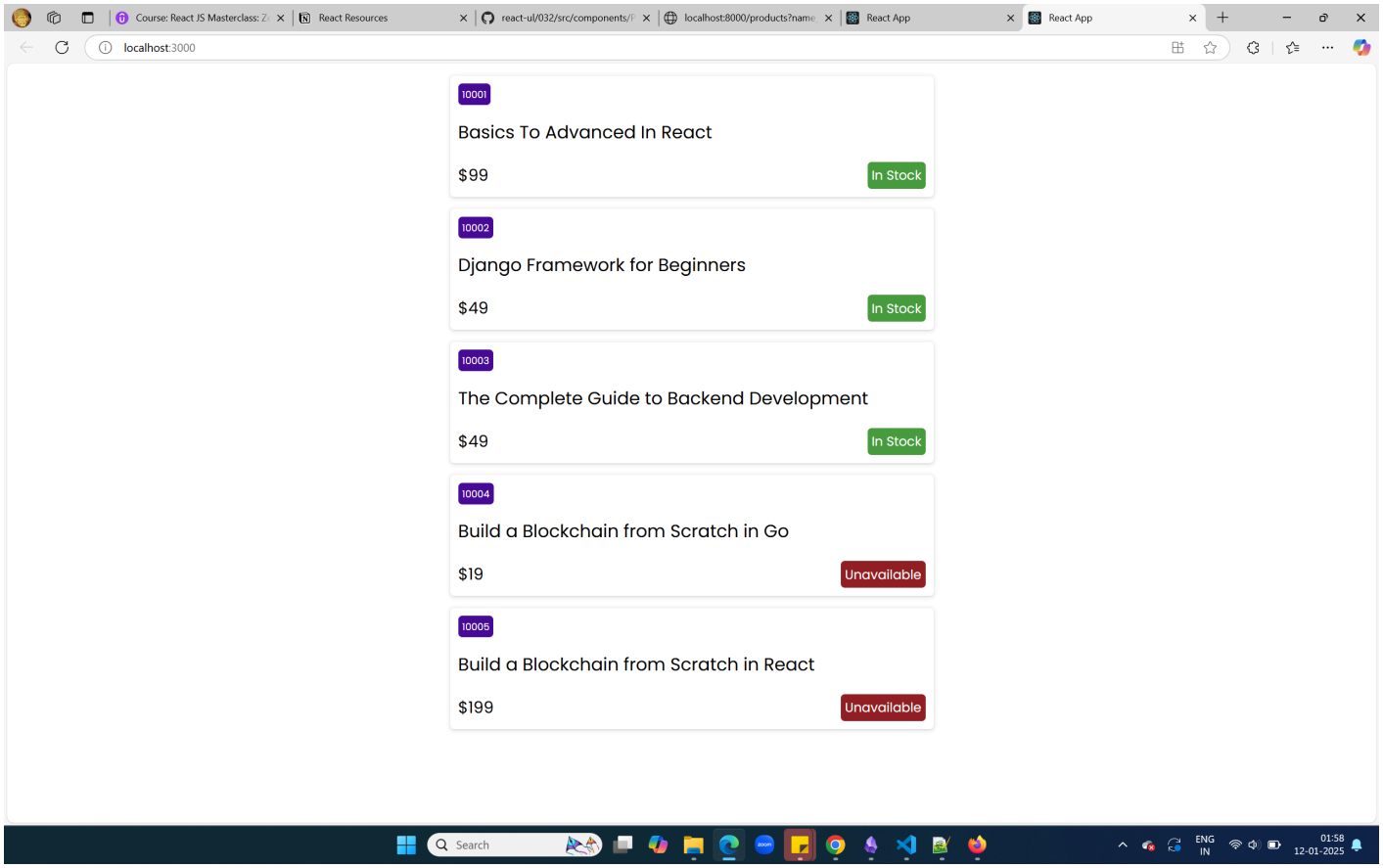
```
@import url('https://fonts.googleapis.com/css2?
family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&f
amily=Roboto&display=swap');

* {
  margin: 0px;
  padding: 0px;
  box-sizing: border-box;
  font-family: 'Poppins', sans-serif;
}
```

App.css

```
.App {
  max-width: 1200px;
  margin: auto;
  padding: 0px 15px;
}

h1{
  text-align: center;
  margin: 30px auto;
}

.heading {
  font-size: 22px;
  align-items: center;
}

.card {
  margin: 15px auto;
  padding: 10px;
  box-shadow: rgba(0, 0, 0, 0.16) 0px 1px 4px;
  max-width: 600px;
  border-radius: 5px;
}

.card .id {
  background-color: #4e06a0;
  font-size: 12px;
  color: #FFFFFF;
  border-radius: 5px;
  margin: 10px 0px;
  padding: 5px;
  display: inline;
}

.card .name {
  font-size: 22px;
  font-weight: 400;
  margin: 20px 0px;
}

.card .info {
  font-size: 20px;
  display: flex;
  justify-content: space-between;
  align-items: center;
}

.card .instock {
  background-color: #06a02d;
  font-size: 16px;
  color: #FFFFFF;
  border-radius: 5px;
  padding: 5px;
  display: inline;
}

.card .unavailable {
  background-color: #a00620;
  font-size: 16px;
```

```
  color: #FFFFFF;
  border-radius: 5px;
  padding: 5px;
  display: inline;
}

button {
  border: 0px;
  border-radius: 5px;
  background-color: #0c68a1;
  color: #FFFFFF;
  cursor: pointer;
  padding: 10px;
  font-size: 20px;
}

button.onlyStock {
  background-color: #0ca14f;
}

button.all {
  background-color: #065ba0;
}
```

and the output is like this



Now in the above coding of video square bracket is empty which means logic inside the function will execute only once

```
ProductList.js
------------------
import { useEffect, useState } from "react"

export const ProductList = () => {
    const [products, setProducts] = useState([]);

    useEffect(() => {
        fetch("http://localhost:8000/products")
        .then(response => response.json())
        .then(data => setProducts(data));
```

```
    }, []);

  return (
    <section>
        { products.map((product) => (
            <div className="card" key={product.id}>
                <p className="id">{product.id}</p>
                <p className="name">{product.name}</p>
                <p className="info">
                    <span>${product.price}</span>
                    <span className={product.in_stock ? "instock" : "unavailable"}>{product.in_stock ? "In Stock" : "Unavailable"}
</span>
                </p>
            </div>
        )) }
    </section>
  )
}
```

in the above coding it is taking the URL and data from that URL is displayed once in the page using setProducts use state value okay now what i want to do is in the square bracket which is empty i will pass the dependency means some parameter i will pass and if that parameter is changed then the code should execute that is what i will do so above coding only fetching of data is happening and now what i will do is on button click i will change the parameter and when the parameter is changed the code of use effect will execute i can put multiple dependency as well means by putting comma after passing first dependency but same unnecessary logic also execute which is not needed so separate use effect i will use here now

so now what i am doing is i will show all values of URL when u click all and in stock if in stock is clicked okay and there it will be change in the URL and now in the same manner i will console log counter dependency as well where i am incrementing the counter on click of button and u can see here i had written all anonymous functions here okay

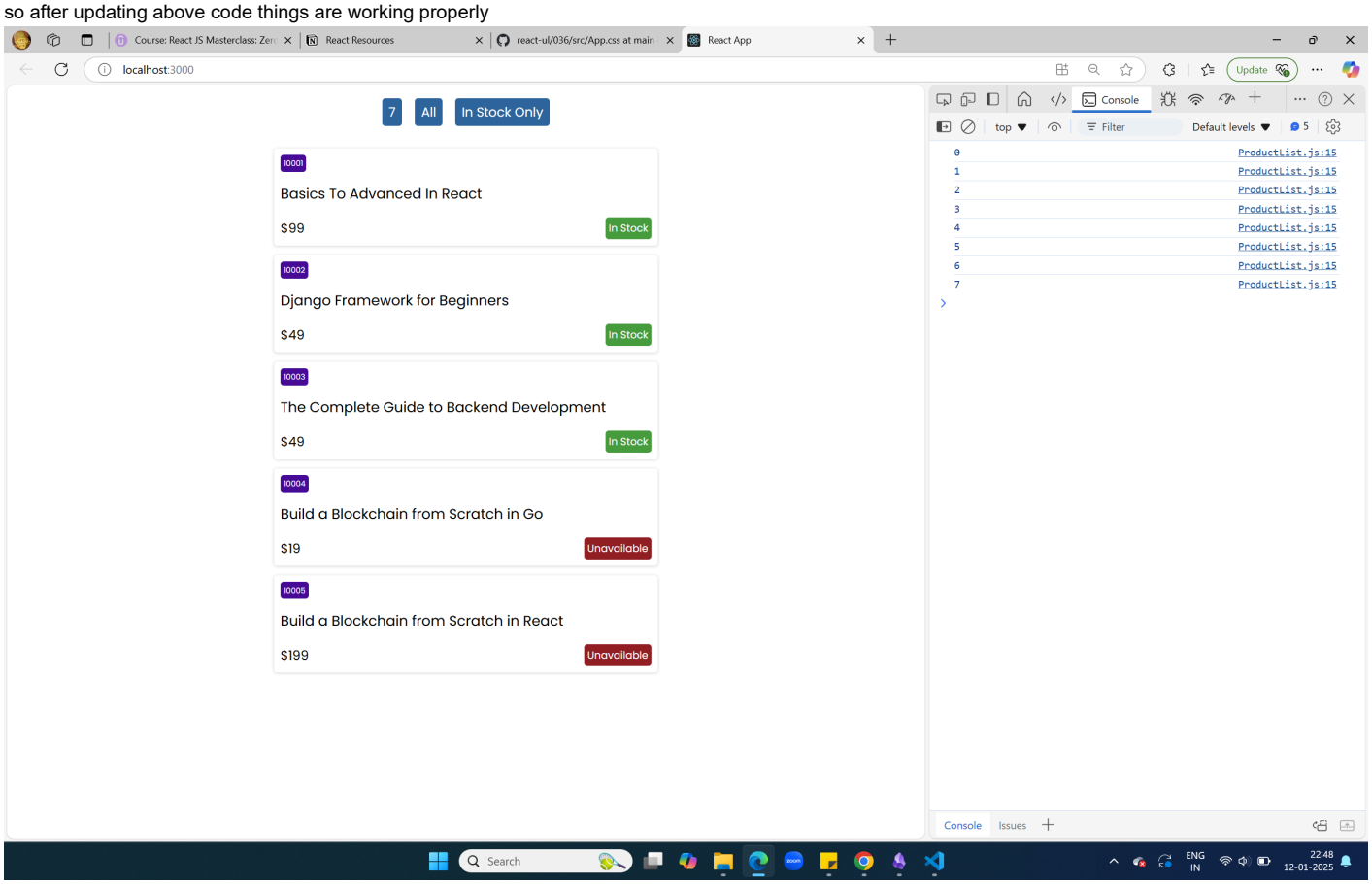
```
ProductList.js
-----------------
import { useEffect, useState } from "react"

export const ProductList = () => {
    const [products, setProducts] = useState([]);
    const [url, setUrl] = useState("http://localhost:8000/products");
    const [counter, setCounter] = useState(0);

    useEffect(() => {
        fetch(url)
        .then(response => response.json())
        .then(data => setProducts(data));
    }, [url]);

    useEffect(() => {
        console.log(counter);
    }, [counter]);

  return (
    <section>
        <div className="filter">
            <button onClick={() => setCounter(counter + 1)}>{counter}</button>
            <button onClick={() => setUrl("http://localhost:8000/products")}>All</button>
            <button onClick={() => setUrl("http://localhost:8000/products?in_stock=true")}>In Stock Only</button>
        </div>

        { products.map((product) => (
            <div className="card" key={product.id}>
                <p className="id">{product.id}</p>
                <p className="name">{product.name}</p>
                <p className="info">
                    <span>${product.price}</span>
                    <span className={product.in_stock ? "instock" : "unavailable"}>{product.in_stock ? "In Stock" : "Unavailable"}
</span>
                </p>
            </div>
        )) }
    </section>
  )
}
```

App.css

```
.App {
  max-width: 1200px;
  margin: auto;
  padding: 0px 15px;
}

h1{
  text-align: center;
  margin: 30px auto;
}

.heading {
  font-size: 22px;
  align-items: center;
}

.filter{
  display: flex;
  justify-content: center;
}

.card {
  margin: 15px auto;
  padding: 10px;
  box-shadow: rgba(0, 0, 0, 0.16) 0px 1px 4px;
  max-width: 600px;
  border-radius: 5px;
}

.card .id {
  background-color: #4e06a0;
  font-size: 12px;
  color: #FFFFFF;
  border-radius: 5px;
  margin: 10px 0px;
  padding: 5px;
  display: inline;
}

.card .name {
  font-size: 22px;
  font-weight: 400;
  margin: 20px 0px;
}

.card .info {
  font-size: 20px;
  display: flex;
  justify-content: space-between;
  align-items: center;
}

.card .instock {
  background-color: #06a02d;
  font-size: 16px;
  color: #FFFFFF;
  border-radius: 5px;
  padding: 5px;
  display: inline;
}

.card .unavailable {
  background-color: #a00620;
  font-size: 16px;
  color: #FFFFFF;
  border-radius: 5px;
  padding: 5px;
  display: inline;
}

button {
  border: 0px;
```

```
  border-radius: 5px;
  background-color: #0c68a1;
  color: #FFFFFF;
  cursor: pointer;
  padding: 7px 10px;
  font-size: 20px;
  display: inline-block;
  margin: 20px 10px;
}

button.onlyStock {
  background-color: #037c3a;
}

button.all {
  background-color: #065ba0;
}
```

so after updating above code things are working properly



now i want to remove counter logic first from the earlier code as i want to concentrate on URL code so which looks like this okay

```
import { useEffect, useState } from "react"

export const ProductList = () => {
    const [products, setProducts] = useState([]);
    const [url, setUrl] = useState("http://localhost:8000/products");

    useEffect(() => {
        fetch(url)
        .then(response => response.json())
        .then(data => setProducts(data));
    }, [url]);


  return (
    <section>
        <div className="filter">

            <button onClick={() => setUrl("http://localhost:8000/products")}>All</button>
```

```
            <button onClick={() => setUrl("http://localhost:8000/products?in_stock=true")}>In Stock Only</button>
        </div>

        { products.map((product) => (
            <div className="card" key={product.id}>
                <p className="id">{product.id}</p>
                <p className="name">{product.name}</p>
                <p className="info">
                    <span>${product.price}</span>
                    <span className={product.in_stock ? "instock" : "unavailable"}>{product.in_stock ? "In Stock" : "Unavailable"}
</span>
                </p>
            </div>
        )) }
    </section>
  )
}
```

Now use Effect is having some code all that code i want to module it or keep it in one function like this and i am calling it

```
useEffect(() => {
    const fetchProducts = async () => {
        const response = await fetch(url);
        const data = await response.json();
        setProducts(data);
    };

    fetchProducts();
}, [url]);
```

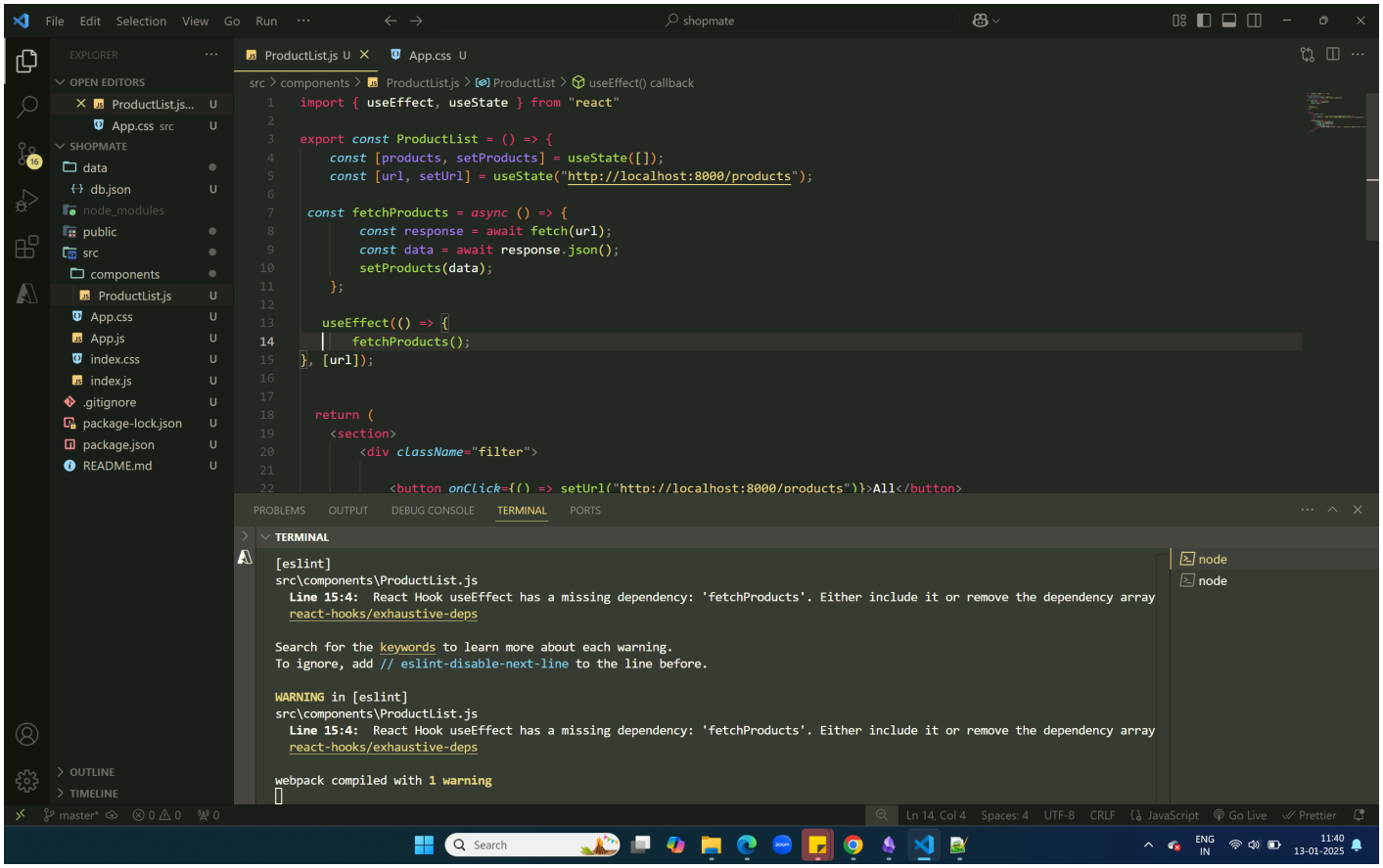
I am using here async function over here okay
so after putting this code and trying out to see the output as same only

so modified code is till now like this

```
import { useEffect, useState } from "react"

export const ProductList = () => {
    const [products, setProducts] = useState([]);
    const [url, setUrl] = useState("http://localhost:8000/products");

   useEffect(() => {
    const fetchProducts = async () => {
        const response = await fetch(url);
        const data = await response.json();
        setProducts(data);
    };

    fetchProducts();
}, [url]);


  return (
    <section>
        <div className="filter">

            <button onClick={() => setUrl("http://localhost:8000/products")}>All</button>
            <button onClick={() => setUrl("http://localhost:8000/products?in_stock=true")}>In Stock Only</button>
        </div>

        { products.map((product) => (
            <div className="card" key={product.id}>
                <p className="id">{product.id}</p>
                <p className="name">{product.name}</p>
                <p className="info">
                    <span>${product.price}</span>
                    <span className={product.in_stock ? "instock" : "unavailable"}>{product.in_stock ? "In Stock" : "Unavailable"}
</span>
                </p>
            </div>
        )) }
    </section>
```

```
  )
}
```

next when i am taking the function outside and just calling it



I am seeing some kind of warning that use effect is missing the dependency as URL is now dependency but URL is not there a function call is there and in side the function that URL is there now it is telling me to put function fetch Products() as dependency into it okay
now program is running properly but i am getting warning here okay

so this is some kind of warning which can create problem at the time of deployment later so we have to rectify this warning okay .

now

```
  const fetchProducts = async () => {
        const response = await fetch(url);
        const data = await response.json();
        setProducts(data);
    };
   useEffect(() => {
       fetchProducts();
}, [url]);
```

now in the above code everything is fine but warning is coming

as URL is now not there in use effect so if i remove URL and make it empty then it will not work and if i put dependency as fetchProducts of what he is saying in warning it is going into infinite loop like earlier so what is the solution for this now

so i am using a call back method which will make function to use same memory so every time same function is getting called and using different memory and also what i am doing here is that
whenever the function is going out of use effect i am getting this error to overcome that i am using this and i am using console log - to know how many times it is being called one time or more just for testing purpose okay .so this is the fresh and final code and after using this all warning and errors which i was getting is gone okay

```
productList.js
---------------
import { useCallback, useEffect, useState } from "react"
```

```
export const ProductList = () => {
    const [products, setProducts] = useState([]);
    const [url, setUrl] = useState("http://localhost:8000/products");

    const fetchProducts = useCallback(async () => {
        const response = await fetch(url);
        const data = await response.json();
        setProducts(data);
    }, [url]);

    useEffect(() => {
        fetchProducts();
        console.log("-");
    }, [fetchProducts]);

  return (
    <section>
        <div className="filter">
            <button onClick={() => setUrl("http://localhost:8000/products")}>All</button>
            <button onClick={() => setUrl("http://localhost:8000/products?in_stock=true")}>In Stock Only</button>
        </div>

        { products.map((product) => (
            <div className="card" key={product.id}>
                <p className="id">{product.id}</p>
                <p className="name">{product.name}</p>
                <p className="info">
                    <span>${product.price}</span>
                    <span className={product.in_stock ? "instock" : "unavailable"}>{product.in_stock ? "In Stock" : "Unavailable"}
</span>
                </p>
            </div>
        )) }
    </section>
  )
}
```

so you can see i had imported call back also so every thing is related here okay

now let us move to topic which is custom hook now what it is actually is that my code i am shifting in another place and calling it so in src create a new folder hooks and there add one file useFetch.js okay and here i will transfer all code of productList.js into which is of use effect into useFetch.js here try to analyze the code once okay .so here it is a convention to use use in naming of file whenever you are creating a custom hook okay

```
useFetch.js
-------------
import { useEffect, useState } from "react";

export const useFetch = (url) => {
    const [data, setData] = useState(null);

    useEffect(() => {
        const fetchData = async () => {
            const response = await fetch(url);
            const result = await response.json();
            setData(result);
        }
        fetchData();
    }, [url]);

  return { data }
}

ProductList.js
--------------
import { useState } from "react"
import { useFetch } from "../hooks/useFetch";

export const ProductList = () => {
    const [url, setUrl] = useState("http://localhost:8000/products/");
    const { data: products } = useFetch(url);
```

```
  return (
    <section>
        <div className="filter">
            <button onClick={() => setUrl("http://localhost:8000/products")}>All</button>
            <button onClick={() => setUrl("http://localhost:8000/products?in_stock=true")}>In Stock Only</button>
        </div>

        { products && products.map((product) => (
            <div className="card" key={product.id}>
                <p className="id">{product.id}</p>
                <p className="name">{product.name}</p>
                <p className="info">
                    <span>${product.price}</span>
                    <span className={product.in_stock ? "instock" : "unavailable"}>{product.in_stock ? "In Stock" : "Unavailable"}
</span>
                </p>
            </div>
        )) }
    </section>
  )
}
```

so there is no change in output here so here useFetch.js code is for generic methods means not only products but also other things also it will work just u need to send the URL to the function okay ..

I am giving an alias name to data as products as in display of productList.js it should match .
if you follow this no need to do call back usage okay as we have done earlier okay so we can avoid that also here okay .

you can see if products is null i am not rendering it okay .

you can try this custom hook for some other functionality also okay .

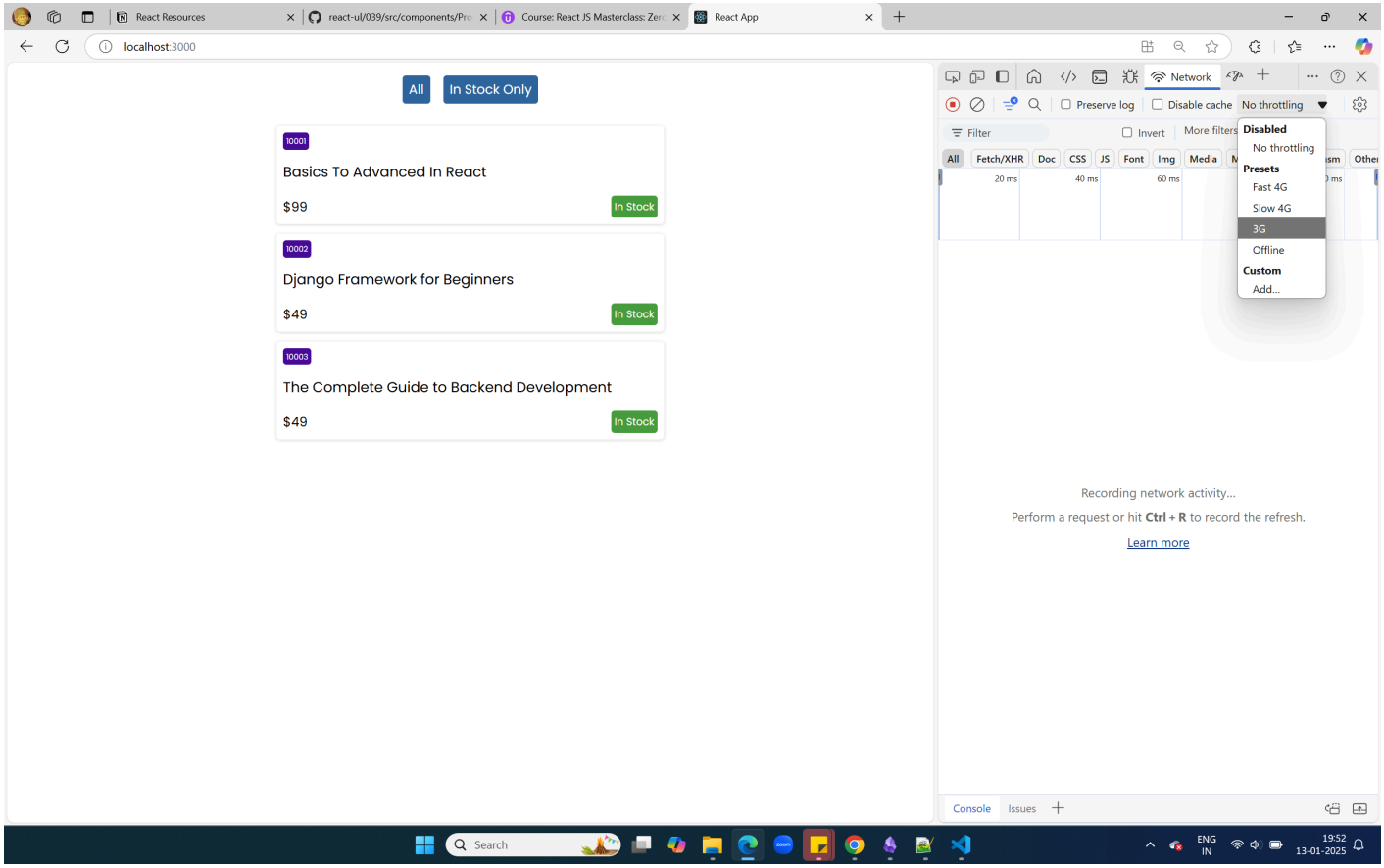
now let us go with loading feature means before loading i want to show some some animation kind of wait which we generally see in the websites okay and here data is very less but when huge amount of data is there then loading will or may be delayed for some seconds and so during that time i want to show some animations so let us work on this now

```
useFetch.js
------------
import { useEffect, useState } from "react";

export const useFetch = (url) => {
    const [data, setData] = useState(null);
    const [loading, setLoading] = useState(false);

    useEffect(() => {
        const fetchData = async () => {
            setLoading(true);
            const response = await fetch(url);
            const result = await response.json();
            setLoading(false);
            setData(result);
        }
        fetchData();
    }, [url]);

  return { data, loading }
}
```

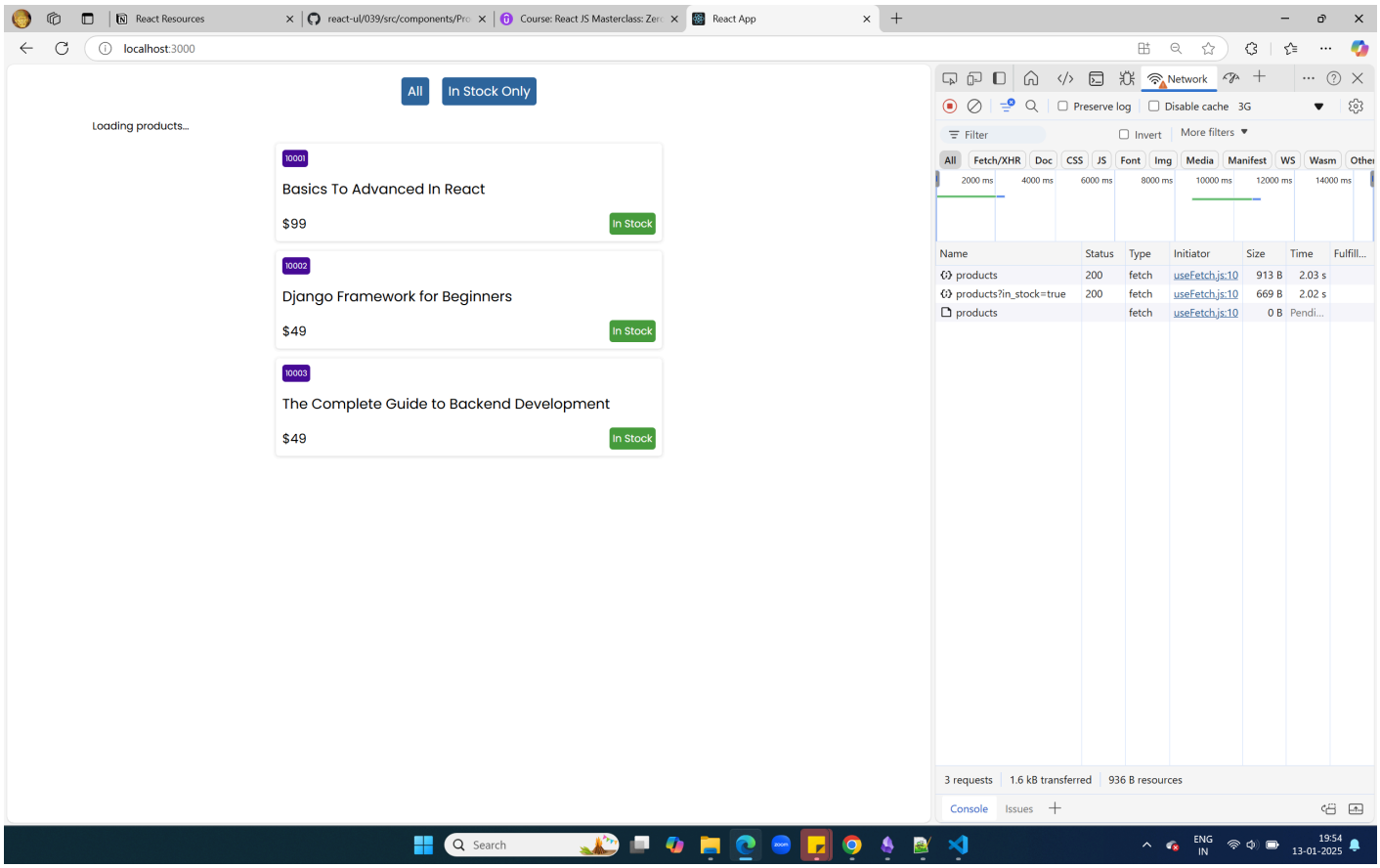
in the above code loading is initially false and when data is about to be fetch from api it will become true and then returned along with data okay ..now in the ProductList.js again i am doing code like this

```
ProductList.js
--------------
import { useState } from "react"
import { useFetch } from "../hooks/useFetch";

export const ProductList = () => {
    const [url, setUrl] = useState("http://localhost:8000/products/");
    const { data: products, loading } = useFetch(url);

  return (
    <section>
```

```
        <div className="filter">
            <button onClick={() => setUrl("http://localhost:8000/products")}>All</button>
            <button onClick={() => setUrl("http://localhost:8000/products?in_stock=true")}>In Stock Only</button>
        </div>
        { loading && <p>Loading products ... </p>}

        { products && products.map((product) => (
            <div className="card" key={product.id}>
                <p className="id">{product.id}</p>
                <p className="name">{product.name}</p>
                <p className="info">
                    <span>${product.price}</span>
                    <span className={product.in_stock ? "instock" : "unavailable"}>{product.in_stock ? "In Stock" : "Unavailable"}
</span>
                </p>
            </div>
        )) }
    </section>
  )
}
```



so i am not seeing the message of Loading products... so i need to slow down the network for that go to inspect and network and 3G as shown in diagram and let us see whether this thing will work or not okay



so again i am changing it to normal and you can put some image also for animation if u want right now i had just kept one message okay .

Now let us talk about how to handle errors okay

```
useFetch.js
---------------
import { useEffect, useState } from "react";

export const useFetch = (url) => {
    const [data, setData] = useState(null);
    const [loading, setLoading] = useState(false);
    const [error, setError] = useState("");

    useEffect(() => {
        const fetchData = async () => {
            setLoading(true);
            try{
                const response = await fetch(url);
                if(!response.ok){
                    throw new Error(response.statusText);
                }
                const result = await response.json();
                setLoading(false);
                setData(result);
                setError("");
            } catch(error){
                setLoading(false);
                setError(error.message);
            }
        }
        fetchData();
    }, [url]);

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

here where there is possibility of error that code is kept in try block and in catch block i am setting the error message and same error i am passing as parameter here

now error can be server side where server is not running you can stop the json server and see how the code is responding and then do client side error where you are passing wrong URL okay

```
ProductList.js
------------------
import { useState } from "react"
import { useFetch } from "../hooks/useFetch";

export const ProductList = () => {
    const [url, setUrl] = useState("http://localhost:8000/products/");
    const { data: products, loading, error } = useFetch(url);

  return (
    <section>
        <div className="filter">
            <button onClick={() => setUrl("http://localhost:8000/products")}>All</button>
            <button onClick={() => setUrl("http://localhost:8000/products?in_stock=true")}>In Stock Only</button>
        </div>
        { loading && <p>Loading products ... </p>}

        { error && <p>{error}</p>}

        { products && products.map((product) => (
            <div className="card" key={product.id}>
                <p className="id">{product.id}</p>
                <p className="name">{product.name}</p>
                <p className="info">
                    <span>${product.price}</span>
                    <span className={product.in_stock ? "instock" : "unavailable"}>{product.in_stock ? "In Stock" : "Unavailable"}
</span>
                </p>
            </div>
        )) }
    </section>
  )
}
```

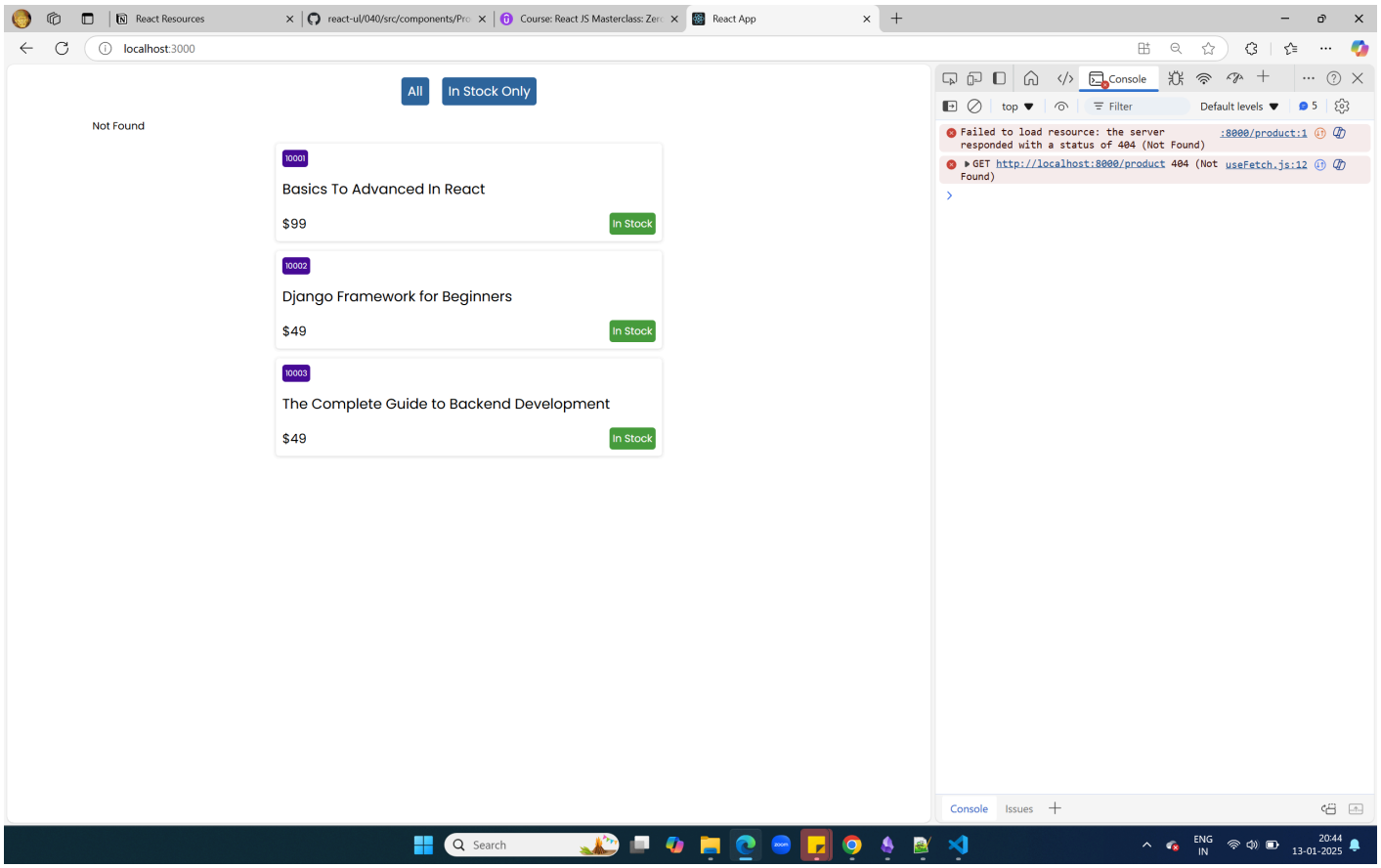
so if error is true print the error message so let us see and how it is handled here okay .

so server is running 8000 json one and wrong URL i am entering means all instead of

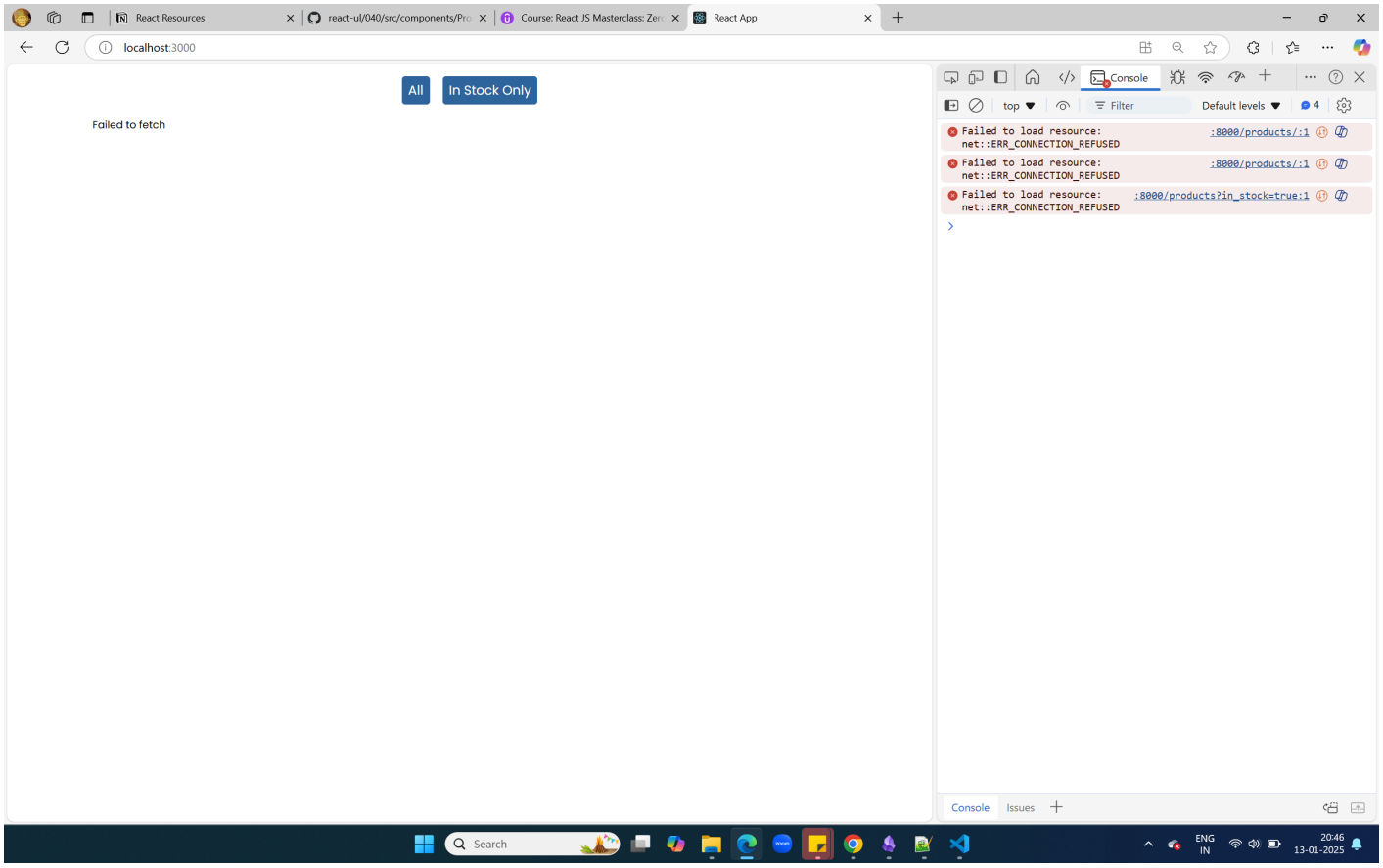
```
      <button onClick={() => setUrl("http://localhost:8000/product")}>All</button>
```

so here above i had put product instead of products

so how it will respond is like this



so now i will stop server and same error i am doing it let us see



so as per coding and what type of error i am doing i am getting the result just analyze the code along with video in which is 43 video okay done now let us move to

Now i want to stop my fetch request in middle okay which has already been called for fetch and it is doing its duty of fetching okay .

so for that one controller you have to create and pass the signal and abort it you can check the video 44 for this okay

```
useFetch.js
---------------
import { useEffect, useState } from "react";

export const useFetch = (url) => {
    const [data, setData] = useState(null);
    const [loading, setLoading] = useState(false);
    const [error, setError] = useState("");

    useEffect(() => {
        const controller = new AbortController()

        const fetchData = async () => {
            setLoading(true);
            try{
                const response = await fetch(url, { signal: controller.signal });
                if(!response.ok){
                    throw new Error(response.statusText);
                }
                const result = await response.json();
                setLoading(false);
                setData(result);
                setError("");
            } catch(error){
                setLoading(false);
                setError(error.message);
            }
        }
        fetchData();

        return () => controller.abort();
    }, [url]);

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

let us see how it behaves now and again go to 3g mode of network and see the effect so after writing above code it goes into unmounting code when i refresh the browser okay .
so last video is not useful and not important so i am skipping that 4 mins video but right now i am done with section 7 now i am moving to new section 8

Section 8 (Project TaskMate)

Now i am doing a full functional project of what we have learned which is TaskMate . This project has got three things header and add a task and show all tasks now we are doing all the things together in the project earlier we did bit by bit now all the things together will do it

I'm having in the project the option to change the theme also we will be deploying the project and it will be online available and also it will be saved into some persistent place as well.

So now I'm creating a new react app by as usual writing the same command

npx create-react-app taskmate ,now earlier also many times i had created taskmate app and modified and now i am creating again for full fledged purpose so giving a different name as taskmateproject so writing the command

npx create-react-app taskmateproject

after creating this app delete these things now one by one which we have done many times okay
reportwebvital.js, logo.svg and apptest.js files all 3 files delete it and then in index.js remove web vitals reference and remove from App.js all jsx code leaving className=App and also remove logo from there from import as logo is also deleted and App.css all code remove it so this much do it and also remove idex.css code as well just make it empty okay .Also remove setuptest.js okay .

just written hello world in App.js and that only displaying so up to this is done of what i had told as initial setup up above okay .

so deletion we have done in src folder and public i have not done if needed will do that also okay
so now first index.css file code is here

```
index.css
---------
@import url('https://fonts.googleapis.com/css2?
family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&f
```

```
amily=Roboto&display=swap');
@import url("https://cdn.jsdelivr.net/npm/bootstrap-icons@1.8.3/font/bootstrap-icons.css");

:root{
  --box-shadow: rgba(0, 0, 0, 0.05) 0px 6px 24px 0px, rgba(0, 0, 0, 0.08) 0px 0px 0px 1px;
}

* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: 'Poppins', sans-serif;
  text-decoration: none;
  list-style: none;
}

/* themes */
.light{
  background-color: #FFFFFF;
  transition: all 500ms;
}

.medium{
  background-color: #EFEFEF;
  transition: all 500ms;
}

.dark{
  background-color: #15202B;
  transition: all 500ms;
}

.gOne{
  background: linear-gradient(90deg, hsla(191, 75%, 60%, 1) 0%, hsla(248, 87%, 36%, 1) 100%);
  transition: all 500ms;
}
.gTwo{
  background: linear-gradient(235deg, #FFFFFF 0%, #000F25 100%), linear-gradient(180deg, #6100FF 0%, #000000 100%), linear-
gradient(235deg, #FFA3AC 0%, #FFA3AC 40%, #00043C calc(40% + 1px), #00043C 60%, #005D6C calc(60% + 1px), #005D6C 70%, #00C9B1 calc(70%
+ 1px), #00C9B1 100%), linear-gradient(125deg, #FFA3AC 0%, #FFA3AC 40%, #00043C calc(40% + 1px), #00043C 60%, #005D6C calc(60% + 1px),
#005D6C 70%, #00C9B1 calc(70% + 1px), #00C9B1 100%);
  background-blend-mode: soft-light, screen, darken, normal;
}

.gThree{
  background: linear-gradient(125deg, #FFFFFF 0%, #000000 100%), linear-gradient(200deg, #FFD9E8 0%, #FFD9E8 50%, #DE95BA calc(50% +
1px), #DE95BA 60%, #7F4A88 calc(60% + 1px), #7F4A88 75%, #4A266A calc(75% + 1px), #4A266A 100%), linear-gradient(113deg, #FFD9E8 0%,
#FFD9E8 40%, #DE95BA calc(40% + 1px), #DE95BA 50%, #7F4A88 calc(50% + 1px), #7F4A88 70%, #4A266A calc(70% + 1px), #4A266A 100%);
  background-blend-mode: overlay, overlay, normal;
}
/* themes */
```

Then now code for App.css added like this

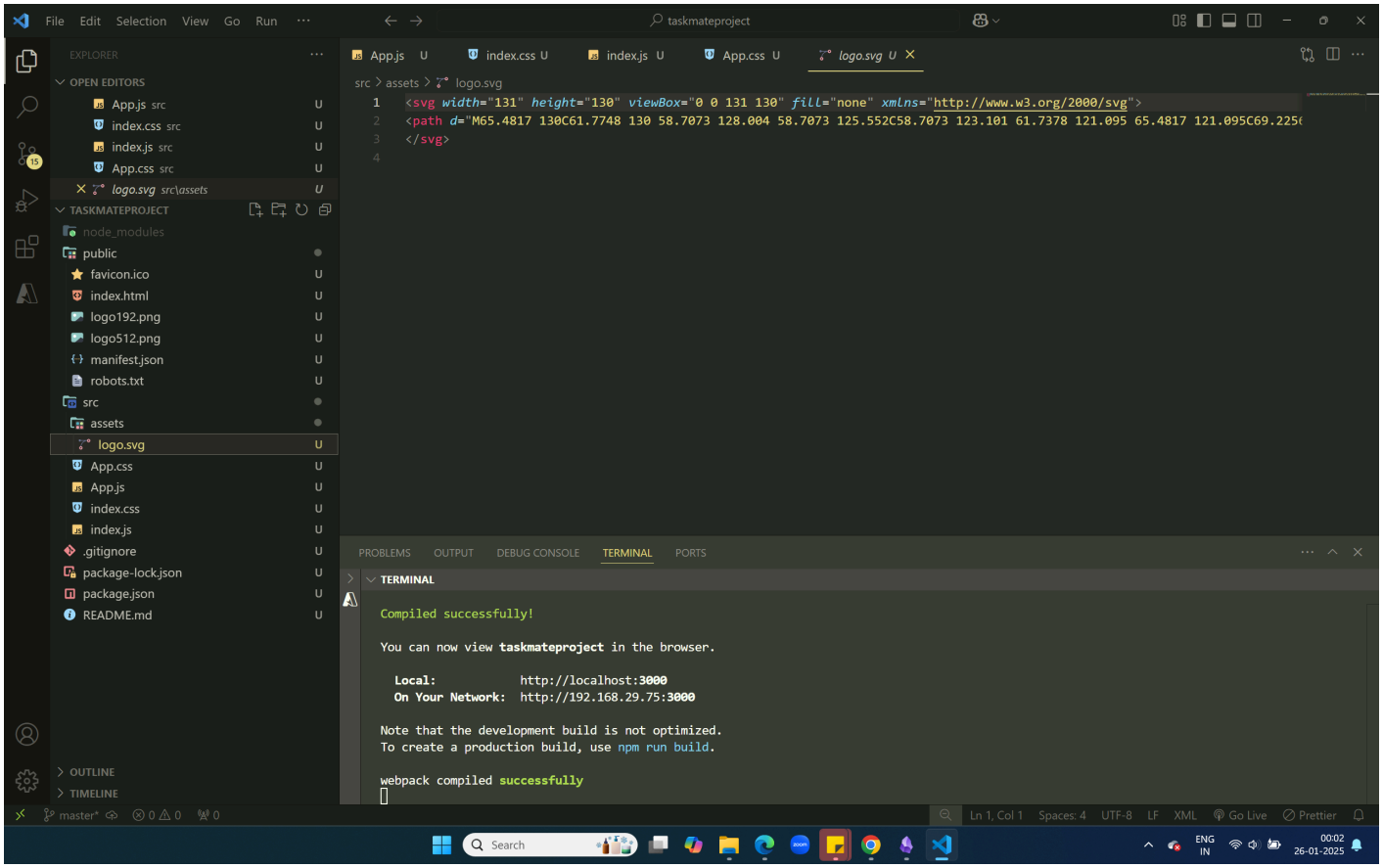
```
App.css
---------
.App{
  min-height: 100vh;
}

.container{
  max-width: 1200px;
  margin: auto;
  padding: 10px 10px;
}

header{
  min-height: 70px;
  display: flex;
  align-items: center;
  justify-content: space-between;
  padding: 15px;
  margin-bottom: 70px;
```

```css
  background-color: #FFFFFF;
  border-radius: 5px;
  box-shadow: var(--box-shadow);
}

header .logo{
  display: flex;
  align-items: center;
}

header .logo img{
  max-width: 40px;
  cursor: pointer;
}

header .logo span{
  font-size: 22px;
  margin-left: 7px;
  cursor: pointer;
  color: #525252;
}

header .themeSelector span{
  height: 20px;
  width: 20px;
  border-radius: 50%;
  display: inline-block;
  margin-left: 5px;
  cursor: pointer;
  border: 1px solid #dfdfdf;
}

header .themeSelector .activeTheme{
  height: 30px;
  width: 30px;
}

.addTask {
  padding: 20px 15px;
  border-radius: 5px;
  background-color: #FFFFFF;
  max-width: 500px;
  margin: 20px auto;
  box-shadow: var(--box-shadow);
}

.addTask form{
  display: flex;
  gap: 7px;
}

.addTask form input{
  flex-grow: 1;
  padding: 5px 5px;
  border: 1px solid #eaeaea;
  border-radius: 5px;
  font-size: 18px;
}

.addTask form input:focus{
  outline: 0.5px solid #eaeaea;
}

.addTask form button{
  background-color: #03ba5c;
  color: #FFFFFF;
  border: none;
  padding: 5px 20px;
  border-radius: 5px;
  cursor: pointer;
  font-size: 18px;
}

.addTask form button:hover{
  background-color: #03b056;
```

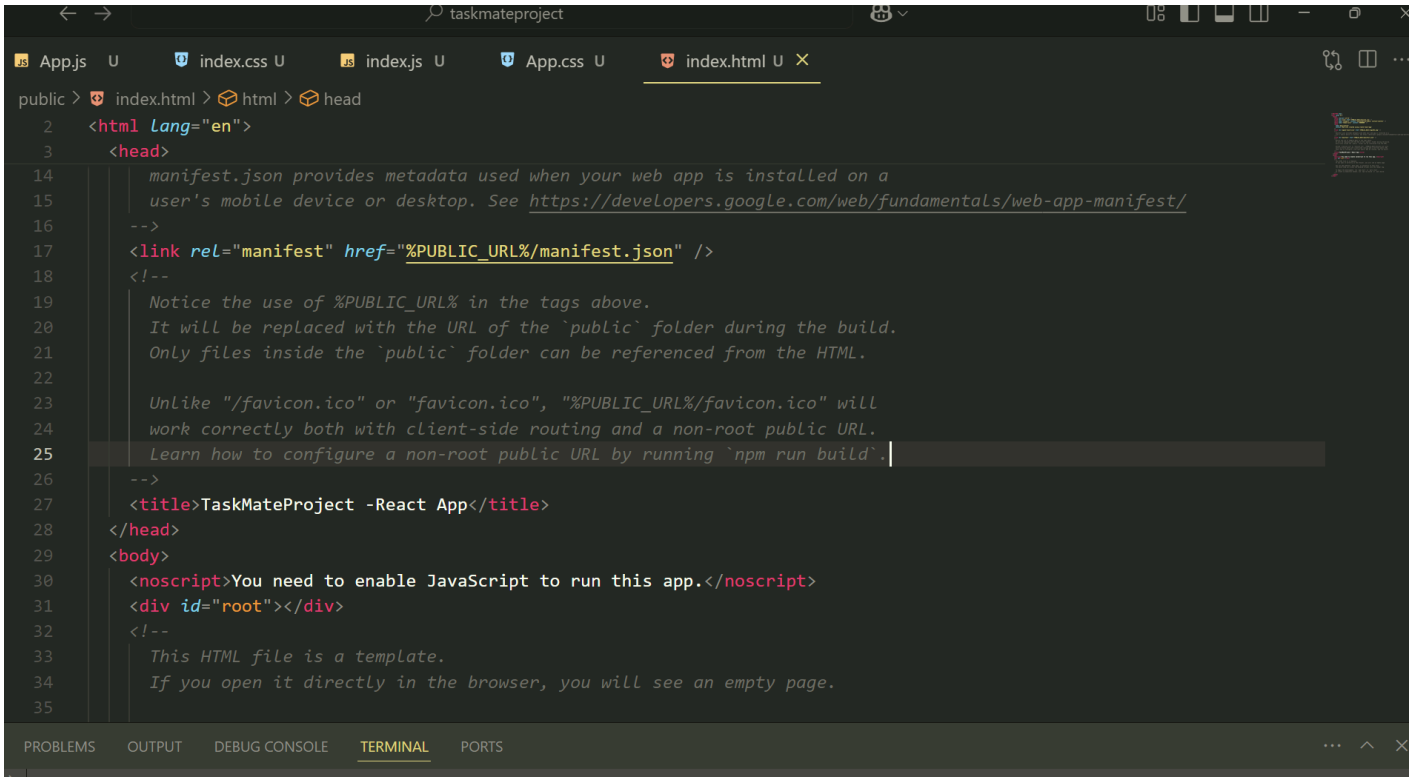
```css
}

.showTask{
  padding: 20px 15px;
  border-radius: 5px;
  background-color: #FFFFFF;
  margin: 20px auto;
  box-shadow: var(--box-shadow);
}

.showTask div.head{
  padding: 10px 0px 20px 0px;
  border-bottom: 1px solid #dfdfdf;
  display: flex;
  align-items: center;
}

.showTask div.head div{
  flex-grow: 1;
}

.showTask .title{
  font-size: 18px;
  font-weight: 600;
  user-select: none;
}

.showTask .count{
  background-color: #dfdfdf;
  color: #525252;
  padding: 5px 10px;
  border-radius: 50%;
  margin-left: 10px;
  user-select: none;
}

.showTask .clearAll{
  background-color: #0f6af3;
  color: #FFFFFF;
  padding: 5px 10px;
  border-radius: 5px;
  cursor: pointer;
  border: 0px;
  font-size: 16px;
}

.showTask .clearAll:hover{
  background-color: #1061db;
}

.showTask ul{
  margin: 30px 5px 20px 5px;
  display: flex;
  flex-direction: row;
  justify-content: space-around;
  flex-wrap: wrap;
  gap: 15px;
}

.showTask li{
  padding: 15px 10px;
  border-bottom: 1px solid #e5e5e5;
  display: flex;
  gap: 5px;
  width: 350px;
  box-shadow: var(--box-shadow);
  border-radius: 5px;
  border-left: 5px solid #1365dfaf;
}

.showTask li:hover{
  background-color: #f8f8f8;
}

.showTask li p{
```

```css
  flex-grow: 1;
  display: flex;
  flex-direction: column;
}

.showTask li .name{
  font-size: 18px;
  padding-bottom: 5px;
}

.showTask li .time{
  font-size: 12px;
  color: #737373;
}

.showTask i.bi-pencil-square{
  font-size: 18px;
  color: #1363DF;
  cursor: pointer;
}

.showTask i.bi-trash{
  font-size: 18px;
  color: #D82148;
  cursor: pointer;
}

@media only screen and (max-width: 800px) {
  .showTask li{
    width: 100%;
  }
}
```

now create a new folder assets in src folder to add my logo and which is pasted here u can see in the image

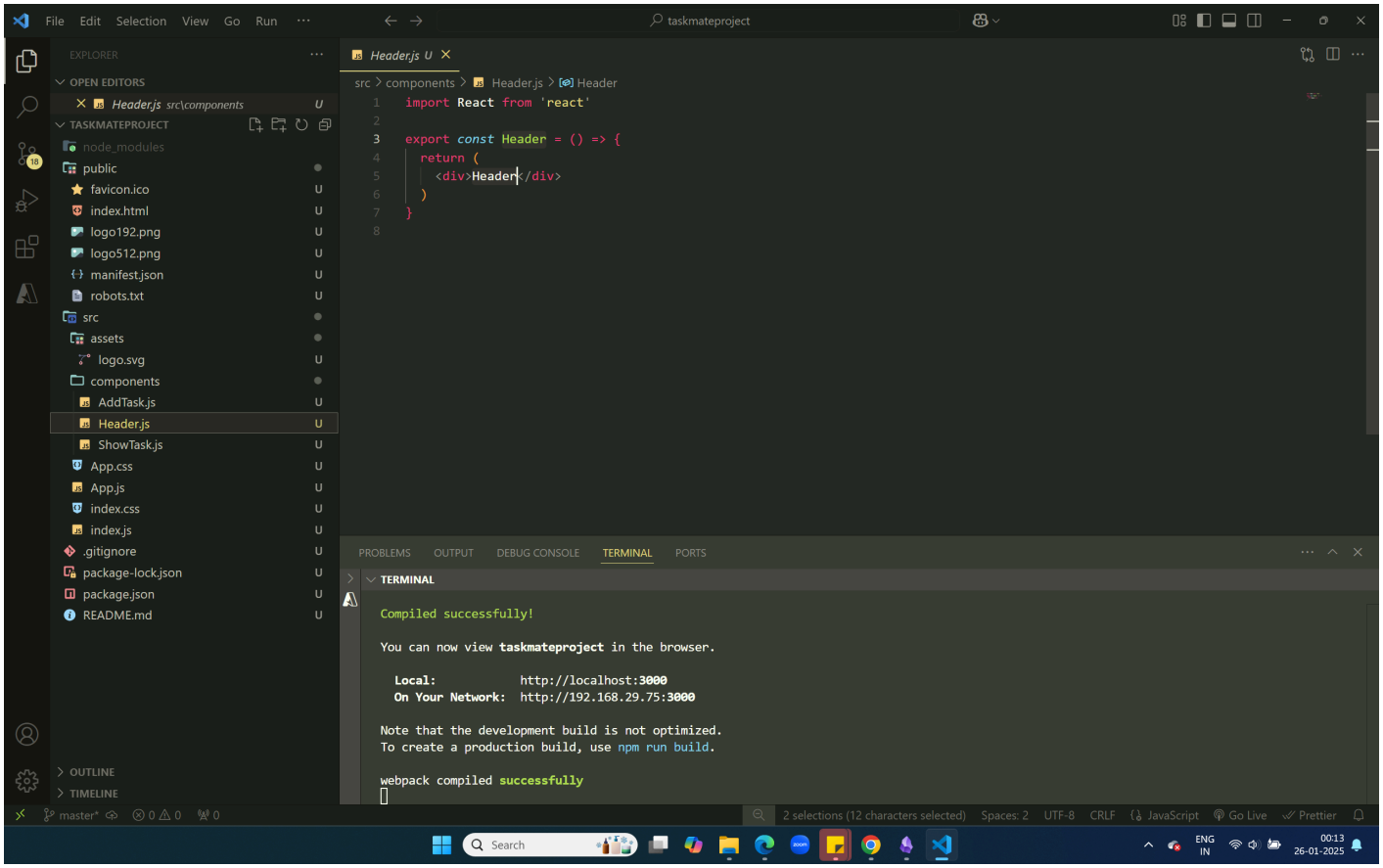


next thing is change the title of the file in index.html



now let us start our tasks for building the UI

now create one folder components in src folder and add three components and do rafc in each page and remove namespace okay

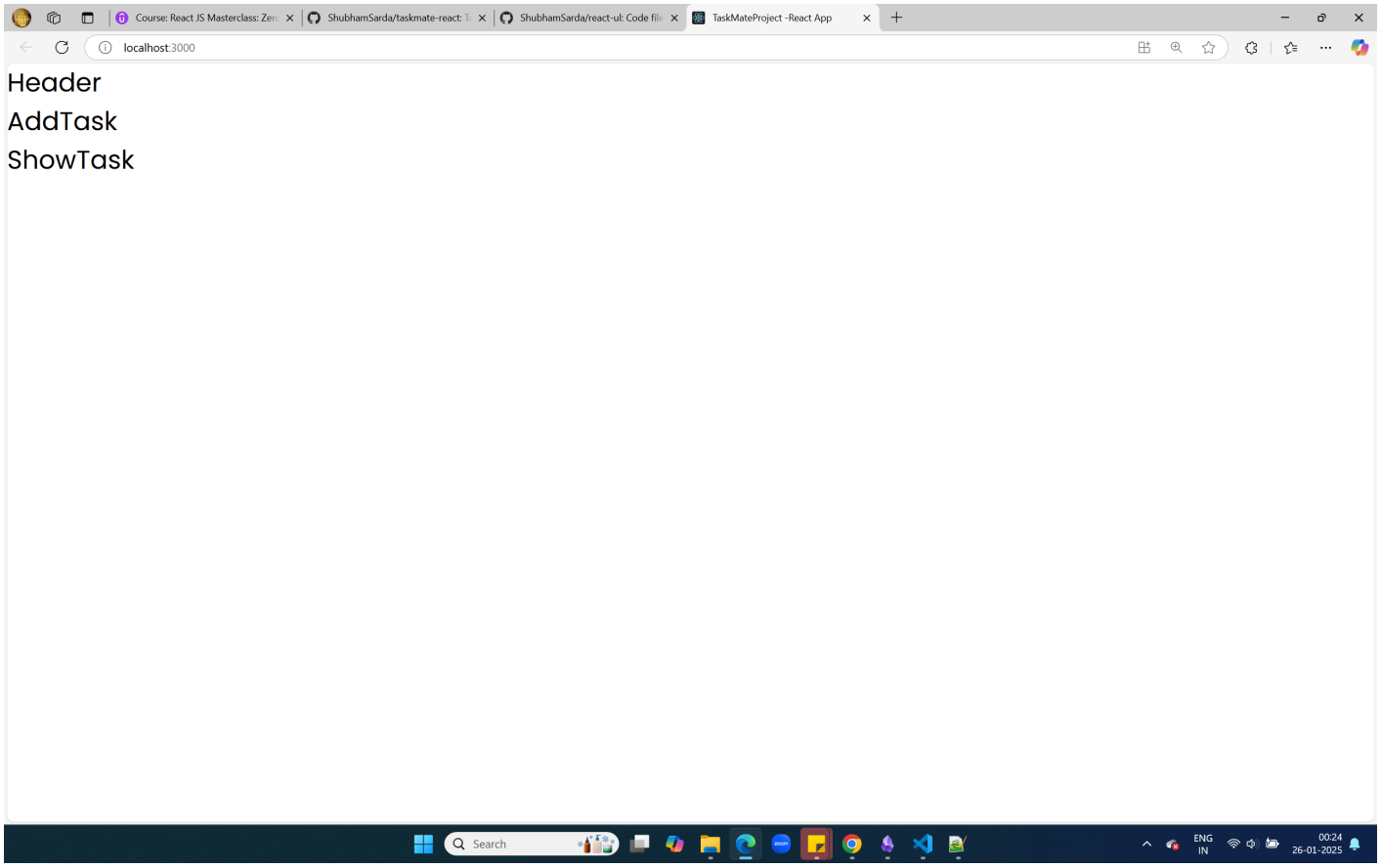


and then import them in App.js like this

```
import './App.css';
import { Header } from "./components/Header";
```

```
import { AddTask } from "./components/AddTask";
import { ShowTask } from "./components/ShowTask";
function App() {
  return (
    <div className="App">
      <Header/>
      <AddTask />
      <ShowTask/>
    </div>
  );
}

export default App;
```

and the output right now i am getting is like this

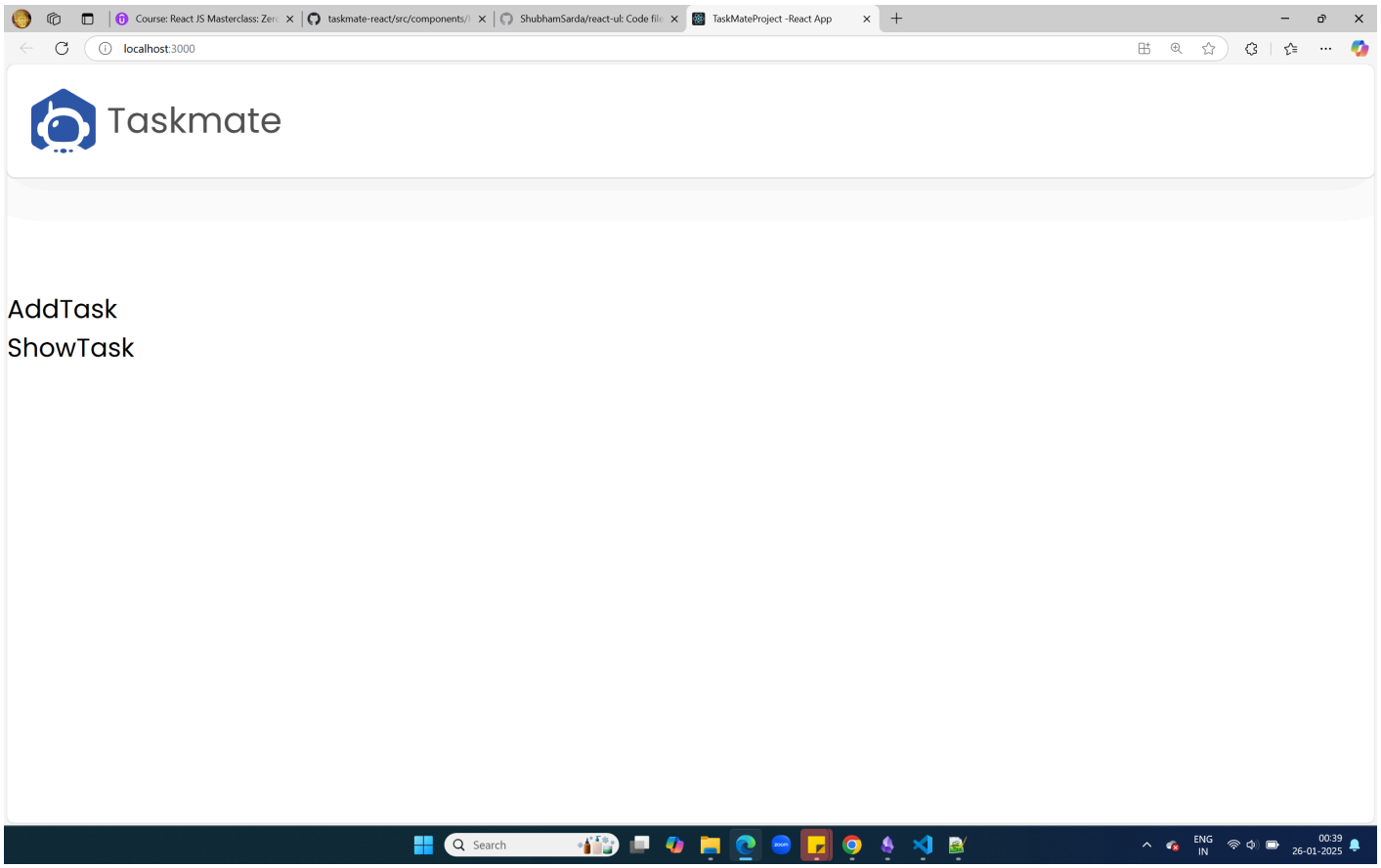


next is i want to work on Header.js now okay

so initial code on Header.js will be like this

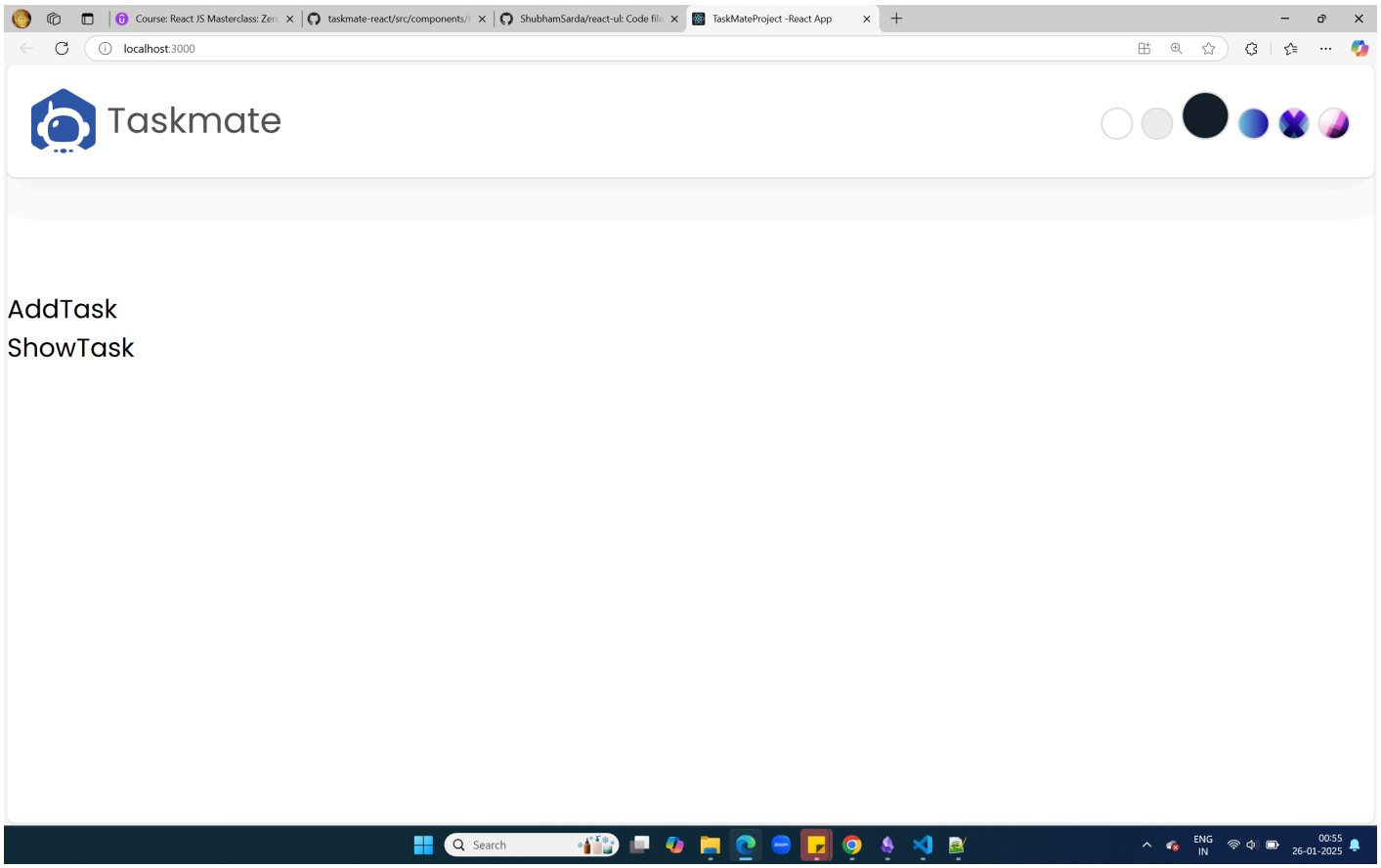
```
import Logo from "../assets/logo.svg";

export const Header = () => {
    return (
        <header>
            <div className="logo">
                <img src={Logo} alt="Taskmate Logo" />
                <span>Taskmate</span>
            </div>
        </header>
    );
};
```

so now running the app it looks like this



adding further theme code

```
import Logo from "../assets/logo.svg";

export const Header = () => {
    return (
        <header>
            <div className="logo">
                <img src={Logo} alt="Taskmate Logo" />
                <span>Taskmate</span>
        </div>
        <div className="themeSelector">
                <span  className="light"></span>
                 <span className="medium"></span>
                 <span  className="dark activeTheme"></span>
                <span  className="gOne"></span>
                  <span  className="gTwo"></span>
                <span  className="gThree"></span>


            </div>
      </header>

    );
};
```

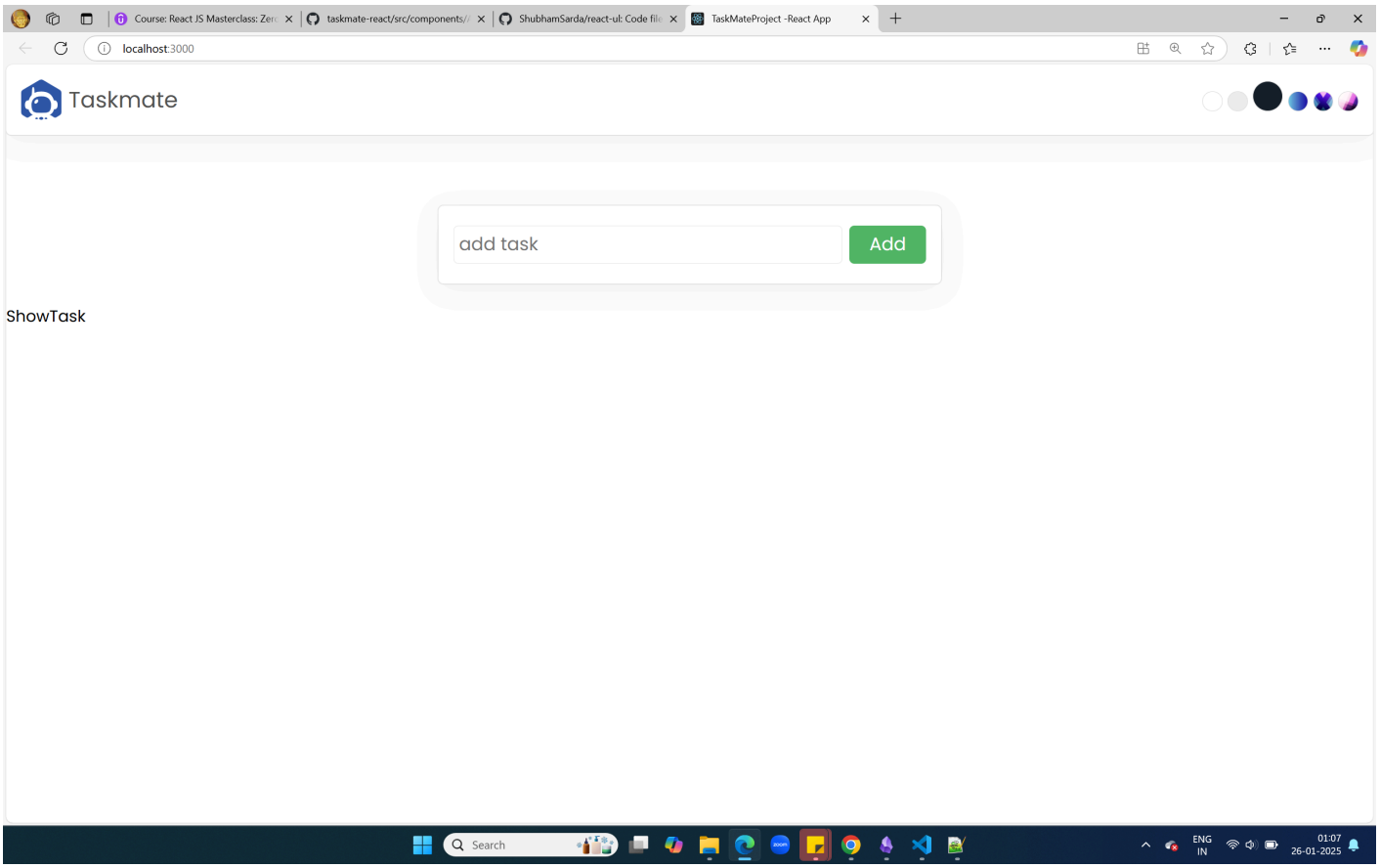


so it will look like this as per above code next

Now let us focus on AddTask.js code

```
export const AddTask = () => {
    return (
        <section className="addTask">
            <form>
                <input type="text" name="task" autoComplete="off" placeholder="add task" maxLength="25" />
                <button type="submit">Add</button>
            </form>
        </section>
    );
};
```

and for above code output is like this



ShowTask

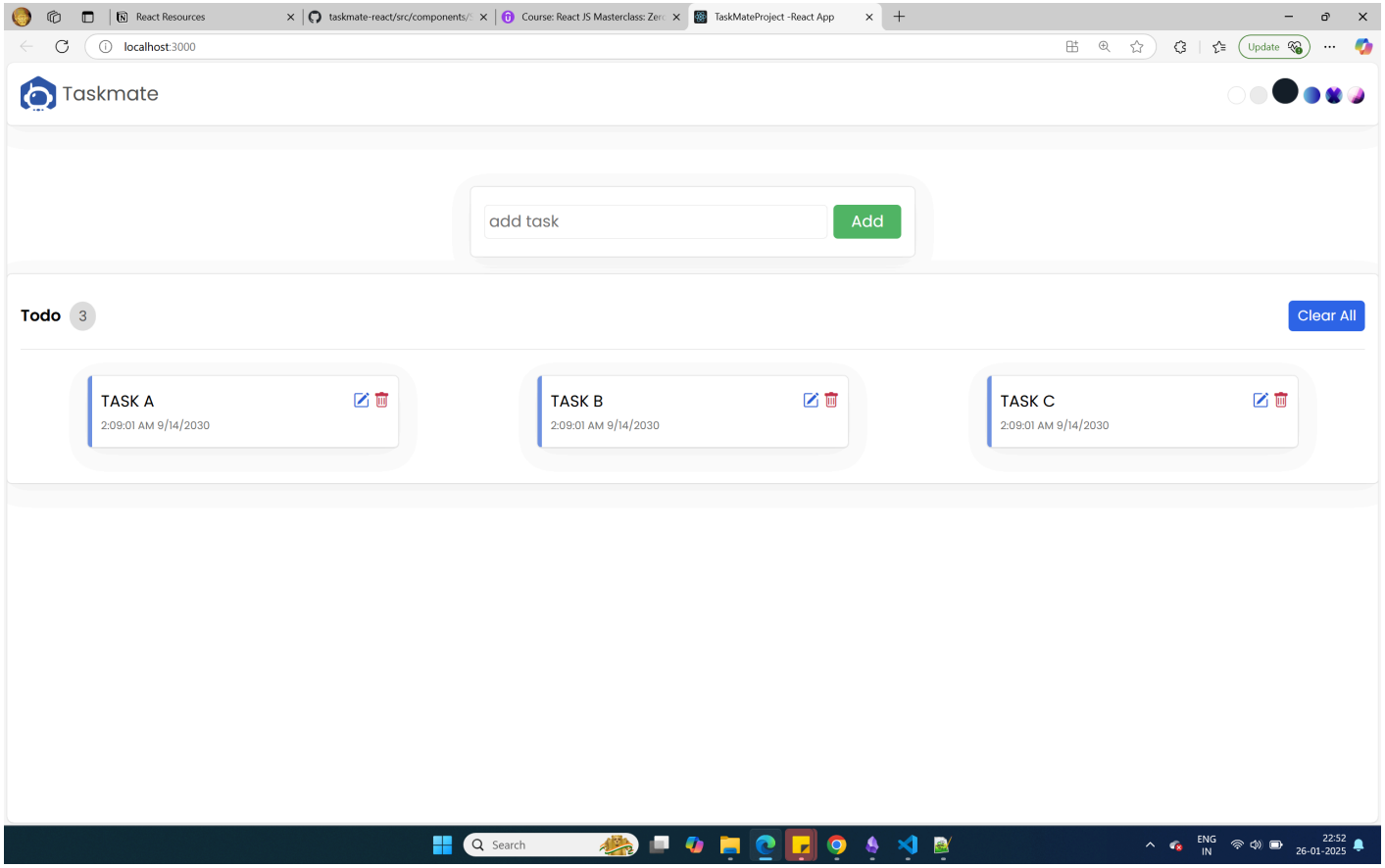
next

focus on ShowTask.js okay

```
export const ShowTask = () => {
  return (
   <section className="showTask">
  <div className="head">
    <div>
      <span className="title">Todo</span>
      <span className="count">0</span>
    </div>
    <button className="clearAll">Clear All</button>
  </div>
  <ul>
    <li>
      <p>
        <span className="name">Task A</span>
        <span className="time">2:09:01 AM 9/14/2030</span>
      </p>
      <i className="bi bi-pencil-square"></i>
      <i className="bi bi-trash"></i>
    </li>
  </ul>
</section>

  )
}


//next addding on top some default coded values of tasks like this



export const ShowTask = () => {

  const tasks = [
  { id: 10001, name: "TASK A", time: "2:09:01 AM 9/14/2030" },
  { id: 10002, name: "TASK B", time: "2:09:01 AM 9/14/2030" },
  { id: 10003, name: "TASK C", time: "2:09:01 AM 9/14/2030" },
];
```

```
  return (
   <section className="showTask">
  <div className="head">
    <div>
      <span className="title">Todo</span>
      <span className="count">0</span>
    </div>
    <button className="clearAll">Clear All</button>
  </div>
  <ul>
    <li>
      <p>
        <span className="name">Task A</span>
        <span className="time">2:09:01 AM 9/14/2030</span>
      </p>
      <i className="bi bi-pencil-square"></i>
      <i className="bi bi-trash"></i>
    </li>
  </ul>
</section>

  )
}

//and then displaying that in show list  of li elements like this

export const ShowTask = () => {
  const tasks = [
    { id: 10001, name: "TASK A", time: "2:09:01 AM 9/14/2030" },
    { id: 10002, name: "TASK B", time: "2:09:01 AM 9/14/2030" },
    { id: 10003, name: "TASK C", time: "2:09:01 AM 9/14/2030" },
  ];

  return (
    <section className="showTask">
      <div className="head">
        <div>
          <span className="title">Todo</span>
          <span className="count">{tasks.length}</span>
        </div>
        <button className="clearAll">Clear All</button>
      </div>
      <ul>
        {tasks.map((task) => (
          <li key={task.id}>
            <p>
              <span className="name">{task.name}</span>
              <span className="time">{task.time}</span>
            </p>
            <i className="bi bi-pencil-square"></i>
            <i className="bi bi-trash"></i>
          </li>
        ))}
      </ul>
    </section>
  );
};
```

so it looks like this till now it is done



Now i am going to create task list state where i can add task inside it and delete and update it so after creating that i can pass them to add tasks and show tasks okay in App.js okay .

```
import './App.css';
import { Header } from "./components/Header";
import { AddTask } from "./components/AddTask";
import { ShowTask } from "./components/ShowTask";
import { useState } from 'react';

function App() {

  const[tasklist,setTasklist]=useState([])

  return (
    <div className="App">
      <Header/>
      <AddTask tasklist={tasklist} setTasklist={setTasklist} />
      <ShowTask tasklist={tasklist} setTasklist={setTasklist}/>
    </div>
  );
}

export default App;
```

```
//after this in show task page it will take the parameters and default value i had removed here
//as i am adding tasklist as parameter that only should be added ..so removed tasks okay
//and at clear all i am making it zero okay using setTasklist parameter and u can see count is
//also defined on tasklist.length it is same as what we have done earlier

ShowTask.js
------------

export const ShowTask = ({tasklist,setTasklist}) => {
```

```
  return (
    <section className="showTask">
      <div className="head">
        <div>
          <span className="title">Todo</span>
          <span className="count">{tasklist.length}</span>
        </div>
        <button className="clearAll" onClick={() => setTasklist([])}>Clear All</button>
      </div>
      <ul>
        {tasklist.map((task) => (
          <li key={task.id}>
            <p>
              <span className="name">{task.name}</span>
              <span className="time">{task.time}</span>
            </p>
            <i className="bi bi-pencil-square"></i>
            <i className="bi bi-trash"></i>
          </li>
        ))}
      </ul>
    </section>
  );
};


//now the same value the add task page also takes and same tasks also earlier done
//so you can check the code in addtaskpage as well



AddTask.js
-------------


export const AddTask = ({tasklist,setTasklist}) => {
   const handleSubmit = (e) => {
  e.preventDefault();

  const date = new Date();
  const newTask = {
    id: date.getTime(),
    name: e.target.task.value,
    time: `${date.toLocaleTimeString()} ${date.toLocaleDateString()}`,
  };

       setTasklist([...tasklist, newTask]);
       e.target.task.value = "";
};

return (
  <section className="addTask">
    <form onSubmit={handleSubmit}>
      <input
        type="text"
        name="task"
        autoComplete="off"
        placeholder="add task"
        maxLength="25"
      />
      <button type="submit">Add</button>
    </form>
  </section>
);

};
```

so code is working fine till now next move to edit and delete functionality in show tasks page okay

now when i click the edit of individual task i should carry the information to add task component okay and from there if click update it should be updated in show task along with date and when i click from show task individual task i should be aware of which task has been clicked for that what we have to do we need to add task use state also in app.js

so i have to create a task use state here as i need to show this individual task every where in the application in both the components in add task and show task as well

so App.js coding is like this

```
App.js
---------
import { useState } from "react";
import { Header } from "./components/Header";
import { AddTask } from "./components/AddTask";
import { ShowTask } from "./components/ShowTask";
import './App.css';


function App() {
  const [tasklist, setTasklist] = useState([]);
  const [task, setTask] = useState({});

  return (
    <div className="App">
      <Header />
      <AddTask
        tasklist={tasklist}
        setTasklist={setTasklist}
        task={task}
        setTask={setTask}
      />
      <ShowTask
        tasklist={tasklist}
        setTasklist={setTasklist}
        task={task}
        setTask={setTask}
      />
    </div>
  );
}

export default App;
```

now get the things in show task now okay

As i am passing task here to show task and there also inside coding

```
tasklist.map((task)
```

I am having task which will be looping thought-out so in order to avoid confusion i will use todo here and change the code okay

and two functions one for delete and one for update is there for which code has been written okay
here in those functions i will use filter thing here means which is needed that list i will catch and based on condition filter the list and will show which is not matching okay .

so id will be passed to function and from the task list matching one is not shown others are shown as matching one will be deleted one or updated one let us see more into this

so you can see in below code i am using find in edit method and filter in delete okay

```
ShowTask.js
------------
export const ShowTask = ({tasklist, setTasklist, task, setTask}) => {

    const handleEdit = (id) => {
        const seletedTask = tasklist.find(todo => todo.id === id);
        setTask(seletedTask);
    }

    const handleDelete = (id) => {
        const updatedTasklist = tasklist.filter(todo => todo.id !== id);
        setTasklist(updatedTasklist);
    }

  return (
    <section className="showTask">
        <div className="head">
```

```
            <div>
                <span className="title">Todo</span>
                <span className="count">{tasklist.length}</span>
            </div>
            <button onClick={() => setTasklist([])} className="clearAll">Clear All</button>
        </div>
        <ul>
            { tasklist.map((todo) => (
                <li key={todo.id}>
                    <p>
                        <span className="name">{todo.name}</span>
                        <span className="time">{todo.time}</span>
                    </p>
                    <i onClick={() => handleEdit(todo.id)} className="bi bi-pencil-square"></i>
                    <i onClick={() => handleDelete(todo.id)} className="bi bi-trash"></i>
                </li>
            )) }
        </ul>
    </section>
  )
}
```

now coming from App.js earlier here task is empty in App.js and now what happens next is

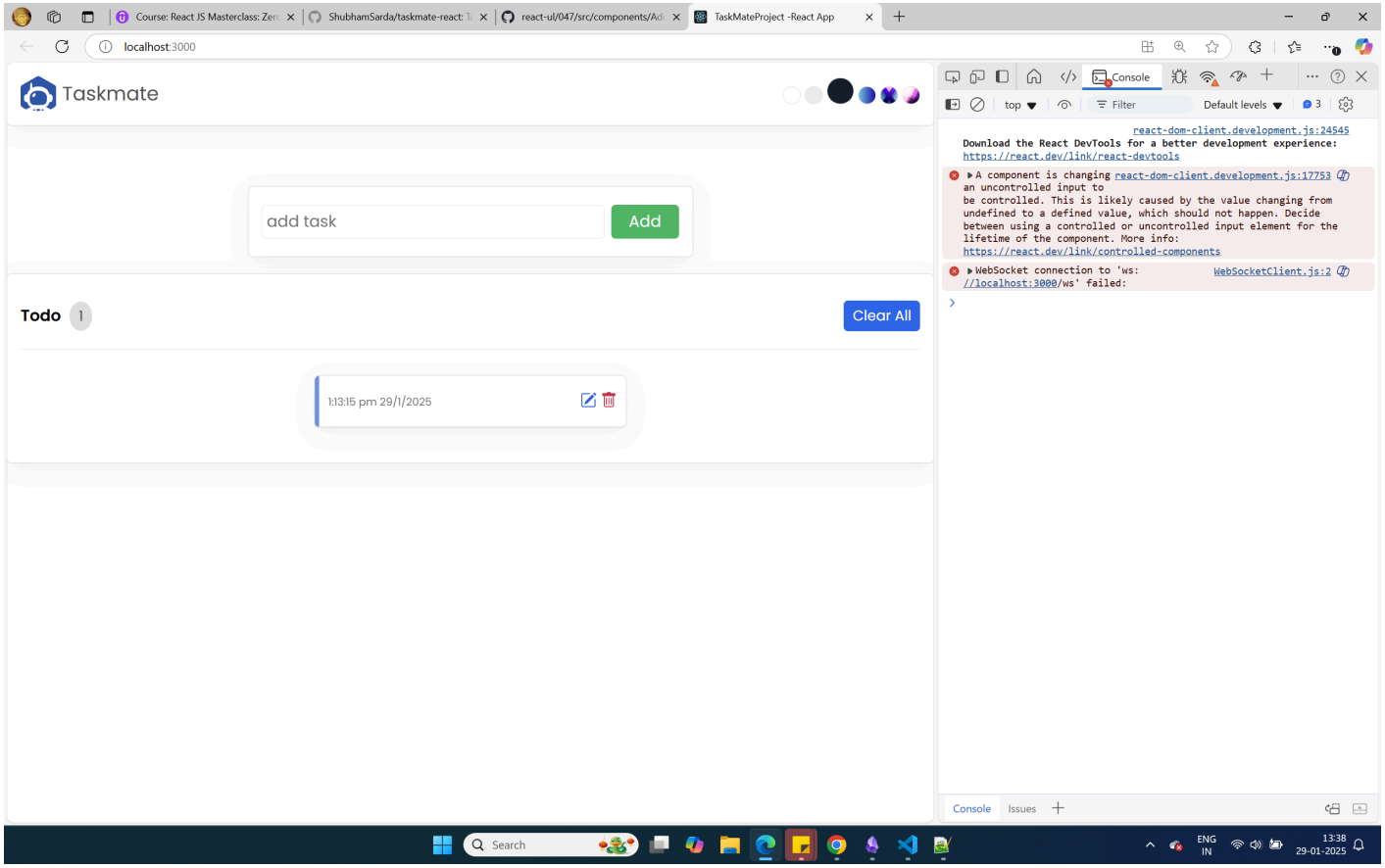
```
const [task, setTask] = useState({}); in App.js
```

when i click the edit button from show tasks this will be filled with some object which is selected task in show task and and in add task we are submitting the form and that control text box is having some value which i will specify and on change event is also specified in add task u can see and in the form if i click edit button means update button which is not there which will come based on condition it will come based on ternary operator it will come will be shown to u
so in submit handle form code if u click edit id is passed and instead if id is there some update code is coming otherwise normal code will come in add task okay
so all explanation i had given above in this paragraph you have to analyze the code why it has been written like that what effect it is showing in code and application that u have to understand okay .

```
AddTask.js
------------
export const AddTask = ({tasklist, setTasklist, task, setTask}) => {

  const handleSubmit = (e) => {
    e.preventDefault();

    if(task.id){
      const date = new Date();
      const updatedTasklist = tasklist.map((todo) => (
        todo.id === task.id ? {id: task.id, name: task.name, time: `${date.toLocaleTimeString()} ${date.toLocaleDateString()}`} : todo
      ));
      setTasklist(updatedTasklist);
    } else {
      const date = new Date();
      const newTask = {
        id: date.getTime(),
        name: e.target.task.value,
        time: `${date.toLocaleTimeString()} ${date.toLocaleDateString()}`
      }
      setTasklist([...tasklist, newTask]);
      e.target.task.value = "";
    }
  }

  return (
    <section className="addTask">
        <form onSubmit={handleSubmit}>
            <input type="text" name="task" value={task.name} autoComplete="off" placeholder="add task" maxLength="25" onChange={e =>
setTask({...task, name: e.target.value})} />
            <button type="submit">Add</button>
        </form>
    </section>
  )
}
```

so this code which i had written till now is fine ..and once task is added or updated i should empty it and also button is in add form only it has to be changed to update let us work on that from now onwards

I am getting some warning also here

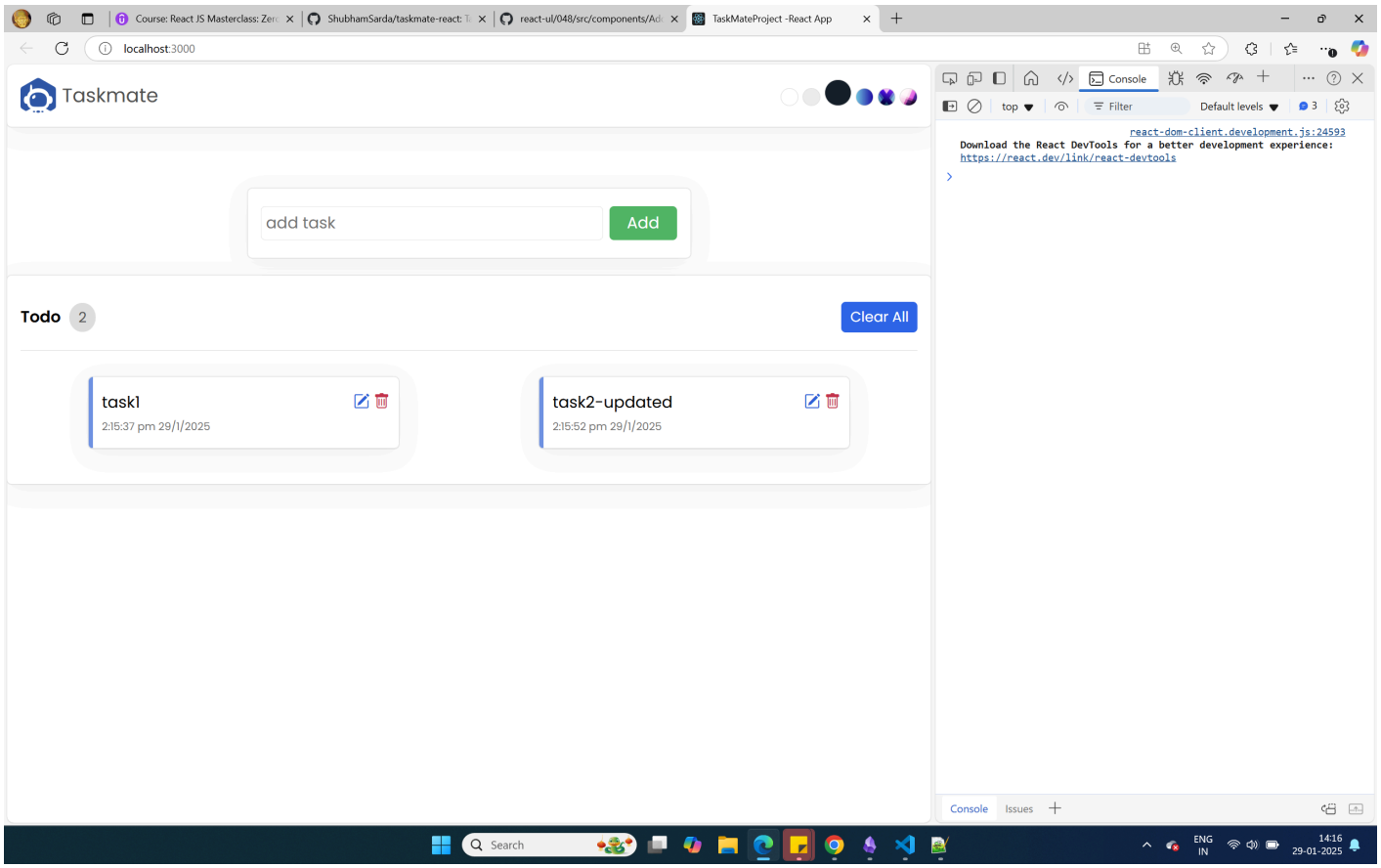


so let us work on that also from now onwards
I have to make my task empty for that make set task to empty both after adding the task and after updating the task also so even i kept this it is not happening or not getting empty and i am getting same kind of error here under inspect okay

As task.name is getting undefined once i delete so use or condition which i had done in code once check it okay so by this below code the inspect error also has gone let us see now

```
AddTask.js
-----------
export const AddTask = ({tasklist, setTasklist, task, setTask}) => {

  const handleSubmit = (e) => {
    e.preventDefault();

    if(task.id){
      const date = new Date();
      const updatedTasklist = tasklist.map((todo) => (
        todo.id === task.id ? {id: task.id, name: task.name, time: `${date.toLocaleTimeString()} ${date.toLocaleDateString()}`} : todo
      ));
      setTasklist(updatedTasklist);
      setTask({});
    } else {
      const date = new Date();
      const newTask = {
        id: date.getTime(),
        name: e.target.task.value,
        time: `${date.toLocaleTimeString()} ${date.toLocaleDateString()}`
      }
      setTasklist([...tasklist, newTask]);
      setTask({});
    }
  }

  return (
    <section className="addTask">
        <form onSubmit={handleSubmit}>
            <input type="text" name="task" value={task.name || ""} autoComplete="off" placeholder="add task" maxLength="25" onChange=
{e => setTask({...task, name: e.target.value})} />
```

```
            <button type="submit">Add</button>
          </form>
      </section>
    )
}
```



so in the image i can see how it is working fine then now i want to store the information in the localhost but before that let us update the button from add to update

```
  <button type="submit">{ task.id ? "Update" : "Add" }</button>
so this change u need to do in AddTask .js
```

```
AddTask.js
--------------
export const AddTask = ({tasklist, setTasklist, task, setTask}) => {

  const handleSubmit = (e) => {
    e.preventDefault();

    if(task.id){
      const date = new Date();
      const updatedTasklist = tasklist.map((todo) => (
        todo.id === task.id ? {id: task.id, name: task.name, time: `${date.toLocaleTimeString()} ${date.toLocaleDateString()}`} : todo
      ));
      setTasklist(updatedTasklist);
      setTask({});
    } else {
      const date = new Date();
      const newTask = {
        id: date.getTime(),
        name: e.target.task.value,
        time: `${date.toLocaleTimeString()} ${date.toLocaleDateString()}`
      }
      setTasklist([...tasklist, newTask]);
      setTask({});
    }
  }
```

```
  return (
    <section className="addTask">
        <form onSubmit={handleSubmit}>
            <input type="text" name="task" value={task.name || ""} autoComplete="off" placeholder="add task" maxLength="25" onChange=
{e => setTask({ ...task, name: e.target.value})} />
            <button type="submit">{ task.id ? "Update" : "Add" }</button>
        </form>
    </section>
  )
}
```

Now for local storage setting i am using using use effect and to get getting the value from use state okay .

```
App.js
--------
import { useEffect, useState } from "react";
import { Header } from "./components/Header";
import { AddTask } from "./components/AddTask";
import { ShowTask } from "./components/ShowTask";
import './App.css';


function App() {
  const [tasklist, setTasklist] = useState(JSON.parse(localStorage.getItem("tasklist")) || []);
  const [task, setTask] = useState({});

  useEffect(() => {
    localStorage.setItem("tasklist", JSON.stringify(tasklist))
  }, [tasklist]);

  return (
    <div className="App">
      <Header />
      <AddTask
        tasklist={tasklist}
        setTasklist={setTasklist}
        task={task}
        setTask={setTask}
      />
      <ShowTask
        tasklist={tasklist}
        setTasklist={setTasklist}
        task={task}
        setTask={setTask}
      />
    </div>
  );
}

export default App;
```

Now for the above code if i refresh also it will not go

now in local storage i want to store theme also for the website let us do that

```
Header.js
----------
import { useEffect } from "react";
import { useState } from "react"
import Logo from "../assets/logo.png"

export const Header = () => {
  const [theme, setTheme] = useState(JSON.parse(localStorage.getItem("theme")) || "medium");

  useEffect(() => {
    localStorage.setItem("theme", JSON.stringify(theme));
    document.documentElement.removeAttribute("class");
    document.documentElement.classList.add(theme);
  }, [theme]);

  return (
    <header>
        <div className="logo">
```

```
            <img src={Logo} alt="Taskmate Logo" />
            <span>Taskmate</span>
        </div>
        <div className="themeSelector">
            <span onClick={() => setTheme("light")} className={ theme === "light" ? "light activeTheme" : "light"}></span>
            <span onClick={() => setTheme("medium")} className={ theme === "medium" ? "medium activeTheme" : "medium"}></span>
            <span onClick={() => setTheme("dark")} className={ theme === "dark" ? "dark activeTheme" : "dark"}></span>
            <span onClick={() => setTheme("gOne")} className={ theme === "gOne" ? "gOne activeTheme" : "gOne"}></span>
            <span onClick={() => setTheme("gTwo")} className={ theme === "gTwo" ? "gTwo activeTheme" : "gTwo"}></span>
            <span onClick={() => setTheme("gThree")} className={ theme === "gThree" ? "gThree activeTheme" : "gThree"}></span>
        </div>
    </header>
  )
}
```

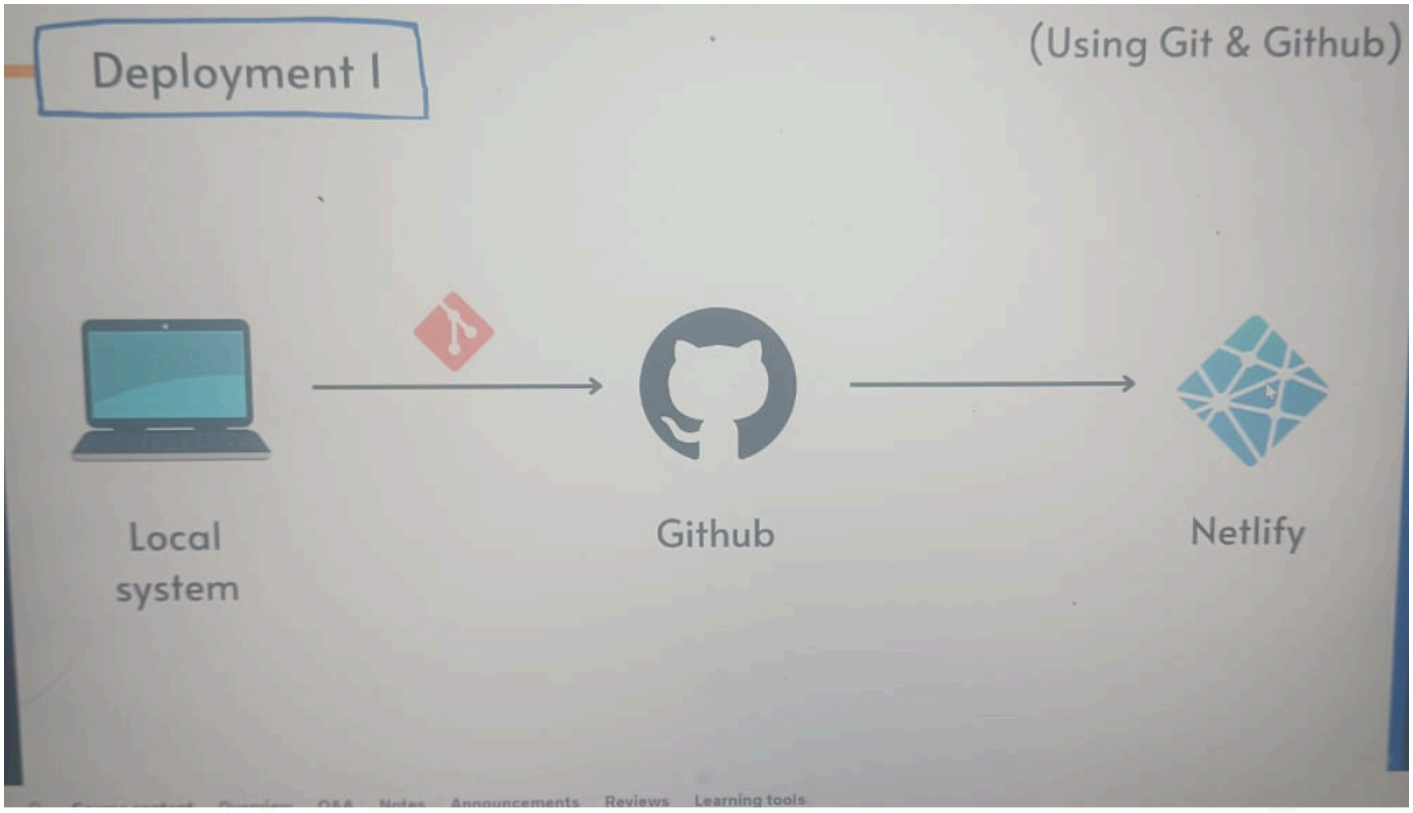
so let us understand above code lightly ..how it is working and all

to remove the previous one already set i am using remove also in use effect so use effect we know already when the theme is changed then what ever logic is there in use effect that will be executed okay .

so using set theme method of use effect i am changing the logic okay .

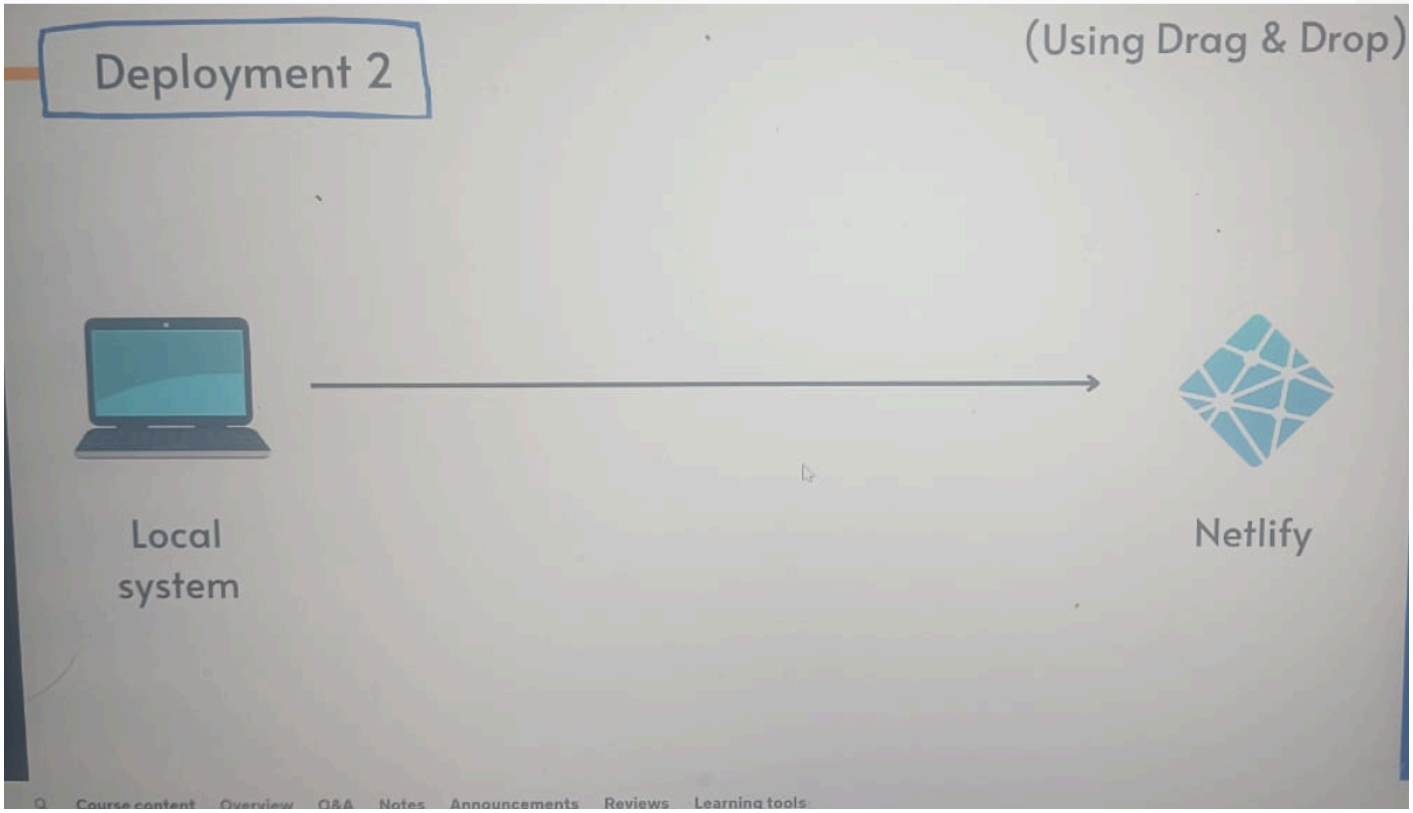
generally on first run of website it will be empty so use or operator here in code above now refer video 55 for more information now let us see how i can deploy this application okay

Two ways of deployment is there is one is from github to netilfy



here no need to create the build again from github any changes u can directly do it means continuously u can work with the website redeployment is not needed here automatically it will take the changes u have done or added it will take automatically okay .

and second thing is directly into netlify like this here for any change u need to re deploy the project here this is used for small project the second kind of deployment okay

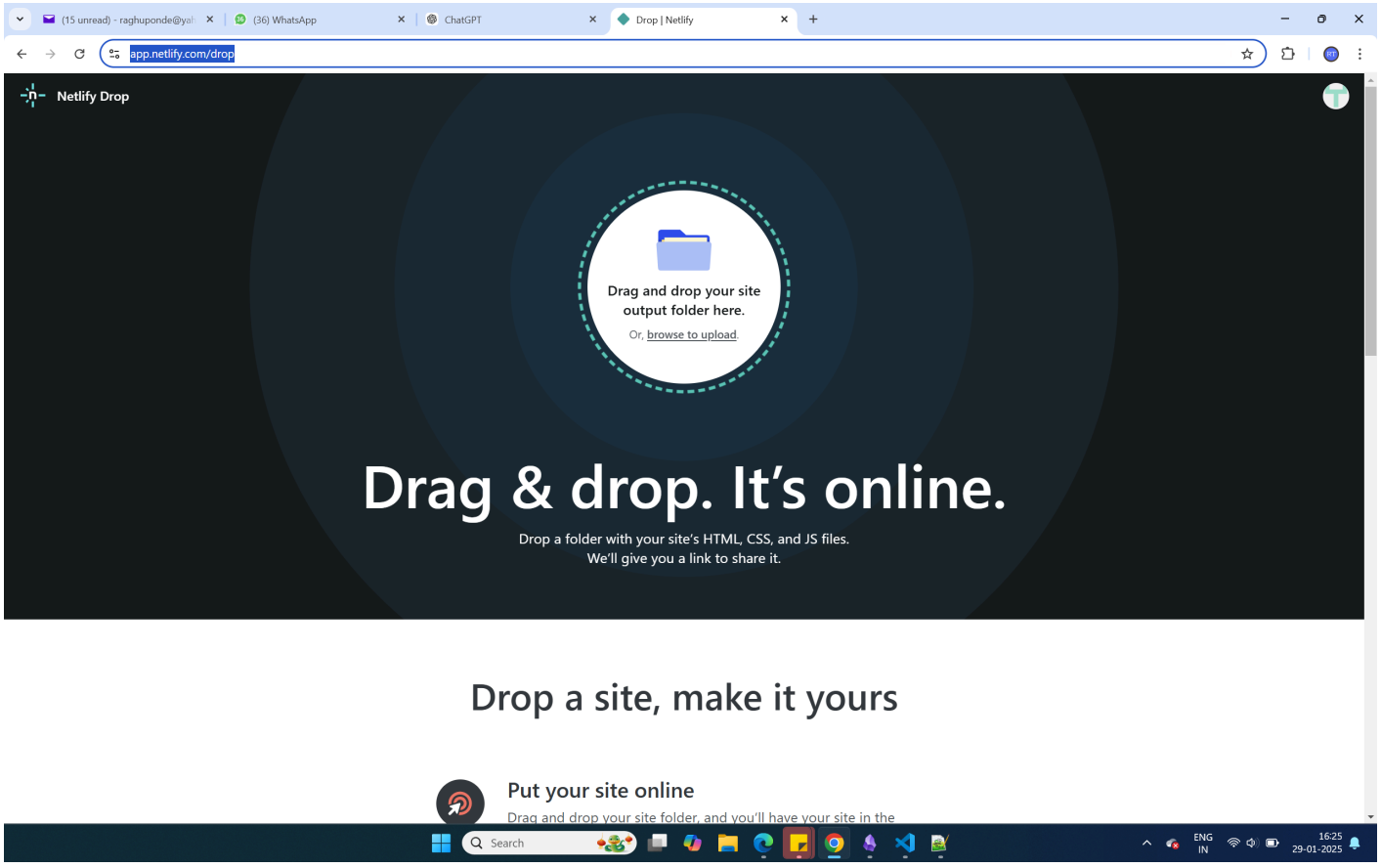


we will follow this here simply create the build and upload that build in netlify okay .

go to website https://www.netlify.com/

after filling basic details i jumped to this page here i had not gone through github personnel i am just logging in okay

https://app.netlify.com/drop



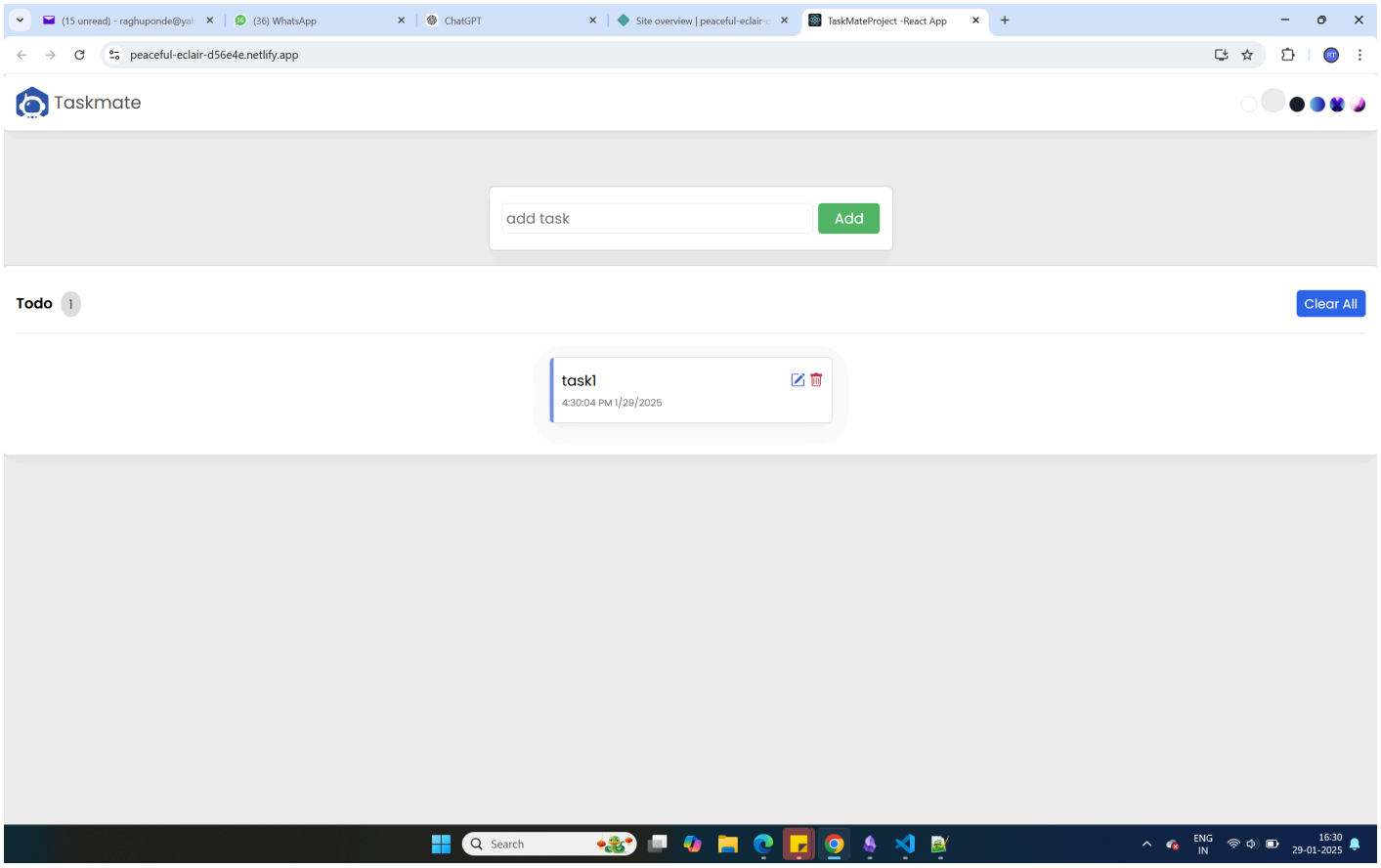
now stop your application using control + c and then run the command npm run build
from your project okay

A build folder will be created in the project select that build folder and drag it in netlify above

so after browse and uploading the folder of build

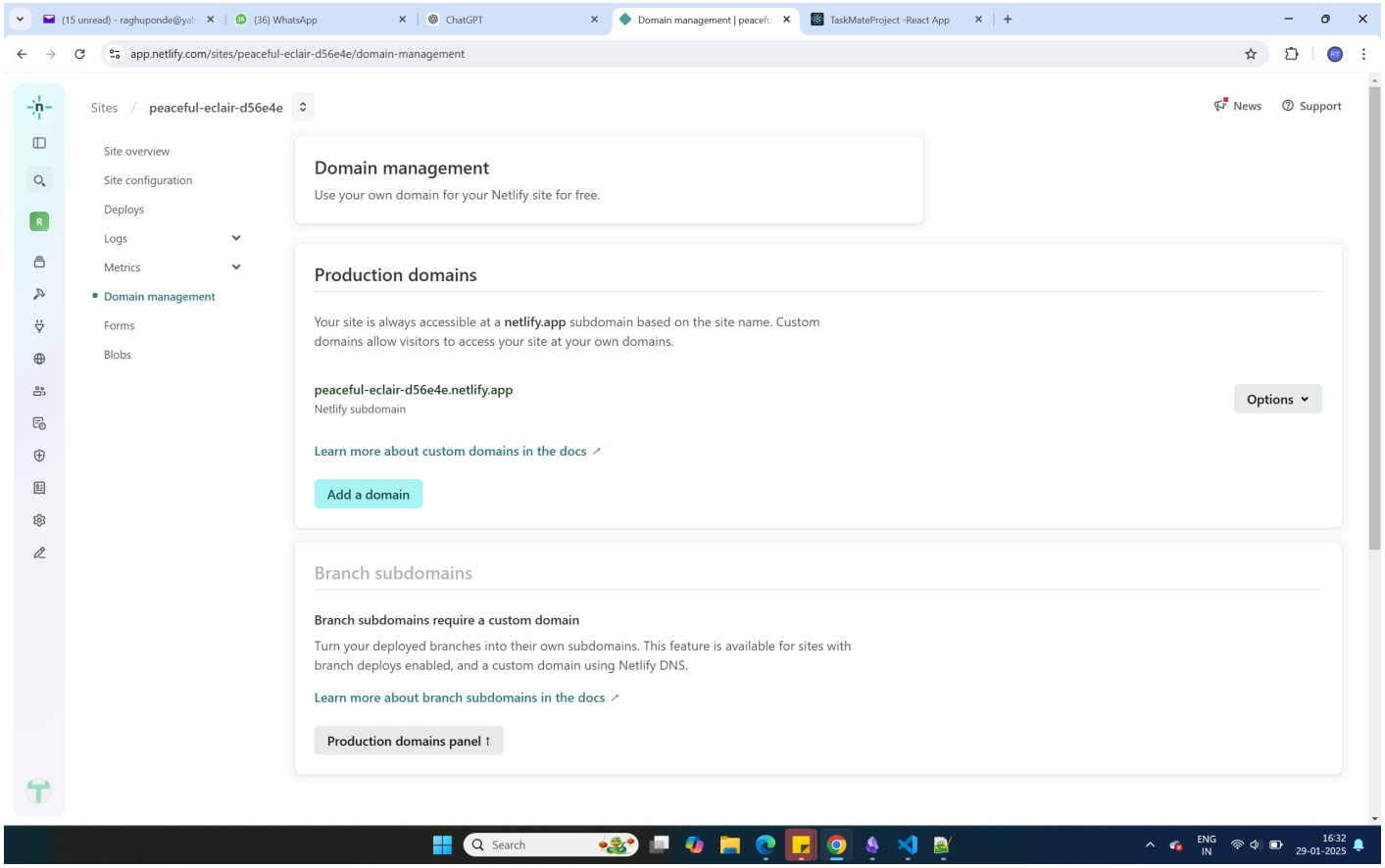
https://peaceful-eclair-d56e4e.netlify.app/

this URL it has got uploaded now next is



next i want to change my URL what can be done let us see here

then go to domain management and then options



say edit site name here okay

https://taskmateproject-raghu.netlify.app/

so updated site URL of mine is like this now ...