

JavaScript Introduction

Ok for this you can refer the notes which have kept already which you have learned so basically now I will start from react only not JavaScript but while teaching i will teach from those topics only okay which is there with me

JavaScript is a scripting language used both on client-side and server-side, that allows you to make web pages interactive.

Using JavaScript

Real-time Networking Apps

Web/Mobile Applications

Command-line Tools

Games

```
{  
var number = 10;
```

```
}
```

```
number = 30;
```

```
console.log(number);
```

so here latest value it will print and i can reassign the value okay

```
{
```

```
let number = 20;
```

```
console.log(number)
```

```
}
```

```
number = 34;
```

```
console.log(number)
```

so based on scope the value is printed just print it and check the output

```
{
```

```
const number = 20;
```

```
console.log(number)
```

```
}
```

```
number = 34;
```

```
console.log(number)
```

This also prints the value

```
{
```

```
const number = 20;
```

```
console.log(number)
```

```
}
```

```
const number = 34;
```

```
console.log(number)
```

This also prints the value

But this will not down one code

```
{
```

```
const number = 20;
```

```
console.log(number)
```

```
number = 45;
```

```
}
```

```
const number = 34;  
console.log(number)
```

okay as const values i cannot reassign it

Now functions i am creating

```
function test()  
{  
    console.log("hello world ");  
}  
test();  
function test2(num1, num2)  
{  
    return (num1 + num2);  
}
```

```
let sum = test2(12, 45);  
console.log(sum);
```

so above two functions using arrow is like this

```
const test = () => console.log("hello world ")  
test();
```

```
const test2 = (n1, n2) => n1 + n2;
```

```
let sum2 = test2(10, 34);
```

```
console.log(sum2);
```

I can write a function where only single value is passing

```
const test3 = (number) => number + 10;
```

```
let sum3 = test3(134);
```

```
console.log(sum3);
```

Now start learning classes

```
class test  
{
```

```
constructor()
{
    console.log("I am from constructor");
}
}

new test();
```

another version

```
class test {
constructor(number1, number2) {
this.number1 = number1;
this.number2 = number2;
}
addition() {
console.log(this.number1 + this.number2)
}
}
```

```
new test(10, 20).addition();
```

by creating the reference also u can give a call like this

```
let ref=new test(10,20)
```

```
ref. addition();
```

classes with inheritance example

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Rectangle and Square Example</title>
</head>
<body>

    <h1>Rectangle and Square Area Calculation</h1>

    <script>
        // Base class: Rectangle
        class Rectangle {
            constructor(width, height) {
                this.width = width;
                this.height = height;
            }
            calculateArea() {
                return this.width * this.height;
            }
        }
        class Square extends Rectangle {
            constructor(side) {
                super(side, side);
            }
            calculateArea() {
                return super.calculateArea();
            }
        }
        const square = new Square(5);
        console.log(`Area of square: ${square.calculateArea()}`);
    </script>

```

```

    }

    // Method to calculate the area of a rectangle
    area() {
        return this.width * this.height;
    }
}

// Derived class: Square inherits from Rectangle
class Square extends Rectangle {
    constructor(side) {
        // Call the parent class (Rectangle) constructor with both
        width and height equal to side
        super(side, side);
    }

    // No need to override the area method since it's inherited from
    Rectangle
}

// Create an instance of Rectangle and calculate its area
const rectangle = new Rectangle(10, 5);
console.log(`Area of Rectangle: ${rectangle.area()}`); // Output:
"Area of Rectangle: 50"

// Create an instance of Square and calculate its area
const square = new Square(4);
console.log(`Area of Square: ${square.area()}`); // Output: "Area of
Square: 16"
</script>

</body>
</html>

```

Now let us learn async and await features in JavaScript

Now let us learn about async and await in JavaScript suppose u are getting some delay in fetching web api earlier web i may get the response data or i may get some network error or I may not get the output only for those kind of functions earlier we use to write promises in js

What Are Promises in JavaScript?

A **Promise** in JavaScript is an object that represents the eventual completion (or failure) of an asynchronous operation and its resulting value. It allows you to associate handlers with an asynchronous action's eventual success value or failure reason. This makes it easier to work with asynchronous operations compared to traditional callback-based code.

A Promise has three states:

1. **Pending**: The initial state—when the operation is neither fulfilled nor rejected.
2. **Fulfilled**: The operation completed successfully, and the promise has a resulting value.
3. **Rejected**: The operation failed, and the promise has a reason for the failure.

Key Promise Methods:

- `then()` : Attaches callbacks for the success case (fulfilled state).
- `catch()` : Attaches callbacks for the failure case (rejected state).
- `finally()` : Attaches callbacks to run regardless of the result (optional).

Example of Promises

Let's look at an example where we simulate a promise that fetches some data after a delay and demonstrate both a success and an error case

```
```<html lang="en">

<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Promises with Web API Example</title>
</head>

<body>
 <h1>Promises with Web API Example</h1>
 <button onclick="fetchData()">Fetch User Data</button>

 <script>
 // Function to fetch user data using a manually created Promise
 function fetchData() {
 new Promise((resolve, reject) => {
 // Fetch user data from the API
 fetch('https://jsonplaceholder.typicode.com/users') // Fetch
all users
 .then(response => {
```

```

 // Check if the response is OK
 if (!response.ok) {
 return reject('Network response was not OK'); //
Reject the promise if response is not OK
 }
 return response.json(); // Parse the JSON data
 })
 .then(data => {
 resolve(data); // Resolve the promise with fetched
data
 })
 .catch(error => {
 reject('Failed to fetch user data: ' + error); //
Reject the promise if an error occurs
 });
})
.then(userData => {
 // Handle and log the fetched data
 console.log("User data fetched successfully:", userData);
})
.catch(error => {
 // Handle any error that occurred during the fetch
 console.error("Error:", error);
})
.finally(() => {
 // This block is executed regardless of success or failure
 console.log("Fetch operation completed.");
});
}
</script>

</body>

</html>

```

Now same code using `async` and `await` in JavaScript

```

<!DOCTYPE html>
<html lang="en">

<head>
 <meta charset="UTF-8">

```

```

<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Async/Await with Web API Example</title>
</head>

<body>
 <h1>Async/Await with Web API Example</h1>
 <button onclick="fetchData()">Fetch User Data</button>

 <script>
 // Async function to fetch user data
 async function fetchData() {
 try {
 // Fetch user data from the API
 const response = await
fetch('https://jsonplaceholder.typicode.com/users'); // Fetch all users

 // Check if the response is OK
 if (!response.ok) {
 throw new Error('Network response was not OK'); // Throw
an error if response is not OK
 }

 // Parse the JSON data
 const userData = await response.json();

 // Log the fetched data
 console.log("User data fetched successfully:", userData);
 } catch (error) {
 // Handle any error that occurred during the fetch
 console.error("Error:", error);
 } finally {
 // This block is executed regardless of success or failure
 console.log("Fetch operation completed.");
 }
 }
 </script>

</body>

</html>

```

new code with both examples

```
<!DOCTYPE html>
```

```

<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Async/Await with Web API Example</title>
</head>

<body>
 <h1>Async/Await with Web API Example</h1>
 <button onclick="consumeWebService()">ConsumeWebService</button>
 <button onclick="consumeWebServiceAsync()">consumewebserviceAsync</button>

 <script>
 function consumeWebService() {
 fetch("https://jsonplaceholder.typicode.com/todos").then(response => response.json())
 .catch(error => alert('something bad just happened: ()')).then(json => console.log(json));
 }

 async function consumeWebServiceAsync() {
 try {
 const response = await fetch("https://jsonplaceholder.typicode.com/todos");
 const json = await response.json();
 console.log(json);
 }
 catch (ex) {
 alert('something bad just happened: ()');
 }
 }
 </script>
</body>
</html>

```

Next concept is of DE structuring

```

const person = {
 name: "john",
 age: 25,
 mail: "john@gmail.com"
}
console.log(age);

```

this code will give me error so i it will work when i write like this

```
console.log(person.age)
```

Now I have to de structure it now

```
const { name, age, mail } = person;
console.log(age);
```

Next let us go with spread operator usage

```
arr11 = [10, 20, 30, 40]
```

```
arr2 = [50, 60]
```

```
arr2 = [...arr11, 50, 60]
```

```
console.log(arr2)
```

and same thing in classes and objects

```
var obj1 = {
 name: 'john'
}
var obj2 = {
 ...obj1, age: 25,
 mail: "mail@gmail.com"
}
console.log(obj2)
```

## Map Syntax

Variable.map( (element) => print(element) )

---

example 1:

```
var arr = [10, 20, 30, 40, 50]
arr.map((element) => console.log(element))
```

example 2:

---

```
const numbers = [1, 2, 3, 4, 5];
const squares = numbers.map(value => value * value);
console.log(squares);
```

example 3:

---

```
const people = [
{ id: 1, name: 'Felipe', country: 'DR' },
{ id: 2, name: 'Scott', country: 'USA' },
{ id: 3, name: 'Jennifer', country: 'Canada' },
]
```

```
const ids = people.map(person => person.id);
console.log(ids);
```

in mapping i can do modification of data of array

here mapping will always return a value but foreach will not return value ..

```
var data = [1, 2, 3, 4, 5];
data.forEach(function(element)
{
 console.log(element);
})
var data2 = [1, 2, 3, 4, 5];
data2.forEach(element => console.log(element));
```

Now I want to check for every value for that every function is there

```
var data = [1, 2, 3, 4]
var output = data.every(
function (element) {
return element > 0;
}
)
console.log(output)
```

here all conditions are getting satisfied so true will come and it will become false if i write >2 or something like that

here in the same place i can use some also then in that even some elements in list are true also means then it is true

```
var data = [1, 2, 3, 4]
var output = data.some(
function (element) {
return element > 2;
}
)
```

```
console.log(output)
```

Now let us try to understand indexOf and lastIndexOf

```
var data = [12, 78, 23, 19]
console.log(data.indexOf(23));
var data2 = [12, 78, 23, 19, 12, 34, 23]
console.log(data2.lastIndexOf(23))
```

here i am checking the positions of number 23 in both arrays

Then stringify is used to convert the data into json format

```
var data = {
name: 'John',
age: 28
}
var json = JSON.stringify(data);
console.log(json);
```

Now let us discuss date functions now

```
var date = new Date();
console.log(date.valueOf());
```

Gives the value of date till now in milliseconds okay

rest operator usage which is ...

This below coding we have seen earlier also in spread concept okay

```
function user(name, ...remaining) {
console.log(name + " : " + remaining)
}
```

```
user('John', 25, "mail@gmail.com", 123456)
```

here in function how to use we can see and you have to use that at the last of the function

## Filter Syntax

### array.filter( (element) => condition )

```
var arr = [10, 20, 30, 40, 50]
```

```
let filtered = arr.filter((element) => element > 20);
```

```
console.log(filtered) // will print in the form of array
```

same thing which is having array u want to print it using map

```
filtered.map((element) => console.log(element));
```

in the form of line by line it will print the value okay

Lets understand now the concept of modules in js which will be transferred to React Js later

JavaScript modules allow you to split your code into separate files and reuse them across different parts of your application. The **ES6 module system** uses the `export` and `import` keywords to handle this. Here's a demo that shows how to use modules in JavaScript.

## Example: JavaScript Modules Demo

We'll create a simple demo where we have a **module** that handles rectangle and square area calculations. This will demonstrate how to use `export` and `import` statements.

/project

  |—— index.html

```
|── shapes.js
└── app.js
```

## Step 1: Create the Module ( shapes.js )

This module will contain two classes: `Rectangle` and `Square`, along with their `area()` methods. We'll export these classes.

```
// shapes.js

// Rectangle class
export class Rectangle {
 constructor(width, height) {
 this.width = width;
 this.height = height;
 }

 // Method to calculate the area of a rectangle
 area() {
 return this.width * this.height;
 }
}

// Square class that extends Rectangle
export class Square extends Rectangle {
 constructor(side) {
 // Call the parent constructor with equal width and height
 super(side, side);
 }
}
```

## Step 2: Create the Main Application ( app.js )

This is where we import the classes from `shapes.js` and use them in our application.

```
// app.js

// Importing the Rectangle and Square classes from shapes.js
import { Rectangle, Square } from './shapes.js';
```

```
// Create a rectangle and calculate its area
const rectangle = new Rectangle(10, 5);
console.log(Area of Rectangle: ${rectangle.area()});

// Create a square and calculate its area
const square = new Square(4);
console.log(Area of Square: ${square.area()});
```

## Step 3: Create the HTML File ( index.html )

In the HTML file, we'll reference `app.js` as a module using the `type="module"` attribute in the script tag.

```
<!DOCTYPE html>
<html lang="en">

<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>JavaScript Modules Demo</title>
</head>

<body>
 <h1>JavaScript Modules Demo</h1>

 <script type="module" src="app.js"></script>
</body>

</html>
```

## React Starting Now

---

some links

<https://www.dotnetmastery.com/Home/Details?courseld=29> (url for practise)

# What is React?

- **React is** Front End Javascript Framework
- **React is** Declarative
- **React is** Composable / Component based
- **React is** FAST!
- **Maintained by** META
- **React is responsible for maintaining/managing the UI Aspect.**

<https://legacy.reactjs.org/docs/react-component.html>

<https://legacy.reactjs.org/docs/getting-started.html#try-react>

<https://legacy.reactjs.org/docs/add-react-to-a-website.html>

so first i created one folder and in that folded added index.html and index.js files and the added some online libraries into html page like this

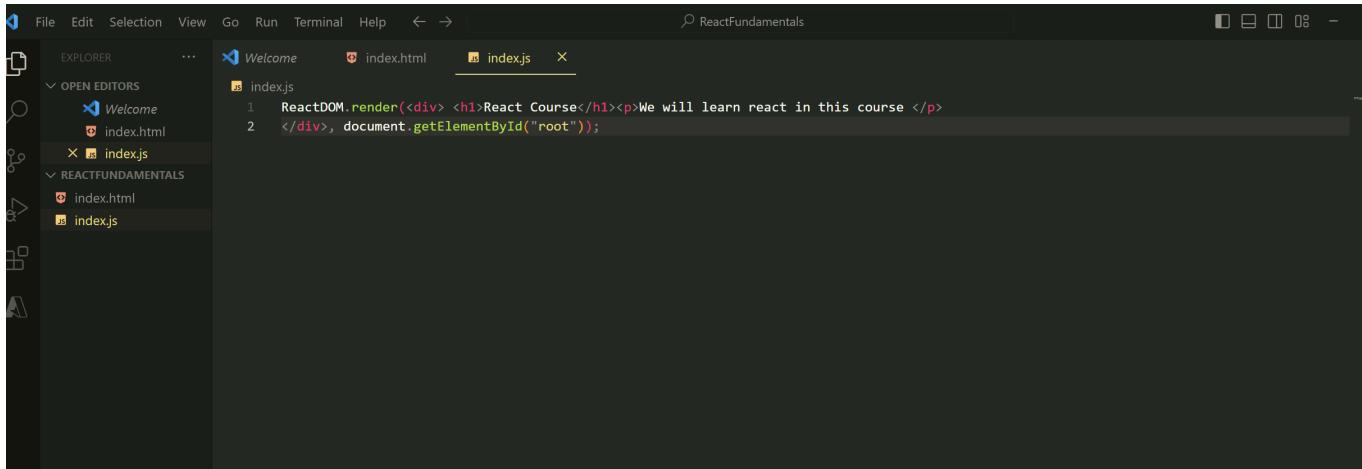
```
<html>
<head>
 <script src="https://unpkg.com/react@18/umd/react.development.js"
crossorigin></script>
 <script src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"
crossorigin></script>
<script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>
 <script src="index.js" type="text/babel"></script>
</head>
<body>
 <h1>Hello World</h1>
</body>
</html>
```

and in index.js file write like this

```
ReactDOM.render(<p>Hello World from react</p>,
document.getElementById("root"));
```

so here in root division tag i will render the paragraph information .Open it in live server so here we have to decide what we have to render and where to render that

further adding of new data in root tag



so recommended approach is like this in js file

```
const root = ReactDOM.createRoot(document.getElementById("root"))

root.render(<div>

 <h1>React Course </h1>

 <p>We will learn react in this course </p>

</div>);
```

Now again open it in live server it will show the output like this

```
ReactDOM.createRoot(document.getElementById("root")).render(<div>

 <h1>React Course </h1>
```

```
<p>We will learn react in this course </p>
</div>);
```

still compact i had made it okay

Now what is JSX here whatever code i had written in

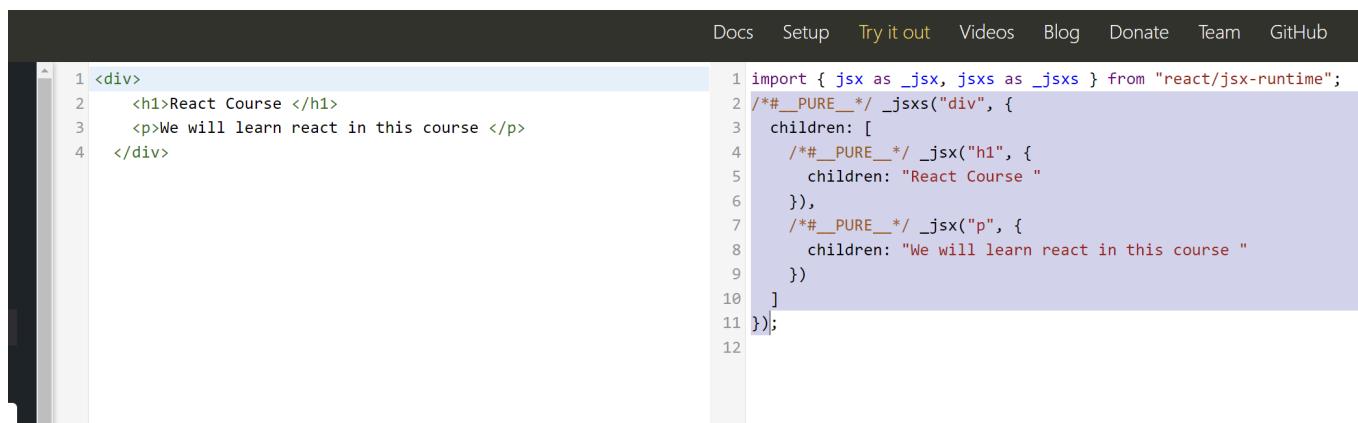
## React Course

We will learn react in this course

in render is JSX syntax open babel and after doing try it out in below url

<https://babeljs.io/>

paste the above code and see the output at right side



The screenshot shows the Babel.js website's "Try it out" feature. On the left, there is an input code editor containing the following JSX:

```
1 <div>
2 <h1>React Course </h1>
3 <p>We will learn react in this course </p>
4 </div>
```

On the right, the output is displayed as transpiled JavaScript:

```
1 import { jsx as _jsx, jsxs as _jsxs } from "react/jsx-runtime";
2 /*#__PURE__*/ _jsxs("div", {
3 children: [
4 /*#__PURE__*/ _jsx("h1", {
5 children: "React Course "
6 }),
7 /*#__PURE__*/ _jsx("p", {
8 children: "We will learn react in this course "
9 })
10],
11});
12
```

now let us go to <https://create-react-app.dev/> website and do create one app

npx create-react-app my-app

npx create-react-app taskopedia

so using above command create one app in ReactappPractise folder

so ctrl+ ` prompt will open and type the above command okay

Success! Created taskopedia at

D:\React\_Bruhugen\ReactFundamentals\ReactappPractise\taskopedia

Inside that directory, you can run several commands:

npm start

Starts the development server.

npm run build

Bundles the app into static files for production.

npm test

Starts the test runner.

npm run eject

Removes this tool and copies build dependencies, configuration files and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

```
cd taskopedia
```

```
npm start
```

Happy hacking!

PS D:\React\_Bruhugen\ReactFundamentals\ReactappPractise>

like this it will show me

The screenshot shows a terminal window in VS Code with the following content:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE

Success! Created taskopedia at D:\React_Bruhugen\ReactFundamentals\ReactappPractise\taskopedia
Inside that directory, you can run several commands:

npm start
Starts the development server.

npm run build
Bundles the app into static files for production.

npm test
Starts the test runner.

npm run eject
Removes this tool and copies build dependencies, configuration files
and scripts into the app directory. If you do this, you can't go back!

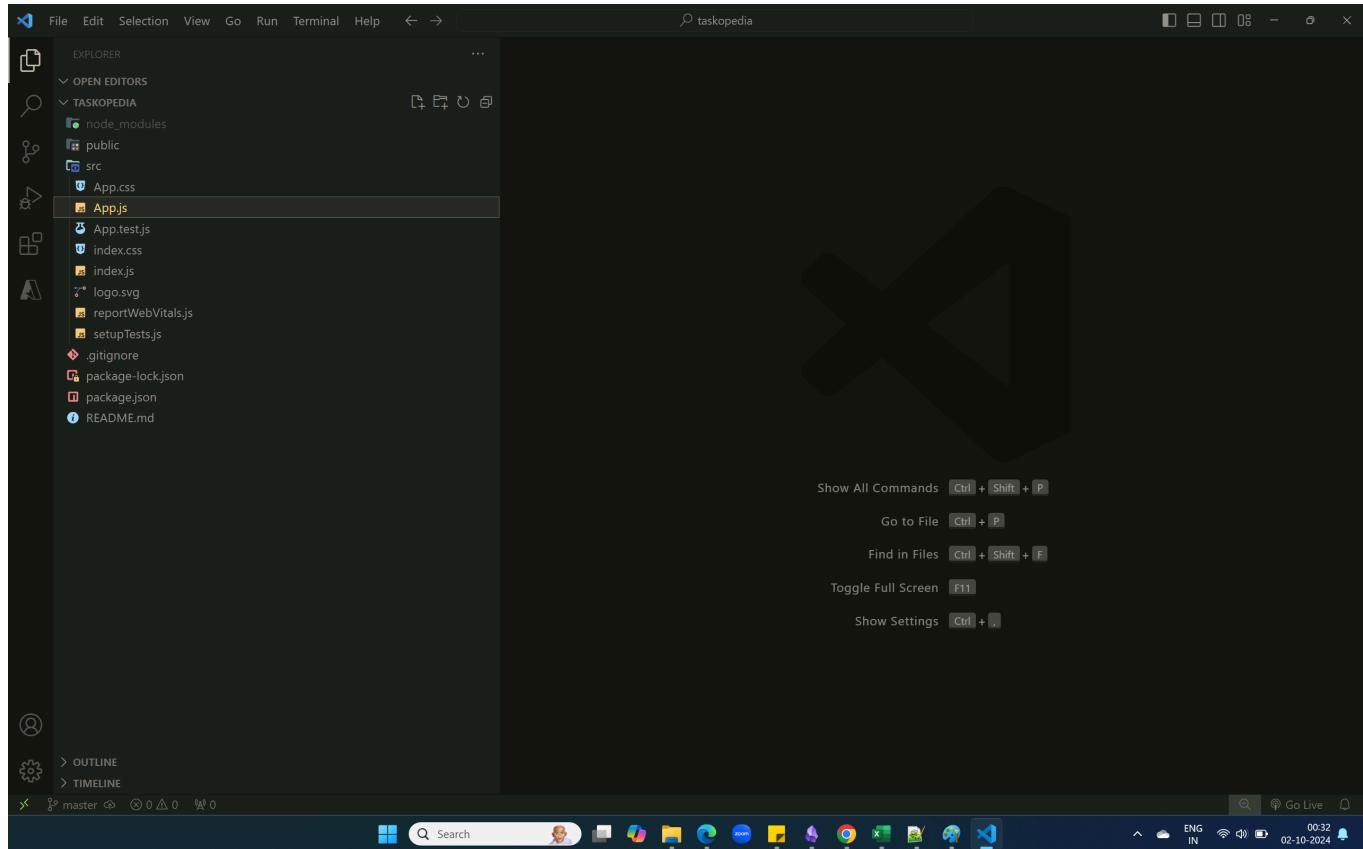
We suggest that you begin by typing:

cd taskopedia
npm start

Happy hacking!
PS D:\React_Bruhugen\ReactFundamentals\ReactappPractise>
```

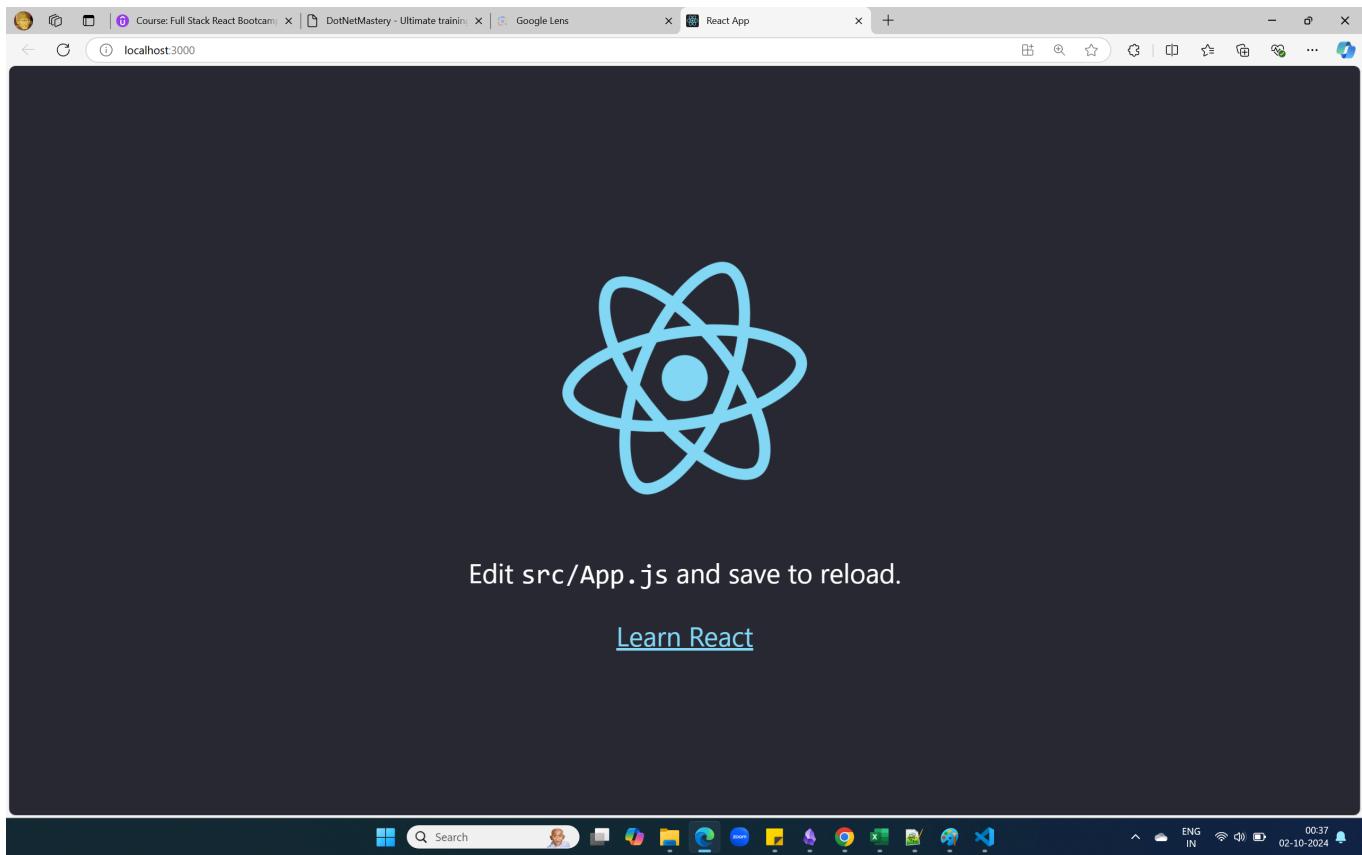
The terminal window has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is selected), PORTS, and AZURE. At the bottom, there are icons for file operations, search, and zoom.

Now if go to folder taskopedia and from vscode say select folder okay so u can close the vscode and open just that folder which is taskopedia okay

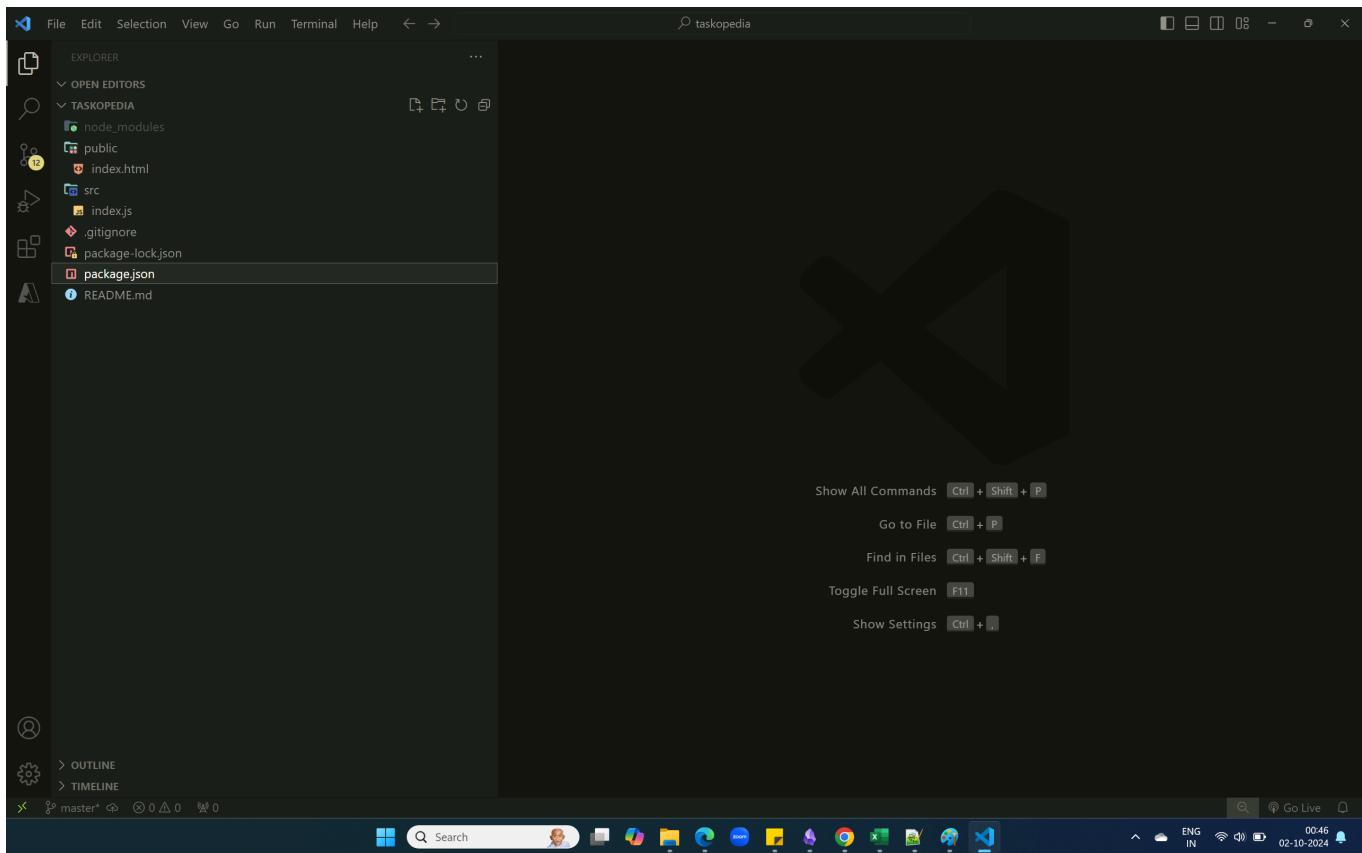


so it will look like this okay

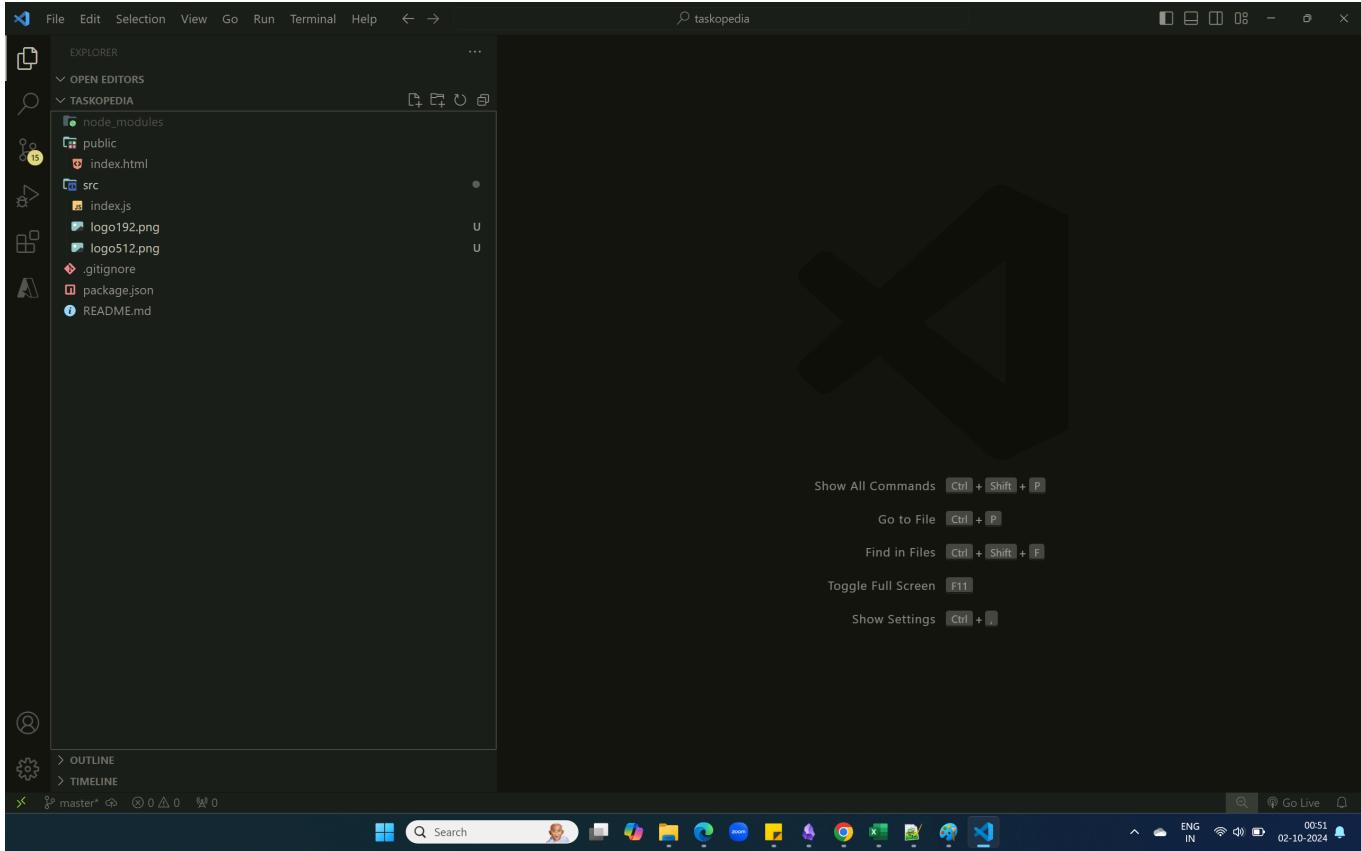
now run the command like this npm start from taskopedia okay



Now i will do some clean up of files and put react logo to src and change the name of it also



so again changing name and keeping it



Here image is getting deleted if i deleted any one so i am keeping them there both of them u can see in the above image now again run npm start now some errors will come try to rectify that now

## Compiled with problems:

```
ERROR in ./src/index.js 6:0-21
Module not found: Error: Can't resolve './index.css' in
'D:\React_Bruhugen\ReactFundamentals\ReactappPractise\taskopedia\src'

ERROR in ./src/index.js 7:0-24
Module not found: Error: Can't resolve './App' in
'D:\React_Bruhugen\ReactFundamentals\ReactappPractise\taskopedia\src'

ERROR in ./src/index.js 8:0-48
Module not found: Error: Can't resolve './reportWebVitals' in
'D:\React_Bruhugen\ReactFundamentals\ReactappPractise\taskopedia\src'
```

so the modified code of js

```
import React from 'react';

import ReactDOM from 'react-dom/client';

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(
 <React.StrictMode>

 </React.StrictMode>
);
```

now run the code with npm start and see okay  
and also change the html file also like this

```
<!DOCTYPE html>

<html lang="en">

 <head>

 <meta charset="utf-8" />

 <meta name="viewport" content="width=device-width, initial-scale=1" />

 <meta name="theme-color" content="#000000" />

 <meta
 name="description"
 content="Web site created using create-react-app"

 />
```

```
<title>React App</title>

</head>

<body>

<noscript>You need to enable JavaScript to run this app.</noscript>

<div id="root"></div>

</body>

</html>
```

To render some thing we have to

write some code so written hello world by removing strict mode etc

```
import React from 'react';

import ReactDOM from 'react-dom/client';

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(

<h1>Hello World!</h1>

);
```

so now instead of rendering common i will create now component now and call it in root.render method like this okay

```
import React from 'react';

import ReactDOM from 'react-dom/client';

function MainBody()
```

```

{

 return (
 <div>
 <p>In this course we learn react by building taskopedia </p>

 Call Ben
 Go to walmart

 </div>
)
}

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(
 <MainBody />
);

```

so now if i change the MainBody to small one like mainBody then it will not show output as react follows pascal notation here okay

```

import React from 'react';

import ReactDOM from 'react-dom/client';

function mainBody()
{

```

```
return (
 <div>
 <p>In this course we learn react by building taskopedia </p>

 Call Ben
 Go to walmart

 </div>
)

}

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(
<mainBody/>
);
```

```
index.js 1, M X index.html M
src > index.js > ...
1 import React from 'react';
2 import ReactDOM from 'react-dom/client';
3
4 function MainBody()
{
5
6 return (
7 <div>
8 <p>In this course we learn react by building taskopedia </p>
9
10 Call Ben
11 Go to walmart
12
13 </div>
14)
15 }
16 const root = ReactDOM.createRoot(document.getElementById('root'));
17 root.render([
18
19 <MainBody />
20 <MainBody />
21 <MainBody />
22
23]);
24
25
26
```

so this will give me error if i print same multiple times things need to enclosed in in some parent tag like this

```
import React from 'react';
import ReactDOM from 'react-dom/client';

function MainBody()

{
 return (
 <div>
 <p>In this course we learn react by building taskopedia </p>

 Call Ben

 </div>
)
}
```

```

 Go to walmart

</div>

)

}

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(
 <div>

 <MainBody />

 <MainBody />

 <MainBody />

 </div>
);

```

so now adding Header and footer it will look like this

```

import React from 'react';

import ReactDOM from 'react-dom/client';

function Header()

{

 return(
 <h1> The react Course </h1>

```

```

)

}

function Footer()
{
 return (
<h1> This is footer </h1>
)
}

function MainBody()
{
 return (
<div>
 <p>In this course we learn react by building taskopedia </p>

 Call Ben
 Go to walmart

</div>
)
}
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
<div>
 <Header />
 <MainBody />
 <Footer />
</div>
);

);

```

Now nested components example is below

```

import React from 'react';

import ReactDOM from 'react-dom/client';

function MainHeader()
{

```

```
return (<h1> The react Course </h1>);

}

function SubHeader()

{

 return (<h2> The course is very exciting </h2>);

}

function Header()

{

 return(

 <div>

 <MainHeader></MainHeader>

 <SubHeader></SubHeader>

 </div>

)

}

function Footer()

{

 return (

 <h1> This is footer </h1>
)
}
```

```
)

}

function MainBody()

{

 return (

 <div>

 <p>In this course we learn react by building taskopedia </p>

 Call Ben

 Go to walmart

 </div>

)

}

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(

 <div>

 <Header />

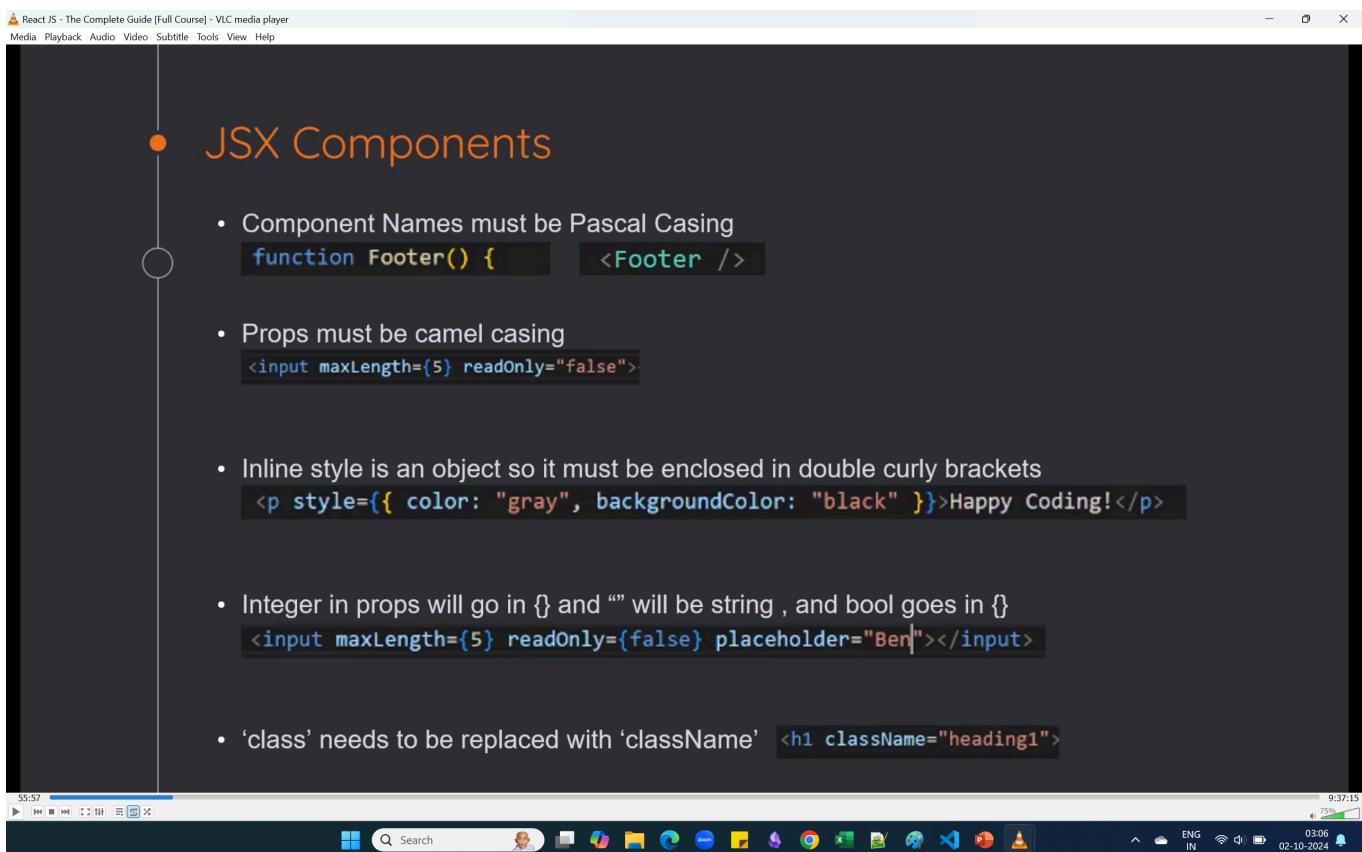
 <MainBody />

 <Footer />

 </div>

```

```
);
```



this is bootstrap link page and in this

<https://getbootstrap.com/docs/5.3/getting-started/introduction/>

css link

js link

and I had also added one style.css file in CSS folder of src folder here CSS is a new folder here i added okay

```
.heading2 {
 color: brown;
 font-size: large;
}
```

```
.h2 {
 color: chocolate;
}
```

so now in total till now both the pages will look like this

index.html

```
<!DOCTYPE html>

<html lang="en">

 <head>

 <meta charset="utf-8" />

 <meta name="viewport" content="width=device-width, initial-scale=1" />

 <meta name="theme-color" content="#000000" />

 <meta
 name="description"
 content="Web site created using create-react-app"

 />

 <!-- <link href="../src/CSS/style.css" rel="stylesheet" />-->

 <link
 href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
 rel="stylesheet"

 integrity="sha384-QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh0JMHjY6hW+ALEwIH"
 crossorigin="anonymous">

 <script
 src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.8/dist/umd/popper.min.js"
 ".
```

```

integrity="sha384-
I7E8VVD/ismYTF4hNIPjVp/Zjvgyol6VFvRkX/vR+Vc4jQkC+hVqc2pM80Dewa9r"
crossorigin="anonymous">></script>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.min.js"

integrity="sha384-
0pUGZvbkm6XF6gxjEnlmuGrJXVbNuzT9qBBavbLwCs0GabYfZo0T0to5eqruptLy"
crossorigin="anonymous">></script>

<title>React App</title>

</head>

<body>

<noscript>You need to enable JavaScript to run this app.</noscript>

<div id="root"></div>

</body>

</html>

```

index.js file

```

import React from 'react';

import ReactDOM from 'react-dom/client';

import './CSS/style.css';

function MainHeader()

{

 return (<div><h2 className='text-primary'> The React Course </h2>

<h2 className='heading2'> The React Course2 </h2>

```

```
</div>

);

}

const subHeaderStyle = {

 color: "blueviolet",

 backgroundColor: "lightgray"

}

function SubHeader()

{

 return (<p style={subHeaderStyle}> The course is very exciting </p>);

}

function Header()

{

 return(

 <div>

 <MainHeader></MainHeader>

 <SubHeader></SubHeader>

 </div>

)

}

function Footer()

{

 return (

<p style={{ color:"gray",backgroundColor:"black" }}> This is footer </p>
```

```
)

}

function MainBody()

{

 return (

 <div>

 <p>In this course we learn react by building taskopedia </p>

 Call Ben

 Go to walmart

 </div>

)

}

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(

 <div>

 <Header />

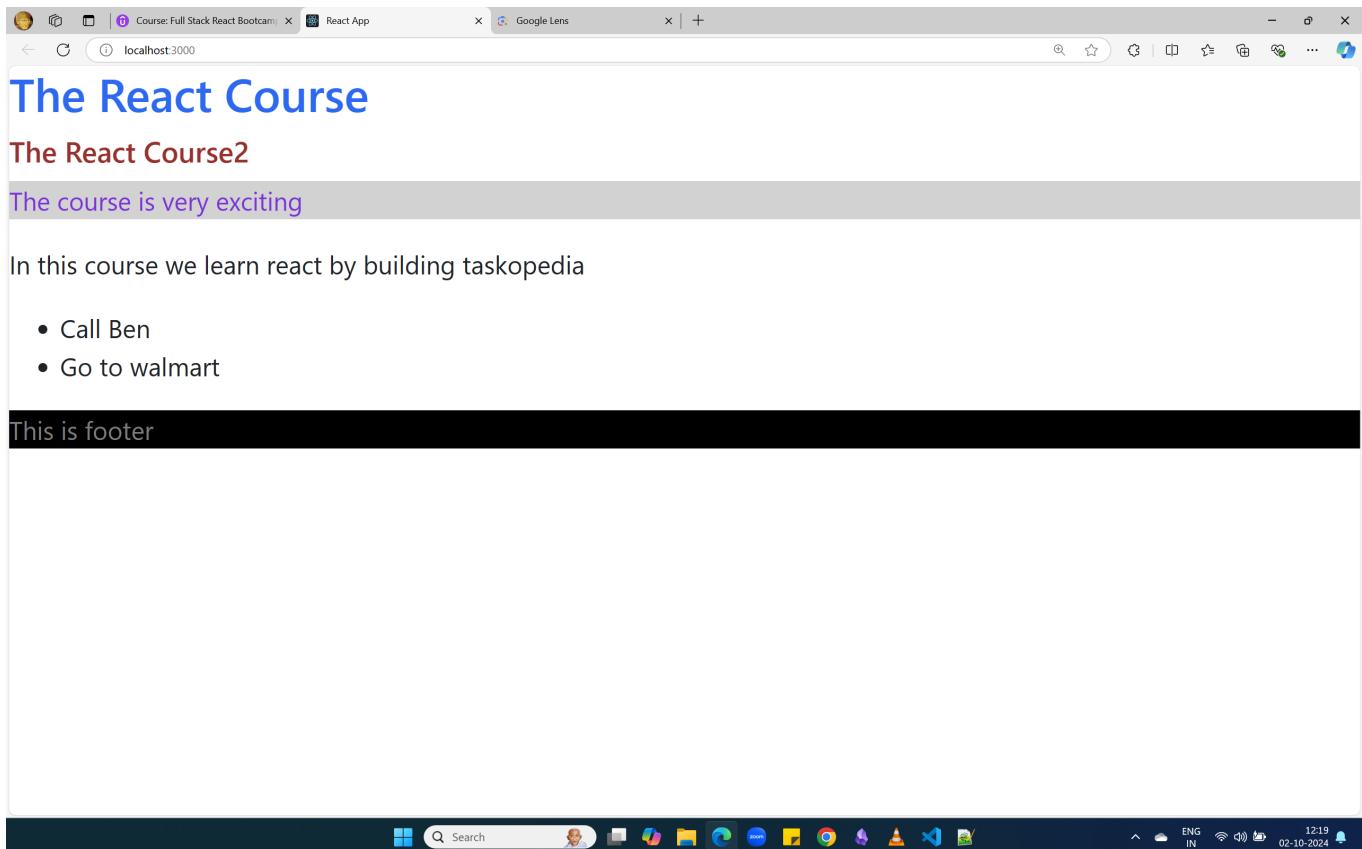
 <MainBody />

 <Footer />

 </div>

);
```

The output will look like this



when the link is inside code src then we cannot put path so that is why i am doing importing in code you can see okay ..

Till now further coding where in students component shown the function of variables okay

```
import React from 'react';

import ReactDOM from 'react-dom/client';

import "./CSS/style.css";

function MainHeader()

{

 return (<div><h2 className='text-primary'> The React Course </h2>

 <h2 className='heading2'> The React Course2 </h2>

 </div>

);
```

```
}

const subHeaderStyle = {

 color: "blueviolet",

 backgroundColor: "lightgray"

}

function SubHeader()

{

 return (<p style={subHeaderStyle}> The course is very exciting </p>);

}

function Header()

{

 return(

<div>

 <MainHeader></MainHeader>

 <SubHeader></SubHeader>

</div>

)

}

function Footer()

{

 return (

<p style={{ color:"gray",backgroundColor:"black" }}> This is footer


```

```
</p>

)

}

function Students(){

 const fullName = "Kris Walley";

 const programmingExp = 2;

 return(
 <div className="container p-4">

 <div className="row">

 Students Enrolled

 </div>

 <div className="row border">

 <div className="col-2">

 <img src={`https://ui-avatars.com/api/?name=${fullName}`}
 className='w-100'>

 </div>

 <div className="col-10">

 {fullName}

 Programming Experience {programmingExp} years </div>

 </div>

 </div>

)
 }
}

function MainBody()
```

```
{

 const WhatWeLearn = "React Js"

 const lectureCount=3

 return (

 <div>

 <p>In this course we learn {WhatWeLearn} by building taskopedia

 Total Lecture :{lectureCount}

 </p>

 Basic Foundation

 Functional and Class Components

 <div>

 Enter Tasks : <input maxLength={5} readOnly={false} placeholder='Ben'>
 </input>

 </div>

 </div>

)

}

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(

 <div>

 <Header />
```

```
<MainBody />

<Students />

<Footer />

</div>

);
```

Now i need to add the components which are related into one file okay

Now take all header components from index.js file and keep it in header.jsx file and also make it export default so that if u give different name also it will get substituted here

```
function MainHeader()

{

 return (<div><h2 className='text-primary'> The React Course </h2>

<h2 className='heading2'> The React Course2 </h2>

</div>

);

}

const subHeaderStyle = {

 color: "blueviolet",

 backgroundColor: "lightgray"

}

function SubHeader()

{

 return (<p style={subHeaderStyle}> The course is very exciting </p>);

}
```

```
function Header()

{
 return(
 <div>
 <MainHeader></MainHeader>
 <SubHeader></SubHeader>
 </div>
)
}

export default Header;
```

same thing in image shown below

The screenshot shows the VS Code interface. The left sidebar displays a file tree with the following structure:

- OPEN EDITORS:
  - index.js M
  - header.jsx U X
  - style.css U
  - index.html M
- TASKOPEDIA:
  - node\_modules
  - public
  - src
  - CSS
  - header.jsx U
  - index.js M
  - logo192.png U
  - logo512.png U
  - .gitignore
  - package.json
  - README.md

The right pane shows the content of the 'header.jsx' file:

```
src > header.jsx > ...
1 function MainHeader()
2 {
3 return (<div><h2 className='text-primary'> The React Course </h2>
4 <h2 className='heading2'> The React Course2 </h2>
5 </div>
6);
7 }
8 const subHeaderStyle = {
9 color: "blueviolet",
10 backgroundColor: "lightgray"
11 }
12 function SubHeader()
13 {
14 return (<p style={subHeaderStyle}> The course is very exciting </p>);
15 }
16 function Header()
17 {
18 return(
19 <div>
20 <MainHeader></MainHeader>
21 <SubHeader></SubHeader>
22 </div>
23)
24 }
25
26 export default Header;
```

and now in Index.js file import it and i had given different name which is possible

```
import React from 'react';

import ReactDOM from 'react-dom/client';

import "./CSS/style.css";

import TempHeader from './header';

function Footer()

{

 return (

 <p style={{ color:"gray",backgroundColor:"black" }}> This is footer
</p>

)

}
```

```
}

function Students(){
 const fullName = "Kris Walley";
 const programmingExp = 2;

 return(
 <div className="container p-4">
 <div className="row">
 Students Enrolled
 </div>
 <div className="row border">
 <div className="col-2">
 <img src={`https://ui-avatars.com/api/?name=${fullName}`}
 className='w-100'>
 </div>
 <div className="col-10">
 {fullName}

 Programming Experience {programmingExp} years </div>
 </div>
 </div>
)
 }
}

function MainBody()
{
 const WhatWeLearn = "React Js"
```

```
const lectureCount=3

return (
 <div>

 <p>In this course we learn {WhatWeLearn} by building taskopedia

 Total Lecture :{lectureCount}

 </p>

 Basic Foundation
 Functional and Class Components

 <div>

 Enter Tasks : <input maxLength={5} readOnly={false} placeholder='Ben'>
 </input>

 </div>

 </div>
)

}

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(
 <div>

 <TempHeader />
 <MainBody />
 <Students />
 </div>
)
```

```
<Footer />

</div>

);
```

here above i can use TempHeader also or normally Header also as it is default i can use it okay but now for time being again I am changing it to Header only okay

```
import React from 'react';

import ReactDOM from 'react-dom/client';

import "./CSS/style.css";

import Header from './header';

function Footer()

{

 return (

 <p style={{ color:"gray",backgroundColor:"black" }}> This is footer

</p>

)

}

function Students(){

 const fullName = "Kris Walley";

 const programmingExp = 2;

 return(

<div className="container p-4">

 <div className="row">

 Students Enrolled

 </div>

</div>

)

}
```

```
</div>

<div className="row border">

 <div className="col-2">

 <img src={`https://ui-avatars.com/api/?name=${fullName}`}
 className='w-100'>

 </div>

 <div className="col-10">

 {fullName}

 Programming Experience {programmingExp} years </div>

 </div>

</div>

)

}

function MainBody()

{

 const WhatWeLearn = "React Js"

 const lectureCount=3

 return (

 <div>

 <p>In this course we learn {WhatWeLearn} by building taskopedia

 Total Lecture :{lectureCount}

 </p>


```

```

 Basic Foundation

 Functional and Class Components

 <div>

 Enter Tasks : <input maxLength={5} readOnly={false} placeholder='Ben'>
 </input>

 </div>

</div>

)

}

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(
 <div>

 <Header />

 <MainBody />

 <Students />

 <Footer />

 </div>
);


```

so here i had separated out one component which is header component from index.js file i am using .jsx file because it is nicely compatible with vscode and i can include babel code here and also can include react components and also can embed the css ,variable and html code in between okay .

so if u don't want to write like this externally u can write like this also it is okay

```

export default function Header()

{
 return(
 <div>

 <MainHeader></MainHeader>

 <SubHeader></SubHeader>

 </div>
)
}

```

In student Component I want to pass experience and name properties i have it with me

```

const fullName = "Kris Walley";

const programmingExp = 2;

```

so what i will do i will delete these two properties and send them as props from the function and that only i will substitute there in index.js file .

```

export default function Student(props){

 return(
 <div className="container p-4">

 <div className="row border">

 <div className="col-2">

 <img src={`https://ui-avatars.com/api/?name=${props.name}`}
 className='w-100'>

 </div>

 <div className="col-10">

```

```

{props.name}

 Programming Experience {props.experience} years </div>

</div>

</div>

)
}

}

```

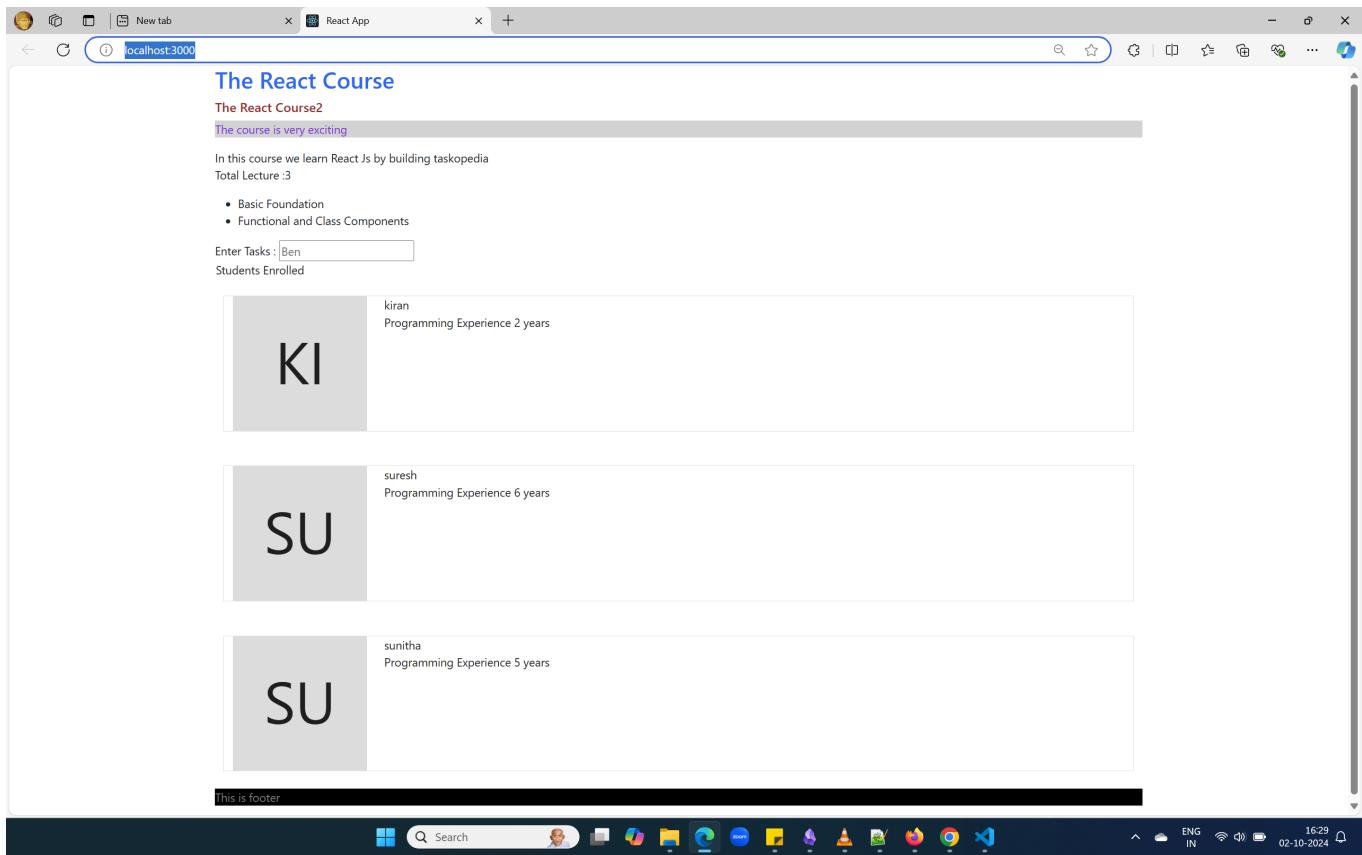
next in index.js file write like this

```

root.render(
 <div className='container'>
 <Header />
 <MainBody />
 <div className="row"> Students Enrolled </div>
 <Student experience={2} name="kiran"/>
 <Student experience={6} name="suresh"/>
 <Student experience={5} name="sunitha" />
 <Footer />
 </div>
);

```

so accordingly u will get the output like this what u have given as props



so props here is the fixed name How to use now images of third party means nugget one  
<https://fakerjs.dev/>

In this go to api and search image in text box then

To install <https://fakerjs.dev/guide/>

npm install @faker-js/faker --save-dev

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure with files like index.js, Header.jsx, Student.jsx, style.css, and index.html.
- Code Editor:** Displays the content of index.js, which includes a MainBody function and a root render block.
- Terminal:** Shows the command `npm install @faker-js/faker --save-dev` being run, followed by audit results and a fix command.
- Bottom Status Bar:** Includes icons for search, file operations, and system status.

now run the above command at taskopedia folder and go to package.json

The screenshot shows the file explorer with package.json selected. The content of package.json is as follows:

```

{
 "name": "taskopedia",
 "version": "1.0.0",
 "description": "A simple React application for managing student enrollment.",
 "main": "index.js",
 "scripts": {
 "start": "node index.js"
 },
 "dependencies": {
 "@faker-js/faker": "^9.0.3"
 },
 "devDependencies": {
 "last": "1.0.0"
 }
}

```

Now go to Student. Jsx component and import the package

A screenshot of Visual Studio Code showing a file named `Student.jsx`. The code imports `@faker-js/faker` and defines a `Student` component. The project structure in the Explorer sidebar shows files like `index.html`, `Header.jsx`, and `package.json`.

```
import faker from "@faker-js/faker";
export default function Student(props) {
 return (
 <div className="container p-4">
 <div className="row border">
 <div className="col-2">
```

in usage of that faker website i had seen the way to import it

A screenshot of the [Faker.js documentation](#). The left sidebar has sections for **Guide** (Getting Started, Usage, Localization, Frameworks, Randomizer, Unique Values, Upgrading to v9), **API**, and **About**. The main content area is titled **Node.js** and shows code examples for importing Faker into Node.js. It includes tabs for `esm` and `cjs`, and a code block with examples:

```
import { faker } from '@faker-js/faker';
// or, if desiring a different locale
// import { fakerDE as faker } from '@faker-js/faker';

const randomName = faker.person.fullName(); // Rowan Nikolaus
const randomEmail = faker.internet.email(); // Kassandra.Haley@erich.biz
```

For more information on selecting and customizing a locale, please refer to our [Localization Guide](#).

now go to API ,images and in that avatar you go

<https://fakerjs.dev/api/image.html#avatar>

The screenshot shows the Faker.js API documentation page. The left sidebar has a navigation menu with 'Guide' and 'API' sections. Under 'API', there are links for 'Overview', 'Faker', 'SimpleFaker', 'Randomizer', 'Utilities', 'Modules' (with 'Airline' and 'Animal' sub-links), and 'Examples'. The main content area is titled 'avatar' and describes it as generating a random avatar image URL. It notes availability since v2.0.1 and returns a string. A code snippet shows the function signature: `function avatar(): string;`. There is also a copy icon.

`faker.image.avatar()`

and paste in code and see the output okay

```
import { faker } from "@faker-js/faker"

export default function Student(props) {

 return(
 <div className="container p-4">

 <div className="row border">

 <div className="col-2">

 </div>

 <div className="col-10">
 {props.name}

 Programming Experience {props.experience} years </div>
 </div>
 </div>
)
)
}
```

```
}
```

another website is there for images <https://lorem.space/>

and from here take any URL: <https://api.lorem.space/image/movie?w=150&h=220>

by choosing any category okay

index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './CSS/style.css';

import Header from './Header';
import Student from './Student';

function Footer()
{
 return (
 <p style={{ color:"gray",backgroundColor:"black" }}> This is footer
</p>
)
}

function MainBody()
{
 const WhatWeLearn = "React Js"
 const lectureCount=3
 return (

```

```
<div>

 <p>In this course we learn {WhatWeLearn} by building taskopedia

 Total Lecture :{lectureCount}

 </p>

 Basic Foundation

 Functional and Class Components

 <div>

 Enter Tasks : <input maxLength={5} readOnly={false} placeholder='Ben'>
 </input>

 </div>

</div>

)

}

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(

<div className='container'>

 <Header />

 <MainBody />

 <div className="row"> Students Enrolled </div>

 <Student experience={2} name="kiran"
headshot="https://randomuser.me/api/portraits/men/10.jpg" />

 <Student experience={6} name="suresh"
```

```

headshot="https://randomuser.me/api/portraits/women/10.jpg" />

<Student experience={5} name="sunitha"
headshot="https://randomuser.me/api/portraits/men/20.jpg" />

<Footer />

</div>

);

```

In the above code u can take values from 0 to 99 and u can use men and women here okay  
student.jsx file

```

import { faker } from "@faker-js/faker"

export default function Student(props) {

return(

{props.name}

Programming Experience {props.experience} years </div>

</div>
)


```

```
}
```

Now faker is not needed we can use this online images without installing any packages okay

Now in src folder add images folder and move the two images into that folder and go and do change for coding in main header

```
function MainHeader()

{
 return (
 <div>

 React Course TaskOPedia -
 </div>
);
}
```

with this code I'm not seeing the logo So I'm changing the code like this below

```
import logo from './images/logo192.png'

function MainHeader()

{
 return (
 <div>

 React Course TaskOPedia -
 </div>
);
}
```

```
}
```

The code is working because i am importing and I am importing because the image is in src folder if it is not in src folder i can do above coding okay so I am having two images i am using only one here okay

earlier for CSS also had done the same thing means importing we have done

so after adding some more styles like this

```
import logo from './images/logo192.png'

function MainHeader()

{

 return (

<div className="pt-3 py-1 pl-2" style={{backgroundColor:"black"}}>

React Course TaskOPedia -

</div>

);

}
```

It looks like this below and further i had added

The screenshot shows a web browser window with the URL `localhost:3000`. The page title is "React Course TaskOPedia -". A sub-header says "The course is very exciting". Below it, a message states "In this course we learn React Js by building taskopedia" and "Total Lecture :3". There is a bulleted list: "Basic Foundation" and "Functional and Class Components". A text input field contains "Enter Tasks : Ben". Below it, a section says "Students Enrolled" with three entries:

-  kiran  
Programming Experience 2 years
-  suresh  
Programming Experience 6 years
-  sunitha  
Programming Experience 5 years

At the bottom, a footer bar has the text "This is footer". The system tray at the bottom right shows the date as 04-10-2024.

done is in index.js method i will remove the container for div and will add to row

```
<div >

<Header />

<MainBody />

<div className="container row"> Students Enrolled </div>

<Student experience={2} name="kiran"
headshot="https://randomuser.me/api/portraits/men/10.jpg" />

<Student experience={6} name="suresh"
headshot="https://randomuser.me/api/portraits/women/10.jpg" />

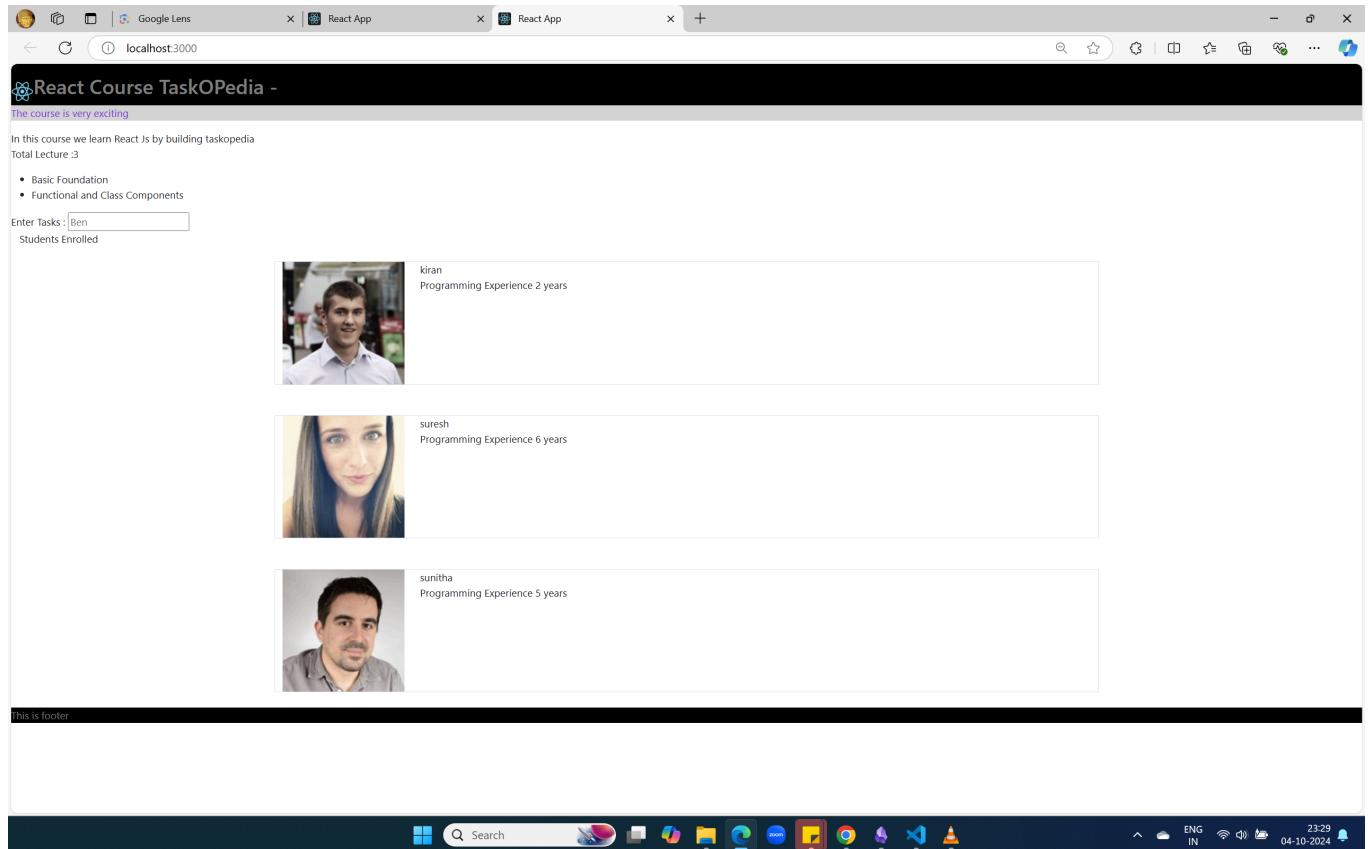
<Student experience={5} name="sunitha"
headshot="https://randomuser.me/api/portraits/men/20.jpg" />

<Footer />

</div>
```

) ;

so it will look like this now



so footer also separated

```
export default function Footer() {

 return (<p style={{ color: "white", backgroundColor: "black" ,marginTop: "10px" , textAlign:"center" }}>Happy Coding!</p>

);

}
```

finally main body also

now some more changes done like this

In students.jsx

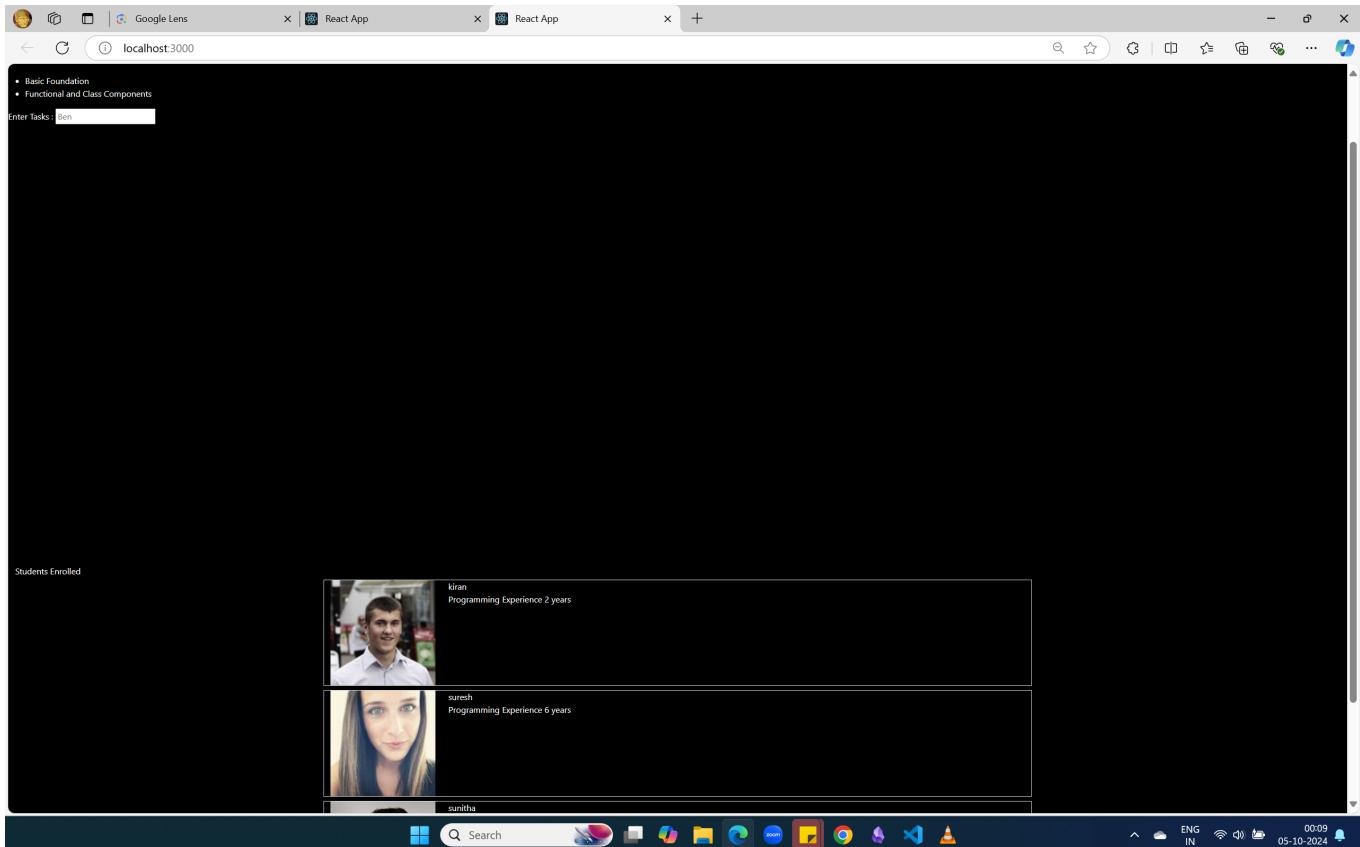
in index.jsx

```
<div style={{ backgroundColor: "black", color:"white"}>
```

in MainBody.jsx

```
<div style={{minHeight:"70vh"}>
```

so after these changes the look is like this



next further changes to be done is

Create one folder layout an in that put header and footer JSX file and then student part should go in the main body

also rename index.js to now .jsx

after doing changes and rectifying the code

so final codes till now

```
export default function Footer() {

 return (<p style={{ color: "white", backgroundColor: "black" ,marginTop:
"10px" , textAlign:"center" }}>Happy Coding!</p>
```

```
);

}

import logo from '../images/logo192.png'

function MainHeader()

{

 return (

<div className="pt-3 py-1 pl-2" style={{backgroundColor:"black"}}>

React Course Task0Pedia -

</div>

);

}

const subHeaderStyle = {

 color: "blueviolet",

 backgroundColor: "lightgray"

}

function SubHeader()

{

 return (<p style={subHeaderStyle} className="text-center"> The course is
very exciting </p>);

}

function Header()
```



```
<div className="px-4">

 <MainBody />

</div>
<Footer />

</div>

);

import Student from './Student';

function MainBody()

{

 const WhatWeLearn = "React Js"

 const lectureCount=3

 return (

 <div style={{minHeight:"70vh"}}>

 <p>In this course we learn {WhatWeLearn} by building taskopedia

 Total Lecture :{lectureCount}

 </p>

 Basic Foundation

 Functional and Class Components

 <div className="container row"> Students Enrolled </div>

 <Student experience={2} name="kiran"
headshot="https://randomuser.me/api/portraits/men/10.jpg" />


```

```
<Student experience={6} name="suresh"
headshot="https://randomuser.me/api/portraits/women/10.jpg" />

<Student experience={5} name="sunitha"
headshot="https://randomuser.me/api/portraits/men/20.jpg" />

</div>

)

}

export default MainBody
```

```
import { faker } from "@faker-js/faker"

export default function Student(props) {

return(

{props.name}

Programming Experience {props.experience} years </div>

</div>
</div>


```

)

}

so finally will look like this

The screenshot shows a web browser window with three tabs open, all titled "React App". The active tab displays a course page for "React Course TaskOPedia -". The page has a dark header with the title and a sub-header "The course is very exciting". Below this, there's a section about the course: "In this course we learn React Js by building taskopedia" and "Total Lecture : 3". A bulleted list includes "Basic Foundation" and "Functional and Class Components". A "Students Enrolled" section follows, featuring three student profiles in a grid:

Students Enrolled
 kiran Programming Experience 2 years
 suresh Programming Experience 6 years
 sunitha Programming Experience 5 years

At the bottom of the page is a "Happy Coding!" message. The browser's taskbar at the bottom shows various pinned icons and the date/time "05-10-2024 00:49".

further in student i changed like this

```
return(
```

then it looks like this

The screenshot shows a web browser window with multiple tabs open, all titled "React App". The active tab displays a dark-themed React application. At the top, there's a header with the title "React Course TaskOPedia -" and a sub-header "The course is very exciting". Below this, a message states "In this course we learn React Js by building taskopedia" and "Total Lecture :3". A bulleted list follows: • Basic Foundation • Functional and Class Components. A section titled "Students Enrolled" lists three students with their profile pictures, names, and programming experience: kiran (Programming Experience 2 years), suresh (Programming Experience 6 years), and sunitha (Programming Experience 5 years). At the bottom right of the app area, the text "Happy Coding!" is visible. The browser's address bar shows "localhost:3000". The system tray at the bottom right indicates the date as 05-10-2024 and the time as 01:06.

in index .html body background color make it black

This screenshot shows the same React application as the first one, but with a black background for the entire body. The UI elements remain the same: header, course details, student enrollment list, and the "Happy Coding!" message. The browser and system tray are also identical to the first screenshot.

Now i want to add child components so from <https://icons.getbootstrap.com/>

copy this CDN url into index.html file

```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-icons.min.css">
```

Add new component now

### StudentReview.jsx

```
export default function StudentReview()

{
 return (
 <div className="p-2">
 <i class="bi bi-hand-thumbs-up-fill text-success"></i>
 <i class="bi bi-hand-thumbs-down-fill text-danger"></i>
 </div>
);
}
```

The above component I want to use it in student component so if i do that then it will be available for all the students so for some specific students only i want to make it available say first two students okay

```
import Student from './Student';
import StudentReview from './StudentReview';

function MainBody()
{
 const WhatWeLearn = "React Js"
 const lectureCount=3
 return (
 <div style={{minHeight:"70vh"}}>
 <p>In this course we learn {WhatWeLearn} by building taskopedia

 Total Lecture :{lectureCount}
 </p>

```

```

 Basic Foundation
 Functional and Class Components

 <div className="container row"> Students Enrolled </div>
 <Student experience={2} name="kiran"
headshot="https://randomuser.me/api/portraits/men/10.jpg">
 <StudentReview />
 </Student>
 <Student experience={6} name="suresh"
headshot="https://randomuser.me/api/portraits/women/10.jpg">
 <StudentReview />
 </Student>
<Student experience={5} name="sunitha"
headshot="https://randomuser.me/api/portraits/men/20.jpg" />
</div>
)
}

export default MainBody

```

so even though added import and for first two added the elements i am not seeing any change in design .

so something needs to be added in student component

```

import { faker } from "@faker-js/faker"

export default function Student(props) {

return(
<div className="col-4 p-1">

 <div className="row border">
 <div className="col-2">

 </div>
 <div className="col-10">
 {props.name}

 Programming Experience {props.experience} years
 </div>
 <div className="col-2">
 {props.children}
 </div>
 </div>
)
}

export default MainBody

```

```
 </div>
 </div>
</div>
)
}
```

so you can see here i am using props. Children which will give me the output

The screenshot shows a web browser window titled "React App" at "localhost:3000". The page has a dark theme with a header featuring the React logo and the text "React Course TaskOPedia -". Below the header, there is a message "The course is very exciting". The main content area displays three student profiles in cards:

- kiran**  
Programming Experience 2 years  
(&nbsp;)
- suresh**  
Programming Experience 6 years  
(&nbsp;)
- sunitha**  
Programming Experience 5 years

At the bottom of the page, the text "Happy Coding!" is centered. The browser's address bar shows "localhost:3000" and the taskbar at the bottom includes icons for various Windows applications like File Explorer, Edge, and Task View.

Now create a folder Components in src folder and in that student folder and add student and student review components

	EXPLORER	...	inde
25	OPEN EDITORS		src > C
	index.html public M		1
	StudentReview.jsx...	U	2
	Header.jsx src\Layo...	U	3
	index.jsx src U		4
	Student.jsx src\Co... U		5
	MainBody.jsx src U		6
	Footer.jsx src\Layout U		7
	TASKOPEDIA	▷ E+ ⌂ ⌂	8
	node_modules		9
	public	●	10
	index.html M		11
	src	●	12
	Components \ Student	●	13
	Student.jsx U		14
	StudentReview.jsx U		15
	CSS	●	16
	images	●	17
	Layout	●	18
	Footer.jsx U		19
	Header.jsx U		20
	index.jsx U		21
	MainBody.jsx U		22
	.gitignore		23



so still design is same no effect okay now

Now Convert all functions which u have created into arrow function okay

```
import { faker } from "@faker-js/faker"

const Student=(props)=> {

 return (
 <div className="col-4 p-1">

 <div className="row border">

 <div className="col-2">

 </div>

 <div className="col-10">
 {props.name}

 Programming Experience {props.experience} years
 </div>

 <div className="col-2">
 {props.children}
 </div>
 </div>
 </div>
)
}
```

```
</div>

</div>

</div>

);

}

Now converting into arrow function
```

```
export default Student;
```

```
const Student=(props)=> {

 return (
 <div className="col-4 p-1">

 <div className="row border">

 <div className="col-2">

 </div>

 <div className="col-10">

 {props.name}

 Programming Experience {props.experience} years

 </div>

 <div className="col-2">

 {props.children}

 </div>

 </div>

```

```
</div>

</div>

);

}

export default Student;

const StudentReview=()=>{

{
 return (
 <div className="p-2">

 <i class="bi bi-hand-thumbs-up-fill text-success p-1"></i>

 <i class="bi bi-hand-thumbs-down-fill text-danger p-1"></i>

 </div>
);
}

export default StudentReview

const Footer=()=>{

{
 return (<p style={{ color: "white", backgroundColor: "black" ,marginTop: "10px", textAlign:"center" }}>Happy Coding!</p>
);
}
}
```

```
export default Footer

import logo from '../images/logo192.png'

function MainHeader()

{

 return (

<div className="pt-3 py-1 pl-2" style={{backgroundColor:"black"}}>

React Course Task0Pedia -

</div>

);

}

const subHeaderStyle = {

 color: "blueviolet",

 backgroundColor: "lightgray"

}

function SubHeader()

{

 return (<p style={subHeaderStyle} className="text-center"> The course is
very exciting </p>);

}

const Header=()=>

{
```

```

return(
 <div>
 <MainHeader></MainHeader>
 <SubHeader></SubHeader>
 </div>
)
}

export default Header

```

so upto now after converting js functions into arrow functions it is running fine

Now let us go through Class Component and Function Component

## Class vs Functional Components

### Class Components

- Less common
- render() method is required, which returns an HTML element
- Stateful components
- Class components have lifecycle methods

### Functional Components

- More common
- render() method is not present, it directly returns HTML element or nothing.
- Stateless components
- Class components do have lifecycle methods
- Hooks!

Next I want to convert the function components into class based components

```

import React from "react";
class StudentReview extends React.Component {
 render() {
 return (
 <div className="p-2">
 <i
 class="bi bi-hand-thumbs-up-fill text-success p-1" style={{ cursor: "pointer" }}></i>
 <i
 class="bi bi-hand-thumbs-down-fill text-danger p-1"
 style={{ cursor: "pointer" }}></i>
 </div>
);
 }
}

export default StudentReview;

```

so above one function based component i had converted into class based can check it okay now i am changing the student component which is having props in this manner here u can use this

```

import React from "react";

class Student extends React.Component{

 render() {

 return (
 <div className="col-4 p-1">

 <div className="row border">
 <div className="col-2">

 </div>
 <div className="col-10">
 {this.props.name}

 Programming Experience {this.props.experience} years
 </div>
 <div className="col-2">
 {this.props.children}
 </div>
 </div>
 </div>
);
 }
}

export default Student;

```

```

 </div>
);
}

export default Student;

```

Now let us convert MainBody into class based component

```

import Student from './Components/Student/Student';
import StudentReview from './Components/Student/StudentReview';
import React from 'react';
class MainBody extends React.Component
{

 render()
 {
 const WhatWeLearn = "React Js"
 const lectureCount=3
 return (
 <div style={{minHeight:"70vh"}}>
 <p>In this course we learn {WhatWeLearn} by building taskopedia

 Total Lecture :{lectureCount}
 </p>

 Basic Foundation
 Functional and Class Components

 <div className="container row">&nbsp&nbsp Students Enrolled </div>
 <Student experience={2} name="kiran"
headshot="https://randomuser.me/api/portraits/men/10.jpg">
 <StudentReview />
 </Student>
 <Student experience={6} name="suresh"
headshot="https://randomuser.me/api/portraits/women/10.jpg">
 <StudentReview />
 </Student>
 <Student experience={5} name="sunitha"
headshot="https://randomuser.me/api/portraits/men/20.jpg" />
 </div>
)
 }
}

```

```
export default MainBody
```

so all the information above project is in this URL : [GitHub - bhrugen/react\\_taskopedia](#)

Now i want to develop a new project

so being in D:\React\_Bruhugen\ReactFundamentals\ReactappPractise

and for the new app the URL is : [GitHub - bhrugen/react\\_countopedia](#)

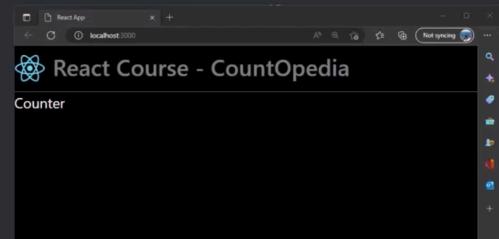
write the command now to develop this app

**npx create-react-app countopedia**

after creating the app ..

## Assignment 7 - CountOPedia Setup

1. Remove all files other than index.html, index.js and react Image
2. Move react image to src/images folder. (Create images folder)
3. Clean index.js and index.html (Remove anything that is not used)
4. Add Bootstrap js/css CDN inside index.html
5. Create 2 components (Design to look as shown in image)
  - Header.jsx (Functional Component)
  - Counter.jsx (Class Component)
6. Copy the images provided in course snippet section 4 to the images folder



so the above tasks which u have to do now just as we have done earlier okay

The screenshot shows the VS Code interface. The left sidebar displays the file structure of the 'countopedia' project. The 'src' folder contains 'index.js'. The 'public' folder contains 'index.html'. Other files visible include '.gitignore', 'package-lock.json', 'package.json', and 'README.md'. The right panel shows the code editor with 'index.js' open. The code imports React and ReactDOM, creates a root element, and renders a component. The code editor has syntax highlighting and a status bar at the top.

```
File Edit Selection View Go Run Terminal Help ← → ⌘ countopedia
EXPLORER OPEN EDITORS index.js M ...
src > index.js > ...
1 import React from 'react';
2 import ReactDOM from 'react-dom/client';
3
4 const root = ReactDOM.createRoot(document.getElementById('root'));
5 root.render(
6
7);
8
```

now after doing changes the look is like this

and now go to index.html file and add these files so from the previous project i am adding it like this

```
<!DOCTYPE html>
<html lang="en">
 <head>
 <meta charset="utf-8" />
 <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
 <meta name="viewport" content="width=device-width, initial-scale=1" />
 <meta name="theme-color" content="#000000" />
 <meta
 name="description"
 content="Web site created using create-react-app"
 />

 <link
 href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
 rel="stylesheet"
 integrity="sha384-QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH"
 crossorigin="anonymous"
 >
 <script
 src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.8/dist/umd/popper.min.js"
 >
 <script
 integrity="sha384-I7E8VVD/ismYTF4hNIPjVp/Zjvgyol6VFvRkX/vR+Vc4jQkC+hVqc2pM80Dewa9r"
 crossorigin="anonymous"
 ></script>
 <script
 src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.min.js"
 >
```

```
 integrity="sha384-
0pUGZvbkm6XF6gxjEnlmuGrJXVbNuzT9qBBavbLwCs0GabYfZo0T0to5eqruptLy"
crossorigin="anonymous">></script>

<title>React App</title>
</head>
<body style="background-color: black;">
<noscript>You need to enable JavaScript to run this app.</noscript>
<div id="root"></div>

</body>
</html>
```

Now go to index.js file

and now first add two components Header.jsx and Counter.jsx files in the src folder

Header.jsx code

---

```
import logo from "./images/logo512.png";

function Header() {
 return (
 <div className="py-2 pl-2" style={{ borderBottom: "1px solid #777" }}>

 CountOpedia
 </div>
);
}

export default Header;
```

Counter.jsx code

---

```
import React from "react";

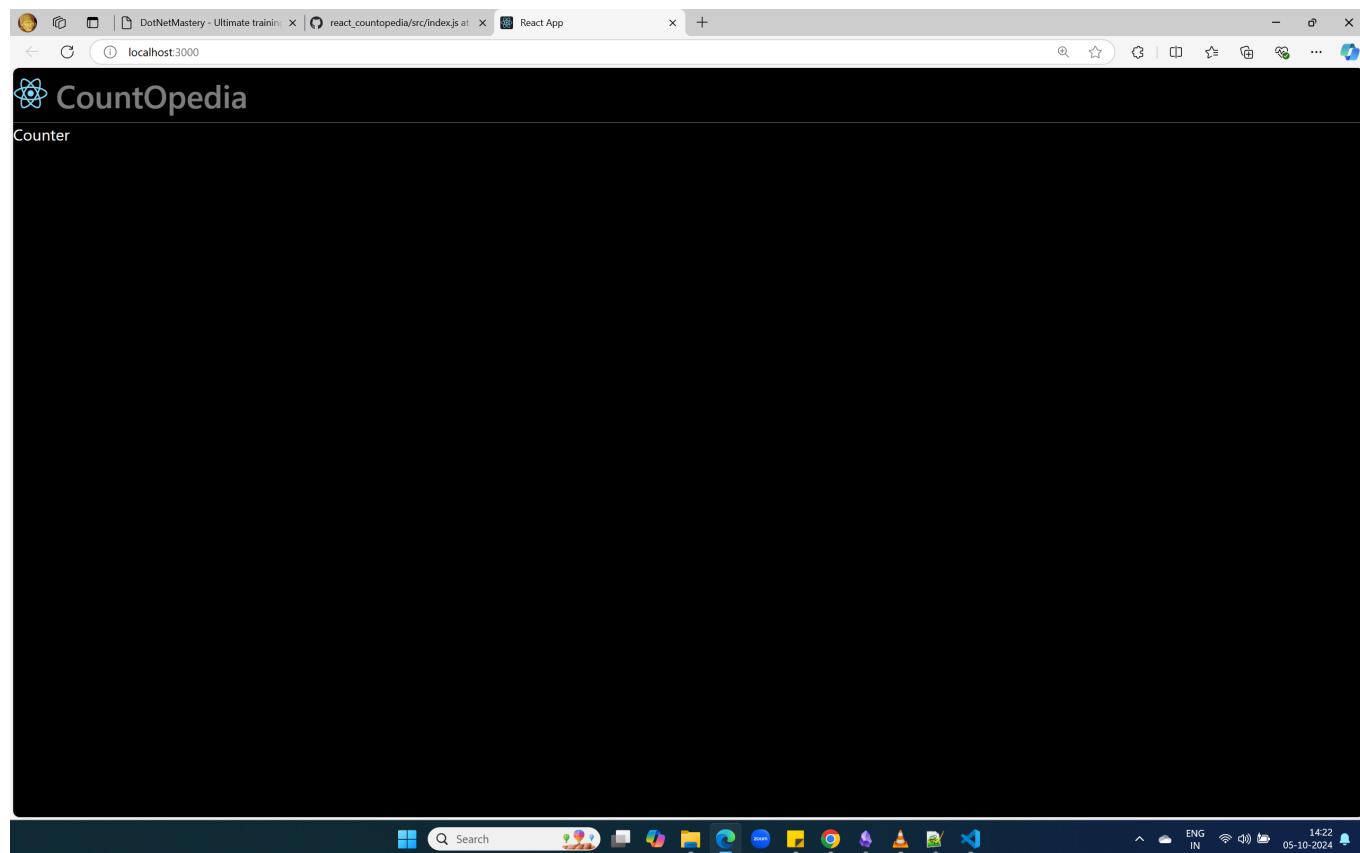
export default class Counter extends React.Component {
 render() {
```

```
 return <div className="text-white">Counter</div>;
 }
}
```

index.js file code

```
import React from "react";
import ReactDOM from "react-dom/client";
import Counter from "./Counter";
import Header from "./Header";

const root = ReactDOM.createRoot(document.getElementById("root"));
root.render(
 <div>
 <Header></Header>
 <Counter></Counter>
 </div>
);
```



Till now it looks like this and now add two images attack and defend in images folder

Now doing changes in counter.jsx it looks like this

```
import React from "react";

export default class Counter extends React.Component {
 render() {
 return (
 <div className="row text-white">
 <h1>Counter: </h1>
 <button style={{ width: "200px" }}>+1</button>
 <button style={{ width: "200px" }}>-1</button>
 </div>
)
 }
}
```

and the Header.jsx

```
import logo from "./images/logo512.png";

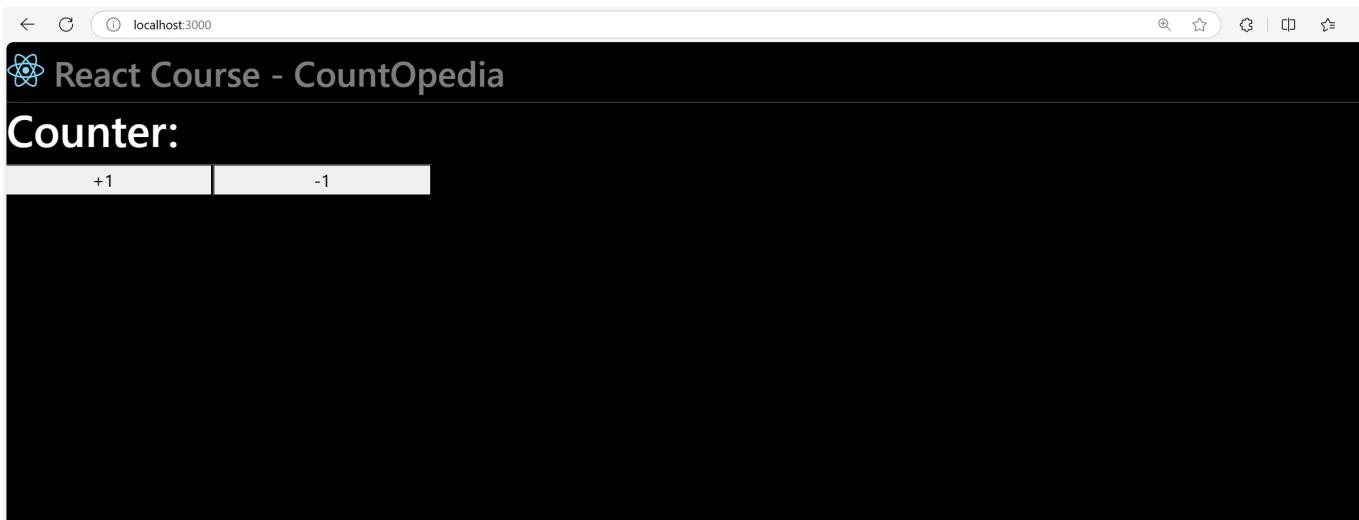
function Header() {
 return (
 <div className="py-2 pl-2" style={{ borderBottom: "1px solid #777" }}>

 React Course - Count0pedia

 </div>
);
}

export default Header;
```

and right now the output is



now again updated code of Counter.jsx

```
import React from "react";

export default class Counter extends React.Component {
 handleAttack() {
 alert("Attack clicked");
 }

 handleDefence() {
 alert("Defend clicked");
 }

 render() {
 return (
 <div className="row text-white">
 <h1>Counter: </h1>
 <button onClick={this.handleAttack} style={{ width: "200px" }}>
 +1
 </button>
 <button onClick={this.handleDefence} style={{ width: "200px" }}>
 -1
 </button>
 </div>
);
 }
}
```

so above click event is defined and methods defined in class only for that

when we are using class component then only we can have states and to the super constructor i have to pass props and here we can define state and there counter what value u give that will be displayed

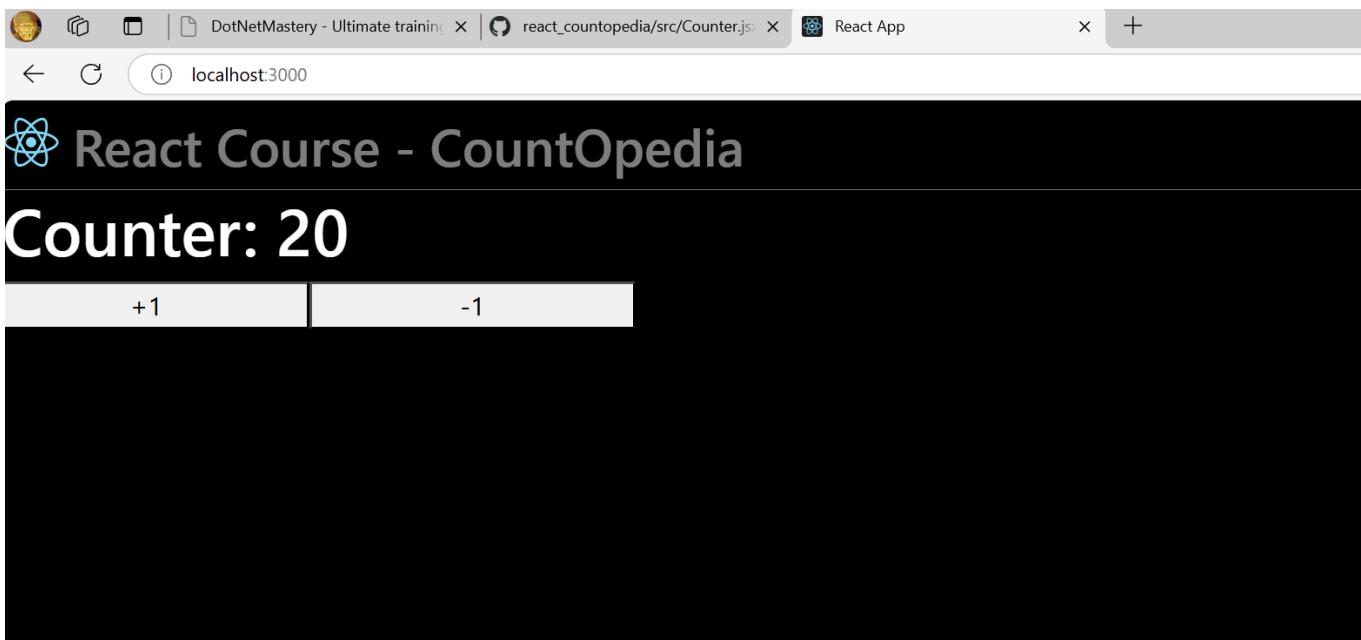
```
import React from "react";

export default class Counter extends React.Component {
 constructor(props) {
 super(props);
 this.state = {
 count: 20,
 };
 }

 handleAttack() {
 alert("Attack clicked");
 }

 handleDefence() {
 alert("Defend clicked");
 }

 render() {
 return (
 <div className="row text-white">
 <h1>Counter: {this.state.count} </h1>
 <button onClick={this.handleAttack} style={{ width: "200px" }}>
 +1
 </button>
 <button onClick={this.handleDefence} style={{ width: "200px" }}>
 -1
 </button>
 </div>
);
 }
}
```



here state is immutable i cannot directly modify the state okay

so after putting 1

```
handleAttack() {

 alert("Attack clicked");

 this.state.count=1

}
```

and clicking the button it will give error okay as what i have told is correct i cannot modify the count status okay u will get error this was for checking the above point

so what i will do now is that we can change the state using setState method but u need to bind on the top the method and then click you can see the effect okay

```
import React from "react";

export default class Counter extends React.Component {
 constructor(props) {
 super(props);
 this.handleAttack = this.handleAttack.bind(this);
 this.state = {
 count: 20,
 };
 }

 handleAttack() {
 alert("Attack clicked");
 this.setState({ count: this.state.count + 1 });
 }
}
```

```

 // this.state.count=1
 this.setState({count:2})
}

handleDefence() {
 alert("Defend clicked");
}

render() {
 return (
 <div className="row text-white">
 <h1>Counter: {this.state.count} </h1>
 <button onClick={this.handleAttack} style={{ width: "200px" }}>
 +1
 </button>
 <button onClick={this.handleDefence} style={{ width: "200px" }}>
 -1
 </button>
 </div>
);
}
}

```

now u do click the button the value will change from 20 to 2

so state cannot be modified like that directly okay so now instead of setting 2 i will increment and decrement on button clicks so code will change like this and also register both the events with bind

```

import React from "react";

export default class Counter extends React.Component {
 constructor(props) {
 super(props);
 this.handleAttack = this.handleAttack.bind(this);
 this.handleDefence= this.handleDefence.bind(this);
 this.state = {
 count: 20,
 };
 }

 handleAttack() {
 // alert("Attack clicked");
 // this.state.count=2
 this.setState({ count: this.state.count + 1 });
 }
}

```

```

}

handleDefence() {
 // alert("Defend clicked");
 this.setState({ count: this.state.count - 1 });
}

render() {
 return (
 <div className="row text-white">
 <h1>Counter: {this.state.count} </h1>
 <button onClick={this.handleAttack} style={{ width: "200px" }}>
 +1
 </button>
 <button onClick={this.handleDefence} style={{ width: "200px" }}>
 -1
 </button>
 </div>
);
}
}

```

so it will increment from 21 and 22 and decrement from 19 etc. like that okay  
 now if want to do like this

```

handleAttack() {

 // alert("Attack clicked");

 // this.state.count=2

 this.setState({ count: this.state.count + 10 });

 this.setState({ count: this.state.count + 100 });

}

```

means it will take 100 so from zero it will take zero and set 100 same thing if i do reverse  
 it will take 10 so how do do this to print both simultaneously that is the task okay so want 110 as  
 output and now in below code i will use arrow syntax for this

```

import React from "react";

export default class Counter extends React.Component {

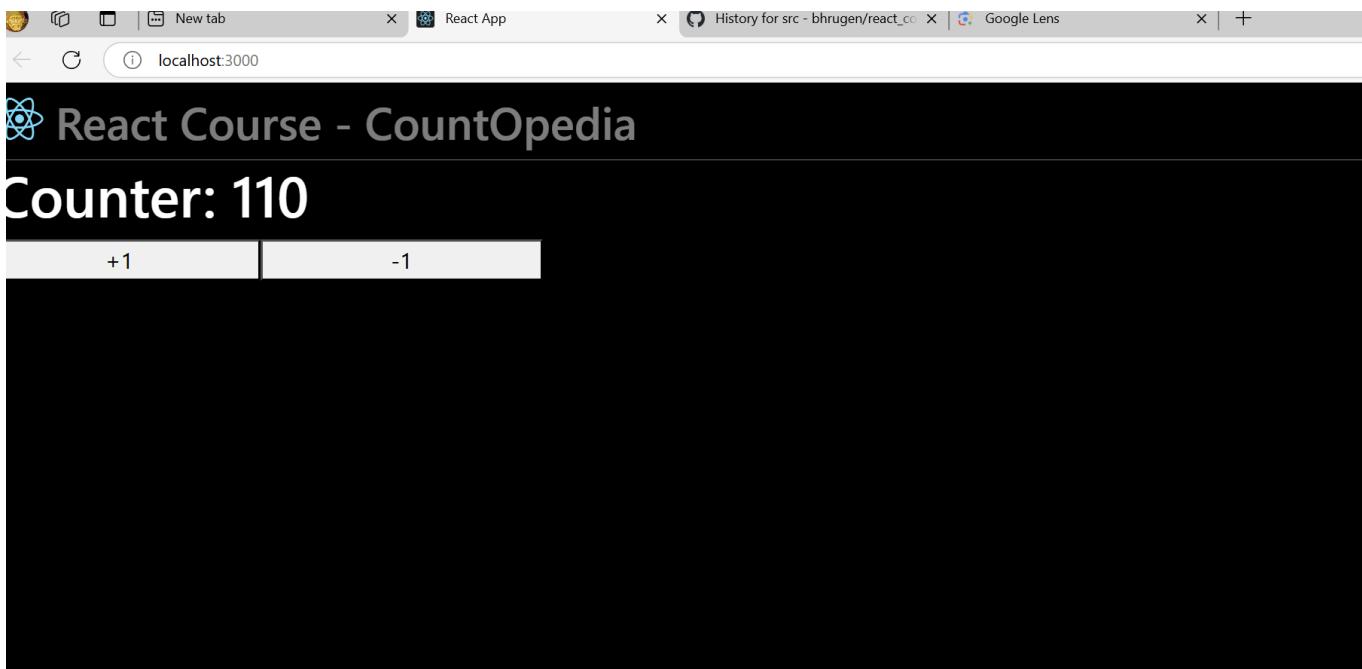
```

```
constructor(props) {
 super(props);
 this.handleAttack = this.handleAttack.bind(this);
 this.handleDefence= this.handleDefence.bind(this);
 this.state = {
 count:0,
 };
}

handleAttack() {
 this.setState((previousState) => {
 return {
 count: previousState.count + 100,
 };
 });
 this.setState((previousState) =>{
 return {
 count: previousState.count + 10,
 };
 });
}

handleDefence() {
 // alert("Defend clicked");
 this.setState({ count: this.state.count - 1 });
}

render() {
 return (
 <div className="row text-white">
 <h1>Counter: {this.state.count} </h1>
 <button onClick={this.handleAttack} style={{ width: "200px" }}>
 +1
 </button>
 <button onClick={this.handleDefence} style={{ width: "200px" }}>
 -1
 </button>
 </div>
);
}
}
```



further changing the functionality we will get like this means same code with arrow

```
import React from "react";

export default class Counter extends React.Component {
 constructor(props) {
 super(props);
 this.handleAttack = this.handleAttack.bind(this);
 this.handleDefence= this.handleDefence.bind(this);
 this.state = {
 count:0,
 };
 }

 handleAttack() {
 this.setState((previousState) => {
 return {
 count: previousState.count + 1,
 };
 });
 }

 handleDefence() {
 this.setState((previousState) => {
 return {
 count: previousState.count - 1,
 };
 });
 }
}
```

```

 });
}

render() {
 return (
 <div className="row text-white">
 <h1>Counter: {this.state.count} </h1>
 <button onClick={this.handleAttack} style={{ width: "200px" }}>
 +1
 </button>
 <button onClick={this.handleDefence} style={{ width: "200px" }}>
 -1
 </button>
 </div>
);
}
}

```

Now further i had added some images and then implemented the logic for random play and reset

```

import React from "react";
import attack from "./images/attack.png";
import defend from "./images/defend.png";

export default class Counter extends React.Component {
 constructor(props) {
 super(props);
 this.handleAttack = this.handleAttack.bind(this);
 this.handleDefence = this.handleDefence.bind(this);
 this.state = {
 count: 0,
 };
 }

 handleAttack = () => {
 //alert("Attack clicked");
 this.setState((previousState) => {
 let newCount = previousState.count + Math.round(Math.random() * 10);
 return {
 count: newCount,
 };
 });
 //this.state.count = 2;
 };
}

```

```
handleDefence = () => {
 //alert("Defend clicked");
 this.setState((previousState) => {
 let newCount = previousState.count - Math.round(Math.random() * 10);
 return {
 count: newCount,
 };
 });
};

handleRandomPlay = () => {
 let playMode = Math.round(Math.random());
 if (playMode == 0) {
 this.handleAttack();
 } else {
 this.handleDefence();
 }
};

handleReset = () => {
 this.setState(() => {
 return {
 count: 0,
 };
 });
};

render() {
 return (
 <div className="row text-white text-center">
 <h1>Game Score: {this.state.count} </h1>
 <p>You win at +10 points and lose at -10 points!</p>
 <p>Last Play: </p>
 <h3>Game Status : </h3>
 <div className="col-6 col-md-3 offset-md-3">
 <img
 style={{
 width: "100%",
 cursor: "pointer",
 border: "1px solid green",
 }}
 className="p-4 rounded"
 src={attack}
 onClick={this.handleAttack}
 />
 </div>
 <div className="col-6 col-md-3 offset-md-3">

```

```

 <img
 style={{
 width: "100%",
 cursor: "pointer",
 border: "1px solid red",
 }}
 className="p-4 rounded"
 src={defend}
 onClick={this.handleDefence}
 />
 </div>
 <div className="col-12 col-md-4 offset-md-4">
 <button
 className="btn btn-secondary w-100 mt-2"
 onClick={this.handleRandomPlay}
 >
 Random Play
 </button>

 <button
 className="btn btn-warning w-100 mt-2"
 onClick={this.handleReset}
 >
 Reset
 </button>
 </div>
</div>
);
}
}

```

Logic for game status updated in code

```

import React from "react";
import attack from "./images/attack.png";
import defend from "./images/defend.png";

export default class Counter extends React.Component {
 constructor(props) {
 super(props);
 this.handleAttack = this.handleAttack.bind(this);
 this.handleDefence = this.handleDefence.bind(this);
 this.state = {
 count: 0,
 gameStatus: "",
 }
 }
 handleAttack() {
 this.setState({
 count: this.state.count + 1
 })
 }
 handleDefence() {
 this.setState({
 count: this.state.count - 1
 })
 }
 handleReset() {
 this.setState({
 count: 0
 })
 }
 handleRandomPlay() {
 const random = Math.floor(Math.random() * 2);
 if (random === 0) {
 this.handleAttack();
 } else {
 this.handleDefence();
 }
 }
 render() {
 return (
 <div>
 <img
 style={{
 width: "100%",
```

```
 lastPlay: "",
 };
}

handleAttack = () => {
 //alert("Attack clicked");
 this.setState((previousState) => {
 let newCount = previousState.count + Math.round(Math.random() * 10);
 return {
 count: newCount,
 lastPlay: "Attack",
 gameStatus: newCount > 10 ? "You Won!!" : previousState.gameStatus,
 };
 });
 //this.state.count = 2;
};

handleDefence = () => {
 //alert("Defend clicked");
 this.setState((previousState) => {
 let newCount = previousState.count - Math.round(Math.random() * 10);
 return {
 count: newCount,
 lastPlay: "Defence",
 gameStatus: newCount < -10 ? "You Lost!!" : previousState.gameStatus,
 };
 });
};

handleRandomPlay = () => {
 let playMode = Math.round(Math.random());
 if (playMode == 0) {
 this.handleAttack();
 } else {
 this.handleDefence();
 }
};
handleReset = () => {
 this.setState(() => {
 return {
 count: 0,
 gameStatus: "",
 lastPlay: "",
 };
 });
};
```

```
render() {
 return (
 <div className="row text-white text-center">
 <h1>Game Score: {this.state.count} </h1>
 <p>You win at +10 points and lose at -10 points!</p>
 <p>Last Play:{this.state.lastPlay} </p>
 <h3>Game Status : {this.state.gameStatus}</h3>
 <div className="col-6 col-md-3 offset-md-3">
 <img
 style={{
 width: "100%",
 cursor: "pointer",
 border: "1px solid green",
 }}
 className="p-4 rounded"
 src={attack}
 onClick={this.handleAttack}
 />
 </div>
 <div className="col-6 col-md-3 ">
 <img
 style={{
 width: "100%",
 cursor: "pointer",
 border: "1px solid red",
 }}
 className="p-4 rounded"
 src={defend}
 onClick={this.handleDefence}
 />
 </div>
 <div className="col-12 col-md-4 offset-md-4">
 <button
 className="btn btn-secondary w-100 mt-2"
 onClick={this.handleRandomPlay}
 >
 Random Play
 </button>

 <button
 className="btn btn-warning w-100 mt-2"
 onClick={this.handleReset}
 >
 Reset
 </button>
 </div>

```

```

 </div>
);
}
}

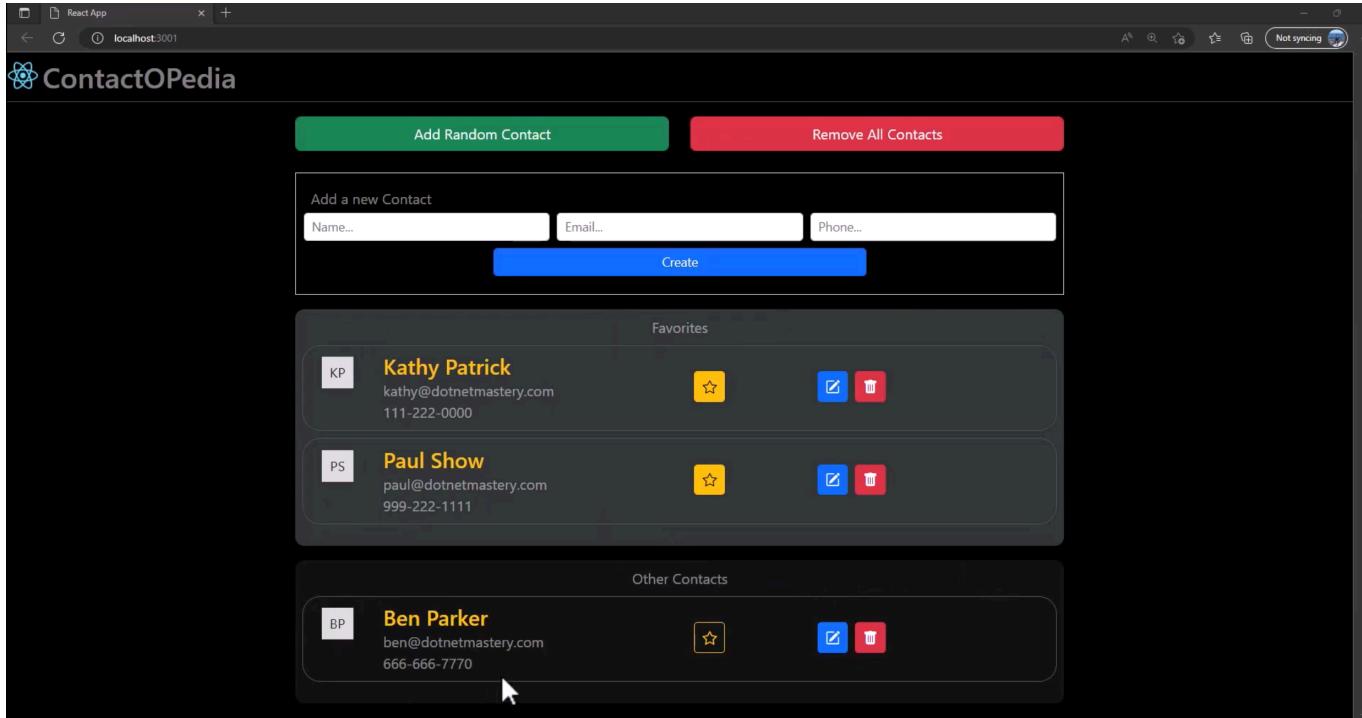
```

This ends the 2nd project of Countopedia and now move on to next project

I had created a new application and now copy Header, images and index.js from countopedia to contactopedia one okay and Even replace index html file in public folder okay

as usual remove everything from public folder and src folder okay I am getting a good output .

we have different components



add random contact ,remove all contacts ,favorites etc. and all so let us now go with a demo now

and now add Footer.jsx in layout folder so the coding is like this

```

const Footer = ()=> {
 return (
 <div
 style={{
 color: "gray",
 marginTop: "10px",
 borderTop: "1px solid #555",
 textAlign: "center",
 }}>

```

```

>
Happy Coding! @ContactOPedia
</div>
);
};

export default Footer;

```

The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows the project structure under "CONTACTOPEDIA". The "src" folder contains "Components" and "Layout". "Components" contains "ContactPages" (with files like AddContact.jsx, AddRandomComponent.jsx, Contact.jsx, ContactIndex.jsx, FavouriteContact.jsx, GeneralContact.jsx, RemoveAllContacts.jsx) and "Layout" (with files like Footer.jsx and Header.jsx). "Layout" also contains "images" with files logo192.png and logo512.png.
- Code Editor:** The file "Footer.jsx" is open. The code is as follows:

```

const Footer = () => {
 return (
 <div
 style={{
 color: "gray",
 marginTop: "10px",
 borderTop: "1px solid #555",
 textAlign: "center",
 }}>
 Happy Coding! @ContactOPedia
 </div>
);
};

export default Footer;

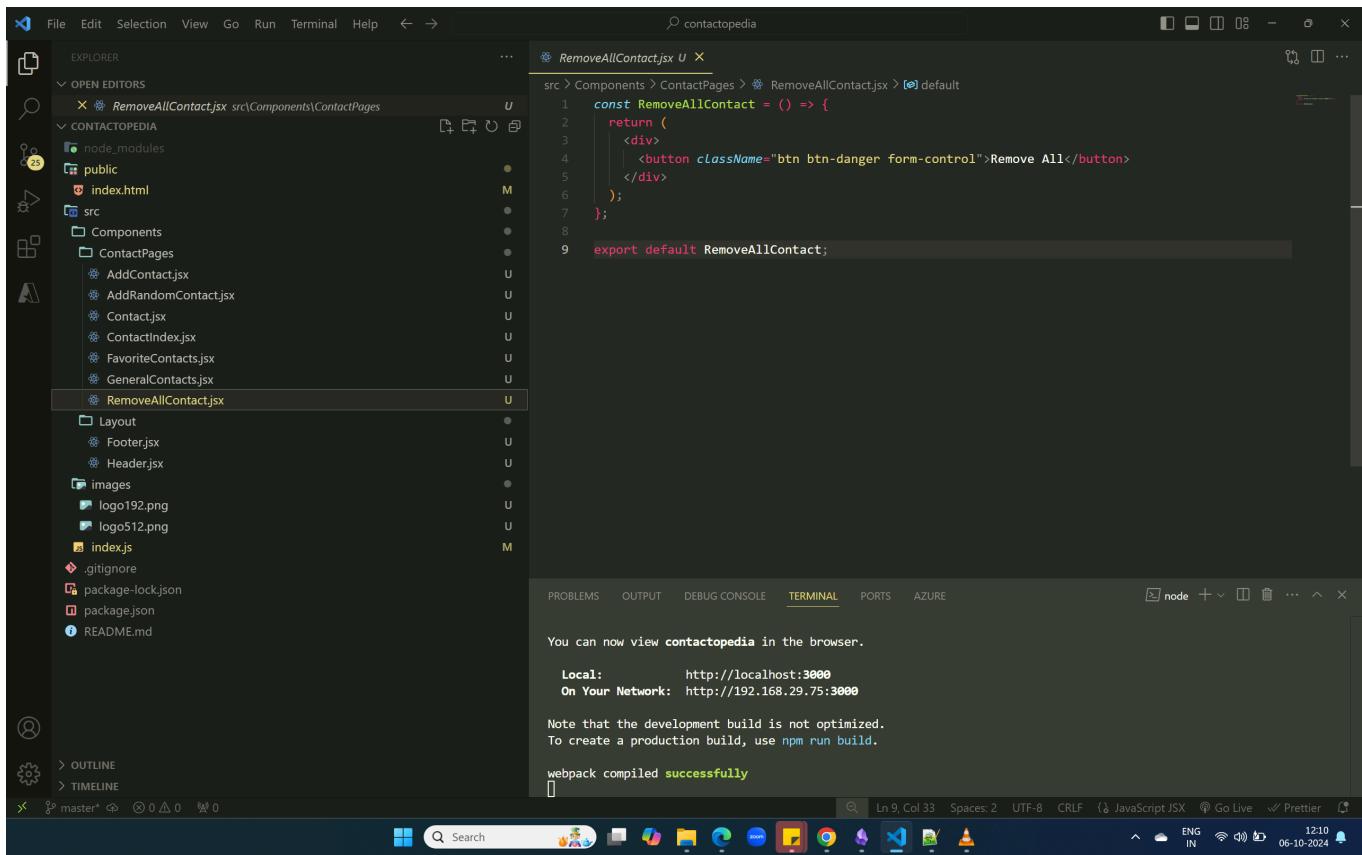
```
- Terminal:** Shows a warning message from Webpack: "Module not found: Error: You attempted to import ../../images/logo512.png which falls outside of the project src/ directory. Relative imports outside of src/ are not supported." It also says "webpack compiled with 1 error".
- Status Bar:** Shows the current file is "Footer.jsx", line 15, column 23, with 4 spaces, using UTF-8 encoding, CRLF line endings, and the file is a JavaScript JSX file.

so you can see the structure of my project okay

```
import logo from "../../images/logo512.png";
```

so i had changed the path in header then the error which was showing in image is gone .So now in all of the JSX files I will create the functional components which you can see in the diagram

Now add some basic codes into all components now



Now as per screen latest let us write all codes now one by one

### RemoveAllContact

```

const RemoveAllContact = () => {
 return (
 <div>
 <button className="btn btn-danger form-control">Remove All</button>
 </div>
);
};

export default RemoveAllContact;
```

### GeneralContacts

```

const GeneralContacts = () => {
 return (
 <div>
 <button className="btn btn-secondary form-control">
 General Contact
 </button>
 </div>
);
};
```

```
export default GeneralContacts;

FavoriteContacts

const FavoriteContacts = () => {
 return (
 <div>
 <button className="btn btn-secondary form-control">
 Favorite Contacts
 </button>
 </div>
);
};

export default FavoriteContacts;

Contact

const Contact = () => {
 return (
 <div>
 <button className="btn btn-secondary form-control">Contact</button>
 </div>
);
};

export default Contact;

AddRandomContact

const AddRandomContact = () => {
 return (
 <div>
 <button className="btn btn-danger form-control">
 Add Random Contact
 </button>
 </div>
);
};

export default AddRandomContact;

AddContact

const AddContact = () => {
```

```
 return (
 <div>
 <button className="btn btn-secondary form-control">Add Contact</button>
 </div>
);
}

export default AddContact;

ContactIndex

import React from "react";
import Footer from "../Layout/Footer";
import Header from "../Layout/Header";
import AddContact from "./AddContact";
import AddRandomContact from "./AddRandomContact";
import FavoriteContacts from "./FavoriteContacts";
import GeneralContacts from "./GeneralContacts";
import RemoveAllContact from "./RemoveAllContact";

class ContactIndex extends React.Component {
 render() {
 return (
 <div>
 <Header />
 <div className="container" style={{ minHeight: "85vh" }}>
 <div className="row py-3">
 <div className="col-4 offset-2">
 <AddRandomContact />
 </div>
 <div className="col-4">
 <RemoveAllContact />
 </div>
 <div className="row py-2">
 <AddContact />
 </div>
 <div className="row py-2">
 <FavoriteContacts />
 </div>
 <div className="row py-2">
 <GeneralContacts />
 </div>
 </div>
 </div>
 <Footer />
 </div>
);
 }
}
```

```

);
}

}

export default ContactIndex;

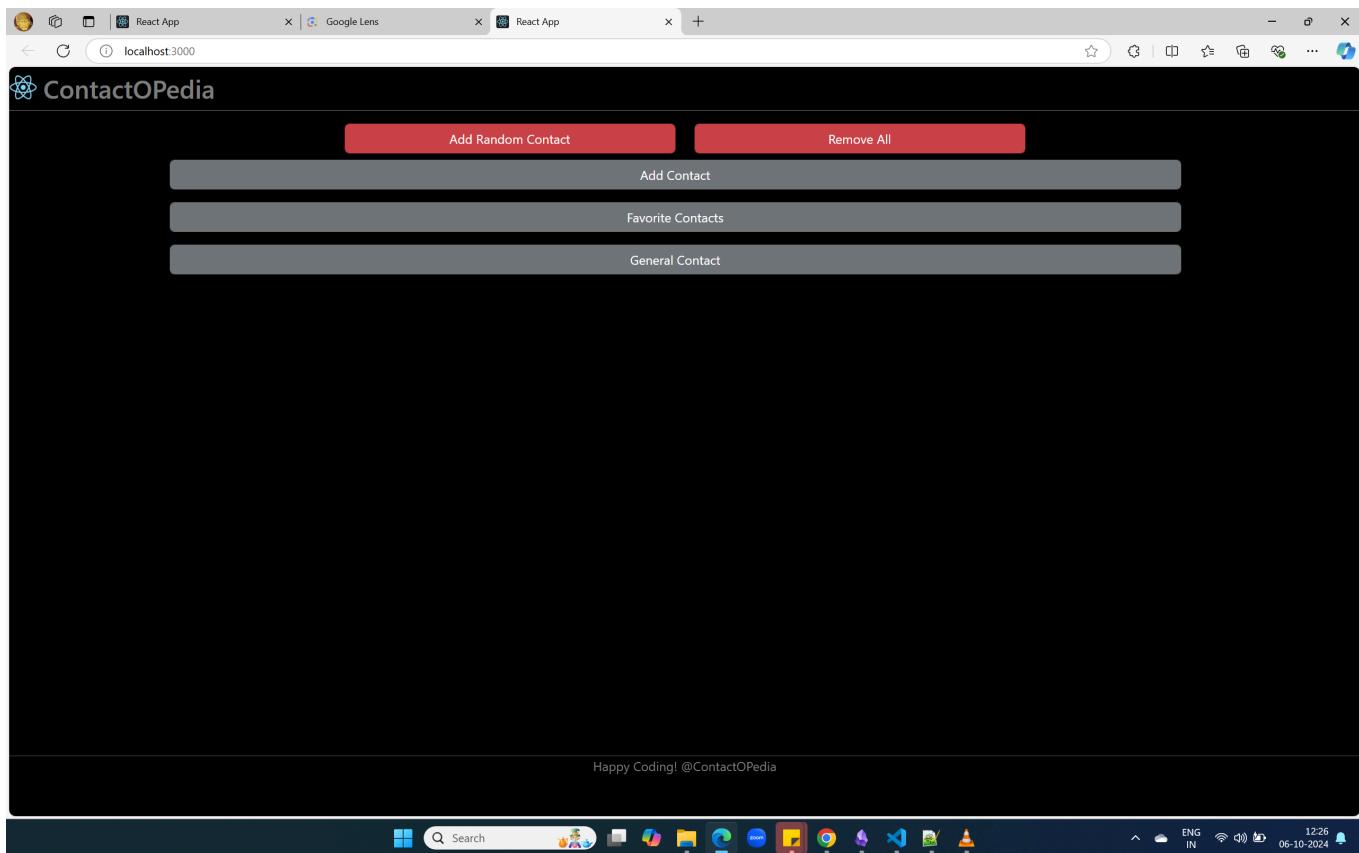
index.js file code

import React from "react";
import ReactDOM from "react-dom/client";
import ContactIndex from "./Components/ContactPages/ContactIndex";
import Header from "./Components/Layout/Header";

const root = ReactDOM.createRoot(document.getElementById("root"));
root.render(<ContactIndex />);

```

so now the image will look like this



Now i will work with AddContact.jsx file now changing the code is like this

```

const AddContact = () => {
 return (
 <div className="border row text-white p-2">

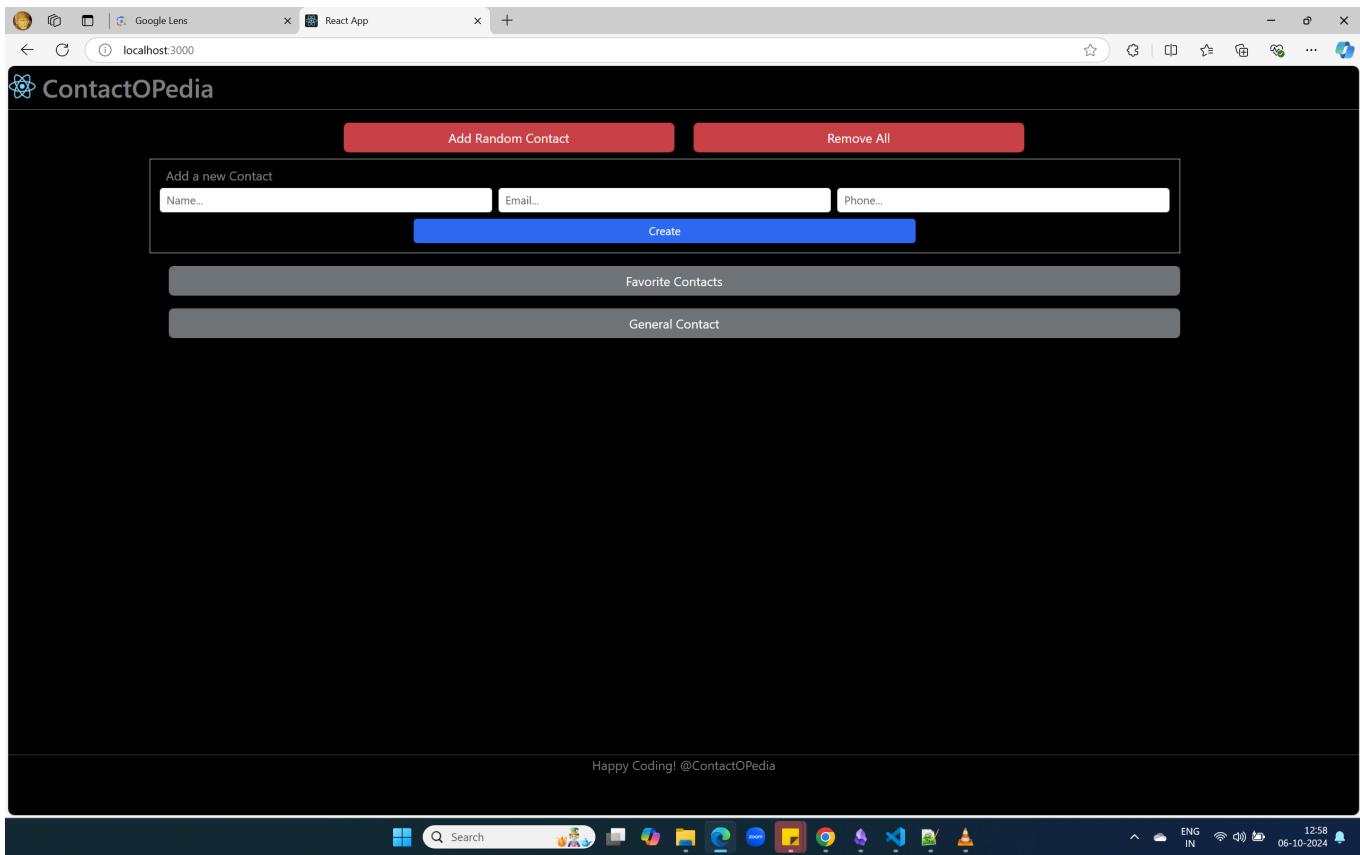
```

```
<div className="col-12 text-white-50">Add a new Contact</div>
<div className="col-12 col-md-4 p-1">
 <input
 className="form-control form-control-sm"
 placeholder="Name...">
 </input>
</div>
<div className="col-12 col-md-4 p-1">
 <input
 className="form-control form-control-sm"
 placeholder="Email...">
 </input>
</div>
<div className="col-12 col-md-4 p-1">
 <input
 className="form-control form-control-sm"
 placeholder="Phone...">
 </input>
</div>
<div className="col-12 col-md-6 offset-md-3 p-1">
 <button className="btn btn-primary btn-sm form-control">Create</button>
</div>
</div>
};

};

export default AddContact;
```

so the output will look now this



Before creating logic for create we have to maintain state we have to store all contacts using state now but in a class we have to maintain state so

## ContactIndex.jsx

A decorative horizontal line consisting of a series of red asterisks (\*).

```
import React from "react";
import Footer from "../Layout/Footer";
import Header from "../Layout/Header";
import AddContact from "./AddContact";
import AddRandomContact from "./AddRandomContact";
import FavoriteContacts from "./FavoriteContacts";
import GeneralContacts from "./GeneralContacts";
import RemoveAllContact from "./RemoveAllContact";

class ContactIndex extends React.Component {
 constructor(props) {
 super(props);
 this.state = {
 contactList: [
 {
 id: 1,
 name: "Ben Parker",
 phone: "666-666-7770",
 },
 {
 id: 2,
 name: "Peter Parker",
 phone: "666-666-7771",
 },
 {
 id: 3,
 name: "J. Jonah Jameson",
 phone: "666-666-7772",
 },
 {
 id: 4,
 name: "May Parker",
 phone: "666-666-7773",
 },
 {
 id: 5,
 name: "Norman Osborn",
 phone: "666-666-7774",
 },
 {
 id: 6,
 name: "Flash Thompson",
 phone: "666-666-7775",
 },
 {
 id: 7,
 name: "H. Peter Parker",
 phone: "666-666-7776",
 },
 {
 id: 8,
 name: "Dale Cooper",
 phone: "666-666-7777",
 },
 {
 id: 9,
 name: "Agent M",
 phone: "666-666-7778",
 },
 {
 id: 10,
 name: "Agent K",
 phone: "666-666-7779",
 },
],
 };
 }
}
```

```
 email: "ben@dotnetmastery.com",
 isFavorite: false,
 },
 {
 id: 2,
 name: "Kathy Patrick",
 phone: "111-222-0000",
 email: "kathy@dotnetmastery.com",
 isFavorite: true,
 },
 {
 id: 3,
 name: "Paul Show",
 phone: "999-222-1111",
 email: "paul@dotnetmastery.com",
 isFavorite: true,
 },
],
};

render() {
 return (
 <div>
 <Header />
 <div className="container" style={{ minHeight: "85vh" }}>
 <div className="row py-3">
 <div className="col-4 offset-2">
 <AddRandomContact />
 </div>
 <div className="col-4">
 <RemoveAllContact />
 </div>
 <div className="row py-2">
 <AddContact />
 </div>
 <div className="row py-2">
 <FavoriteContacts
 contacts={this.state.contactList.filter(
 (u) => u.isFavorite == true
)}
 />
 </div>
 <div className="row py-2">
 <GeneralContacts
 contacts={this.state.contactList.filter(

```

```
 (u) => u.isFavorite == false
)}
 />
</div>
</div>
</div>
<Footer />
</div>
);
}
}
```

```
export default ContactIndex;
```

### FavoriteContacts.jsx

```

import Contact from "./Contact";

const FavoriteContacts = (props) => {
 return (
 <div>
 {props.contacts.map((contact, index) => (
 <Contact contact={contact} key={index}></Contact>
))}
 </div>
);
};

export default FavoriteContacts;
```

### GeneralContacts.jsx

```

import Contact from "./Contact";

const GeneralContacts = (props) => {
 return (
 <div>
 {props.contacts.map((contact, index) => (
 <Contact contact={contact} key={index}></Contact>
))}
 </div>
);
};
```

```
};

export default GeneralContacts;
```

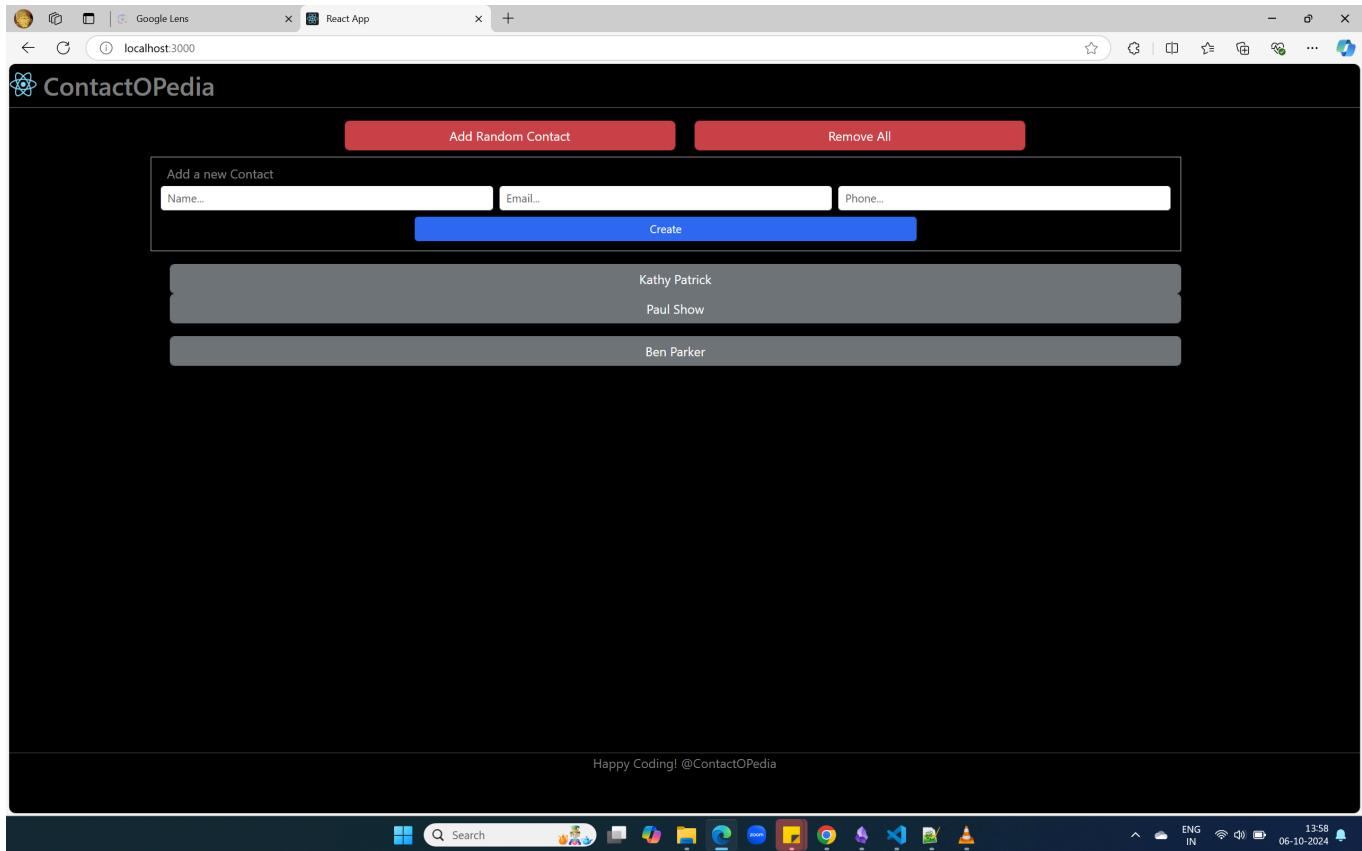
### Contact.jsx

```

```

```
const Contact = (props) => {
 return (
 <div>
 <button className="btn btn-secondary form-control">
 {props.contact.name}
 </button>
 </div>
);
};

export default Contact;
```



How many elements we have in contact. Jsx that many will be shown in design okay

Now as usual added Bootstrap icons link in index and and also installed faker and doing all this

and

```
npm install @faker-js/faker --save-dev
```

code updated like this for contact. Jsx file

```
const Contact = (props) => {
 return (
 <div
 className="row p-md-2 mb-2"
 style={{ borderRadius: "20px", border: "1px solid #555" }}
 >
 <div className="col-2 col-md-1 pt-2 pt-md-1">
 <img
 src={`https://ui-avatars.com/api/?name=${props.contact.name}`}
 style={{ width: "80%" }}
 alt=""
 />
 </div>
 <div className="col-6 col-md-5 text-warning pt-0">
 {props.contact.name}

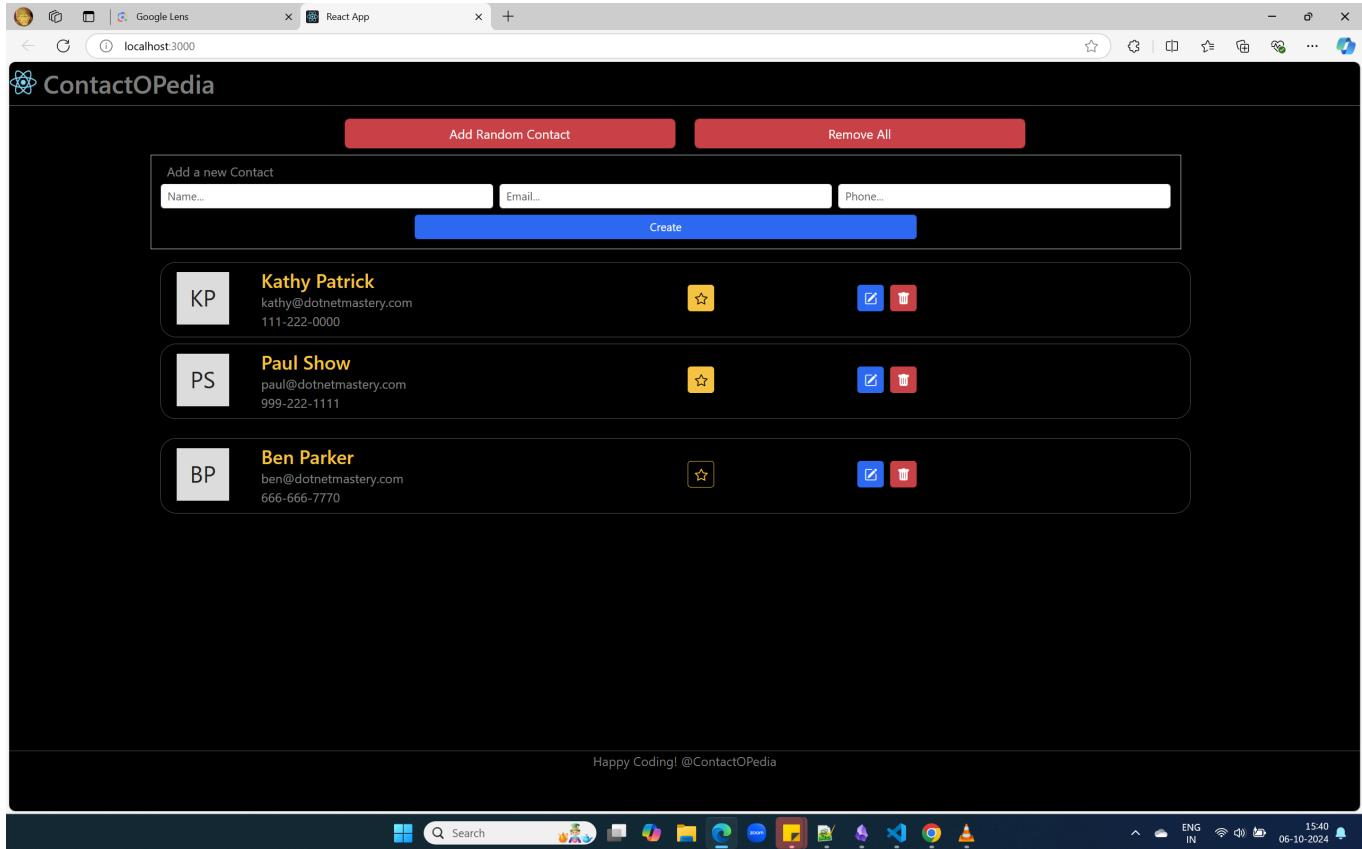
 <div className="text-white-50">
 {props.contact.email}

 {props.contact.phone}
 </div>
 </div>
 <div className="col-2 col-md-2 pt-md-3">
 <button
 className={`${'btn btn-sm m-1 ${props.contact.isFavorite ? "btn-warning" : "btn-outline-warning"}`}
 }>
 <i class="bi bi-star" style={{ fontSize: "1rem" }}></i>
 </button>
 </div>
 <div className="col-2 col-md-3 pt-md-3">
 <button className="btn btn-primary btn-sm m-1">
 <i class="bi bi-pencil-square" style={{ fontSize: "1rem" }}></i>
 </button>
 <button className="btn btn-danger btn-sm m-1">
 <i class="bi bi-trash-fill" style={{ fontSize: "1rem" }}></i>
 </button>
 </div>
 </div>
)
}
```

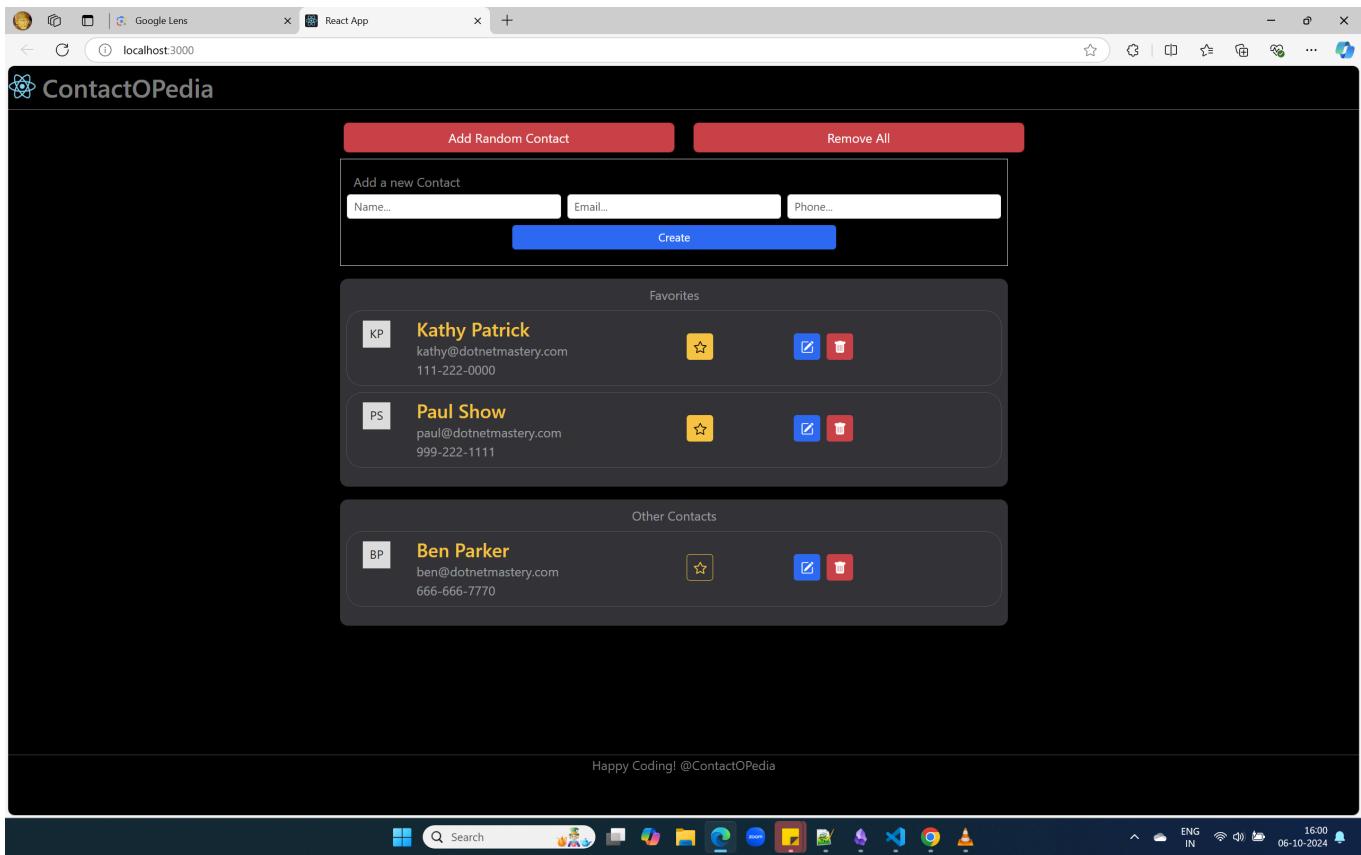
```
)
};

export default Contact;
```

and the output now will be like this



after further changing add contact ,contact index and favorite and general codes we will get the screen like this and \



the changes codes code is below which is doing change in look and feel plug

```
const AddContact = () => {
 return (
 <div className="border col-12 text-white p-2">
 <div className="row p-2">
 <div className="col-12 text-white-50">Add a new Contact</div>
 <div className="col-12 col-md-4 p-1">
 <input
 className="form-control form-control-sm"
 placeholder="Name..."></input>
 </div>
 <div className="col-12 col-md-4 p-1">
 <input
 className="form-control form-control-sm"
 placeholder="Email..."></input>
 </div>
 <div className="col-12 col-md-4 p-1">
 <input
 className="form-control form-control-sm"
 placeholder="Phone..."></input>
 </div>
 </div>
)
}
```

```

 <div className="col-12 col-md-6 offset-md-3 p-1">
 <button className="btn btn-primary btn-sm form-control">
 Create
 </button>
 </div>
 </div>
);
}

export default AddContact;

```

\*\*\*\*\*

```

import React from "react";
import Footer from "../Layout/Footer";
import Header from "../Layout/Header";
import AddContact from "./AddContact";
import AddRandomContact from "./AddRandomContact";
import FavoriteContacts from "./FavoriteContacts";
import GeneralContacts from "./GeneralContacts";
import RemoveAllContact from "./RemoveAllContact";

class ContactIndex extends React.Component {
 constructor(props) {
 super(props);
 this.state = {
 contactList: [
 {
 id: 1,
 name: "Ben Parker",
 phone: "666-666-7770",
 email: "ben@dotnetmastery.com",
 isFavorite: false,
 },
 {
 id: 2,
 name: "Kathy Patrick",
 phone: "111-222-0000",
 email: "kathy@dotnetmastery.com",
 isFavorite: true,
 },
 {
 id: 3,
 name: "Paul Show",
 phone: "999-222-1111",
 }
]
 };
 }

 render() {
 return (
 <div>
 <Header />
 <AddContact />
 <AddRandomContact />
 <GeneralContacts />
 <FavoriteContacts />
 <RemoveAllContact />
 </div>
);
 }
}

```

```
 email: "paul@dotnetmastery.com",
 isFavorite: true,
 },
],
};

render() {
 return (
 <div>
 <Header />
 <div className="container" style={{ minHeight: "85vh" }}>
 <div className="row py-3">
 <div className="col-4 offset-2">
 <AddRandomContact />
 </div>
 <div className="col-4">
 <RemoveAllContact />
 </div>
 <div className="row py-2">
 <div className="col-8 offset-2 row">
 <AddContact />
 </div>
 </div>
 <div className="row py-2">
 <div className="col-8 offset-2 row">
 <FavoriteContacts
 contacts={this.state.contactList.filter(
 (u) => u.isFavorite == true
)}
 />
 </div>
 </div>
 <div className="row py-2">
 <div className="col-8 offset-2 row">
 <GeneralContacts
 contacts={this.state.contactList.filter(
 (u) => u.isFavorite == false
)}
 />
 </div>
 </div>
 </div>
 <Footer />
 </div>
)
}
```

```
);
 }
}

export default ContactIndex;

```

```
import Contact from "./Contact";

const FavoriteContacts = (props) => {
 return (
 <div
 className="col-12 py-2"
 style={{ borderRadius: "10px", backgroundColor: "#323637" }}
 >
 <div className="text-center text-white-50">Favorites</div>
 <div className="p-2">
 {props.contacts.map((contact, index) => (
 <Contact contact={contact} key={index}></Contact>
)));
 </div>
 </div>
);
}

export default FavoriteContacts;
```

```

```

```
**
import Contact from "./Contact";

const GeneralContacts = (props) => {
 return (
 <div
 className="col-12 py-2"
 style={{ borderRadius: "10px", backgroundColor: "#323637" }}
 >
 <div className="text-center text-white-50">Other Contacts</div>
 <div className="p-2">
 {props.contacts.map((contact, index) => (
 <Contact contact={contact} key={index}></Contact>
)));
 </div>
 </div>
);
}
```

```

);
}

export default GeneralContacts;

```

Now work on contact handler where i will be adding contact so go to ContactIndex.jsx

Now in the ContactIndex if u see i had passed to child component the collection property contacts with that name and in that name i am sending the state values okay and as it is class based component if i put this it is through props now in the same manner to the child AddContact i can send one function code also as props means from the child component addcontact i will call this method okay and in order to prevent default on submit event i will use preventDefaults okay

Let us code now

```

handleAddContact=()=>{
{
 alert("Hello")
}
//above render i am defining the method okay
render() {
// down then in addcontact
<div className="row py-2">
 <div className="col-8 offset-2 row">
 <AddContact handleAddContact={this.handleAddContact} />
 </div>

```

Now go to add contact and define one form over there and onclick of form again one function okay like this

```

<form onSubmit={this.handleAddContactFormSubmit}>
 ...
 ...
 ...
</form>
// then again above render method but u need to convert this to class again as
i want to store again some status properties now okay so the complete code how
it changes is like this which is given down okay

handleAddContactFormSubmit = (e) => {
 e.preventDefault();
 this.props.handleAddContact();
};

```

```
// if i comment preventdefault then button form submit will called u can see
the web page will refresh but if comment that line and click create form
submit the web page will not refresh telling that foorm submit is not called
once test it practicaally now i am putting complete code of ContactIndex and
AddContact now

import React from "react";
import Footer from "../Layout/Footer";
import Header from "../Layout/Header";
import AddContact from "./AddContact";
import AddRandomContact from "./AddRandomContact";
import FavoriteContacts from "./FavoriteContacts";
import GeneralContacts from "./GeneralContacts";
import RemoveAllContact from "./RemoveAllContact";

class ContactIndex extends React.Component {
 constructor(props) {
 super(props);
 this.state = {
 contactList: [
 {
 id: 1,
 name: "Ben Parker",
 phone: "666-666-7770",
 email: "ben@dotnetmastery.com",
 isFavorite: false,
 },
 {
 id: 2,
 name: "Kathy Patrick",
 phone: "111-222-0000",
 email: "kathy@dotnetmastery.com",
 isFavorite: true,
 },
 {
 id: 3,
 name: "Paul Show",
 phone: "999-222-1111",
 email: "paul@dotnetmastery.com",
 isFavorite: true,
 },
],
 };
 }
 handleAddContact=()=>=>
 {
 alert("Hello")
 }
}
```

```
}

render() {
 return (
 <div>
 <Header />
 <div className="container" style={{ minHeight: "85vh" }}>
 <div className="row py-3">
 <div className="col-4 offset-2">
 <AddRandomContact />
 </div>
 <div className="col-4">
 <RemoveAllContact />
 </div>
 </div>
 <div className="row py-2">
 <div className="col-8 offset-2 row">
 <AddContact handleAddContact={this.handleAddContact} />
 </div>
 </div>
 <div className="row py-2">
 <div className="col-8 offset-2 row">
 <FavoriteContacts
 contacts={this.state.contactList.filter(
 (u) => u.isFavorite == true
)}
 />
 </div>
 </div>
 <div className="row py-2">
 <div className="col-8 offset-2 row">
 <GeneralContacts
 contacts={this.state.contactList.filter(
 (u) => u.isFavorite == false
)}
 />
 </div>
 </div>
 <div>
 <Footer />
 </div>
 </div>
);
}

export default ContactIndex;
```

```
import React from "react";

class AddContact extends React.Component {
 constructor(props) {
 super(props);
 this.state = {
 errorMessage: undefined,
 successMessage: undefined,
 };
 }
 handleAddContactFormSubmit = (e) => {
 e.preventDefault();
 this.props.handleAddContact();
 };
 render() {
 return (
 <div className="border col-12 text-white p-2">
 <form onSubmit={this.handleAddContactFormSubmit}>
 <div className="row p-2">
 <div className="col-12 text-white-50">Add a new Contact</div>
 <div className="col-12 col-md-4 p-1">
 <input
 className="form-control form-control-sm"
 placeholder="Name..."></input>
 </div>
 <div className="col-12 col-md-4 p-1">
 <input
 className="form-control form-control-sm"
 placeholder="Email..."></input>
 </div>
 <div className="col-12 col-md-4 p-1">
 <input
 className="form-control form-control-sm"
 placeholder="Phone..."></input>
 </div>
 <div className="col-12 col-md-6 offset-md-3 p-1">
 <button className="btn btn-primary btn-sm form-control">
 Create
 </button>
 </div>
 </div>
 </form>
 </div>
);
 }
}
```

```

);
}

export default AddContact;

```

Now in add Contact i will define name properties and will add using target and will pass to ContactIndex which will take that object and using spread operator it will add into existing contact list and Id is updated by taking so all coding goes down try to understand it in one go

```

import React from "react";

class AddContact extends React.Component {
 constructor(props) {
 super(props);
 this.state = {
 errorMessage: undefined,
 successMessage: undefined,
 };
 }
 handleAddContactFormSubmit = (e) => {
 e.preventDefault();
 const name = e.target.elements.contactName.value.trim();
 const email = e.target.elements.contactEmail.value.trim();
 const phone = e.target.elements.contactPhone.value.trim();
 this.props.handleAddContact({ name, email, phone });
 };

 render() {
 return (
 <div className="border col-12 text-white p-2">
 <form onSubmit={this.handleAddContactFormSubmit}>
 <div className="row p-2">
 <div className="col-12 text-white-50">Add a new Contact</div>
 <div className="col-md-4 p-1">
 <input
 className="form-control form-control-sm"
 placeholder="Name..."
 name="contactName"
 ></input>
 </div>
 <div className="col-12 col-md-4 p-1">
 <input
 className="form-control form-control-sm"
 placeholder="Email..."
 ></input>
 </div>
 </div>
 </form>
 </div>
);
 }
}

```

```

 name="contactEmail"
 ></input>
 </div>
 <div className="col-12 col-md-4 p-1">
 <input
 className="form-control form-control-sm"
 placeholder="Phone..."
 name="contactPhone"
 ></input>
 </div>
 <div className="col-12 col-md-6 offset-md-3 p-1">
 <button className="btn btn-primary btn-sm form-control">
 Create
 </button>
 </div>
 </div>
</form>
</div>
);
}
}

export default AddContact;

```

```

import React from "react";
import Footer from "../Layout/Footer";
import Header from "../Layout/Header";
import AddContact from "./AddContact";
import AddRandomContact from "./AddRandomContact";
import FavoriteContacts from "./FavoriteContacts";
import GeneralContacts from "./GeneralContacts";
import RemoveAllContact from "./RemoveAllContact";

class ContactIndex extends React.Component {
 constructor(props) {
 super(props);
 this.state = {
 contactList: [
 {
 id: 1,
 name: "Ben Parker",
 phone: "666-666-7770",
 email: "ben@dotnetmastery.com",
 isFavorite: false,
 },
 {

```

```

 id: 2,
 name: "Kathy Patrick",
 phone: "111-222-0000",
 email: "kathy@dotnetmastery.com",
 isFavorite: true,
 },
 {
 id: 3,
 name: "Paul Show",
 phone: "999-222-1111",
 email: "paul@dotnetmastery.com",
 isFavorite: true,
 },
],
};

handleAddContact = (newContact) => {
 const newFinalContact = {
 ...newContact,
 id: this.state.contactList[this.state.contactList.length - 1].id + 1,
 isFavorite: false,
 };
 this.setState((prevState) => {
 return {
 contactList: prevState.contactList.concat([newFinalContact]),
 };
 });
 alert("hello");
};

render() {
 return (
 <div>
 <Header />
 <div className="container" style={{ minHeight: "85vh" }}>
 <div className="row py-3">
 <div className="col-4 offset-2">
 <AddRandomContact />
 </div>
 <div className="col-4">
 <RemoveAllContact />
 </div>
 <div className="row py-2">
 <div className="col-8 offset-2 row">
 <AddContact handleAddContact={this.handleAddContact} />
 </div>
 </div>
 </div>
 </div>
 </div>
);
}

```

```

 </div>
 </div>
 <div className="row py-2">
 <div className="col-8 offset-2 row">
 <FavoriteContacts
 contacts={this.state.contactList.filter(
 (u) => u.isFavorite == true
)}
 />
 </div>
 </div>
 <div className="row py-2">
 <div className="col-8 offset-2 row">
 <GeneralContacts
 contacts={this.state.contactList.filter(
 (u) => u.isFavorite == false
)}
 />
 </div>
 </div>
 <div>
 <Footer />
 </div>
);
}
}

export default ContactIndex;

```

so things are working fine next what to do is i need to add validation and display notification so at a time for both i am keeping the updated code understand it from code or see the video from 3:24:21 to 3:32:21

```

import React from "react";

class AddContact extends React.Component {
 constructor(props) {
 super(props);
 this.state = {
 errorMessage: undefined,
 successMessage: undefined,

```

```
 };
}

handleAddContactFormSubmit = (e) => {
 e.preventDefault();
 const name = e.target.elements.contactName.value.trim();
 const email = e.target.elements.contactEmail.value.trim();
 const phone = e.target.elements.contactPhone.value.trim();
 const response = this.props.handleAddContact({
 name: name,
 email: email,
 phone: phone,
 });

 if (response.status == "success") {
 this.setState({ errorMessage: undefined, successMessage: response.msg });
 } else {
 this.setState({ errorMessage: response.msg, successMessage: undefined });
 }
};

render() {
 return (
 <div className="border col-12 text-white p-2">
 <form
 onSubmit={this.handleAddContactFormSubmit}
 className="contact-form"
 >
 <div className="row p-2">
 <div className="col-12 text-white-50">Add a new Contact</div>
 <div className="col-12 col-md-4 p-1">
 <input
 className="form-control form-control-sm"
 placeholder="Name..."
 name="contactName"
 ></input>
 </div>
 <div className="col-12 col-md-4 p-1">
 <input
 className="form-control form-control-sm"
 placeholder="Email..."
 name="contactEmail"
 ></input>
 </div>
 </div>
 </form>
 </div>
);
}
```

```

 <div className="col-12 col-md-4 p-1">
 <input
 className="form-control form-control-sm"
 placeholder="Phone..."
 name="contactPhone"
 ></input>
 </div>

 {this.state.errorMessage == undefined ? (
 <div></div>
) : (
 <div className="col-12 text-center text-danger">
 {this.state.errorMessage}
 </div>
)}
 }

 {this.state.successMessage == undefined ? (
 <div></div>
) : (
 <div className="col-12 text-center text-success">
 {this.state.successMessage}
 </div>
)}
 }

 <div className="col-12 col-md-6 offset-md-3 p-1">
 <button className="btn btn-primary btn-sm form-control">
 Create
 </button>
 </div>
 </div>
</form>
</div>
);

}
}

export default AddContact;

```

```

import React from "react";
import Footer from "../Layout/Footer";
import Header from "../Layout/Header";
import AddContact from "./AddContact";
import AddRandomContact from "./AddRandomContact";
import FavoriteContacts from "./FavoriteContacts";
import GeneralContacts from "./GeneralContacts";
import RemoveAllContact from "./RemoveAllContact";

```

```
class ContactIndex extends React.Component {
 constructor(props) {
 super(props);
 this.state = {
 contactList: [
 {
 id: 1,
 name: "Ben Parker",
 phone: "666-666-7770",
 email: "ben@dotnetmastery.com",
 isFavorite: false,
 },
 {
 id: 2,
 name: "Kathy Patrick",
 phone: "111-222-0000",
 email: "kathy@dotnetmastery.com",
 isFavorite: true,
 },
 {
 id: 3,
 name: "Paul Show",
 phone: "999-222-1111",
 email: "paul@dotnetmastery.com",
 isFavorite: true,
 },
],
 };
 }

 handleAddContact = (newContact) => {
 if (newContact.name === "") {
 return { status: "failure", msg: "Please Enter a valid Name" };
 } else if (newContact.phone === "") {
 return { status: "failure", msg: "Please Enter a valid Phone Number" };
 }
 const duplicateRecord = this.state.contactList.filter((x) => {
 if (x.name === newContact.name && x.phone === newContact.phone) {
 return true;
 }
 });
 if (duplicateRecord.length > 0) {
 return { status: "failure", msg: "Duplicate Record" };
 } else {
 const newFinalContact = {
```

```
 ... newContact,
 id: this.state.contactList[this.state.contactList.length - 1].id + 1,
 isFavorite: false,
};

this.setState((prevState) => {
 return {
 contactList: prevState.contactList.concat([newFinalContact]),
 };
});

return { status: "success", msg: "Contact was added successfully" };
}

};

render() {
 return (
 <div>
 <Header />
 <div className="container" style={{ minHeight: "85vh" }}>
 <div className="row py-3">
 <div className="col-4 offset-2">
 <AddRandomContact />
 </div>
 <div className="col-4">
 <RemoveAllContact />
 </div>
 <div className="row py-2">
 <div className="col-8 offset-2 row">
 <AddContact handleAddContact={this.handleAddContact} />
 </div>
 </div>
 <div className="row py-2">
 <div className="col-8 offset-2 row">
 <FavoriteContacts
 contacts={this.state.contactList.filter(
 (u) => u.isFavorite == true
)}
 />
 </div>
 </div>
 <div className="row py-2">
 <div className="col-8 offset-2 row">
 <GeneralContacts
 contacts={this.state.contactList.filter(
 (u) => u.isFavorite == false
)}
 />
 </div>
 </div>
 </div>
 </div>
);
}
```

```

 </div>
 </div>
</div>
</div>
<Footer />
</div>
);
}
}

export default ContactIndex;

```

Now i want to implement toggle effect refer video from 3:32:21 to 3:41:41

I am just pasting the updated code now and will analyze by myself

you can see how this event is moving from ContactIndex to Favorite's and Genralcontacts and then to Contacts understand the flow

```

import React from "react";
import Footer from "../Layout/Footer";
import Header from "../Layout/Header";
import AddContact from "./AddContact";
import AddRandomContact from "./AddRandomContact";
import FavoriteContacts from "./FavoriteContacts";
import GeneralContacts from "./GeneralContacts";
import RemoveAllContact from "./RemoveAllContact";

class ContactIndex extends React.Component {
 constructor(props) {
 super(props);

 this.state = {
 contactList: [
 {
 id: 1,
 name: "Ben Parker",
 phone: "666-666-7770",
 email: "ben@dotnetmastery.com",
 isFavorite: false,
 },
 {
 id: 2,
 name: "Kathy Patrick",
 phone: "111-222-0000",
 }
]
 };
 }

 handleToggleFavorite = (id) => {
 const { contactList } = this.state;
 const index = contactList.findIndex(contact => contact.id === id);
 const newContactList = [...contactList];
 newContactList[index].isFavorite = !newContactList[index].isFavorite;
 this.setState({ contactList: newContactList });
 }

 render() {
 return (
 <div>
 <Header />
 <div>
 <AddContact />
 <AddRandomContact />
 <FavoriteContacts />
 <GeneralContacts />
 <RemoveAllContact />
 </div>
 <Footer />
 </div>
);
 }
}

export default ContactIndex;

```

```

 email: "kathy@dotnetmastery.com",
 isFavorite: true,
 },
 {
 id: 3,
 name: "Paul Show",
 phone: "999-222-1111",
 email: "paul@dotnetmastery.com",
 isFavorite: true,
 },
],
};

handleAddContact = (newContact) => {
 if (newContact.name == "") {
 return { status: "failure", msg: "Please Enter a valid Name" };
 } else if (newContact.phone == "") {
 return { status: "failure", msg: "Please Enter a valid Phone Number" };
 }
 const duplicateRecord = this.state.contactList.filter((x) => {
 if (x.name == newContact.name && x.phone == newContact.phone) {
 return true;
 }
 });
 if (duplicateRecord.length > 0) {
 return { status: "failure", msg: "Duplicate Record" };
 } else {
 const newFinalContact = {
 ...newContact,
 id: this.state.contactList[this.state.contactList.length - 1].id + 1,
 isFavorite: false,
 };
 this.setState((prevState) => {
 return {
 contactList: prevState.contactList.concat([newFinalContact]),
 };
 });
 return { status: "success", msg: "Contact was added successfully" };
 }
};

handleToggleFavorites = (contact) => {
 this.setState((prevState) => {
 return {
 contactList: prevState.contactList.map((obj) => {

```

```
 if (obj.id == contact.id) {
 return { ... obj, isFavorite: !obj.isFavorite };
 }
 return obj;
 },
},
);
};

render() {
 return (
 <div>
 <Header />
 <div className="container" style={{ minHeight: "85vh" }}>
 <div className="row py-3">
 <div className="col-4 offset-2">
 <AddRandomContact />
 </div>
 <div className="col-4">
 <RemoveAllContact />
 </div>
 </div>
 <div className="row py-2">
 <div className="col-8 offset-2 row">
 <AddContact handleAddContact={this.handleAddContact} />
 </div>
 </div>
 <div className="row py-2">
 <div className="col-8 offset-2 row">
 <FavoriteContacts
 contacts={this.state.contactList.filter(
 (u) => u.isFavorite == true
)}
 favoriteClick={this.handleToggleFavorites}
 />
 </div>
 </div>
 <div className="row py-2">
 <div className="col-8 offset-2 row">
 <GeneralContacts
 contacts={this.state.contactList.filter(
 (u) => u.isFavorite == false
)}
 favoriteClick={this.handleToggleFavorites}
 />
 </div>
 </div>
 </div>
 </div>
);
}
```

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

export default ContactIndex;

import Contact from './Contact';

const FavoriteContacts = (props) => {
return (
<div
 className="col-12 py-2"
 style={{ borderRadius: "10px", backgroundColor: "#323637" }}
>
 <div className="text-center text-white-50">Favorites</div>
 <div className="p-2">
 {props.contacts.map((contact, index) => (
 <Contact
 contact={contact}
 key={index}
 favoriteClick={props.favoriteClick}
 ></Contact>
))}
 </div>
</div>
);
};

export default FavoriteContacts;

import Contact from './Contact';

const GeneralContacts = (props) => {
return (
<div
 className="col-12 py-2"
 style={{ borderRadius: "10px", backgroundColor: "#323637" }}
>
 <div className="text-center text-white-50">Other Contacts</div>
 <div className="p-2">
 {props.contacts.map((contact, index) => (
 <Contact
```

```

 contact={contact}
 key={index}
 favoriteClick={props.favoriteClick}
 ></Contact>
)})
 </div>
</div>
);
};

export default GeneralContacts;

const Contact = (props) => {
 return (
 <div
 className="row p-md-2 mb-2"
 style={{ borderRadius: "20px", border: "1px solid #555" }}
 >
 <div className="col-2 col-md-1 pt-2 pt-md-1">
 <img
 src={`https://ui-avatars.com/api/?name=${props.contact.name}`}
 style={{ width: "80%" }}
 alt=""
 />
 </div>
 <div className="col-6 col-md-5 text-warning pt-0">
 {props.contact.name}

 <div className="text-white-50">
 {props.contact.email}

 {props.contact.phone}
 </div>
 </div>
 <div className="col-2 col-md-2 pt-md-3">
 <button
 className={`${`btn btn-sm m-1 ${(
 props.contact.isFavorite ? "btn-warning" : "btn-outline-warning"
)}`}`}
 onClick={() => props.favoriteClick(props.contact)}
 >
 <i className="bi bi-star" style={{ fontSize: "1rem" }}></i>
 </button>
 </div>
 <div className="col-2 col-md-3 pt-md-3">
 <button className="btn btn-primary btn-sm m-1">

```

```

 <i className="bi bi-pencil-square" style={{ fontSize: "1rem" }}></i>
 </button>
 <button className="btn btn-danger btn-sm m-1">
 <i className="bi bi-trash-fill" style={{ fontSize: "1rem" }}></i>
 </button>
</div>
</div>
);
};

export default Contact;

```

Now implementing delete refer video from 3:41:41 to 3:45:37

so pasting code same logic it is and things are moving to same components as favorites we have done just change the logic

```

import React from "react";
import Footer from "../Layout/Footer";
import Header from "../Layout/Header";
import AddContact from "./AddContact";
import AddRandomContact from "./AddRandomContact";
import FavoriteContacts from "./FavoriteContacts";
import GeneralContacts from "./GeneralContacts";
import RemoveAllContact from "./RemoveAllContact";

class ContactIndex extends React.Component {
 constructor(props) {
 super(props);

 this.state = {
 contactList: [
 {
 id: 1,
 name: "Ben Parker",
 phone: "666-666-7770",
 email: "ben@dotnetmastery.com",
 isFavorite: false,
 },
 {
 id: 2,
 name: "Kathy Patrick",
 phone: "111-222-0000",
 email: "kathy@dotnetmastery.com",
 }
]
 };
 }

 render() {
 return (
 <div>
 <Header />
 <AddContact />
 <GeneralContacts />
 <FavoriteContacts />
 <Footer />
 </div>
);
 }
}

export default ContactIndex;

```

```

 isFavorite: true,
 },
 {
 id: 3,
 name: "Paul Show",
 phone: "999-222-1111",
 email: "paul@dotnetmastery.com",
 isFavorite: true,
 },
],
};

handleAddContact = (newContact) => {
 if (newContact.name == "") {
 return { status: "failure", msg: "Please Enter a valid Name" };
 } else if (newContact.phone == "") {
 return { status: "failure", msg: "Please Enter a valid Phone Number" };
 }
 const duplicateRecord = this.state.contactList.filter((x) => {
 if (x.name == newContact.name && x.phone == newContact.phone) {
 return true;
 }
 });
 if (duplicateRecord.length > 0) {
 return { status: "failure", msg: "Duplicate Record" };
 } else {
 const newFinalContact = {
 ...newContact,
 id: this.state.contactList[this.state.contactList.length - 1].id + 1,
 isFavorite: false,
 };
 this.setState((prevState) => {
 return {
 contactList: prevState.contactList.concat([newFinalContact]),
 };
 });
 return { status: "success", msg: "Contact was added successfully" };
 }
};

handleToggleFavorites = (contact) => {
 this.setState((prevState) => {
 return {
 contactList: prevState.contactList.map((obj) => {
 if (obj.id == contact.id) {

```

```
 return { ... obj, isFavorite: !obj.isFavorite };
 }
 return obj;
),
);
};
);

handleDeleteContact = (contactId) => {
this.setState((prevState) => {
return {
 contactList: prevState.contactList.filter((obj) => {
 return obj.id !== contactId;
),
},
);
});
};

render() {
return (
<div>
 <Header />
 <div className="container" style={{ minHeight: "85vh" }}>
 <div className="row py-3">
 <div className="col-4 offset-2">
 <AddRandomContact />
 </div>
 <div className="col-4">
 <RemoveAllContact />
 </div>
 <div className="row py-2">
 <div className="col-8 offset-2 row">
 <AddContact handleAddContact={this.handleAddContact} />
 </div>
 </div>
 <div className="row py-2">
 <div className="col-8 offset-2 row">
 <FavoriteContacts
 contacts={this.state.contactList.filter(
 (u) => u.isFavorite == true
)}
 favoriteClick={this.handleToggleFavorites}
 deleteContact={this.handleDeleteContact}
 />
 </div>
 </div>
 </div>
)
```

```

 <div className="row py-2">
 <div className="col-8 offset-2 row">
 <GeneralContacts
 contacts={this.state.contactList.filter(
 (u) => u.isFavorite == false
)}
 favoriteClick={this.handleToggleFavorites}
 deleteContact={this.handleDeleteContact}
 />
 </div>
 </div>
 </div>
 <Footer />
 </div>
);
}

}

export default ContactIndex;

import Contact from "./Contact";

const FavoriteContacts = (props) => {
 return (
 <div
 className="col-12 py-2"
 style={{ borderRadius: "10px", backgroundColor: "#323637" }}
 >
 <div className="text-center text-white-50">Favorites</div>
 <div className="p-2">
 {props.contacts.map((contact, index) => (
 <Contact
 contact={contact}
 key={index}
 favoriteClick={props.favoriteClick}
 deleteContact={props.deleteContact}
 ></Contact>
)));
 </div>
 </div>
);
}

export default FavoriteContacts;

```

```
import Contact from './Contact';

const GeneralContacts = (props) => {
 return (
 <div
 className="col-12 py-2"
 style={{ borderRadius: "10px", backgroundColor: "#323637" }}
 >
 <div className="text-center text-white-50">Other Contacts</div>
 <div className="p-2">
 {props.contacts.map((contact, index) => (
 <Contact
 contact={contact}
 key={index}
 favoriteClick={props.favoriteClick}
 deleteContact={props.deleteContact}
 ></Contact>
))}
 </div>
 </div>
);
};

export default GeneralContacts;

const Contact = (props) => {
 return (
 <div
 className="row p-md-2 mb-2"
 style={{ borderRadius: "20px", border: "1px solid #555" }}
 >
 <div className="col-2 col-md-1 pt-2 pt-md-1">
 <img
 src={`https://ui-avatars.com/api/?name=${props.contact.name}`}
 style={{ width: "80%" }}
 alt=""
 />
 </div>
 <div className="col-6 col-md-5 text-warning pt-0">
 {props.contact.name}

 <div className="text-white-50">
 {props.contact.email}

 {props.contact.phone}
 </div>
 </div>
 </div>
);
};
```

```
</div>
</div>
<div className="col-2 col-md-2 pt-md-3">
 <button
 className={`btn btn-sm m-1 ${props.contact.isFavorite ? "btn-warning" : "btn-outline-warning"}`}
 onClick={() => props.favoriteClick(props.contact)}
 >
 <i className="bi bi-star" style={{ fontSize: "1rem" }}></i>
 </button>
</div>
<div className="col-2 col-md-3 pt-md-3">
 <button className="btn btn-primary btn-sm m-1">
 <i className="bi bi-pencil-square" style={{ fontSize: "1rem" }}></i>
 </button>
 <button className="btn btn-danger btn-sm m-1">
 <i
 className="bi bi-trash-fill"
 onClick={() => props.deleteContact(props.contact.id)}
 style={{ fontSize: "1rem" }}
 ></i>
 </button>
</div>
</div>
);
};

export default Contact;
```