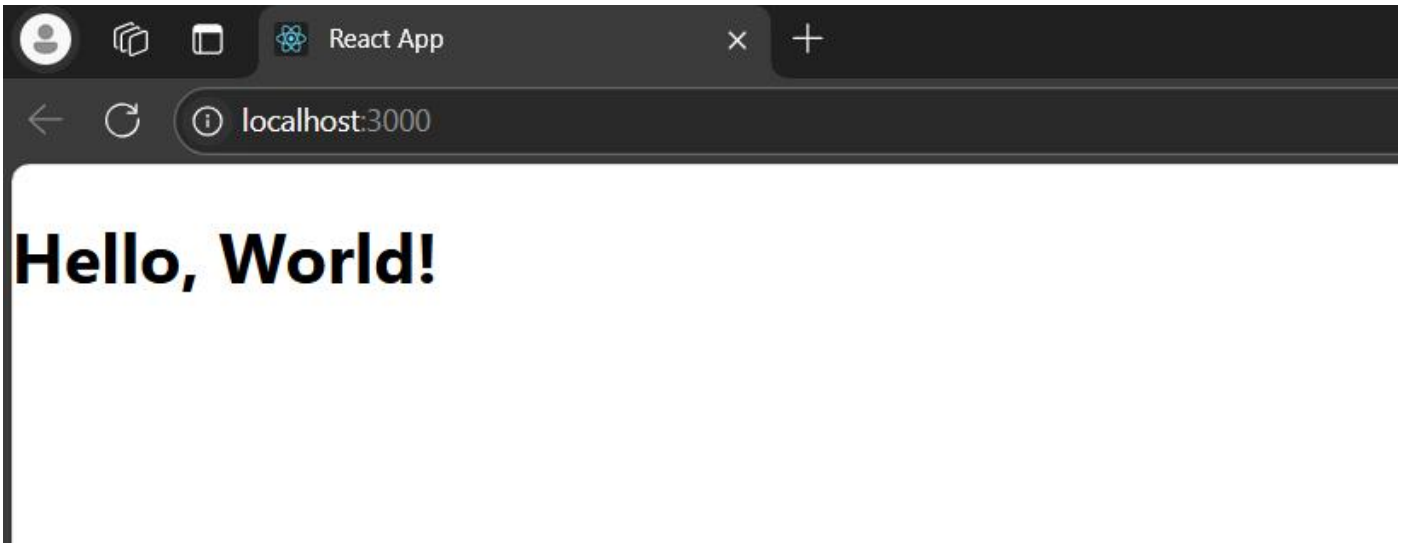
JEYSAN.V
717823F225
MERN TASK day 5 (React)

1. How to write your first React component

Tasks:

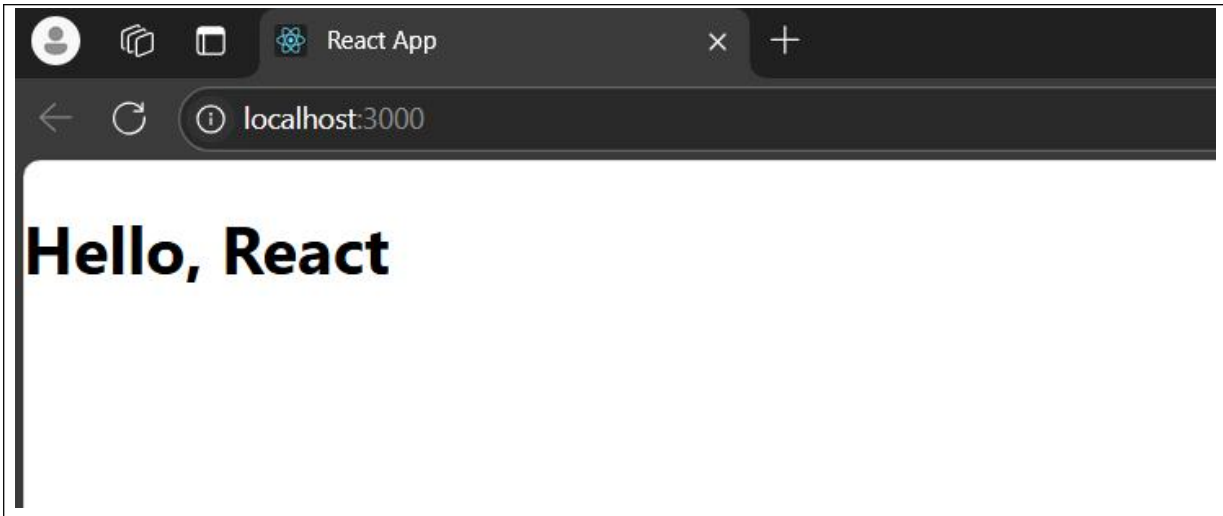
1. Create a simple functional component named Greeting that returns “Hello, World!” within a <h1> tag.

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


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



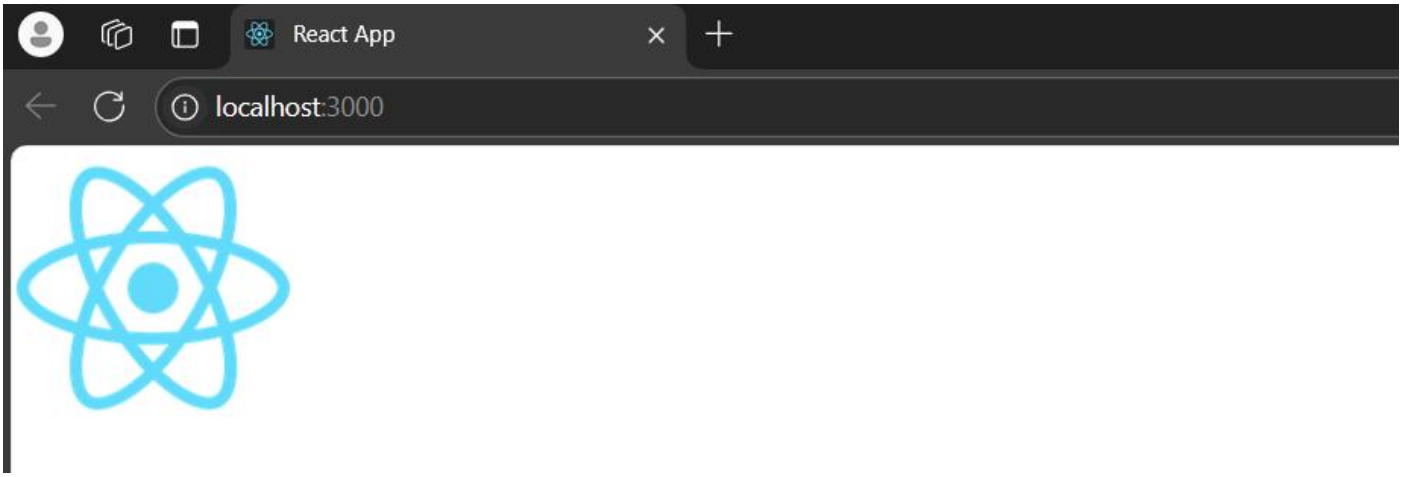
2. Modify the Greeting component to display “Hello, React!”.

```


import React from 'react';
import './App.css';
import Greeting from './Components/Greeting';


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



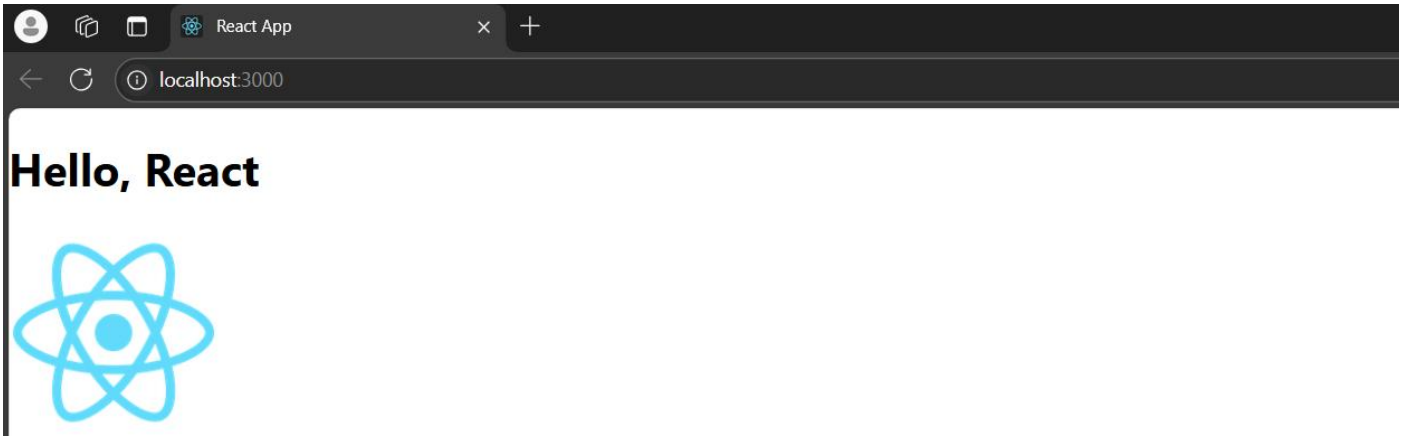
3. Create a Gallery functional component to display an image.

```
import React from 'react';

function Gallery() {
    return (
        <div>
            <img
                src="logo192.png"
                alt="Placeholder"
                style={{ width: '150px', height: '150px' }}
            />
        </div>
    );
}
export default Gallery;
```



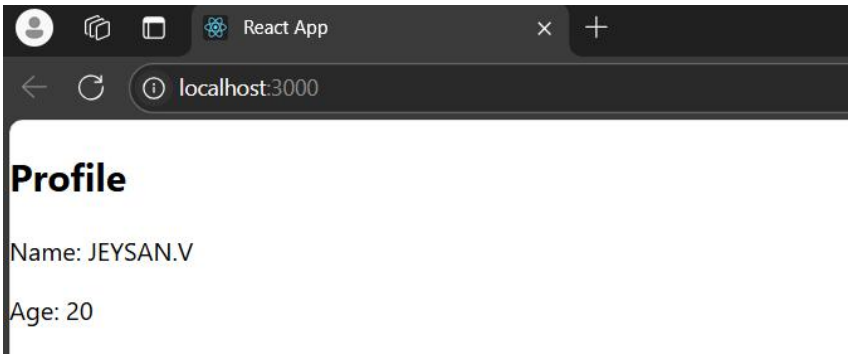
4. Add Greeting to the Gallery component and display the image and greeting.

```
import React from 'react';
import Greeting from './Greeting';

function Gallery() {
    return (
        <div>
            <Greeting />
            <img
                src="logo192.png"
                alt="Placeholder"
                style={{ width: '150px', height: '150px' }}
            />
        </div>
    );
}
export default Gallery;
```



5. Write a component called Profile which displays a hardcoded user's name and age.

```
import React from 'react';

function Profile() {
    return (
        <div>
            <h2>Profile</h2>
            <p>Name: JEYSAN.V</p>
            <p>Age: 20</p>
        </div>
    );
}
export default Profile;
```

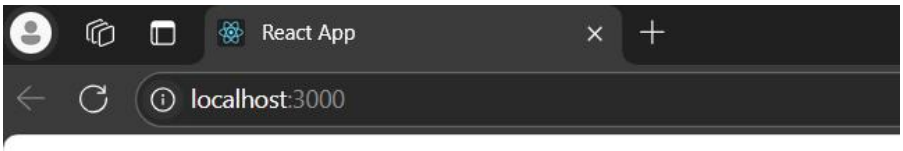


2. When and how to create multi-component files

Tasks:

1. Create a file named UserComponents.js and inside it, define two components: UserName and UserAge that display hardcoded names and ages respectively.

```
import React from 'react';

export function UserName() {
    return <p>Name: JEYSAN.V</p>;
}
export function UserAge() {
    return <p>Age: 20</p>;
}
```

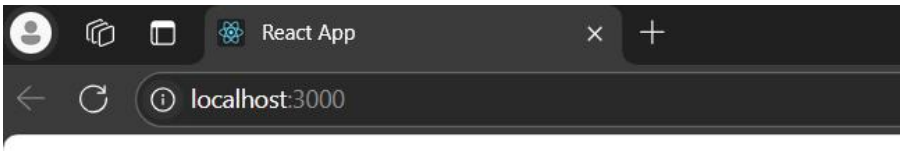


Name: JEYSAN.V

Age: 20

2. Export both UserName and UserAge from UserComponents.js.

```
import React from 'react';
export function UserName() {
    return <p>Name: JEYSAN.V</p>;
}
export function UserAge() {
    return <p>Age: 20</p>;
}
```



Name: JEYSAN.V

Age: 20

3. In a separate file, import and use both UserName and UserAge components using named imports.

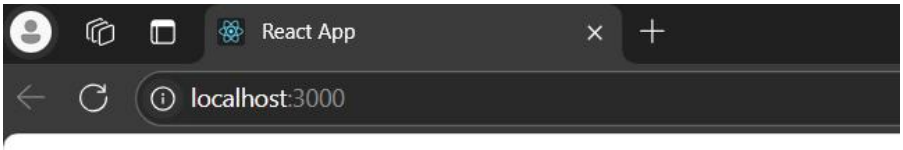
```
import React from 'react';
import { UserName, UserAge } from './UserComponent';
function UserInfo() {
    return (
        <div>
            <UserName />
            <UserAge />
        </div>
    );
}
export default UserInfo;
```

Name: JEYSAN.V

Age: 20

4. Convert UserAge into a default export and modify the importing file to accommodate the change.

```
import React from 'react';
import { UserName} from './UserComponent';
import Age from './UserComponent';
function UserInfo() {
    return (
        <div>
            <UserName />
            <Age />
        </div>
    );
}
export default UserInfo;
```



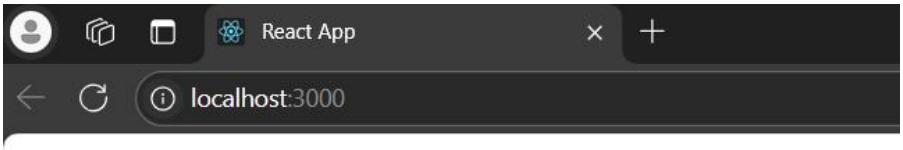
Name: JEYSAN.V

Age: 20

5. Split UserName and UserAge into separate files and adjust your imports.

```
function UserAge() {
    return <p>Age: 20</p>;
}
export default UserAge;
```

```
function UserName() {
    return <p>Name: JEYSAN.V</p>;
}
export default UserName;
```



Name: JEYSAN.V

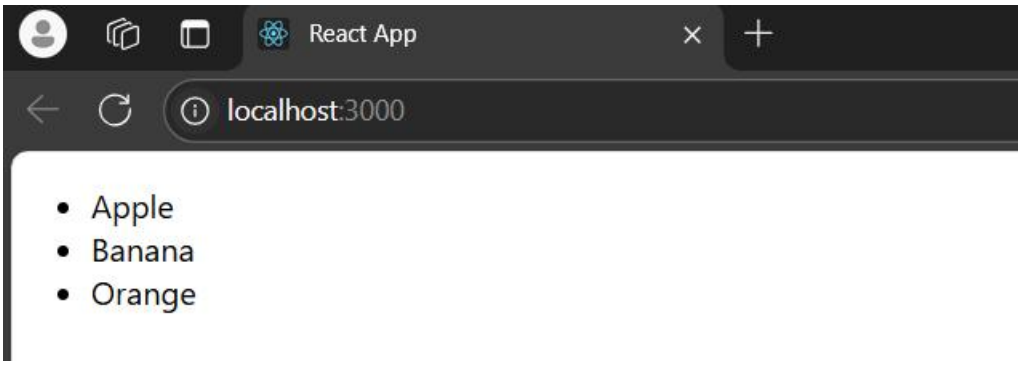
Age: 20

3. How to add markup to JavaScript with JSX

Tasks:

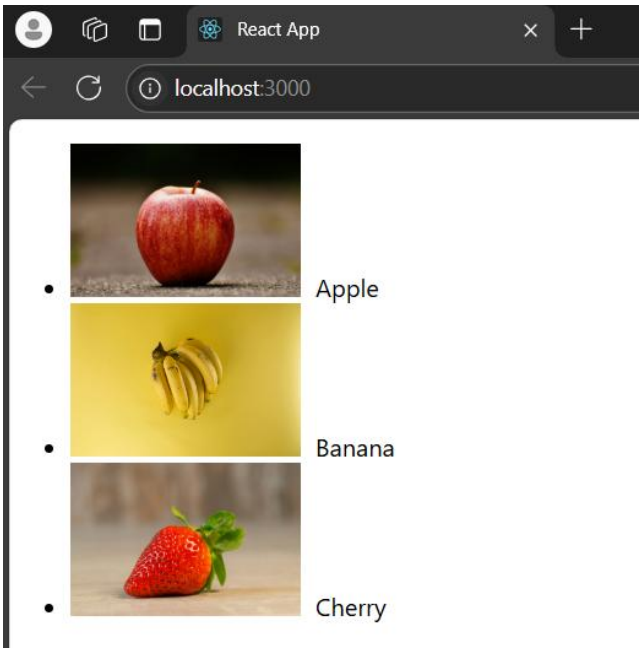
1. Create a component that displays an unordered list (<ul>) of 3 favorite fruits.

```
import React from "react";
import "./App.css";
import FavoriteFruits from "./Task3/FavoriteFruits";
function App() {
  return (
    <FavoriteFruits/>
  );
}
export default App;
```



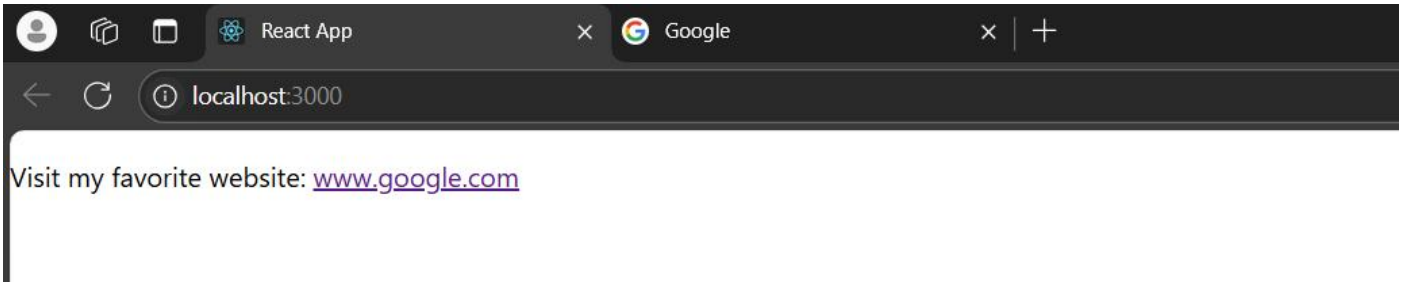
2. Update the above component to display a picture (<img>) of each fruit next to its name. (Use hardcoded image URLs for now.)

```
import React from 'react';
function FavoriteFruits() {
    return (
        <ul>
            <li>
                <img
                    src="apple.jpg"
                    alt="Apple"
                    style={{ marginRight: '10px',height:100,width:150 }}
                />
                Apple
            </li>
            <li>
                <img
                    src="banana.jpg"
                    alt="Banana"
                    style={{ marginRight: '10px' ,height:100,width:150 }}
                />
                Banana
            </li>
            <li>
                <img
                    src="strawberry.jpg"
                    alt="Strawberry"
                    style={{ marginRight: '10px',height:100,width:150  }}
                />
                Cherry
            </li>
        </ul>
    );
}
export default FavoriteFruits;
```



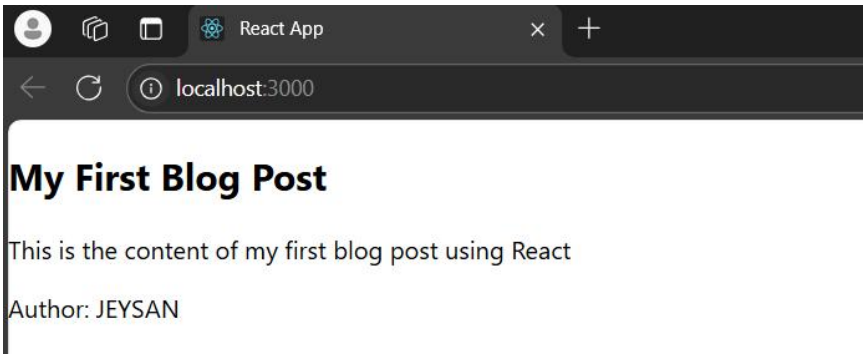
3. Create a component WebsiteLink that displays a hardcoded URL in an anchor (<a>) tag.

```
import React from 'react';
function WebsiteLink() {
    return (
        <p>
            Visit my favorite website:{" "}
            <a href="https://www.google.com" target="_blank" rel="noopener noreferrer">
                www.google.com
            </a>
        </p>
    );
}
export default WebsiteLink;
```



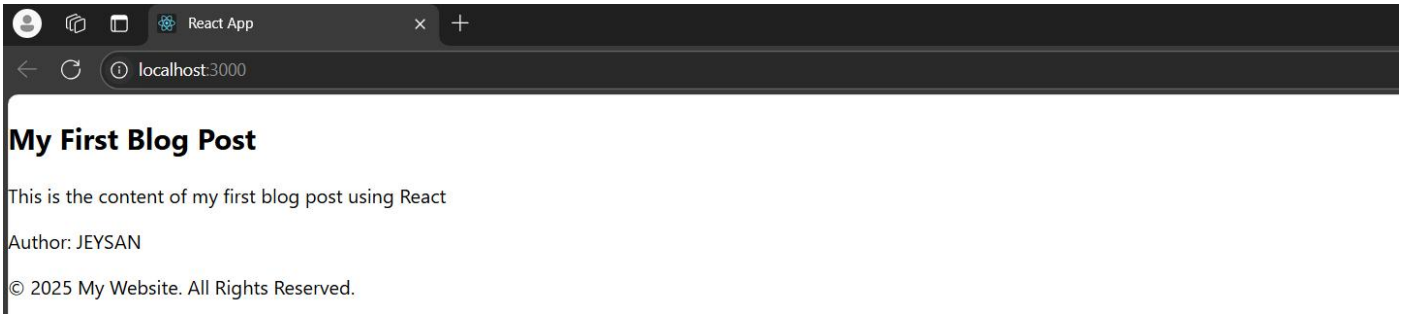
4. Make a JSX component that mimics a simple blog post with a title, content, and author. (All hardcoded.)

```jsx
import React from 'react';
function BlogPost() {
    return (
        <div>
            <h2>My First Blog Post</h2>
            <p>
                This is the content of my first blog post using React
            </p>
            <p>Author: JEYSAN</p>
        </div>
    );
}
export default BlogPost;
```



5. Design a Footer component with hardcoded copyright information using JSX.

```jsx
import React from 'react';
function Footer() {
    return (
        <footer>
            <p>© 2025 My Website. All Rights Reserved.</p>
        </footer>
    );
}
export default Footer;
```



4. JavaScript in JSX with Curly Braces

Tasks:

1. Display today's date in a component using the JavaScript Date object.

```
import React from 'react';
function CurrentDate() {
    const today = new Date().toLocaleDateString();
    return <p>Today's Date: {today}</p>;
}
export default CurrentDate;
```

2. Create a component that displays a random quote from a hardcoded list of quotes.

```
import React from 'react';
function RandomQuote() {
    const quotes = [
        "The only limit to our realization of tomorrow is our doubts of today.",
        "Do what you can, with what you have, where you are.",
        "Success is not final, failure is not fatal: It is the courage to continue that counts.",
    ];
    const randomQuote = quotes[Math.floor(Math.random() * quotes.length)];
    return <p>Random Quote: "{randomQuote}"</p>;
}
export default RandomQuote;
```

3. Write a component called MathResult that displays the result of a simple arithmetic operation (e.g., addition) of two hardcoded numbers.

```
import React from 'react';
function MathResult() {
    const number1 = 15;
    const number2 = 25;
    const sum = number1 + number2;
    return (
        <p>
            The sum of {number1} and {number2} is {sum}.
        </p>
    );
}
export default MathResult;
```

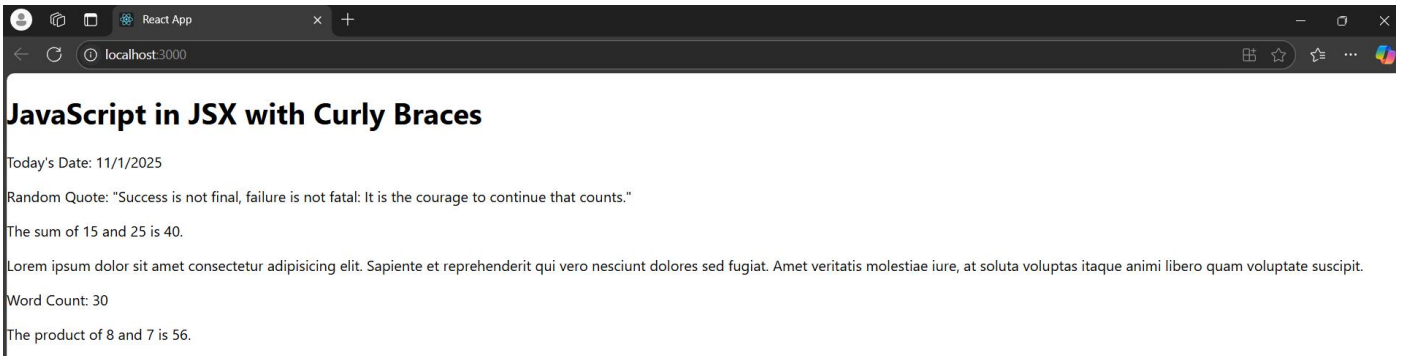
4. Create a component that displays the word count of a hardcoded paragraph.

```
import React from 'react';
function WordCount() {
    const paragraph ="Lorem ipsum dolor sit amet consectetur adipisicing elit. Sapiente et reprehenderit qui vero
nesciunt dolores sed fugiat. Amet veritatis molestiae iure, at soluta voluptas itaque animi libero quam voluptate
suscipit.";
    const wordCount = paragraph.split(' ').length;
    return (
        <div>
            <p>{paragraph}</p>
            <p>Word Count: {wordCount}</p>
        </div>
    );
}
export default WordCount;
```

5. Create a component that calculates and displays the product of two hardcoded numbers.

```
import React from 'react';

function CalculateProduct() {
    const number1 = 8;
    const number2 = 7;
    const product = number1 * number2;
    return (
        <p>
            The product of {number1} and {number2} is {product}.
        </p>
    );
}
export default CalculateProduct;
```

```
import React from 'react';
import CurrentDate from './Task4/CurrentDate';
import RandomQuote from './Task4/RandomQuote';
import MathResult from './Task4/MathResult';
import WordCount from './Task4/WordCount';
import CalculateProduct from './Task4/CaclulateProduct';
function App() {
return (
        <div>
            <h1>JavaScript in JSX with Curly Braces</h1>
            <CurrentDate />
            <RandomQuote />
            <MathResult />
            <WordCount />
            <CalculateProduct />
        </div>
    );
}
export default App;
```

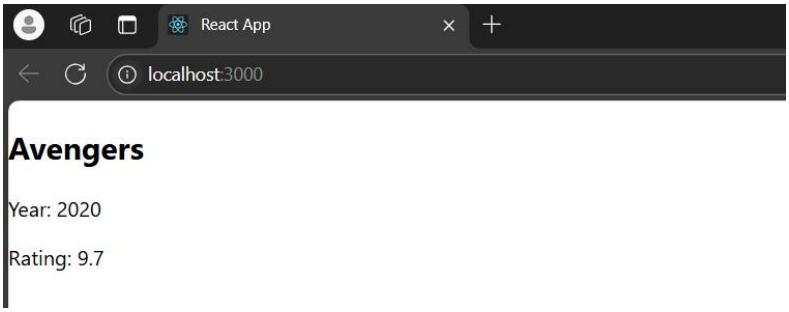


5. Passing Props to a Component

Tasks:

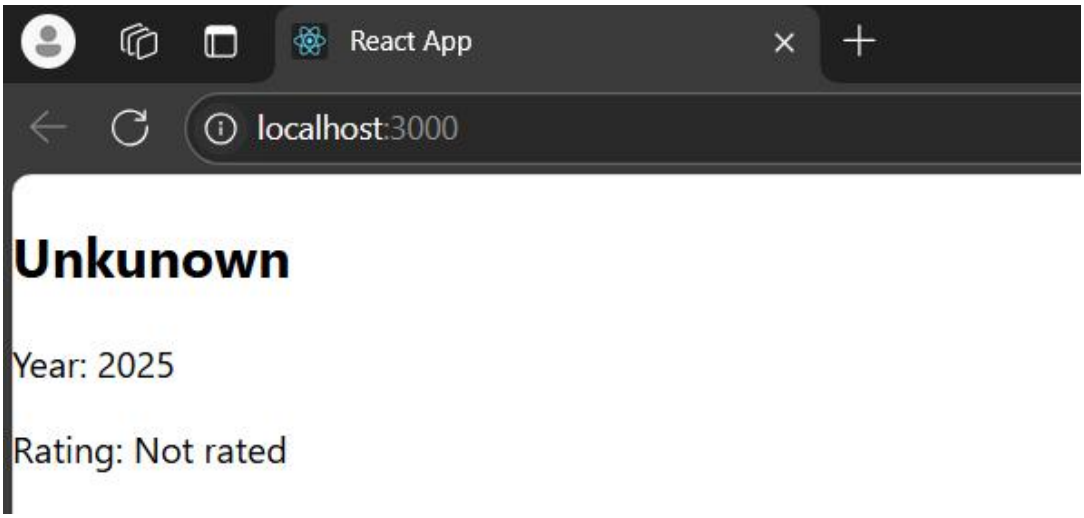
1. Create a Movie component that displays the title, year, and rating of a movie using props.

```
import React from 'react';
function Movie({ title, year, rating }) {
    return (
        <div>
            <h2>{title}</h2>
            <p>Year: {year}</p>
            <p>Rating: {rating}</p>
        </div>
    );
}
export default Movie;
```



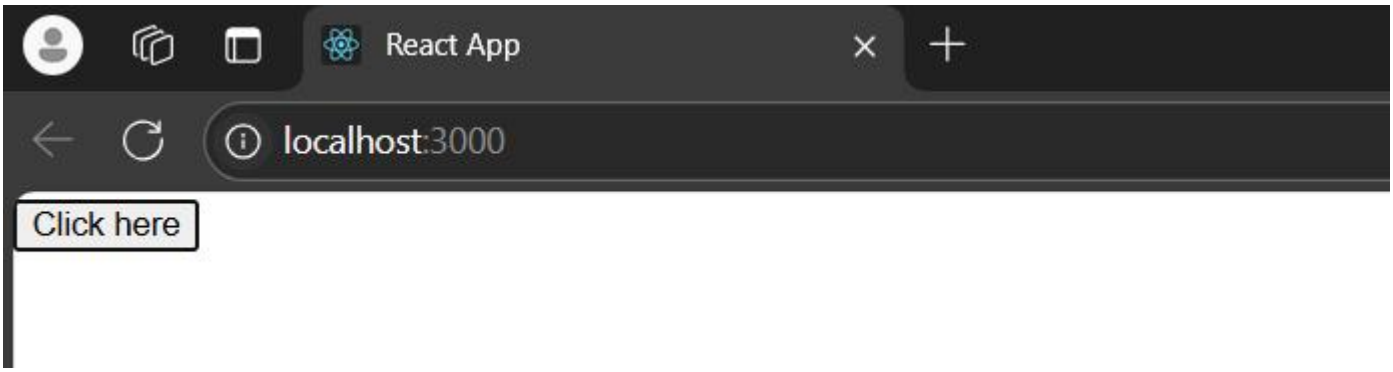
2. Update the Movie component to have a default prop for rating as “Not Rated”.

```
import React from 'react';
import Movie from './Task5/Movie';
function App() {
return (
        <Movie title="Unkunown" year={2025}/>
    );
}
export default App;
```



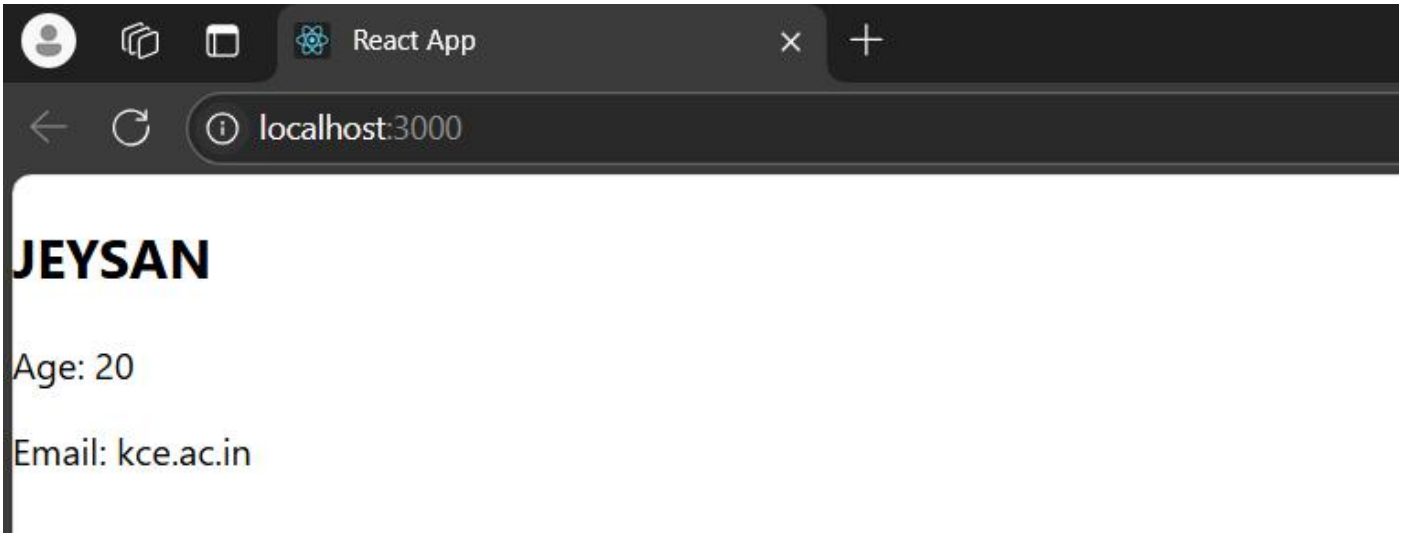
3. Design a Button component that takes in a label prop and displays the label on the button.

```
import React from 'react';
import Button from './Task5/Button';
function App() {
return (
        <Button label= "Click here"/>
    );
}
export default App;
```



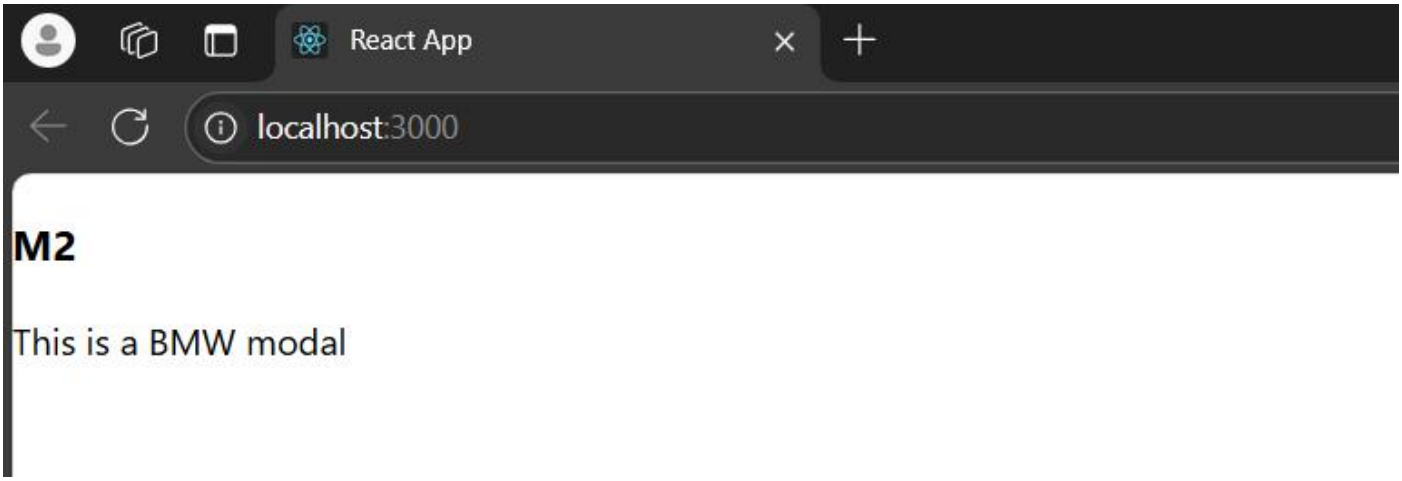
4. Make a UserProfile component and pass an object containing user details as props and display them.

```
import React from 'react';
function UserProfile({ user }) {
    return (
        <div>
            <h2>{user.name}</h2>
            <p>Age: {user.age}</p>
            <p>Email: {user.email}</p>
        </div>
    );
}
export default UserProfile;
```



5. Develop a Modal component that accepts and displays a title and some content passed as props.

```
import React from 'react';
function Modal({ title, content }) {
    return (
        <div>
            <h3>{title}</h3>
            <p>{content}</p>
        </div>
    );
}
export default Modal;
```



6. Conditional Rendering
Tasks:

1. Design a UserStatus component that displays “Online” or “Offline” based on a isOnline prop.

```
import React from 'react';
function UserStatus({ isOnline }) {
    return <p>{isOnline ? "Online" : "Offline"}</p>;
}
export default UserStatus;
```

2. Write a component AgeCheck that displays “Adult” or “Minor” based on an age prop.

```
import React from 'react';
function AgeCheck({ age }) {
    return <p>{age >= 18 ? "Adult" : "Minor"}</p>;
}
export default AgeCheck;
```

3. Create a Loading component that either displays “Loading…” or content based on a isLoading prop.

```
import React from 'react';
function Loading({ isLoading, children }) {
    return <div>{isLoading ? "Loading…" : children}</div>;
}
export default Loading;
```

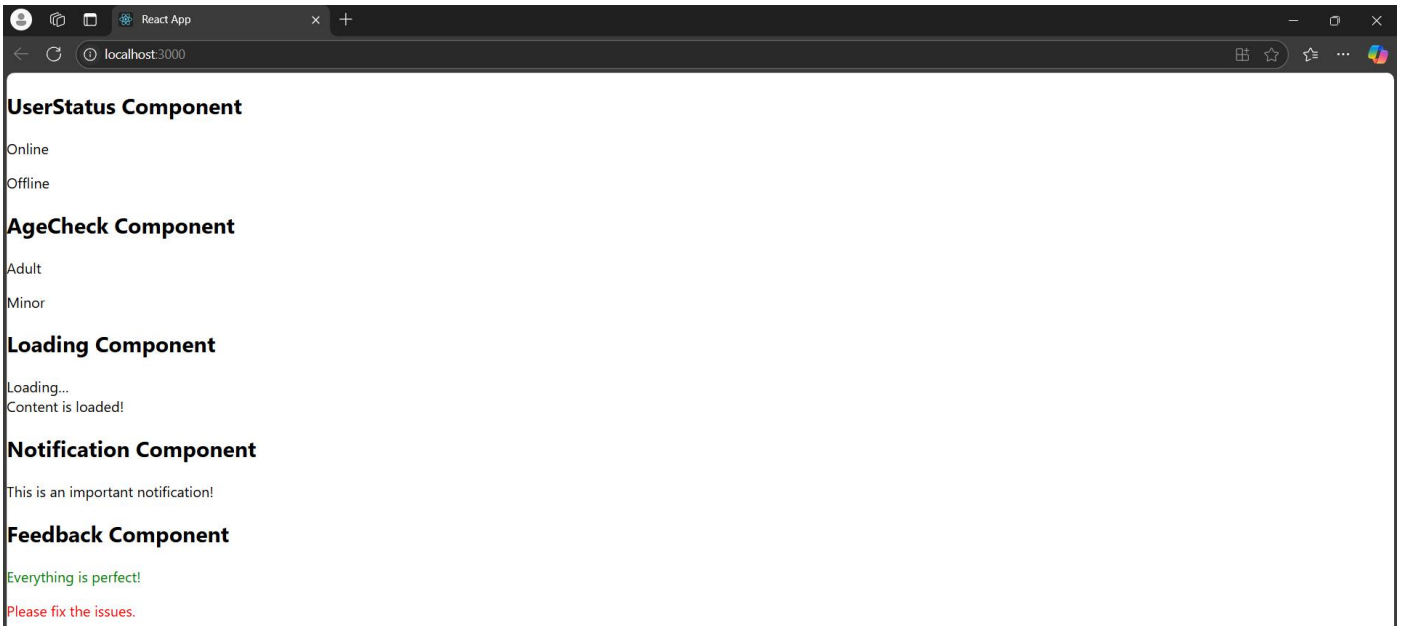
4. Make a Notification component that conditionally displays a message if a message prop is provided.

```
import React from 'react';
function Notification({ message }) {
    return message ? <p>{message}</p> : null;
}
export default Notification;
```

5. Design a Feedback component that displays feedback in either green (positive) or red (negative) based on a type prop.

```
import React from 'react';
function Feedback({ type, message }) {
    const style = {
        color: type === "positive" ? "green" : "red",
    };
    return <p style={style}>{message}</p>;
}
export default Feedback;
```

```
import React from 'react';
import UserStatus from './Task6/UserStatus';
import AgeCheck from './Task6/AgeCheck';
import Loading from './Task6/Loading';
import Notification from './Task6/Notification';
import Feedback from './Task6/FeedBack';

function App() {
    return (
        <div>
            <h2>UserStatus Component</h2>
            <UserStatus isOnline={true} />
            <UserStatus isOnline={false} />
            <h2>AgeCheck Component</h2>
            <AgeCheck age={20} />
            <AgeCheck age={16} />
            <h2>Loading Component</h2>
            <Loading isLoading={true} />
            <Loading isLoading={false}>Content is loaded!</Loading>
            <h2>Notification Component</h2>
            <Notification message="This is an important notification!" />
            <Notification message="" />
            <h2>Feedback Component</h2>
            <Feedback type="positive" message="Everything is perfect!" />
            <Feedback type="negative" message="Please fix the issues." />
        </div>
    );
}
export default App;
```

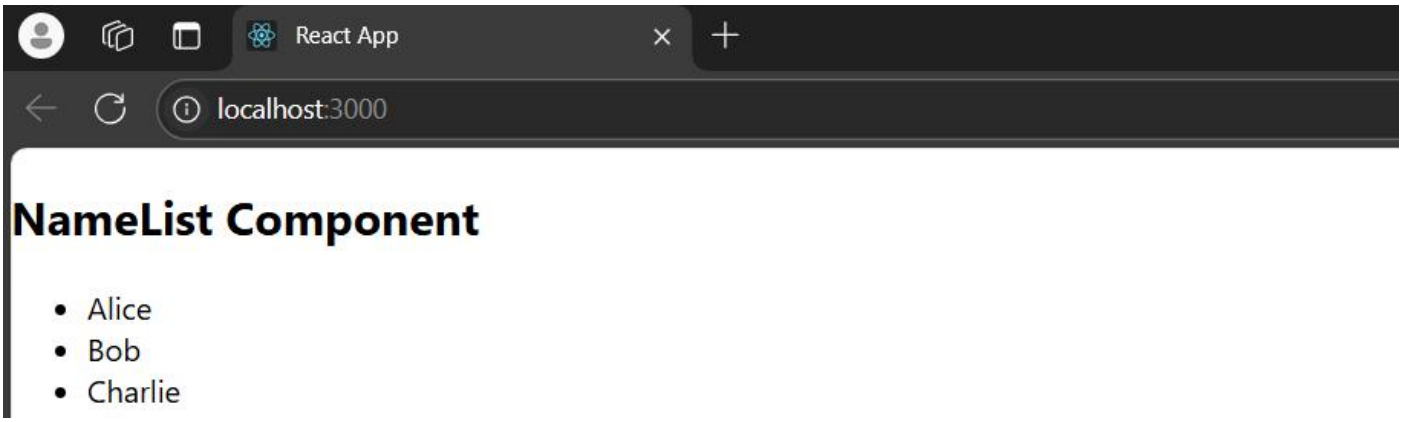


7. Rendering Lists

Tasks:

1. Write a component that takes an array of names as a prop and displays them in a list.

```
import React from 'react';
function NameList({ names }) {
    return (
        <ul>
            {names.map((name, index) => (
                <li key={index}>{name}</li>
            ))}
        </ul>
    );
}
export default NameList;
```

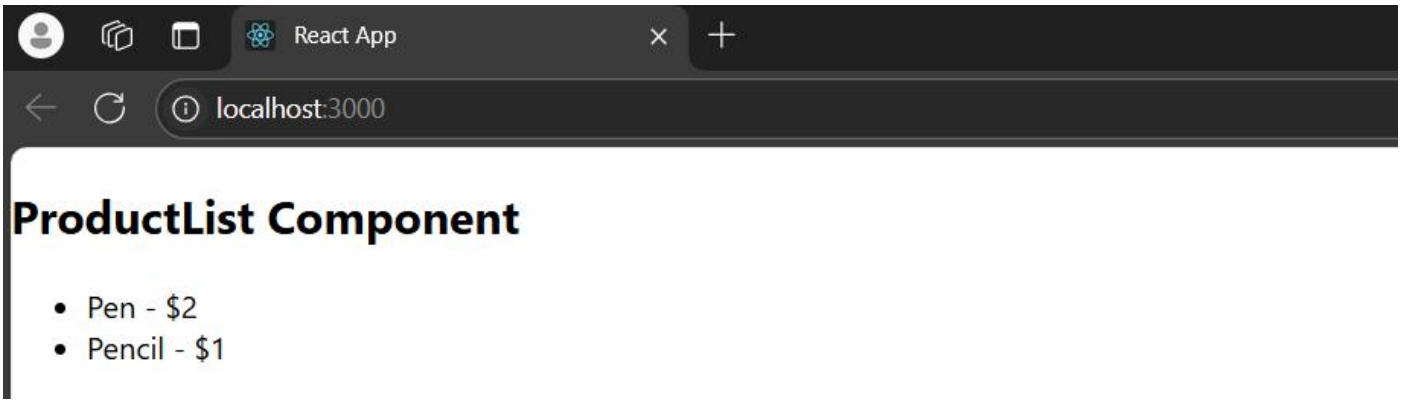


2. Create a TodoList component that displays a list of tasks and marks the completed ones.

```
import React from 'react';
function TodoList({ tasks }) {
    return (
        <ul>
            {tasks.map((task, index) => (
                <li key={index} style={{ textDecoration: task.completed ? 'line-through' : 'none' }}>
                    {task.text}
                </li>
            ))}
        </ul>
    );
}
export default TodoList;
```

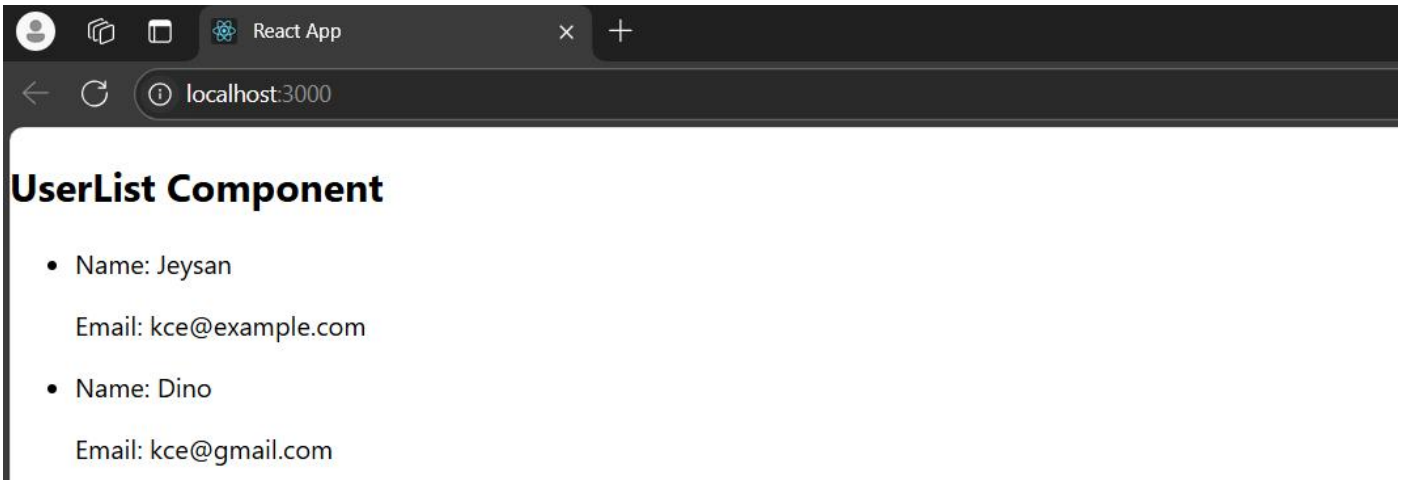
3. Design a ProductList component that only displays products with a price less than $10 using the filter() method.

```
import React from 'react';
function ProductList({ products }) {
    const affordableProducts = products.filter(product => product.price < 10);
    return (
        <ul>
            {affordableProducts.map((product, index) => (
                <li key={index}>
                    {product.name} - ${product.price}
                </li>
            ))}
        </ul>
    );
}
export default ProductList;
```



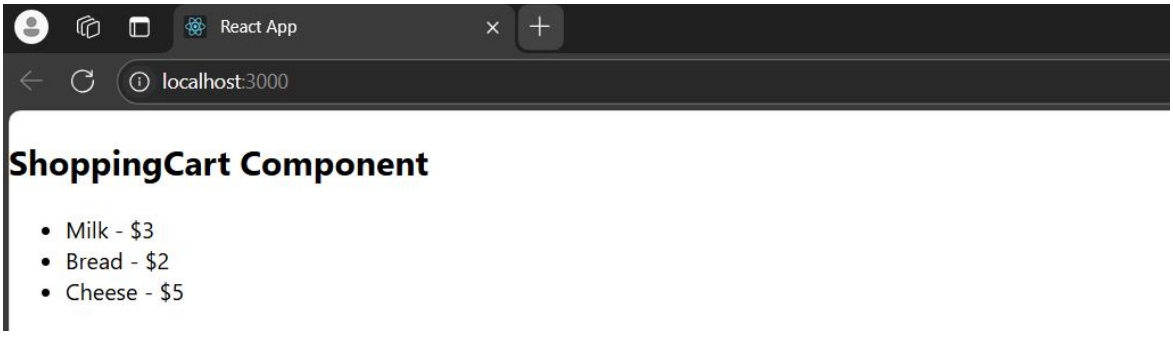
4. Make a UserList component that takes an array of user objects and displays their names and emails.

```
import React from 'react';
function UserList({ users }) {
    return (
        <ul>
            {users.map((user, index) => (
                <li key={index}>
                    <p>Name: {user.name}</p>
                    <p>Email: {user.email}</p>
                </li>
            ))}
        </ul>
    );
}
export default UserList;
```



5. Create a ShoppingCart component that displays a list of items and their prices. Ensure each item has a unique key.

```
import React from 'react';
function ShoppingCart({ items }) {
    return (
        <ul>
            {items.map(item => (
                <li key={item.id}>
                    {item.name} - ${item.price}
                </li>
            ))}
        </ul>
    );
}
export default ShoppingCart;
```

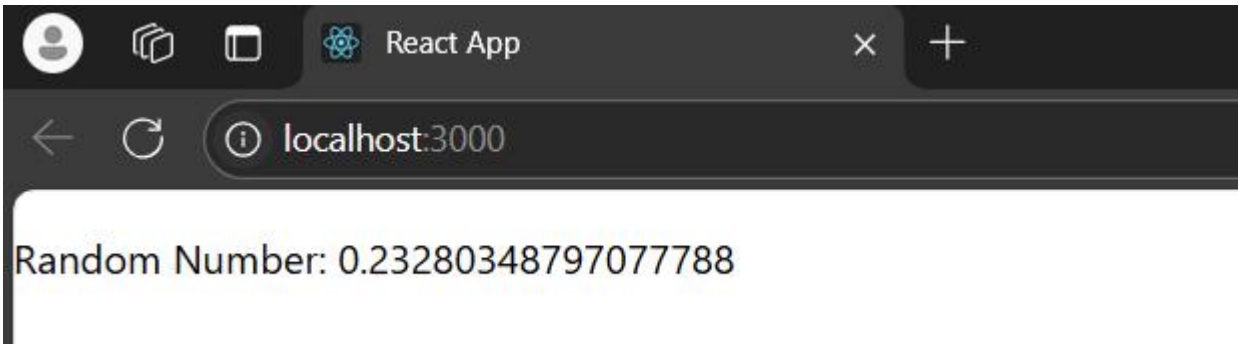


8. Keeping Components Pure

Tasks:

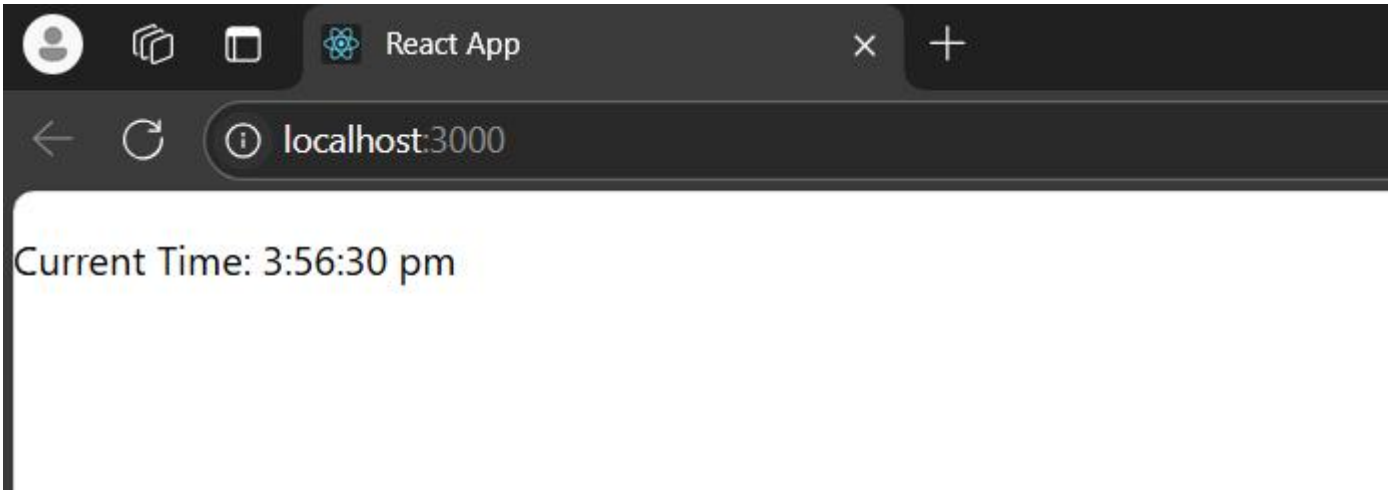
1. Convert an impure component that uses Math.random() within the render phase to a pure one.

```
import React from "react";
function ImpureComponent() {
    return <p>Random Number: {Math.random()}</p>;
}

export default ImpureComponent;
```



2. Create a pure component Clock that displays the current time and updates every second without causing side-effects during the render phase.

```
import React from 'react';
import Clock from './Task8/Clock';
function App() {
    return(
        <>
          <Clock/>
        </>
    );
}
export default App;
```

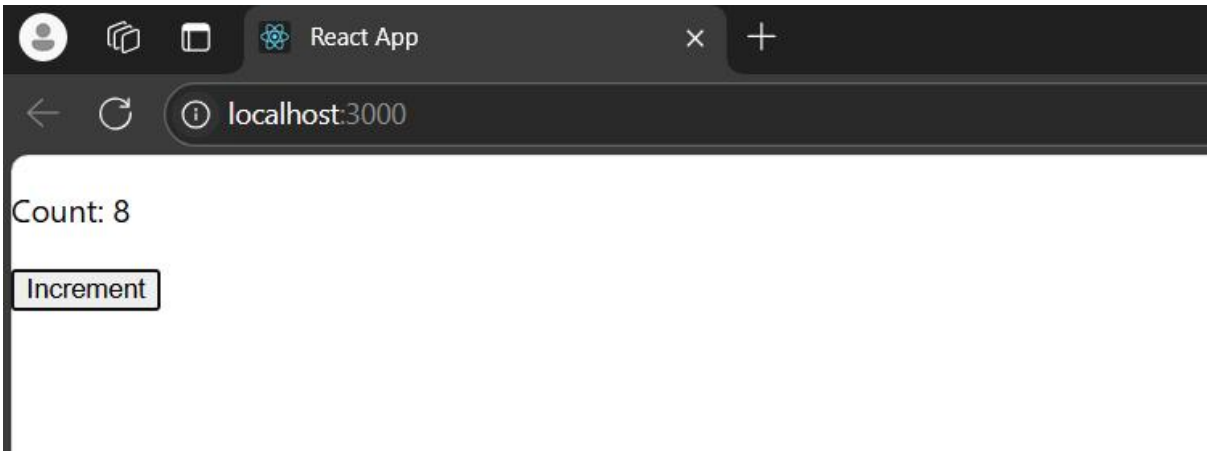


3. Use Strict Mode in an existing application and identify any warnings in the console.

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

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

4. Convert a class-based component with side effects in its lifecycle methods to a pure functional component using hooks.

```
import React, { useState, useEffect } from 'react';
function FunctionalComponent() {
    const [count, setCount] = useState(0);
    useEffect(() => {
        console.log('Component Mounted or Updated');
    }, [count]);
    return (
        <div>
            <p>Count: {count}</p>
            <button onClick={() => setCount(count + 1)}>Increment</button>
        </div>
    );
}
export default FunctionalComponent;
```



5. Make a pure ProfilePic component that takes a user ID as a prop and fetches the user's profile picture URL from an array without side-effects during rendering.

```jsx
import React from 'react';
const profiles = [
    { id: 1, url: 'apple.jpg' },
    { id: 2, url: 'banana.jpg' },
    { id: 3, url: 'strawberry.jpg' },
];
function ProfilePic({ userId }) {
    const profile = profiles.find(profile => profile.id === userId);
    const profilePicUrl = profile ? profile.url : 'https://example.com/default.jpg';

    return <img src={profilePicUrl} alt="Profile" height={100} width={150} />;
}
export default ProfilePic;
```