***Cognizant Java FSE – (Deep Skilling)***

***(WEEK-7)***

# **MODULE: React**

*Submitted by*

Name: S Subhashri

Email: [22102152@rmd.ac.in](mailto:22102152@rmd.ac.in)

College: RMD ENGINEERING COLLEGE

Batch: Java FSE – 2026

Superset ID: 6397078

**Exercise 9 :**

**Create a React Application named “cricketapp”**

Folder Structure

cricketapp/

├── src/

│ ├── components/

│ │ ├── ListofPlayers.js

│ │ └── IndianPlayers.js

│ ├── App.js

│ └── index.js

**ListofPlayers.js**

import React from 'react';

function ListofPlayers() {

const players = [

{ name: 'Virat Kohli', score: 95 },

{ name: 'Rohit Sharma', score: 65 },

{ name: 'Jasprit Bumrah', score: 85 },

{ name: 'Hardik Pandya', score: 60 },

{ name: 'Shubman Gill', score: 75 },

{ name: 'KL Rahul', score: 55 },

{ name: 'Ravindra Jadeja', score: 80 },

{ name: 'Shreyas Iyer', score: 45 },

{ name: 'Mohammed Siraj', score: 90 },

{ name: 'Kuldeep Yadav', score: 50 },

{ name: 'Axar Patel', score: 77 },

];

const highScorers = players.map((player, index) => (

<li key={index}>{player.name} - {player.score}</li>

));

const lowScorers = players

.filter(player => player.score < 70)

.map((player, index) => (

<li key={index}>{player.name} - {player.score}</li>

));

return (

<div>

<h2>All Players</h2>

<ul>{highScorers}</ul>

<h2>Players with Score below 70</h2>

<ul>{lowScorers}</ul>

</div>

);

}

export default ListofPlayers;

**IndianPlayers.js**

import React from 'react';

function IndianPlayers() {

const oddPlayers = ['Virat', 'Rohit', 'Bumrah', 'Hardik', 'Gill'];

const evenPlayers = ['Rahul', 'Jadeja', 'Iyer', 'Siraj', 'Kuldeep'];

const [firstOdd, secondOdd, ...restOdd] = oddPlayers;

const [firstEven, secondEven, ...restEven] = evenPlayers;

const T20Players = ['Virat', 'Rohit', 'Bumrah'];

const RanjiPlayers = ['Pujara', 'Saha'];

const allPlayers = [...T20Players, ...RanjiPlayers]; // Merge feature

return (

<div>

<h2>Odd Team Players (Destructured)</h2>

<p>1st: {firstOdd}, 2nd: {secondOdd}, Others: {restOdd.join(', ')}</p>

<h2>Even Team Players (Destructured)</h2>

<p>1st: {firstEven}, 2nd: {secondEven}, Others: {restEven.join(', ')}</p>

<h2>All Players (Merged)</h2>

<p>{allPlayers.join(', ')}</p>

</div>

);

}

export default IndianPlayers;

**App.js**

import React from 'react';

import ListofPlayers from './components/ListofPlayers';

import IndianPlayers from './components/IndianPlayers';

function App() {

const flag = true; // change to false to view IndianPlayers

return (

<div className="App">

<h1>Cricket App</h1>

{flag ? <ListofPlayers /> : <IndianPlayers />}

</div>

);

}

export default App;

**index.js**

import React from 'react';

import ReactDOM from 'react-dom';

import App from './App';

ReactDOM.render(

<App />,

document.getElementById('root')

);

OUTPUT  
A screenshot of a computer

AI-generated content may be incorrect.

**When flag=false**

A screenshot of a computer

AI-generated content may be incorrect.

**Exercise 10 :**

**Create a React Application named “officespacerentalapp” which uses React JSX to create elements, attributes and renders DOM to display the page.**

**1. Install and Setup React App**

npx create-react-app officespacerentalapp

cd officespacerentalapp

npm start

**2. App.js**

import React from 'react';

import './App.css'; // For optional styling

import officeImage from './office.jpg'

function App() {

const heading = "Office Space";

const imageJSX = <img src={officeImage} width="25%" height="25%" alt="Office Space" />;

const offices = [

{ Name: "DBS", Rent: 50000, Address: "Chennai" },

{ Name: "WeWork", Rent: 65000, Address: "Bangalore" },

{ Name: "Regus", Rent: 45000, Address: "Mumbai" }

];

return (

<div className="App" style={{ padding: "20px" }}>

<h1><b>{heading} , at Affordable Range</b></h1>

{offices.map((item, index) => {

const rentStyle = {

color: item.Rent <= 60000 ? 'red' : 'green'

};

return (

<div key={index} style={{ marginBottom: "40px" }}>

{imageJSX}

<h1><b>Name: {item.Name}</b></h1>

<h3 style={rentStyle}><b>Rent: Rs. {item.Rent}</b></h3>

<h3><b>Address: {item.Address}</b></h3>

</div>

);

})}

</div>

);

}

export default App;

**OUTPUT**  
A screenshot of a computer

AI-generated content may be incorrect.

**Exercise 11 :**

**Create a React Application “eventexamplesapp” to handle various events of the form elements in HTML.**

**App.js**

import React, { useState } from 'react';

import './App.css';

import CurrencyConverter from './CurrencyConverter';

function App() {

const [counter, setCounter] = useState(0);

const increment = () => {

setCounter(prev => prev + 1);

sayHello();

};

const decrement = () => {

setCounter(prev => prev - 1);

};

const sayHello = () => {

alert("Hello! Member1");

};

const sayWelcome = (msg) => {

alert(msg);

};

const handlePress = () => {

alert("I was clicked");

};

return (

<div className="App">

<h2>{counter}</h2>

<button onClick={increment}>Increment</button>

<button onClick={decrement}>Decrement</button>

<br /><br />

<button onClick={() => sayWelcome("Hello! Member1")}>Say welcome</button>

<br /><br />

<button onClick={handlePress}>Click on me</button>

<br /><br />

<CurrencyConverter />

</div>

);

}

export default App;

**CurrencyConverter.js**

import React, { useState } from 'react';

function CurrencyConverter() {

const [euro, setEuro] = useState('');

const handleSubmit = (e) => {

e.preventDefault();

if (!euro || isNaN(euro)) {

alert("Please enter a valid amount in Euro.");

return;

}

const rupees = parseFloat(euro) \* 80; // 1 Euro = 80 INR

alert(`Converting to Euro Amount is ${rupees}`);

};

return (

<div>

<h2 style={{ color: "green" }}>Currency Converter!!!</h2>

<form onSubmit={handleSubmit}>

<label>Amount:</label>

<input

type="text"

value={euro}

onChange={(e) => setEuro(e.target.value)}

/>

<br /><br />

<label>Currency:</label>

<select>

<option>Euro</option>

</select>

<br /><br />

<button type="submit">Submit</button>

</form>

</div>

);

}

export default CurrencyConverter;

**OUTPUT**

1.Create “Increment” button to increase the value of the counter and “Decrement” button to decrease the value of the counter. The “Increase” button should invoke multiple methods.

* 1. To increment the value
  2. Say Hello followed by a static message

A screenshot of a computer

AI-generated content may be incorrect.

1. Create a button “Say Welcome” which invokes the function which takes “welcome” as an argument.

A screenshot of a computer

AI-generated content may be incorrect.

1. Create a button which invokes synthetic event “OnPress” which display “I was clicked”

A screenshot of a computer

AI-generated content may be incorrect.

4. Create a “CurrencyConvertor” component which will convert the Indian Rupees to Euro when the Convert button is clicked.

Handle the Click event of the button to invoke the handleSubmit event and handle the conversion of the euro to rupees.

A screenshot of a computer

AI-generated content may be incorrect.

**Exercise 12 :**

**Create a React Application named “ticketbookingapp” where the guest user can browse the page where the flight details are displayed whereas the logged in user only can book tickets.**

The Login and Logout buttons should accordingly display different pages. Once the user is logged in the User page should be displayed. When the user clicks on Logout, the Guest page should be displayed.

**Folder Structure**

├── src/

│ ├── App.js

│ ├── index.js

├── public/

├── package.json

**Create the React App**

**Open terminal and run:**

npx create-react-app ticketbookingapp

cd ticketbookingapp

code .

**App.js**

import React, { useState } from 'react';

function LoginButton({ onClick }) {

return <button onClick={onClick}>Login</button>;

}

function LogoutButton({ onClick }) {

return <button onClick={onClick}>Logout</button>;

}

function GuestGreeting() {

return <h2>Please sign up.</h2>;

}

function UserGreeting() {

return <h2>Welcome back.</h2>;

}

function Greeting({ isLoggedIn }) {

return isLoggedIn ? <UserGreeting /> : <GuestGreeting />;

}

function App() {

const [isLoggedIn, setIsLoggedIn] = useState(false);

const handleLoginClick = () => setIsLoggedIn(true);

const handleLogoutClick = () => setIsLoggedIn(false);

return (

<div style={{ padding: '20px', fontFamily: 'Arial' }}>

<Greeting isLoggedIn={isLoggedIn} />

{isLoggedIn ? (

<LogoutButton onClick={handleLogoutClick} />

) : (

<LoginButton onClick={handleLoginClick} />

)}

</div>

);

}

export default App;

**Greeting.js**

import React from "react";

function UserGreeting() {

return <h1>Welcome back</h1>;

}

function GuestGreeting() {

return <h1>Please sign up.</h1>;

}

function Greeting(props) {

const isLoggedIn = props.isLoggedIn;

if (isLoggedIn) {

return <UserGreeting />;

}

return <GuestGreeting />;

}

export default Greeting;

**LoginControl.js**

import React from "react";

import Greeting from "./Greeting";

function LoginButton(props) {

return <button onClick={props.onClick}>Login</button>;

}

function LogoutButton(props) {

return <button onClick={props.onClick}>Logout</button>;

}

class LoginControl extends React.Component {

constructor(props) {

super(props);

this.handleLoginClick = this.handleLoginClick.bind(this);

this.handleLogoutClick = this.handleLogoutClick.bind(this);

this.state = { isLoggedIn: false };

}

handleLoginClick() {

this.setState({ isLoggedIn: true });

}

handleLogoutClick() {

this.setState({ isLoggedIn: false });

}

render() {

const isLoggedIn = this.state.isLoggedIn;

let button;

if (isLoggedIn) {

button = <LogoutButton onClick={this.handleLogoutClick} />;

} else {

button = <LoginButton onClick={this.handleLoginClick} />;

}

return (

<div>

<Greeting isLoggedIn={isLoggedIn} />

{button}

</div>

);

}

}

export default LoginControl;

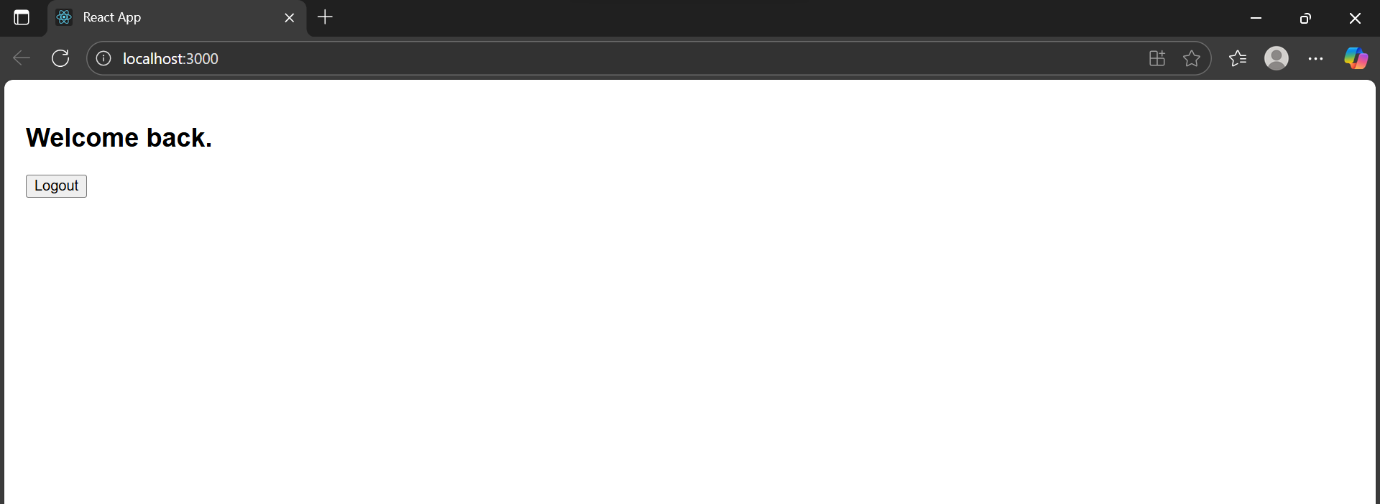
**OUTPUT**

**login output**

A screen shot of a computer

AI-generated content may be incorrect.

**logout output**



**Exercise 13 :**

**Create a React App named “bloggerapp” in with 3 components.**

1. Book Details
2. Blog Details
3. Course Details

**1.Setup:**

Open terminal and run:

npx create-react-app bloggerapp

cd bloggerapp

npm start

**2. App.js**

import React, { useState } from 'react';

import CourseDetails from './CourseDetails';

import BookDetails from './BookDetails';

import BlogDetails from './BlogDetails';

function App() {

const [showCourses, setShowCourses] = useState(true);

const [showBooks, setShowBooks] = useState(true);

const [showBlogs, setShowBlogs] = useState(true);

return (

<div style={{ display: 'flex', justifyContent: 'space-around', padding: '30px' }}>

{showCourses && <CourseDetails />}

{showBooks && <BookDetails />}

{showBlogs && <BlogDetails />}

</div>

);

}

export default App;

**3. CourseDetails.js**

import React from 'react';

const courses = [

{ name: "Angular", date: "4/5/2021" },

{ name: "React", date: "6/3/20201" }

];

function CourseDetails() {

return (

<div style={{ borderRight: '4px solid green', padding: '0 20px' }}>

<h2>Course Details</h2>

{courses.map((course, index) => (

<div key={index}>

<h3>{course.name}</h3>

<p>{course.date}</p>

</div>

))}

</div>

);

}

export default CourseDetails;

**4. BookDetails.js**

import React from 'react';

const books = [

{ title: "Master React", price: 670 },

{ title: "Deep Dive into Angular 11", price: 800 },

{ title: "Mongo Essentials", price: 450 }

];

function BookDetails() {

return (

<div style={{ borderRight: '4px solid green', padding: '0 20px' }}>

<h2>Book Details</h2>

{books.map((book, index) => (

<div key={index}>

<h3>{book.title}</h3>

<p>{book.price}</p>

</div>

))}

</div>

);

}

export default BookDetails;

**5. BlogDetails.js**

import React from 'react';

const blogs = [

{

title: "React Learning",

author: "Stephen Biz",

content: "Welcome to learning React!"

},

{

title: "Installation",

author: "Schwezdenier",

content: "You can install React from npm."

}

];

function BlogDetails() {

return (

<div style={{ padding: '0 20px' }}>

<h2>Blog Details</h2>

{blogs.map((blog, index) => (

<div key={index}>

<h3>{blog.title}</h3>

<p><strong>{blog.author}</strong></p>

<p>{blog.content}</p>

</div>

))}

</div>

);

}

export default BlogDetails;

**6. How it works:**

* App.js conditionally renders 3 components.
* Each component uses .map() and key for rendering lists.
* Code is modular, simple, and reflects best practices for reusable components.

**OUTPUT**

A screenshot of a computer

AI-generated content may be incorrect.