**Week-7 Hands-On Solutions**

**App 1 — officespacerentalapp (JSX & inline CSS demonstration)**

Folder: officespacerentalapp/src

Files: index.js, App.js, OfficeList.js, index.css

Description: Demonstrates JSX, inline CSS, looping through objects and conditional inline styles (rent color).

src/index.js

import React from 'react';

import ReactDOM from 'react-dom/client';

import App from './App';

import './index.css';

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

root.render(<App />);

src/App.js

import React from 'react';

import OfficeList from './OfficeList';

const App = () => {

return (

<div style={{ padding: 20, fontFamily: 'Arial, sans-serif' }}>

<h1>Office Space Rental</h1>

<OfficeList />

</div>

);

};

export default App;

src/OfficeList.js

import React from 'react';

const offices = [

{ id: 1, name: 'Apex Tower', rent: 55000, address: 'MG Road, Bangalore' },

{ id: 2, name: 'Skyline Center', rent: 75000, address: 'Connaught Place, Delhi' },

{ id: 3, name: 'Harbour Plaza', rent: 45000, address: 'Marine Drive, Mumbai' },

];

const OfficeList = () => {

return (

<div>

<h2>Available Offices</h2>

<ul>

{offices.map((office) => {

const rentStyle = { color: office.rent < 60000 ? 'red' : 'green', fontWeight: 'bold' };

return (

<li key={office.id} style={{ marginBottom: 10 }}>

<div><strong>{office.name}</strong></div>

<div>Address: {office.address}</div>

<div>Rent: <span style={rentStyle}>₹{office.rent.toLocaleString()}</span></div>

</li>

);

})}

</ul>

</div>

);

};

export default OfficeList;

**App 2 — eventexamplesapp (Events, synthetic events & currency converter)**

Folder: eventexamplesapp/src

Files: index.js, App.js, Counter.js, CurrencyConverter.js

Description: Demonstrates React event handling, synthetic events, using this-like patterns via hooks, passing arguments to event handlers and a simple currency converter.

src/index.js

import React from 'react';

import ReactDOM from 'react-dom/client';

import App from './App';

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

root.render(<App />);

src/App.js

import React from 'react';

import Counter from './Counter';

import CurrencyConverter from './CurrencyConverter';

const App = () => {

return (

<div style={{ padding: 20 }}>

<h1>Event Examples App</h1>

<Counter />

<hr />

<CurrencyConverter />

</div>

);

};

export default App;

src/Counter.js

import React, { useState } from 'react';

const Counter = () => {

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

const [message, setMessage] = useState('');

const sayHello = (extra) => {

// multiple methods example - function called from event handler

setMessage(`Hello - ${extra}`);

};

const handleIncrement = () => {

setCount(prev => prev + 1);

};

const handleIncrementMultiple = () => {

// Invoke multiple functions from a single click handler

handleIncrement();

sayHello('Static message');

};

const handleDecrement = () => {

setCount(prev => prev - 1);

};

const handleWelcome = (text) => {

setMessage(text);

};

const handleSynthetic = (e) => {

// React synthetic event

setMessage('I was clicked');

};

return (

<div>

<h2>Counter</h2>

<div>Value: {count}</div>

<div style={{ marginTop: 8 }}>

<button onClick={handleIncrementMultiple}>Increase (calls multiple methods)</button>{' '}

<button onClick={handleDecrement}>Decrease</button>{' '}

<button onClick={() => handleWelcome('Welcome')}>Say Welcome</button>{' '}

<button onClick={handleSynthetic} onKeyPress={handleSynthetic}>Synthetic Event</button>

</div>

<div style={{ marginTop: 10 }}>Message: {message}</div>

</div>

);

};

export default Counter;

src/CurrencyConverter.js

import React, { useState } from 'react';

const CurrencyConverter = () => {

const [rupees, setRupees] = useState('');

const [result, setResult] = useState(null);

// example conversion rate; update as needed

const INR\_TO\_EUR = 0.011; // 1 INR = 0.011 EUR ~ (approx 1/90)

const handleSubmit = (e) => {

e.preventDefault();

const r = parseFloat(rupees);

if (isNaN(r)) {

setResult('Please enter a valid number');

return;

}

setResult((r \* INR\_TO\_EUR).toFixed(2) + ' EUR');

};

return (

<div style={{ marginTop: 20 }}>

<h2>Currency Converter (INR → EUR)</h2>

<form onSubmit={handleSubmit}>

<input

type="text"

value={rupees}

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

placeholder="Enter amount in INR"

/>

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

</form>

<div style={{ marginTop: 8 }}>Result: {result}</div>

</div>

);

};

export default CurrencyConverter;

**App 3 — ticketbookingapp (Conditional rendering & element variables)**

Folder: ticketbookingapp/src

Files: index.js, App.js, GuestPage.js, UserPage.js

Description: Guest user can browse flights; logged-in user can book tickets. Login/Logout toggles pages. Demonstrates element variables and preventing rendering (returning null).

src/index.js

import React from 'react';

import ReactDOM from 'react-dom/client';

import App from './App';

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

root.render(<App />);

src/App.js

import React, { useState } from 'react';

import GuestPage from './GuestPage';

import UserPage from './UserPage';

const App = () => {

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

return (

<div style={{ padding: 20 }}>

<h1>Ticket Booking</h1>

<div style={{ marginBottom: 10 }}>

{isLoggedIn ? (

<button onClick={() => setIsLoggedIn(false)}>Logout</button>

) : (

<button onClick={() => setIsLoggedIn(true)}>Login</button>

)}

</div>

{/\* Element variable example \*/}

{isLoggedIn ? <UserPage /> : <GuestPage />}

</div>

);

};

export default App;

src/GuestPage.js

import React from 'react';

const flights = [

{ id: 1, from: 'Bangalore', to: 'Mumbai', fare: 3500 },

{ id: 2, from: 'Delhi', to: 'Chennai', fare: 4200 },

];

const GuestPage = () => {

return (

<div>

<h2>Guest View - Browse Flights</h2>

<ul>

{flights.map(f => (

<li key={f.id}>{f.from} → {f.to} — ₹{f.fare}</li>

))}

</ul>

<p>Please log in to book tickets.</p>

</div>

);

};

export default GuestPage;

src/UserPage.js

import React from 'react';

const UserPage = () => {

return (

<div>

<h2>User View - Book Tickets</h2>

<p>Select a flight and proceed to book (example placeholder).</p>

<button>Book Now</button>

</div>

);

};

export default UserPage;

**App 4 — bloggerapp (Multiple conditional rendering approaches)**

Folder: bloggerapp/src

Files: index.js, App.js, BookDetails.js, BlogDetails.js, CourseDetails.js

Description: Demonstrates several ways to conditionally render multiple components: if/else, ternary, logical AND, switch mapping.

src/index.js

import React from 'react';

import ReactDOM from 'react-dom/client';

import App from './App';

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

root.render(<App />);

src/App.js

import React, { useState } from 'react';

import BookDetails from './BookDetails';

import BlogDetails from './BlogDetails';

import CourseDetails from './CourseDetails';

const App = () => {

// change method to 'if', 'ternary', 'and', or 'switch' to see different styles

const [method, setMethod] = useState('if');

return (

<div style={{ padding: 20 }}>

<h1>Blogger App - Conditional Rendering Demo</h1>

<div style={{ marginBottom: 12 }}>

<label>Select method: </label>

<select value={method} onChange={e => setMethod(e.target.value)}>

<option value="if">If/Else</option>

<option value="ternary">Ternary</option>

<option value="and">Logical AND</option>

<option value="switch">Switch</option>

</select>

</div>

{/\* If / Else \*/}

{method === 'if' && (

<div>

<h3>If / Else Example</h3>

<BookDetails />

<BlogDetails />

</div>

)}

{/\* Ternary \*/}

{method === 'ternary' ? (

<div>

<h3>Ternary Example (Book or Course)</h3>

<BookDetails />

</div>

) : null}

{/\* Logical AND \*/}

{method === 'and' && (

<div>

<h3>Logical AND Example</h3>

<BlogDetails />

</div>

)}

{/\* Switch-style rendering using mapping \*/}

{method === 'switch' && (

<div>

<h3>Switch-style Example</h3>

<CourseDetails />

</div>

)}

</div>

);

};

export default App;

src/BookDetails.js

import React from 'react';

const BookDetails = () => {

return (

<div>

<h4>Book Details</h4>

<p>Title: Learning React</p>

<p>Author: Someone</p>

</div>

);

};

export default BookDetails;

src/BlogDetails.js

import React from 'react';

const BlogDetails = () => {

return (

<div>

<h4>Blog Details</h4>

<p>Recent Posts: React basics, Hooks, Routing</p>

</div>

);

};

export default BlogDetails;

src/CourseDetails.js

import React from 'react';

const CourseDetails = () => {

return (

<div>

<h4>Course Details</h4>

<p>Course: Frontend Development</p>

<p>Duration: 6 weeks</p>

</div>

);

};

export default CourseDetails;

**App 5 — cricketapp (ES6 features: map, arrow functions, destructuring & merge)**

Folder: cricketapp/src

Files: index.js, App.js, components/ListOfPlayers.js, components/IndianPlayers.js

Description: Create an array of 11 players (name & score) and demonstrate map(), filter (arrow functions), destructuring, and merging arrays with spread operator. Toggle display using a flag variable.

src/index.js

import React from 'react';

import ReactDOM from 'react-dom/client';

import App from './App';

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

root.render(<App />);

src/App.js

import React, { useState } from 'react';

import ListOfPlayers from './components/ListOfPlayers';

import IndianPlayers from './components/IndianPlayers';

const initialPlayers = [

{ name: 'Player 1', score: 85 },

{ name: 'Player 2', score: 45 },

{ name: 'Player 3', score: 78 },

{ name: 'Player 4', score: 66 },

{ name: 'Player 5', score: 92 },

{ name: 'Player 6', score: 30 },

{ name: 'Player 7', score: 74 },

{ name: 'Player 8', score: 59 },

{ name: 'Player 9', score: 81 },

{ name: 'Player 10', score: 68 },

{ name: 'Player 11', score: 99 },

];

const App = () => {

const [flag, setFlag] = useState(true); // Toggle between components

return (

<div style={{ padding: 20 }}>

<h1>Cricket App</h1>

<div style={{ marginBottom: 10 }}>

<button onClick={() => setFlag(true)}>Show ListOfPlayers</button>{' '}

<button onClick={() => setFlag(false)}>Show IndianPlayers</button>

</div>

{flag ? <ListOfPlayers players={initialPlayers} /> : <IndianPlayers players={initialPlayers} />}

</div>

);

};

export default App;

src/components/ListOfPlayers.js

import React from 'react';

const ListOfPlayers = ({ players }) => {

// Use map() to render list

return (

<div>

<h2>All Players</h2>

<ul>

{players.map((p, idx) => (

<li key={idx}>{p.name} — Score: {p.score}</li>

))}

</ul>

<h3>Players with scores below 70 (using arrow function & filter)</h3>

<ul>

{players.filter(p => p.score < 70).map((p, idx) => (

<li key={'f' + idx}>{p.name} — {p.score}</li>

))}

</ul>

<h3>Merging sample arrays (T20 + Ranji)</h3>

<div>

{(() => {

const t20 = ['Rahul', 'Rohit', 'Hardik'];

const ranji = ['Vijay', 'Azhar', 'Gautam'];

const merged = [...t20, ...ranji]; // spread operator (merge)

return (

<ul>

{merged.map((name, i) => <li key={i}>{name}</li>)}

</ul>

);

})()}

</div>

</div>

);

};

export default ListOfPlayers;

src/components/IndianPlayers.js

import React from 'react';

const IndianPlayers = ({ players }) => {

// Destructuring example: extract first two players and the rest

const [firstPlayer, secondPlayer, ...rest] = players;

// Using original index to separate odd/even team players (0-based index)

const evenTeam = players.filter((\_, i) => i % 2 === 0); // indices 0,2,4...

const oddTeam = players.filter((\_, i) => i % 2 === 1); // indices 1,3,5...

return (

<div>

<h2>Indian Players (Destructuring example)</h2>

<p><strong>Captain:</strong> {firstPlayer.name} — Score: {firstPlayer.score}</p>

<p><strong>Vice Captain:</strong> {secondPlayer.name} — Score: {secondPlayer.score}</p>

<h3>Even Team Players</h3>

<ul>

{evenTeam.map((p, i) => <li key={i}>{p.name} — {p.score}</li>)}

</ul>

<h3>Odd Team Players</h3>

<ul>

{oddTeam.map((p, i) => <li key={i}>{p.name} — {p.score}</li>)}

</ul>

</div>

);

};

export default IndianPlayers;