MANDATORY EXERCISES WEEK 7

## **Objectives**

* List the features of ES6
* Explain JavaScript let
* Identify the differences between var and let
* Explain JavaScript const
* Explain ES6 class fundamentals
* Explain ES6 class inheritance
* Define ES6 arrow functions
* Identify set(), map()

In this hands-on lab, you will learn how to:

* Use map() method of ES6
* Apply arrow functions of ES6
* Implement Destructuring features of ES6

## **Prerequisites**

The following is required to complete this hands-on lab:

* Node.js
* NPM
* Visual Studio Code

## **Notes**

Estimated time to complete this lab: **60 minutes.**

Create a React Application named “cricketapp” with the following components:

1. ListofPlayers

* Declare an array with 11 players and store details of their names and scores using the map feature of ES6



* Filter the players with scores below 70 using arrow functions of ES6.



1. IndianPlayers
   1. Display the Odd Team Player and Even Team players using the Destructuring features of ES6



* 1. Declare two arrays T20players and RanjiTrophy players and merge the two arrays and display them using the Merge feature of ES6



Display these two components in the same home page using a simple if else in the flag variable.

**Output:**

When Flag=true



When Flag=false



**Hint:**

CODE:

import React from 'react';

const ListofPlayers = () => {

const players = [

{ name: "Virat", score: 95 },

{ name: "Rohit", score: 65 },

{ name: "Dhoni", score: 80 },

{ name: "Ashwin", score: 50 },

{ name: "Hardik", score: 74 },

{ name: "Bumrah", score: 68 },

{ name: "Surya", score: 85 },

{ name: "Gill", score: 45 },

{ name: "Iyer", score: 76 },

{ name: "Pant", score: 55 },

{ name: "Jadeja", score: 90 },

];

// Use map

const allPlayers = players.map((p, index) => (

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

));

// Use arrow function + filter

const below70 = players.filter(p => p.score < 70).map((p, index) => (

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

));

return (

<div>

<h2>All Players</h2>

<ul>{allPlayers}</ul>

<h2>Players with Score &lt; 70</h2>

<ul>{below70}</ul>

</div>

);

};

export default ListofPlayers;

import React from 'react';

const IndianPlayers = () => {

const players = ["Virat", "Rohit", "Dhoni", "Ashwin", "Hardik", "Bumrah", "Surya", "Gill"];

// Destructuring

const oddTeam = players.filter((\_, i) => i % 2 !== 0);

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

// Merge arrays

const T20players = ["Samson", "Kishan"];

const RanjiTrophy = ["Shaw", "Pujara"];

const mergedPlayers = [...T20players, ...RanjiTrophy];

return (

<div>

<h2>Odd Team</h2>

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

<h2>Even Team</h2>

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

<h2>Merged Players (T20 + Ranji)</h2>

<ul>{mergedPlayers.map((p, i) => <li key={i}>{p}</li>)}</ul>

</div>

);

};

export default IndianPlayers;

import React from 'react';

import ListofPlayers from './components/ListofPlayers';

import IndianPlayers from './components/IndianPlayers';

function App() {

const flag = true; // Change this to false to test the other component

return (

<div className="App">

<h1>🏏 Cricket Dashboard</h1>

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

</div>

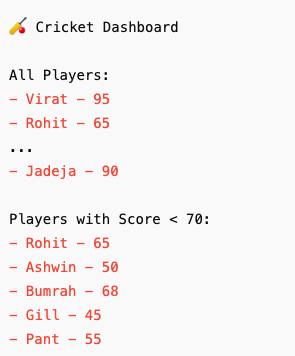
);

}

export default App;

OUTPUT:

FLAG=TRUE



FLAG=FALSE



## **Objectives**

* Define JSX
* Explain about ECMA Script
* Explain React.createElement()
* Explain how to create React nodes with JSX
* Define how to render JSX to DOM
* Explain how to use JavaScript expressions in JSX
* Explain how to use inline CSS in JSX

In this hands-on lab, you will learn how to:

* Use JSX syntax in React applications
* Use inline CSS in JSX

## **Prerequisites**

The following is required to complete this hands-on lab:

* Node.js
* NPM
* Visual Studio Code

## **Notes**

Estimated time to complete this lab: **60 minutes.**

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

Create an element to display the heading of the page.

Attribute to display the image of the office space

Create an object of office to display the details like Name, Rent and Address.

Create a list of Object and loop through the office space item to display more data.

To apply Css, Display the color of the Rent in Red if it’s below 60000 and in Green if it’s above 60000.

Output:



**Hint:**





CODE:

import React from 'react';

import './App.css';

function App() {

// 1. Create heading element using JSX

const heading = <h1>🏢 Office Space Rental Listings</h1>;

// 2. Sample image URL

const imageUrl = "https://images.unsplash.com/photo-1581090700227-4c4c2e6e0c01?auto=format&fit=crop&w=800&q=80";

// 3. Office object

const office = {

name: "Indigo Tech Park",

rent: 55000,

address: "Koramangala, Bangalore"

};

// 4. List of office spaces

const officeList = [

{ name: "Indigo Tech Park", rent: 55000, address: "Koramangala, Bangalore" },

{ name: "WeWork Residency", rent: 75000, address: "HSR Layout, Bangalore" },

{ name: "Cowork Central", rent: 42000, address: "MG Road, Bangalore" },

{ name: "Innov8 Infinity", rent: 82000, address: "Indiranagar, Bangalore" }

];

// 5. Render list with conditional inline styling

const officeCards = officeList.map((office, index) => (

<div key={index} style={{

border: '1px solid #ccc',

padding: '12px',

margin: '10px',

borderRadius: '8px',

width: '300px',

display: 'inline-block',

boxShadow: '2px 2px 6px rgba(0,0,0,0.1)'

}}>

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

<p><strong>Address:</strong> {office.address}</p>

<p style={{ color: office.rent < 60000 ? 'red' : 'green' }}>

<strong>Rent:</strong> ₹{office.rent}

</p>

</div>

));

return (

<div className="App" style={{ textAlign: 'center', fontFamily: 'Arial' }}>

{heading}

<img

src={imageUrl}

alt="Office Space"

style={{ width: '70%', borderRadius: '12px', margin: '20px 0' }}

/>

<h2>Featured Office</h2>

<p><strong>Name:</strong> {office.name}</p>

<p><strong>Address:</strong> {office.address}</p>

<p style={{ color: office.rent < 60000 ? 'red' : 'green' }}>

<strong>Rent:</strong> ₹{office.rent}

</p>

<hr style={{ width: '80%', margin: '30px auto' }} />

<h2>All Listings</h2>

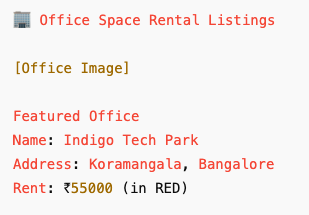
{officeCards}

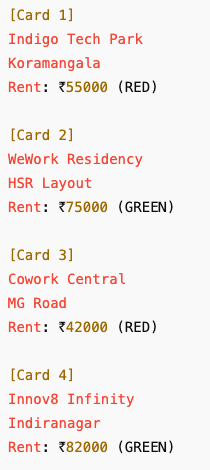
</div>

);

}

export default App;





## **Objectives**

* Explain React events
* Explain about event handlers
* Define Synthetic event
* Identify React event naming convention

In this hands-on lab, you will learn how to:

* Implement Event handling concept in React applications
* Use this keyword
* Use synthetic event

## **Prerequisites**

The following is required to complete this hands-on lab:

* Node.js
* NPM
* Visual Studio Code

## **Notes**

Estimated time to complete this lab: **90 minutes.**

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

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.



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



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



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.



CODE:

import React, { Component } from 'react';

class Counter extends Component {

constructor() {

super();

this.state = { count: 0 };

this.increment = this.increment.bind(this);

this.sayHello = this.sayHello.bind(this);

}

increment() {

this.setState({ count: this.state.count + 1 });

this.sayHello();

}

decrement = () => {

this.setState({ count: this.state.count - 1 });

};

sayHello() {

console.log("Hello! You clicked Increment.");

}

render() {

return (

<div>

<h2>Counter: {this.state.count}</h2>

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

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

</div>

);

}

}

export default Counter;

import React from 'react';

function WelcomeButton() {

const greet = (message) => {

alert(`Message: ${message}`);

};

return (

<div>

<button onClick={() => greet("Welcome")}>Say Welcome</button>

</div>

);

}

export default WelcomeButton;

import React from 'react';

function SyntheticClick() {

const handleClick = (event) => {

console.log("Synthetic Event Object:", event);

alert("I was clicked");

};

return (

<div>

<button onClick={handleClick}>OnPress</button>

</div>

);

}

export default SyntheticClick;

import React, { useState } from 'react';

function CurrencyConverter() {

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

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

const handleSubmit = (e) => {

e.preventDefault();

const rate = 0.011; // ₹1 = €0.011 (example)

setEuro((rupees \* rate).toFixed(2));

};

return (

<div>

<h2>Currency Converter</h2>

<form onSubmit={handleSubmit}>

<label>Amount in ₹:

<input

type="number"

value={rupees}

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

/>

</label>

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

</form>

{euro && <p>Equivalent in €: {euro}</p>}

</div>

);

}

export default CurrencyConverter;

import React from 'react';

import Counter from './components/Counter';

import WelcomeButton from './components/WelcomeButton';

import SyntheticClick from './components/SyntheticClick';

import CurrencyConverter from './components/CurrencyConverter';

function App() {

return (

<div className="App" style={{ textAlign: "center" }}>

<h1>🎯 Event Examples App</h1>

<Counter />

<hr />

<WelcomeButton />

<hr />

<SyntheticClick />

<hr />

<CurrencyConverter />

</div>

);

}

export default App;

OUTPUT:



## **Objectives**

* Explain about conditional rendering in React
* Define element variables
* Explain how to prevent components from rendering

In this hands-on lab, you will learn how to:

* Implement conditional rendering in React applications

## **Prerequisites**

The following is required to complete this hands-on lab:

* Node.js
* NPM
* Visual Studio Code

## **Notes**

Estimated time to complete this lab: **60 minutes.**

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.





**Hint:**







CODE:

import React from 'react';

function GuestPage() {

return (

<div>

<h2>Welcome Guest!</h2>

<p>Please browse flights. Login to book tickets.</p>

</div>

);

}

export default GuestPage;

import React from 'react';

function UserPage() {

return (

<div>

<h2>Welcome User!</h2>

<p>You are logged in. You can now book tickets.</p>

<button>Book a Flight</button>

</div>

);

}

export default UserPage;

import React, { useState } from 'react';

import GuestPage from './GuestPage';

import UserPage from './UserPage';

function LoginControl() {

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

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

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

// Element variable for button

let button;

if (isLoggedIn) {

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

} else {

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

}

// Conditional rendering using if-else

let page;

if (isLoggedIn) {

page = <UserPage />;

} else {

page = <GuestPage />;

}

return (

<div style={{ textAlign: 'center' }}>

<h1>✈️ Ticket Booking App</h1>

{button}

<hr style={{ margin: '20px' }} />

{page}

</div>

);

}

export default LoginControl;

import React from 'react';

import './App.css';

import LoginControl from './components/LoginControl';

function App() {

return (

<div className="App">

<LoginControl />

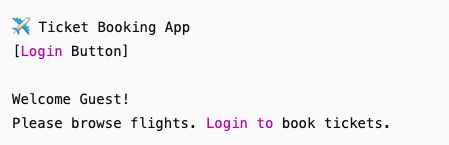
</div>

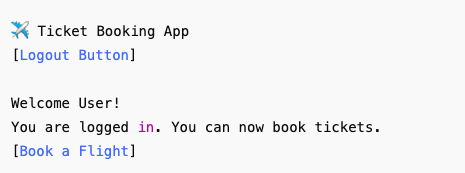
);

}

export default App;

OUTPUT:





## **Objectives**

* Explain various ways of conditional rendering
* Explain how to render multiple components
* Define list component
* Explain about keys in React applications
* Explain how to extract components with keys
* Explain React Map, map() function

In this hands-on lab, you will learn how to:

* Implement conditional rendering in React applications

## **Prerequisites**

The following is required to complete this hands-on lab:

* Node.js
* NPM
* Visual Studio Code

## **Notes**

Estimated time to complete this lab: **60 minutes.**

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

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

Implement this with as many ways possible of Conditional Rendering.



**Hint:**







CODE:

import React from 'react';

function BookDetails({ books }) {

return (

<div>

<h2>📚 Book Details</h2>

<ul>

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

<li key={index}>

<strong>{book.title}</strong> by {book.author}

</li>

))}

</ul>

</div>

);

}

export default BookDetails;

import React from 'react';

function BlogDetails({ blogs }) {

return (

<div>

<h2>📝 Blog Details</h2>

<ul>

{blogs.map((blog) => (

<li key={blog.id}>

<strong>{blog.title}</strong>: {blog.snippet}

</li>

))}

</ul>

</div>

);

}

export default BlogDetails;

import React from 'react';

function CourseDetails({ courses }) {

return (

<div>

<h2>🎓 Course Details</h2>

{courses.length > 0 ? (

<ul>

{courses.map((course) => (

<li key={course.name}>

<strong>{course.name}</strong> — {course.duration} weeks

</li>

))}

</ul>

) : (

<p>No courses available right now.</p>

)}

</div>

);

}

export default CourseDetails;

import React, { useState } from 'react';

import './App.css';

import BookDetails from './components/BookDetails';

import BlogDetails from './components/BlogDetails';

import CourseDetails from './components/CourseDetails';

function App() {

const [activeComponent, setActiveComponent] = useState("books");

const books = [

{ title: "Atomic Habits", author: "James Clear" },

{ title: "Clean Code", author: "Robert C. Martin" },

];

const blogs = [

{ id: 1, title: "React Basics", snippet: "Introduction to React..." },

{ id: 2, title: "State Management", snippet: "Handling state in apps..." },

];

const courses = [

{ name: "ReactJS", duration: 6 },

{ name: "JavaScript ES6", duration: 4 },

];

// Method 1: if-else (classic)

const renderComponent = () => {

if (activeComponent === "books") {

return <BookDetails books={books} />;

} else if (activeComponent === "blogs") {

return <BlogDetails blogs={blogs} />;

} else if (activeComponent === "courses") {

return <CourseDetails courses={courses} />;

} else {

return <p>Please select a section.</p>;

}

};

return (

<div className="App" style={{ textAlign: 'center' }}>

<h1>📖 BloggerApp Dashboard</h1>

{/\* Buttons to toggle views \*/}

<div>

<button onClick={() => setActiveComponent("books")}>Show Books</button>

<button onClick={() => setActiveComponent("blogs")}>Show Blogs</button>

<button onClick={() => setActiveComponent("courses")}>Show Courses</button>

</div>

<hr />

{/\* Method 2: Element variable \*/}

{renderComponent()}

{/\* Method 3: Ternary operator \*/}

<div style={{ marginTop: "30px" }}>

<h3>Quick Check:</h3>

{activeComponent === "books" ? (

<p>📘 Viewing Books</p>

) : activeComponent === "blogs" ? (

<p>🗒️ Viewing Blogs</p>

) : (

<p>🎓 Viewing Courses</p>

)}

</div>

</div>

);

}

export default App;

OUTPUT:

