**React\_HandsOn\_9**

**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:**



Creating The React App

npx create-react-app cricketapp

cd cricketapp

npm start

// src/components/ListofPlayers.js

import React from 'react';

const ListofPlayers = () => {

const players = [

{ name: 'Virat', score: 120 },

{ name: 'Rohit', score: 85 },

{ name: 'Sachin', score: 65 },

{ name: 'Dhoni', score: 90 },

{ name: 'Yuvraj', score: 45 },

{ name: 'Kohli', score: 110 },

{ name: 'Sharma', score: 55 },

{ name: 'Tendulkar', score: 75 },

{ name: 'Dravid', score: 60 },

{ name: 'Ganguly', score: 80 },

{ name: 'Sehwag', score: 30 },

];

// Filter players with scores below 70 using arrow function

const lowScorers = players.filter(player => player.score < 70);

return (

<div>

<h2>All Players</h2>

<ul>

{players.map((player, index) => (

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

))}

</ul>

<h2>Players with Scores Below 70</h2>

<ul>

{lowScorers.map((player, index) => (

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

))}

</ul>

</div>

);

};

export default ListofPlayers;

// src/components/IndianPlayers.js

import React from 'react';

const IndianPlayers = () => {

const players = [

{ name: 'Virat', team: 'Odd' },

{ name: 'Rohit', team: 'Even' },

{ name: 'Sachin', team: 'Odd' },

{ name: 'Dhoni', team: 'Even' },

];

// Destructuring to separate odd and even team players

const [oddPlayers, evenPlayers] = [

players.filter(p => p.team === 'Odd'),

players.filter(p => p.team === 'Even'),

];

// Merge T20 and RanjiTrophy players

const t20Players = ['Kohli', 'Sharma'];

const ranjiPlayers = ['Dravid', 'Ganguly'];

const allPlayers = [...t20Players, ...ranjiPlayers];

return (

<div>

<h2>Odd Team Players</h2>

<ul>

{oddPlayers.map((player, index) => (

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

))}

</ul>

<h2>Even Team Players</h2>

<ul>

{evenPlayers.map((player, index) => (

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

))}

</ul>

<h2>Merged T20 and Ranji Players</h2>

<ul>

{allPlayers.map((player, index) => (

<li key={index}>{player}</li>

))}

</ul>

</div>

);

};

export default IndianPlayers;

// src/App.js

import React, { useState } from 'react';

import ListofPlayers from './components/ListofPlayers';

import IndianPlayers from './components/IndianPlayers';

import './App.css';

function App() {

const [flag, setFlag] = useState(true);

return (

<div className="App">

<h1>Cricket App</h1>

<button onClick={() => setFlag(!flag)}>

Toggle Components (Flag: {flag.toString()})

</button>

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

</div>

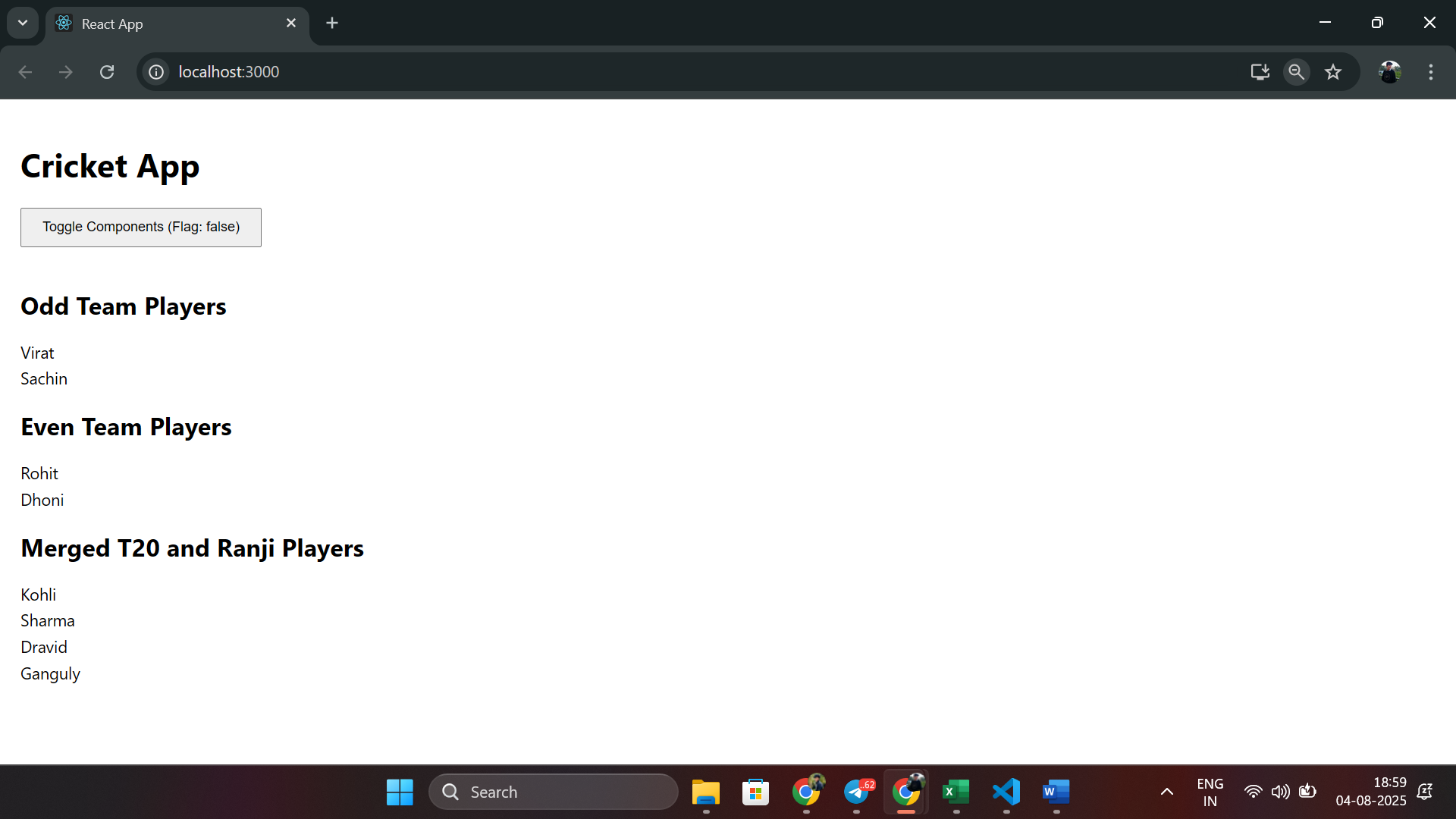
);

}

export default App;

A screenshot of a computer

AI-generated content may be incorrect.



React\_HandsOn\_10

## **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:

A screenshot of a computer

AI-generated content may be incorrect.

**Hint:**

A black background with white text

AI-generated content may be incorrect.

A computer screen shot of text

AI-generated content may be incorrect.

Create the React App:

bashnpx create-react-app officespacerentalapp

cd officespacerentalapp

npm start

Implement App Component:

jsx// src/App.js

import React from 'react';

import './App.css';

function App() {

const offices = [

{ name: 'Office A', rent: 50000, address: '123 Main St' },

{ name: 'Office B', rent: 70000, address: '456 Park Ave' },

{ name: 'Office C', rent: 55000, address: '789 Broad St' },

];

return (

<div className="App">

<h1>Office Space Rental</h1>

<img

src="https://via.placeholder.com/300"

alt="Office Space"

style={{ width: '300px', margin: '20px' }}

/>

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

<div key={index} style={{ margin: '20px 0' }}>

<h2>{office.name}</h2>

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

Rent: ₹{office.rent}

</p>

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

</div>

))}

</div>

);

}

export default App;

Uses JSX to create elements (h1, img, div).

Uses an array of objects and map() to loop through office details.

Applies inline CSS to color rent based on the condition (< 60000: red, >= 60000: green).

Basic Styling:

css// src/App.css

.App {

text-align: center;

padding: 20px;

}

A screenshot of a computer

AI-generated content may be incorrect.

React\_HandsOn\_11

**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.

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.

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.

Create the React App:

bashnpx create-react-app eventexamplesapp

cd eventexamplesapp

npm start

Set Up Components:

Create src/components/Counter.js for increment/decrement buttons.

Create src/components/WelcomeButton.js for the “Say Welcome” button.

Create src/components/OnPressButton.js for the synthetic event.

Create src/components/CurrencyConverter.js for currency conversion.

Modify src/App.js to include all components.

Implement Counter Component:

jsx// src/components/Counter.js

import React, { useState } from 'react';

const Counter = () => {

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

const handleIncrement = () => {

setCount(count + 1);

console.log('Hello, this is a static message!');

};

const handleDecrement = () => {

setCount(count - 1);

};

return (

<div>

<h2>Counter: {count}</h2>

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

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

</div>

);

};

export default Counter;

The Increment button calls handleIncrement, which increments the count and logs a message.

Implement WelcomeButton Component:

jsx// src/components/WelcomeButton.js

import React from 'react';

const WelcomeButton = () => {

const sayWelcome = (message) => {

alert(message);

};

return (

<div>

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

</div>

);

};

export default WelcomeButton;

The button calls sayWelcome with the argument 'welcome'.

Implement OnPressButton Component:

jsx// src/components/OnPressButton.js

import React from 'react';

const OnPressButton = () => {

const handlePress = (event) => {

console.log('Synthetic Event:', event);

alert('I was clicked');

};

return (

<div>

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

</div>

);

};

export default OnPressButton;

Uses a synthetic event (onClick) to display a message and log the event.

Implement CurrencyConverter Component:

jsx// src/components/CurrencyConverter.js

import React, { useState } from 'react';

const CurrencyConverter = () => {

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

const [euros, setEuros] = useState(null);

const handleSubmit = (event) => {

event.preventDefault();

const conversionRate = 0.011; // 1 INR = 0.011 EUR (approx)

setEuros((rupees \* conversionRate).toFixed(2));

};

return (

<div>

<h2>Currency Converter</h2>

<form onSubmit={handleSubmit}>

<input

type="number"

value={rupees}

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

placeholder="Enter amount in INR"

/>

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

</form>

{euros && <p>{rupees} INR = {euros} EUR</p>}

</div>

);

};

export default CurrencyConverter;

Implement App Component:

jsx// src/App.js

import React from 'react';

import Counter from './components/Counter';

import WelcomeButton from './components/WelcomeButton';

import OnPressButton from './components/OnPressButton';

import CurrencyConverter from './components/CurrencyConverter';

import './App.css';

function App() {

return (

<div className="App">

<h1>Event Examples App</h1>

<Counter />

<WelcomeButton />

<OnPressButton />

<CurrencyConverter />

</div>

);

}

export default App;

Basic Styling:

css// src/App.css

.App {

text-align: center;

padding: 20px;

}

button {

margin: 10px;

padding: 10px 20px;

cursor: pointer;

}

input {

margin: 10px;

padding: 5px;

}

A screenshot of a computer

AI-generated content may be incorrect.

React\_HandsOn\_12

Create the React App:

bashnpx create-react-app ticketbookingapp

cd ticketbookingapp

npm start

Set Up Components:

Create src/components/GuestPage.js for the guest page (displays flight details).

Create src/components/UserPage.js for the logged-in user page (allows ticket booking).

Modify src/App.js to handle login state and conditional rendering.

Implement GuestPage Component:

jsx// src/components/GuestPage.js

import React from 'react';

const GuestPage = ({ onLogin }) => {

return (

<div>

<h2>Flight Details</h2>

<p>Flight: AI-202 | Destination: New York | Time: 10:00 AM</p>

<p>Flight: AI-303 | Destination: London | Time: 2:00 PM</p>

<button onClick={onLogin}>Login</button>

</div>

);

};

export default GuestPage;

Displays sample flight details and a Login button.

The onLogin prop will trigger the login state change.

Implement UserPage Component:

jsx// src/components/UserPage.js

import React from 'react';

const UserPage = ({ onLogout }) => {

return (

<div>

<h2>Book Tickets</h2>

<p>Welcome, User! Select a flight to book:</p>

<button>Book Flight AI-202</button>

<button>Book Flight AI-303</button>

<button onClick={onLogout}>Logout</button>

</div>

);

};

export default UserPage;

Displays a booking interface and a Logout button.

The onLogout prop will trigger the logout state change.

Implement App Component with Conditional Rendering:

jsx// src/App.js

import React, { useState } from 'react';

import GuestPage from './components/GuestPage';

import UserPage from './components/UserPage';

import './App.css';

function App() {

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

const handleLogin = () => {

setIsLoggedIn(true);

};

const handleLogout = () => {

setIsLoggedIn(false);

};

return (

<div className="App">

<h1>Ticket Booking App</h1>

{isLoggedIn ? (

<UserPage onLogout={handleLogout} />

) : (

<GuestPage onLogin={handleLogin} />

)}

</div>

);

}

export default App;

Uses the useState hook to manage login state.

Conditionally renders GuestPage or UserPage based on isLoggedIn.

Basic Styling:

css// src/App.css

.App {

text-align: center;

padding: 20px;

}

button {

margin: 10px;

padding: 10px 20px;

cursor: pointer;

}

A screenshot of a computer

AI-generated content may be incorrect.

React\_HandsOn\_13

Create the React App:

bashnpx create-react-app bloggerapp

cd bloggerapp

npm start

Set Up Components:

Create src/components/BookDetails.js, src/components/BlogDetails.js, and src/components/CourseDetails.js.

Modify src/App.js to implement different conditional rendering techniques.

Implement BookDetails Component:

jsx// src/components/BookDetails.js

import React from 'react';

const BookDetails = () => {

return (

<div>

<h2>Book Details</h2>

<p>Title: React Basics</p>

<p>Author: John Doe</p>

</div>

);

};

export default BookDetails;

Implement BlogDetails Component:

jsx// src/components/BlogDetails.js

import React from 'react';

const BlogDetails = () => {

return (

<div>

<h2>Blog Details</h2>

<p>Title: Learning React</p>

<p>Author: Jane Smith</p>

</div>

);

};

export default BlogDetails;

Implement CourseDetails Component:

jsx// src/components/CourseDetails.js

import React from 'react';

const CourseDetails = () => {

return (

<div>

<h2>Course Details</h2>

<p>Title: React for Beginners</p>

<p>Instructor: Alex Brown</p>

</div>

);

};

export default CourseDetails;

Implement App Component with Multiple Conditional Rendering Techniques:

jsx// src/App.js

import React, { useState } from 'react';

import BookDetails from './components/BookDetails';

import BlogDetails from './components/BlogDetails';

import CourseDetails from './components/CourseDetails';

import './App.css';

function App() {

const [view, setView] = useState('book');

// Method 1: If-else (using element variables)

let content;

if (view === 'book') {

content = <BookDetails />;

} else if (view === 'blog') {

content = <BlogDetails />;

} else {

content = <CourseDetails />;

}

return (

<div className="App">

<h1>Blogger App</h1>

<div>

<button onClick={() => setView('book')}>Show Book</button>

<button onClick={() => setView('blog')}>Show Blog</button>

<button onClick={() => setView('course')}>Show Course</button>

</div>

{/\* Method 1: Element Variables \*/}

<h3>Method 1: If-else</h3>

{content}

{/\* Method 2: Ternary Operator \*/}

<h3>Method 2: Ternary Operator</h3>

{view === 'book' ? (

<BookDetails />

) : view === 'blog' ? (

<BlogDetails />

) : (

<CourseDetails />

)}

{/\* Method 3: Logical && Operator \*/}

<h3>Method 3: Logical &&</h3>

{view === 'book' && <BookDetails />}

{view === 'blog' && <BlogDetails />}

{view === 'course' && <CourseDetails />}

{/\* Method 4: Switch Statement (using a function) \*/}

<h3>Method 4: Switch Statement</h3>

{(() => {

switch (view) {

case 'book':

return <BookDetails />;

case 'blog':

return <BlogDetails />;

default:

return <CourseDetails />;

}

})()}

</div>

);

}

export default App;

Basic Styling:

css// src/App.css

.App {

text-align: center;

padding: 20px;

}

button {

margin: 10px;

padding: 10px 20px;

cursor: pointer;

}

div {

margin: 20px 0;

}

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.