**13. ReactJS-HOL**

The objective of this HandsOn is to practice React component creation and conditional rendering within a React application named ‘bloggerapp’.

**Steps followed :**

Created the React App –

I created a new React app ‘bloggerapp’ using command :

A screenshot of a computer program

AI-generated content may be incorrect.

Opened the Project in VS Code –

Using command in command prompt :

A black background with white text

AI-generated content may be incorrect.

Components used –

Inside the src folder, we created 5 .js files :

Created CourseDetails.js :

function CourseDetails() {

const courses = [

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

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

];

return (

<div>

<h1>Course Details</h1>

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

<div key={index}>

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

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

</div>

))}

</div>

);

}

export default CourseDetails;

Created BookDetails.js –

function BookDetails(props) {

const { books } = props;

return (

<div>

<h1>Book Details</h1>

{books.map((book) => (

<div key={book.id}>

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

<h4>{book.price}</h4>

</div>

))}

</div>

);

}

export default BookDetails;

Created BlogDetails.js –

function BlogDetails() {

const blogs = [

{

title: 'React Learning',

author: 'Stephen Biz',

desc: 'Welcome to learning React!',

},

{

title: 'Installation',

author: 'Schwezdenier',

desc: 'You can install React from npm.',

},

];

return (

<div>

<h1>Blog Details</h1>

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

<div key={index}>

<h2>{blog.title}</h2>

<h4><strong>{blog.author}</strong></h4>

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

</div>

))}

</div>

);

}

export default BlogDetails;

Updated App.js –

import './App.css';

import CourseDetails from './CourseDetails';

import BookDetails from './BookDetails';

import BlogDetails from './BlogDetails';

const books = [

{ id: 101, bname: 'Master React', price: 670 },

{ id: 102, bname: 'Deep Dive into Angular 11', price: 800 },

{ id: 103, bname: 'Mongo Essentials', price: 450 },

];

function App() {

return (

<div className="main-container">

<div><CourseDetails /></div>

<div><BookDetails books={books} /></div>

<div><BlogDetails /></div>

</div>

);

}

export default App;

Updated App.css –

.App {

text-align: center;

}

.App-logo {

height: 40vmin;

pointer-events: none;

}

@media (prefers-reduced-motion: no-preference) {

.App-logo {

animation: App-logo-spin infinite 20s linear;

}

}

.App-header {

background-color: #282c34;

min-height: 100vh;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

font-size: calc(10px + 2vmin);

color: white;

}

.App-link {

color: #61dafb;

}

@keyframes App-logo-spin {

from {

transform: rotate(0deg);

}

to {

transform: rotate(360deg);

}

}

.main-container {

display: flex;

justify-content: space-around;

padding: 40px;

font-family: Arial, sans-serif;

}

.main-container > div {

border-left: 4px solid green;

padding-left: 20px;

width: 25%;

}

h1 {

font-size: 24px;

font-weight: bold;

margin-bottom: 15px;

}

h2 {

font-size: 20px;

font-weight: bold;

margin: 10px 0 5px;

}

h3, h4 {

margin: 0 0 10px;

}

p {

margin-bottom: 15px;

}

Ran the App –

A screenshot of a computer

AI-generated content may be incorrect.

Output –

A screenshot of a computer

AI-generated content may be incorrect.

**12. ReactJS-HOL**

The objective of this HandsOn is to create a React application named ‘ticketbookingapp’ that demonstrates login-based conditional rendering. The application shows different content based on whether the user is logged in or not.

**Steps followed :**

Created the React App –

I created a new React app ‘ticketbookingapp’ using command :

A screenshot of a computer program

AI-generated content may be incorrect.

Opened the Project in VS Code –

Using command in command prompt :

A black background with white text

AI-generated content may be incorrect.

Components used –

Inside the src folder, we created 5 .js files :

Created LoginButton.js :

import React from 'react';

function LoginButton({ onClick }) {

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

}

export default LoginButton;

Created LogoutButton.js –

import React from 'react';

function LogoutButton({ onClick }) {

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

}

export default LogoutButton;

Created Greeting.js –

import React from 'react';

import UserGreeting from './UserGreeting';

import GuestGreeting from './GuestGreeting';

function Greeting({ isLoggedIn }) {

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

}

export default Greeting;

Created UserGreeting.js –

import React from 'react';

function UserGreeting() {

return <h1>Welcome back</h1>;

}

export default UserGreeting;

Created GuestGreeting.js –

import React from 'react';

function GuestGreeting() {

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

}

export default GuestGreeting;

Updated App.js –

import React, { useState } from 'react';

import LoginButton from './LoginButton';

import LogoutButton from './LogoutButton';

import Greeting from './Greeting';

function App() {

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

const handleLoginClick = () => {

setIsLoggedIn(true);

};

const handleLogoutClick = () => {

setIsLoggedIn(false);

};

return (

<div style={{ textAlign: 'center', marginTop: '50px' }}>

<Greeting isLoggedIn={isLoggedIn} />

{isLoggedIn ? (

<LogoutButton onClick={handleLogoutClick} />

) : (

<LoginButton onClick={handleLoginClick} />

)}

</div>

);

}

export default App;

Ran the App –

A screenshot of a computer

AI-generated content may be incorrect.

Output –

A screen shot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

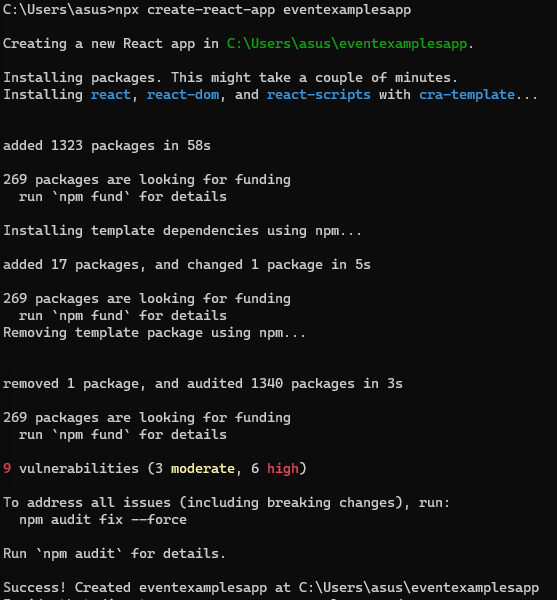
**11. ReactJS-HOL**

The objective of this HandsOn is to explore various event handling techniques in React using functional components and hooks within a React application “eventexamplesapp”.

**Steps followed :**

Created the React App –

I created a new React app ‘eventexamplesapp’ using command :



Opened the Project in VS Code –

Using command in command prompt :

A black background with white text

AI-generated content may be incorrect.

Components used –

Inside the src folder, we created 2 .js files :

Created Counter.js :

import React, { useState } from 'react';

function Counter() {

const [count, setCount] = useState(1);

const increment = () => {

setCount(prev => prev + 1);

sayHello();

};

const decrement = () => {

setCount(prev => prev - 1);

};

const sayHello = () => {

alert("Hello! Static message.");

};

const sayWelcome = (message) => {

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

};

const handleClick = () => {

alert("I was clicked!");

};

return (

<div>

<h2>{count}</h2>

<div>

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

</div>

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

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

</div>

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

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

</div>

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

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

</div>

</div>

);

}

export default Counter;

Created CurrencyConvertor.js –

import React, { useState } from 'react';

function CurrencyConvertor() {

const [amount, setAmount] = useState('');

const [currency, setCurrency] = useState('Euro');

const handleSubmit = (e) => {

e.preventDefault();

const convertedAmount = amount \* 80;

alert(`Converting to ${currency} Amount is ${convertedAmount}`);

};

return (

<div>

<h2 style={{ color: 'green' }}>Currency Convertor!!!</h2>

<form onSubmit={handleSubmit}>

<div>

<label>Amount: </label>

<input

type="text"

value={amount}

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

/>

</div>

<div>

<label>Currency: </label>

<input

type="text"

value={currency}

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

/>

</div>

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

</form>

</div>

);

}

export default CurrencyConvertor;

Updated App.js –

import React from 'react';

import Counter from './Counter';

import CurrencyConvertor from './CurrencyConvertor';

function App() {

return (

<div className="App">

<Counter />

<hr />

<CurrencyConvertor />

</div>

);

}

export default App;

Ran the App –

A screenshot of a computer

AI-generated content may be incorrect.

Output –

A screenshot of a computer

AI-generated content may be incorrect.

**10. ReactJS-HOL**

The objective of this HandsOn to build a React app using JSX, render dynamic elements, and apply conditional styling using CSS classes within a React application “officespacerentalapp”.

**Steps followed :**

Created the React App –

I created a new React app ‘officespacerentalapp’ using command :

A screenshot of a computer program

AI-generated content may be incorrect.

Opened the Project in VS Code –

Using command in command prompt :

A black background with white text

AI-generated content may be incorrect.

Updated App.js –

import './App.css';

const OfficeSpaceRental = () => {

const heading = "Office Space";

const officeList = [

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

{ Name: "Smartworks", Rent: 62000, Address: "Bangalore" },

{ Name: "Regus", Rent: 58000, Address: "Mumbai" }

];

return (

<div className="App">

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

<img

src="https://images.unsplash.com/photo-1585241936936-0b3c4e6aee4e?auto=format&fit=crop&w=800&q=60"

width="25%"

height="25%"

alt="Office Space"

/>

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

const rentColor = item.Rent <= 60000 ? 'textRed' : 'textGreen';

return (

<div key={index}>

<h2>Name: {item.Name}</h2>

<h3 className={rentColor}>Rent: Rs. {item.Rent}</h3>

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

</div>

);

})}

</div>

);

};

export default OfficeSpaceRental;

Updated App.css –

.App {

text-align: center;

}

.App-logo {

height: 40vmin;

pointer-events: none;

}

@media (prefers-reduced-motion: no-preference) {

.App-logo {

animation: App-logo-spin infinite 20s linear;

}

}

.App-header {

background-color: #282c34;

min-height: 100vh;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

font-size: calc(10px + 2vmin);

color: white;

}

.App-link {

color: #61dafb;

}

@keyframes App-logo-spin {

from {

transform: rotate(0deg);

}

to {

transform: rotate(360deg);

}

}

.textRed {

color: red;

font-weight: bold;

}

.textGreen {

color: green;

font-weight: bold;

}

Ran the App –

A screenshot of a computer program

AI-generated content may be incorrect.

Output –

A screenshot of a computer

AI-generated content may be incorrect.

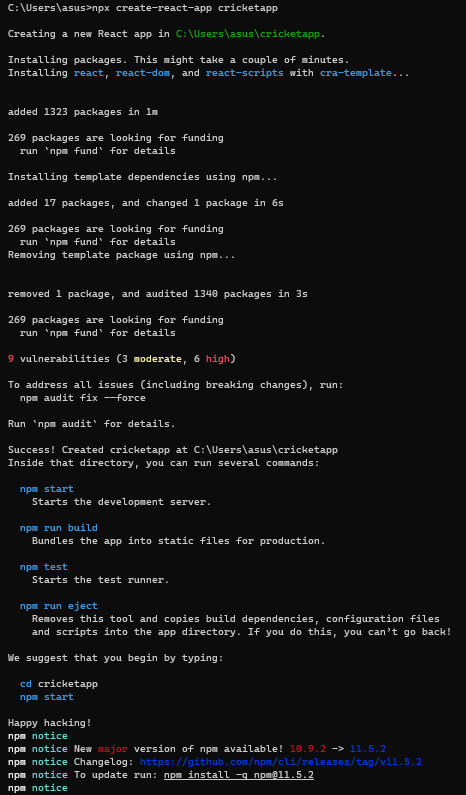
**9. ReactJS-HOL**

The objective of this HandsOn is to practice ES6 features such as arrow functions, array destructuring, merging arrays, and conditional rendering within a React application “cricketapp”.

**Steps followed :**

Created the React App –

I created a new React app ‘cricketapp’ using command :



Opened the Project in VS Code –

Using command in command prompt :



Components used –

Inside the src folder, we created 5 .js files :

Created ListofPlayers.js :

function ListofPlayers({ players }) {

return (

<div>

<ul>

{

players.map((item) => {

return (

<li>

Mr. {item.name} <span> {item.score}</span>

</li>

);

})

}

</ul>

</div>

);

}

export default ListofPlayers;

Created Scorebelow70.js –

function Scorebelow70({ players }) {

let players70 = [];

players.map((item) => {

if (item.score <= 70) {

players70.push(item);

}

});

return (

<div>

<ul>

{

players70.map((item) => {

return (

<li>

Mr. {item.name} <span> {item.score}</span>

</li>

);

})

}

</ul>

</div>

);

}

export default Scorebelow70;

Created OddPlayers.js –

function OddPlayers({ players }) {

const [first, , third, , fifth] = players;

return (

<div>

<h3>Odd Players</h3>

<li>First: {first}</li>

<li>Third: {third}</li>

<li>Fifth: {fifth}</li>

</div>

);

}

export default OddPlayers;

Created EvenPlayers.js –

function EvenPlayers({ players }) {

const [, second, , fourth, , sixth] = players;

return (

<div>

<h3>Even Players</h3>

<li>Second: {second}</li>

<li>Fourth: {fourth}</li>

<li>Sixth: {sixth}</li>

</div>

);

}

export default EvenPlayers;

Created ListofIndianPlayers.js –

export function ListofIndianPlayers({ IndianPlayers }) {

return (

<div>

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

<li key={index}>Mr. {item}</li>

))}

</div>

);

}

Modified App.js to Include All Components –

import React from 'react';

import ListofPlayers from './ListofPlayers';

import Scorebelow70 from './Scorebelow70';

import OddPlayers from './OddPlayers';

import EvenPlayers from './EvenPlayers';

import { ListofIndianPlayers } from './ListofIndianPlayers';

const players = [

{ name: 'Jack', score: 50 },

{ name: 'Michael', score: 70 },

{ name: 'John', score: 40 },

{ name: 'Ann', score: 61 },

{ name: 'Elisabeth', score: 61 },

{ name: 'Sachin', score: 95 },

{ name: 'Dhoni', score: 100 },

{ name: 'Virat', score: 84 },

{ name: 'Jadeja', score: 64 },

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

{ name: 'Rohit', score: 80 }

];

const IndianTeam = ['Sachin', 'Dhoni', 'Virat', 'Rohit', 'Raina', 'Yuvraj'];

const T20Players = ['First Player', 'Second Player', 'Third Player'];

const RanjiTrophyPlayers = ['Fourth Player', 'Fifth Player', 'Sixth Player'];

const IndianPlayers = [...T20Players, ...RanjiTrophyPlayers];

function App() {

const flag = true;

if (flag === true) {

return (

<div>

<h1>List of Players</h1>

<ListofPlayers players={players} />

<hr />

<h1>List of Players having Scores Less than 70</h1>

<Scorebelow70 players={players} />

</div>

);

} else {

return (

<div>

<div>

<h1>Indian Team</h1>

<h1>Odd Players</h1>

<OddPlayers players={IndianTeam} />

<hr />

<h1>Even Players</h1>

<EvenPlayers players={IndianTeam} />

</div>

<hr />

<div>

<h1>List of Indian Players Merged:</h1>

<ListofIndianPlayers IndianPlayers={IndianPlayers} />

</div>

</div>

);

}

}

export default App;

Ran the App –

A screenshot of a computer program

AI-generated content may be incorrect.

Output –

When flag = true –

A screenshot of a computer

AI-generated content may be incorrect.

When flag = false –

A screenshot of a sports team

AI-generated content may be incorrect.