**9.Reactjs-HOL**

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

npx create-react-app cricketapp

cd cricketapp

code .

**Create Components**

 ListofPlayers.js

 IndianPlayers.js

**ListofPlayers.js**

import React from 'react';

function ListofPlayers() {

const players = [

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

{ name: 'Rohit Sharma', score: 88 },

{ name: 'KL Rahul', score: 60 },

{ name: 'Shubman Gill', score: 73 },

{ name: 'Rishabh Pant', score: 35 },

{ name: 'Hardik Pandya', score: 82 },

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

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

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

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

{ name: 'Surya Kumar', score: 99 }

];

// Use arrow function to filter players below 70

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 Scoring Below 70</h2>

<ul>

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

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

))}

</ul>

</div>

);

}

export default ListofPlayers;

**IndianPlayers.js**

import React from 'react';

function IndianPlayers() {

const T20players = ['Rohit', 'Virat', 'Pant', 'Bumrah'];

const RanjiTrophy = ['Pujara', 'Rahane', 'Jadeja', 'Shami'];

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

const oddPlayers = allPlayers.filter((\_, index) => index % 2 === 0);

const evenPlayers = allPlayers.filter((\_, index) => index % 2 !== 0);

return (

<div>

<h2>All Indian Players</h2>

<ul>

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

</ul>

<h2>Odd Team Players</h2>

<ul>

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

</ul>

<h2>Even Team Players</h2>

<ul>

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

</ul>

</div>

);

}

export default IndianPlayers;

**App.js**

import React from 'react';

import ListofPlayers from './ListofPlayers';

import IndianPlayers from './IndianPlayers';

function App() {

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

return (

<div className="App">

<h1>Cricket App</h1>

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

</div>

);

}

export default App;

**Run the Application**

npm start

**Output**

Compiled successfully!

You can now view cricketapp in the browser.

Local: http://localhost:3000

On Your Network: http://10.38.85.110:3000

Note that the development build is not optimized.

To create a production build, use npm run build.

webpack compiled successfully

When flag value is true

A screenshot of a computer

Description automatically generated

When flag value is false

A screenshot of a computer

Description automatically generated

**10.Reactjs-HOL**

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

**Create React App**

npx create-react-app officespacerentalapp

cd officespacerentalapp

code .

**App.js**

import React from 'react';

function App() {

const officeList = [

{

name: 'Regus Workspace',

rent: 55000,

address: 'Mumbai',

image: 'https://images.unsplash.com/photo-1573164574572-cb89e39749b4?auto=format&fit=crop&w=800&q=60'

},

{

name: 'WeWork Central',

rent: 67000,

address: 'Hyderabad',

image: 'https://images.unsplash.com/photo-1581091870622-2b0c6c7b2e4c?auto=format&fit=crop&w=800&q=60'

},

{

name: 'Tidel Tech Park',

rent: 48000,

address: 'Chennai',

image: 'https://images.unsplash.com/photo-1596075780750-d2bdc8cdaac8?auto=format&fit=crop&w=800&q=60'

},

{

name: 'Mindspace IT Park',

rent: 72000,

address: 'Pune',

image: 'https://images.unsplash.com/photo-1581074817984-b4e70c6a4f1d?auto=format&fit=crop&w=800&q=60'

},

{

name: 'Cowork Station',

rent: 59000,

address: 'Bengaluru',

image: 'https://images.unsplash.com/photo-1573164713988-8665fc963095?auto=format&fit=crop&w=800&q=60'

}

];

return (

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

<h1>Office Space , at Affordable Range</h1>

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

const rentStyle = {

color: office.rent < 60000 ? 'red' : 'green',

fontWeight: 'bold'

};

return (

<div key={index} style={{

marginBottom: '30px',

borderBottom: '1px solid #ccc',

paddingBottom: '20px',

maxWidth: '350px'

}}>

<img

src={office.image}

alt={office.name}

width="300"

height="200"

style={{ borderRadius: '5px' }}

/>

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

<p style={rentStyle}>Rent: Rs. {office.rent}</p>

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

</div>

);

})}

</div>

);

}

export default App;

**Run the App**

npm start

**Output**

Compiled successfully!

You can now view officespacerentalapp in the browser.

Local: http://localhost:3001

On Your Network: http://10.38.85.110:3001

Note that the development build is not optimized.

To create a production build, use npm run build.

webpack compiled successfully

A screenshot of a computer

Description automatically generated

**11.Reactjs-HOL**

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

npx create-react-app eventexamplesapp

cd eventexamplesapp

code .

**Create Component Files**

 Counter.js

 CurrencyConvertor.js

**Counter.js**

import React, { useState } from 'react';

function Counter() {

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

// Multiple methods: increment and say hello

const handleIncrement = () => {

increment();

sayHello();

};

const increment = () => {

setCount(prev => prev + 1);

};

const sayHello = () => {

alert("Hello! This is a static message.");

};

const handleDecrement = () => {

setCount(prev => prev - 1);

};

const sayWelcome = (message) => {

alert(message);

};

const handleSynthetic = () => {

alert("I was clicked");

};

return (

<div style={{ marginBottom: '40px' }}>

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

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

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

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

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

</div>

);

}

export default Counter;

**CurrencyConvertor.js**

import React, { useState } from 'react';

function CurrencyConvertor() {

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

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

const handleChange = (e) => {

setRupees(e.target.value);

};

const handleSubmit = () => {

const euroValue = parseFloat(rupees) / 90;

setEuros(euroValue.toFixed(2));

};

return (

<div>

<h2>Currency Convertor</h2>

<input

type="number"

placeholder="Enter amount in INR"

value={rupees}

onChange={handleChange}

/>

<button onClick={handleSubmit}>Convert</button>

<p>Value in Euro: €{euros}</p>

</div>

);

}

export default CurrencyConvertor;

**App.js**

import React from 'react';

import Counter from './Counter';

import CurrencyConvertor from './CurrencyConvertor';

function App() {

return (

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

<h1>Event Examples App</h1>

<Counter />

<CurrencyConvertor />

</div>

);

}

export default App;

**Run the App**

npm start

**Output**

Compiled successfully!

You can now view eventexamplesapp in the browser.

Local: http://localhost:3001

On Your Network: http://10.38.85.110:3001

Note that the development build is not optimized.

To create a production build, use npm run build.

webpack compiled successfully

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**12.Reactjs-HOL**

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.

npx create-react-app ticketbookingapp

cd ticketbookingapp

npm start

Create a components folder inside src.

FlightList.js

import React from 'react';

const FlightList = () => {

const flights = [

{ id: 1, airline: "IndiGo", from: "Delhi", to: "Mumbai", time: "10:00 AM" },

{ id: 2, airline: "Air India", from: "Chennai", to: "Bangalore", time: "12:00 PM" },

{ id: 3, airline: "SpiceJet", from: "Hyderabad", to: "Kolkata", time: "2:00 PM" },

];

return (

<div>

<h2>Available Flights</h2>

<ul>

{flights.map((flight) => (

<li key={flight.id}>

✈ {flight.airline} - {flight.from} → {flight.to} at {flight.time}

</li>

))}

</ul>

</div>

);

};

export default FlightList;

GuestPage.js

import React from 'react';

import FlightList from './FlightList';

const GuestPage = () => {

return (

<div>

<h1>Welcome Guest!</h1>

<FlightList />

<p>Please login to book tickets.</p>

</div>

);

};

export default GuestPage;

UserPage.js

import React from 'react';

import FlightList from './FlightList';

const UserPage = () => {

return (

<div>

<h1>Welcome User!</h1>

<FlightList />

<button>Book Tickets</button>

</div>

);

};

export default UserPage;

App.js

// src/App.js

import React, { useState } from 'react';

import GuestPage from './components/GuestPage';

import UserPage from './components/UserPage';

function App() {

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

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

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

return (

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

<h1>Ticket Booking App</h1>

{isLoggedIn ? (

<>

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

<UserPage />

</>

) : (

<>

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

<GuestPage />

</>

)}

</div>

);

}

export default App;

**Run the App**

npm start

**Output**

Compiled successfully!

You can now view ticketbookingapp in the browser.

Local: http://localhost:3002

On Your Network: http://10.38.85.110:3002

Note that the development build is not optimized.

To create a production build, use npm run build.

webpack compiled successfully

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**13.Reactjs-HOL**

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.

**Create the react app**

npx create-react-app bloggerapp

cd bloggerapp

**BookDetails.js**

import React from 'react';

const BookDetails = ({ category }) => {

const techBooks = [

{ title: 'Learning Node.js', price: 500 },

{ title: 'Master React', price: 700 }

];

const designBooks = [

{ title: 'UI/UX Essentials', price: 650 },

{ title: 'Figma Basics', price: 480 }

];

const books = category === 'tech' ? techBooks :

category === 'design' ? designBooks : null;

if (!books) return <p>No books found in this category.</p>;

return (

<div style={{ borderLeft: '3px solid green', padding: '10px' }}>

<h2>Book Details ({category})</h2>

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

<div key={idx}>

<strong>{book.title}</strong><br />

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

</div>

))}

</div>

);

};

export default BookDetails;

**CourseDetails.js**

import React from 'react';

const CourseDetails = ({ level }) => {

const beginnerCourses = [

{ name: 'HTML Basics', date: '1/1/2021' },

{ name: 'CSS Fundamentals', date: '1/5/2021' }

];

const advancedCourses = [

{ name: 'React Advanced', date: '3/2/2022' },

{ name: 'Angular Mastery', date: '6/3/2022' }

];

let courses;

if (level === 'beginner') {

courses = beginnerCourses;

} else if (level === 'advanced') {

courses = advancedCourses;

} else {

return <p>No courses available.</p>;

}

return (

<div style={{ borderLeft: '3px solid green', padding: '10px' }}>

<h2>Course Details ({level})</h2>

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

<div key={idx}>

<strong>{course.name}</strong><br />

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

</div>

))}

</div>

);

};

export default CourseDetails;

**BlogDetails.js**

import React from 'react';

const BlogDetails = ({ topic }) => {

let content;

switch (topic) {

case 'react':

content = {

title: 'React Learning',

author: 'Jane Doe',

body: 'React is a powerful library for building UIs.'

};

break;

case 'vue':

content = {

title: 'Vue Insights',

author: 'John Vue',

body: 'Vue is simple yet powerful for small to mid-size apps.'

};

break;

default:

return <p>No blog available on this topic.</p>;

}

return (

<div style={{ borderLeft: '3px solid green', padding: '10px' }}>

<h2>Blog Details ({topic})</h2>

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

<strong>{content.author}</strong>

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

</div>

);

};

export default BlogDetails;

**App.js**

import React, { useState } from 'react';

import CourseDetails from './components/CourseDetails';

import BookDetails from './components/BookDetails';

import BlogDetails from './components/BlogDetails';

function App() {

const [courseLevel, setCourseLevel] = useState('beginner'); // 'beginner' or 'advanced'

const [bookCategory, setBookCategory] = useState('tech'); // 'tech' or 'design'

const [blogTopic, setBlogTopic] = useState('react'); // 'react' or 'vue'

return (

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

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

<div style={{ marginBottom: '20px' }}>

<button onClick={() => setCourseLevel('beginner')}>Beginner Courses</button>

<button onClick={() => setCourseLevel('advanced')}>Advanced Courses</button>

<button onClick={() => setBookCategory('tech')}>Tech Books</button>

<button onClick={() => setBookCategory('design')}>Design Books</button>

<button onClick={() => setBlogTopic('react')}>React Blog</button>

<button onClick={() => setBlogTopic('vue')}>Vue Blog</button>

</div>

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

<CourseDetails level={courseLevel} />

<BookDetails category={bookCategory} />

<BlogDetails topic={blogTopic} />

</div>

</div>

);

}

export default App;

**Output**

Compiled successfully!

You can now view bloggerapp in the browser.

Local: http://localhost:3002

On Your Network: http://10.38.85.110:3002

Note that the development build is not optimized.

To create a production build, use npm run build.

webpack compiled successfully

A screenshot of a computer

Description automatically generated