**WEEK -7 ASSIGNMENT**

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

Programs:

App.jsx  
import React from 'react'

import ListOfPlayers from './components/ListOfPlayers'

import IndianPlayers from './components/IndianPlayers'

const flag = true // Change to false to switch output

export default function App() {

return (

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

<h1>Cricket App</h1>

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

</div>

)

}

ListOfPlayers.jsx  
import React from 'react'

export default function ListOfPlayers() {

const players = [

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

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

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

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

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

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

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

{ name: 'Mr. Virat', score: 64 },

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

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

{ name: 'Mr. Rohit', score: 80 }

]

const lessThan70 = players.filter(player => player.score < 70)

return (

<div>

<h2>List of Players</h2>

<ul>

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

<li key={index}>

{player.name} {player.score}

</li>

))}

</ul>

<h2>List of Players having Scores Less than 70</h2>

<ul>

{lessThan70.map((player, index) => (

<li key={index}>

{player.name} {player.score}

</li>

))}

</ul>

</div>

)

}

IndianPlayers.jsx  
import React from 'react'

export default function IndianPlayers() {

const T20players = ['Sachin', 'Dhoni2', 'Virat3', 'Rohit4', 'Yuvraj5']

const RanjiTrophy = [

'First Player',

'Second Player',

'Third Player',

'Fourth Player',

'Fifth Player'

]

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

const [oddPlayers, evenPlayers] = mergedPlayers.reduce(

(acc, player, index) => {

if ((index + 1) % 2 === 0) acc[1].push(player)

else acc[0].push(player)

return acc

},

[[], []]

)

return (

<div>

<h2>Odd Players</h2>

<ul>

{oddPlayers.map((p, i) => (

<li key={i}>

{i === 0 ? 'First' : i === 1 ? 'Third' : 'Fifth'} : {p}

</li>

))}

</ul>

<h2>Even Players</h2>

<ul>

{evenPlayers.map((p, i) => (

<li key={i}>

{i === 0 ? 'Second' : i === 1 ? 'Fourth' : 'Sixth'} : {p}

</li>

))}

</ul>

<h2>List of Indian Players Merged:</h2>

<ul>

{mergedPlayers.map((p, i) => (

<li key={i}>{p}</li>

))}

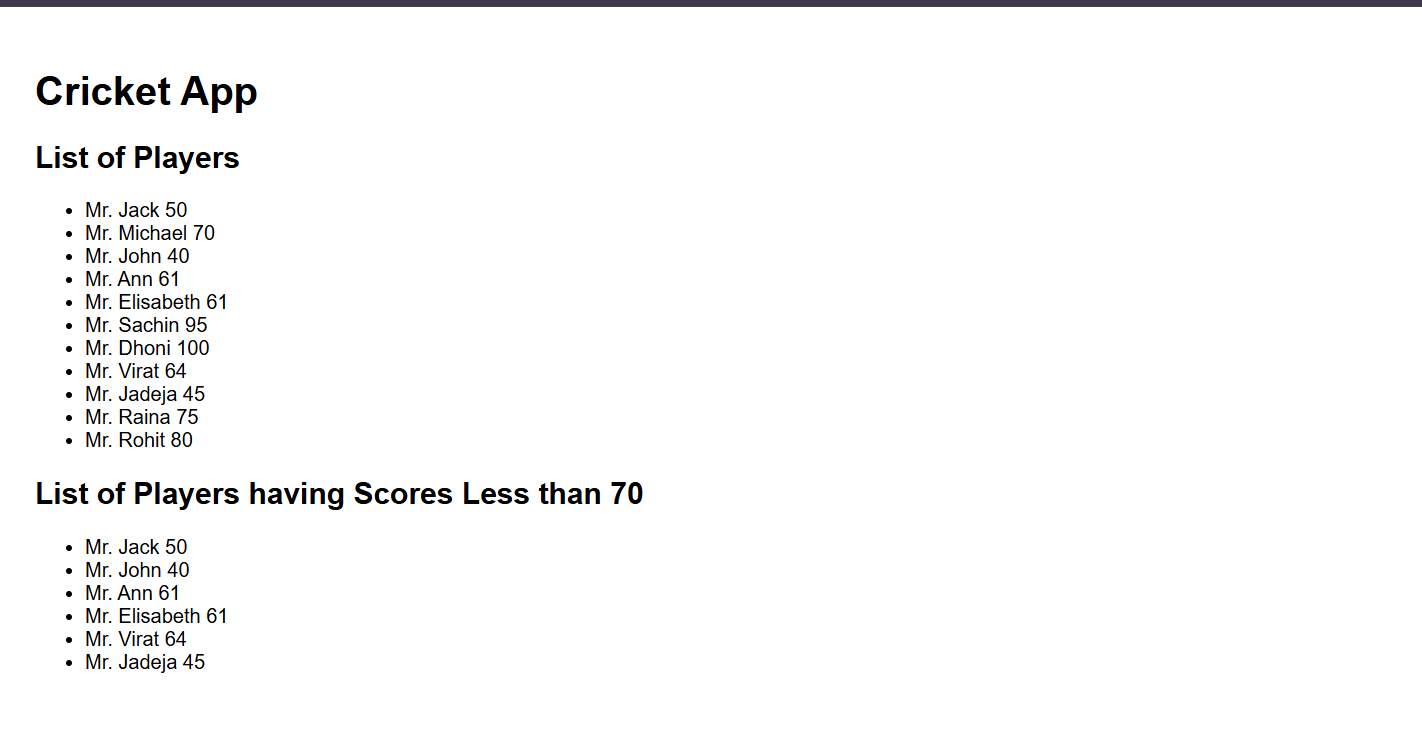
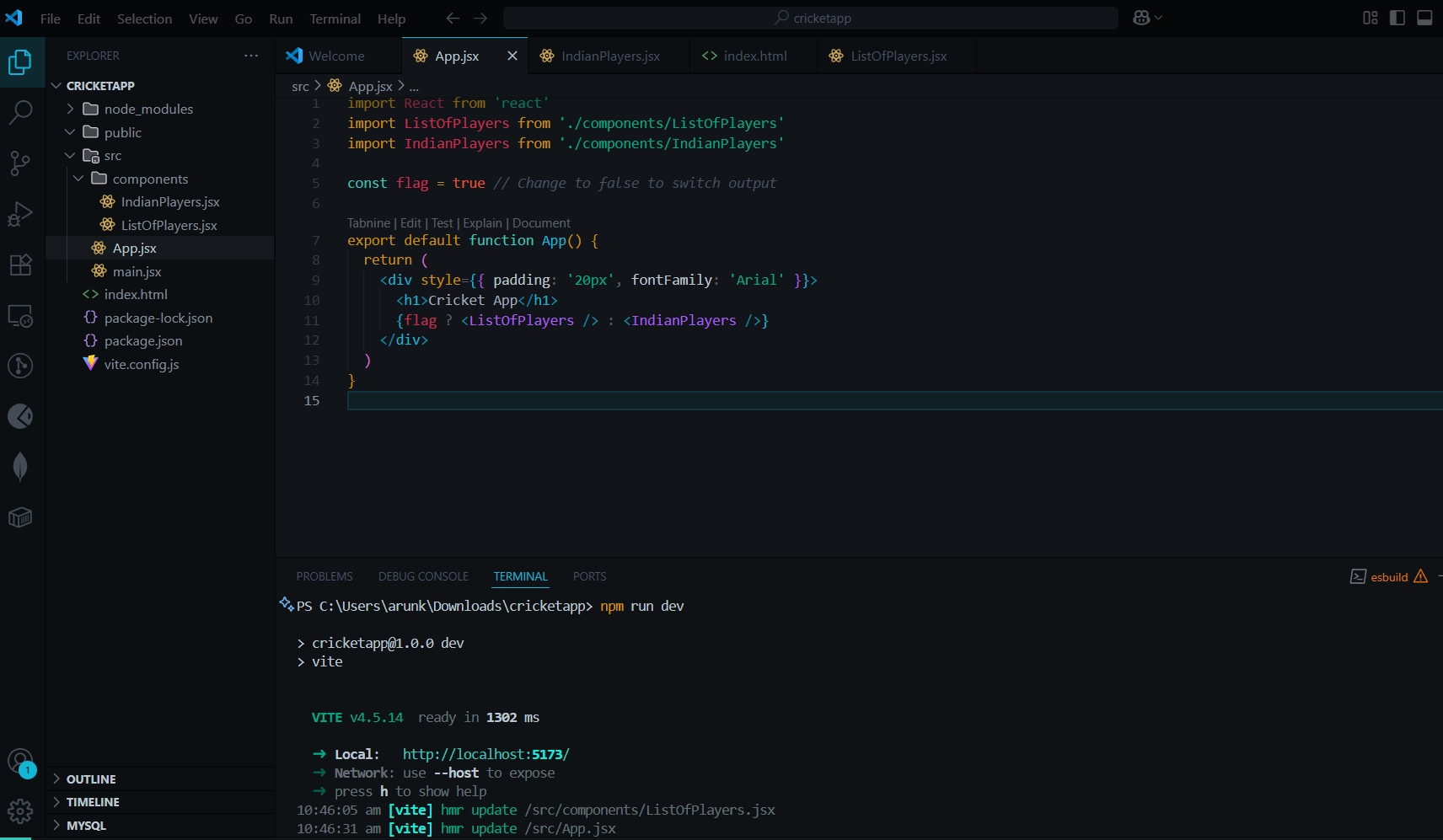
</ul>

</div>

)

}

Output:

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

Programs  
App.jsx:

import React from "react";

import office1 from './assets/images.jpeg'

export default function App() {

// Single office object

const office = {

name: "Tech Park Tower",

rent: 55000,

address: "MG Road, Bangalore",

image: office1 // Use the imported image path here

};

// Multiple office spaces

const officeList = [

{

name: "Tech Park Tower",

rent: 55000,

address: "MG Road, Bangalore",

image: office1 // You can also use local images here

},

{

name: "Global Business Hub",

rent: 75000,

address: "DLF Cyber City, Gurgaon",

image: "https://via.placeholder.com/200x150.png?text=Office+2"

},

{

name: "Startup Square",

rent: 45000,

address: "Hitech City, Hyderabad",

image: "https://via.placeholder.com/200x150.png?text=Office+3"

}

];

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

return (

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

{heading}

{/\* Single Office \*/}

<h2>Featured Office</h2>

<div

style={{

border: "1px solid #ccc",

padding: "10px",

marginBottom: "20px",

}}

>

<img src={office.image} alt={office.name} />

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

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

Rent: ₹{office.rent}

</p>

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

</div>

{/\* List of Offices \*/}

<h2>Available Offices</h2>

<div style={{ display: "flex", gap: "20px", flexWrap: "wrap" }}>

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

<div

key={index}

style={{

border: "1px solid #ccc",

padding: "10px",

width: "220px",

textAlign: "center",

}}

>

<img src={o.image} alt={o.name} />

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

<p style={{ color: o.rent < 60000 ? "red" : "green" }}>

Rent: ₹{o.rent}

</p>

<p>{o.address}</p>

</div>

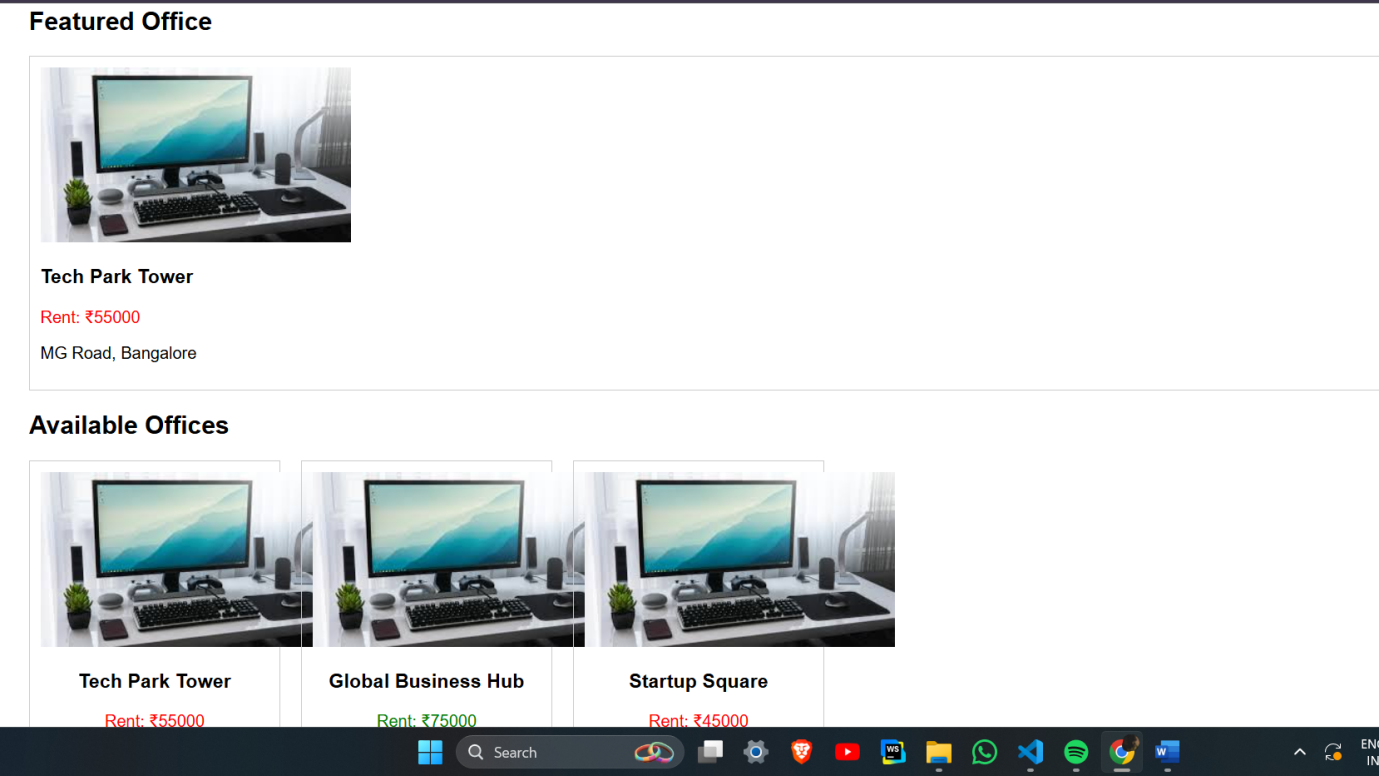
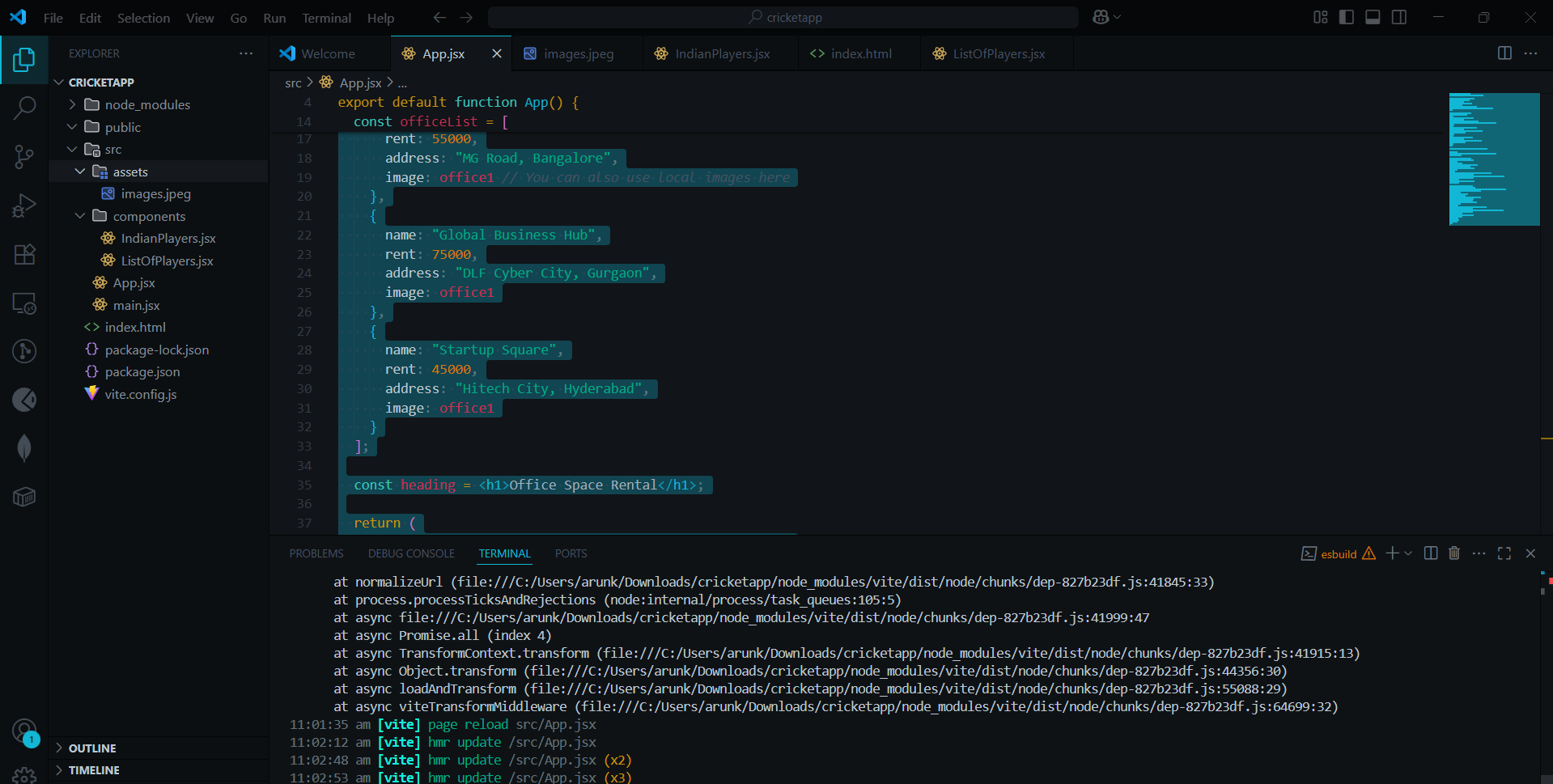
))}

</div>

</div>

);

}

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

Program:

App.jsx:

import React, { useState } from "react";

export default function CurrencyConvertor() {

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

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

const handleSubmit = (e) => {

e.preventDefault();

const euro = (parseFloat(rupees) / 90).toFixed(2); // 1 Euro ≈ 90 INR

setResult(`€${euro}`);

};

return (

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

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

<form onSubmit={handleSubmit}>

<input

type="number"

value={rupees}

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

placeholder="Enter amount in INR"

/>

<button type="submit" style={{ marginLeft: "10px" }}>

Convert

</button>

</form>

{result && <p>Converted Amount: {result}</p>}

</div>

);

}

### CurrencyConvertor.jsx

import React, { useState } from "react";

export default function CurrencyConvertor() {

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

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

const handleSubmit = (e) => {

e.preventDefault();

const euro = (parseFloat(rupees) / 90).toFixed(2); // 1 Euro ≈ 90 INR

setResult(`€${euro}`);

};

return (

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

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

<form onSubmit={handleSubmit}>

<input

type="number"

value={rupees}

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

placeholder="Enter amount in INR"

/>

<button type="submit" style={{ marginLeft: "10px" }}>

Convert

</button>

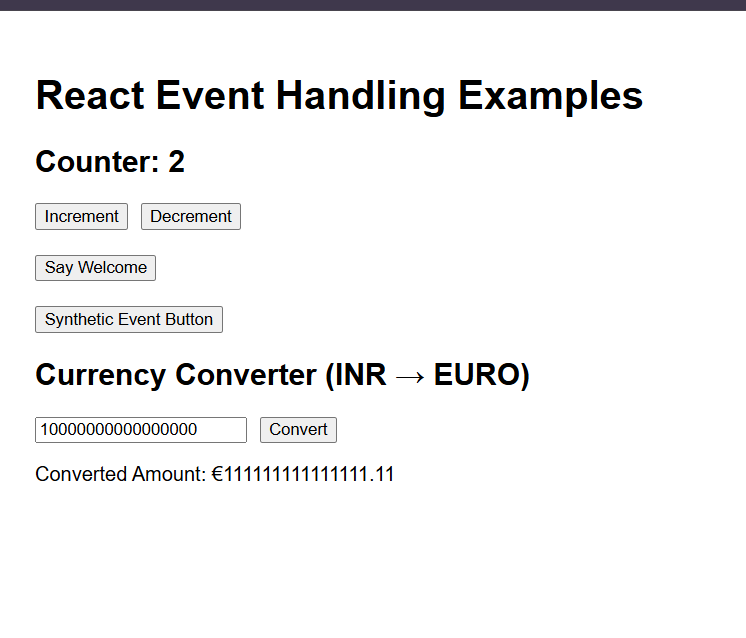
</form>

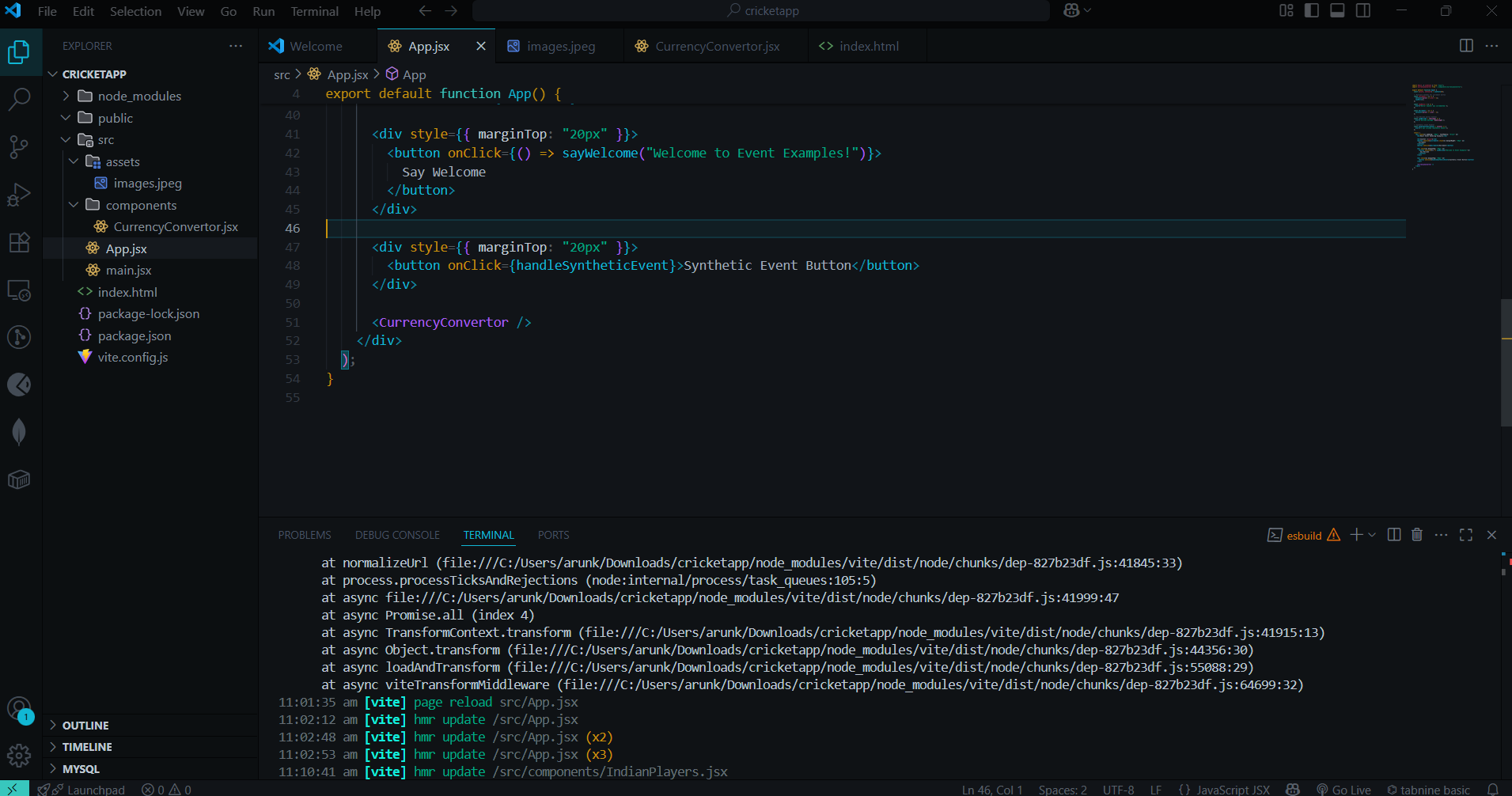
{result && <p>Converted Amount: {result}</p>}

</div>

);

}





4.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.

Programs:

GuestPage.jsx  
import React from "react";

export default function GuestPage() {

const flights = [

{ flight: "AI-202", from: "Delhi", to: "Bangalore", time: "10:00 AM" },

{ flight: "6E-502", from: "Mumbai", to: "Hyderabad", time: "1:00 PM" },

{ flight: "SG-105", from: "Chennai", to: "Pune", time: "4:30 PM" },

];

return (

<div>

<h2>Welcome Guest</h2>

<p>You can browse flight details below:</p>

<table border="1" cellPadding="5">

<thead>

<tr>

<th>Flight</th>

<th>From</th>

<th>To</th>

<th>Time</th>

</tr>

</thead>

<tbody>

{flights.map((f, i) => (

<tr key={i}>

<td>{f.flight}</td>

<td>{f.from}</td>

<td>{f.to}</td>

<td>{f.time}</td>

</tr>

))}

</tbody>

</table>

</div>

);

}  
UserPage.jsx

import React from "react";

export default function UserPage() {

return (

<div>

<h2>Welcome User</h2>

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

<button>Book Ticket</button>

</div>

);

}  
APP.jsx:

import React, { useState } from "react";

import GuestPage from "./components/GuestPage";

import UserPage from "./components/UserPage";

export default function App() {

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

let page; // element variable for conditional rendering

if (isLoggedIn) {

page = <UserPage />;

} else {

page = <GuestPage />;

}

return (

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

<h1>Ticket Booking App</h1>

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

{isLoggedIn ? (

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

) : (

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

)}

</div>

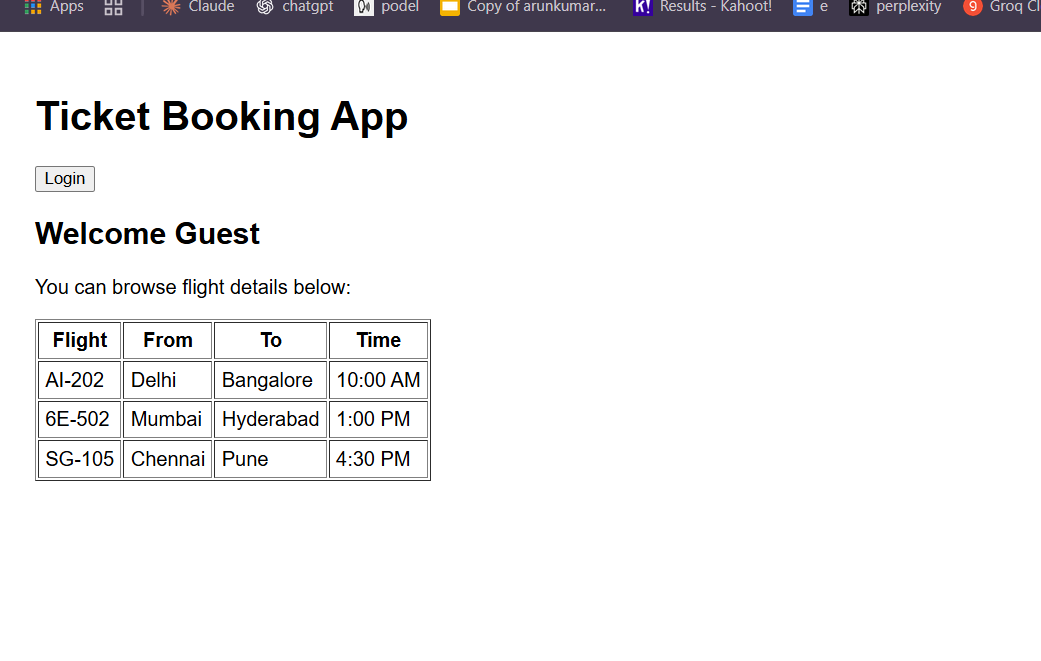
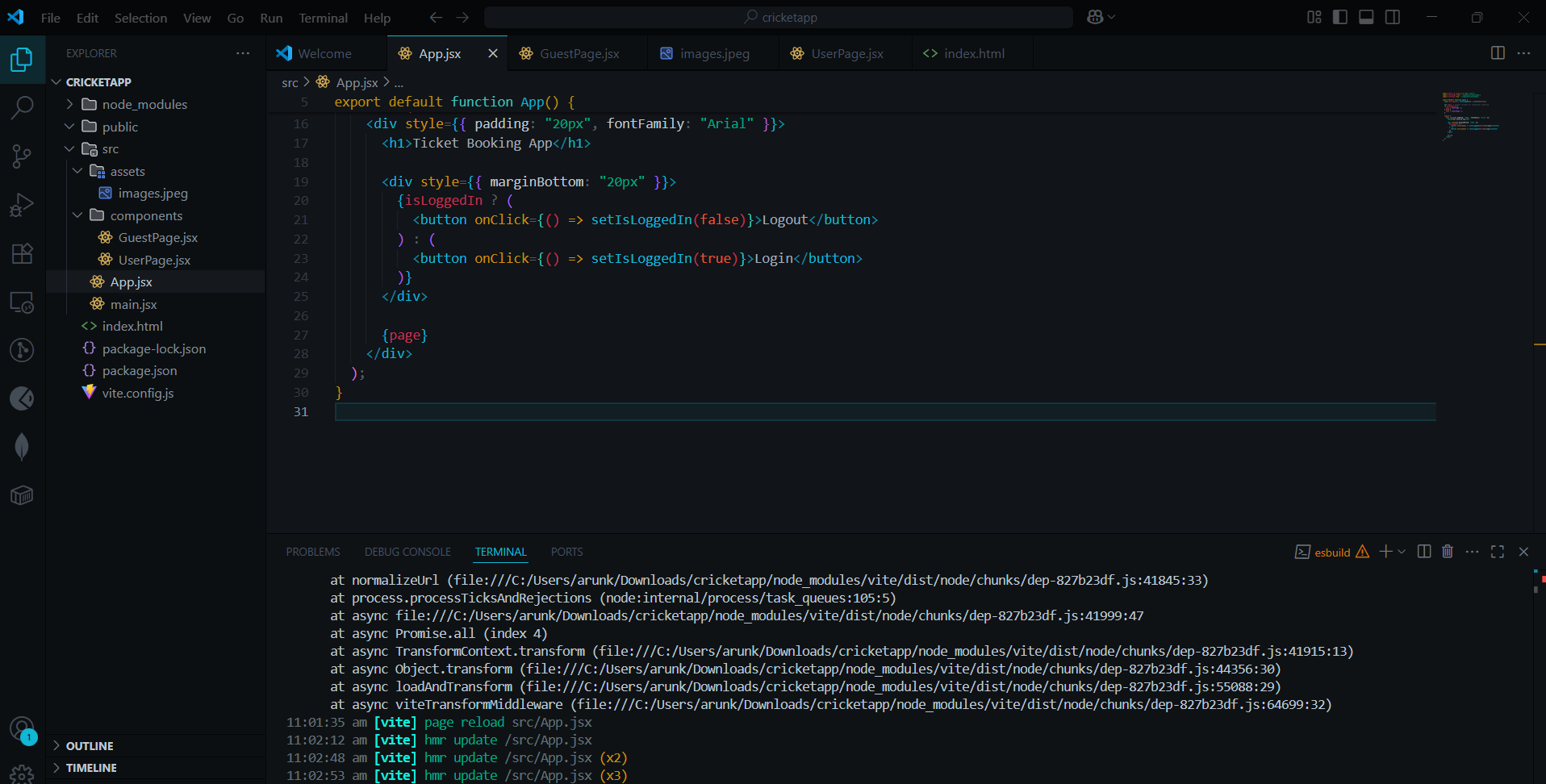
{page}

</div>

);

}

Output:

5.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.

Programs:

BookDetails.jsx

import React from "react";

export default function BookDetails() {

const books = [

{ id: 1, title: "JavaScript Guide", author: "John Doe" },

{ id: 2, title: "React Basics", author: "Jane Smith" },

];

return (

<div>

<h2>Book Details</h2>

<ul>

{books.map((book) => (

<li key={book.id}>

{book.title} - {book.author}

</li>

))}

</ul>

</div>

);

}

BlogDetails.jsx

import React from "react";

export default function BlogDetails() {

const blogs = [

{ id: 1, title: "Learning React", date: "2024-01-01" },

{ id: 2, title: "JavaScript Tips", date: "2024-02-15" },

];

return (

<div>

<h2>Blog Details</h2>

{blogs.map((blog) => (

<div key={blog.id}>

<h4>{blog.title}</h4>

<p>Date: {blog.date}</p>

</div>

))}

</div>

);

}

CourseDetails.jsx:

import React from "react";

export default function CourseDetails() {

const courses = ["React", "JavaScript", "Node.js"];

return (

<div>

<h2>Course Details</h2>

<ul>

{courses.map((c, i) => (

<li key={i}>{c}</li>

))}

</ul>

</div>

);

}

App.jsx:

import React, { useState } from "react";

import BookDetails from "./components/BookDetails";

import BlogDetails from "./components/BlogDetails";

import CourseDetails from "./components/CourseDetails";

export default function App() {

const [selected, setSelected] = useState("books"); // default page

// Approach 1: if/else

let content;

if (selected === "books") {

content = <BookDetails />;

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

content = <BlogDetails />;

} else {

content = <CourseDetails />;

}

return (

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

<h1>Blogger App</h1>

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

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

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

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

</div>

<hr />

{/\* Approach 1: Using element variable \*/}

{content}

<hr />

{/\* Approach 2: Using ternary operator \*/}

{selected === "books" ? (

<BookDetails />

) : selected === "blogs" ? (

<BlogDetails />

) : (

<CourseDetails />

)}

<hr />

{/\* Approach 3: Short-circuit && \*/}

{selected === "books" && <BookDetails />}

{selected === "blogs" && <BlogDetails />}

{selected === "courses" && <CourseDetails />}

</div>

);

}

