WEEK 7 HANDSON

Q1) npx create-react-app cricketapp

->ListofPlayers.js

import React from 'react';

function ListofPlayers() {

// Array of 11 players with names and scores

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 }

];

// Filter players with scores less than 70

const playersWithScoreLessThan70 = players.filter(item => item.score < 70);

return (

<div>

<h1>List of Players</h1>

<ul>

{/\* Display all players using map \*/}

{players.map((item) => (

<li key={item.name}>Mr. {item.name}<span> {item.score}</span></li>

))}

</ul>

<hr />

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

<ul>

{/\* Display filtered players using map \*/}

{playersWithScoreLessThan70.map((item) => (

<li key={item.name}>Mr. {item.name}<span> {item.score}</span></li>

))}

</ul>

</div>

);

}

export default ListofPlayers;

-> IndianPlayers.js

import React from 'react';

// Helper component for Odd players using array destructuring

function OddPlayers({ players }) {

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

return (

<div>

<h2>Odd Players</h2>

<ul>

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

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

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

</ul>

</div>

);

}

// Helper component for Even players using array destructuring

function EvenPlayers({ players }) {

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

return (

<div>

<h2>Even Players</h2>

<ul>

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

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

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

</ul>

</div>

);

}

// Main component to display Indian players

function IndianPlayers() {

const IndianTeam = ['Sachin1', 'Dhoni2', 'Virat3', 'Rohit4', 'Yuvraj5', 'Raina6'];

// Declare two arrays to be merged

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

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

// Merge arrays using the ES6 spread operator

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

return (

<div>

<h1>Indian Team</h1>

<OddPlayers players={IndianTeam} />

<EvenPlayers players={IndianTeam} />

<hr />

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

<ul>

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

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

))}

</ul>

</div>

);

}

export default IndianPlayers;

-> App.js

import React from 'react';

import './App.css';

import ListofPlayers from './ListofPlayers';

import IndianPlayers from './IndianPlayers';

function App() {

// Set this flag to true or false to switch the view

const flag = true;

// Conditional rendering based on the flag

if (flag) {

return (

<div className="App">

<ListofPlayers />

</div>

);

} else {

return (

<div className="App">

<IndianPlayers />

</div>

);

}

}

export default App;

Output) 1



Output 2)



Q2) npx create-react-app officespacerentalapp

1. App.js

import './App.css';

import officeImage from './office-space.jpg'; // We will add this image in the next step

function App() {

// Create an object to display details for a single office

const officeDetails = {

Name: "DBS",

Rent: 50000,

Address: "Chennai"

};

// Style object for the rent price

const rentStyle = {

color: officeDetails.Rent <= 60000 ? 'red' : 'green'

};

return (

<div className="App">

{/\* Create an element to display the heading \*/}

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

{/\* Attribute to display the image \*/}

<img src={officeImage} alt="Modern office space" width="300" />

{/\* Display details from the office object \*/}

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

{/\* Apply inline CSS to change the rent color \*/}

<h3 style={rentStyle}>Rent: Rs. {officeDetails.Rent}</h3>

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

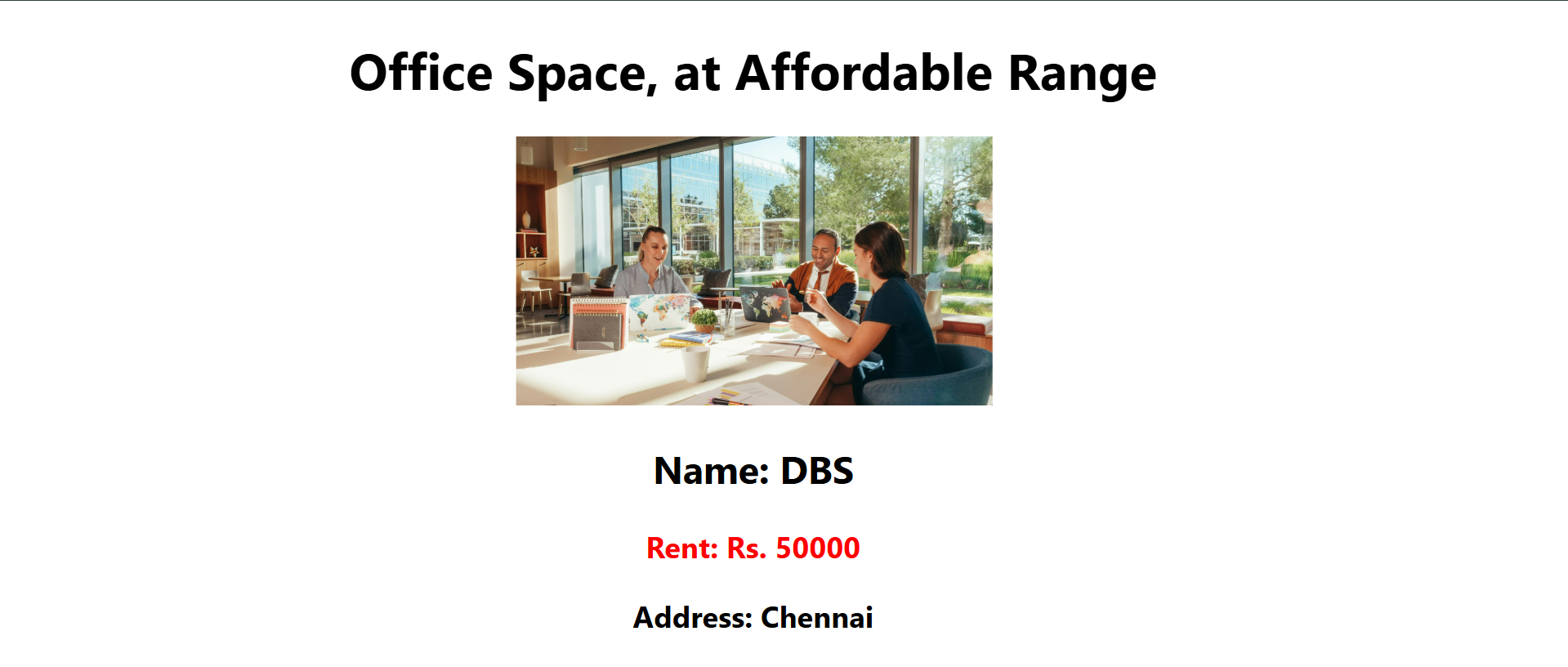
</div>

);

}

export default App;

Output)



Q3) npx create-react-app eventexamplesapp

1. CurrencyConvertor.js

import React, { useState } from 'react';

// We'll use a fixed rate for this lab. (1 Euro ≈ 90 INR)

const INR\_TO\_EURO\_RATE = 90;

function CurrencyConvertor() {

const [inr, setInr] = useState('');

const [euro, setEuro] = useState(0);

// Updates state as the user types in the input field

const handleInputChange = (event) => {

setInr(event.target.value);

};

// This function handles the conversion logic when the button is clicked

const handleSubmit = (event) => {

event.preventDefault(); // Prevents the form from reloading the page

const convertedAmount = parseFloat(inr) / INR\_TO\_EURO\_RATE;

setEuro(convertedAmount.toFixed(2)); // Round to 2 decimal places

};

return (

<div>

<h2>Currency Convertor</h2>

<form onSubmit={handleSubmit}>

<label>

INR:

<input type="number" value={inr} onChange={handleInputChange} placeholder="Enter amount in INR" />

</label>

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

</form>

<h3>Converted Amount: €{euro}</h3>

</div>

);

}

export default CurrencyConvertor;

1. EventsComponent.js

import React, { useState } from 'react';

function EventsComponent() {

const [counter, setCounter] = useState(0);

// --- Counter Logic ---

// Method to increment the counter value [cite: 46]

const incrementCounter = () => {

setCounter(prevCounter => prevCounter + 1);

};

// A second method that will also be called on increment

const sayHello = () => {

alert('Hello! The value will be incremented.'); // [cite: 47]

};

// The button will invoke multiple methods with this single function [cite: 45]

const handleIncrementClick = () => {

incrementCounter();

sayHello();

};

// Method to decrement the counter value

const decrementCounter = () => {

setCounter(prevCounter => prevCounter - 1); // [cite: 44]

};

// --- Other Event Examples ---

// This function takes an argument and is called by the "Say Welcome" button

const sayWelcome = (message) => {

alert(message); // [cite: 48]

};

// This function handles the click and receives React's SyntheticEvent

const handlePress = (event) => {

console.log(event); // The 'event' object is a SyntheticEvent [cite: 35]

alert('I was clicked!'); // [cite: 49]

};

return (

<div>

<h2>Counter</h2>

<h3>Value: {counter}</h3>

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

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

<hr />

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

<hr />

{/\* The document mentions "OnPress", but for web React, the standard is onClick, which uses a SyntheticEvent \*/}

<button onClick={handlePress}>Invoke Synthetic Event</button>

</div>

);

}

export default EventsComponent;

1. App.js

import './App.css';

import EventsComponent from './EventsComponent';

import CurrencyConvertor from './CurrencyConvertor';

function App() {

return (

<div className="App">

<h1>React Event Handling Examples</h1>

{/\* This is the main component for event examples \*/}

<EventsComponent />

{/\* This is the currency convertor component \*/}

<hr className="component-divider" />

<CurrencyConvertor />

</div>

);

}

export default App;

1. App.css

.App {

text-align: center;

max-width: 600px;

margin: 40px auto;

padding: 20px;

border: 1px solid #ddd;

border-radius: 8px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

}

button {

margin: 5px;

padding: 10px 15px;

font-size: 16px;

cursor: pointer;

border-radius: 5px;

border: 1px solid #ccc;

background-color: #f0f0f0;

}

button:hover {

background-color: #e0e0e0;

}

hr {

border: none;

height: 1px;

background-color: #eee;

margin: 25px 0;

}

.component-divider {

border-style: dashed;

}

input {

padding: 8px;

margin: 0 10px;

font-size: 16px;

border-radius: 5px;

border: 1px solid #ccc;

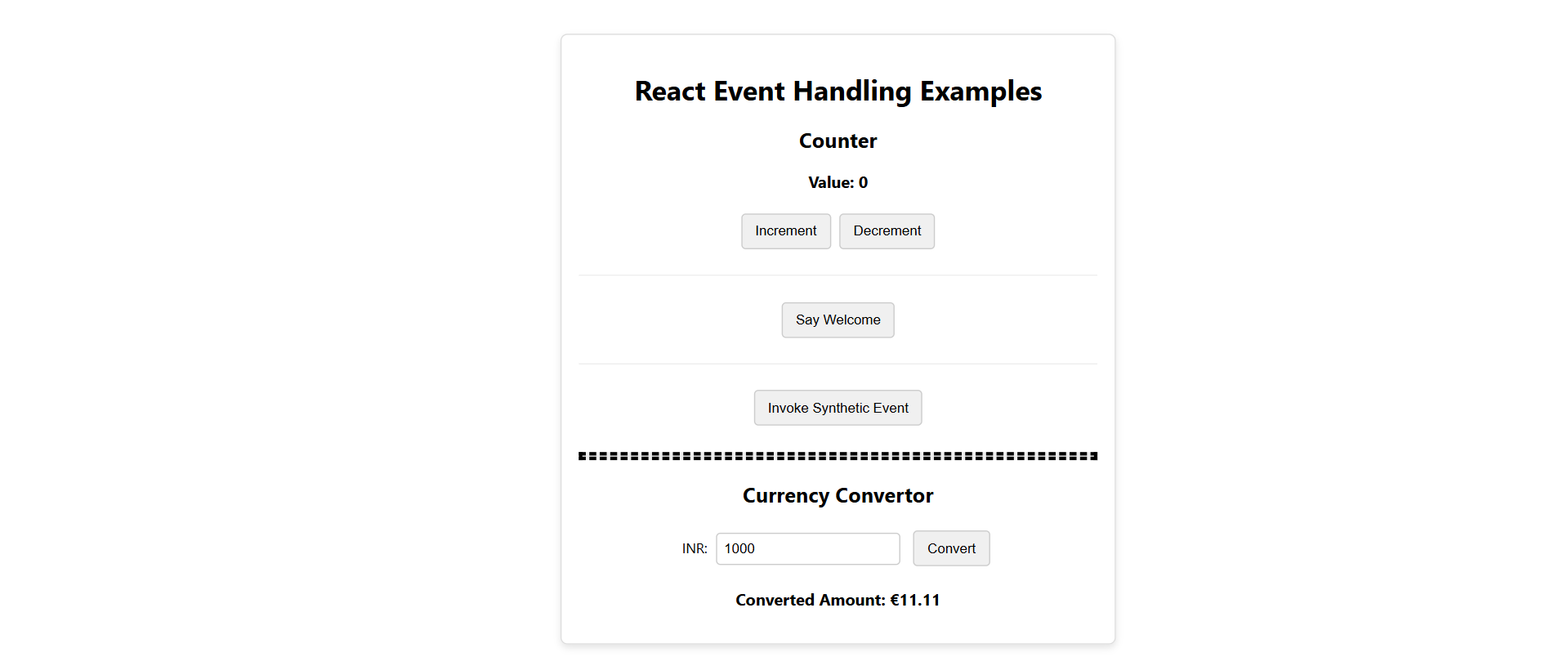
}

form {

margin-bottom: 20px;

}

Output)



Q4) npx create-react-app ticketbookingapp

1. App.css

/\* Reset and base styles \*/

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

body {

font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',

'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',

sans-serif;

-webkit-font-smoothing: antialiased;

-moz-osx-font-smoothing: grayscale;

}

.App {

min-height: 100vh;

display: flex;

flex-direction: column;

}

.container {

max-width: 1200px;

margin: 0 auto;

padding: 0 20px;

}

/\* Header Styles \*/

.app-header {

background: white;

box-shadow: 0 2px 4px rgba(0,0,0,0.1);

border-bottom: 1px solid #e5e7eb;

padding: 1rem 0;

}

.header-content {

display: flex;

justify-content: space-between;

align-items: center;

}

.app-title h1 {

font-size: 1.8rem;

color: #1f2937;

margin-bottom: 0.25rem;

}

.user-status {

font-size: 0.875rem;

color: #6b7280;

}

/\* Button Styles \*/

.login-btn, .logout-btn {

padding: 0.75rem 1.5rem;

border: none;

border-radius: 0.5rem;

font-weight: 600;

cursor: pointer;

transition: all 0.2s;

font-size: 1rem;

}

.login-btn {

background: #3b82f6;

color: white;

}

.login-btn:hover {

background: #2563eb;

}

.logout-btn {

background: #ef4444;

color: white;

}

.logout-btn:hover {

background: #dc2626;

}

/\* Page Styles \*/

.guest-page {

background: #f9fafb;

min-height: calc(100vh - 200px);

flex: 1;

}

.user-page {

background: #eff6ff;

min-height: calc(100vh - 200px);

flex: 1;

}

.page-header {

text-align: center;

padding: 2rem 0;

}

.page-header h1 {

font-size: 2rem;

color: #1f2937;

margin-bottom: 0.5rem;

}

.page-header p {

color: #6b7280;

font-size: 1.1rem;

}

/\* Booking Status \*/

.booking-status {

margin-bottom: 1.5rem;

padding: 1rem;

background: #dcfce7;

border: 1px solid #86efac;

color: #166534;

border-radius: 0.5rem;

text-align: center;

font-weight: 500;

}

/\* Flights Grid \*/

.flights-grid {

display: grid;

gap: 1.5rem;

grid-template-columns: repeat(auto-fit, minmax(320px, 1fr));

padding-bottom: 2rem;

}

/\* Flight Card \*/

.flight-card {

background: white;

border-radius: 0.75rem;

box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);

padding: 1.5rem;

border: 1px solid #e5e7eb;

transition: transform 0.2s, box-shadow 0.2s;

}

.flight-card:hover {

transform: translateY(-2px);

box-shadow: 0 8px 15px rgba(0, 0, 0, 0.15);

}

.flight-header {

display: flex;

justify-content: space-between;

align-items: flex-start;

margin-bottom: 1rem;

}

.flight-header h3 {

font-size: 1.25rem;

color: #1f2937;

font-weight: 600;

}

.price {

font-size: 1.5rem;

font-weight: bold;

color: #3b82f6;

}

.user-price {

color: #059669;

}

/\* Flight Details \*/

.flight-details {

margin-bottom: 1.5rem;

}

.detail-row {

display: flex;

justify-content: space-between;

margin-bottom: 0.75rem;

padding: 0.25rem 0;

}

.detail-row span:first-child {

color: #6b7280;

}

.detail-row span:last-child {

font-weight: 500;

color: #1f2937;

}

/\* Book Button \*/

.book-btn {

width: 100%;

padding: 0.75rem;

border: none;

border-radius: 0.5rem;

font-weight: 600;

font-size: 1rem;

cursor: pointer;

transition: all 0.2s;

}

.book-btn.disabled {

background: #d1d5db;

color: #6b7280;

cursor: not-allowed;

}

.book-btn.active {

background: #10b981;

color: white;

}

.book-btn.active:hover {

background: #059669;

transform: translateY(-1px);

}

/\* Footer \*/

.app-footer {

background: #1f2937;

color: white;

padding: 1.5rem 0;

margin-top: auto;

}

.app-footer p {

text-align: center;

color: #d1d5db;

}

/\* Responsive Design \*/

@media (max-width: 768px) {

.header-content {

flex-direction: column;

gap: 1rem;

}

.app-title {

text-align: center;

}

.flights-grid {

grid-template-columns: 1fr;

}

.page-header h1 {

font-size: 1.5rem;

}

.container {

padding: 0 15px;

}

}

1. App.js

import React, { useState } from 'react';

import './App.css';

function App() {

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

const [bookingStatus, setBookingStatus] = useState('');

// Sample flight data

const flights = [

{

id: 1,

from: 'New York',

to: 'Los Angeles',

departure: '10:00 AM',

arrival: '1:00 PM',

price: '$299',

airline: 'SkyWings'

},

{

id: 2,

from: 'Chicago',

to: 'Miami',

departure: '2:30 PM',

arrival: '6:45 PM',

price: '$189',

airline: 'AirFlow'

},

{

id: 3,

from: 'Seattle',

to: 'Boston',

departure: '8:15 AM',

arrival: '4:30 PM',

price: '$349',

airline: 'CloudLine'

}

];

const handleLogin = () => {

setIsLoggedIn(true);

setBookingStatus('');

};

const handleLogout = () => {

setIsLoggedIn(false);

setBookingStatus('');

};

const handleBookTicket = (flight) => {

setBookingStatus(`Ticket booked successfully for ${flight.from} to ${flight.to}!`);

};

// Element variable for conditional rendering

let navigationButton;

if (isLoggedIn) {

navigationButton = (

<button

onClick={handleLogout}

className="logout-btn"

>

Logout

</button>

);

} else {

navigationButton = (

<button

onClick={handleLogin}

className="login-btn"

>

Login

</button>

);

}

// Guest Component - Only displays flight information

const GuestPage = () => {

return (

<div className="guest-page">

<div className="container">

<div className="page-header">

<h1>Flight Information</h1>

<p>Browse available flights (Login required to book tickets)</p>

</div>

<div className="flights-grid">

{flights.map(flight => (

<div key={flight.id} className="flight-card">

<div className="flight-header">

<h3>{flight.airline}</h3>

<span className="price">{flight.price}</span>

</div>

<div className="flight-details">

<div className="detail-row">

<span>From:</span>

<span>{flight.from}</span>

</div>

<div className="detail-row">

<span>To:</span>

<span>{flight.to}</span>

</div>

<div className="detail-row">

<span>Departure:</span>

<span>{flight.departure}</span>

</div>

<div className="detail-row">

<span>Arrival:</span>

<span>{flight.arrival}</span>

</div>

</div>

<button

disabled

className="book-btn disabled"

>

Login Required to Book

</button>

</div>

))}

</div>

</div>

</div>

);

};

// User Component - Displays flight information with booking capability

const UserPage = () => {

return (

<div className="user-page">

<div className="container">

<div className="page-header">

<h1>Book Your Flight</h1>

<p>Welcome! You can now book tickets for available flights</p>

</div>

{bookingStatus && (

<div className="booking-status">

{bookingStatus}

</div>

)}

<div className="flights-grid">

{flights.map(flight => (

<div key={flight.id} className="flight-card">

<div className="flight-header">

<h3>{flight.airline}</h3>

<span className="price user-price">{flight.price}</span>

</div>

<div className="flight-details">

<div className="detail-row">

<span>From:</span>

<span>{flight.from}</span>

</div>

<div className="detail-row">

<span>To:</span>

<span>{flight.to}</span>

</div>

<div className="detail-row">

<span>Departure:</span>

<span>{flight.departure}</span>

</div>

<div className="detail-row">

<span>Arrival:</span>

<span>{flight.arrival}</span>

</div>

</div>

<button

onClick={() => handleBookTicket(flight)}

className="book-btn active"

>

Book Ticket

</button>

</div>

))}

</div>

</div>

</div>

);

};

return (

<div className="App">

{/\* Header with Navigation \*/}

<header className="app-header">

<div className="container">

<div className="header-content">

<div className="app-title">

<h1>✈️ Ticket Booking App</h1>

<span className="user-status">

{isLoggedIn ? 'Logged in as User' : 'Guest Mode'}

</span>

</div>

{/\* Element variable usage for conditional rendering \*/}

{navigationButton}

</div>

</div>

</header>

{/\* Main Content - Conditional Rendering \*/}

<main>

{isLoggedIn ? <UserPage /> : <GuestPage />}

</main>

{/\* Footer \*/}

<footer className="app-footer">

<div className="container">

<p>

This app demonstrates conditional rendering in React -

{isLoggedIn ? ' User can book tickets' : ' Guest can only view flights'}

</p>

</div>

</footer>

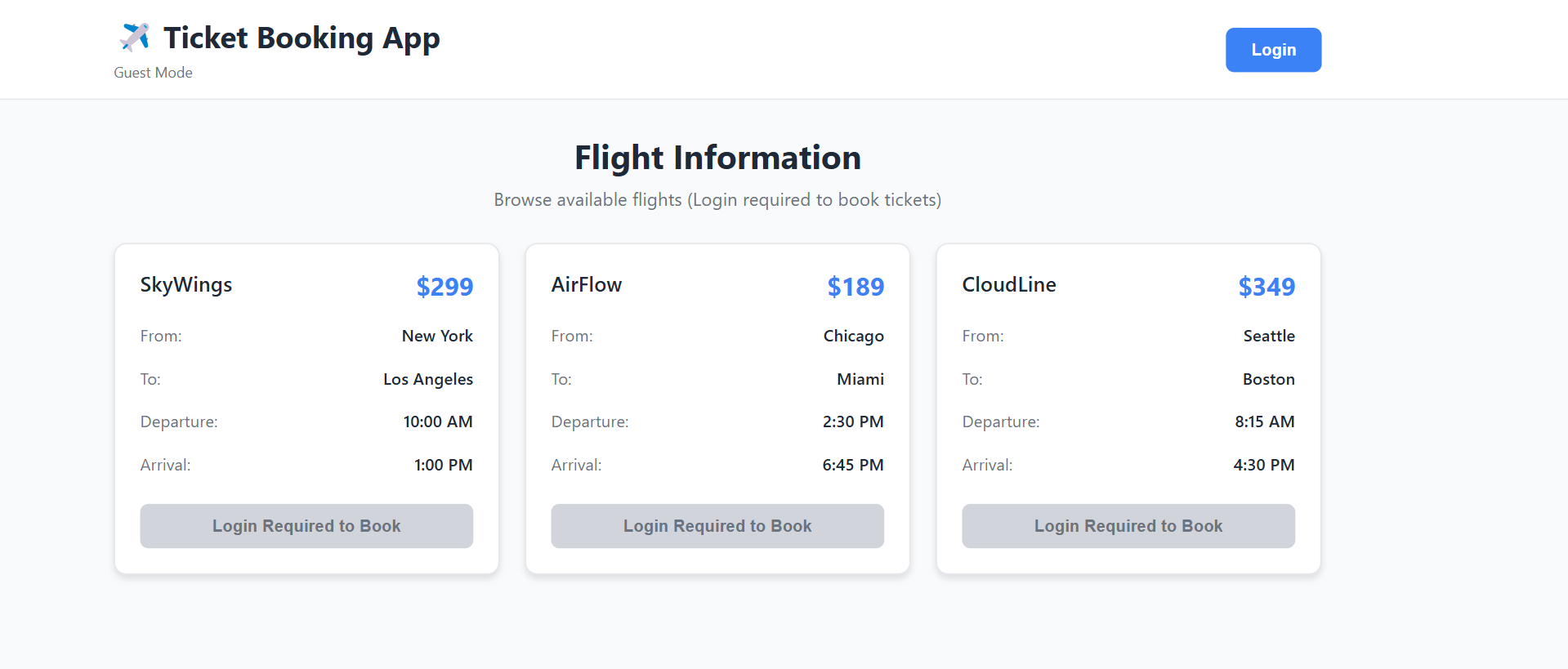
</div>

);

}

export default App;

Output



Q5) npx create-react-app bloggerapp

1. CourseDetails.js

import React from 'react';

// Data for courses

const courses = [

{ id: 1, name: 'Angular', date: '4/5/2021' },

{ id: 2, name: 'React', date: '6/3/2020' },

];

function CourseDetails() {

return (

<div className="component-container">

<h2>Course Details</h2>

{/\* Map through the courses array to display each one \*/}

{courses.map(course => (

<div key={course.id} className="item-container">

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

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

</div>

))}

</div>

);

}

export default CourseDetails;  
  
2) BookDetails.js

import React from 'react';

// Data for books [cite: 47]

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 BookDetails() {

return (

<div className="component-container">

<h2>Book Details</h2>

{/\* Map through books array and use a key for each item \*/}

{books.map(book => (

<div key={book.id} className="item-container">

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

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

{/\* Example of Conditional Rendering: Ternary Operator \*/}

{book.price > 700 ? <span className="highlight">Bestseller!</span> : null}

</div>

))}

</div>

);

}

export default BookDetails;

3) BlogDetails.js

import React from 'react';

// Data for blogs

const blogs = [

{ id: 1, title: 'React Learning', author: 'Stephen Biz', content: 'Welcome to learning React!' },

{ id: 2, title: 'Installation', author: 'Schewzdenier', content: 'You can install React from npm.' },

];

function BlogDetails() {

return (

<div className="component-container">

<h2>Blog Details</h2>

{blogs.map(blog => (

<div key={blog.id} className="item-container">

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

{/\* Example of Conditional Rendering: Logical && Operator \*/}

{blog.author && <p className="author">By: {blog.author}</p>}

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

</div>

))}

</div>

);

}

export default BlogDetails;

1. App.js

import './App.css';

import BookDetails from './BookDetails';

import BlogDetails from './BlogDetails';

import CourseDetails from './CourseDetails';

function App() {

return (

<div className="app-container">

<CourseDetails />

<BookDetails />

<BlogDetails />

</div>

);

}

export default App;

5)App.css

.app-container {

display: flex;

justify-content: space-around;

padding: 20px;

font-family: sans-serif;

}

.component-container {

width: 30%;

padding: 0 20px;

/\* Add the vertical green border \*/

border-right: 2px solid green;

}

/\* Remove the border from the last component \*/

.component-container:last-child {

border-right: none;

}

.item-container {

margin-bottom: 25px;

}

h2, h3 {

margin-bottom: 5px;

}

p {

margin-top: 0;

}

.author {

font-style: italic;

color: #555;

}

.highlight {

color: #28a745; /\* Green color for the highlight \*/

font-weight: bold;

}

Output)

