**WEEK 7 - React**

**LAB 9 :** ***Create a React Application named “cricketapp”.***  
  
*Code:*  
Src -> Components -> ListofPlayers.js

import React from 'react';

const players = [

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

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

{ name: 'KL Rahul', score: 67 },

{ name: 'Shikhar Dhawan', score: 54 },

{ name: 'Rishabh Pant', score: 72 },

{ name: 'Hardik Pandya', score: 46 },

{ name: 'Jasprit Bumrah', score: 12 },

{ name: 'Ravindra Jadeja', score: 33 },

{ name: 'Suryakumar Yadav', score: 81 },

{ name: 'Shreyas Iyer', score: 69 },

{ name: 'Ishan Kishan', score: 74 }

];

const ListOfPlayers = () => {

// Filter players with score below 70 using arrow function

const belowSeventy = players.filter(p => p.score < 70);

return (

<div className="card">

<h2>All Players (using map + destructuring)</h2>

<ul>

{players.map(({ name, score }, i) => (

<li key={i}>{i + 1}. {name} — {score}</li>

))}

</ul>

<h3>Players with score &lt; 70 (filter + arrow function)</h3>

<ul>

{belowSeventy.map(({ name, score }, i) => (

<li key={name}>{name} — {score}</li>

))}

</ul>

</div>

);

};

export default ListOfPlayers;

Src -> Components -> IndianPlayers.js

import React from 'react';

const players = [

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

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

{ name: 'KL Rahul', score: 67 },

{ name: 'Shikhar Dhawan', score: 54 },

{ name: 'Rishabh Pant', score: 72 },

{ name: 'Hardik Pandya', score: 46 },

{ name: 'Jasprit Bumrah', score: 12 },

{ name: 'Ravindra Jadeja', score: 33 },

{ name: 'Suryakumar Yadav', score: 81 },

{ name: 'Shreyas Iyer', score: 69 },

{ name: 'Ishan Kishan', score: 74 }

];

const IndianPlayers = () => {

// Split into odd / even teams (index-based). Demonstrates destructuring in map later.

const oddTeam = players.filter((\_, idx) => idx % 2 === 0); // indexes 0,2,4...

const evenTeam = players.filter((\_, idx) => idx % 2 === 1); // indexes 1,3,5...

// Example arrays for merge demo:

const T20players = ['Rohit Sharma', 'Suryakumar Yadav', 'Ishan Kishan'];

const RanjiPlayers = ['LocalStar A', 'LocalStar B', 'LocalStar C'];

// Merge using spread operator

const merged = [...T20players, ...RanjiPlayers];

// Example of array destructuring: pick first two players

const [firstPlayer, secondPlayer, ...restPlayers] = players;

return (

<div className="card">

<h2>Indian Players — Odd / Even teams</h2>

<h3>Odd Team (1st, 3rd, 5th ...)</h3>

<ul>

{oddTeam.map(({ name, score }, i) => <li key={i}>{name} — {score}</li>)}

</ul>

<h3>Even Team (2nd, 4th, 6th ...)</h3>

<ul>

{evenTeam.map(({ name, score }, i) => <li key={i}>{name} — {score}</li>)}

</ul>

<h3>Merged T20 + Ranji Players (using ...spread)</h3>

<ul>

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

</ul>

<h4>Array destructuring example</h4>

<p>First: {firstPlayer.name} — Second: {secondPlayer.name}</p>

</div>

);

};

export default IndianPlayers;

index.js

import React from 'react';

import { createRoot } from 'react-dom/client';

import App from './App';

import './index.css';

const container = document.getElementById('root');

const root = createRoot(container);

root.render(<App />);

App.js

import React from 'react';

import ListOfPlayers from './Components/ListofPlayers';

import IndianPlayers from './Components/IndianPlayers';

import './App.css';

function App() {

const flag = true; // change to false to test the other output

return (

<div className="App">

<h1>Cricket App — ES6 features demo</h1>

{flag ? (

<div className="container">

<ListOfPlayers />

<IndianPlayers />

</div>

) : (

<div className="container">

<IndianPlayers />

</div>

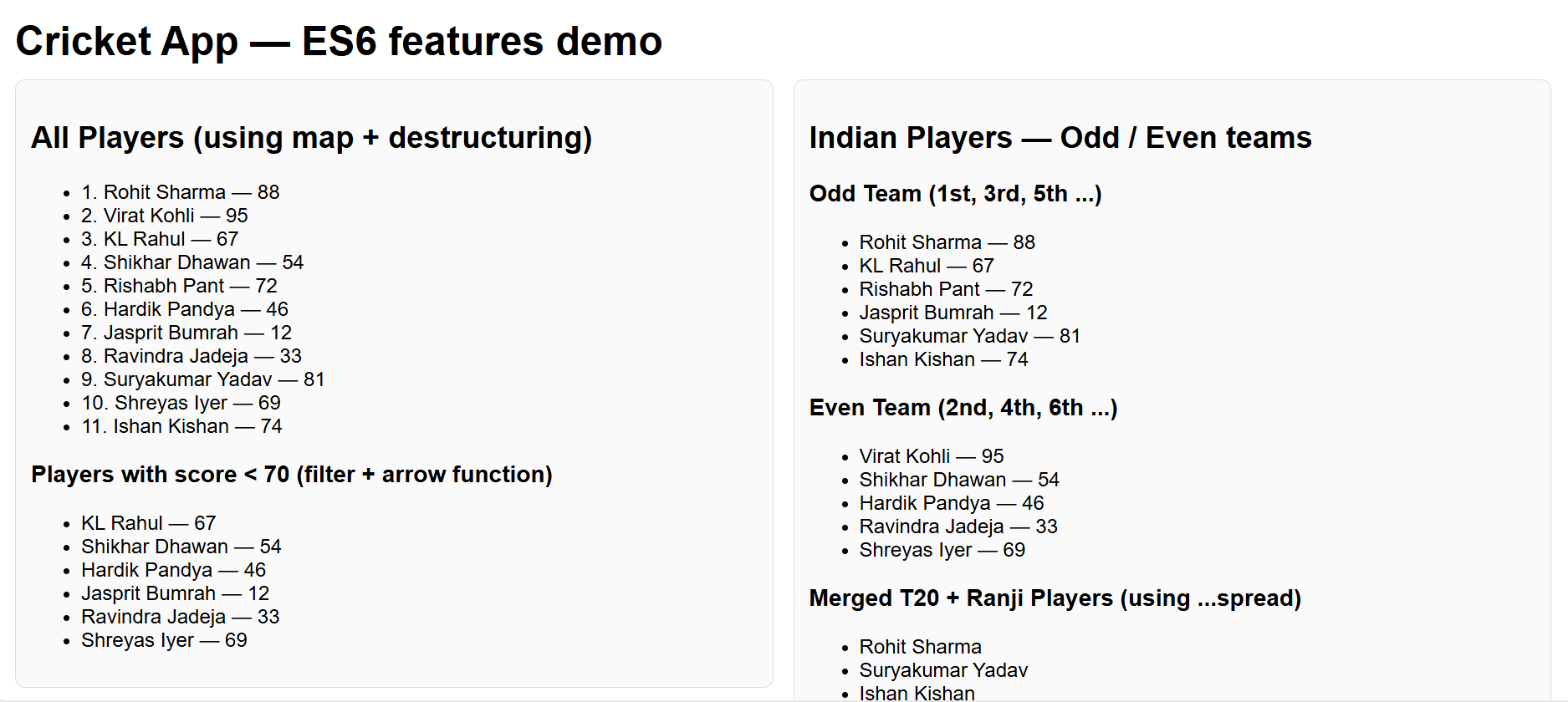
)}

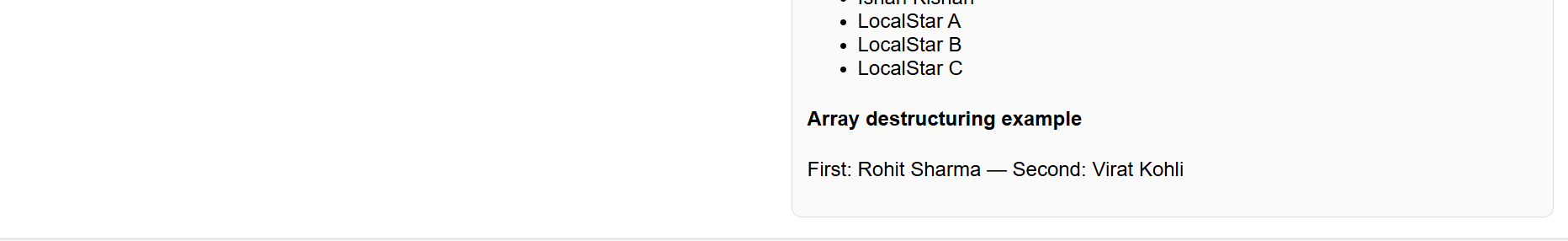
</div>

);

}

export default App;

*Output:*



**LAB 10 :** ***Create a React Application named “officespacerentalapp” which uses React JSX to create elements, attributes and renders DOM to display the page.***

*Code:*   
App.js

import React from 'react';

function App() {

// single office object

const office = { name: 'Sunrise Towers', rent: 55000, address: '123 Marina Blvd' };

// list of offices

const offices = [

{ id: 1, name: 'Sunrise Towers', rent: 55000, address: '123 Marina Blvd' },

{ id: 2, name: 'Bayview Plaza', rent: 75000, address: '45 Ocean Ave' },

{ id: 3, name: 'Lake Heights', rent: 45000, address: '78 Lake Rd' },

];

// helper for inline style based on rent

const rentColor = (r) => ({ color: r < 60000 ? 'red' : 'green' });

return (

<div style={{ padding: 20, fontFamily: 'Arial, sans-serif' }}>

{/\* Heading \*/}

<h1>Office Space Rental</h1>

{/\* Image from public folder \*/}

<img

src="/office.jpg"

alt="Office"

style={{ width: 300, height: 'auto', display: 'block', marginBottom: 12 }}

/>

{/\* Single object \*/}

<section>

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

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

<p style={rentColor(office.rent)}>Rent: ₹{office.rent}</p>

</section>

{/\* List of objects \*/}

<section>

<h3>Other office listings</h3>

<ul>

{offices.map((o) => (

<li key={o.id} style={{ marginBottom: 8 }}>

<strong>{o.name}</strong> — {o.address} —{' '}

<span style={rentColor(o.rent)}>₹{o.rent}</span>

</li>

))}

</ul>

</section>

</div>

);

}

export default App;

index.js

import React from 'react';

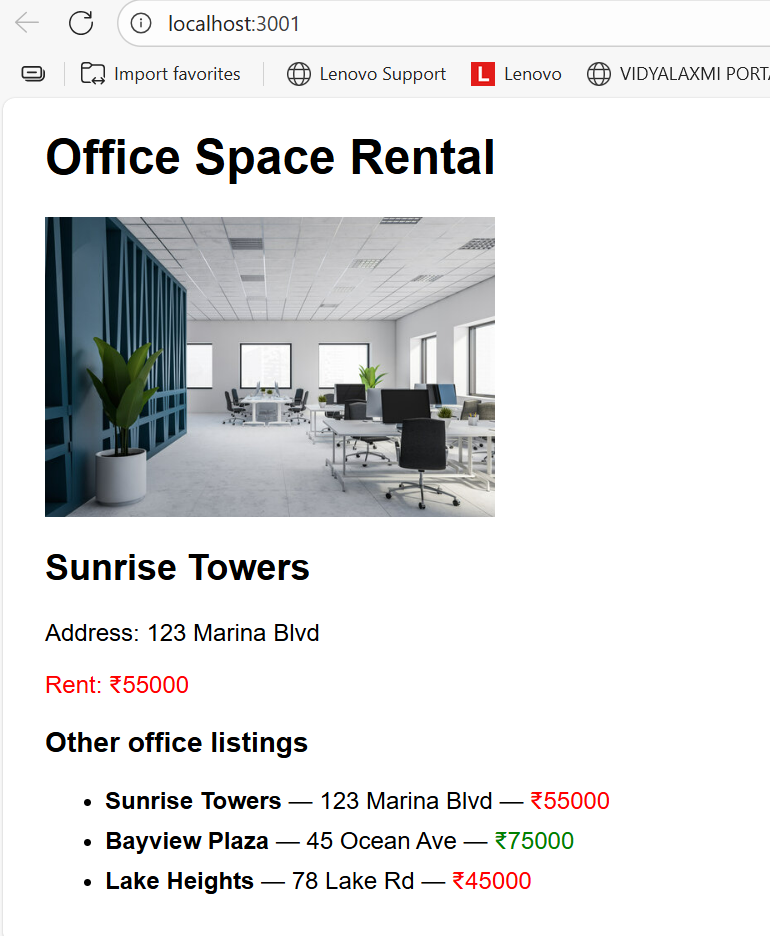
import ReactDOM from 'react-dom/client';

import App from './App';

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(<App />);

*Output:*



**LAB 12: *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.***  
*Code:*

Src -> components -> FlightList.js

// src/components/FlightList.js

import React from 'react';

import FlightCard from './FlightCard';

export default function FlightList({ flights = [], loggedIn = false, onBook, bookings = [] }) {

// Prevent rendering when there are no flights (example of returning early)

if (!flights || flights.length === 0) return <p>No flights available.</p>;

return (

<div className="flight-list">

{flights.map(f => (

<FlightCard

key={f.id}

flight={f}

loggedIn={loggedIn}

onBook={onBook}

booked={bookings.includes(f.id)}

/>

))}

</div>

);

}

FlightCard.js

import React from 'react';

export default function FlightCard({ flight, loggedIn, onBook, booked }) {

// Prevent rendering if flight data missing

if (!flight) return null;

// Element variable: compute the Book button node conditionally

const bookButton = loggedIn

? <button onClick={() => onBook(flight.id)} disabled={booked}>{booked ? 'Booked' : 'Book'}</button>

: null; // guest sees no button

return (

<div className="flight-card">

<h4>{flight.airline}</h4>

<p>{flight.from} → {flight.to}</p>

<p>Departure: {flight.date} • Fare: ₹{flight.fare}</p>

{/\* Insert the element variable \*/}

<div className="actions">

{bookButton}

</div>

</div>

);

}

GuestPage.js

import React from 'react';

import FlightList from './FlightList';

import flights from '../data/flights';

export default function GuestPage() {

return (

<div className="page guest-page">

<h2>Welcome, Guest</h2>

<p>Browse available flights. Please login to book tickets.</p>

<FlightList flights={flights} loggedIn={false} />

</div>

);

}

UserPage.js

import React from 'react';

import FlightList from './FlightList';

import flights from '../data/flights';

export default function UserPage({ bookings = [], onBook }) {

return (

<div className="page user-page">

<h2>Welcome, User</h2>

<p>Select a flight to book.</p>

<FlightList flights={flights} loggedIn={true} onBook={onBook} bookings={bookings} />

<section className="my-bookings">

<h3>My Bookings</h3>

{bookings.length === 0 ? (

<p>No bookings yet.</p>

) : (

<ul>

{bookings.map(id => {

const f = flights.find(x => x.id === id);

return <li key={id}>{f.airline} — {f.from} → {f.to} on {f.date} (₹{f.fare})</li>;

})}

</ul>

)}

</section>

</div>

);

}

Src -> data -> flight.js

const flights = [

{ id: 1, airline: 'Air India', from: 'Delhi', to: 'Mumbai', date: '2025-09-01', fare: 3500 },

{ id: 2, airline: 'IndiGo', from: 'Bengaluru', to: 'Chennai', date: '2025-09-03', fare: 2200 },

{ id: 3, airline: 'SpiceJet', from: 'Kolkata', to: 'Hyderabad', date: '2025-09-10', fare: 2800 }

];

export default flights;

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 [bookings, setBookings] = useState([]); // store flight ids

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

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

const handleBook = (flightId) => {

setBookings(prev => prev.includes(flightId) ? prev : [...prev, flightId]);

};

return (

<div className="app">

<header className="app-header">

<h1>ticketbookingapp</h1>

{isLoggedIn

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

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

}

</header>

{/\* Conditional rendering: show Guest or User page \*/}

{isLoggedIn

? <UserPage bookings={bookings} onBook={handleBook} />

: <GuestPage />

}

</div>

);

}

export default App;

App.css

.app { font-family: Arial, sans-serif; padding: 20px; max-width: 900px; margin: 0 auto; }

.app-header { display:flex; justify-content:space-between; align-items:center; margin-bottom:16px; }

.page { border: 1px solid #eee; padding: 12px; border-radius: 6px; margin-bottom: 12px; }

.flight-card { border:1px solid #ddd; padding:10px; border-radius:6px; margin:8px 0; }

.actions { margin-top:8px; }

button { padding:6px 10px; cursor:pointer; }

index.js

import React from 'react';

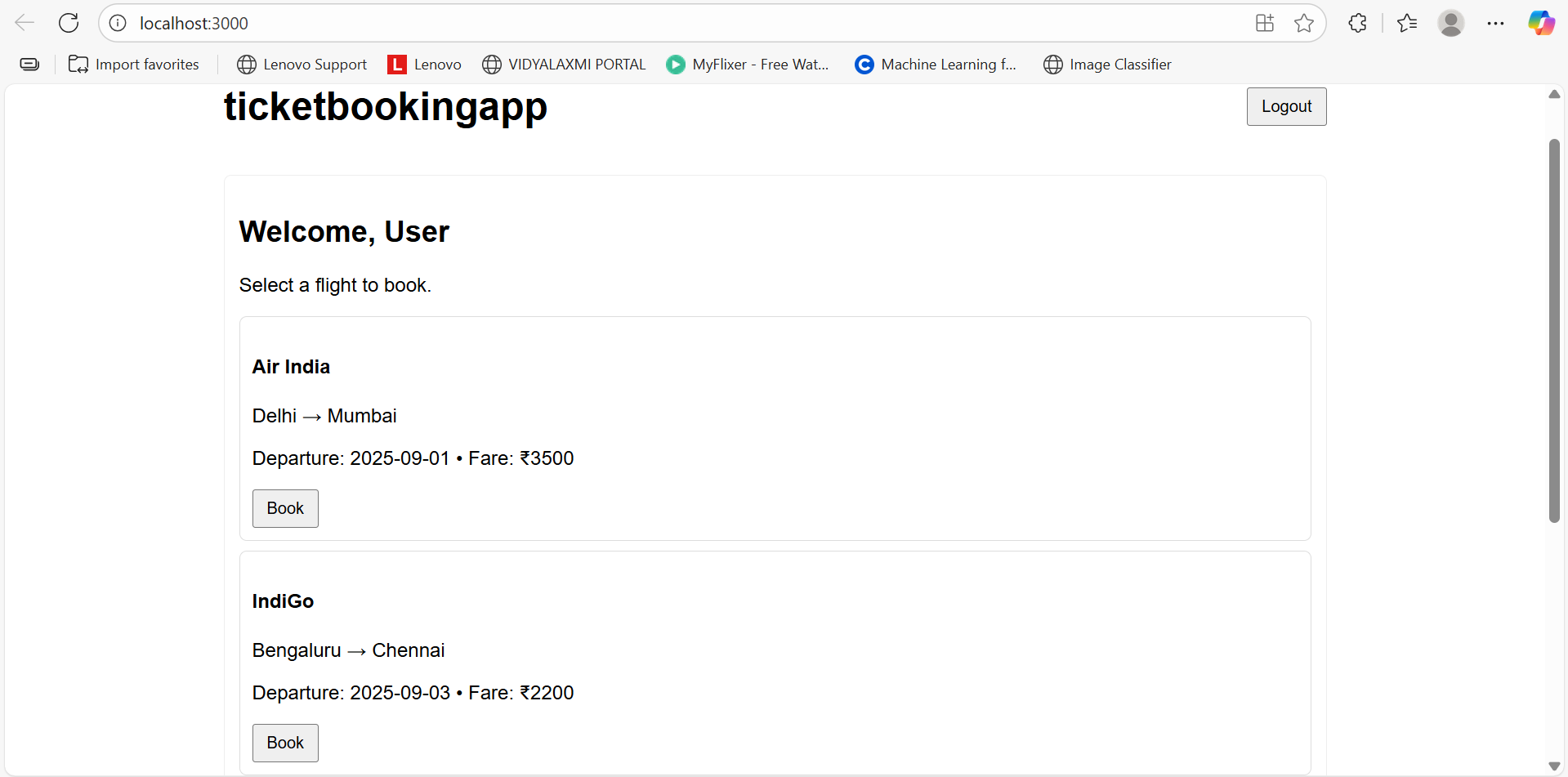
import { createRoot } from 'react-dom/client';

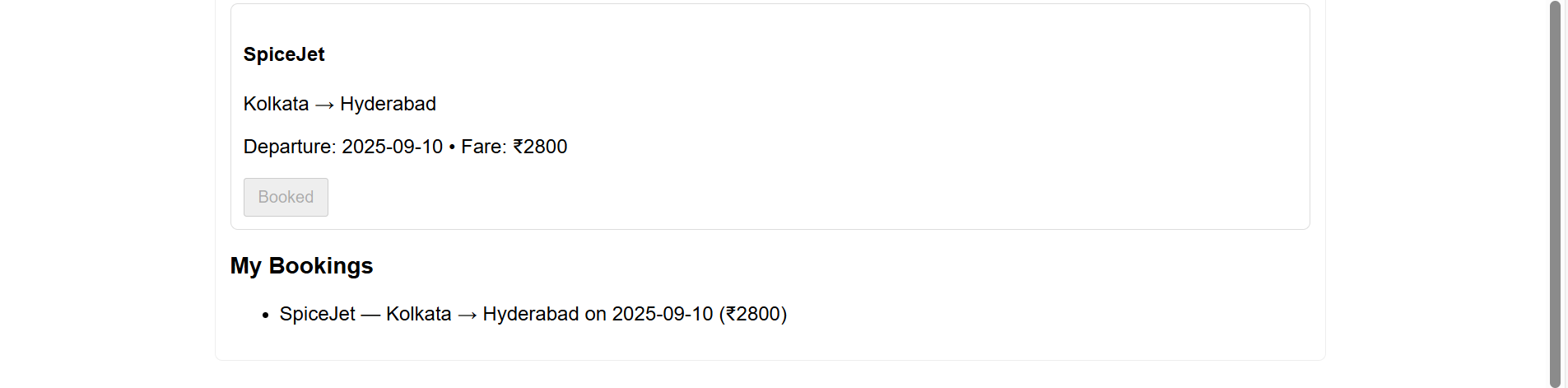
import App from './App';

import './App.css';

createRoot(document.getElementById('root')).render(<App />);

*Output:*





**LAB 13:**  ***Create a React App named “bloggerapp” in with 3 components.  
 Book Details  
 Blog Details  
 Course Details  
 Implement this with as many ways possible of Conditional Rendering.***  
*Code:*  
src// components // BlogDetails.js

import React from "react";

const BlogDetails = ({ blogs = [] }) => {

// logical && short-circuit

return (

<section>

<h2>Blog Details</h2>

{/\* logical AND: if blogs exist show list \*/}

{blogs.length > 0 && (

<div>

{blogs.map(b => (

<article key={b.id}>

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

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

</article>

))}

</div>

)}

{/\* ternary example (alternate content when empty) \*/}

{blogs.length === 0 ? <p>No blogs right now</p> : null}

</section>

);

};

export default BlogDetails;

BookDetails.js

import React from "react";

import BookItem from "./BookItem";

const BookDetails = ({ books }) => {

// element variable pattern

if (!books || books.length === 0) {

return <p>No books available</p>;

}

return (

<section>

<h2>Book Details</h2>

<ul>

{/\* map() with stable key (id) and extracted BookItem component \*/}

{books.map(book => (

<BookItem key={book.id} book={book} />

))}

</ul>

</section>

);

};

export default BookDetails;

BookItem.js

import React from "react";

const BookItem = ({ book }) => {

return (

<li>

<strong>{book.title}</strong> — {book.author}

{/\* ternary operator for availability \*/}

{book.available ? <span> (Available)</span> : <span> (Out of stock)</span>}

</li>

);

};

export default BookItem;

CourseDetails.js

import React from "react";

const CourseDetails = ({ courses = [] }) => {

return (

<section>

<h2>Course Details</h2>

<ul>

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

// prefer c.id; fallback to idx only if absolutely needed

<li key={c.id ?? idx}>

{c.name} — {c.duration}

</li>

))}

</ul>

</section>

);

};

export default CourseDetails;

src// data.js

const data = {

books: [

{ id: 1, title: "React Basics", author: "Alice", available: true },

{ id: 2, title: "Advanced React", author: "Bob", available: false }

],

blogs: [

{ id: 1, title: "Why React?", content: "React makes UIs..." },

{ id: 2, title: "Hooks Deep Dive", content: "useState and useEffect..." }

],

courses: [

{ id: 1, name: "React Mastery", duration: "4 weeks" },

{ id: 2, name: "JS Essentials", duration: "3 weeks" }

]

};

export default data;

App.js

import React, { useState } from "react";

import BookDetails from "./components/BookDetails";

import BlogDetails from "./components/BlogDetails";

import CourseDetails from "./components/CourseDetails";

import data from "./data";

function App() {

const [showBooks, setShowBooks] = useState(true);

const [showBlogs, setShowBlogs] = useState(true);

return (

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

<h1>Blogger App</h1>

<button onClick={() => setShowBooks(prev => !prev)}>

{showBooks ? "Hide Books" : "Show Books"}

</button>

<button onClick={() => setShowBlogs(prev => !prev)} style={{ marginLeft: 8 }}>

{showBlogs ? "Hide Blogs" : "Show Blogs"}

</button>

{/\* Conditional rendering examples \*/}

{showBooks && <BookDetails books={data.books} />}

{showBlogs ? <BlogDetails blogs={data.blogs} /> : <p>No blogs to display</p>}

<CourseDetails courses={data.courses} />

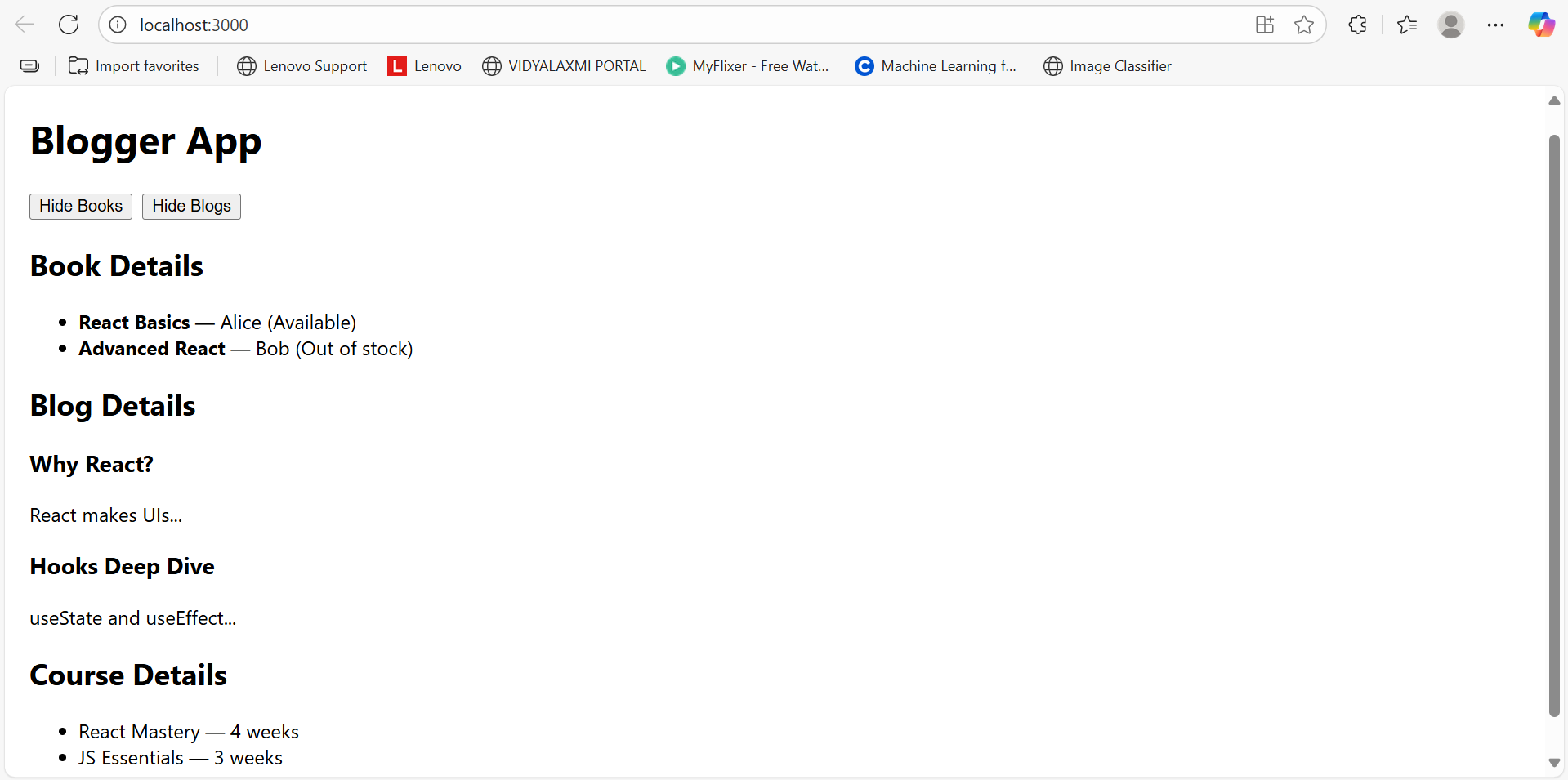
</div>

);

}

export default App;

*Output:*



**LAB 15: *Create a React App named “ticketraisingapp” which will help to raise a complaint and get it resolved.  
Create a component named “ComplaintRegister” with a form containing a textbox to enter the employee name and a textarea to enter the complaint. Use “handleSubmit” event of the button to submit the complaint and generate a Reference number for further follow ups in the alert box.***

*Code:*  
ComplaintRegister.js

import React, { useState } from 'react';

function ComplaintRegister() {

const [name, setName] = useState('');

const [complaint, setComplaint] = useState('');

const handleSubmit = (e) => {

e.preventDefault();

// Basic validation

if (name.trim() === '' || complaint.trim() === '') {

alert('Please fill in both the employee name and the complaint.');

return;

}

// Generate a reference number (simple, readable)

const ref = 'REF-' + Math.random().toString(36).substring(2, 8).toUpperCase();

// Show reference number in alert for follow-up (as per lab)

alert(`Complaint submitted successfully!\nReference number: ${ref}`);

// Reset form

setName('');

setComplaint('');

};

return (

<div className="complaint-register">

<h2>Raise a Complaint</h2>

<form onSubmit={handleSubmit}>

<div className="form-row">

<label htmlFor="empName">Employee Name</label>

<input

id="empName"

type="text"

value={name}

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

placeholder="Enter employee name"

/>

</div>

<div className="form-row">

<label htmlFor="complaint">Complaint</label>

<textarea

id="complaint"

rows="5"

value={complaint}

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

placeholder="Describe your complaint"

/>

</div>

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

</form>

</div>

);

}

export default ComplaintRegister;

App.js

import React from 'react';

import ComplaintRegister from './ComplaintRegister';

import './App.css';

function App() {

return (

<div className="App">

<ComplaintRegister />

</div>

);

}

export default App;

App.css

.App {

font-family: Arial, sans-serif;

padding: 2rem;

}

.complaint-register {

max-width: 600px;

margin: 0 auto;

border: 1px solid #e0e0e0;

padding: 1.2rem;

border-radius: 8px;

box-shadow: 0 2px 6px rgba(0,0,0,0.03);

}

.complaint-register h2 {

margin-top: 0;

}

.form-row {

margin-bottom: 1rem;

}

.form-row label {

display: block;

margin-bottom: .4rem;

font-weight: 600;

}

.form-row input,

.form-row textarea {

width: 100%;

padding: .5rem;

border: 1px solid #ccc;

border-radius: 4px;

box-sizing: border-box;

}

button[type="submit"] {

padding: .6rem 1rem;

border: none;

border-radius: 4px;

cursor: pointer;

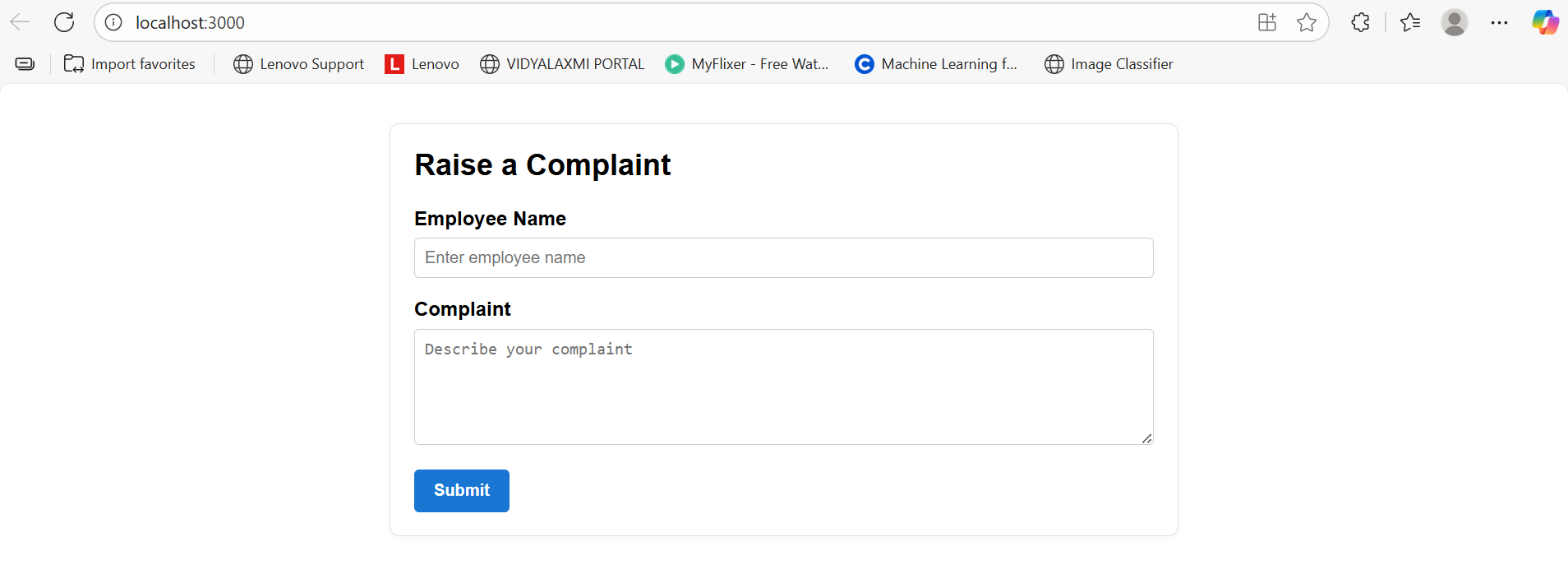
background-color: #1976d2;

color: white;

font-weight: 600;

}

*Output:*



**LAB 16: *Create a React App named “mailregisterapp” which will have a component named “register.js”. Create a form which accepts the name, email and password and validate the fields as per the following:  
1. Name should have atleast 5 characters  
2. Email should have @ and .  
3. Password should have atleast 8 characters.  
Ensure that validations are implemented through eventhandle and eventsubmit of a form.***

*Code:*register.js  
import React, { useState } from 'react';

export default function Register() {

const [form, setForm] = useState({ name: '', email: '', password: '' });

const [errors, setErrors] = useState({});

const [submitted, setSubmitted] = useState(false);

// Validate a single field (used on change) and also to validate all on submit

const validateField = (fieldName, value) => {

if (fieldName === 'name') {

if (!value.trim()) return 'Name is required';

if (value.trim().length < 5) return 'Name must be at least 5 characters';

return '';

}

if (fieldName === 'email') {

if (!value.trim()) return 'Email is required';

// simple email check ensuring at least '@' and '.'

if (!/\S+@\S+\.\S+/.test(value)) return 'Email must contain "@" and "." and be valid';

return '';

}

if (fieldName === 'password') {

if (!value) return 'Password is required';

if (value.length < 8) return 'Password must be at least 8 characters';

return '';

}

return '';

};

// Event handler (eventhandle) — updates state and does per-field validation

const handleChange = (e) => {

const { name, value } = e.target;

setForm(prev => ({ ...prev, [name]: value }));

// immediate per-field validation

setErrors(prev => ({ ...prev, [name]: validateField(name, value) }));

setSubmitted(false);

};

// Event submit (eventsubmit) — validate all fields, prevent default submit

const handleSubmit = (e) => {

e.preventDefault();

const newErrors = {};

Object.keys(form).forEach(key => {

const err = validateField(key, form[key]);

if (err) newErrors[key] = err;

});

setErrors(newErrors);

if (Object.keys(newErrors).length === 0) {

// All good

setSubmitted(true);

// optionally send `form` to server here

// clear form if desired:

// setForm({ name: '', email: '', password: '' });

} else {

setSubmitted(false);

}

};

return (

<div className="register-container">

<h2>Register</h2>

<form onSubmit={handleSubmit} noValidate>

<div className="form-group">

<label htmlFor="name">Name</label>

<input

id="name"

name="name"

type="text"

value={form.name}

onChange={handleChange}

placeholder="Enter name (min 5 chars)"

/>

{errors.name && <div className="error">{errors.name}</div>}

</div>

<div className="form-group">

<label htmlFor="email">Email</label>

<input

id="email"

name="email"

type="email"

value={form.email}

onChange={handleChange}

placeholder="example@domain.com"

/>

{errors.email && <div className="error">{errors.email}</div>}

</div>

<div className="form-group">

<label htmlFor="password">Password</label>

<input

id="password"

name="password"

type="password"

value={form.password}

onChange={handleChange}

placeholder="Min 8 characters"

/>

{errors.password && <div className="error">{errors.password}</div>}

</div>

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

</form>

{submitted && (

<div className="success">

Registration successful! (Name: {form.name}, Email: {form.email})

</div>

)}

</div>

);

}

App.js  
import React from 'react';

import './App.css';

import Register from './register';

function App() {

return (

<div className="App">

<Register />

</div>

);

}

export default App;

App.css  
.App { font-family: Arial, sans-serif; padding: 2rem; background:#f7f7f7; min-height:100vh; }

.register-container { max-width:420px; margin: 2rem auto; background:white; padding:1.5rem; border-radius:8px; box-shadow:0 2px 6px rgba(0,0,0,0.06); }

.form-group { margin-bottom:1rem; }

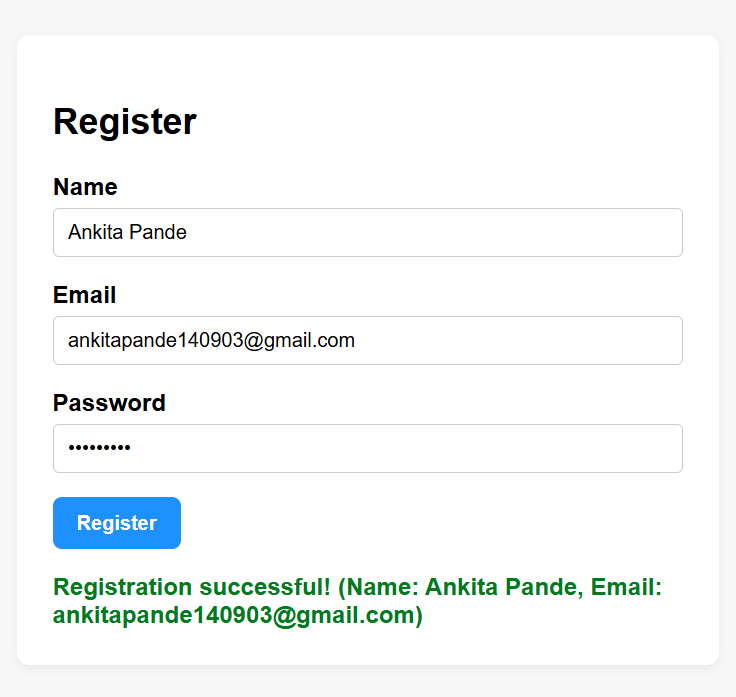
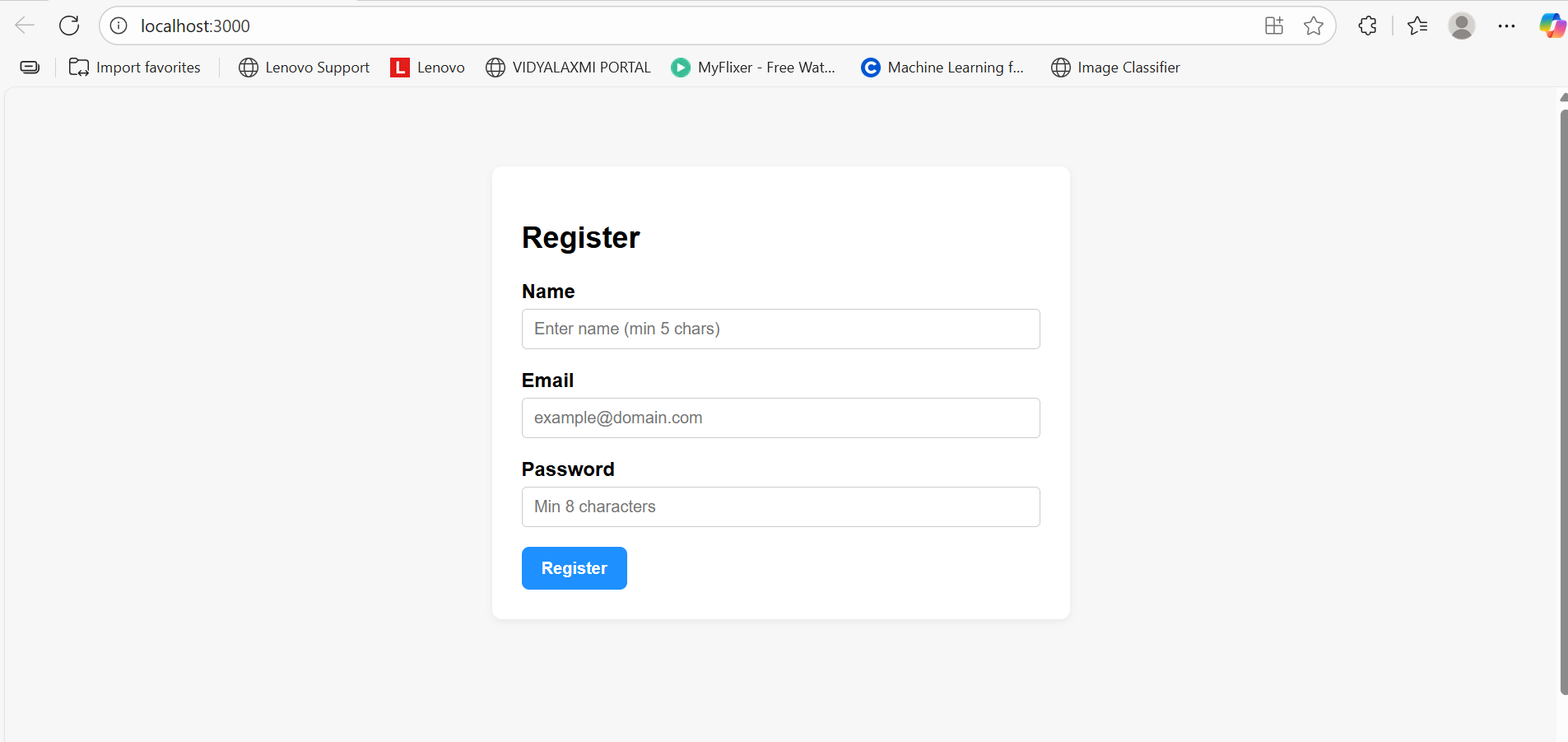
label { display:block; margin-bottom:.3rem; font-weight:600; }

input { width:100%; padding:.5rem .6rem; border:1px solid #ccc; border-radius:4px; box-sizing:border-box; }

button { padding:.6rem 1rem; border:none; border-radius:6px; background:#1e90ff; color:white; cursor:pointer; font-weight:600; }

.error { color:#b00020; margin-top:.35rem; font-size:.9rem; }

.success { margin-top:1rem; color:#007a1f; font-weight:700; }

*Output:  
***LAB 17: *Create a React Application “fetchuserapp” which will retrieve the user details from*** [***https://api.randomuser.me/***](https://api.randomuser.me/) ***and display the title, firstname and image of a user.  
Create a component named “Getuser” and in the asynchronous method “ComponentDidMount ()” invoke the URL using fetch method and the response can be displayed in the render method of the component.****Code:*Getuser.js  
import React, { Component } from 'react';

export default class Getuser extends Component {

state = {

user: null,

loading: true,

error: null,

};

// fetch in lifecycle

async componentDidMount() {

try {

const res = await fetch('https://api.randomuser.me/');

if (!res.ok) throw new Error(`HTTP ${res.status}`);

const json = await res.json();

const user = json.results?.[0] ?? null;

this.setState({ user, loading: false });

} catch (err) {

this.setState({ error: err.message, loading: false });

}

}

render() {

const { user, loading, error } = this.state;

if (loading) return <div>Loading user…</div>;

if (error) return <div>Error: {error}</div>;

if (!user) return <div>No user found</div>;

const { name, picture } = user; // name.title, name.first, picture.large

return (

<div className="user-card">

<h3>{name.title} {name.first}</h3>

<img src={picture.large} alt={`${name.title} ${name.first}`} />

</div>

);

}

}

App.js  
import React from 'react';

import './App.css';

import Getuser from './Getuser';

function App() {

return (

<div className="App">

<h1>Fetch User App</h1>

<Getuser />

</div>

);

}

export default App;

App.css  
.App {

font-family: Arial, sans-serif;

padding: 2rem;

text-align: center;

background: #f5f7fa;

min-height: 100vh;

}

.user-card {

display: inline-block;

padding: 1rem 1.5rem;

border-radius: 10px;

background: white;

box-shadow: 0 6px 18px rgba(0,0,0,0.06);

margin-top: 1rem;

text-align: center;

}

.user-card img {

border-radius: 50%;

width: 120px;

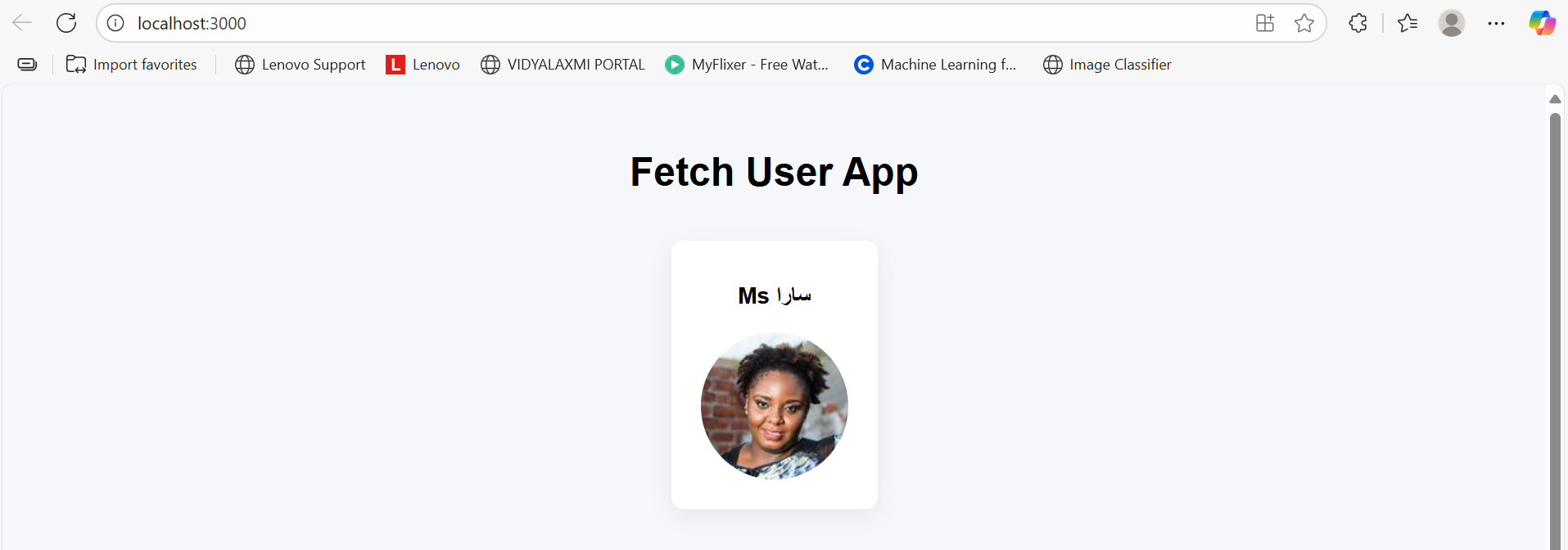
height: 120px;

object-fit: cover;

display:block;

margin: 0.5rem auto;

}

*Output:*  


**Ankita Pande  
ID-6362381**