|  |  |
| --- | --- |
| **NAME-Konika Khurana** | **Sub-Full Stack** |
| **UID-23ICS12925** | **Class-622A** |

# Practice1 CODE-

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<title>Product Card</title>

<script src="[https://unpkg.com/react@18/umd/react.development.js"](https://unpkg.com/react%4018/umd/react.development.js) crossorigin></script>

<script src="[https://unpkg.com/react-dom@18/umd/react-dom.development.js](https://unpkg.com/react-dom%4018/umd/react-dom.development.js)" crossorigin></script>

<script src="[https://unpkg.com/@babel/standalone/babel.min.js"](https://unpkg.com/%40babel/standalone/babel.min.js)></script>

<style> .container { border: 2px solid #333; padding: 20px; text-

align: center; width: 80%; margin: 20px auto;

}

.product-list { display: flex; justify- content: center; gap: 20px; flex-wrap: wrap;

}

.card { border: 1px solid #ddd; border-radius: 10px; padding: 15px; width: 200px; text-align: center; margin: 10px; box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);

}

.in-stock {

color: green; font- weight: bold;

}

.out-stock { color: red; font- weight: bold;

}

</style>

</head>

<body>

<div id="root"></div>

<script type="text/babel"> const ProductCard = ({ name, price, status }) => {

return (

<div className="card">

<h3>{name}</h3>

<p>Price: ${price}</p>

<p className={status === "In Stock" ? "in-stock" : "out-stock"}> Status: {status}

</p>

</div>

);

};

function App() { return (

<div className="container">

<h2>Products List</h2>

<div className="product-list">

<ProductCard name="Wireless Mouse" price="25.99" status="In Stock" />

<ProductCard name="Keyboard" price="45.5" status="Out of Stock" />

<ProductCard name="Monitor" price="199.99" status="In Stock" />

</div>

</div>

);

}

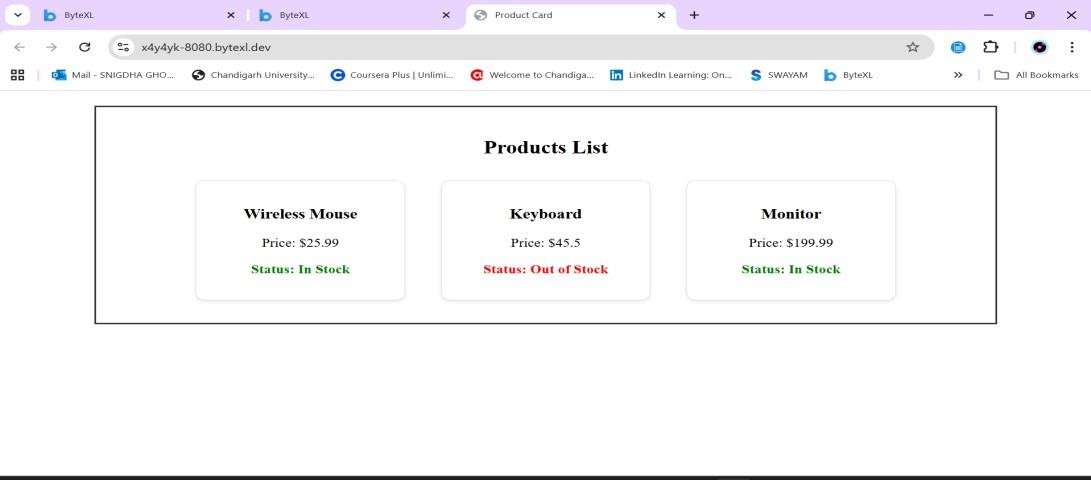
const root = ReactDOM.createRoot(document.getElementById("root")); root.render(<App />);

</script>

</body>

</html>

# OUTPUT



**Practice2 CODE-**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Library Management</title>

<!-- React, ReactDOM, and Babel -->

<script src="[https://unpkg.com/react@18/umd/react.development.js"><](https://unpkg.com/react%4018/umd/react.development.js)/script>

<script src="https://unpkg.com/react- dom@18/umd/reactdom.development.js"></script>

<script src="[https://unpkg.com/@babel/standalone/babel.min.js"](https://unpkg.com/%40babel/standalone/babel.min.js)></script>

</head>

<body>

<div id="root"></div>

<script type="text/babel"> const { useState } = React;

function Library() { const [books, setBooks] = useState([

{ title: "1984", author: "George Orwell" },

{ title: "The Great Gatsby", author: "F. Scott Fitzgerald" },

{ title: "To Kill a Mockingbird", author: "Harper Lee" },

]);

const [search, setSearch] = useState(""); const [newTitle, setNewTitle] = useState(""); const [newAuthor, setNewAuthor] = useState("");

const addBook = () => { if (newTitle.trim() && newAuthor.trim()) { setBooks([...books, { title: newTitle, author: newAuthor }]); setNewTitle(""); setNewAuthor("");

}

};

const removeBook = (index) => { setBooks(books.filter((\_, i) => i !== index));

};

const filteredBooks = books.filter( (book) =>

book.title.toLowerCase().includes(search.toLowerCase()) || book.author.toLowerCase().includes(search.toLowerCase())

);

return (

<div style={styles.container}>

<h2>Library Management</h2>

<input type="text" placeholder="Search by title or author" value={search} onChange={(e) => setSearch(e.target.value)} style={styles.input}

/>

<div style={styles.addBookSection}>

<input type="text" placeholder="New book title" value={newTitle} onChange={(e) => setNewTitle(e.target.value)} style={styles.inputSmall}

/> <input type="text" placeholder="New book author" value={newAuthor} onChange={(e) => setNewAuthor(e.target.value)} style={styles.inputSmall}

/>

<button onClick={addBook} style={styles.button}> Add Book

</button>

</div>

<ul style={styles.bookList}>

{filteredBooks.map((book, index) => (

<li key={index} style={styles.bookItem}>

<span>

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

</span>

<button onClick={() => removeBook(index)} style={styles.removeBtn}

>

Remove

</button>

</li>

))}

</ul>

</div>

);

}

const styles = {

container: { border: "2px

solid #333", padding:

"20px", width: "80%", margin: "20px auto",

}, input: {

width: "100%", padding: "8px", marginBottom:

"10px", border: "1px solid #ccc",

},

addBookSection: { display: "flex", gap: "10px", marginBottom: "15px",

},

inputSmall: { flex: 1, padding: "8px", border: "1px solid #ccc",

}, button: { padding: "8px 12px", background: "#333",

color: "#fff", border: "none", cursor: "pointer",

},

bookList: { listStyle: "none",

padding: 0,

},

bookItem: { display: "flex", justifyContent: "space-between", padding: "8px", border: "1px solid #ddd", marginBottom:

"5px", borderRadius: "5px",

},

removeBtn: { background: "#f44336", color: "white", border:

"none", padding: "5px

10px", cursor:

"pointer", borderRadius: "4px",

},

};

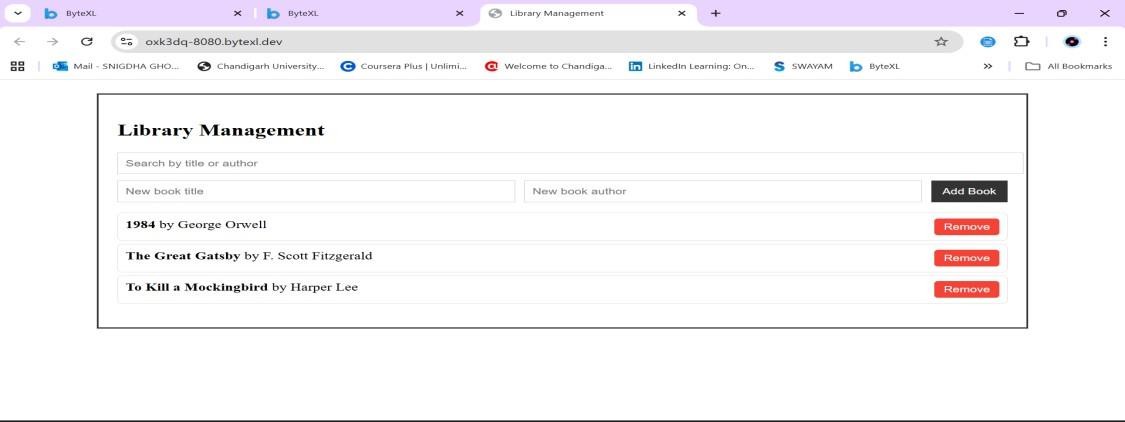
ReactDOM.createRoot(document.getElementById("root")).render(<Library />);

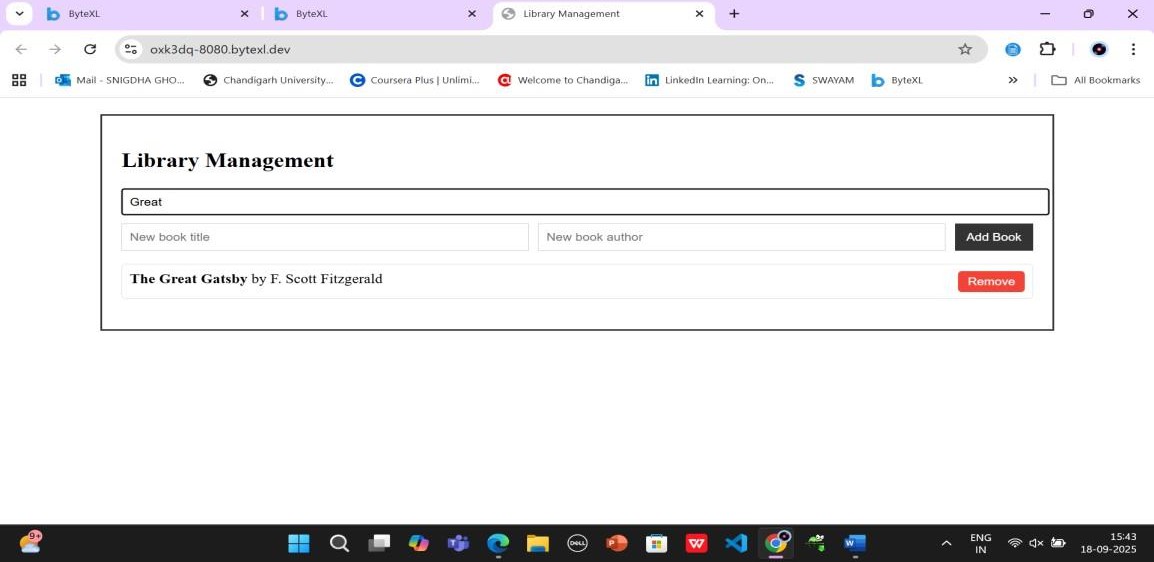
</script>

</body>

</html>

# OUTPUT

****



**Practice3 CODE-**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<title>Person / Student / Teacher Demo</title>

<meta name="viewport" content="width=device-width,initial-scale=1" />

<style>

body { font-family: system-ui, -apple-system, "Segoe UI", Roboto, Arial; padding: 20px; } h1 { margin-

bottom: 8px; }

pre { background:#f6f8fa; border:1px solid #e1e4e8; padding:12px; white- space:pre-wrap; }

.log { margin-top:12px; color:#444; }

</style>

</head>

<body>

<h1>Person / Student / Teacher Demo</h1>

<p>This page runs the ES6 classes and prints results below (also logged to console).</p>

<div>

<strong>Output:</strong>

<pre id="output">(running...)</pre>

</div>

<script>

// --- Person base class class Person { constructor(name, age) { this.name = name; this.age = age;

}

displayInfo() { return `Name:

${this.name}, Age: ${this.age}`;

}

}

// --- Student subclass class Student extends Person { constructor(name, age, course) { super(name, age); this.course = course;

}

// override displayInfo() { return

`${super.displayInfo()}, Course: ${this.course}`;

}

}

// --- Teacher subclass

class Teacher extends Person { constructor(name, age, subject) {

super(name, age); this.subject = subject;

}

// override displayInfo() { return

`${super.displayInfo()}, Subject: ${this.subject}`;

}

}

// --- create instances and display results const student1 = new Student("Alice", 20, "Computer Science"); const teacher1 = new Teacher("Mr. Smith", 45, "Mathematics");

const outText = [ student1.displayInfo(), teacher1.displayInfo()

].join("\n");

// show on page document.getElementById("output").textContent = outText;

// also log to console (open devtools to see) console.log(outText);

</script>

</body>

</html>

**OUTPUT**

