Case Study 1: Online Food Ordering System

Scenario:

A customer browses a food delivery app.

They can add food items to a cart, see the total cost, and finally place an order.

//App.Js

```
import React, { useState } from "react";
import Menu from "./Menu";
import Cart from "./Cart";


const App = () => {
  const [cart, setCart] = useState([]);


  const addToCart = (food) => {
    setCart((prev) => {
      const existing = prev.find((item) => item.id === food.id);
      if (existing) {
        // increase quantity if already in cart
        return prev.map((item) =>
          item.id === food.id
            ? { ...item, qty: item.qty + 1 }
            : item
        );
      } else {
        // add new item
        return [...prev, { ...food, qty: 1 }];
      }
    });
  };
```

```
  return (
    <div style={{ display: "flex", gap: "50px", padding: "20px" }}>
      <Menu addToCart={addToCart} />
      <Cart cart={cart} />
    </div>
  );
};

export default App;
```

//Menu.js

```
import React from "react";
import FoodCard from "./FoodCard";

const foods = [
  { id: 1, name: "Pizza", price: 250, description: "Cheesy and delicious" },
  { id: 2, name: "Burger", price: 150, description: "Crispy and juicy" },
  { id: 3, name: "Pasta", price: 200, description: "Creamy and spicy" }
];

const Menu = ({ addToCart }) => {
  return (
    <div>
      <h2>🍴 Menu</h2>
      {foods.map((food) => (
        <FoodCard key={food.id} food={food} addToCart={addToCart} />
      ))}
    </div>
  );
};
```

```
export default Menu;
```

//FoodCard.js

```
import React from "react";

const FoodCard = ({ food, addToCart }) => {
  return (
    <div style={{
      border: "1px solid gray",
      borderRadius: "10px",
      padding: "10px",
      marginBottom: "10px",
      width: "200px"
    }}>
      <h3>{food.name}</h3>
      <p>{food.description}</p>
      <p>₹{food.price}</p>
      <button onClick={() => addToCart(food)}>Add to Cart</button>
    </div>
  );
};

export default FoodCard;
```

//Cart.js

```
import React from "react";

const Cart = ({ cart }) => {
  const total = cart.reduce((sum, item) => sum + item.price * item.qty, 0);

  return (
```

```
    <div>
      <h2> Cart</h2>
      {cart.length === 0 && <p>No items in cart.</p>}
      <ul>
        {cart.map((item) => (
          <li key={item.id}>
            {item.name} × {item.qty} = ₹{item.price * item.qty}
          </li>
        ))}
      </ul>
      <h3>Total: ₹{total}</h3>
    </div>
  );
};

export default Cart;
```

Case Study 2:

E-Learning Platform (Course Enrollment)

Scenario: A student logs into an e-learning site, views available courses, and enrolls in some.

//App.js

```
import React, { useState } from "react";
import CourseList from "./CourseList";
import EnrolledCourses from "./EnrolledCourses";

const App = () => {
  const [enrolled, setEnrolled] = useState([]);

  const handleEnroll = (course) => {
```

```
  setEnrolled((prev) => {
    if (prev.find((c) => c.id === course.id)) {
      return prev; // already enrolled
    }
    return [...prev, { ...course, progress: 0, favorite: false }];
  });
};

const toggleFavorite = (id) => {
  setEnrolled((prev) =>
    prev.map((c) =>
      c.id === id ? { ...c, favorite: !c.favorite } : c
    )
  );
};

const updateProgress = (id, value) => {
  setEnrolled((prev) =>
    prev.map((c) =>
      c.id === id ? { ...c, progress: value } : c
    )
  );
};

return (
  <div style={{ display: "flex", gap: "50px", padding: "20px" }}>
    <CourseList enrolled={enrolled} handleEnroll={handleEnroll} />
    <EnrolledCourses
      enrolled={enrolled}
      toggleFavorite={toggleFavorite}
      updateProgress={updateProgress}
```

```
      />
    </div>
  );
};

export default App;
```

//CourseList.js

```
import React from "react";
import CourseCard from "./CourseCard";

const courses = [
  { id: 1, title: "React Basics", author: "John Doe", duration: "3h" },
  { id: 2, title: "Node.js Fundamentals", author: "Jane Smith", duration: "5h" },
  { id: 3, title: "Database Design", author: "Alex Lee", duration: "4h" },
];

const CourseList = ({ enrolled, handleEnroll }) => {
  return (
    <div>
      <h2> Available Courses</h2>
      {courses.map((course) => (
        <CourseCard
          key={course.id}
          course={course}
          isEnrolled={!!enrolled.find((c) => c.id === course.id)}
          handleEnroll={handleEnroll}
        />
      ))}
    </div>
  );
```

```
};

export default CourseList;
```

//CourseCard.js

```
import React from "react";

const CourseCard = ({ course, isEnrolled, handleEnroll }) => {
  return (
    <div
      style={{
        border: "1px solid gray",
        borderRadius: "10px",
        padding: "10px",
        marginBottom: "10px",
        width: "250px",
      }}
    >
      <h3>{course.title}</h3>
      <p>{course.author}</p>
      <p>⏱️ Duration: {course.duration}</p>
      <button
        onClick={() => handleEnroll(course)}
        disabled={isEnrolled}
      >
        {isEnrolled ? " Already Enrolled" : "Enroll"}
      </button>
    </div>
  );
};
```

```
export default CourseCard;
```

//EnrolledCourses.js

```
import React from "react";

const EnrolledCourses = ({ enrolled, toggleFavorite, updateProgress }) => {
  return (
    <div>
      <h2> Enrolled Courses</h2>
      {enrolled.length === 0 && <p>No courses enrolled yet.</p>}
      {enrolled.map((course) => (
        <div
          key={course.id}
          style={{
            border: "1px solid blue",
            borderRadius: "10px",
            padding: "10px",
            marginBottom: "10px",
            width: "300px",
          }}
        >
          <h3>
            {course.title}{" "}
            {course.favorite ? " ⭐ " : ""}
          </h3>
          <p>Progress: {course.progress}%</p>
          <input
            type="range"
            min="0"
            max="100"
            value={course.progress}
```

```
          onChange={(e) =>
            updateProgress(course.id, Number(e.target.value))
          }
        />
        <br />
        <button onClick={() => toggleFavorite(course.id)}>
          {course.favorite ? "Unfavorite" : "Mark Favorite"}
        </button>
      </div>
    ))}
  </div>
 );
};

export default EnrolledCourses;
```

Case Study 3:

Movie Ticket Booking

Scenario: A user books tickets for a movie by selecting seats and confirming the booking.

//App.js

```
import React, { useState } from "react";
import SeatSelector from "./SeatSelector";
import Summary from "./Summary";

const App = () => {
 const movie = {
  title: "Inception",
  timing: "7:00 PM",
  price: 150,
 };
```

```
const [selectedSeats, setSelectedSeats] = useState([]);
const [isConfirmed, setIsConfirmed] = useState(false);

const toggleSeat = (seat) => {
  setSelectedSeats((prev) =>
    prev.includes(seat)
      ? prev.filter((s) => s !== seat) // remove seat if already selected
      : [...prev, seat] // add seat if not selected
  );
};

const confirmBooking = () => {
  if (selectedSeats.length === 0) {
    alert("Please select at least one seat!");
    return;
  }
  setIsConfirmed(true);
};

return (
  <div style={{ padding: "20px", fontFamily: "Arial" }}>
    <h1> {movie.title}</h1>
    <p> Show Time: {movie.timing}</p>
    <p> Price per Seat: ₹{movie.price}</p>

    <SeatSelector
      movie={movie}
      selectedSeats={selectedSeats}
      toggleSeat={toggleSeat}
    />
```

```
      <Summary
        movie={movie}
        selectedSeats={selectedSeats}
        isConfirmed={isConfirmed}
      />

      {!isConfirmed && (
        <button
          style={{
            marginTop: "20px",
            padding: "10px 20px",
            backgroundColor: "green",
            color: "white",
            border: "none",
            borderRadius: "5px",
            cursor: "pointer",
          }}
          onClick={confirmBooking}
        >
          Confirm Booking
        </button>
      )}
    </div>
  );
};

export default App;
```

//SeatSelector.js

```
import React from "react";
```

```
const seats = ["A1", "A2", "A3", "A4", "A5", "B1", "B2", "B3", "B4", "B5"];

const SeatSelector = ({ selectedSeats, toggleSeat }) => {
  return (
    <div>
      <h2> Select Seats</h2>
      <div style={{ display: "grid", gridTemplateColumns: "repeat(5, 60px)", gap: "10px" }}>
        {seats.map((seat) => (
          <button
            key={seat}
            onClick={() => toggleSeat(seat)}
            style={{
              padding: "10px",
              borderRadius: "5px",
              border: "1px solid gray",
              backgroundColor: selectedSeats.includes(seat) ? "orange" : "white",
              cursor: "pointer",
            }}
          >
            {seat}
          </button>
        ))}
      </div>
    </div>
  );
};

export default SeatSelector;
```

//Summary.js

```
import React from "react";


const Summary = ({ movie, selectedSeats, isConfirmed }) => {
  const totalCost = selectedSeats.length * movie.price;


  return (
    <div style={{ marginTop: "20px" }}>
      <h2> Booking Summary</h2>
      <p>Selected Seats: {selectedSeats.length > 0 ? selectedSeats.join(", ") : "None"}</p>
      <p>Total Tickets: {selectedSeats.length}</p>
      <p>Total Cost: ₹{totalCost}</p>
      {isConfirmed && <h3 style={{ color: "green" }}> Booking Confirmed!</h3>}
    </div>
  );
};


export default Summary;
```

Case Study 4:

Fitness Tracker Dashboard

Scenario: A fitness app tracks user workouts (steps, calories burned, water intake).

//App.js

```
import React, { useState } from "react";
import StepsTracker from "./StepsTracker";
import CaloriesTracker from "./CaloriesTracker";
import WaterTracker from "./WaterTracker";
import Summary from "./Summary";


const App = () => {
```

```
  const user = {
    name: "Ammu",
    age: 22,
    weight: 55,
  };

  const [steps, setSteps] = useState(0);
  const [calories, setCalories] = useState(0);
  const [water, setWater] = useState(0);

  return (
    <div style={{ padding: "20px", fontFamily: "Arial" }}>
      <h1> Fitness Tracker Dashboard</h1>
      <p>
        {user.name}, Age: {user.age}, Weight: {user.weight}kg
      </p>

      <StepsTracker steps={steps} updateSteps={setSteps} />
      <CaloriesTracker calories={calories} updateCalories={setCalories} />
      <WaterTracker water={water} updateWater={setWater} />

      <Summary steps={steps} calories={calories} water={water} />
    </div>
  );
};

export default App;
```

//StepsTracker.js

```
import React from "react";
```

```
const StepsTracker = ({ steps, updateSteps }) => {
  return (
    <div style={{ marginTop: "20px" }}>
      <h2> Steps Tracker</h2>
      <p>Steps Walked: {steps}</p>
      <button onClick={() => updateSteps(steps + 500)}>+500 Steps</button>
      <button onClick={() => updateSteps(steps + 1000)} style={{ marginLeft: "10px" }}>
        +1000 Steps
      </button>
    </div>
  );
};

export default StepsTracker;
```

//CaloriesTracker.js

```
import React from "react";

const CaloriesTracker = ({ calories, updateCalories }) => {
  return (
    <div style={{ marginTop: "20px" }}>
      <h2> Calories Tracker</h2>
      <p>Calories Burned: {calories}</p>
      <button onClick={() => updateCalories(calories + 100)}>+100</button>
      <button onClick={() => updateCalories(calories + 250)} style={{ marginLeft: "10px" }}>
        +250
      </button>
    </div>
  );
};
export default CaloriesTracker;
```

//WaterTracker.js

```
import React from "react";


const WaterTracker = ({ water, updateWater }) => {
  return (
    <div style={{ marginTop: "20px" }}>
      <h2> Water Tracker</h2>
      <p>Glasses Drank: {water}</p>
      <button onClick={() => updateWater(water + 1)}>+1 Glass</button>
      <button onClick={() => updateWater(water + 2)} style={{ marginLeft: "10px" }}>
        +2 Glasses
      </button>
    </div>
  );
};

export default WaterTracker;
```

//Summary.js

```
import React from "react";


const Summary = ({ steps, calories, water }) => {
  return (
    <div style={{ marginTop: "30px", padding: "15px", border: "1px solid gray", borderRadius: "10px"
}}>
      <h2> Daily Summary</h2>
      <p>Total Steps: {steps}</p>
      <p>Total Calories Burned: {calories}</p>
      <p>Total Water Intake: {water} glasses</p>
    </div>
  );
```

```
};

export default Summary;
```

Case Study 5:

Hotel Room Reservation

Scenario: A hotel booking website lets customers select rooms, choose dates, and view total price before confirming.

//App.js

```
import React, { useState } from "react";
import RoomList from "./RoomList";
import Summary from "./Summary";

const App = () => {
  const rooms = [
    { id: 1, type: "Deluxe Room", price: 2000, available: true },
    { id: 2, type: "Suite Room", price: 3500, available: true },
    { id: 3, type: "Standard Room", price: 1500, available: false },
  ];

  const [checkIn, setCheckIn] = useState("");
  const [checkOut, setCheckOut] = useState("");
  const [selectedRoom, setSelectedRoom] = useState(null);
  const [totalPrice, setTotalPrice] = useState(0);

  const handleSelectRoom = (room) => {
    setSelectedRoom(room);
    if (checkIn && checkOut) {
      calculateTotal(room, checkIn, checkOut);
    }
```

```
};

const calculateTotal = (room, inDate, outDate) => {
  const days = (new Date(outDate) - new Date(inDate)) / (1000 * 3600 * 24);
  if (days > 0) {
    setTotalPrice(days * room.price);
  } else {
    setTotalPrice(0);
  }
};

const handleDateChange = (inDate, outDate) => {
  setCheckIn(inDate);
  setCheckOut(outDate);
  if (selectedRoom) {
    calculateTotal(selectedRoom, inDate, outDate);
  }
};

return (
  <div style={{ padding: "20px", fontFamily: "Arial" }}>
    <h1> Hotel Room Reservation</h1>

    <div style={{ marginBottom: "20px" }}>
      <label>
        Check-in Date:{" "}
        <input
          type="date"
          value={checkIn}
          onChange={(e) => handleDateChange(e.target.value, checkOut)}
        />
```

```
      </label>
      <br />
      <label>
        Check-out Date:{" "}
        <input
          type="date"
          value={checkOut}
          onChange={(e) => handleDateChange(checkIn, e.target.value)}
        />
      </label>
    </div>

    <RoomList rooms={rooms} onSelectRoom={handleSelectRoom} selectedRoom={selectedRoom} />

    <Summary
      checkIn={checkIn}
      checkOut={checkOut}
      selectedRoom={selectedRoom}
      totalPrice={totalPrice}
    />
  </div>
 );
};

export default App;
```

//RoomList.js

```
import React from "react";
import RoomCard from "./RoomCard";

const RoomList = ({ rooms, onSelectRoom, selectedRoom }) => {
```

```
  return (
    <div>
      <h2>Available Rooms</h2>
      <div style={{ display: "flex", gap: "20px" }}>
        {rooms.map((room) => (
          <RoomCard
            key={room.id}
            room={room}
            isSelected={selectedRoom?.id === room.id}
            onSelect={() => onSelectRoom(room)}
          />
        ))}
      </div>
    </div>
  );
};

export default RoomList;
```

//RoomCard.js

```
import React from "react";

const RoomCard = ({ room, onSelect, isSelected }) => {
  return (
    <div
      onClick={room.available ? onSelect : null}
      style={{
        border: isSelected ? "2px solid green" : "1px solid gray",
        padding: "15px",
        borderRadius: "8px",
        backgroundColor: room.available ? "white" : "#f5f5f5",
```

```
      cursor: room.available ? "pointer" : "not-allowed",
    }}
  >
    <h3>{room.type}</h3>
    <p>Price per Night: ₹{room.price}</p>
    <p>Status: {room.available ? "Available " : "Not Available "}</p>
  </div>
 );
};

export default RoomCard;
```

//Summary.js

```
import React from "react";

const Summary = ({ checkIn, checkOut, selectedRoom, totalPrice }) => {
 return (
  <div style={{ marginTop: "30px", padding: "15px", border: "1px solid gray", borderRadius: "10px"
}}>
    <h2> Booking Summary</h2>
    {selectedRoom ? (
     <>
      <p>Room: {selectedRoom.type}</p>
      <p>Check-in: {checkIn || "Not selected"}</p>
      <p>Check-out: {checkOut || "Not selected"}</p>
      <p>Total Price: ₹{totalPrice}</p>
      {totalPrice > 0 ? <p>Status: Ready to Confirm </p> : <p>Status: Select valid dates </p>}
     </>
    ) : (
     <p>No room selected yet.</p>
    )}
```

```
    </div>
  );
};

export default Summary;
```