Practical Task – Event Ticket Booking System (React + Node.js)

Duration: 2–3 Hours
Tech Stack: React (frontend), Node.js + Express (backend), optional MongoDB or in-memory data store

You are required to develop a **mini ticket booking system** that allows users to view events, check seat availability by section and row, and book tickets. This system should include backend APIs and a functional frontend that communicates with those APIs.

The challenge will test your skills in **data structure design, concurrency handling, and logical UI updates**.

Features to Implement

** Backend

Implement the following REST APIs:

1. POST /events

Create a new event with:

```
{
          "name": "Concert XYZ",
          "date": "2025-07-10T19:00:00Z",
          "sections": [
           {
             "name": "Section A",
             "rows": [
               { "name": "Row 1", "totalSeats": 10 },
               { "name": "Row 2", "totalSeats": 8 }
             ]
           },
           {
             "name": "Section B",
             "rows": [
               { "name": "Row 1", "totalSeats": 6 }
             ]
           }
          ]
         }
```

2. GET /events

Returns all events with basic details.

3. GET /events/:id/availability

Returns availability of each section and row:

4. POST /events/:id/purchase

Purchase tickets. Input:

Rules:

- Must validate availability and prevent overbooking
- Apply group discount if quantity ≥ 4 (respond with groupDiscount: true)
- Ensure atomic seat updates to avoid race conditions

** Frontend

1. Home Page

- **Fetch and display all events**
- **Button/link to view individual event**

2. Event Detail Page

- **Show availability section-wise and row-wise**
- **Form to select section, row, number of tickets**
- **Button to submit ticket purchase**
- **Display success/failure message**
- **Show "Group Discount Applied" if 4 or more tickets booked**

Rules & Constraints

- Backend can use in-memory store (like a JS object or array) or MongoDB
- Don't use large UI libraries (e.g., no Material UI); simple CSS or Bootstrap is fine
- Do not expose internal booking logic on the frontend
- Do not allow rebooking already booked seats

Practical Task – Event Ticket Booking System (React + Node.js)

Bonus Points

- Added unit tests (backend logic)
- Real-time UI update without page reload
- Dockerized app setup