Final Project Requirements – DRDO Venue & IT Resource Booking System

This document includes the exact instructions and flow structure as explained for building a full-stack web application for DRDO internal use. This system is used to book venues and IT resources (e.g., projectors, VC setup) for meetings and internal activities. The UI and logic must strictly follow the attached PDF files given by the user.

■ Main Purpose

This application is made for internal use at DRDO to manage and book venues and IT materials for meetings (online/offline) and other official activities.

■ Meeting Types

- Offline Meetings – Require physical venues (Lecture Hall, Auditorium, Others).
- Online Meetings – May need VC support, projector, mic, etc.
- Miscellaneous Activities – For internal sessions, training, events at different DRDO locations.

■ Booking Flow (Must Follow This)

1. User requests a booking with date, time, venue, and required IT support.
2. System checks whether time and venue are available. If either is already booked, show a message and stop the process.
3. If available, booking request goes to Group Director (GD) for approval.
4. If GD approves, request is sent to Director's Secretary.
5. If Secretary approves, request goes to IT Team for final setup confirmation.
6. Once IT marks setup complete, the event is ready.
7. After event, user will submit feedback and attach any files (PPT, videos, etc.).

■ Real-time Booking Status Tracking

There must be a tracking page for users to check the live status of their booking request. This page should show step-by-step status: ✔ Submitted → ✔ GD Approved → ✔ Secretary Approved → ✔ Setup Done If rejected at any stage, show red cross with reason. It should work using real-time data using WebSocket or polling API.

■ UI & Design Structure (Must be Same as PDF)

- Use exact layout, components, and Tailwind CSS classes as shown in the PDF.
- Sidebar on the left must be dark blue (#1e293b), with active tab blue (#1e90ff).
- All cards must be white with shadow and rounded corners.
- Status tags: Green = Approved, Yellow = Pending, Red = Rejected.
- Form, modals, alerts, all elements must match the attached design brief.

■ Pages (Frontend Routes)

- /login – Role-based login
- /register – Register as user
- /user – User dashboard
- /user/book – Book venue
- /user/bookings – Booking history
- /user/track – Live approval tracking
- /user/feedback – Submit feedback after event
- /director – Group Director's approval panel
- /secretary – Director Secretary final approval panel
- /it – IT team's setup confirmation dashboard

■■ Backend (Node.js + Express)

Use Express.js with role-based authentication (JWT). Implement REST APIs for login, bookings, feedback, and real-time status using WebSocket (Socket.IO).

■ Frontend (React + Tailwind)

Use React with Tailwind CSS for all pages. Use Tailwind UI and grid system exactly as shown in design PDF. Make sure all screens are mobile-responsive and pixel-perfect to the design.

■ Final Deliverables

- Frontend (React) – All pages styled as per design PDF
- Backend (Node.js) – Secure APIs with role-based control
- Tracking System – Real-time status viewer with WebSocket
- Booking Logic – Reject if time/venue not free
- Feedback System – Text + optional file uploads
- PDF Design Matching – All pages, colors, elements must match uploaded PDF