Work Breakdown Structure (WBS) - Advanced Movie Booking Application

1. Project Setup

1.1 Initialize Git repository and .gitignore

1.2 Create monorepo or separate folders for frontend & backend

1.3 Set up package managers (npm/yarn)

1.4 Install core dependencies:

- Frontend: React, React Router, Axios, Redux/Context, TailwindCSS
- Backend: Express.js, Mongoose, dotenv, cors, JWT
- Dev tools: ESLint, Prettier, nodemon, concurrently

2. Authentication Module (JWT + RBAC)

■ You can start working on Login/Register here

2.1 User Authentication

2.1.1 Register API (name, email, password → hashed)

2.1.2 Login API (email + password validation → JWT generation)

2.1.3 JWT Middleware (auth + role guard)

2.1.4 Password hashing (bcrypt)

2.1.5 Role-based access control middleware (admin, user, theater)

2.2 Frontend Forms

2.2.1 Login Page (role selector optional)

2.2.2 Register Page (dynamic: user/theater registration)

2.2.3 Input validation (client + server)

2.2.4 Error handling and toast notifications

2.2.5 JWT storage (localStorage or secure cookies)

2.2.6 Protected routes (React Router + role check)

3. User Module

Browse Movies, Filters

View Movie Details + Showtimes

Real-time seat selection UI

Book Ticket and Booking History

Invoice Download and Notifications

4. Theater Module

Theater Registration/Login

Manage Screens and Seat Layout

Add Movies with Showtimes

View Bookings and Revenue

5. Admin Module

Admin Login (Super/Admin roles)

Manage Users, Theaters, Movies

Assign Admin Roles

Admin Dashboard with Charts

6. Booking System

Booking Flow Implementation

Seat Lock Mechanism (Redis or in-memory)

Booking Status Handling

7. Payments Module

Stripe/PayPal integration

Handle success/failure

Map payments to bookings

Store transaction details

8. Notification System

Email (SendGrid/Nodemailer)

SMS (Twilio)

Booking confirmation alerts

9. Admin Analytics Dashboard

Monthly Bookings, Revenue by Theater/Movie

Most Booked Shows

User growth stats

Export reports (CSV/Excel)

10. Multi-language Support (i18n)

Setup i18n (react-i18next)

Translate static labels

Language switcher UI

11. UI/UX Design

Responsive design, Theme support

Accessible forms and dashboards

Theater seat layout UI

12. Deployment & DevOps

Docker, PM2, Nginx setup

MongoDB Atlas setup

CI/CD pipelines

Logging & Error tracking

■ Suggested Development Order

1. Setup Backend (Express, MongoDB, Auth APIs)
2. Setup Frontend (React pages for login/register)
3. Implement RBAC and route protection
4. Build admin/theater/user dashboards
5. Add movie/showtime/booking flow
6. Add payment and real-time seat logic
7. Polish with notifications, i18n, analytics