

Movie Booking Application - Project Documentation

Work Breakdown Structure (WBS)

1. Planning & Setup

- Project structure
- GitHub & .env
- Install dependencies

2. Frontend (React.js)

- Login/Signup UI
- Role Dashboards
- Movie list & filters
- Seat selection UI

3. Backend (Node.js)

- Express server setup
- Auth & Roles
- Admin, User, Theater APIs

4. Database (MongoDB)

- User schema
- Theater schema
- Movies/Shows schema
- Booking schema

5. Core Modules

- Admin: User/Theater mgmt
- Theater: Movies/Shows mgmt
- User: Browse/Book/History

6. Testing

- Unit/API/UI testing
- Bug fixing

7. Deployment

- Vercel/Render
- Docs & API guide

Module-wise Field Requirements

User Module

- Username (text)
- Email (email)
- Password (password)
- Confirm Password (password)
- Phone (tel)
- Role (hidden: user)
- Profile Picture (file)

Theater Module

- Theater Name (text)
- Email (email)
- Password (password)
- Address (textarea)
- Screens (number)
- Owner Name (text)
- Role (hidden: theater)

Movie Management (Theater)

- Movie Title (text)
- Language (select)
- Genre (select)
- Duration (number, min)
- Show Timings (datetime)
- Poster (file)

Booking Module

- User ID (hidden)
- Movie ID (hidden)
- Showtime (datetime)
- Selected Seats (array)
- Payment Method (radio/select)
- Total Price (auto-calculated)

Admin Module

- Login ID (email)
- Password (password)
- Theater Approval (toggle)
- User Ban (toggle)

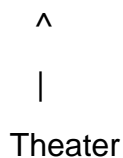
Data Flow Diagram (DFD) - Overview

Actors:

- User: Registers, books tickets, views bookings
- Theater: Adds shows, manages movies & screens
- Admin: Approves theaters, manages users & global movies

DFD Level 0 (Context):

User <--> [Movie Booking System] <--> Admin



DFD Level 1 (Expanded):

[Login/Register] --> Auth Service --> [User DB]

[Browse Movies] --> Movie Service --> [Movie DB]

[Select Showtime] --> Show Handler --> [Show DB]

[Seat Selection] --> Booking Engine --> [Booking DB] --> Payment API --> [Payment Logs]