📌 Project Title:

UniTicket SDK – One SDK to Book Them All

🌟 Vision Statement:

To unify the fragmented global ticketing systems across industries (airlines, railways, events, cinemas, concerts) with a **developer-first SDK** that provides seamless backend infrastructure to build, manage, and scale any ticket booking system.

🧠 Problem Statement:

Ticketing systems today are:

- **Industry-specific** (e.g. IRCTC, BookMyShow, Eventbrite).
- **Redundant in development efforts** (every new startup builds from scratch).
- **Closed APIs or monolithic solutions**.
- Difficult to integrate for developers building **custom platforms**.

💡 Proposed Solution:

Create an **industry-agnostic**, plug-and-play **Go-based SDK** that:

- Handles ticket creation, booking, payment, seat availability.
- Supports events, transport, cinemas, and more.
- Uses Appwrite for auth, MongoDB for storage, Razorpay for payments.
- Deployable on any cloud with REST APIs.
- Open to integration into any frontend.

🎯 Target Users:

Segment	**Use Case**
🧑‍💻 Developers	Integrate full booking backend into apps
🏢 Startups	Build ticket-based platforms rapidly
🚆 Transport Companies	Airlines, Railways, Bus Booking Engines
🎟️ Event Hosts	Conferences, concerts, exhibitions
🍿 Cinemas	Movie chains & OTT screenings

⚙️ Core Features:

Module	**Description**
🎫 Ticket Booking	Select shows/events/flights and book with seat management
🪑 Seat Management	Real-time seat lock and update
💳 Payments	Integrated with Razorpay
🔐 Auth & Roles	Appwrite-based login, user roles
📦 SDK Package	Developer SDK to integrate easily
📊 Booking Logs	MongoDB log of each transaction
🌍 Cross-Domain Support	Can be used for events, movies, railways, etc.
☁️ Docker Support	Easily deployable via containers

🧱 Tech Stack:

Layer	**Technology**
Language	Go (1.21)
Authentication	Appwrite
Payments	Razorpay
Database	MongoDB
API Framework	Gorilla Mux (REST)
Deployment	Docker, Docker Compose
API Testing	Postman / Thunder Client
Documentation	Swagger (OpenAPI Spec)
Version Control	GitHub

🏗️ Architecture Workflow:

User App → SDK/API → Appwrite (Auth)
↓
MongoDB (Booking DB)
↓
Razorpay (Payment)

🧩 Example Use Case:

Frontend: A startup building a movie ticket booking app.

Backend: They plug in your SDK, create show endpoints, manage seat locking, integrate Razorpay, and go live in 2 days — all without writing complex backend logic.

💰 Monetization Model:

Model	**Explanation**
💳 SaaS Subscription	Monthly fees for hosted SDK (Cloud offering)
💼 Enterprise Licensing	Large companies can white-label the backend
💸 Per Booking Commission	Earn ₹1–₹5 per ticket booked via SDK
🛠️ Plugin Marketplace	Analytics, Loyalty Points, QR Code Add-ons
🚀 Developer Tiering	Free → Pro → Enterprise API access limits

📊 Business Strategy:

Stage	**Activities**
MVP (Now)	Bookings, seat manager, payments
Q2 2025	Multi-domain templates (flights, events, cinemas)
Q3 2025	Add webhook support, web dashboard
Q4 2025	AI-based demand analytics + mobile SDK
2026	Enterprise integrations (Amadeus, Sabre, IRCTC APIs)

🆚 Comparison with Existing Global Companies:

Feature	**UniTicket SDK**	**BookMyShow**	**Eventbrite**	**Ticketmaster**	**Amadeus**
Domain-Agnostic	✅	❌	❌	❌	❌
SDK for Devs	✅	❌	❌	❌	Partially
API-first	✅	❌	✅	❌	✅
White-label Support	✅	❌	❌	❌	✅
Easy Deployment	✅ (Docker)	❌	❌	❌	❌
Monetization	B2B SaaS	B2C	Event-based fee	Closed	GDS licensing

🏆 Unique Selling Points (USP):

- 💡 **First Go-based universal ticketing SDK**
- 🔓 **Open, plug-and-play REST APIs**
- 🌍 **Industry-agnostic** (works across domains)
- 💼 **Modular + scalable + secure**
- 📦 **Self-hosted or SaaS option**
- ⚙️ **No frontend required — backend focused**

💡 Long-Term Potential:

⍰ Integration with **flight APIs**, **rail networks**, **event discovery engines**

⍰ Partnering with ticket printers, kiosks, POS devices

⍰ Enabling **decentralized ticketing** with Web3 in the future

⍰ Real-time seat heatmaps, resale ticket marketplaces