TicketMarket Content

TicketMarket – Cross-Platform Mobile Ticketing Platform (iOS & Android)

TicketMarket is a production-grade, cross-platform mobile ticketing application built using Expo (React Native) for both iOS and Android, supported by a scalable backend and administrative system.

The project was designed, implemented, and deployed as a real commercial product — covering mobile clients, backend APIs, authentication, administration, and operational tooling.

Product Status & Strategic Decision

TicketMarket was successfully built and uploaded to the app distribution pipeline (iOS & Android).

After several months of development and internal testing, the decision was made to pause and shut down the product in order to reassess long-term business strategy, market positioning, and monetization models.

This decision was intentional and strategic, not technical.

The platform architecture was designed to be re-activated in the future, following updated business and go-to-market decisions.

Core Vision

TicketMarket was created to serve as a full mobile-first ticket marketplace, enabling:

End users to browse events and securely purchase tickets
Event organizers to manage events and ticket inventory
Administrators to oversee users, events, and system integrity
The system reflects real-world product thinking, not a demo or academic project.

Mobile Application (Expo – iOS & Android)

The mobile app was built using Expo (React Native) to deliver a single codebase across platforms.

Key mobile capabilities:

iOS & Android support from a unified codebase
Secure authentication flow
Role-based access to screens
Event browsing and ticket purchasing
Ticket viewing and digital ticket access
API-driven state management
Environment-based configuration (dev / production)

The app architecture follows production mobile standards, including authentication-aware navigation and protected routes.

User Authentication & Authorization

The platform includes a robust authentication and authorization system:

Secure user registration and login
Token-based authentication (JWT / Firebase Auth)
Role-based permissions:
Ticket buyers
Event organizers
Administrators
Ownership validation on every sensitive operation
Backend-enforced authorization (not client-trusted)

Security and access control were treated as core system requirements.

User Features (Ticket Buyers)

End users can:

Create and manage accounts
Browse events
Purchase tickets
View owned tickets
Access digital / PDF ticket files
View purchase history
Manage personal data securely

All actions are validated on both client and server sides.

Event & Ticket Management

The system includes a complete event lifecycle:
Event creation and editing
Ticket type configuration (price, limits, availability)
Ticket inventory tracking
User-to-ticket ownership binding
Ticket status management
Secure file uploads per ticket
Tickets are treated as secure, auditable assets, not static files.

Admin Dashboard

A dedicated admin dashboard enables full operational control:
User management and auditing
Event moderation
Ticket monitoring
Soft-deletion and recovery
System-level oversight
The admin system mirrors real operational tools used in commercial platforms.

Backend Architecture

The backend was built with a clean, scalable architecture:
RESTful API
Controllers / Services / Data layers
Strong entity modeling
Input validation & sanitization
Authorization checks on all endpoints
Centralized error handling
Audit-ready data models
The architecture supports future expansion and reactivation.

Technologies Used

Mobile
Expo (React Native)
TypeScript
iOS & Android deployment
Secure navigation & protected routes
Backend
REST API
JWT / Firebase Authentication
File upload & storage handling
General
Role-based authorization
Mobile-first architecture
Production-ready project structure

TicketMarket demonstrates my ability to:
Build and deploy a real mobile product (iOS & Android)
Design scalable backend systems
Implement secure authentication & authorization
Build admin dashboards for real operational use
Make product and business-driven decisions, including when to pause development
This project reflects real-world software engineering and product ownership, from architecture to business strategy.