System Architecture Specification

Restaurant Event Ticket Booking & Entry Verification System

PRODUCTION RELEASE CANDIDATE

Architecture Team

December 3, 2025

EXECUTIVE SUMMARY

This document outlines the final, production-safe architecture for the Event Ticketing System. This is a **fraud-aware** system designed to eliminate fake payment screenshots, duplicate entries, and manual gate errors. It utilizes a three-layer control mechanism: Public Website, AI Automation (LLaMA 4), and Manual Human Verification.

1. System Layers & Core Workflow

The system operates on three distinct, controlled layers:

1. **Customer Website (Public):** User registration, ticket selection, and proof of payment upload.
2. **Backend Core (Secure):** Supabase Database + LLaMA 4 Scout (OCR) for data extraction.
3. **Admin & Bouncer System (Control):** Manual financial verification and physical access control.

The Core Identity: Ticket ID + QR Code + Manual Verification + Gate Confirmation.

2. Customer Website (Public Interface)

Step 1: Registration
Mandatory fields: Full Name, Phone Number, Email ID. (Optional OTP via Supabase Auth).
Step 2: Ticket Selection & Payment
User selects ticket tier (e.g., Silver: 2000 INR, Gold: 3500 INR).

- **Payment Method:** Static UPI QR (Owner's Account). No Gateway APIs.
- **Action:** User scans manually and transfers funds.

Step 3: Proof of Payment
User uploads the transaction screenshot. The system immediately:

- Creates unique `Ticket ID`.
- Generates a QR Code.
- Sets Status to **PENDING**.

Step 4: The Ticket Cart
The user receives the ticket, but it is marked **"Under Verification"**. It cannot be used for entry yet.

3. AI Automation (Backend Intelligence)

Role: Assistant, not decision maker.
Upon screenshot upload, the backend triggers **LLaMA 4 Scout Vision**:

- **Input:** Payment Screenshot.
- **Extraction:** UTR Number, Amount, Date.
- **Storage:** Data saved to Admin-only fields (`utr_number`, `extracted_amount`).

SECURITY PROTOCOL

The customer **NEVER** sees the extracted UTR or Amount. These fields are visible **ONLY** to the Admin to prevent users from editing or faking data based on AI errors.

4. Admin Dashboard (Verification Layer)

Location: /admin/dashboard
The Admin (Owner) reviews all `PENDING` tickets. The dashboard presents:

1. Ticket Details (Name, Price).
2. The Actual Screenshot.
3. **AI Hints:** Extracted UTR and Amount (for quick reference).

The Decision: The owner checks their bank app.

- **VERIFY**: Money received. Status → `VERIFIED`.
- **REJECT**: Fraud/Mismatch. Status → `REJECTED`.

5. Ticket Lifecycle (The Locked System)

This lifecycle is rigid. No other transitions are possible.

STATUS	MEANING	ENTRY ALLOWED?
PENDING	Screenshot uploaded, not checked	**NO** ×
VERIFIED	Owner confirmed payment	**YES** ✓
USED	Entry completed at Gate	**NO** ×
REJECTED	Fake or invalid payment	**NO** ×

Logic Flow:
`PENDING` → (Admin Verifies) → `VERIFIED` → (Bouncer Scans) → `USED`

6. Bouncer & Gate Scanner System

Location: /admin/scan
The final line of defense. The Bouncer scans the QR code using the integrated camera.
Possible Outcomes:

- **INVALID ID:** Ticket not found. Access Denied.
- **PENDING:** Payment not verified. Access Denied. (User asked to wait).
- **REJECTED:** Fraud flagged. Access Denied.
- **VERIFIED:** VALID STATE.

The ”Confirm Entry” Protocol:
When a `VERIFIED` ticket is scanned, the Bouncer sees a **”CONFIRM ENTRY”** button.

- Tapping this updates status to `USED` and logs the timestamp.
- Second scan results in: **ALREADY USED**.

7. Threat Model & Scope

WHAT THIS PREVENTS

- Fake payment screenshots.
- Reusing screenshots for multiple tickets.
- Sharing QR codes via WhatsApp (One-time use).
- Duplicate entries.
- Manual gate confusion.

WHAT IT DOES NOT DO

- Does NOT connect to Bank APIs directly.
- Does NOT trust AI for financial decisions.
- Does NOT auto-approve payments.

8. Technical Stack

Layer	Technology
Frontend	React / Next.js, Tailwind CSS
Scanning	html5-qrcode / zxing
Backend Logic	Node.js / FastAPI
Database	Supabase (PostgreSQL)
AI/OCR	LLaMA 4 Scout Vision API
Storage	Supabase Storage Buckets