

Here's a **step-by-step roadmap** for building your ticketing system from scratch, optimized for clarity and efficiency. I'll present it as a **visual diagram** followed by detailed explanations:

## Key Stages Explained

### 1. Core Infrastructure (Weeks 1-3)

- **Auth System (7 days)**
  - JWT with refresh tokens
  - Role-based access control (RBAC)
  - Biometric login setup
- **Database Design (5 days)**
  - Optimized schemas (Users, Tickets, Events)
  - Indexes for critical queries (QR scans, user tickets)
- **API Framework (5 days)**
  - Express.js with rate limiting
  - Winston logging + request tracing

### 2. Ticket Flow (Weeks 3-5)

- **Purchase Logic (5 days)**
  - Tiered pricing (VIP/Early Bird/General)
  - Idempotency keys for duplicate purchases
- **QR Generation (3 days)**
  - Offline-validable codes with expiration
  - Base64 image generation
- **Validation API (5 days)**
  - Staff-facing endpoint
  - Anti-replay protection

### 3. Payments (Weeks 5-7)

- **MoMo/Visa Integration (7 days)**
  - Sandbox testing

- Webhook security (signature verification)
- **Refunds (5 days)**
  - Escrow management
  - Idempotency key tracking

#### ***4. Admin Tools (Weeks 7-10)***

- **Organizer Dashboard (7 days)**
  - Real-time ticket scans
  - Manual override controls
- **Super Admin (5 days)**
  - Force refunds
  - User impersonation
- **Analytics (7 days)**
  - Revenue reports
  - Attendee demographics

#### ***5. Testing (Weeks 10-12)***

- **Unit Tests (5 days)**
  - 100% auth/payment coverage
  - Mock offline scenarios
- **Load Tests (5 days)**
  - 500+ concurrent users
  - 2G throttling checks
- **Security Audit (3 days)**
  - OWASP Top 10 scan
  - JWT secret rotation

#### ***6. Deployment (Week 12)***

- **Staging (3 days)**
  - UAT with real devices
  - Payment dry-runs
- **Production (2 days)**
  - Blue/green deployment
  - 24-hour monitoring

## Critical Path

flowchart LR

```
A[Auth] --> B[Ticket Purchase]
B --> C[Payment Webhooks]
C --> D[Validation]
D --> E[Refunds]
E --> F[Analytics]
```

## Pro Tips for Execution

1. **Parallelize Work**
  - a. Build frontend (React Native) alongside APIs after Week 3
  - b. Run security scans during integration testing
2. **Risk Mitigation**
  - a. Payment provider fallbacks (e.g., switch to Cash if MoMo fails)
  - b. Daily database backups from Day 1
3. **Toolchain**
  - a. **GitHub Actions:** Automated testing on every commit
  - b. **Sentry:** Real-time error monitoring
  - c. **Postman:** Shared API collection for team testing

## Suggested Adjustments for Your Context

1. **Prioritize Offline-First**
  - a. Extend QR validation logic to work without internet
  - b. Add AsyncStorage caching early (Week 2)
2. **Tourist Flow**
  - a. Email-only signup option (bypass phone requirement)
3. **Localization**
  - a. Add French/Portuguese labels in API responses

Would you like me to adapt this for a specific tool (e.g., Jira, ClickUp) or focus on refining a particular phase?

