

# 1. Business Concept

- **Problem you're solving:**
  - Customers wait too long for service.
  - Cafés/restaurants spend on staff taking orders manually.
  - Errors occur in handwritten orders.
- **Solution:**
  - Each table has a QR code.
  - Customer scans → sees digital menu → places order → order goes directly to kitchen + billing system.
  - Restaurant staff only need to serve.
- **Target customers:**
  - Independent cafés and restaurants (easy to onboard).
  - Later expansion to franchises, cloud kitchens, bars, lounges.
- **Revenue Model (SaaS):**
  - Monthly subscription per restaurant (based on no. of tables/outlets).
  - Free tier (limited menus/tables) to attract small players.
  - Add-ons: Payment gateway integration, analytics, loyalty points, marketing tools.

## 2. Features Roadmap

### MVP (Phase 1)

- QR code generation for each table.
- Digital menu (mobile-friendly web app).
- Online ordering (with table no. auto-captured).
- Dashboard for restaurant (new orders, status, order history).
- Basic admin (add/edit menu, mark item unavailable).

### Phase 2

- Online payments (UPI, cards, wallets).
- Multi-language menus.
- Order status updates (e.g., Preparing → Ready → Served).
- Split bills.
- Role-based access (waiters vs managers vs admin).

### Phase 3

- Analytics (popular items, peak hours, repeat customers).
- Loyalty program & discounts.
- POS integration (if restaurant already has billing software).
- AI-powered upselling (e.g., “People also ordered fries with this burger”).
- White-label option (restaurants can brand the menu).

### 3. Tech Stack

- **Frontend:**
  - Angular for restaurant dashboard.
  - Mobile-first menu UI (Angular).
- **Backend:**
  - Java / Spring Boot
  - REST API.
- **Database:**
  - MySQL for structured data.
  - Redis for caching orders in real-time.
- **Infrastructure:**
  - Cloud (AWS / GCP / Azure).
  - Containerized with Docker + Kubernetes (scalability).
- **Other Essentials:**
  - Authentication (OAuth, JWT).
  - Payment Gateway (Stripe, Razorpay, PayPal).
  - Notification system (email/SMS/WhatsApp for order confirmation).

## 4. Business Model

- **Pricing ideas:**
  - Small cafés: \$20–30/month.
  - Mid-size restaurants: \$50–100/month.
  - Enterprise chains: Custom pricing.
- **Free trial:** 14–30 days.
- **Upsell:** Payment processing fees, custom branding, advanced analytics.

## 5. GTM (Go-To-Market) Strategy

- **Step 1: Early adopters**
  - Partner with 2–3 local cafés. Build MVP and run pilot. Collect feedback.
- **Step 2: Scale in local region**
  - Market in your city → build word-of-mouth credibility.
  - Use WhatsApp/LinkedIn outreach to restaurant owners.
- **Step 3: SaaS growth**
  - Website with self-signup flow (restaurants register, upload menu, get QR codes).
  - Build reseller/partner network (POS resellers, food consultants).
- **Step 4: Expansion**
  - Target international markets where QR dining is popular (Middle East, SE Asia, EU).

## 6. Financials (High-level)

- **Initial Costs:**
  - Development (in-house or outsourced).
  - Cloud hosting & infra (~\$100–300/month initially).
  - Payment gateway setup.
  - Marketing (digital ads, outreach).
- **Revenue Projection (example):**
  - Year 1: 50 restaurants × \$30/month = \$18,000 ARR.
  - Year 2: 300 restaurants = ~\$100,000 ARR.
  - Year 3: 1,000+ restaurants = \$300k+ ARR.

## 7. Execution Roadmap

1. **Market Research:** Talk to 10–15 local restaurant owners. Validate pain points.
2. **MVP Build** (2–3 months): Focus only on QR → Menu → Order → Dashboard.
3. **Pilot Launch:** Run in 2–3 cafés, refine.
4. **Go-to-Market:** Create a landing page + sales pitch deck.
5. **Iterate & Scale:** Add payments, analytics, integrations.