

PROJECT ABSTRACT

QR Code Digital Menu and Table Ordering System

An Angular PWA for restaurants: QR-code table menus, order-from-table with live cart, a kitchen display system for live orders, and table billing with UPI — built-to-order with source code, report, PPT and viva kit.

01. Title

QR Code Digital Menu and Table Ordering System

02. Introduction / Background

Restaurant tables still depend on printed menus — expensive to reprint, impossible to update for sold-out dishes — and on waiters ferrying orders to the kitchen, where tickets get lost and priorities blur at rush hour. Diners wait to be noticed; kitchens work from shouted instructions. This project digitizes the table: each table carries a QR code that opens a no-install menu (a PWA) bound to that table; diners browse with photos and prices, build a cart and send the order straight to the kitchen display. The kitchen sees live order tickets with elapsed-time highlighting, fires and completes them, and the table bill — payable by UPI — is always current. Menu updates, price changes and sold-out flags propagate instantly to every table.

03. Problem Statement

Restaurants bleed money on menu reprints, lose orders to inattentive service, and run kitchens on paper tickets that cannot show urgency or be re-prioritized. Diners face stale menus and slow billing. The system replaces paper menus and order slips with a QR-bound digital flow from table to kitchen display to bill, with the kitchen's live state visible to staff.

04. Objectives

- Generate per-table QR codes binding the menu session to a table number.
- Build a mobile-first digital menu: categories, photos, veg/non-veg marks, prices, sold-out flags.
- Implement table cart and order placement routed directly to the kitchen display.
- Build a kitchen display system: live tickets, elapsed-time urgency highlighting, start/done states.
- Implement table billing with GST and UPI/card/cash settlement.
- Provide an admin panel: menu management, table/QR management and sales reports.

05. Proposed Methodology

Built as an Angular PWA with a Node.js/Express API and MongoDB. (1) Table binding: QR encodes restaurant + table ID; scanning opens the PWA scoped to that table's session. (2) Ordering: cart and order APIs write to an orders collection; Socket.io pushes new tickets to

the kitchen display in real time. (3) KDS: tickets carry timestamps; the display highlights ageing orders and supports start/complete transitions that notify the table session. (4) Billing: the table's open orders aggregate into a bill with GST; payment mode is recorded on settlement. The PWA works from the browser with no install, and the menu is fully manageable from the admin panel — no reprints, ever.

06. System Architecture / Working

A diner scans the table QR and the menu opens, already knowing it is Table T-07. They browse categories, add dishes to the cart and tap Send to Kitchen — the order lands on the kitchen display as a new ticket with a running timer. The kitchen starts preparation (status visible to the captain's view) and marks it done when served. Sold-out dishes are flagged by the kitchen in one tap and vanish from all menus instantly. At the end, the table bill aggregates every order with GST; the diner pays via UPI QR or settles in cash, and the table session closes for the next guests.

07. Hardware / Software Requirements

- Node.js 18+, Angular 17, MongoDB 6+, Socket.io — all open source
- Printed QR code standees per table (generated by the system)
- Kitchen display: any tablet/TV with a browser; diners use their own phones
- Any modern laptop/PC for the admin panel and demo

08. Expected Outcomes

- Working system: QR table menus, ordering, live kitchen display, table billing.
- Real-time ticket flow demonstrable with two browser windows (table + kitchen).
- Admin panel for menus, tables, QR generation and sales reports.
- Complete documentation: project report, PPT and viva Q&A.;

09. Applications

- Casual-dining restaurants, cafes and food courts cutting waiter load.
- Cloud kitchens adding a dine-in ordering layer.
- College canteens and office cafeterias with token-style flow.
- A base for adding loyalty or pre-ordering later.

10. Future Scope

- Pre-ordering with scheduled pickup for takeaway.
- Customer loyalty and feedback per visit.
- Multi-branch menu management from one admin.
- Integration with food-aggregator order streams.