

Restaurant Table Reservation System with Floor Plan Booking

Project Abstract | K-1221

1. Introduction / Background

Restaurants lose revenue every night from two avoidable problems: empty tables that could have been filled, and double bookings that turn guests away at the door. Most small and mid-sized restaurants still manage reservations on paper registers or phone calls, which gives the host no real-time picture of the floor. This project builds a web-based reservation system whose centrepiece is a live floor plan: tables drawn as they physically sit on the floor, coloured by live status, so booking a table feels like choosing a seat on a cinema app.

2. Problem Statement

Paper and phone-based reservation handling leads to double bookings, missed special requests, long host queues at peak hours, and no data on table utilisation. Walk-in guests wait because the host cannot see which tables are about to free up. There is a need for a single system where guests book their own tables visually, the host manages the floor live, and no-show patterns are visible.

3. Objectives

- Let guests browse a live floor plan and book a specific table for a date and time slot, with name, phone and party size.
- Give the host a real-time table-status view (free, reserved, occupied) and a reservation management panel.
- Enforce slot rules: booking window up to 30 days ahead, hold policy, and double-booking protection at the database level.
- Provide an admin dashboard with utilisation statistics, upcoming reservations and a no-show tracker.

4. Proposed Methodology

The system is developed as a full-stack web application. The floor plan is modelled as a coordinate grid of tables, each with capacity, zone and status. A relational database stores tables, reservations and admin users. The guest frontend renders the floor plan as an interactive SVG grid, the host view re-renders from the same data, and the backend enforces uniqueness of table + date + time slot. Development proceeds through requirement analysis, database schema design, frontend and backend implementation, integration, and functional testing of the booking rules.

5. System Architecture / Working

- Guest layer: floor-plan viewer, reservation form, booking confirmation, cancellation flow.
- Host layer: live floor status, reservation list, check-in / check-out actions, manual booking entry.
- Admin layer: utilisation dashboard, no-show tracking, table management.

- Backend: REST API with booking validation, slot conflict detection and status state machine.
- Database: tables (tables, reservations, users) with unique constraint on table + date + slot.

6. Hardware / Software Requirements

Item	Specification
Processor	Any modern dual-core processor (development)
RAM	4 GB minimum
Storage	500 MB for code, database and media
Operating system	Windows / Linux / macOS
Backend	Node.js with Express framework
Database	MySQL
Frontend	HTML5, CSS3, JavaScript (ES6)
Browser	Modern Chrome / Firefox / Edge / Safari

7. Expected Outcomes

- A working web application where guests can visually select and book tables.
- A host dashboard showing live table status and the day's reservation list.
- An admin view with utilisation statistics and no-show tracking.
- Documented schema, setup guide and test results for the booking rules.

8. Applications

- Fine-dining and casual restaurants taking advance reservations.
- Cafes and lounges managing limited seating during peak hours.
- Banquet halls and event venues with zone-based seating.
- Any walk-in business where floor-space inventory must be booked visually.

9. Limitations

- Single restaurant per deployment; multi-branch chains are future scope.
- Real-time updates in the demo use periodic refresh; WebSocket push is documented as the production upgrade.
- Payments or table deposits are not integrated; they are a documented extension.
- The system assumes the host updates statuses honestly; sensor-based occupancy detection is not included.

10. Future Scope

- Online payment and table-deposit collection at booking time.

- SMS/WhatsApp confirmation and reminder integration.
- Customer loyalty and visit-history analytics.
- Multi-branch support with a central admin console.
- Waitlist management with automatic table-offer notifications.