**Abstract**

This is the restaurant web app Reference Manual. It documents the restaurant web app.

This restaurant web app is developed and tested by google Server and but some of the restaurant web app features may not function depending on the version of the web browser used.

Restaurant web app platform support evolves over time. For the latest platform support information, see http://www. restaurant web app l.com/support/supportedplatforms/.html.

For notes on the restaurant web app see the changes in each release, see the  [Release Notes](https://dev.mysql.com/doc/relnotes/workbench/en/).

For legal information, including licensing information, see the [Preface and Legal Notices](https://dev.mysql.com/doc/workbench/en/wb-preface.html).

from the My restaurant web app. The the restaurant web app release is available on all supported browsers on Microsoft Windows, macOS, and Linux.

For help with using My restaurant web app, please visit the my restaurant web app.com, where you can discuss your issues with other users.

Document generated on: 2025-03-28 (revision: 81314)

Table of Contents

[System Overview 2](#_Toc194056438)

[Key Components 2](#_Toc194056439)

[Step-by-Step Implementation 2](#_Toc194056440)

[Step 1: QR Code Authentication 2](#_Toc194056441)

[Step 2: Menu Browsing & Order Placement 2](#_Toc194056442)

[Step 3: Payment Integration 3](#_Toc194056443)

[Step 4: Real-Time Order Tracking & Notifications 3](#_Toc194056444)

**Documentation**

# System Overview

The web app allows customers to:

1. Log in by scanning a QR code at their table.
2. Browse the menu, add items to a cart, and place orders.
3. Pay securely via integrated payment gateways.
4. Receive real-time notifications when their order is prepared.

## Key Components

* QR Code Authentication: Unique codes per table for instant login.
* Dynamic Menu: Categorized dishes with images, descriptions, and prices.
* Order Management: Cart functionality, order submission, and kitchen tracking.
* Payment Processing: Integration with services like Stripe or PayPal.
* Real-Time Alerts: Notify customers when orders are ready.

# Step-by-Step Implementation

## Step 1: QR Code Authentication

1. Generate QR Codes for Tables
   * Assign a unique, short-lived token to each table.
   * Embed the token into a QR code URL
   * Print QR codes and place them on tables.

Validate Scanned QR Codes

* + When a user scans the code, the frontend sends the token to the backend for validation.
  + If valid, the backend returns the table ID, and the user is logged into the app.

1. Session Management
   * Store the table ID locally (e.g., browser storage) to maintain the session.

## Step 2: Menu Browsing & Order Placement

1. Display the Menu
   * Fetch menu data (categories, items, prices) from the backend.
   * Organize items with filters (e.g., vegetarian, gluten-free).
2. Cart Functionality
   * Allow users to add/remove items, adjust quantities, and view the total.
   * Save cart state temporarily in the browser.
3. Submit Orders
   * Send the order details (table ID, items, special requests) to the backend.
   * Assign a unique order ID and mark it as "pending" in the database.

# Step 3: Payment Integration

1. Connect to a Payment Gateway
   * Use Stripe, PayPal, or similar services to handle transactions.
   * Securely collect card details or enable digital wallets (e.g., Apple Pay).
2. Process Payments
   * After order submission, redirect users to a secure payment page.
   * Confirm payment success/failure and update the order status.

# Step 4: Real-Time Order Tracking & Notifications

1. Kitchen Dashboard
   * Build an interface for staff to view and update order statuses (e.g., "preparing," "ready").
2. Real-Time Updates
   * Use WebSocket (e.g., Socket.io) to push order status changes to the frontend.
   * When the kitchen marks an order as "ready," notify the customer instantly.
3. Notifications
   * Display browser alerts or use push notifications (e.g., Firebase Cloud Messaging).
   * Include order details and preparation time in the notification.

After the notification has been received indicating to the customer the order is ready, depending on the pick up option the customer chose the order will be taken to the customer or they will personally pick it up.

**Tinyiko Muxlhanga**

**202202957**