

College Canteen Pre-Order System

Project Abstract | K-1230

1. Introduction / Background

Every college knows the 1 PM canteen crush: a long queue for a quick snack, orders shouted across the counter, paper tokens that get lost, and a kitchen cooking blind with no idea what is coming. This project moves the whole transaction online — students browse a digital menu, build a cart, pay by UPI and pick a pickup slot, then collect against a token number while the kitchen works through a live order queue sorted by slot.

2. Problem Statement

Peak-hour canteen queues waste student time, paper tokens cause order mix-ups, and kitchens cannot plan preparation because they see orders only when customers arrive. A pre-order system is needed where students order and pay ahead, the kitchen sees a slot-sorted live queue, and token-based collection replaces the physical queue.

3. Objectives

- Provide a digital menu with category filters, veg/non-veg marks and prices.
- Implement cart, UPI-recorded checkout and pickup-slot selection.
- Issue token numbers per order for queue-free collection.
- Give the kitchen a live queue dashboard with one-tap stage advancement (received, preparing, ready).
- Provide live order tracking for students and daily revenue reporting for the manager.

4. Proposed Methodology

The system is developed as a full-stack web application with a student frontend and a kitchen dashboard. Orders are header-plus-items records with slot and token fields; a stage state machine governs kitchen progress; the kitchen queue is sorted by pickup slot. UPI payments are recorded against orders. The methodology covers requirement analysis, schema and queue design, implementation, testing of the order lifecycle and slot logic, and documentation.

5. System Architecture / Working

- Student layer: menu browser, cart, checkout with slot picker, live token tracker, order history.
- Kitchen layer: live queue sorted by slot, one-tap stage advancement, menu availability toggles.
- Manager layer: daily orders, revenue, average prep-time reporting.
- Backend: REST API with order state machine and token generation.
- Database: menu items, slots, orders, order items, payments, users.

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 pre-order web app with menu, cart, slots and token tracking.
- A kitchen queue dashboard with live stage advancement.
- Daily revenue and prep-time reporting for the canteen manager.
- Full documentation: schema, manuals, report, PPT and viva Q&A.;

8. Applications

- College and school canteens managing peak-hour rush.
- Corporate office cafeterias with pre-order lunch.
- Food courts with multi-counter token systems (adapted).
- Hostel mess halls with slot-based meal collection.

9. Limitations

- UPI payments are recorded at checkout; a production UPI collect API integration is a documented extension.
- The kitchen display is a web dashboard; dedicated KDS hardware or printer integration is future scope.
- Item sold-out status is toggled manually by staff.
- Delivery to rooms is not included; the model is counter pickup.

10. Future Scope

- UPI collect API integration for real payment collection.
- WebSocket-based real-time kitchen display updates.
- Multi-outlet campus support with outlet selection.
- Demand forecasting from historical order data.

- Loyalty points and meal-subscription plans for hostellers.