



SKH PRIVATE LIMITED

INTERNSHIP PROJECT REPORT

CANTEEN MANAGEMENT SYSTEM

Submitted By

Goldi
Bachelor of Technology (B.Tech)
Cyber Security
Roll No. 27506

Submitted To

HR Anil Chaudhary
HR Dinesh Singh
SKH Private Limited

Internship Duration

15 June 2026 – 24 July 2026

Academic Session

2024–2028

Department of Computer Applications

Dronacharya College of Engineering
Gurugram University

Abstract

The **Canteen Management System** is a web-based application developed during an internship at **SKH Group** to digitize and streamline the daily meal-tracking process of the corporate canteen.

Prior to this system, meal counts were tracked manually, which regularly led to disputes between the Human Resources (HR) department and the canteen vendor: the vendor would report serving more plates than HR had authorised, often because employees brought guests or family members, or because meals eaten outside the canteen premises were still being counted.

This ambiguity made it difficult to reconcile bills and resulted in recurring friction between the two parties.

To resolve this, the system was designed to separate every meal into two clearly defined categories: *regular* meals, which are auto-generated from the daily default meal plan uploaded by HR for every employee, and *extra* meals, which employees must explicitly request and which HR must explicitly approve before the vendor can serve them. The vendor then marks each meal as served against the matched employee record, so that regular and extra counts can never be confused.

The application was built with an Angular frontend, a Node.js/Express REST API backend, a MySQL database, and JWT-based authentication with role-based access control for three distinct user roles: Employee, HR, and Vendor.

The system further provides both HR and the vendor with analytics dashboards, exportable PDF/Excel reports, and a feedback channel between employees and HR, with the overall goal of reducing food wastage and eliminating the ordered-versus-served discrepancy that previously caused conflict.

Keywords: Canteen Management, Meal Booking System, Web Application, Angular, Node.js, MySQL, Role-Based Access Control.

Acronyms

API	Application Programming Interface
CMS	Canteen Management System
CRUD	Create, Read, Update, Delete
HR	Human Resources
JWT	JSON Web Token
REST	Representational State Transfer
SQL	Structured Query Language
UI	User Interface

Contents

Abstract	1
Acronyms	2
1 Introduction	5
1.1 Context	5
1.2 Problem Statement and Motivation	5
1.3 Objectives	6
1.4 Organization	7
2 Background	8
2.1 The Manual Process	8
2.2 Requirements of Similar Systems	8
2.3 Summary	8
3 Requirements and System Architecture	9
3.1 Functional Requirements	9
3.1.1 Employee Role	9
3.1.2 Employee Login	9
3.1.3 HR Role	9
3.1.4 HR Login	10
3.1.5 Real HR Login Credentials	10
3.1.6 Vendor Role	10
3.1.7 Vendor Login Credentials	11
3.2 Non-Functional Requirements	11
3.3 System Architecture	11
4 System Design	13
4.1 Database Design	13
4.2 Workflow Design	13
5 Implementation	15
5.1 Technology Stack	15
5.2 Authentication Module	15
5.3 Meal Booking and Approval Module	15
5.4 Vendor Verification Module	16
5.5 Analytics and Reporting Module	17
5.6 Notifications and Feedback Module	18
5.7 Deployment	18

6	Testing and Results	20
6.1	Testing Approach	20
6.2	Results	20
7	Conclusion and Future Work	21
7.1	Conclusion	21
7.2	Future Work	21

Chapter 1

Introduction

1.1 Context

SKH Group operates a corporate canteen that serves breakfast, lunch, and snacks to its employees on a daily basis. Historically, the number of meals served was tracked manually, either on paper or through informal communication between the vendor and HR staff. Because this process depended entirely on the vendor's own count of plates served, there was no independent record that HR could use to verify how many meals were actually authorised versus how many were consumed. This project was carried out as an internship under the supervision of the Corporate IT department, with the requirement specification provided directly by HR Anil Chaudhary. The resulting system, the Canteen Management System (CMS), replaces the manual process with a digital workflow shared by three types of users: employees, HR staff, and the canteen vendor.

1.2 Problem Statement and Motivation

Kindly review the below requirement and proceed with the development of the applications.		
Please analyze the feasibility, effort estimation (not mandatory), and dependencies involved. In case of any queries or clarifications required, kindly let me know.		
Functionality	Present Challenges	Benefits to the Organisations
Canteen Management System	Manual tracking of employee meals and subsidies Limited data on consumption patterns and wastage Complex reconciliation with vendors	Better control on food subsidy costs Reduction in wastage through data-driven planning Accurate vendor billing and faster reconciliation Improved hygiene, compliance, and employee satisfaction
This module is to be developed as per existing modules in our intranet application.		
Technology Stack:		
• Angular • TypeScript • Node.js • Express.js • MySQL • JWT Authentication		

Figure 1.1: Problem Statement

The core problem addressed by this project is the lack of a verifiable, single source of truth for canteen meal counts. In the manual process, the vendor would frequently claim to have served more meals than HR had authorised. Investigation of these disputes typically traced back to two recurring causes: employees bringing guests or family members who were then counted as regular employee meals, and employees who were marked present but had eaten outside the canteen while still being included in the vendor's served count. Because there was no digital record distinguishing *authorised* meals from *extra* or *unauthorised* ones, HR had no reliable way to challenge the vendor's figures, and the vendor had no way to prove their figures were accurate. This created a recurring monthly billing conflict and, indirectly, contributed to food wastage, since the kitchen had no accurate advance count of how many meals to prepare.

The motivation for this project was therefore to build a system that removes this ambiguity entirely by requiring every meal to be explicitly accounted for: regular meals are generated

automatically from HR’s approved daily default plan, and any meal outside that plan must be requested by the employee and approved by HR before the vendor is even permitted to serve it. This turns the previously informal, dispute-prone process into an auditable digital record.

1.3 Objectives

The general objective of this project is to design and implement a role-based web application that digitises the entire canteen meal-tracking workflow for SKH Group. The specific objectives are to:

- Allow HR to upload a daily default meal plan that automatically generates one regular meal entry (breakfast, lunch, and/or snacks) per employee.

The screenshot shows the 'Upload Today's Meal' interface. On the left is a sidebar with navigation links: Dashboard, Today's Menu, Upload Meal (active), Upload Meal History, All Notifications, Feedback received, Extra Booking Details, Canteen History, and All Report Export. The main form area has a title 'Upload Today's Meal' and a subtitle 'Upload today's meal for employees and vendors'. It contains three input fields: 'Current Meal' (Breakfast), 'Booking Window' (14:00 PM - 14:30 PM), and 'Meal Window' (14:00 PM - 14:30 PM). Below these are fields for 'Meal Type' (Breakfast), 'Date' (07/21/2024), 'Food Item' (chai), 'Qty' (200), and a 'Veg' dropdown menu. There is an 'Add Food Item' button. A 'Today's Menu Preview' section shows a list of items: chai, Veg, and Qty: 200. At the bottom right are 'Save Draft' and 'Publish Menu' buttons.

Figure 1.2: Upload Meal

The screenshot shows the 'Uploaded Meal History' interface. It has a search bar and a dropdown menu. Below is a table with the following data:

Date	Meal	Food Item	Qty	Uploaded By	Status	Action
21 Jul 2024, 10:00 AM	Breakfast	chai (Qty: 200)	200	HR	Published	View
21 Jul 2024, 09:30 AM	Breakfast	Qty: 0	0	HR	Published	View
20 Jul 2024, 08:13 PM	Snacks	vanilla (Qty: 500)	500	HR	Published	View
20 Jul 2024, 08:50 PM	Snacks	vanilla (Qty: 100), brown pudding (Qty: 100)	1000	HR	Published	View
20 Jul 2024, 08:49 PM	Snacks	vanilla (Qty: 100), brown pudding (Qty: 100)	1000	HR	Published	View

Figure 1.3: Uploaded Meal History

- Allow employees to request additional (extra) meals, with a quantity and a reason, through a self-service booking form.
- Require HR approval before any extra meal request becomes visible to the vendor.
- Allow the vendor to verify and mark meals as served, matched against the employee’s regular and approved-extra entitlements, so that regular and extra counts are never conflated.
- Provide HR and the vendor with analytics dashboards showing ordered, served, extra, regular, and pending meal counts, broken down by meal type and by day, week, year, or custom date range.

- Provide exportable PDF and Excel reports of meal history and analytics, for both auditing and reimbursement purposes.
- Provide a feedback channel that lets employees rate and comment on meals, with responses visible from HR.
- Secure the system with JWT-based authentication and role-based access control across the three user roles: Employee, HR, and Vendor.

1.4 Organization

The remainder of this report is organised as follows. Chapter 2 reviews the background and the shortcomings of the manual process the system replaces. Chapter 3 describes the functional and non-functional requirements gathered from HR and the overall system architecture. Chapter 4 presents the system design, including the database structure and the workflow between the three user roles. Chapter 5 describes the implementation, covering the technology stack and each functional module of the application. Chapter 6 discusses the testing process and the results obtained. Chapter 7 concludes the report and outlines possible future improvements.

Chapter 2

Background

2.1 The Manual Process

Before this project, the canteen operated on a manual sign-off process. HR would give the vendor an approximate daily headcount based on the number of active employees, and the vendor would prepare and serve meals accordingly. At the end of each billing cycle, the vendor would submit a total plate count for reimbursement. Because there was no itemised, per-employee record of who had eaten what, HR had only the vendor’s word to validate against a rough headcount estimate, and any discrepancy between the two figures could not be resolved with evidence from either side.

2.2 Requirements of Similar Systems

Canteen and cafeteria management is a well-established category of enterprise software, typically offering meal pre-booking, RFID or QR-code based verification at the point of service, and integration with payroll for deduction of meal costs. For this internship, the scope was deliberately narrowed to the specific problem SKH Group needed solved: distinguishing regular from extra meals and giving both HR and the vendor a shared, tamper-evident record. Features such as RFID scanning or payroll integration were considered out of scope for the current phase and are discussed as future work in Chapter 7.

2.3 Summary

The background review confirmed that the central gap in the existing manual process was not a lack of general canteen-management concepts, but the absence of a shared digital record that both HR and the vendor could trust. This directly shaped the core design decision described in the next chapter: splitting every meal into a regular, auto-generated entry and an extra, explicitly-approved entry.

Chapter 3

Requirements and System Architecture

3.1 Functional Requirements

3.1.1 Employee Role

- View the daily menu for breakfast, lunch, and snacks.
- Book an extra meal by submitting a quantity and a reason.
- View a personal meal booking history, filterable by meal type.
- Export personal meal history as a PDF, for salary or reimbursement purposes.
- Submit feedback on meals (rating and comments) and view HR's reply.
- Receive notifications relevant to their bookings and HR announcements.

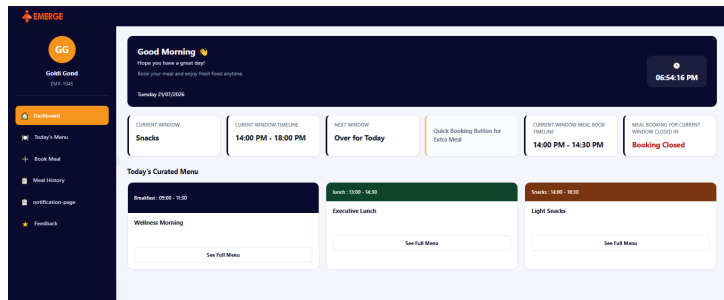


Figure 3.1: Employee Dashboard

3.1.2 Employee Login

Login credentials have been shared separately and are not included in this document for security reasons.

Table 3.1: Employee Login Credentials

3.1.3 HR Role

- Upload the daily default meal plan for all employees.
- Review and approve or reject employees' extra meal requests.
- View analytics on ordered, served, extra, regular, and pending meal counts, filterable by meal type and by day, week, year, or a custom date range.

- Export analytics as PDF and Excel reports.
- View and respond to employee feedback.
- Broadcast notifications (e.g. menu publication) visible to employees and the vendor.

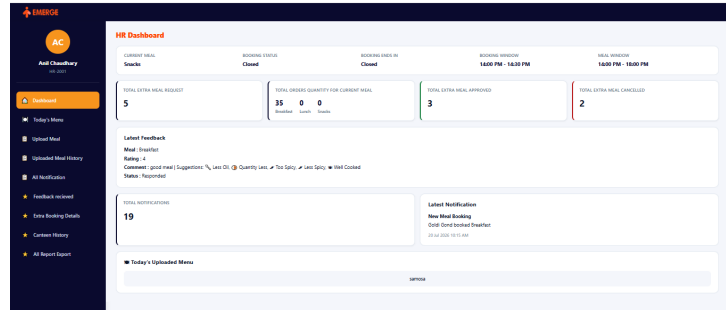


Figure 3.2: HR Dashboard

3.1.4 HR Login

Login credentials have been shared separately and are not included in this document for security reasons.

Table 3.2: HR Login Credentials

3.1.5 Real HR Login Credentials

Login credentials have been shared separately and are not included in this document for security reasons.

Table 3.3: Real HR Login Credentials

3.1.6 Vendor Role

- View the list of regular meals generated from HR's daily plan.
- View the list of HR-approved extra meal requests.
- Mark meals as served, matched by employee identifier, against the combined regular and approved-extra entitlement.
- View summary cards showing regular served, extra approved/served/rejected, and pending counts per meal type.
- View analytics and export reports in the same manner as HR.

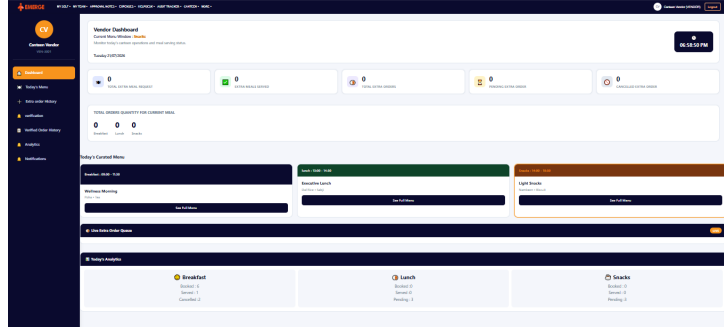


Figure 3.3: Vendor Dashboard

3.1.7 Vendor Login Credentials

Login credentials have been shared separately and are not included in this document for security reasons.

Table 3.4: Vendor Login Credentials

3.2 Non-Functional Requirements

- **Security:** All API endpoints must be protected by JWT-based authentication, with role-based route guards ensuring each user can only access functionality permitted for their role.
- **Data integrity:** Regular and extra meal counts must remain independently traceable at all times, so that served counts can always be reconciled against HR's original authorisation.
- **Usability:** The interface should be simple enough for non-technical canteen and HR staff to use with minimal training.
- **Auditability:** All report exports (PDF/Excel) must reflect the same underlying data as the on-screen analytics, so that printed reports can be used as evidence in HR-vendor reconciliation.

3.3 System Architecture

The system follows a standard three-tier web architecture:

- **Presentation layer:** An Angular single-page application provides three separate portals – Employee, HR, and Vendor – each showing only the navigation and functionality relevant to that role.
- **Application layer:** A Node.js back end built with the Express.js framework exposes a REST API. Authentication is handled via JWT access and refresh tokens, and every route is protected by role-based middleware.

- **Data layer:** A MySQL relational database stores users, refresh tokens, meals, meal requests (bookings), feedback, notifications, and the daily menu.

Communication between the Angular front end and the Express back end takes place over HTTPS using JSON payloads. On the front end, each functional area (meals, bookings, feedback, notifications, menu) is backed by its own Angular service, which performs an optimistic local update to keep the UI responsive and persists the change to the back end asynchronously.

Chapter 4

System Design

4.1 Database Design

The database schema is centred on the following tables:

- **users** – stores employee, HR, and vendor accounts, including role, employee code, and hashed credentials.
- **refresh_tokens** – stores active JWT refresh tokens to support secure session renewal and logout.
- **meals** – the daily menu items uploaded by HR for breakfast, lunch, and snacks.
- **meal_requests** – the central booking table, holding both auto-generated regular entries (quantity defaulted to one per employee) and employee-submitted extra requests, each with a status (Pending, Approved, Rejected, Served, Cancelled).
- **feedback** – employee feedback per meal, including a numeric rating and optional taste/quantity/hygiene/taste fields, plus HR’s reply.
- **notifications** – cross-role notifications, such as menu publication broadcasts from HR.
- **today_menu** – the current day’s published menu, used to drive the employee and vendor menu views.

4.2 Workflow Design

The overall meal lifecycle follows a clear sequence:

1. HR uploads the default daily meal plan, which automatically creates one regular **meal_requests** entry per employee for each applicable meal type.
2. An employee who needs an additional meal submits a booking request with a quantity and reason from their own portal. This request is created with status *Pending*.
3. HR reviews pending extra requests and either approves or rejects them. Only approved requests become visible on the vendor’s extra-orders view.
4. At meal time, the vendor sees both the regular list (from HR’s default plan) and the approved-extra list, and marks each employee’s entry as *Served* once the meal has actually been provided.

5. Both HR and the vendor can then review analytics that break down, for any date range and meal type, how many meals were ordered, approved, rejected, served, or are still pending – resolving the ambiguity that previously caused billing disputes.

Chapter 5

Implementation

5.1 Technology Stack

The system was implemented using the following technologies:

- **Front end:** Angular (latest version) with TypeScript.
- **Back end:** Node.js with the Express.js framework, exposing a REST API.
- **Database:** MySQL.
- **Authentication:** JSON Web Tokens (JWT), with access/refresh token pairs and role-based route guards.

5.2 Authentication Module

The authentication module provides login, registration, token refresh, and logout endpoints. On successful login, the back end issues a short-lived access token and a longer-lived refresh token, the latter persisted in the `refresh_tokens` table so that sessions can be revoked on logout. Role-based guards on both the Angular routes and the Express middleware ensure that employees, HR staff, and vendor staff can only reach the views and endpoints appropriate to their role.

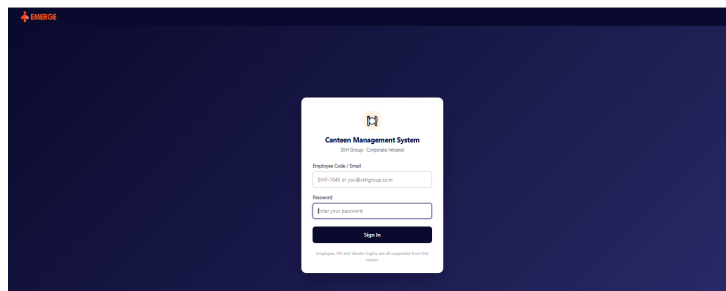


Figure 5.1: Authentication Page

5.3 Meal Booking and Approval Module

This module implements the core business logic described in Chapter 4: automatic generation of regular meal entries from HR’s daily plan, employee-submitted extra requests, and the HR approval step that gates whether an extra request becomes visible to the vendor. Both the employee and HR portals present this data through a shared `MealRequestService`, which keeps the front end responsive through optimistic local updates while persisting changes to the back end.

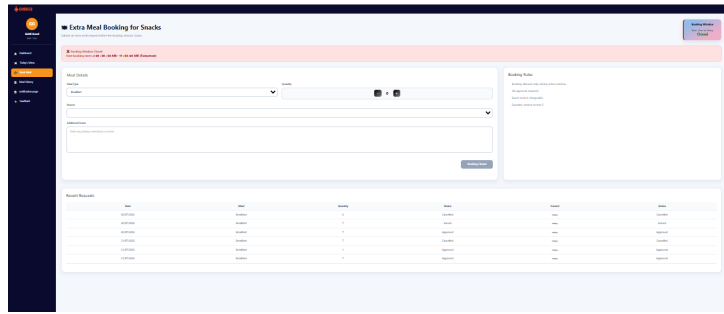


Figure 5.2: Book Meal Request

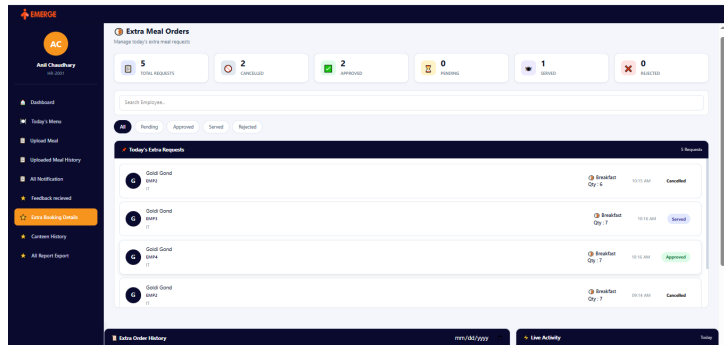


Figure 5.3: Approve Meal Request

5.4 Vendor Verification Module

The vendor portal presents regular and approved-extra requests together in a single verification view, allowing vendor staff to mark each employee's entry as served. Summary cards on this view show, per meal type, how many regular meals have been served, and how many extra requests are approved, rejected, served, or still pending – giving the vendor an at-a-glance reconciliation view for each meal period.

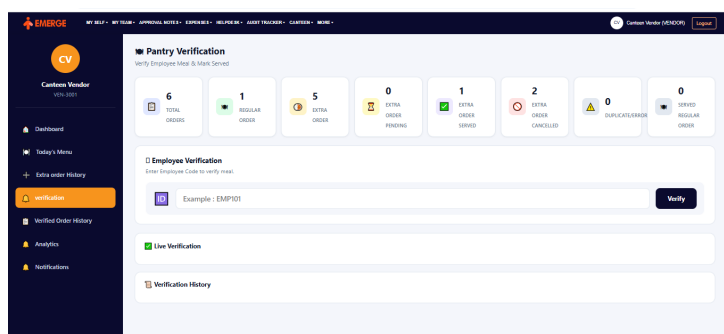


Figure 5.4: Verify Meal Request

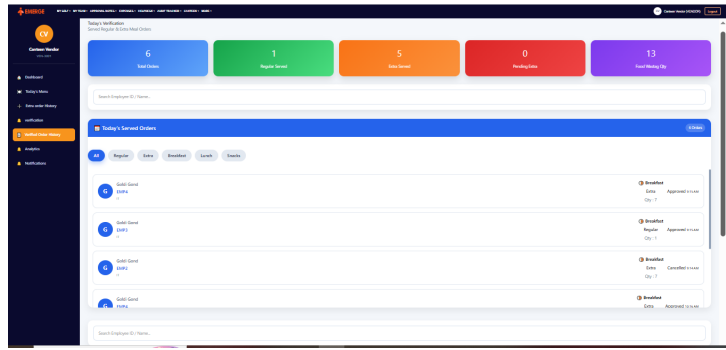


Figure 5.5: Verified Meal Request

5.5 Analytics and Reporting Module

Both the HR and vendor portals include an analytics view that aggregates meal-request data by status (ordered, approved, rejected, served, pending) and by meal type (breakfast, lunch, snacks), over a day, week, year, or custom date range selected by the user. The same filtered dataset drives both the on-screen chart and the underlying table, and can be exported as a structured PDF or Excel report with clearly named columns (employee ID, employee name, date, meal type, quantity, status, and order type), so that reports can be used directly in HR-vendor reconciliation meetings.

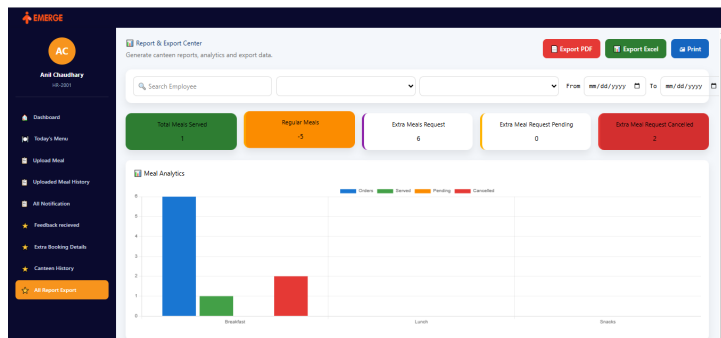


Figure 5.6: Analytics View

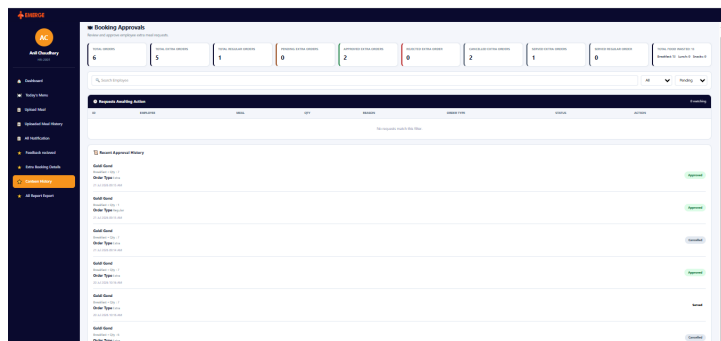


Figure 5.7: All About Canteen History

5.6 Notifications and Feedback Module

The notification service allows HR to broadcast announcements (such as menu publication) that are visible to both employees and the vendor, and it notifies employees of the outcome of their extra meal requests. The feedback module lets employees rate meals and leave comments, which HR can review and respond to, closing the loop between kitchen quality and the people consuming the food.

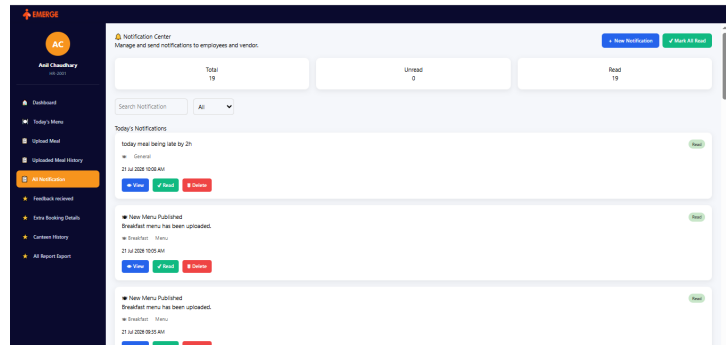


Figure 5.8: Notification Tab

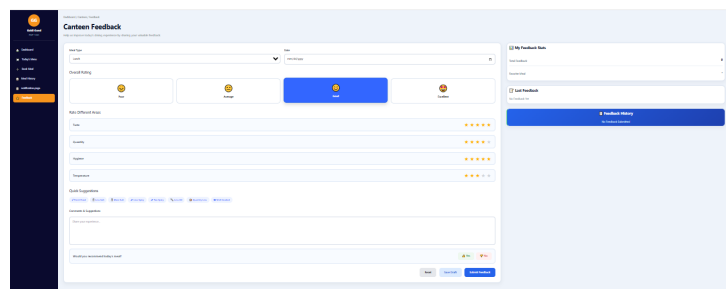


Figure 5.9: Feedback Form

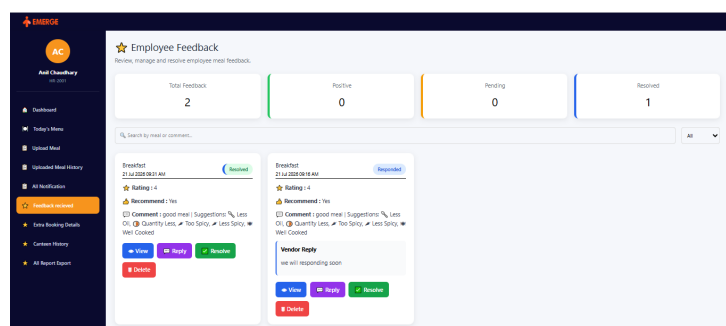


Figure 5.10: Feedback Recieved

5.7 Deployment

The completed application was deployed to production hosting so that it could be accessed and evaluated over the internet rather than only on a local development machine. The front end, back end, and database were hosted separately, each on a platform suited to that layer:

- **Front end (Angular):** deployed on Vercel.
- **Back end (Node.js/Express API):** deployed on Render.
- **Database (MySQL):** hosted on TiDB Cloud, a MySQL-compatible managed database service.

The deployed application is available at: <https://skh-internship-seven.vercel.app>

Layer	Hosting Platform
Front end	Vercel
Back end	Render
Database	TiDB Cloud

Table 5.1: Deployment Platforms

Chapter 6

Testing and Results

6.1 Testing Approach

Testing was carried out iteratively throughout development. Each module was exercised end-to-end – from the Angular UI, through the Express API, to the MySQL database – with particular attention paid to the points where the system’s core guarantee (accurate separation of regular and extra meal counts) could be silently broken by a mismatch between the front end and the back end.

6.2 Results

End-to-end testing confirmed that:

- Login, session refresh, and role-based access control function correctly for all three roles.
- HR’s daily meal plan correctly generates one regular entry per employee per applicable meal type.
- Extra meal requests are correctly withheld from the vendor’s view until approved by HR.
- The vendor’s served counts are correctly matched against employee identifiers, keeping regular and extra totals independently traceable.
- Analytics dashboards and their PDF/Excel exports reflect the same filtered data, for any selected meal type, status, and date range.

Overall, the system successfully replaces the previous manual process with an auditable digital workflow, giving both HR and the vendor a shared, verifiable record of every meal served.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

This internship project delivered a working Canteen Management System for SKH Group that directly addresses the recurring HR-vendor billing conflict caused by an unverifiable manual meal-tracking process. By requiring every meal to be either an HR-authorised regular entry or an explicitly HR-approved extra request, and by having the vendor mark meals as served against a matched employee record, the system provides a single, auditable source of truth that both departments can rely on. Beyond resolving disputes, the system also supports HR and the vendor with analytics and exportable reports intended to reduce food wastage by giving the kitchen a more accurate picture of expected demand.

7.2 Future Work

Potential extensions to the system include:

- Integration with an RFID or QR-code scanner at the serving counter, to further speed up and verify the vendor's mark-as-served step.
- Integration with payroll, to automatically account for reimbursable extra meals.
- Predictive analytics to estimate expected meal demand from historical booking patterns, further reducing food wastage.
- A mobile application version of the employee portal for booking and viewing meal history on the go.

Bibliography

- [1] Auth0, *JSON Web Tokens Introduction*, Auth0 Documentation, 2024.
- [2] Google, *Angular Documentation*, angular.dev, 2024.
- [3] OpenJS Foundation, *Express.js Documentation*, expressjs.com, 2024.
- [4] Oracle Corporation, *MySQL 8.0 Reference Manual*, dev.mysql.com, 2024.