

PROJECT ABSTRACT

Mess Menu Feedback and Rating System

A feedback loop for hostel messes: weekly menus, per-meal star ratings, dish-level analytics and contractor scorecards. Delivered built-to-order with the full application, report, PPT and viva kit.

01. Title

02. Introduction / Background

Mess food is the most complained-about and least measured service on any residential campus. Committees argue from anecdotes — 'the dal is always bad' — while contractors dismiss complaints as a vocal minority. Nobody has numbers. This project builds a mess feedback system: the weekly menu published in advance, per-meal star ratings from verified residents, dish-level analytics (which dishes consistently score below 3.6), and a monthly contractor scorecard that turns opinions into data the committee can act on.

03. Problem Statement

Without structured feedback, menu decisions are political and quality problems persist because they are never quantified. The project addresses this with a rating loop: residents rate each meal once (one rating per meal per student keeps it fair), tag what stood out, and write comments; the system aggregates ratings by meal slot, day and dish, auto-flags chronic underperformers, and publishes committee responses — so students see that feedback causes change.

04. Objectives

- Publish the weekly mess menu with per-dish ratings from prior weeks.
- Collect one-rating-per-meal-per-student feedback with tags and comments.
- Compute dish-level, slot-level and trend analytics with auto-flagging of weak dishes.
- Generate a monthly contractor scorecard for the mess committee.
- Let the committee publish responses to feedback visible to all residents.
- Deliver the complete student kit: application, report, PPT and viva Q&A.;

05. Proposed Methodology

The system is developed in three stages. (1) Menu management: the committee enters the weekly menu (breakfast/lunch/snacks/dinner per day); residents browse it with historical dish ratings. (2) Rating: after each meal window, verified residents submit a star rating plus structured tags (tasty, bland, served late, less quantity) and optional comments; duplicates are blocked per student per meal. (3) Analytics: rolling averages by dish/slot/day, trend detection, automatic flagging of dishes below threshold for two consecutive weeks, and scorecard export.

06. System Architecture / Working

A role-based web application: residents (rate, comment, browse menus), committee members (publish menus, respond to feedback, view analytics) and read-only contractor view of the scorecard. The backend stores menus, ratings and comments in a relational database and serves aggregate queries for the analytics dashboards; a scheduler opens and closes rating windows per meal slot.

07. Hardware / Software Requirements

- Server: shared hosting or small VPS
- Backend: Python (Django/Flask) or Node.js with PostgreSQL/MySQL
- Frontend: HTML5, CSS3, JavaScript (mobile-first; rating takes under 30 seconds)
- Scheduled jobs for meal-window rating open/close
- Email or notice-board integration for committee responses

08. Expected Outcomes

A working system where a week's menu can be published, rated meal-by-meal, and analyzed by dish. Expected behavior: duplicate ratings are rejected, analytics update as ratings arrive, and the scorecard reflects the real data. Participation-rate expectations are design targets; the report documents how the buyer can run a pilot to measure actual adoption.

09. Applications

- College and university hostel messes
- Corporate cafeterias and canteens
- School meal programs
- PG and co-living kitchens

10. Future Scope

- Nutrition tracking per meal for health-conscious residents
- Pre-order counts to reduce food wastage
- Multilingual feedback intake
- Integration with biometric attendance for verified diner counts