

NAME	SAPID	ROLL-NO
Umed Jogi	70612400043	A022
Purva Sagar Kamerkar	70612400055	A025
Maria Lokhandwala	70612400009	A034

Microservices and Architecture

PROJECT-SYNOPSIS

MealMonitor

We are developing a mobile application tailored for college campuses to improve food quality awareness and empower users to make informed dining decisions. The app serves as a collaborative platform where students, faculty, and staff can share real-time reviews of canteen food items, complete with photos, ratings, and optional anonymity.

Users can report negative experiences such as stale ingredients or poor preparation instantly, alerting others and helping them avoid similar issues. Positive reviews also guide peers toward recommended dishes, creating a dynamic and helpful food discovery experience. Community polls allow users to validate shared experiences, making feedback more reliable and democratic.

A dedicated canteen admin monitors feedback, responds to concerns, and posts updates when issues are resolved (e.g., fresh ingredients now in use). They can also announce new menu items and seasonal specials. An app admin ensures platform integrity by moderating content and managing user roles.

This app fosters a transparent, community-driven feedback loop that promotes food safety, enhances user satisfaction, and encourages accountability within campus dining services. It transforms everyday food choices into a shared experience, making the canteen not just a place to eat but a place to connect, improve, and enjoy.

Aim:

To create a community-driven food review platform within an educational institution that empowers students, faculty, and staff to share real-time feedback on canteen food quality, helping others make informed choices and encouraging accountability from the canteen administration.

Objectives:

1. Improve food safety by flagging poor-quality items.
2. Enable better food choices through peer reviews and ratings.
3. Establish a feedback loop between users and canteen admin.
4. Support transparency with updates on issue resolutions and new menu items.
5. Encourage engagement via polls, anonymous reviews, and dish recommendations.

Flow of App:

Step	Action	User Role
1	User orders food and experiences it	Student / Faculty
2	Posts a review with optional photo and anonymity	Student / Faculty
3	Other users get notified and vote on the review	All Users
4	Canteen admin sees flagged items and posts resolution	Canteen Admin
5	Admin announces new dishes or improvements	Canteen Admin
6	App admin moderates content and manages roles	App Admin

Conclusion:

This app fosters a culture of shared responsibility and transparency in campus dining. It empowers users to speak up, helps others make smarter choices, and holds the canteen accountable all while keeping the experience engaging and respectful.

Tech Stack Overview (Spring Boot Microservices)

◆ Frontend (React Web App)

Layer	Technology / Tools
Framework	React.js
Routing	React Router
State Management	Redux Toolkit or Context API
UI Library	Material-UI or Tailwind CSS
API Communication	Axios
Authentication	Firebase Auth or JWT-based custom auth
Hosting	Vercel / Netlify (Free Tier)

◆ Backend Microservices (Spring Boot)

Each microservice is built using **Spring Boot**, with independent responsibilities and databases.

Microservice	Stack
User Service	Spring Boot + Spring Security + JWT
Review & Poll Service	Spring Boot + Spring Data JPA

Canteen Service	Spring Boot + Spring Web
Notification Service	Spring Boot + Firebase Cloud Messaging (FCM) or Email API
Moderation & Admin Service	Spring Boot + Spring AOP + Admin APIs

Communication:

- REST APIs secured with **JWT**
- Optional **Spring Cloud OpenFeign** or **gRPC** for internal service calls

◆ **Databases**

Type	Technology / Provider
NoSQL	MongoDB Atlas (for flexible schemas like reviews, media)
SQL	PostgreSQL via Supabase or Neon.tech (for structured data like users, roles, issues)
ORM	Spring Data JPA with Hibernate

◆ **Infrastructure & DevOps**

Layer	Technology / Tools
API Gateway	Spring Cloud Gateway or NGINX
Service Discovery	Spring Cloud Eureka
Configuration	Spring Cloud Config Server
Containerization	Docker
Orchestration	Docker Compose (local), Kubernetes (cloud)
CI/CD	GitHub Actions , GitLab CI , or Jenkins

ERD DIAGRAM



