

Name	Dheva Dharshni S
Roll no	7376222IT133
Seat no	86
Project ID	10
Problem Statement	Faculty Log (Mailer BIT)

Technical Components:

Component	Tech Stack
Front End	React
Back End	Node.js with Express.js
Database	MongoDB (NOSQL Database)
API	OpenAPI

PROBLEM STATEMENT:

The decentralized nature of email communication within educational institutions leads to several challenges which includes:

- **Inconsistent messaging:** Different departments and administrative units send emails independently, leading to duplication of information and inconsistent messaging.
- **Schedule conflicts:** Students and faculty receive multiple emails with overlapping schedules and events, leading to confusion and missed opportunities.
- **Fragmented communication:** Important announcements and updates get lost in the volume of emails, making it difficult for recipients to stay informed and engaged.
- **Administrative burden:** Managing email distribution lists, resolving conflicts, and ensuring timely delivery of critical information impose a significant administrative burden on staff and faculty.

PROJECT FLOW:

Purpose:

To develop a centralized mailing system that efficiently manages communication regarding student, faculty schedules and activities, resolving existing issues of schedule conflicts and communication inconsistencies.

Scope:

This system includes user authentication, a mailer request form, conflict checks, and a real-time dashboard for viewing and managing schedules. It integrates with existing email systems to ensure scheduled and conflict-free messaging.

Business Context:

The centralized mailing system is aimed at enhancing communication clarity and timeliness across BIT, thus boosting organizational efficiency by minimizing scheduling conflicts. Primary stakeholders include students, faculty, administrative staff, and the IT department.

Consideration:

- All users possess active Google accounts for authentication.
- Users have regular access to internet-enabled devices.

User personas:

- **Faculty:** Requires the ability to send out and receive schedule updates and notices efficiently.
- **Admin Staff:** Manages system operations, resolves conflicts, and approves mail requests.

Functional Requirements:

- **User Authentication:** Secure login using Google OAuth.
- **Mailer Request Form:** Users input mail content, scheduling time, category, and recipients.
- **Dynamic Dashboard:** Real-time schedule viewing and interaction.
- **Priority Algorithm:** Automated prioritization of communications based on rules.
- **Conflict Check:** The system checks for scheduling conflicts.

FLOW CHART:

