



Student Name: TARRUN S

Seat No: 434

Project ID: 9

Project title: IQAC Mailer

Technical Components

Component	Tech Stack
Backend	Node.js with Express.js
Frontend	React.js
Database	MongoDB(NOSQL Database)
API	OpenAPI

Implementation Timeline

Phase	Deadline	Status	Notes
Stage 1		Under Review	Planning and Requirement gathering
Stage 2		In progress	Design and Prototyping
Stage 3		Not started	DB Designing
Stage 4		Not started	Backend Implementation
Stage 5		Not started	Testing & Implementation

PROBLEM STATEMENT:

The decentralized nature of email communication within educational institutions leads to several challenges, including:

Inconsistent messaging: Various departments and administrative units send emails separately, resulting in information duplication and inconsistent messaging.

Schedule conflicts: Students and faculty members receive multiple emails containing conflicting schedules and events, causing confusion and missed opportunities.

Fragmented communication: Important announcements and updates are easily overlooked in the large volume of emails, making it challenging for recipients to stay informed and engaged.

Administrative burden: Managing email distribution lists, resolving conflicts, and ensuring prompt delivery of critical information place a significant administrative workload on staff and faculty members.

PROJECT-FLOW:

Purpose:

The purpose is to create a centralized email system that effectively handles communication related to student schedules and activities, addressing current issues of conflicting schedules and inconsistent communication.

Scope:

The scope of the project includes features such as user authentication, a form for sending emails, conflict checks, and a real-time dashboard for managing and viewing schedules. The system will be integrated with existing email systems to ensure scheduled and conflict-free messaging.

Business Context:

The centralized email system aims to improve communication clarity and timeliness within BIT, thereby increasing organizational efficiency by minimizing scheduling conflicts. The primary stakeholders include students, faculty, administrative staff, and the IT department.

Consideration:

- All users must have active Google accounts for authentication.
- Users should have regular access to devices with internet connectivity.

Dependencies:

- Integration with Google OAuth for secure user authentication.
- The existing email server should consistently perform well and be available.

User Personas:

- Student: Requires an up-to-date schedule to effectively plan activities.
- Faculty: Needs the ability to send schedule updates and notices efficiently.
- Administrative Staff: Responsible for managing system operations, resolving conflicts, and approving email requests.

User Stories:

- As a student, I would like to easily access a comprehensive schedule of my classes and events to plan my day effectively.
- As a faculty member, I need to ensure that my communications reach students without conflicting with their other scheduled activities.

Functional Requirements:

- User Authentication: Implement a secure login system using Google OAuth.
- Mailer Request Form: Provide users with a form to input email content, scheduling details, categories, and recipients.
- Conflict Resolution: Automatically detect scheduling conflicts and provide options for resolution.
- Dynamic Dashboard: Develop a real-time dashboard for users to view and interact with schedules.
- Priority Algorithm: Create an automated algorithm to prioritize communications based on predefined rules.

FLOW CHART:



