Communication Systems for Schools

A major project report submitted in partial fulfilment of the requirement

for the award of degree of

Bachelor of Technology

in

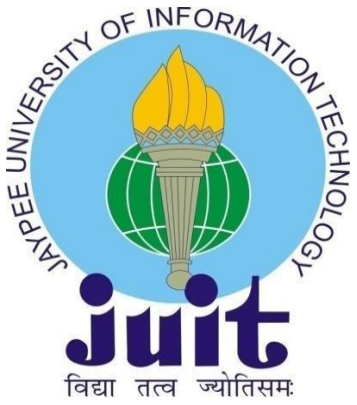
Computer Science & Engineering

Submitted by

Aayush Girdhar (211548)

Under the guidance & supervision of

Ms Seema Rani & Mr Gaurav Negi



Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology,

Waknaghat, Solan - 173234 (India)

May 2025

Supervisor's Certificate

This is to certify that the major project report entitled '*Communication System for Schools*', submitted in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Information Technology**, in the Department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology, Waknaghat, is a bona fide project work carried out under my supervision during the period from January 2025 to May 2025.

I have personally supervised the research work and confirmed that it meets the standards required for submission. The project work has been conducted in accordance with ethical guidelines, and the matter embodied in the report has not been submitted elsewhere for the award of any other degree or diploma.

Date:	Supervisor Name: Ms Seema Rani	Supervisor Name: Mr Gaurav Negi
Place:	Designation: Assistant Professor	Designation: Assistant Professor
	Department: Dept. of CSE & IT	Department: Dept. of CSE & IT

Candidate's Declaration

I hereby declare that the work presented in this report entitled *'Communication System for Schools'* in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Information Technology** submitted in the Department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology, Waknaghat is an authentic record of my own work carried out over a period from January 2025 to May 2025 under the supervision of **Ms Seema Rani & Mr Gaurav Negi**.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Name: Aayush Girdhar

Roll No.: 211548

Date:

This is to certify that the above statement made by the candidates is true to the best of my knowledge.

Supervisor Name: Ms Seema Rani	Supervisor Name: Mr Gaurav Negi
Designation: Assistant Professor	Designation: Assistant Professor
Department: Dept. of CSE & IT	Department: Dept. of CSE & IT

Date

Acknowledgment

First and foremost, I extend my heartfelt gratitude to the Almighty for His divine blessings, which allowed me to complete the project *'Communication System for Schools'.*

I am really grateful and wish my profound indebtedness to supervisors **Ms Seema Rani** Assistant Professor (Grade-I), and **Mr Gaurav Negi**, Assistant Professor (Grade-I), Department of CSE Jaypee University of Information Technology, Waknaghat. The deep Knowledge & keen interest of my supervisors helped me to carry out this project in the field of "Education Technology". Their endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

Additionally, I express gratitude to the department's teaching and non-teaching staff, whose timely help and support facilitated the smooth progress of our work.

I am also thankful to **Toddle**, the company where I am currently interning, for their resources and support, which played a vital role in enriching my experience during this project.

Lastly, I acknowledge and deeply appreciate my parents' enduring patience, encouragement, and unconditional support, which have been a constant source of strength and motivation throughout this journey.

With gratitude,

Aayush Girdhar (211548)

Table of Contents

List of Figures

List of Tables

Abstract

Proper communication is indispensable for facilitating coordination among teachers, students, and parents in academic institutions. Traditional methods such as paper notices and verbal announcements are unproductive, personal, and not scalable. This report presents a centralized Communication System for Schools that overcomes such disadvantages by embedding sophisticated features such as announcement broadcasting, real-time messaging.

The system can deliver accurate and quality content by class, program, and grade targeting, soft delete with recovery, and a systematic review process. The system also supports daily collaboration by broadcast announcements and secure, asynchronous discussions.

The system, which has been developed based on React.js, GraphQL, Apollo Client, and PostgreSQL, has been deeply optimized to enhance data integrity, scalability, and performance. The system is adaptive, supports multi-devices, and has been well tested to ensure reliability for concurrent use. In general, this approach significantly enhances communication between educational institutions, simplifies administrative procedures, and facilitates a more integrated and educated academic community.

Chapter 1

Introduction

1.1 Introduction

There is no question that communication is a foundation of a conducive and collaborative learning environment in schools. It acts as the binding tissue among learners, parents, and educators ensuring clarity, mutual understanding, and timely intervention. Communication is not just an ancillary role in the contemporary education environment but a key enabler of both academic achievement and administrative effectiveness. Older forms of dissemination e.g., printed flyers, oral directions, or blackboard classroom announcements usually fail in the fast-paced and digitally interconnected environment of today. These are generally inefficient, unscalable, and inherently unable to provide personalized or real-time interaction with stakeholders.

Having realized these inadequacies, schools have begun migrating towards combined digital platforms which can optimize and enhance their internal and external networks of communication. For this purpose, the combined School Communication System with both the Announcement Module and the Messaging Module provides a timely and revolutionary remedy.

The Announcement Module fills the need for formal communication that is one-to-many. It provides for teachers and administrators to make necessary announcements to the proper groups. Class-targeting, live-updating, and easy deletion of messages have been accomplished well. These features aid in sending timely, appropriate, and prompt communications based on changing needs of the institution. The system has also been designed with regard to cognitive load, thereby providing users with only relevant information without exposing them to cognitive overload.

Aiding this is the Messaging Module, which introduces the aspect of interactive, two-way communication. This module supports both one-to-one discussions and group discussions through an element referred to as channels. One such enhancement is the Staff Directory

Submodule, which enables the school administration to create pre-defined communication directories, like finance, academics, or counselling, by assigning appropriate faculty members to corresponding categories. The feature allows parents or guardians to communicate directly with the concerned staff by selecting the appropriate directory, thus eliminating delay, misdirection, and unnecessary complexity in the resolution of queries. It allows a contextual and effective model of communication between the school and parents, thus enhancing trust and transparency.

Together, such modules embody a holistic method of school communication - the integration of top-down broadcasting with bottom-up conversation. This two-way channel model has the double benefit of ensuring not just that information passes effectively from the institution to its stakeholders, but also that concerns, questions, and feedback emerge in an organized way from below. The overall communication architecture thus reflects a proactive and adaptive stance, one that aligns with the requirements of 21st-century education.

This combined system is a deliberate step away from old-fashioned, fragmented communication methods to an integrated, tech-enabled paradigm. It aims to make school communication seamless, complete, and actionable - contributing significantly to forging an educated, interactive, and cohesive learning environment.

1.2 Problem Statement

Good communication among students, their families, and institutions is the basis for the effective functioning of any school system. The traditional means of communication such as printed notices, email, or word of mouth never quite do the job with expanding requirements for scale, individualization, and accuracy. Legacy systems reliant on these traditional means are sure to lack the ability to support dynamic, real-time, and role-based communication. Greater institutional complexity and greater expectations regarding transparency and responsiveness generate a self-evident demand for a better and more complete solution.

1.2.1 Inefficiency and Scalability

Traditional modes of communication are slow and cumbersome. Educators and administrative staff have to bear the hassle of preparing and distributing materials, which can, in most cases, result in significant delays or even breakdowns in transmission.

- **Time-Consuming Tasks:** Activities such as updating address lists, printing out a notice, or message delivery confirmation waste valuable time and energy that can otherwise be devoted to the main scholarly work.
- **Limited Reach and Delays**: These systems are plagued by scalability constraints as enrolments increase among students. The dissemination of critical information to various stakeholders is often slow, inaccurate, or loses updates.

1.2.2 Lack of Personalization and Contextual Communication

One-size-fits-all generic messages are unlikely to meet the needs of various stakeholders. Role-based, class-based, or department-based messages are hard to handle with conventional tools.

- **Increased Teacher Workload**: Customizing the notifications for various groups, like students, parents, or internal staff, adds to the already heavy workload of teachers.
- **Low Relevance and Engagement**: Generalized messages do not appeal to everyone, and therefore, they lose interest and are not effective.

1.2.3 Risks of Miscommunication and Fragmentation

The lack of an integrated communication structure is one reason for the risk of miscommunication. Manually delivering can result in information not being delivered correctly or at all.

- **Confusion and Delays**: Notifications that get to the wrong audience or don't arrive on time may result in chaos, late deadlines, or being unable to fulfil responsibilities.
- **Inconsistent Messaging**: Unconnected communication channels are responsible for inconsistency in presenting information to students, parents, and workers.

1.2.4 Lack of Functional Depth and Control

Legacy systems don't possess the high-level functionalities required to effectively process and optimize communications. Elementary features like increased search, filtering of content, or the determination of user involvement are not normally available.

- **Organizational Challenges**: Without categorization or history management, teachers have difficulty navigating or consulting previous communications shared with parents, students, other faculty members, etc..
- **Reduced Accessibility**: Users frequently need to search long lists of messages to find an individual message, and may take longer to do this when no indexes or filters exist.
- **No Visibility on Engagement**: Traditional methods offer no clues about whether the message was seen, read, or comprehended, making follow-up and accountability difficult.

1.2.5 Absence of Real-Time, Two-Way Communication

The most important limitation of the traditional systems may be the lack of support for real-time, interactive communication. Parents or students tend to use indirect, delayed channels to communicate with particular departments or staff members.

- **Delayed Query Resolution**: Without formal, structured communication channels or directories, parent or student inquiries can get misdirected, delayed, or go unanswered.
- **Lack of Structured Interaction**: There is no uniform method to enable dialogue with specific departments (e.g., finance, counselling), and institutional communication becomes reactive instead of proactive.

1.3 Objectives

The goal of this project is to modernize the school communication by removing inefficiencies of the traditional channels and replacing them with a responsive and inclusive communication platform. By bringing the interactions of teachers, students, and parents under a single platform, this system encourages prompt, contextual, and accessible exchange of information, ultimately augmenting stakeholder engagement and educational coordination.

1.3.1 Simplify Communication Processes

- **Centralized Hub**: Create one digital platform for announcements, queries by parents or students, information, and communications and make it accessible to all stakeholders, parents, students, and teachers through one common platform.
- **Instant Feedback**: Guarantee real-time delivery and receipt of messages, minimizing delays and keeping every user informed constantly.
- **Multi-Channel Distribution**: Support multiple channels of communication such as in-app messaging, emails, to support user choices and ensure high reach.

1.3.2 Enable Personalized Communication

- **Effortless Posting**: Give teachers and administrators intuitive tools to initiate communication with concerned students, parents, and faculty members quickly and accurately.
- **Customizable Messaging**: Targeted messaging to classes, departments, or even parents to allow for relevance and clarity of communication.

1.3.3 Enhance Information Discovery

- **Advanced Sorting and Filtering**: Equip users with options to filter announcements and messages based on criteria such as class, date, category, or recipient.
- **Search Functionality**: Include the features of keyword and time search to make it easy locating specific announcements or updates.

1.3.4 Improve Accessibility and System Features

- **Role-Based Dashboards**: Provide customized dashboards according to the user students teachers, or the parents: showing only the most relevant information.
- **Real-Time Syncing**: Synchronize all changes instantly in all connected devices in order to maintain consistency and remove information gaps.
- **Message Recovery**: Implement a soft delete or recovery mechanism to prevent accidental deletion and ensure preservation of critical communications.
- **Multilingual Support**: Offer language options to accommodate varied linguistic backgrounds in school communities and promote inclusivity.

1.3.5 Ensure Ease of Use and Broad Access

- **User-Friendly Interface**: Provide a smooth and intuitive access using web and mobile applications to make users always connected wherever they are.
- **Cross-Platform Accessibility**: Provide seamless access through web and mobile applications to help the users stay connected anywhere anytime.

1.4 Significance and Motivation of the Project Work

Effective communication is the foundation of all successful institutions and schools are not an exception either. Open information interchange among educators, students, and parents is necessary for creating a supportive and collaborative learning environment. Nevertheless, traditional communication mechanisms are usually inadequate to serve the dynamic needs of the contemporary communities that are connected digitally. This project attempts to close that gap by introducing a smart, scalable, inclusive communication platform targeted towards educational institutions.

1.4.1 Significance

- **Enhancing Communication Efficiency**: Through automation and centralized distribution, the system ensures timely and correct dissemination of announcements and up-dates. This reduces confusion, removes redundancy, and enhances stakeholder coordination.

- **Personalized Engagement**: Targeted messaging makes it possible to send appropriate and specific information to the target audience whether students, parents or staff thus creating greater engagement, inclusion and relevance of every message.
- **Time and Resource Optimization**: Outsourcing the monotonous administrative functions of categorizing, filtering, and circulating announcements, automates a lot of work for educators and administrative staff, giving them more time to concentrate on pedagogy and mentorship.
- **Inclusivity through Accessibility**: With multilingual respect and cross-platform accessibility (web and mobile), the system provides for the multifaceted technological and linguistic cultures in school communities, making the platform accessible and fair for all.

1.4.2 Motivation

- **Addressing Legacy System Challenges**: The core motivation of this project is to eliminate the inefficiencies, the scalability issues, and frequent miscommunications in the conventional systems. The project offers a simplified and dependable alternative by reinventing the way schools communicate.
- **Leveraging Technology in Education**: By reimagining how schools communicate, the project provides a streamlined and reliable alternative.
- **Fostering a Connected School Community**: Strong communication cultivates trust, transparency and collaboration. This platform seeks to help between the educators, the students and the parents to become closer, which will result in better educational outcomes and a closer cohesive academic ecosystem.
- **Scalable and Future-Ready Infrastructure**: The project is designed as a scalable solution that can be adapted to shifting user needs, institutional needs, technological trends. It sets the scene for a strong system of communication that is timeless.

1.5 Organization of Project Report

This report is organised in a way that it is able to give a wide picture of the project, how it developed, and what it led to.

Chapter 1: Introduction

The target gives an overview of the project's goals, purpose, and importance, addressing the problems of traditional announcement systems and the necessity for a new solution.

Chapter 2: Literature Survey

In this chapter, the existing research and systems of school communication are discussed in the light of their strengths, weaknesses and limitations. It sets the context for the project by pointing out the gaps of the existing solutions.

Chapter 3: System Development

This chapter outlines the design and implementation of the communication system and its architecture, tools and technologies, and the problems encountered during implementation. It also gives an overview of methodologies used.

Chapter 4: Testing

The procedures and materials used in the evaluation of the system's functionality and performance are described in the testing chapter. It also outlines results of diversified tests to ensure effectiveness of the system.

Chapter 5: Results and Evaluation

This chapter shows the results of the project, evaluating the effect of the system on the communication practices in schools. It also compares the system with the existing solutions to show its benefits and effectiveness.

Chapter 6: Conclusions and Future Scope

The last chapter provides a summary of the key findings of the project and lists its contributions, as well as suggestions for future improvements and scalability.

Chapter 2

Literature Review

2.1 Overview of Relevant Literature

School announcement and messaging management systems are now integral parts of a winning and integrated learning environment. These are the building blocks for enabling parent-teacher-student communication, especially in the light of accessibility, standards, real time collaboration models and web technologies.

Electronic platforms for announcements and messaging are an excellent scalable alternative to the traditional practice, which is always inefficient and myopic. Apart from announcement management, the new systems also have messaging capabilities which allow teachers and students and their parents to communicate immediately and personally.

Messaging is an essential pipeline to control urgent necessities, ask questions, or report updates that require one-to-one communication. They make it easier to communicate in real time and receive important messages such as reminders for events, assignments or breaking news notifications etc.

The chapter reviews critically the literature on such integrated systems, in terms of technological strategy, platforms and methodologies in the management of both school messaging and announcements. The review presents the background of the manner in which the systems have developed with time as far as specific needs of schools are concerned in the dissemination of timely, relevant, and targeted information. This section reviews loopholes in current systems based on analysis of the merits and demerits of the current solutions and highlights the need for more sophisticated and intuitive solutions that offer announcements management and direct messaging functionalities. The target is to develop an integrated communications platform which not only maximizes the distribution of critical information but also allows direct, personal communication thereby maximizing the overall level of communications effectiveness in schools.

Table 2.1: Literature Review

S. NO.	Source	Tools/ Techniques	Key Findings/ Results	Limitations/ Gaps Identified
1.	Organization's Internal Documentation	ReactJs [1], Redux, Git, Apollo Client, GraphQL [2], GitHub, Version Control	All the needed documents used to facilitate a head start in the contribution of the existing platform are found in the organization's internal documentation. It is from the data on the front end, backend and version control, cross-platform development and also dev-ops, which is ordinarily based on CI/CD methods, or continuous integration/continuous development. Also, contains the API for the design system used in the product.	As the documentation is non-disclosable and is internal to the organization therefore, it is not possible to share it with persons who are outside the organization. Also, the documentation is interested in the required use cases of the technologies in the application failing to pick on the potential better use cases.

2.	ReactJS Documentation [1]	ReactJS, State Management [1]	React is a JavaScript library that allows us to create scalable applications through JavaScript. It simplifies the management of the state because of the in-borne functions and the architecture that is provided. It offers a strong library for developing interactive UIs with sexy state management needed for dynamic behaviour management systems.	React is a mere frontend library; it does not have a solution for backend integration and real time collaboration. Most of the time, it requires third-party tools such as Apollo Client for complex applications.

3.	React Native Documentation	React Native, Cross-Platform Development	React Native, an upgrade of React, allows cross platform mobile app development with a single code base which greatly reduces time and effort for development and learning.	Since the code is not written in the platform's native language, it cannot provide impressive performance of compute intensive applications compared to native apps. It relies on integration of third party libraries, for high-order features, for implementation.
4.	Apollo GraphQL Docs [2]	Apollo Client, GraphQL [2]	Apollo GraphQL [2] is a very resourceful tool that allows very efficient data fetching and state synchronization between the front and back end, using GraphQL. It also facilitates real time updates using GraphQL	The performance of the applications can suffer and degrade due to poorly optimized queries, or significantly bigger data sets. It also mandates a certain setup to manage several error states and

			subscriptions.	consistency of data.
5.	PostgreSQL Documentation [3]	Relational Databases	A relational database, PostgreSQL provides an efficient database solution for managing relational data, that also supports usage of complex queries and maintains data integrity ranging from small applications, all the way to large-scale applications.	While it is effective for relational data, managing scalability in real-time applications with high concurrent loads may require additional caching layers or database optimizations that users generally ignore which is considered a bad practice.
6.	Web Content Accessibility Guidelines (WCAG)	Accessibility Standards	These guidelines provide truly comprehensive guidelines for creating access to the application for those with disabilities. They are necessary for keyboard navigation and usability	Provides extensive guidelines on how to have applications accessible to users with disabilities. Critical for keyboard navigability as well as usability improvements.

			improvements on the platform because there are times when the user cannot use the mouse or wants to use only a keyboard to navigate.	
7.	AWS Cloud White Paper [10]	Cloud Storage, Scalability	It offers a scalable storage option for handling large datasets and data availability across all regions all over the world, which is essential to institutions of learning that have international users.	Higher costs for small-scale institutions. Integration also poses a problem with the on-premise data systems.
8.	Jira Documentation	Task Management	Allows to gain insights about handling the incidents with state-based workflows, focusing on draft, published and resolved states.	Few real-time features in standard set up. Each has the plugins or customizations needed in specific needs in education management.

2.2 Key Gaps in Literature

The examination of the given technologies and frameworks still has the following gaps to be filled even though they provide precious information on how to develop communication modules:

- **Real-Time Scalability**: While existing systems such as Firebase [5] and Apollo client have real-time capability, these systems will have to be greatly improved to manage the number of concurrent users of the big institutions.
- **Cross-Platform Usability**: Maintaining smooth user experience on internet and mobile platforms, in line with accessibility standards, is still problematic.
- **Incident Categorization and Reporting**: Comprehensive reports have to be produced manually because automated methods for classifying and judging behavioural events are in their infancy.
- **Cost-effectiveness**: Scalable solutions in the form of cloud based platforms such as AWS may be out of reach for smaller schools with little resources. [10]
- **Integration with Current Systems**: It can be challenging to incorporate new technologies into old school management systems that may lead to inconsistent or useless practices.

Chapter 3

System Development

3.1 Requirements and Analysis

The Announcement and Messaging System for Schools is used to have a centralized communication platform for smooth and effective communication among teachers, students and parents. The system is designed to address communication by providing a consolidated solution for announcements and direct messaging that is also easy to use and can be scaled.

3.1.1 Functional Requirements

- **Creation, Editing, Deletion and Restoration of Announcements**: Teachers and administrators should be able to be able to create, edit, delete and restore announcements easily. The platform should make it easy to manage announcements smoothly, and flexible in communication.
- **Class-Specific Filtering**: Announcements should be filtered by specific classes so that only information that is relevant to the designated audience, whether specific class announcements or general updates for parents are viewed.
- **Real-Time Syncing**: All the changes in the system whether announcements or messages should be saved and synchronized immediately. This guarantees that any data is not lost, and updates are indicated immediately in all devices.
- **Soft-Delete and Restore Functionality**: The system should enable users to soft-delete announcements, which, upon the user's restoration command, will be sent to the bin where they can be retrieved. This feature, on the other hand, must be dynamic to enable user control and recovery of deleted data without any challenge.
- **User Authentication and Role-Based Access**: Role-based authentication must also be deployed to the system where only teachers or administrators, who are authorized users, can create, edit and manage announcements. Students and parents will have limited access, mainly to read announcements and get messages.

- **Messaging Integration**: Besides announcements the system should allow direct message sending between teachers, students and parents. This feature should have real-time support to encourage more personalised one to one interactions when required.
- **Real-Time Messaging**: Message exchange between teachers, students and parents should be in real time. The system should be able to complete instant message delivery and update to eliminate delays on communication. This permits teachers to send either direct messages having to do with assignments, feedback, or urgent updates to students or parents.
- **Two-Way Communication**: The messaging support should work both ways (send and receive messages), i.e. a teacher should be able to initiate conversation with students or parents, and vice versa. This may be quite handy for personalized communication, for example clarifications relating to assignments or to pointing out a specific concern, etc.
- **Group Messaging**: Apart from messaging in a one-on-one mode, the system should support group chats. Teachers can form groups for certain classes or projects allowing for communication of all participants (teachers, students and parents) at once. This is fundamental for class wide upgrades, discussion groups or planning of events.
- **Push Notifications for New Messages**: Users should be notified about new messages, so they can respond timely. This may be in-app notifications, email notifications depending on user preferences. This feature serves to keep all the stakeholders in the know at all times.

3.1.2 Non-Functional Requirements

- **Scalability**: The system should tolerate many users concurrently without appreciable performance and because of that the system should not degrade. As more users join the platform it should keep functioning smoothly and effectively.
- **Accessibility**: The platform needs to adhere to the Web Content Accessibility Guidelines (WCAG) in order to assure page accessibility for all users including the disabled. This comprises screen readers support and keyboard navigation so that everyone has equal access.

- **Cross-Platform Compatibility**: The system should be working perfectly well in different devices such as tablets, smartphones and PCs. It should offer a uniform user experience on all platforms.
- **Reliability**: The system needs to be very reliable with least downtime. Announcements and messages should be available at all times, with little disruptions to the users.
- **Usability**: The interface should be intuitive, so that teachers, administrators, students, and parents are able to use the platform easily. Creating and managing announcements and messages should be relatively simple, and relatively easy to learn for new users.
- **Security**: The system should use secure storage mechanisms in order to secure the information of the user. This includes secure communications and privileged access to ensure that only approved people can access such information.

3.2 Project Design and Architecture

Scalability and maintainability are ensured by the system's modular design. There are three primary layers to the architecture.

3.2.1 Frontend

- **Framework**: GraphQL [2], Apollo Client [2], SASS, and ReactJs.
- **UI/UX**: Done using a unified design system, layered on top of an Ant design for responsiveness and dynamic interfaces. [11]
- **State Management**: Utilization of Redux State Management API to manage user and other application data.
- **Accessibility Features**: Included keyboard navigation, focus management, and WCAG-compliant components.

3.2.2 Backend

- **API**: GraphQL with Apollo server for handling queries and mutations. [2]

- **Database**: Used PostgreSQL for database management. [3]

3.2.3 Database

- **Database Management System**: PostgreSQL [3]
- **Schema Design**:
    - Tables for users, announcements, messages, attachments.
    - Relationships are defined to associate announcements and messages with multiple users (teachers and students).
    - Indexing for faster retrieval of data.
- **Data Integrity**: Implemented constraints and triggers to enforce consistency.

3.2.4 System Architecture

The system follows a three-tier architecture:

1. **Presentation Layer**: ReactJS for the client-side interface.[1]
2. **Business Logic Layer**: Apollo Server and custom logic for handling data operations and workflows.
3. **Data Layer**: PostgreSQL for relational data management. [3]

3.2.5 Frontend Folder Structure

The folder structure of the front end is incorporated into the application as follows.

- **Routes**: This folder contains the child routes of the said route.
- **Modules**: This folder contains the core JS logic of the route and its children's routes, which typically range from common functions, utils, GraphQL queries, mutations, fragments, etc. [2]
- **Components**: This folder contains the files that contain JSX, which is the HTML rendered in JS on the webpage. These files also contain other JS functions used in React to build the required components.

3.3 Data Preparation

The success of the system is based on structured and correct data. Key steps taken for data preparation for the Announcement System for Schools were:

- **Data Cleaning**: Deleted unnecessary records from imported announcement datasets. Standardised date, titles and recipient groups to maintain standards.
- **Data Validation**: Performed verification of records on announcement records for completeness and correctness. Edge cases that were tested include duplicate announcements and invalid recipients.
- **Initial Data Seeding**: Populated the database with sample data, including predefined users, classes, and test announcements, to facilitate system testing and demonstration.

3.4 Implementation

The implementation phase of the School Communication System concentrated on the development of modular scalable and user-centric sections to optimize announcement and messaging workflows. Two key modules, Announcements and Messaging, were developed for the unique but interrelated communication needs of educational institutions. Each module was designed in a way to support real time interactivity, easy use and institutional transparency. The Announcements module allows structured, AI enabled, planning,

scheduling and targeted publishing of school wide announcements while the Messaging module offers Instant, Secure, yet context rich communication between teachers, students and parents. PostgreSQL is used as the main database [3], taking advantage of the full text search option to enable better querying and as a result increased reliability of data integrity.

Powered by GraphQL with Apollo Server [2], a ReactJS frontend [1] using Material-UI guarantees responsive and accessible user experience [6], and app backend interactions made flexible and high performance. The total system elicits a balanced display of robust engineering, careful UX design, and carefully constructed functionality for dynamic school environments.

3.4.1 Key Modules

The Announcements and Messaging platforms at the core of the platform are used to promote clarity, responsiveness, and inclusiveness throughout the school community. Carefully crafted for teachers, students and parents, each module was designed to support the key functions of their characters, thus guaranteeing intuitive interrelations and presenting workable information exchange.

3.4.1.1 Announcements

Teachers can use AI to make announcements and come up with specific and individual for the class, topic or urgency. They can attach something if needed and choose only certain recipients or the entire class to whom to send the announcement. The system also enables saving of announcements as drafts or scheduling them for future publication.

Users can easily filter announcements using several filters such as:

- Publish Date
- Created By
- Class or Grade
- Recipient Type

There are the following categories into which each announcement may fall to streamline the cascade of the announcement:

- Draft
- Sent for Review
- Review Rejected
- Scheduled
- Published
- Deleted

3.4.1.2 Messaging

The messaging module allows real time direct communication between teachers, students and parents to facilitate ongoing collaboration and engagement.

- **Message Creation**: Users are able to open one-to-one or group chats. Teachers can contact individual students or parent groups to talk about academic issues or events or personal feedback.
- **Attachments Support**: Files, images, and documents can be posted directly on the interface of the chat for more context or resources.
- **Message Search**: Conversations can be filtered and searched by participant name, keywords or date range for easy navigation and retrieving purpose.
- **Notifications**: Through in-app, email or push notifications, users will be notified the moment they receive a message.
- **Chat History & Archiving**: All conversations are recorded and can be later retrieved to enhance continuity and traceability of school communications.
- **Message Categories**:
    - Unread
    - Read
    - Replied
    - Archived
    - Flagged for Follow-Up

3.4.2 Algorithms:

- **Search Optimization**: Used PostgreSQL's Full-Text Search feature for efficient query execution. [3]
- **Data Consistency Checks**: Automated scripts to identify and resolve missing or inconsistent data.
- **3.4.3 Technologies Used**
- **Frontend**: ReactJS [1], Material-UI [6], SCSS
- **Backend**: Apollo Server, GraphQL [2]
- **Database**: PostgreSQL [3]
- **Other Tools**: Git and GitHub for version control, JIRA for ticket management and bug reporting slack for team communication.

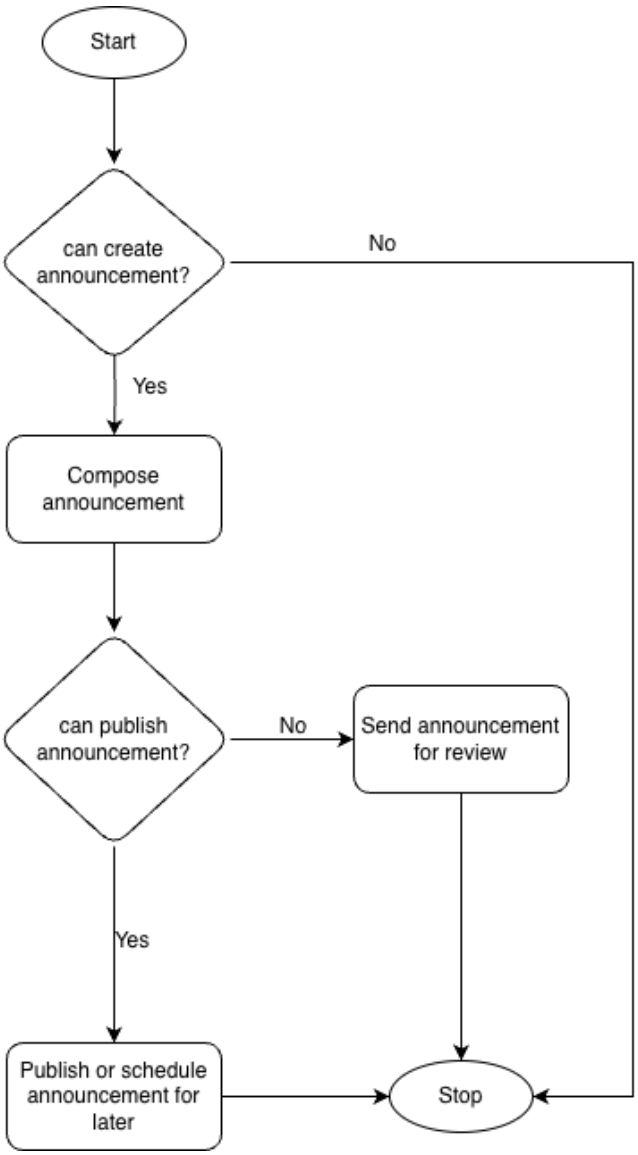


Fig 3.1 High-Level Overview of Announcement Creation

This flowchart illustrates the end-to-end process of how a teacher can make an announcement. First, it checks for the user's permission if they can create an announcement, this is done by checking the teacher's access to the mapped classes. After this, publishing permissions are checked, if the teacher has permission to publish, then the announcement is published directly to the recipients else it is first sent to the assigned reviewer and after reviewing the announcement, it gets published.

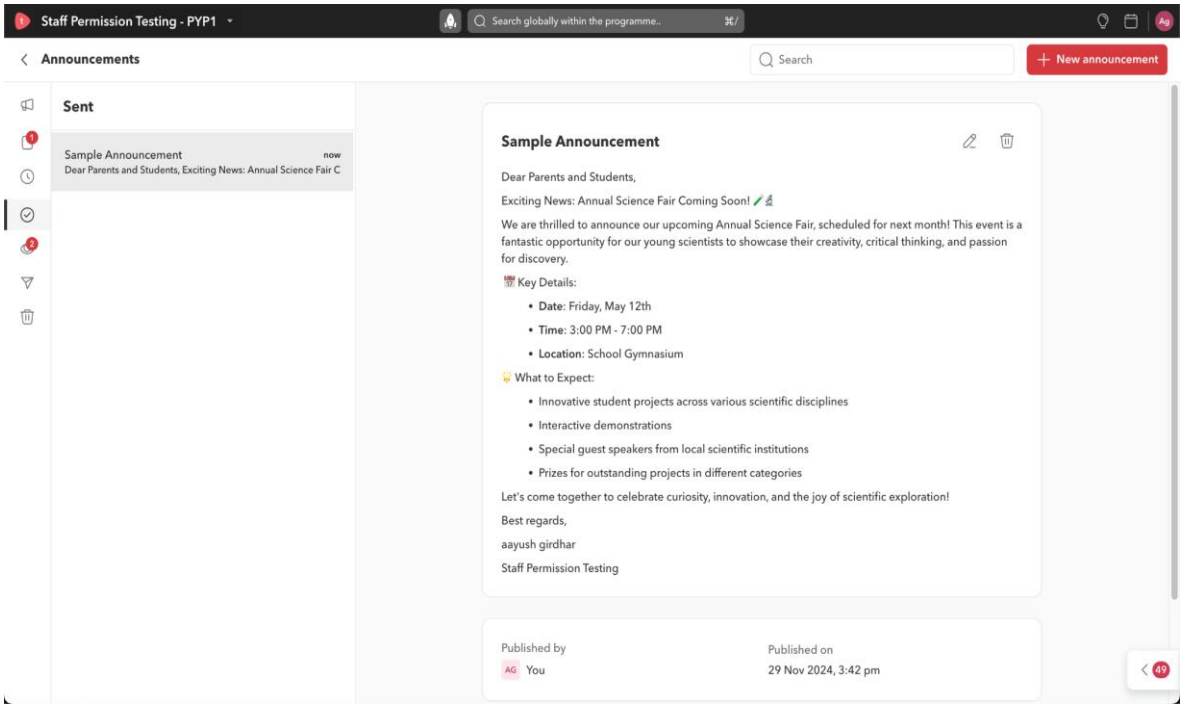


Fig 3.2: Announcement Module with a sample announcement

This figure displays the user interface of the announcement module, showcasing a sample announcement as seen by the staff member. The sent announcement can be edited or moved to the bin by the options present on the top left of the announcement body.

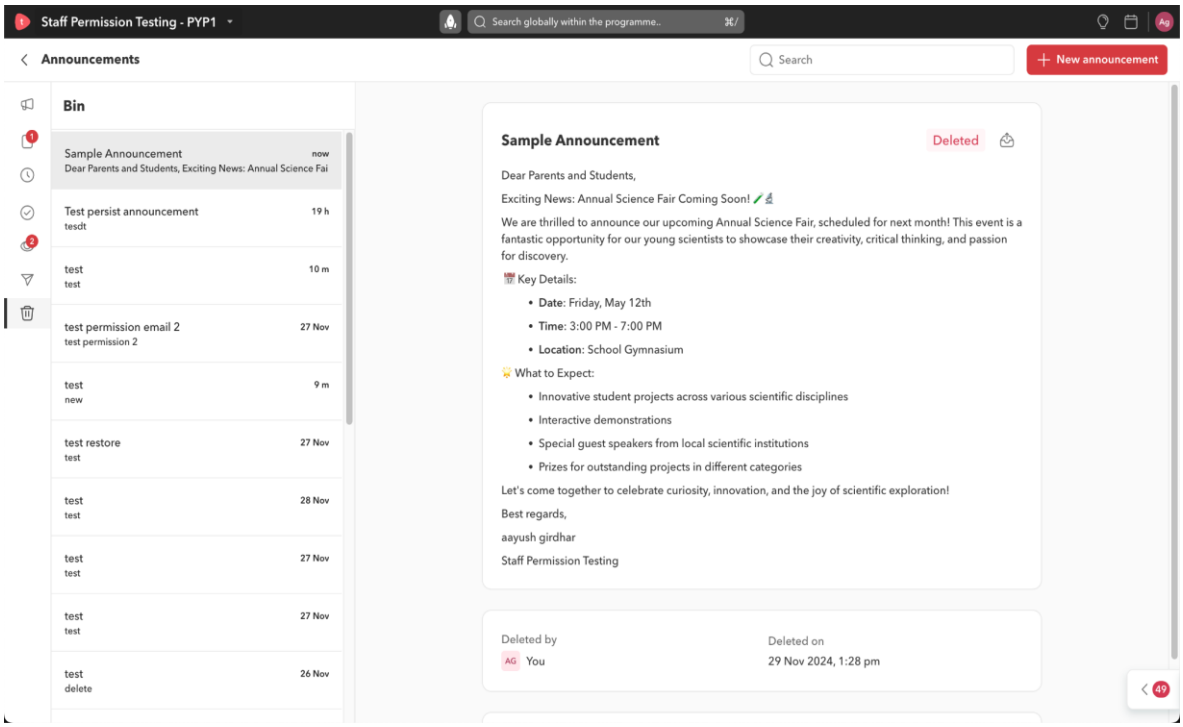


Fig 3.3: Announcement Bin and Restore Functionality

This figure shows the UI for the Bin and Restore feature, all the announcements that are deleted by the staff member are shown here, they can be restored using the restore button present on the top left side of the announcement body.

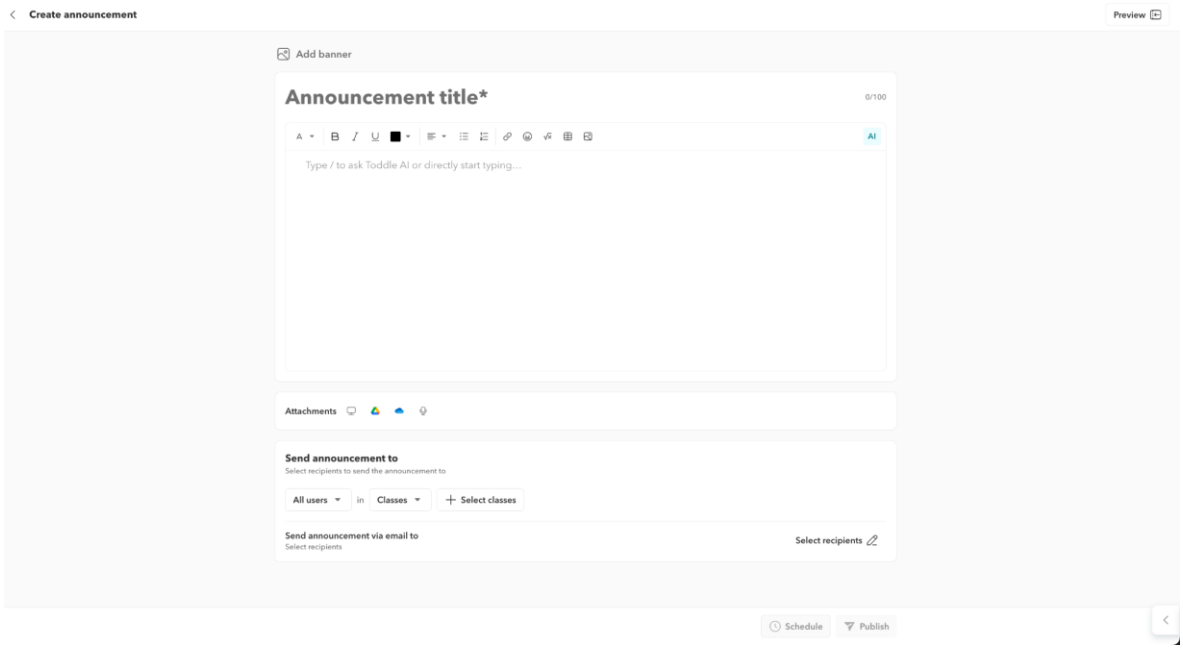


Fig 3.4: Announcement Compose Screen

This figure showcases the UI of the compose screen, the staff member can add banner image, title, react-text content, leveraging formatting tools and AI assistance. The interface also allows the staff member to add relevant attachments, be it images, docs, pdfs, videos or even voice recordings. Lastly the user can specify the recipients and whom to send the announcement via email.

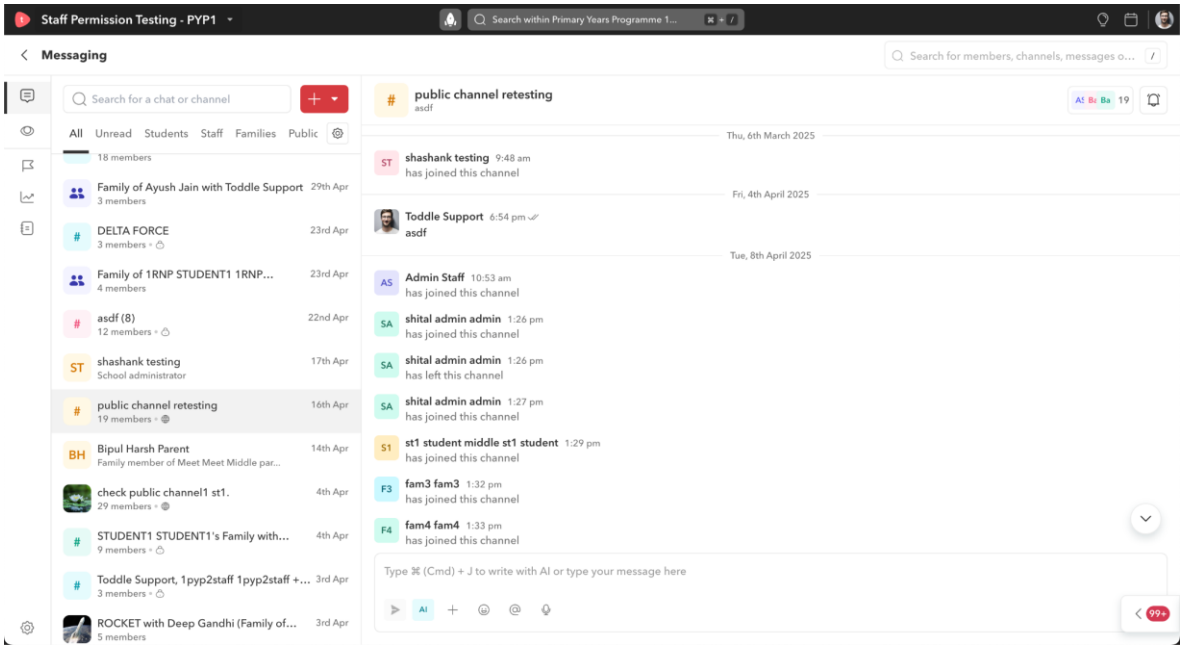


Fig 3.5: Messaging Module with a sample channel

This figure displays the user interface of the messaging module, showcasing a sample messaging thread as seen by the staff member.

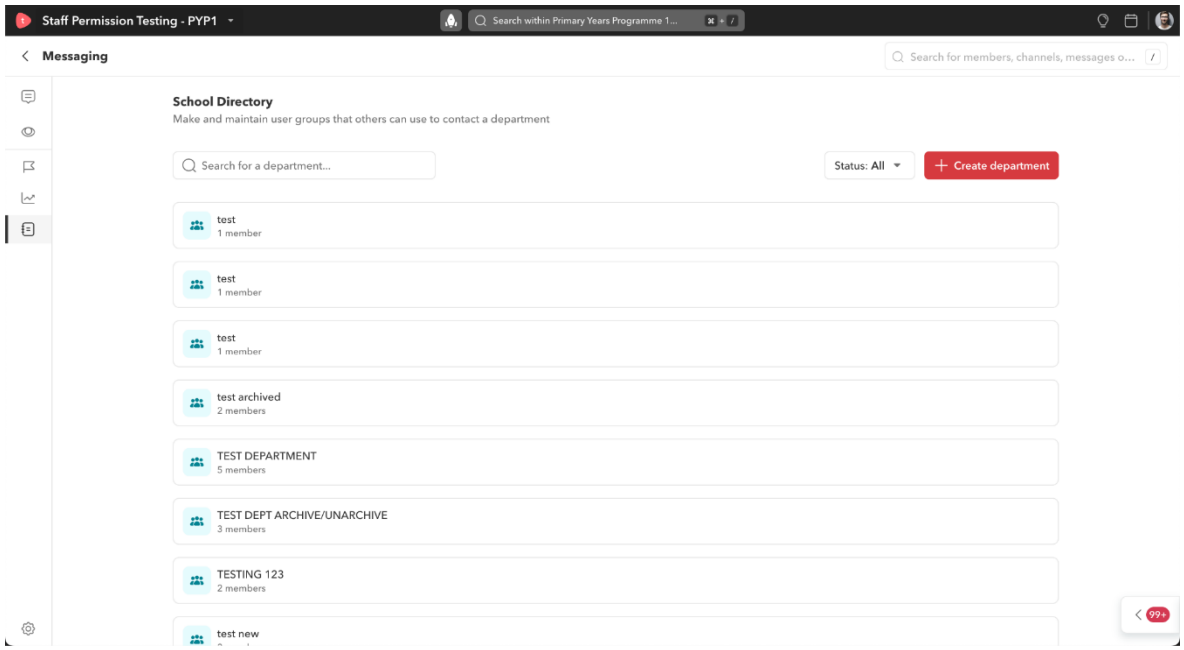


Fig 3.6: School Directory Submodule

This figure illustrates the Staff Directory submodule, the list shows the created directories. The user can search through the numerous directories using the search input field, they can also filter them according to their status like active or archived.

3.5 Key Challenges

In the course of the platform's development, a number of technical and user-centric challenges emerged that would require novel and scalable solutions. From achieving real-time synchronizations of messages and announcements to facilitating rapid, effective filters in a high-volume data situation, every problem required careful architectural choices. Besides the backend complexity of the technology, seamless and inclusive user experience for different user roles aimed at (teachers, parents and students) was also central. Furthermore, ensuring that the standards of accessibility standards were met without locating at the expense of aesthetics added another dimension of complications. This part reflects the most critical issues faced during implementation and the ways of dealing with them, which significantly supports the development of a strong, reactive and tourist-friendly communication system.

3.5.1 Real-Time Synchronization

- **Challenge**: One of the major hurdles to the development of the platform was the issue of guaranteeing that all announcements and messages were current for various users' devices in real time. This necessitated removing delays, preventing data conflict, and consistency in simultaneous edits by several users did not override the other.
- **Solution**: To overcome this, a GraphQL based synchronization system was implemented. This enabled full-duplex communication between the client and the server; hence, updates were immediately broadcasted to all connected users. Also, autosave was implemented periodically to minimise the possibility of data loss in cases of unforeseen interruptions, such as browser or a network crash. Versioning and timestamp-based control methods that are used in conflict-resolutions were also used to promote data integrity.

3.5.2 Efficient Filtering

- **Challenge**: The need for filtering fast and accurately increased as there were more announcements and messages across multiple classes and user roles. Maintaining this

became a performance bottleneck. Fetching and presenting targeted results especially in large datasets required a lot of time and resources.

- **Solution**: For performance optimization, filtering with queries were made at the database levels with indexed queries. Class, date, category and sender fields were indexed to drastically lower query response time. On the frontend, debounce mechanisms were implemented to prevent the server being overwhelmed by the same repetitive request for filters while the user is inputting. Pagination and lazy loading techniques were implemented in order to increase responsiveness and decrease initial load times.

3.5.3 User Accessibility

- **Challenge**: Considering the variety of users, teachers, parents, and students with differing degrees of technical expertise it was important to create an intuitive and user-friendly interface for them all.
- **Solution**: There were extensive usability testing sessions with the stakeholders in the form of all user roles. These sessions were conducted on task completion time, user satisfaction and error rate. The collected feedback was used to iteratively improve UI/UX. Important smooth user experience contributions such as guided walkthroughs, contextual tooltips, and homogeneous navigation patterns can be identified in order to enhance ease of use and lower the learning curve.

3.5.4 Accessibility Compliance

- **Challenge**: To make the application WCAG 2.1 compliant, while preserving the visual look-and-feel and minimally effective content, was a dual challenge. Accessories of accessibility nature quite often call for additional markup; and sometimes that markup can interfere with aesthetic decisions or dynamic content rendering.
- **Solution**: Development team took a design first, accessibility by default approach. Keyboard navigation and screen readers were purpose-designed from the components. ARIA (Accessible Rich Internet Applications) attributes had been added to interactive elements and wherever possible semantic HTML tags were used.

Regular audits for accessibility were done using Ax and Lighthouse tools and also with manual testing through screen readers like NVDA and Voiceover ensuring full compliance.

Chapter 4

Testing

Efficient testing is a bedrock of the software development life cycle ensuring that the developed system works reliably, securely, and consistent with the idea behind its development. Testing phase in this project was planned and conducted with a lot of focus on quality and assurance, user satisfaction and system robustness. The main aim was to ensure that the Announcement and Messaging System for Schools passed all functional and nonfunctional requirements, provided smooth user experience and ran with no critical defects.

To accomplish this a structured testing workflow was put in place and included a separate QA team. The QA engineers collaborated with developers to develop and run test cases, discover bugs and ensure that the bugs are solved in time. A combination of manual and automated testing methods were used to cover a wide range of usage scenarios.

4.1 Testing Strategy

The process of testing the Announcement and Messaging System followed a highly structured procedure, and was integrated in a number of deployment environments to maximize system reliability and performance. The environments were Staging, Pre-Production, and Production, and changes could only move incrementally to these layers once a comprehensive verification procedure was passed. This multi layered strategy helps pass all features through intense quality tests before release to end users.

One of the greatest merits of this approach was the division of responsibilities and assurance between environments:

- Staging acted as a first playground for combined changes.
- Pre-Production was applied to more stable and user-near simulations.
- Production received the ultimate, refined features to be provided to users.

This strategy was used to reduce the possibility of introducing regressions or incomplete features into live surfaces.

4.1.1 Quality Assurance (QA) Engineer Testing

- **Role of QA Engineers:** QA engineers who were committed were instrumental to this validation exercise. They had the duty of subjecting every part of the system to an orderly testing process ranging from:
    - Announcement tagging and categorization
    - Real-time event broadcasting
    - Role-based feature access
    - Messaging capabilities including threaded conversations and role based permissions.
    - User interface consistency and performance
- **Testing Methodology**
    - **QA engineers used a hybrid testing model**: a combination of automated and manual testing (while automated testing proved efficient in quickly verifying business logic and API behaviours, manual testing helped to test visual, UX issues that automation failed to pick up). One rather divisive improvement in this project was the ability to incorporate data-testid attributes into key UI elements. This greatly simplified test automation as QA was now able to hook up and mimic user interactions (clicks, inputs and navigation) with greater accuracy and stability.
    - Test IDs for each of the major components or user facing features whether it is a button, a form or a modal were systematically created to make sure repeated element selection during the testing is consistent eliminating flak test cases and improving maintainability of the test suite.
- **Ticketing and Workflow**
    - **When bugs were identified**:
        - QA engineers would report the issue as Jira tickets by classifying it by priority, platform (frontend/backend), and severity.

- Screenshots and links to related logs or Sentry traces were attached to the ticket.
- Developers would then fix the bug, commit the fix locally and raise a pull request.
- Fixes were checked by Managers before merging into Staging which exacerbated another set of validation before advancing to Pre-Prod and finally Production.

4.1.2 Live bug reporting via Sentry

In order to catch unexpected behaviours in real time, Sentry was implemented into all 3 environments. It was the foundation for live error tracking and performance tracking which logged automatically:

- JavaScript or backend exceptions
- Faulty API calls
- Incorrect or missing parameters
- Broken data flows
- Performance bottlenecks
- User context and stack traces
- Each error in Sentry is automatically associated with the relevant code owner/team using defined ownership rules. Once an issue was detected:
- The assigned team would investigate the root cause, replicate the issue locally, and apply the necessary fixes.
- Fixes would be validated in Staging and then promoted downstream following the test release cycle.

Sentry greatly reduced the mean time to detect and resolve bugs and helped developers and QA remain proactive rather than reactive.

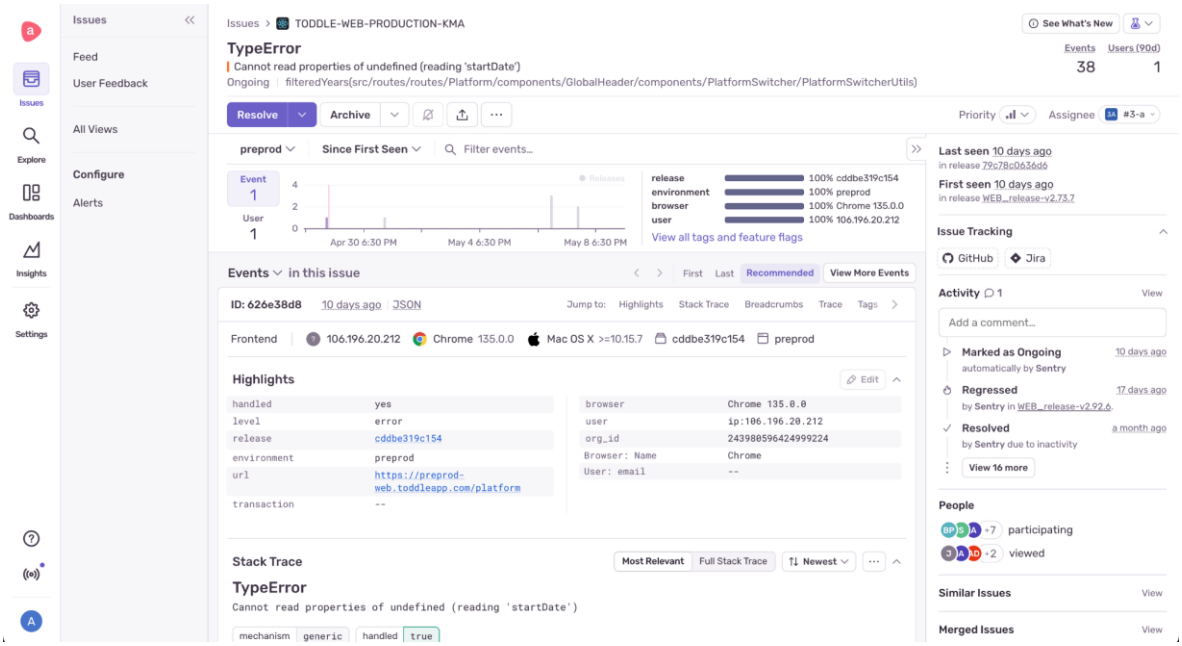


Fig 4.1: Sample Sentry bug report

4.1.3 Task Management on Jira

Jira acted as the epicentre of controlling all quality and engineering – processes. Every bug detected in Sentry or identified in manual testing was represented as a Jira issue. The flow was as follows:

- **QA provided a full bug ticket as follows**:
    - A clear title and description
    - Priority and severity
    - Reproduction steps
    - Screenshots or screen recordings
    - Sentry logs (if applicable)
- Instead of the managers triaging and routing tickets to the developers based on domain (frontend/backend) expertise, they did it. Developers solved the problems in the specified sprints or hot fixing cycles. With peer reviews and local testing, the fix was rolled out to Staging for QA verification.
- When passing Staging checks changes were being moved into Pre-Production, and then to Production. The integration of test ID usage, Sentry integration [8] and Jira

based task assignment created a robust ecosystem where bugs were not only captured and fixed but prevented to recur.

4.1.5 Testing Tools

An array of sophisticated modern-day tools were incorporated into the process of the testing workflow to verify the system on multiple levels of the logic up to the user interaction and communicating with the API. Every tool had a definite role to play in the multi-tiered testing endeavour, and when put together, they added to the stability, performance, and correctness of the platform.

4.1.5.1 Jest – Unit Testing Framework

Purpose:

Jest has been mostly used for unit testing the JavaScript and TypeScript codebases, particularly in frontend.

Usage:

- Focused on component/function tests in isolation.
- Maintained the internal logic of UI components like a button, modals, input validations, and hooks.
- Use of testId attributes in integration enabled better and scoped querying in DOM based tests.

Benefits:

- Fast execution with an integrated mocking possibility.
- Works perfectly with React, supporting snapshot testing and state check.
- The way out to regressions was to ensure that every small logic unit worked as changes were being introduced.

4.1.5.2 Cypress – End-to-End Testing

Purpose:

Cypress was used to get end-to-end (E2E) - user journey simulation and confirmation the system Hypothesis performs as expected across all modules.

Usage:

- Real-life user flows, which were covered, included the following:
- Creating, sending, and reviewing announcements
- Performing search and filter operations
- Navigating through messaging threads
- Working with validations from form and permissions
- Used testId attributes to the maximum extent possible in element selection for a stable interface, shunning the use of weak DOM hierarchies.

Benefits:

- Real browser testing that gave high fidelity simulation of how the application is used by real users.
- Ease of debugging was made possible on the automatic screenshots and video capture for every run of the test.
- Previously, Continuous integration (CI) was used to execute a full regression cycle before every deployment cycle.

4.1.5.3 Postman – API Testing

Purpose:

Postman was applied to test backend APIs such as, announcements, messaging, user authentication, role permissions, search functionalities.

Usage:

- Manually tested APIs in the development phase in order to guarantee:

- Correct request and response structures
- Aspects that should be properly handled are codes and messages of errors.
- Access control and response behavior based on roles
- Collections were distributed between team members for the sake of consistency and traceability.
- Part of CI pipelines to automate API regression testing with Newman (Postman's CLI companion).

Benefits:

- Externally assisted early validation of APIs, even before the procedure of frontend integration.
- Within the project, simplified the debugging of backend issues by giving immediate access to the logs, headers, and payloads.
- Allowed for the creation of test environments and variables, streamlining testing across dev/staging/prod.

4.2 Test Cases and Outcomes

The comprehensive testing was performed to verify functional reliability, user experience, and robustness of both Announcements and Messaging modules. The test process included manual and automated test cases covering such key scenarios such as data validation, in the real time communication, filtering, searching, permission checks and recovery mechanism. For each test case, an attempt to represent it like a usage of the system in real life or an edge case was made in order for the system to behave properly in different scenarios. The publication, scheduling, deletion and restoration and filtering of announcements was checked on the Announcement module. Likewise, the Messaging module received extensive testing for real-time jabbering and attachment processing, and message state and access controls. Every test case operated with success proving the stability and reliability of the system for smooth communication around the platform.

4.2.1 Announcements

Table 4.1: Announcement: Test cases and their outcomes

Test Case ID	**Description**	**Expected Outcome**	**Result**
TC-A01	Create and publish an announcement with all mandatory fields filled	The announcement is saved, published, and visible to selected recipients	Pass
TC-A01A	Create and schedule an announcement	The announcement is saved and scheduled for the correct future date and time	Pass
TC-A02	Attempt to publish an announcement without a description	A toast message is displayed: “Description is required”; form submission is blocked	Pass
TC-A03	Refresh the page while creating an announcement	The draft is autosaved, and the previously entered data is restored upon refresh	Pass
TC-A04	Soft delete an announcement	The announcement is removed from recipient views and moved to the creator's “Deleted” category	Pass

TC-A04A	Permanently delete a draft announcement	The announcement is fully removed from the system and no longer visible to any user	Pass
TC-A05	Restore a deleted announcement	The announcement is moved back to its previous state and becomes visible to all intended recipients	Pass
TC-A06	Search announcements using a text keyword	Announcements containing the search term are listed; term is highlighted in the announcement card	Pass
TC-A07	Filter announcements by creator, publish date, and recipients	Filtered announcements are displayed accurately based on selected parameters	Pass

4.2.2 Messaging

Table 4.2: Messaging: test cases and their outcomes

Test Case ID	**Description**	**Expected Outcome**	**Result**
TC-M01	Send a message in an existing conversation thread	The message appears instantly in the thread and is marked as delivered	Pass
TC-M02	Send a message to a new recipient	A new conversation thread is created, and the message is delivered to the intended user	Pass
TC-M03	Attempt to send a message with empty text	An error toast is shown: “Message content cannot be empty.”	Pass
TC-M04	Send a message with attachments (image/PDF/doc)	Attachment uploads successfully, is displayed as a preview or downloadable link in the conversation	Pass
TC-M05	Refresh the messaging interface during typing	Draft message content is preserved due to autosave functionality	Pass

TC-M06	Receive a real-time message from another user	The incoming message appears instantly in the conversation view without requiring refresh	Pass
TC-M07	Search for a message within a thread using keywords	Relevant messages are highlighted and listed chronologically	Pass
TC-M08	Filter conversation threads by class or participant	Only threads involving the selected class or user are displayed	Pass
TC-M09	Delete a message from the sender's end	The message is marked as "Deleted for You" but remains visible to the recipient	Pass
TC-M10	Delete a message for everyone (within the allowed time window)	The message is replaced with a “This message was deleted” note in both sender and receiver's threads	Pass
TC-M11	Attempt to send a message as a user without permission	Access is blocked, and an error toast is shown: “You do not have permission to send messages in this group”	Pass
TC-M12	Use test IDs in automated testing to interact with message components	All user-interactable components respond correctly using data-testid selectors in Cypress or Jest-based scripts	Pass

Chapter 5

Results and Evaluation

This chapter is a comprehensive summary in one coherence of all the results the Communication System elicited through the delicate achievements of the devoted team. Finally, this chapter will spend some time evaluating the overall functionality of the system and also determine some major findings that appeared in the wake of user input as well as meticulous testing procedures.

5.1 Results

The Communication System for Schools, which comprised the Announcements and Messaging modules respectively, successfully achieved its goal of providing structured, secure, and real-time communication between teachers, children and parents. By incorporating intelligent workflows and automation into every day tasks the system has markedly decreased errors in previously manual and error-prone processes. Among other features, real-time synchronization, role-based access, message threading and announcement reviews have simplified communication, minimized administrative overhead, and promoted collaboration.

Educators and administrative staff feedback had high levels of satisfaction, which included ease of use, reliability, and coordination improvement. The architecture of the system is scalable with integrity of data, even at high concurrent usage thus making it a reliable instrument in digital transformation of educational communication.

5.1.1 System Performance

A strong communication platform needs to be resilient in diverse load conditions and provide ubiquitous fast user experience. System performance was checked with reference to performance indicators such as the response time, concurrency to handle, latency, and state

consistency. Through extensive testing in simulated natural use environments both Announcements as well as Messaging performed quite well within acceptable thresholds. With proper utilization of autosave approaches, peak use lagged minimally on responsiveness terms while the system is stable. These optimisations guaranteed smooth flow of real-time interaction across devices and networks.

5.1.1.1 Announcements

- **Response Time**: Mean system response for creating, publishing and retrieving announcements was kept below 350 ms even during moderate to high user load.
- **Concurrent Usage**: System integrity was maintained at the same time supporting hundreds of users performing multiple operations without performance degradation.
- **Data Consistency**: Automated validation and recovery logic had made state management uniform across edge cases like abrupt re-loads, or disconnections.

5.1.1.2 Messaging

- **Low-Latency Delivery**: Messages were always sent with low latency (~120-180 ms average), so that interaction was real time.
- **Scalable Thread Handling**: Individual and group chat threads were scalable, within delivery order and consistency even with concurrent inflow of messages.
- **WebSocket Stability**: Kept stable, bi-directional communication via WebSocket connections, in different environments.

5.1.2 Functionality Validation

In order to verify that every module operated as it was supposed to in a wide variety of situations and user categories, functionality of each one was tested intensely. The correctness of state transitions, ability to respond to user actions (optimistic updates) and full accessibility support were checked for the Announcements module. Similarly, the Messaging module's threaded conversation support, delivery receipts, and role based access policies were thoroughly validated. The edge case test was carried out on all workflows to ensure the

assurance of the resilience of all workflows thus upholding functional integrity at diverse operational conditions. Such a level of verification helped achieve a stable and reliable experience for users in general.

5.1.2.1 Announcements

- **Optimistic Updates**: Confirmed that the UI behaved as expected when all changes (draft, edit, delete) appear immediately to give a perceived response speed and usability.
- **State Management**: Tested transitions between Draft, Review, Published and Archived with 100% accuracy in all environments.
- **Accessibility Compliance**: Complex WCAG 2.1 compliance was achieved, including appropriate keyboard navigation and the compatibility with screen readers. [4]

5.1.2.2 Messaging

- **Threaded Conversations**: Functionality for viewing, replying to, and traversing threaded conversations was confirmed under multiple edge cases.
- **Role-based Restrictions**: Verified sender-receiver access control according to user roles (students cannot initiate messages to other students etc.).
- **Delivery Receipts**: Message delivery and read indicators all worked well across all the supported platforms.

5.1.3 User Feedback

Involvement of direct input from stakeholders in all user groups was important for investigating the system's practicality. Feedback from Announcement users praised its easy navigation, intuitive workflows, and clarity provided by categorization of the content views. In Messaging, customers enjoyed real-time interaction, announcement integration, and a privacy-conscious design. Teachers and administrators, especially, noted that the interface matched closely with their daily activities. The feedback loop on this level aided in refining

user journeys, as well as, it validated that our design decisions were not just theoretical best practices, but were aligned to user needs.

5.1.3.1 Announcements

- **Ease of Use**: However, educators liked the intuitive layout, particularly such aspects as bin/restore, search filters, and announcement viewing preview.
- **Review Workflow**: The "Sent for Review" flow brought the governance level, which has been well adopted in schools with hierarchical approval processes.
- **Content Categorization**: Such classified views as "Scheduled," "Published" and "Deleted" made things clear and under control.

5.1.3.2 Messaging

- **Instant Engagement**: The speed of communication was appreciated by teachers and parents, especially in time critical communications.
- **Unified Experience**: The fact of the integration of messaging with announcements in one interface was appreciated for the lack of third-party tools.
- **Privacy Assurance**: Feedback reported users felt more secure that messages could only be seen by the recipient.

5.1.4 Usability Metrics

- **Ease of Use**: The average score of the system's usability was given 4.7/5 after surveys on user feedback.
- **Compliance with Accessibility**: WCAG 2.1 requirements were fulfilled at all points, ensuring inclusion. [4]

5.2 Comparison with Existing Solutions

The following table summarizes a comparison between the traditional techniques, the current available digital platforms and the Proposed Communication System. It features the system's innovations and effectiveness in correcting the deficiencies of the available solutions.

Table 5.1: Comparison with existing solutions

Feature	**Traditional Methods**	**Existing Digital Solutions**	**Proposed System**
Real-Time Syncing	Absent; announcements and messages are static and often delivered with delays.	Limited support; often reliant on manual refresh or polling.	Fully integrated real-time syncing for both modules for zero data loss.
Class-Specific Filtering	Requires manual sorting and physical grouping, prone to error.	Basic filtering available, typically limited to predefined groups.	Dynamic filters by class, role, and custom metadata tags for precise targeting and message delivery.
Bin and Restore Functionality	Not available; once discarded, messages/announcements are lost permanently.	Rarely implemented; typically lacks intuitive recovery flow.	Soft delete with dedicated "Bin" and one-click restore for both announcements and messages.

Announcement Review Workflow	Informal processes; no clear review or approval structure.	Minimal to no support; content is often posted directly.	Robust review system allowing draft submission by teachers and publishing by approvers to maintain content quality.
Threaded Messaging	Not applicable; communication is linear and fragmented (e.g., paper notes, calls).	Limited or absent in most school platforms.	Fully threaded conversations for organized discussions, supporting 1:1 and group messaging.

Chapter 6

Conclusions and Future Scope

6.1 Conclusion

Traditional school communication methods have long suffered from inefficiencies, delays, and a lack of coordination. The Communication System for Schools, integrating both Announcements and Messaging, effectively addresses these limitations by harnessing modern web technologies and real-time data handling. Through this project, the reliability, accessibility, and timeliness of internal communication within educational institutions have been significantly enhanced.

By combining thoughtful UX design, robust architecture, and real-time capabilities, the system offers a structured yet flexible communication platform. Noteworthy improvements brought about by this project include

6.1.1 Improved Coordination and Accountability

The Announcement Review System ensures that only approved and quality-checked content reaches the end recipients. Teachers, staff, and administrators can now collaborate more transparently, with designated reviewers overseeing the publication of sensitive or high-priority communications. Similarly, the role-based Messaging System ensures secure, guided conversations, preventing communication misuse while encouraging meaningful dialogue.

6.1.2 Streamlined Workflows

State-driven logic simplifies the lifecycle of announcements drafting, publishing, archiving, and restoring thus reducing ambiguity and administrative overhead. The threaded messaging experience further aids clarity by organizing discussions in a conversational format rather than scattered threads.

The ineffectiveness, delays, and imbalance of coordination have always plagued school communications through the ancient means of doing things. The Communication System for Schools, with Announcements and Messaging all in one, provides a powerful solution to these limitations because of its utilization of the latest in web technologies and real-time data handling. Through this project the reliability, accessibility and timeliness of the internal communication of the educational institutions was improved.

6.1.4 Scalability and High Performance

The supporting infrastructure allows quick response time (below 350ms) and data consistency and integrity even with parallel users and large data sets. The system is easily scalable throughout schools of different sizes and requirements, and thus a future proof solution.

6.2 Future Scope

Though the present system has been able to meet its basic goals and provide a usable communication platform for schools, there is an array of strategic improvements that could take its functionality, flexibility and integration into larger educational ecosystems to the next level.

6.2.1 Roles and Permissions Refinement

Existing Limitation:

Today's system uses some predefined roles with a limited granularity of permissions. This restriction impacts its ability to accommodate throughputs of subtle hierarchical structures especially in larger or more specialized learning institutions.

Proposed Enhancements:

- Schedule fine-grained Role Based access control (RBAC) where differentiation of permissions across non-teachers, counsellors, administrators, educators is possible.

- Make it possible to create customized roles, so institutions would be able to create roles consistent with their internal structures, and policies.
- Assign super-administrator roles in order to govern system-wide activities, audit trails, and policy enforcement.
- Expand permission rules to Messaging i.e. restrict who can start conversations with students or create read only for counsellors.
- **Benefits**:
- Provides users with a data security that only allows access to sensitive information according to roles.
- Streamlines method of task delegation and minimizes operational bottlenecks.
- Supports regulatory compliance: Improved traceability and control are provided.

6.2.2 Integration with Third-Party Platforms

Existing Limitation:

At present, the system is positioned as an independent tool of communication and does not have a smooth interaction with other digital instruments applied in educational institutions.

Proposed Enhancements:

- Create integrations with Learning Management Systems (LMS) like Google Classroom, Canvas and your own learning platform, to enable automatic sharing of announcements and messages with class activities and homework.
- Connect to School Management Systems (SMS) so as to feed and pull attendance, academics, and behaviour reports to us.
- Introduce RESTful APIs and webhook support in allowing real time data sharing across custom school infrastructures.
- Connect Messaging to services such as Zoom or Meet in order to start making instant video calls as required.

Benefits:

- Empowers a single dashboard unified by announcements, messaging, academic performance and conduct monitoring.
- Promotes the adoption of digital by providing plug-and-play experience with existing platforms.
- Combines multiple data streams to support teachers' and school leaders' data-informed decision making.

Such improvements are meant primarily to future-proof the system not only increasing its applicability on a wider range of educational settings but taking it as a cornerstone of school digital transformation. The Communication System for Schools can develop into a full-fledged collaborative ecosystem, thanks to such modular enhancements and greater integrations.

6.3 Summary

The Communication System for Schools including the Announcement and Messaging modules is an important step in the modernization of institutional communication. With an elegant combination of real-time synchronization, class specific targeting, intelligent drafting of messages and an innovative Announcement Review System, the platform directly fills the glaring lacuna of clarity, velocity and accountability in educational environments.

With a modern, scalable stack (ReactJs [1], GraphQL [2], and PostgreSQL [1]) the system provides a friendly user interface and a responsive one to be used regardless of the devices. Some interesting features include soft-delete bins, keyboard accessibility compliance, and AI assisted content generation that make a system useful to teachers and administrators, while allowing for inclusivity and ease of use. Messaging module, introduced as a supplementary channel, encourages direct and controlled interactions between stakeholders allowing issue escalation, feedback loops, and parent-teacher interaction beyond the static announcements.

In its turn, the system’s scalability, ability to operate reliably under concurrent use, and intuitive design were confirmed through rigorous performance testing and the opinion of users. High satisfaction was validated via usability metrics, compliance to WCAG 2.1 standards ratified through accessibility evaluation.[4] Comparison to both conventional and currently available digital solutions is obvious with regard to the platform’s great agility, control and integration potential.

Future perspective, the system provides the framework for future improvements including, granular role-based access control, API integrations with LMS and SMS platforms and data driven personalization of communication workflows. Such developments will allow institutions to develop toward an intelligent communication ecosystem that adjusts to organizational needs dynamically.

Finally, this project addresses existing communications problems in schools and sets a strong foundation for ongoing transformation in schools. By creating an open, inclusive, and collaborative work environment amongst educators, students and their parents, it establishes new standards in terms of operational excellence and active community involvement in education.

Chapter 7

References

1. ReactJS Documentation, "*React – A JavaScript library for building user interfaces*" 2024. [Online]. Available: https://react.dev/. [Accessed: Nov. 30, 2024].

2. Apollo GraphQL, "*Apollo Client Documentation*" 2024. [Online]. Available: https://www.apollographql.com/docs/react/. [Accessed: Nov. 30, 2024].

3. PostgreSQL Documentation, "*PostgreSQL: The World's Most Advanced Open-Source Relational Database*" 2024. [Online]. Available: https://www.postgresql.org/. [Accessed: Nov. 30, 2024].

4. Web Content Accessibility Guidelines (WCAG), "*Web Content Accessibility Guidelines 2.1*" W3C, 2024. [Online]. Available: https://www.w3.org/WAI/standards-guidelines/wcag/.[Accessed: Nov. 30, 2024].

5. Firebase Documentation, "Firebase: Real-time Database and Backend-as-a-Service," 2024. [Online]. Available: https://firebase.google.com/docs/. [Accessed: Nov. 30, 2024].

6. Material-UI Documentation, "*MUI: The React component library you always wanted*" 2024. [Online]. Available: https://mui.com/. [Accessed: Nov. 30, 2024].

7. Toddle Internal Documentation, "I*nternal guidelines and system documentation*," Proprietary Documentation, 2024. (Note: Non-disclosable proprietary documentation used for internal system design and implementation guidance.)

8. Sentry Documentation, "*Sentry: Application Monitoring and Error Tracking Software*," 2024. [Online]. Available: https://sentry.io/. [Accessed: Nov. 30, 2024].

9. Jira Documentation, "*Jira Software by Atlassian*," 2024. [Online]. Available: https://www.atlassian.com/software/jira/. [Accessed: Nov. 30, 2024]. 37

10. AWS Documentation, "*AWS Cloud Computing White Papers*," Amazon Web Services, 2024. [Online]. Available: https://aws.amazon.com/whitepapers/. [Accessed: Nov. 30, 2024].

11. Slack API Documentation, "*Slack API: Build your integration*," 2024. [Online]. Available: https://api.slack.com/. [Accessed: Nov. 30, 2024].

12. Minimalist Design Blog, "*Creating Intuitive and User-Friendly UI/UX Designs*," 2024. [Online]. Available: https://minimalist.design/. [Accessed: Nov. 30, 2024].

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING AND INFORMATION TECHNOLOGY

PLAGIARISM VERIFICATION REPORT

Date: May, 2025**.**

Type of Document: B.Tech. (CSE / IT) Major Project Report

Name: _____ **Enrollment No.:** _____

Contact No: _____ **E-mail:** _____

Name of the Supervisor (s): _____

Title of the Project Report (in capital letters): _____

_____

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/regulations, if I found guilty of any plagiarism and copyright violations in the above major project report even after award of degree, the University reserves the rights to withdraw/revoke my major project report. Kindly allow me to avail plagiarism verification report for the document mentioned above.

- Total No. of Pages:
- Total No. of Preliminary Pages:
- Total No. of Pages including Bibliography/References:

Signature of Student

FOR DEPARTMENT USE

We have checked the major project report as per norms and found **Similarity Index** .....%. Therefore, we are forwarding the complete major project report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

Signature of Supervisor **Signature of HOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

<table>
<tr><th>Copy Received On</th><th>Excluded</th><th>Similarity Index (%)</th><th colspan="2">Abstract & Chapters Details</th></tr>
<tr><td rowspan="2"></td><td rowspan="4">• All Preliminary Pages
• Bibliography/ Images/Quotes
• 14 Words String</td><td rowspan="2"></td><td>Word Count</td><td></td></tr>
<tr><td>Character Count</td><td></td></tr>
<tr><td>Report Generated On</td><td>Submission ID</td><td>Page Count</td><td></td></tr>
<tr><td></td><td></td><td>File Size (in MB)</td><td></td></tr>
</table>

Checked by

Name & Signature **Librarian**

*% detected as AI

AI detection includes the possibility of false positives. Although some text in this submission is likely AI generated, scores below the 20% threshold are not surfaced because they have a higher likelihood of false positives.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?

The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?

Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.

G123.pdf

ORIGINALITY REPORT

6%	5%	1%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Jaypee University of Information Technology Student Paper	4%
2	ir.juit.ac.in:8080 Internet Source	1%
3	www.ir.juit.ac.in:8080 Internet Source	<1%
4	www.juit.ac.in Internet Source	<1%
5	Submitted to University of Hertfordshire Student Paper	<1%
6	opus4.kobv.de Internet Source	<1%
7	Submitted to University of Wales Institute, Cardiff Student Paper	<1%
8	upcommons.upc.edu Internet Source	<1%
	tudr.thapar.edu:8080	

9	Internet Source	<1%
10	gupea.ub.gu.se Internet Source	<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		