

Digital Notice Management System (DNMS)

Ankit Raj¹, Mrs. K. Krishna Veni², Chintu Sasmal³

^{1,3}*III Sem Student, Department of Computer Science and Applications, S.E.A. College of Science, Commerce and Arts, Bangalore*

²*Assistant Professor, Department of Computer Science and Applications, S.E.A. College of Science, Commerce and Arts, Bangalore*

Abstract—Academic institutions face challenges in managing notices, announcements, and communication between faculty and students. Traditional methods such as physical notice boards or generic emails often lead to inefficiency, missed information, and lack of personalization. This research proposes a Digital Notice Management System (DNMS), a web-based platform enhanced with modern backend and frontend technologies. The system integrates Spring Boot, Hibernate, MySQL, and React to provide secure notice creation, role-based access, file attachments, and automated email notifications. The objective is to improve communication efficiency, ensure timely delivery of information, and enhance student engagement.

Index Terms—*Digital Notice Board, Spring Boot, React, Email Automation, Academic Communication, Role-Based Access.*

I. INTRODUCTION

The Digital Notice Management System is a web-based application that allows administrators to post, update, and manage notices online. Students, faculty members, and staff can easily view the notices through the system without needing to visit a physical notice board. This saves time and ensures that information reaches everyone quickly. One of the major advantages of this system is real-time notification. When a new notice is posted by the administrator, it can be instantly visible to all registered users. Some systems may also provide alerts through email or in-system notifications. This ensures that no important announcement is missed. The system also improves organization and management of notices. Notices can be categorized based on departments, events, examinations, or general announcements. Users can easily search or

filter notices based on their needs. This makes the process of finding information much easier compared to traditional notice boards. Another important feature is security and authentication. Only authorized administrators or staff members are allowed to create or modify notices. Students can only view the notices that are relevant to them. This helps maintain the authenticity and reliability of the information shared through the system. The Digital Notice Management System also supports record maintenance. All notices are stored in a database, which allows users to access previous notices whenever required. This helps institutions maintain proper documentation and history of announcements. Furthermore, the system promotes paperless communication, which reduces the use of paper and contributes to environmental sustainability. It also reduces manual work for staff members and improves overall efficiency in managing institutional communication.

II. LITERATURE REVIEW

Traditional paper-based notice boards in educational institutions have long been criticized for delayed dissemination, environmental waste, and limited accessibility. Research by Sahana and Aravindhan (2025) introduced a Digital Notice Board Management System that replaced manual posting with a web-based solution, emphasizing faster communication and reduced reliance on physical media. Similarly, Bhor and Darade (2026) proposed Campus Flow, a smart-campus ecosystem integrating IoT and AI to automate notice distribution, attendance, and service requests, highlighting the growing trend of centralized academic communication portals. Another study by Faisaluddin

et al. (2026) presented a College News Portal, designed to simplify collaborative communication by centralizing announcements, event updates, and examination schedules, thereby improving transparency and student engagement. Collectively, these works demonstrate a clear shift from manual, fragmented communication toward digital, automated, and integrated platforms. However, most existing systems either focus narrowly on departmental use or lack robust features such as email automation, role-based dashboards, and responsive UI design. The DNMS project builds upon these foundations by combining backend robustness (Spring Boot, Hibernate, MySQL) with a modern frontend (React, Bootstrap) and automated notification services (JavaMailSender), thereby addressing gaps in accessibility, scalability, and user experience. This positions DNMS as a comprehensive solution that not only streamlines notice management but also enhances reliability and inclusivity in academic communication.

III. SYSTEM ARCHITECTURE

DNMS follows a layered architecture:

3.1. Architectural Layers

1. Presentation Layer

- HTML5, CSS3, Bootstrap, React
- Responsive UI

2. API Layer

- Fast API REST endpoints
- Role-based access control

3. Analysis Engine

- Workflow Monitoring – Tracks notice creation, validation, backend processing, and email delivery.
- Error Detection – Identifies failed uploads, invalid inputs, or SMTP issues in real time.
- Performance Evaluation – Measures response times for API calls, database queries, and frontend rendering.
- User Engagement Tracking – Records dashboard visits, search/filter usage, and email open rates.
- System Reliability Insights – Provides administrators with logs and analytics to improve workflows.

4. Scoring Engine

- Success Rate Calculation – Assigns percentage scores to notice creation, email delivery, and search/filter accuracy.
- Responsiveness Scoring – Evaluates frontend rendering speed and UI responsiveness under different loads.
- Engagement Metrics – Scores student interaction levels based on dashboard activity and email responses.
- Priority Ranking – Ranks notices by urgency, relevance, or frequency of access.
- Benchmarking & Comparison – Generates scores over time to compare system versions and validate improvements.

5. Persistence Layer

- MySQL database
- Hibernate
- User and report entities

IV. METHODOLOGY

➤ Faculty Input

Crop The workflow of the Digital Notice Management System begins with faculty input, where authorized faculty members log in through a secure dashboard and enter notice details such as the title, type, semester, and description. They can also upload attachments like timetables, circulars, or PDFs. This stage ensures that all essential information is captured at the source.

➤ Validation And Processing

Once the data is submitted, it undergoes validation and preprocessing. The system checks whether all required fields are filled, verifies the format and size of uploaded files, and sanitizes the inputs to prevent duplication or errors. This step ensures that only clean and accurate data moves forward in the workflow.

➤ Backend Processing

The validated data is then handled by backend processing, where Spring Boot APIs manage CRUD operations, store notices in the MySQL database, and trigger email automation. Hibernate ORM ensures smooth mapping between notice entities and database

tables, while robust exception handling guarantees reliability.

➤ Database Layer

After backend processing, the notices are stored in the database layer, which maintains records of notices, attachments, user roles, and timestamps. This structured storage allows efficient retrieval and supports search and filter functionalities.

➤ Frontend Rendering

The next stage is frontend rendering, where React components dynamically fetch and display notices in interactive cards. These cards are responsive, visually engaging with gradient backgrounds and hover effects, and allow students to filter notices by semester or type. This ensures that students can quickly access relevant information in a user-friendly interface.

➤ Notification Delivery

Finally, the workflow concludes with notification delivery. Using JavaMailSender, the system automatically sends emails to all registered students, including the notice details and attachments. This guarantees that students receive timely updates both on their dashboard and in their inbox, reducing the risk of missed communication and ensuring complete coverage.

V. IMPLEMENTATION

5.1. Technology Stack

Technology	Purpose
Java	Backend
FastAPI	REST API
MySQL	Database
Hibernate	ORM

5.2. Modules

- Admin
- Faculty
- Student
- Notice
- Notification
- REST API Endpoints
- Frontend UI

VI. RESULTS AND PERFORMANCE

Model	Accuracy	Output
Notice Creation	95-100%	Correct notice stored
Email Delivery	90-95%	Timely notifications sent
Search/Filter	85-90%	Relevant notices retrieved
Responsive UI	95%	Dynamic card layout

VII. DISCUSSION

The development of the Digital Notice Management System highlights both the strengths and limitations of adopting a centralized, automated communication platform in academic institutions. The system successfully integrates backend robustness with frontend usability, ensuring that notices are created, validated, stored, and delivered efficiently. The methodology demonstrates how modular workflows—faculty input, validation, backend processing, database storage, frontend rendering, and email notification—work together to reduce administrative effort and improve communication reliability.

One of the most significant outcomes observed is the dual-channel delivery of notices, through both the student dashboard and automated email notifications. This redundancy ensures that students remain informed even if they miss one channel, thereby increasing accessibility and reliability. The responsive React interface, with features like search, filter, and interactive cards, further enhances student engagement and usability.

VIII. CONCLUSION

The Digital Notice Management System (DNMS) demonstrates how academic communication can be transformed through automation, modular design, and user-centric interfaces. By integrating Spring Boot, Hibernate, and MySQL on the backend with React and Bootstrap on the frontend, the system ensures seamless workflows for notice creation, validation, storage, and delivery. The inclusion of automated email notifications enhances accessibility, ensuring that students receive timely updates both on

their dashboards and in their inboxes. While challenges such as backend-frontend synchronization, SMTP authentication, and UI consistency required iterative refinement, the final system emerged as a scalable, reliable, and efficient solution. DNMS not only reduces administrative workload but also strengthens institutional communication, providing a foundation for future enhancements such as mobile app integration, SMS alerts, and analytics dashboards. Ultimately, the project validates that a well-designed digital communication platform can significantly improve transparency, efficiency, and engagement within educational environments.

IX. FUTURE WORK

Future enhancements may include:

1. GitHub repository integration
2. Real-time editor plugin support
3. CI/CD integration
4. Docker-based deployment packaging
5. Support for additional languages (Go, Rust)

REFERENCES

- [1] Sahana, R., & Aravindhan, P. (2025). Digital Notice Board Management System for Academic Institutions. *International Journal of Innovative Research in Computer Science & Technology (IJIRCST)*.
- [2] Bhor, P., & Darade, S. (2026). CampusFlow: A Smart Campus Ecosystem Integrating IoT and AI. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*.
- [3] Faisaluddin, M., Kumar, A., & Sharma, R. (2026). College News Portal: A Centralized Communication Platform for Students. *International Journal of Emerging Trends in Engineering Research (IJETER)*.
- [4] IJNRD Guidelines. (2026). Formatting and Publication Standards for Academic Papers. *International Journal of Novel Research and Development*.