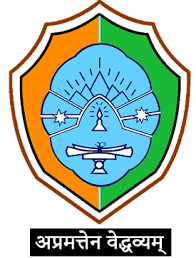
**PROGRESS REPORT ON**

**ONLINE NOTICE BOARD SYSTEM(BulletinX)**



**DEPARTMENT OF COMPUTER SCIENCE AND IT**

**COTTON UNIVERSITY, GHY-01**

**BCA 6th SEMESTER**

**SESSION-2022-25**

**SUBMITED BY**

SAEMI LANGTHASA

ENROLL: BCA2233014

MRIGANKA SARMA

ENROLL: BCA223303

**SUPERVISOR**

Dr. HITEN CHOUDHURY

HEAD OF DEPARTMENT, CSIT

COTTON UNIVERSITY, GHY-01

* **INTRODUCTION**

In the digital age, traditional paper-based notice boards are becoming outdated due to their limitations in accessibility, maintainability, and reach. To overcome these drawbacks, **BulletinX** is designed as a **web-based digital notice board** system that simplifies the dissemination of important announcements and notices for colleges and institutions.

This project provides a platform where **admins** can create, manage, and share notices in various formats (text/PDF), and **users** can easily view, filter, and subscribe to receive timely updates via email. The system is divided into two major parts: the **Admin Panel** and the **User/Public Interface**.

### **OBJECTIVE OF THE PROJECT**

The primary objectives of BulletinX are:

* To digitize the traditional notice board system with real-time access.
* To enable multiple administrators to manage content securely.
* To allow users to view and filter notices without logging in.
* To integrate a subscription model that notifies users about new notices via email.
* To create a clean, responsive, and scalable web application using modern technologies.

### **USED TECHNOLOGIES AND TECHNIQUE**

|  |  |
| --- | --- |
|  |  |
| Technology/Tools | **Purpose** |
| Next.js (React) | Full-stack framework for frontend + backend. |
| Typescript | Strongly typed language for reliability. |
| Prisma ORM | Database modeling and access. |
| PostgreSQL | Relational database for storing notices, users. |
| Nodemailer | Sending subscription-based email notifications. |
| Tailwind CSS | Utility-first CSS framework for styling. |
| Zod + React Hook Form | Form validation and error handling |
| Clerk | Manages admin authentication, sessions securely. |
| Cloudinary | Manages PDF uploads |

### **IMPLEMENTATION**

#### **User Section (Public Interface)**

Public users can:

* Access the notice board directly without login
* Filter notices using various filters
* View notices and download/view attached PDFs
* Subscribe with their name and email for future updates and send feedback
* Receive automatic confirmation and update emails

A discrete **"Admin Login"** link is present for administrative access.

The entire UI/UX is optimized for all devices (mobile, tablet, desktop) using Tailwind CSS and responsive layouts.

#### **Admin Panel**

The admin section includes a secured login/register mechanism. If an admin is not registered, they are redirected to a registration form. Upon successful login, the admin is taken to the **dashboard**, which displays recent notices and system stats.

Main admin features:

* Upload/edit/view notices (with PDF)
* Filter/search notices by type/date/title
* Manage user subscriptions
* Send custom messages and email notifications
* Settings panel for admin profile or system configurations

Navigation is handled using a **responsive navbar/sidebar** with links like Dashboard, Content, Notices, Users, and Settings.

### **CONCLUSION**

The development of **BulletinX** successfully replaces traditional notice systems with a dynamic, secure, and scalable platform. The admin system is fully functional, and the user interface ensures a smooth and accessible experience for public users. All critical modules have been completed and integrated, and testing confirms full functionality across common use cases.

### **FUTURE PLAN**

In the next phase of development, the following goals are identified:

* **Improve UI Design:** Enhance the visual aesthetics and branding consistency across all pages.
* **Make Entire System Mobile Responsive:** Complete the remaining design adjustments to ensure full responsiveness on mobile and tablet devices.
* **Integrate Public User Section with Real-Time Data**: Connect the user interface with the backend/database using API calls so that users can see dynamically updated notices uploaded by admins.
* **Implement Admin-Based Filtering:** Allow individual admins to view and filter only the notices they have uploaded, improving content control and data segregation.