**Project Title**

**Project Report**

*Submitted in partial fulfillment of the requirement of the degree of*

**BACHELORS OF TECHNOLOGY**

*to*

**K.R Mangalam University**

*by*

**ANSHIKA KASHYAP (2401010021)**

**Khushi(2401010089)**

Under the supervision of

**Dr. Pankaj Agarwal**



Department of Computer Science and Engineering

School of Engineering and Technology

K.R Mangalam University, Gurugram- 122001, India

**STUDENT CERTIFICATE**

**This is to certify that the Mini Project Synopsis entitled,  
“AI FOR MENTAL HEALTH CARE CHATBOT”**

**submitted by the undersigned students:**

* **Anshika Kashyap (Roll Number: 2401010021)**
* **Khushi (Roll Number:2401010089)**

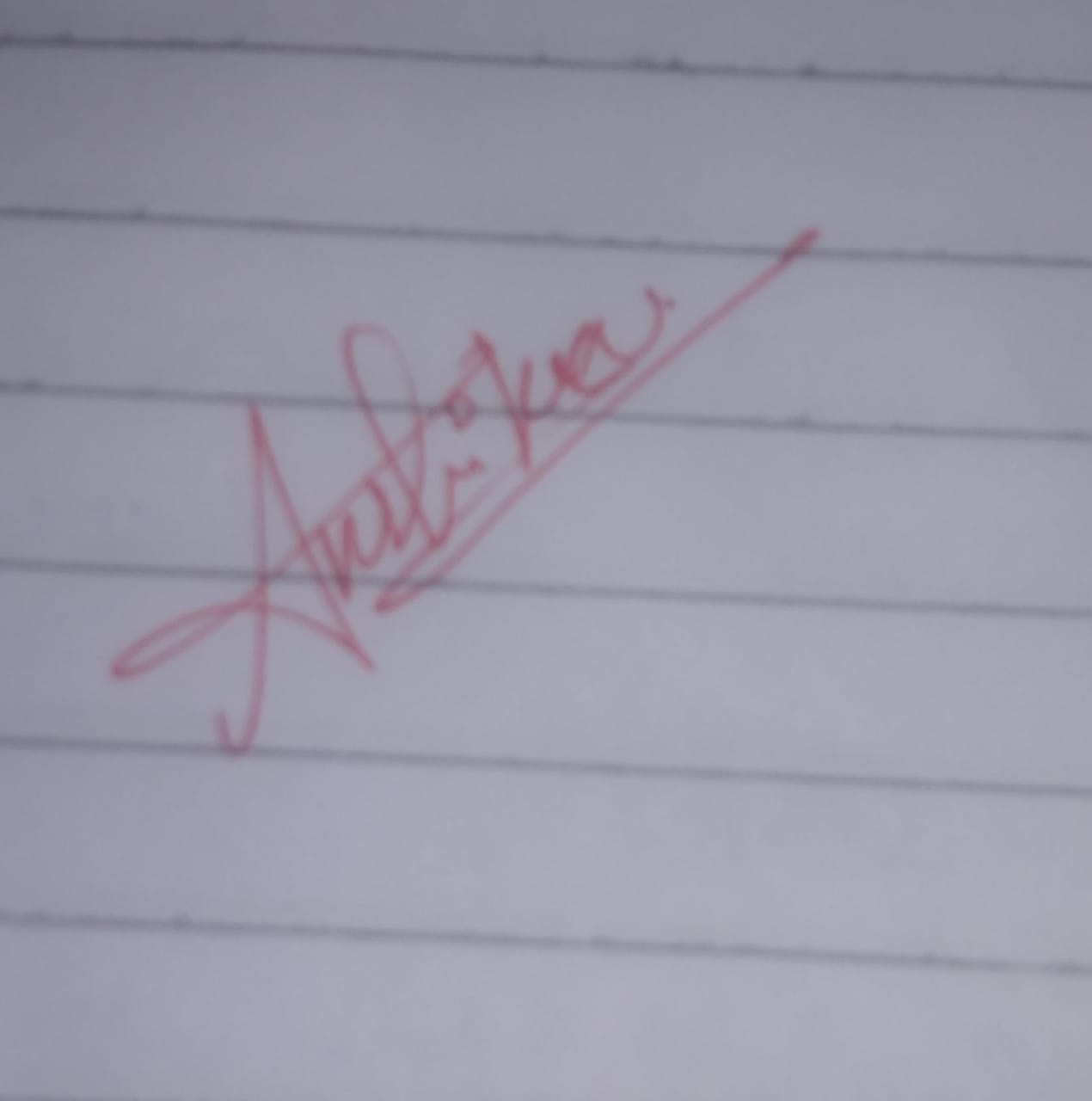
**is a bona fide record of original project work carried out by us during the academic session 2024-2028, as a partial requirement for the subject “MINI PROJECT” under the B.Tech CSE program at K.R. Mangalam University, Gurugram, India.**

**We further certify that:**

* **The project work is our own creation and has not been copied or reproduced from any other source.**
* **The content of this project is free from plagiarism and does not contain any content generated by AI tools, unless explicitly permitted and appropriately cited.**
* **All external references, tools, or frameworks used during the development of this project have been properly acknowledged.**

**We understand that any violation of academic integrity, including plagiarism or unauthorized use of AI-generated content, may lead to disciplinary action as per university policy.**

**Student 1 Signature:**

****

**Student 2 Signature:**

****

**Acknowledgement**

I would like to thank all those people who guided me during this mini project entitled "College Event Management System."

First and foremost, I would like to thank [Faculty Guide Name], my project guide, for their continuous guidance, valuable suggestions, and encouragement throughout the project. Their advice and knowledge played a major role in moulding this project into its final shape.

This project has been a valuable learning experience, and I am thankful to everyone who contributed to its success.

[ANSHIKA KASHYAP] [2401010021]

[KHUSHI] [2401010089]

[B.TECH CSE (CORE),SEC-A)]

**ABSTRACT**:

The College Event Management System is a comprehensive and user-centric web-based application designed to streamline the planning, registration, and administration of various events within an academic institution. Managing multiple college events manually often leads to workshops, upcoming department wise programs, downloaded certificate, missed deadlines, and lack of student engagement. This project aims to convert something into a digital and centralize event-related processes, making it easier for both students and administrators to collaborate, participate, and track event activities. The system provides secure user authentication with separate login/sign up functionalities for students and administrators. Once logged in, students can browse all upcoming events, view detailed descriptions including event name, date, time, venue, and category, and easily register their participation and also download directly certificate into their. In future email confirmations, message directly on their registered number and reminders ensure proper communication. The admin dashboard empowers administrators with tools to create, update, and delete events, as well as view a united list of registered participants. A built-in event calendar or timeline visualization helps in effective schedule management, while integrated on-screen notifications or email alerts inform users about recent updates or newly added events. The system may also include optional modules such as a feedback system, allowing students to submit event reviews, and a certificate generator to auto-create participation certificates for attendees. The project follows a full-stack architecture using HTML, CSS, and JavaScript (or Bootstrap) for the frontend interface, Python (Flask) for backend logic, and MySQL or SQLite for data storage. The development environment is Visual Studio Code, and optional integrations like SMTP for email notifications can be included to enhance user interaction. In conclusion, the College Event Management System serves as a robust, scalable, and beginner-friendly platform that not only digitizes event workflows but also enhances student participation and engagement in campus activities.

**Introduction**

The College Event Management System is a user-friendly, web-based application developed to streamline and modernize the planning, registration, and management of college events. Traditional event coordination often involves manual paperwork, lack of centralized communication, and inefficient registration processes, which can lead to confusion and mismanagement. This system seeks to resolve such issues by offering a centralized digital platform accessible to both students and administrators. Through this system, students can easily view upcoming events, register online, and receive real-time updates or notifications. Meanwhile, administrators are equipped with tools to create, update, and manage events, as well as monitor participant registrations and feedback. The inclusion of features such as user authentication, automated confirmation emails, and optional certificate generation makes the platform robust, secure, and highly functional. The project promotes digital transformation within college campuses, making event handling more structured, transparent, and scalable. It not only reduces the administrative burden but also enhances student engagement by ensuring a smoother experience in participating and managing events effectively. This system not only reduces administrative workload and minimizes the risk of errors but also enhances student participation by providing a streamlined and user-friendly experience. In addition, the digital platform ensures that all event related data is stored securely and is easily accessible, allowing for better planning and execution of future events.

**Objective**

• To develop a user-friendly and efficient platform for managing college events.

• To provide a seamless interface for both students and administrators.2

• To digitize manual processes such as:

o Event announcements

o Event registrations

o Administrative coordination

• To improve accessibility, transparency, and operational efficiency.

• To allow students to:

o Stay updated with upcoming events

o Register for events quickly and easily

o Receive timely updates and changes

• To enable administrators to:

o Manage all events from a centralized dashboard

o Track student registrations and feedback Oversee event schedules and coordination

• To enhance student engagement in campus activities.

• To reduce paperwork and manual errors in event handling.

• To create a modern, digital solution that benefits both students and faculty.

**Tools Used**

**Frontend:**

• HTML, CSS, JavaScript, and Bootstrap are utilized to design and develop an interactive, responsive, and visually appealing frontend that ensures an intuitive user experience for both students and administrators.

**Backend:**

• Python, using frameworks like Flask or Django, is employed to manage the server-side logic, ensuring smooth communication between the frontend and the database, handling form submissions, and executing necessary operations.

**Database:**

• MySQL or SQLite is used for storing event-related information, user data, and registration details in a structured manner.

**IDE(Integrated Development Environment):**

• Visual Studio Code (VS Code) serves as the integrated development environment (IDE) to write and test the code effectively.

**Modules**

• Authentication Module: Handles login and registration for students and admin

• Event Management Module: Allows admins to create, update, or delete events

• Event Listing Module: Displays upcoming events with detailed information

• Registration Module: Allows students to register for events

• Schedule Management Module: Shows events in calendar format

• Feedback Module (optional): Collects post-event feedback

from participants

• Certificate Generation Module (optional): Generates

certificates for attendees

**System Design**

The system follows a client-server architecture. The

frontend handles user interactions and sends requests to the

backend server. The backend processes the logic, interacts

with the database, and returns responses. The system ensures

proper separation of concerns and data flow through MVC

principles.

Data Flow Overview:

• User login/sign up

• Admin adds or edits events

• Students view and register for events

• Notifications are triggered before event dates

• Admins can view registered students

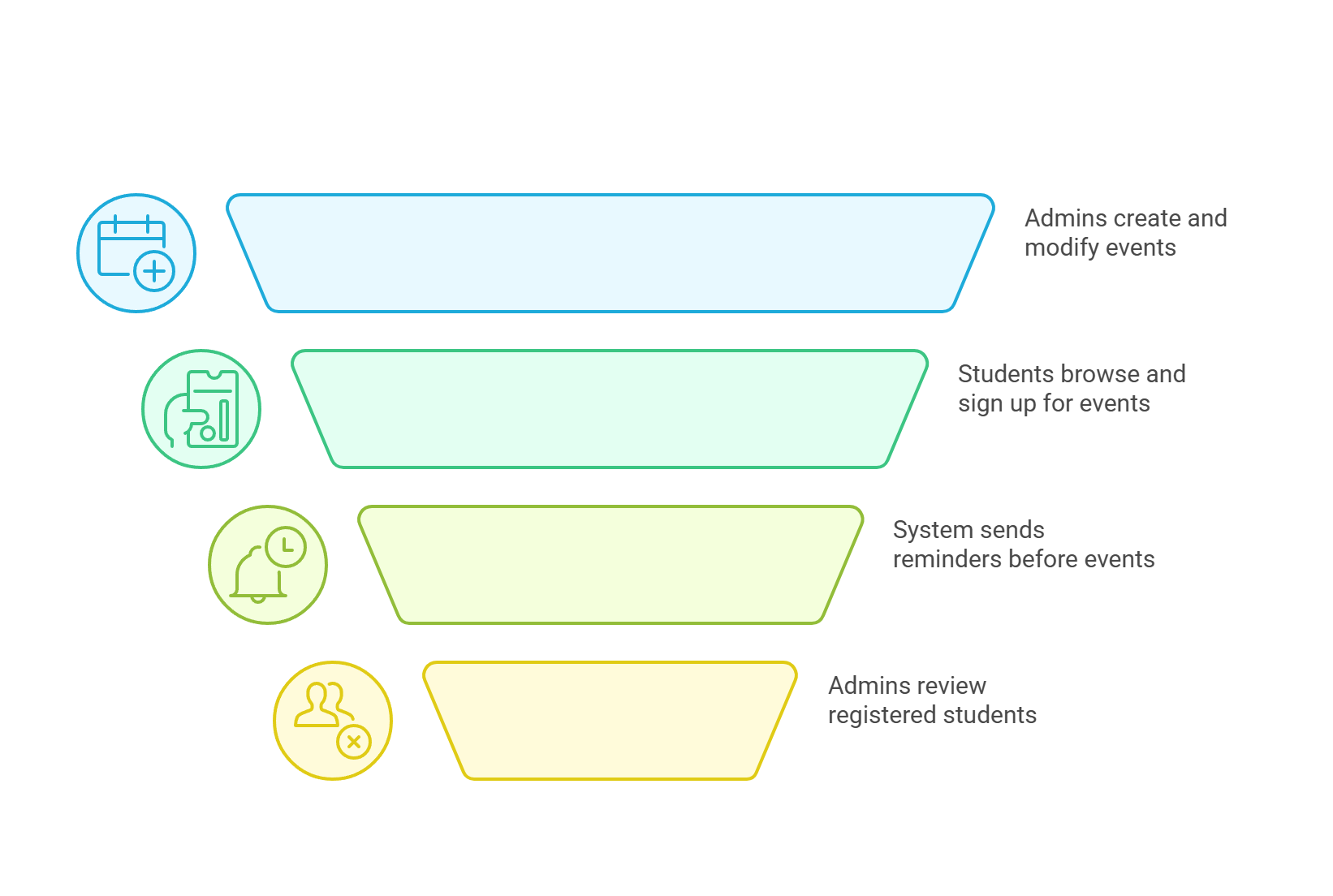


Fig 1:data flow

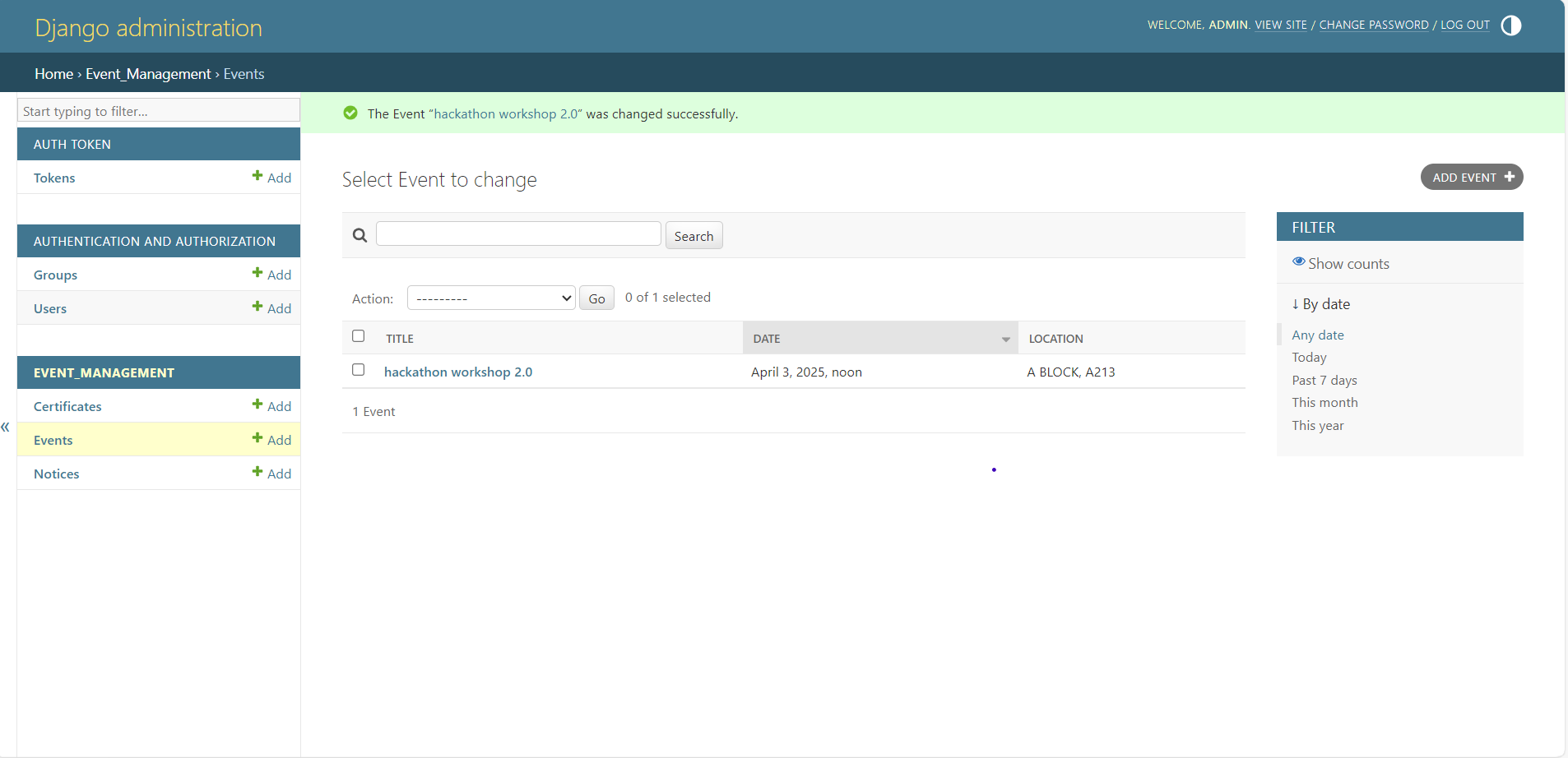


Fig 2:admin added events

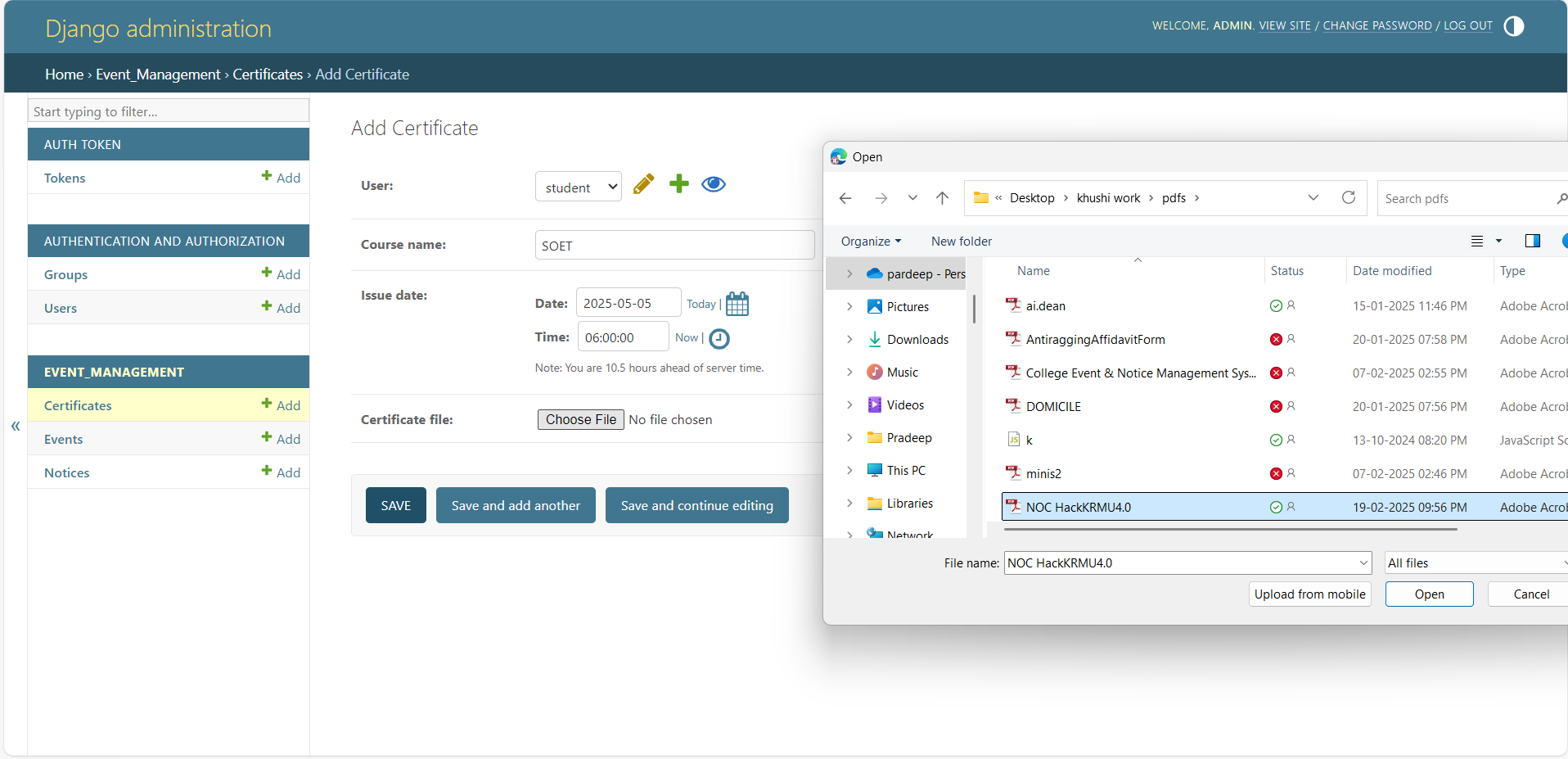


Fig 3:certification download

**Methodology**

The development of the College Event Management System follows a structured and iterative approach using the Software Development Life Cycle (SDLC) model. The key stages involved are:

1. Requirement Analysis

• Identified the primary needs of students and administrators in managing college events.

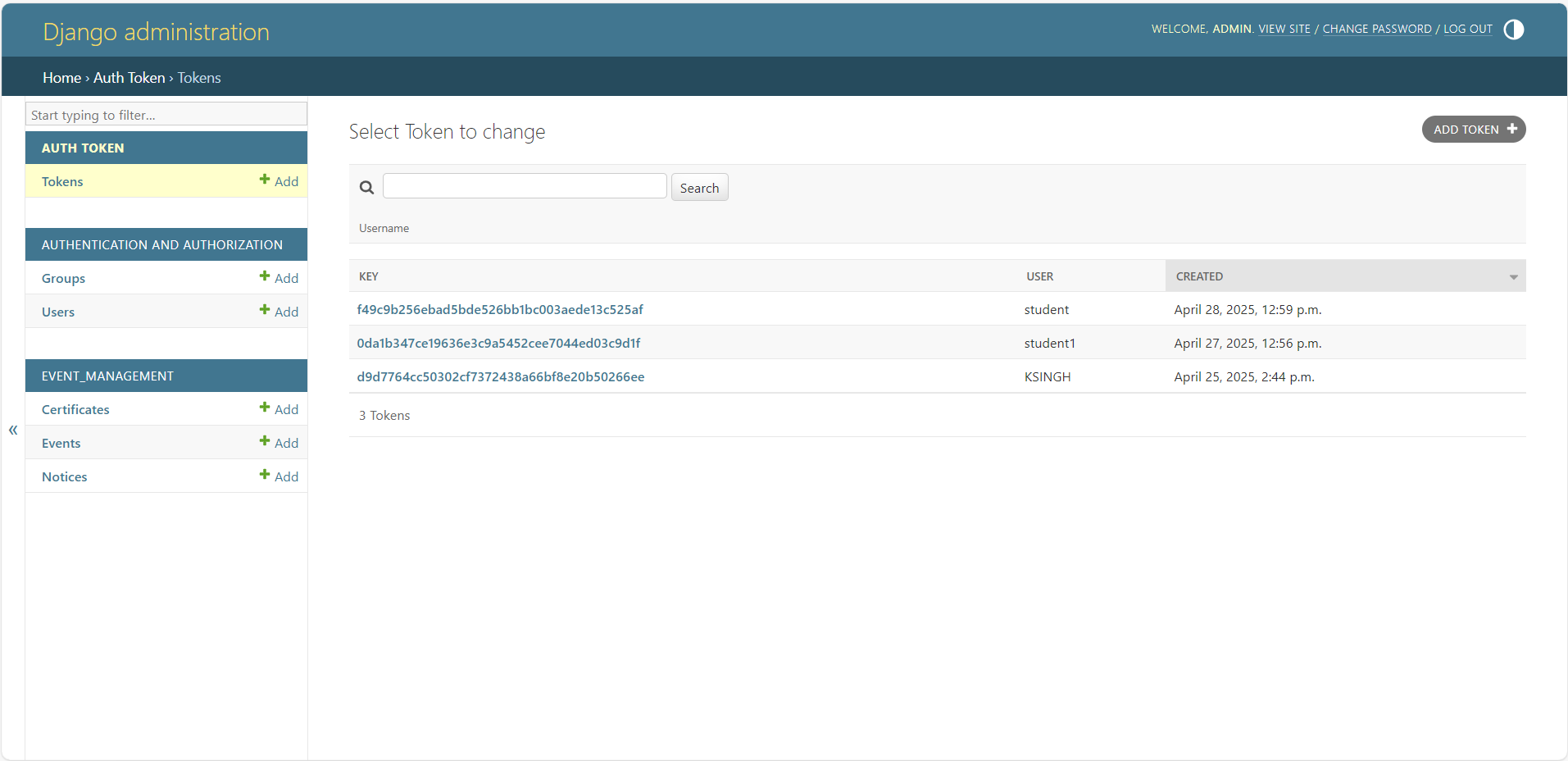
• Conducted informal surveys and discussions to understand common event-related challenges and user expectations.

1. System Design

• Designed the system architecture based on a client-server model.

• Created wire frames and flow diagrams to visualize user interfaces and system workflows.

• Applied the MVC (Model-View-Controller) design pattern to maintain separation of concerns.



1. Frontend Development

• Built user interfaces using HTML, CSS, Bootstrap, and JavaScript for responsiveness and ease of use.

• Developed pages for login, registration, event listings, event calendar, and admin dashboard.

1. Backend Development

• Used Python with Flask (or Django) for server-side programming.

• Created routes for handling user requests and business logic like user authentication, event creation, and registration.

1. Database Integration

• Used MySQL or SQLite to store data such as user details, events, and registrations.

• Established secure connections between backend logic and database using ORM or SQL queries.

1. Testing and Debugging

• Performed unit testing on modules like login, event registration, and admin panel.

• Fixed bugs related to form validation, database connectivity, and user session management.

1. Deployment and Review

• Ran the application on local server using VS Code.

• Collected feedback from test users and improved the system iteratively.

• Ensured cross-browser compatibility and responsive design for better usability.

**Implementation**

The system was implemented using Python (Flask framework). HTML, CSS, and JavaScript were used for creating interactive and responsive web pages. The backend handles routes, form submissions, and database operations. MySQL was used to store user and event information.

**Features**

• User Authentication (Student/Admin)

• Event Listings with details

• Online Event Registration

• Confirmation via email (optional)

• Admin Panel to manage events and users

• Notifications and reminders

• Event Calendar/Schedule view

• Feedback form (optional)

• Auto Certificate Generator (optional)Limitations

• Does not support payment integration

• Email notifications may require manual SMTP setup

• Limited security features in the basic version

• Feedback and certificate modules are optional and may need enhancements.

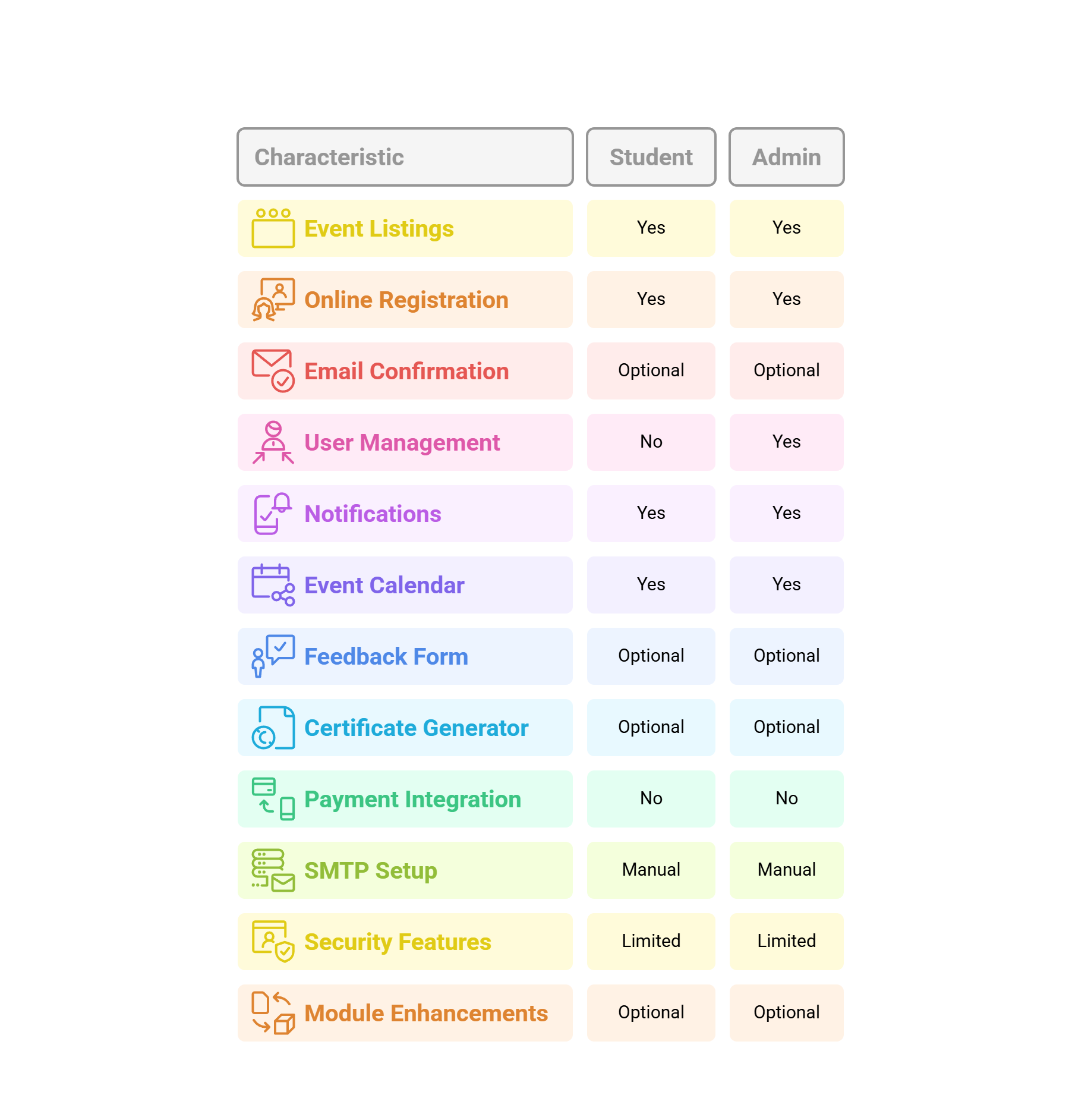


Fig 2:student/admin authentication

**Future Directions**

To improve the functionality, reach, and user experience of the College Event Management System, the following future enhancements are proposed:

1. Mobile Application Integration A dedicated mobile app for Android and iOS can be developed to increase accessibility and user engagement. Students will be able to register for events, receive updates, and view schedules directly on their smartphones. Push notifications can be used for real-time alerts.
2. Advanced Email Configuration A robust email service using professional APIs like Send Grid or Gmail API can be integrated to automate confirmation emails, reminders, and event updates. This will enhance the communication experience between the system and users.
3. Comprehensive Notification System In addition to email and on-screen alerts, SMS and in-app push notifications can be implemented to ensure that students do not miss any important event-related updates.
4. Admin Analytics Dashboard An advanced dashboard for administrators can display statistics such as event participation trends, most active users, and feedback ratings, helping in better decision-making.
5. QR Code-Based Event Check-In A QR code system can be implemented for quick and secure attendance marking during events, reducing the need for manual registers.
6. ERP and Calendar Integration Integration with the college ERP system and Google Calendar would allow better synchronization of academic and extracurricular activities.
7. AI-based Recommendations (Optional) AI can be used in the future to suggest events to students based on their interests, past registrations, and departmental preferences.

**Challenges Faced**

During the development of the College Event Management System, several technical and operational challenges were encountered:

1. User Authentication Security Ensuring secure login and signup for both students and administrators was a key concern. Basic authentication was implemented, but stronger measures like hashed passwords and session management could be added in future iterations.
2. Database Management Designing a relational database that supports multiple event registrations per student, and allows efficient querying by admin users, required careful planning. Normalization and indexing helped improve performance.
3. Frontend-Backend Integration Connecting the user interface with backend routes and ensuring seamless form submissions and validations posed initial difficulties, especially in form data handling and AJAX-based updates.

4. Notification System Initially, integrating automated email or SMS nnotifications presented issues with SMTP configuration and spam filtering. This feature was simplified to on-screen notifications to ensure system stability.

5. Responsive Design Making the application visually consistent across various devices and screen sizes took extra effort. Bootstrap helped mitigate these issues and create a mobile-friendly layout.

6. Testing and Debugging Manual testing of different user flows and identifying bugs like incorrect form submissions, duplicate registrations, or session timeouts required a thorough QA approach.

7. Limited Resources and Time As a student project, time and technical resources were limited. As a result, optional features like certificate generation and feedback analytics were either simplified or marked for future enhancement.

**Results**

The College Event Management System was successfully developed and tested using Python (Flask), HTML, CSS, JavaScript, and MySQL. Upon deployment, the system provided the following measurable and qualitative results:

• Efficient Event Handling: All event-related tasks such as event creation, editing, and deletion were streamlined and managed from a single admin panel.

• Improved Scheduling: The calendar and schedule view helped students and administrators keep track of all upcoming events with ease.

• Simplified Registration: Students were able to register for events through an intuitive and quick online form, which reduced paperwork and saved time.

• User Authentication: Login and registration systems for both students and admins worked securely and as intended, ensuring role-based access.

• Database Accuracy: MySQL integration allowed proper storage and retrieval of event and user data, eliminating manual errors.

• Optional Features Performance: The feedback module and certificate generator (if implemented) operated smoothly and enhanced post-event engagement.

• Responsive UI: The frontend was responsive and accessible on both desktops and mobile browsers (tested with Bootstrap), improving the user experience.

**Conclusion**

The College Event Management System serves as a practical and innovative solution to the traditional, manual methods of event handling in academic institutions. It significantly reduces the administrative burden, provides an intuitive user interface for both students and administrators, and ensures a streamlined approach to event registration, scheduling, and communication. By offering digital management of events, registrations, and notifications, it improves not only the user experience but also administrative control, making event management more efficient and organized. Furthermore, this system enhances transparency and accessibility by allowing students to easily register for events and stay updated with event schedules, eliminating the need for cumbersome paperwork and manual tracking. Administrators can manage events effortlessly through a centralized platform, giving them more time to focus on enhancing the quality of events and student engagement. As the system evolves, there is immense potential for future improvements. Integrating features such as enhanced security and integration with college ERP platforms could enable seamless data sharing and reporting for better administrative oversight. In conclusion, the College Event Management System not only optimizes event management processes but also offers scalability and flexibility for future growth. It is a significant step toward transforming the way academic institutions manage events, bringing them into the digital age. With future enhancements, the system can serve as a comprehensive tool for improving student participation, administrative workflow, and overall event organization.

**DISCUSSION**

The development of the College Event Management System provided valuable insights into the design and implementation of real-world web applications. During the development process, both technical and functional challenges were encountered. One of the primary challenges was ensuring that the system remained user-friendly for students with varying levels of technical proficiency. This was addressed by using a simple and intuitive UI designed with HTML, CSS, and Bootstrap. Another issue was handling secure user authentication and data storage. We resolved this by implementing login systems with role-based access control and by using MySQL for reliable data management. The use of the Flask framework in Python allowed for fast and modular backend development, making the system scalable and easier to maintain. Although modules like email notifications and certificate generation were optional, they highlighted the importance of user engagement and event followup. However, to keep the system lightweight, mobile and email push notifications were excluded in this version.This system improves event visibility, simplifies registration, and reduces manual administrative tasks. It has the potential to be adopted and expanded across different colleges or departments, with future upgrades like mobile app integration, QR-based attendance, and analytics dashboards to further improve functionality.\

**REFRENCES**

1. Flask Documentation Official documentation for Flask, the Python web framework used for backend development. <https://flask.palletsprojects.com/>
2. MySQL Documentation Official MySQL documentation for understanding database setup, queries, and management. https://dev.mysql.com/doc/
3. W3Schools A valuable resource for tutorials on HTML, CSS, JavaScript, and other web technologies. <https://www.w3schools.com/>
4. Bootstrap Documentation Official documentation for the Bootstrap framework, which was used for responsive design in the frontend. <https://getbootstrap.com/>
5. Python Documentation Official Python documentation for understanding Python’s libraries and syntax. https://docs.python.org/
6. SMTP Email Guide (Real Python) A detailed guide on how to send emails using Python and the SMTP protocol. <https://realpython.com/python-send-email/>
7. Flask Documentation Official documentation for Flask, the Python web framework used for backend development. <https://flask.palletsprojects.com/>
8. MySQL Documentation Official MySQL documentation for understanding database setup, queries, and management. https://dev.mysql.com/doc
9. W3Schools A valuable resource for tutorials on HTML, CSS, JavaScript, and other web technologies. <https://www.w3schools.com>
10. Bootstrap Documentation Official documentation for the Bootstrap framework, which was used for responsive design in the frontend. <https://getbootstrap.com>
11. Python Documentation Official Python documentation for understanding Python’s libraries and syntax. https://docs.python.org/
12. SMTP Email Guide (Real Python) A detailed guide on how to send emails using Python and the SMTP protocol. https://realpython.com/python-send-email/