A

Project Report

On

**Smart College Event Management System**

Submitted in partial fulfillment of the requirement for the degree of

**Bachelor of Technology**

**In**

**Computer Science and Engineering**

By

**Ansh Rana 2261107**

**Ayush Rana 2261134**

**Shobhit Rana 2261531**

**Amandeep Kaur 2261089**

**Under the Guidance of**

**Mr. Prince Kumar**

**ASSISTANT PROFESSOR**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**GRAPHIC ERA HILL UNIVERSITY, BHIMTAL CAMPUS**

**SATTAL ROAD, P.O. BHOWALI,**

**DISTRICT- NAINITAL-263132**

**2024-2025**

**STUDENT’S DECLARATION**

We, **Ansh Rana, Ayush Rana, Shobhit Rana, Amandeep Kaur**  hereby declare the work, which is being presented in the project, entitled ‘ **Smart College Event Management System** ’ in partial fulfillment of the requirement for the award of the degree **Bachelor of Technology (B.Tech.)** in the session **2024-2025**, is an authentic record of my work carried out under the supervision of Mr. Prince Kumar.

The matter embodied in this project has not been submitted by me for the award of any other degree.

Date: 26/05/2025 Ansh Rana

Ayush Rana

Shobhit Rana

Amandeep Kaur

**CERTIFICATE**

**The project report entitled “Smart College Even Management System” being submitted by Ansh Rana S/o Surinder Singh Rana, 2261107 of B.Tech.(CSE) to Graphic Era Hill University Bhimtal Campus for the award of bonafide work carried out by them. They have worked under my guidance and supervision and fulfilled the requirement for the submission of a report.**

**Mr. Prince Kumar Dr. Ankur Singh Bisht**

**(Project Guide) (Head, CSE)**

**ACKNOWLEDGEMENT**

We take immense pleasure in thanking the Honorable Director **‘Prof. (Col.) Anil Nair (Retd.)’,** GEHU Bhimtal Campus to permit me and carry out this project work with his excellent and optimistic supervision. This has all been possible due to his novel inspiration, able guidance, and useful suggestions that helped me to develop as a creative researcher and complete the research work, in time.

Words are inadequate in offering my thanks to GOD for providing me with everything that we need. We again want to extend thanks to our president **‘Prof. (Dr.) Kamal Ghanshala’** for providing us with all infrastructure and facilities to work in need without which this work could not be possible.

Many thanks to **‘Dr. Ankur Singh Bisht’** (Head, Department of Computer Science and Engineering, GEHU Bhimtal Campus), our project guide **‘Mr. Prince Kumar’** (Assistant Professor, Department of Computer Science and Engineering, GEHU Bhimtal Campus) and other faculties for their insightful comments, constructive suggestions, valuable advice, and time in reviewing this report.

Finally, yet importantly, We would like to express my heartiest thanks to our beloved parents, for their moral support, affection, and blessings. We would also like to pay our sincere thanks to all my friends and well-wishers for their help and wishes for the successful completion of this project.

**Ansh Rana 2261107**

**Abstract**

The College Event Management System Project is a web-based application project designed to streamline the organization and management of college events. The main goal of this project is to provide college administrators, event coordinators, and participants with an efficient platform to manage and participate in college events online. The login section will have three login facilities for the Administrator, Event Coordinator, and Participant. The application features a simple and pleasant user interface, leveraging the Bootstrap Framework v4, along with user-friendly functionalities.

The Administrator plays a crucial role in overseeing and managing the system. This encompasses various tasks, including adding new events, editing existing ones, and overseeing overall event activity. When creating new events, the Administrator collects pertinent information from coordinators and inputs it into the system to generate event profiles. Additionally, the Administrator is empowered to modify event details, such as schedules, venues, or participant information, as needed. Monitoring event activity is also within the Administrator's purview. Furthermore, the Administrator communicates with coordinators and participants, providing feedback and guidance as necessary. Whether addressing event-related issues or disseminating important updates, the Administrator ensures a smooth and efficient event management experience.

The Event Coordinator, upon logging in through a secure platform provided by the college, can manage specific events. Coordinators can view event details, which include a comprehensive record of event-related activities and logistics. This includes information such as dates, descriptions, and the status of each task related to the event. The coordinator can also track participant registration, manage schedules, and handle any logistical issues that arise. This information allows Coordinators to ensure that events run smoothly, monitor participant engagement, and identify any areas needing attention.

The Participants can log into the system to view upcoming events, register for events, and receive updates. They can access event schedules, venue details, and any pertinent information regarding their participation. This platform provides participants with the ability to track their event activities, receive notifications about changes or updates, and stay informed about college events efficiently.

The Event Coordinators assist participants and other stakeholders by encoding event registration and management details into the college's event management system. For event registrations, they enter details like the participant's name, student ID, event name, and registration date. For event updates or cancellations, they input similar information along with any pertinent details regarding the changes. This process ensures that event records are accurately updated. The coordinator’s responsibilities include handling event registrations, managing event logistics, and ensuring participants receive necessary information. Throughout these processes, Coordinators follow strict protocols to maintain the integrity and smooth execution of events.

**TABLE OF CONTENTS**

Declaration…………………………………………………………………………………ii

Certificate…………………………………………………………………………………..iii

Acknowledgement………………………………………………………………………….iv

Abstract……………………………………………………………………………………..v

Table of Contents…………………………………………………………………………..vii

List of Abbreviations……………………………………………………………………….ix

**CHAPTER 1 INTRODUCTION**……………...………………………...10

1. Prologue……………………………………………………………………………10
2. Background and Motivations………………………………………………………11
3. Problem Statement………………………………………………………………....12
4. Objectives and Research Methodology…………………………………………….13
5. Project Phases……...……………………………………………………………….16

**CHAPTER 2 PHASES OF SOFTWARE DEVELOPMENT CYCLE**

1. Hardware Requirements……………………………………………………………18
2. Software Requirements…………………………………………………………….18

**CHAPTER 3 SNAPSHOT………………………………………………..19**

**CHAPTER 4 LIMITATIONS (WITH PROJECT) ……………..……...20**

**CHAPTER 5 ENHANCEMENTS……………………………………......21**

**CHAPTER 6 CONCLUSION………………………………………….....22**

**REFERENCES………………………………………….....23**

**LIST OF ABBREVIATIONS**

|  |  |  |
| --- | --- | --- |
| **Abbreviation** | **Full Form** | **Purpose/Use in the Document** |
| PHP | Hypertext Preprocessor | Used for server-side scripting in the web-based event management system. |
| MySQL | My Structured Query Language | The database management system used for storing event data. |
| HTML | HyperText Markup Language | Used for structuring web pages in the frontend of the system. |
| CSS | Cascading Style Sheets | Used for styling the frontend interface of the application. |
| XAMPP | Cross-platform Apache, MySQL, PHP, Perl | A local server environment used for development and testing of the web application. |
| ER Diagram | Entity-Relationship Diagram | Used for data modeling and system architecture planning. |
| RBAC | Role-Based Access Control | Proposed enhancement to customize access permissions for different user roles dynamically. |
| 2FA | Two-Factor Authentication | A security enhancement mentioned under limitations; not yet implemented. |
| SMTP | Simple Mail Transfer Protocol | Referenced as an example protocol for email notification integration. |
| API | Application Programming Interface | Suggested for future integration of features like SMS and email notifications. |
| UI | User Interface | Refers to the design and layout of the application as experienced by the user. |

**INTRODUCTION**

1.1 Prologue

In the digital age of today, seamless imbibing of technology in academic and administrative functions has become a requirement, not an extravagance. Functions—be they technical fests and cultural events, or academic seminars—constitute an energetic component of college life, adding to the learning experience and encouraging interaction between students and teachers alike. Yet, organizing such events conventionally entails convoluted logistics, scattered communication, and tedious labor through manual interventions.

The College Event Management System, represents a solution to such difficulties in the form of an intelligent and systematic digital platform for organizing and conducting college events. Having separate modules for administrators, coordinators, and participants, the system facilitates tasks like scheduling, registration, communication, and feedback management. Built utilizing contemporary web technologies such as PHP, MySQL, HTML, and CSS, and run on a local server through XAMPP, the system makes the platform accessible, efficient, and secure.

This report describes the process of conceptualization, design, and deployment of the Preranothsava system—from the realization of the requirement for automation of event handling to the provisioning of a robust and user-friendly solution. Throughout the subsequent chapters that you read, you will learn about the project's mission, system architecture, technical tools, implementation process, and the impact it seeks to generate within institutions of higher learning

1.2 Background and Motivations

College campuses are dynamic environments that host a variety of events—academic, cultural, technical, and extracurricular—each requiring meticulous planning and coordination. Traditionally, the management of such events has relied heavily on manual processes: physical registration forms, word-of-mouth announcements, printed schedules, and direct communication between organizers and participants. While these methods may suffice for small-scale events, they often lead to inefficiencies, communication breakdowns, and logistical errors as the scale and complexity of events grow.

In the age of digitization, where technology has transformed virtually every aspect of human activity, it becomes imperative for educational institutions to modernize how they manage events. The need for a centralized, accessible, and user-friendly platform that can handle event scheduling, participant registration, feedback collection, and real-time communication is more evident than ever. Such a system not only enhances operational efficiency but also elevates the overall user experience for both organizers and participants.

The motivation behind developing the **College Event Management System** stems from the desire to bridge the gap between traditional event coordination and modern digital tools. By automating core functions and enabling seamless interaction among administrators, coordinators, and students, the system aims to reduce manual workload, minimize errors, and ensure timely dissemination of information. It also supports transparency, accountability, and improved decision-making through structured data management and feedback analysis.

Furthermore, the project serves as a learning experience in applying software development principles to a real-world problem, utilizing technologies such as PHP, MySQL, HTML, and CSS, and integrating them into a cohesive application hosted on a local development environment using XAMPP. Ultimately, Preranothsava aspires to contribute to the digital transformation of campus life, fostering a more organized, engaging, and inclusive environment for all stakeholders.

1.3 Problem Statement

Colleges regularly organize a wide range of events such as cultural fests, technical workshops, seminars, competitions, and awareness programs. These events play a crucial role in enhancing student engagement, promoting talent, and fostering a vibrant campus culture. However, the management of these events is often conducted through outdated, manual processes that are inefficient, time-consuming, and prone to human error.

Key challenges include difficulties in coordinating between multiple stakeholders (administrators, coordinators, and participants), maintaining accurate and up-to-date registration data, communicating event updates effectively, and collecting structured feedback for future improvement. Event organizers frequently struggle with compiling participant lists, managing schedules, and handling last-minute changes, while participants lack real-time access to event information and confirmation of their registration status.

These limitations result in poor coordination, reduced participation, miscommunication, and ultimately, a suboptimal experience for everyone involved. Additionally, the absence of a centralized system makes it difficult to maintain historical event data or generate meaningful reports for decision-making and performance evaluation.

To address these challenges, there is a need for a comprehensive and user-friendly digital platform that can automate and streamline all aspects of event management. The **College Event Management System (Preranothsava)** is designed to meet this need by providing a web-based solution that supports event creation, participant registration, schedule management, real-time notifications, and feedback collection. It aims to enhance operational efficiency, ensure data accuracy, and deliver a seamless experience to users across different roles and devices.

By implementing this system, educational institutions can modernize their event management processes, reduce administrative overhead, and create a more engaging and responsive event environment for the college community.

1.4 Objectives and Research Methodology

The objectives of the College Event Management System encompass enhancing participant experience, improving event coordination efficiency, and ensuring the security and reliability of event-related services. Specific objectives include:

* Event Scheduling and Management: Efficiently schedule and manage events, ensuring accurate tracking and reporting of event details such as dates, venues, and participants.
* Participant Information Retrieval: Provide users with quick and secure access to their event information, facilitating seamless registration, inquiries, and participation.
* Registration Processing: Streamline registration processing for timely and accurate enrolment in events, enhancing participant satisfaction and operational efficiency.
* Event History Display: Display detailed event history to users, enabling them to track and review past events for better engagement and transparency.
* Feedback Mechanism: Allow participants and coordinators to provide feedback, enhancing user engagement and enabling the college to gather valuable insights for improving event services.

Research Methodology

The development of the **College Event Management System** is grounded in a systematic research methodology that combines both theoretical study and practical implementation. The methodology followed in this project includes the following key phases:

1. Problem Identification and Requirement Analysis

The initial phase involved identifying the inefficiencies and limitations of traditional event management processes in educational institutions. This was achieved through:

* Observations of existing event coordination practices.
* Informal discussions with college administrators, student coordinators, and participants.
* Analysis of common challenges faced during event planning, registration, and execution.

Based on this, the core system requirements were outlined, including user roles, features, data flow, and security needs.

2. Literature Review and Technology Research

To ensure an informed and effective solution, a review of existing literature and tools was conducted. This included:

* Studying academic resources on database systems and web development.
* Exploring documentation and tutorials for relevant technologies like PHP, MySQL, HTML, CSS, and XAMPP.
* Evaluating existing event management systems to identify gaps and inspiration for improvements.

This helped in defining the scope of the project and selecting the appropriate tech stack.

3. System Design and Planning

In this phase, the structural and functional aspects of the system were planned using:

* Entity-Relationship (E-R) Diagrams to represent data relationships.
* Schema Diagrams for database architecture.
* Module Division to assign clear responsibilities to each component (Admin, Coordinator, Participant).

User interaction and data flow were mapped to ensure a logical, intuitive user experience.

4. Development and Implementation

Using an iterative development approach, the system was built in modules:

* Frontend: Developed using HTML and CSS for structure and styling.
* Backend: Built using PHP for server-side logic and MySQL for database management.
* XAMPP: Used as the local development environment for testing before deployment.

Each module (Participant, Coordinator, Administrator) was tested individually and then integrated to form a cohesive application.

5. Testing and Evaluation

The system underwent rigorous testing to ensure its reliability and functionality:

* Unit Testing: Each function and feature was tested in isolation.
* Integration Testing: Ensured that different modules worked together smoothly.
* User Testing: Simulated real-user interaction to test usability, navigation, and performance.

Bugs were documented and resolved, and user feedback was collected to refine the interface and experience.

6. Documentation and Review

Throughout the project lifecycle, detailed documentation was maintained to track progress, decisions, and changes. The final system was reviewed against initial objectives to ensure:

* Functional completeness
* Data accuracy and security
* User satisfaction and ease of use

1.5 Project Phases

The project was divided into the following key phases to streamline development:

1. **Planning and Requirement Gathering**
   * Identified goals, scope, stakeholders, and core functionalities.
   * Consulted with faculty and end users for feedback.
2. **System Design**
   * Created E-R diagrams, schema diagrams, and module layouts.
   * Defined the relationships between entities and user roles.
3. **Development**
   * Built the backend using PHP and MySQL (via XAMPP).
   * Developed front-end interfaces using HTML and CSS.
   * Integrated modules for Admin, Event Coordinators, and Participants.
4. **Testing**
   * Conducted functional testing of each module.
   * Verified system performance and resolved identified issues.
5. **Deployment and Review**
   * Deployed on a local server for demonstration.
   * Collected feedback from users (students/coordinators).
   * Final system review with project mentor.
6. **Documentation**
   * Prepared final project report, user manual, and presentation materials.

**3. Tools and Technologies Used**

|  |  |
| --- | --- |
| **Tool/Technology** | **Purpose** |
| **XAMPP** | Local development server environment |
| **PHP** | Server-side scripting |
| **MySQL** | Relational database management |
| **HTML/CSS** | User interface design |
| **ER Diagram Tool** | System architecture modeling |
| **Microsoft Word** | Project documentation |

**SOFTWARE AND HARDWARE REQUIREMENTS**

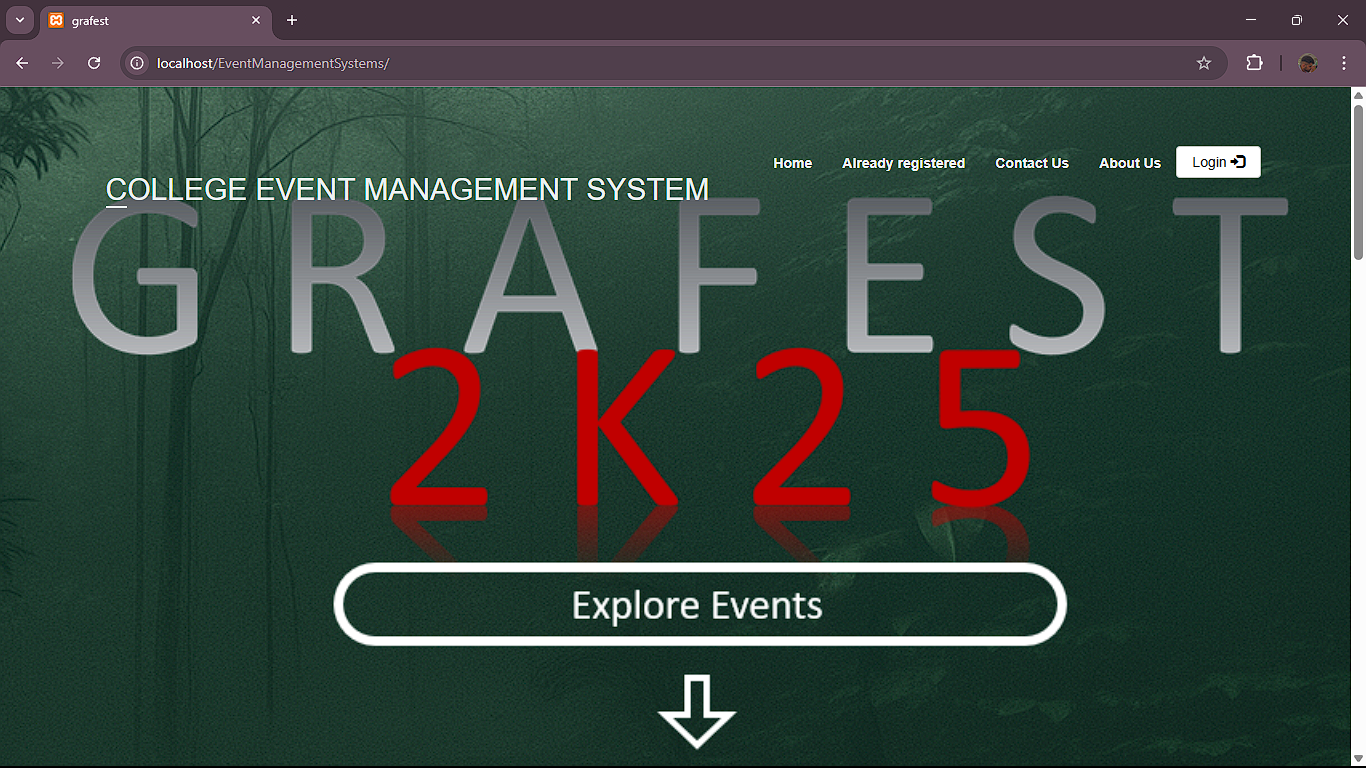
2.1 Software Requirements

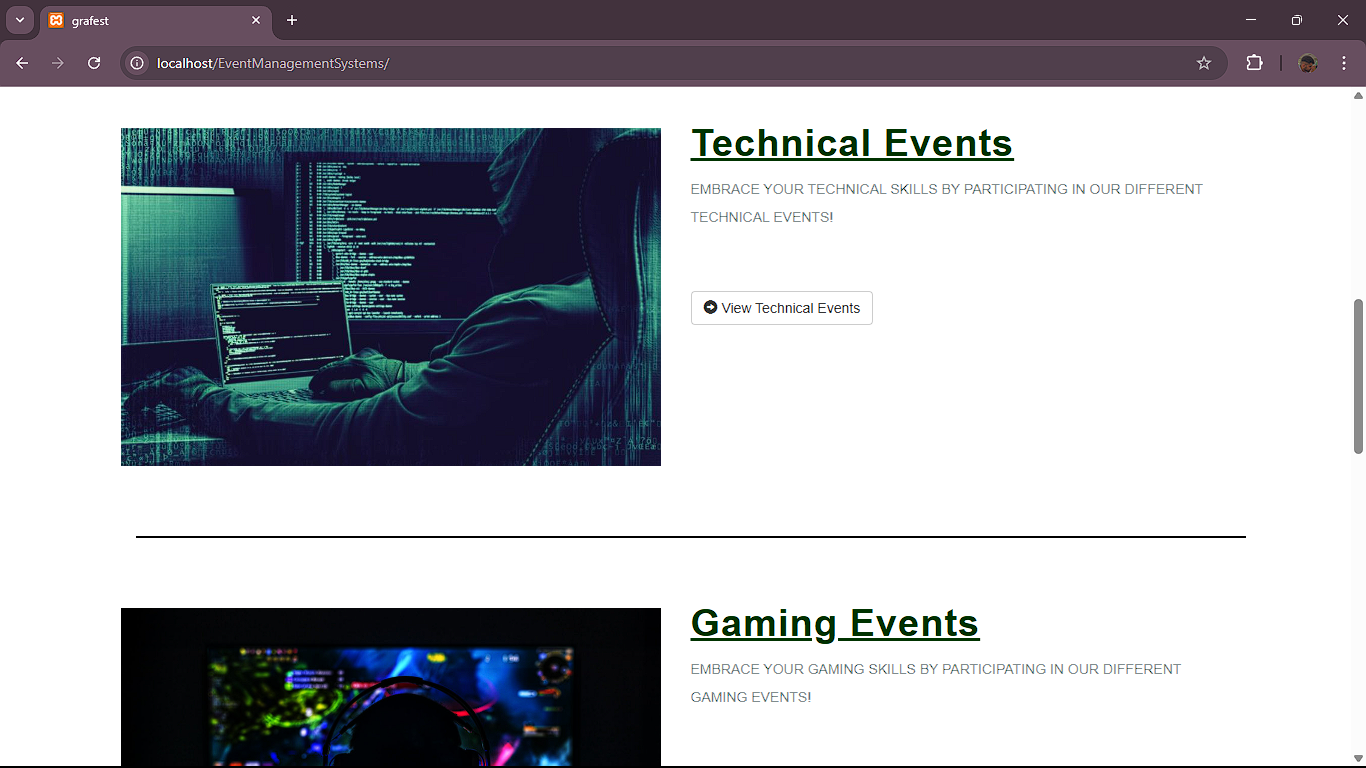
|  |  |
| --- | --- |
| **Softwre Component** | **Specification / Use** |
| **Operating System** | Windows 10 or above / Linux / macOS |
| **Web Server** | Apache (included in XAMPP) |
| **Database** | MySQL (included in XAMPP) |
| **Server-Side Language** | PHP (included in XAMPP) |
| **Frontend Technologies** | HTML5, CSS3, JavaScript |
| **Development Platform** | XAMPP (Cross-platform Apache, MySQL, PHP, Perl) |
| **Browser** | Google Chrome / Mozilla Firefox / Microsoft Edge (for testing/viewing) |
| **Code Editor** | Visual Studio Code |
| **Diagram Tools** | Draw.io / Lucidchart (for ER & schema diagrams) |
| **Documentation Tool** | Microsoft Word / Google Docs |

2.2 Hardware Requirements

|  |  |
| --- | --- |
| **Hardware Component** | **Minimum Specification** |
| **Processor** | Intel i3 or equivalent (dual-core or higher) |
| **RAM** | Minimum 1 GB (Recommended: 4 GB or more) |
| **Storage** | Minimum 3 GB free space (for database and application) |
| **Display** | 1024×768 resolution or higher |
| **Network** | Internet connection (for cloud testing or deployment) |
| **Input Devices** | Keyboard and Mouse |

**SNAPSHOTS**

****

****

**LIMITATIONS**

Lack of Role-Based Access Customization

* Explanation: While the system includes roles like Administrator, Coordinator, and Participant, the permission levels are fixed and not dynamically configurable.
* Impact: This reduces flexibility for institutions that may require more granular access control or additional user roles in the future.

No Integration with Email or SMS Services

* Explanation: The system does not include automated email or SMS notifications for registration confirmations, reminders, or updates.
* Impact: Users must rely solely on in-system notifications, which could affect timely communication and engagement.

Basic Security Features

* Explanation: Security measures like input validation, session management, and basic authentication are implemented, but advanced features like:
  + Two-Factor Authentication (2FA)
  + Encryption of sensitive data
  + CAPTCHA protection are not included.
* Impact: The system is vulnerable to basic security threats if deployed online without further enhancement.

No Automated Scheduling or Conflict Detection

* Explanation: The event scheduling feature does not check for clashes in timing or venue usage.
* Impact: In large-scale environments, overlapping event scheduling may occur, causing logistical issues.

**ENHANCEMENTS**

Online Hosting and Cloud Deployment

* Enhancement: Deploy the system on a live web server or cloud platform (e.g., AWS, Azure).
* Benefit: Enables remote access, scalability, and broader usage across campuses and devices.

2. Advanced Role-Based Access Control (RBAC)

* Enhancement: Implement a dynamic RBAC system where user roles and permissions can be customized by the administrator.
* Benefit: Greater flexibility and security for institutions with diverse user hierarchies.

3. Email and SMS Notification Integration

* Enhancement: Integrate email and SMS APIs (e.g., SMTP, Twilio) for automatic notifications for registration confirmations, event reminders, and updates.
* Benefit: Improves communication efficiency and participant engagement.

4. Mobile-Responsive Design or Dedicated Mobile App

* Enhancement: Optimize the UI for mobile devices or develop a native Android/iOS mobile app.
* Benefit: Enhances accessibility and usability for students on smartphones and tablets.

5. Event Scheduling Conflict Detection

* Enhancement: Add logic to check for clashes in event timing, venues, or resources.
* Benefit: Prevents double bookings and ensures smoother coordination.

**CONCLUSION**

The **College Event Management System** marks a significant step toward modernizing how educational institutions organize and manage events. By leveraging web technologies such as PHP, MySQL, HTML, CSS, and XAMPP, the system provides a centralized, user-friendly platform that simplifies the coordination of events, enhances participant engagement, and ensures smoother communication among administrators, coordinators, and students.

Throughout the project development lifecycle—ranging from problem identification and requirement gathering to design, implementation, and testing—careful attention was paid to real-world challenges faced during event organization in academic settings. The result is a system that addresses core functionalities like event scheduling, participant registration, feedback collection, and role-based access control.

While the current version of the system fulfills the primary objectives, it also lays the foundation for future upgrades. There are numerous opportunities to expand and enhance the platform by integrating advanced features such as mobile responsiveness, cloud deployment, automated notifications, conflict detection, and enhanced security protocols.

In conclusion, *Preranothsava* not only improves operational efficiency but also fosters a more connected, informed, and responsive campus community. It stands as a practical demonstration of how thoughtful application of technology can solve everyday institutional challenges and significantly enrich the academic experience.

**REFERENCES**

1. **Silberschatz, A., Korth, H. F., & Sudarshan, S.** (2010). *Database System Concepts* (6th ed.). McGraw-Hill.
2. **Ullman, J. D., & Widom, J.** (2008). *A First Course in Database Systems*. Pearson Education.
3. **XAMPP by Apache Friends**. (n.d.). Official Documentation and Downloads. Retrieved from: <https://www.apachefriends.org/>
4. **MySQL**. (n.d.). MySQL Documentation. Retrieved from: <https://dev.mysql.com/doc/>
5. **PHP Documentation**. (n.d.). Retrieved from: <https://www.php.net/manual/en/>