MAR ATHANASIUS COLLEGE OF ENGINEERING
(Affiliated to APJ Abdul Kalam Technological University, TVM)
KOTHAMANGALAM

Department of Computer Applications

Main Project Report

CampusConnect
Event Registration System

Done by

Akhil Rock Babu

Reg No:MAC23MCA-2010

Under the guidance of

Prof. Biju Skaria

MAR ATHANASIUS COLLEGE OF ENGINEERING
(Affiliated to APJ Abdul Kalam Technological University, TVM)
KOTHAMANGALAM

CERTIFICATE

CampusConnect
Event Registration System

Certified that this is the bonafide record of project work done by

Akhil Rock Babu
Reg No: MAC23MCA-2010

during the final semester, in partial fulfilment of requirements for award of the degree

Master of Computer Applications

of

APJ Abdul Kalam Technological University Thiruvananthapuram

Faculty Guide
Prof. Biju Skaria

Head of the Department
Prof. Biju Skaria

Project Coordinator
Prof. Sonia Abraham

External Examiners

ACKNOWLEDGEMENT

With heartfelt gratitude, I extend my deepest thanks to the Almighty for his unwavering grace and blessings that have made this journey possible. May his guidance continue to illuminate my path in the years ahead.

I am immensely thankful to Prof. Biju Skaria, Head of the Department of Computer Applications, and my project guide, also Prof. Sonia Abraham, our dedicated project coordinator, for their invaluable guidance and timely advice, which played a pivotal role in shaping this project. Their guidance, constant supervision, and provision of essential information were instrumental in the successful completion of the mini project.

I extend my profound thanks to all the professors in the department and the entire staff at MACE for their unwavering support and inspiration throughout my academic journey. My sincere appreciation goes to my beloved parents, whose guidance has been a beacon in every step of my path.

I am also grateful to my friends and individuals who generously shared their expertiseand assistance, contributing significantly to the fulfillment of this endeavor.

ABSTRACT

The CampusConnect - Event Registration System is a product-driven, web-based solution meticulously designed to streamline and automate event management processes for educational institutions. This innovative and customizable platform allows colleges and institutions to purchase the system, tailor the web interface to reflect their unique branding, and harness its extensive suite of features for more efficient event handling. By addressing key challenges associated with tech fests, cultural programs, and academic activities—such as participant registration, event scheduling, and real-time communication—the product enhances operational effectiveness and reduces administrative workload.

Unlike conventional systems that depend on static websites, spreadsheets, or generic registration forms, CampusConnect introduces advanced features, including a Dynamic Form Builder that empowers event organizers to design tailored registration forms based on the specific needs of different events. This dynamic system simplifies participant registration, facilitates profile management, streamlines event schedules, and incorporates secure fee payment functionality. Additional features, such as ticket generation with QR codes and real-time notifications via email, enhance the user experience for both organizers and participants. The adaptable interface further allows institutions to deliver a personalized and professional digital experience.

Built using modern web technologies such as HTML, CSS, JavaScript, PHP, and MongoDB, CampusConnect ensures a responsive, interactive frontend, robust server-side logic, and optimized data storage and management. The system's scalability supports multi-event handling, enabling organizers to manage multiple events concurrently with ease. By offering a flexible, centralized, and feature-rich product, CampusConnect empowers educational institutions to modernize their event management processes, reduce inefficiencies, and create a seamless and engaging experience for all users.

LIST OF TABLES

LIST OF FIGURES

CONTENTS

1. INTRODUCTION

The CampusConnect - Event Registration System is a web-based application developed as a product to streamline and automate the management of events hosted by colleges and educational institutions. This product aims to replace traditional, manual methods of event registration and coordination with a centralized, efficient, and customizable digital solution. Events conducted as part of tech fests, cultural programs, and other academic activities often face challenges such as managing participant registrations, organizing event schedules, and ensuring seamless communication. The need for an interactive, user-friendly, and adaptable system is essential for enhancing the overall experience of event management while reducing administrative workload.

The existing system for event management in most institutions relies on basic websites, spreadsheets, or Google Forms. These static websites primarily function as informational portals, offering only limited functionalities, such as event details, contact information, and basic registration forms. They lack the capability of a Dynamic Form Builder, which is essential for creating tailored forms that meet the specific requirements of different events. This limitation makes it difficult to collect varied participant information. Furthermore, these traditional systems do not provide customization options for the web interface, which prevents institutions from delivering a personalized and professional experience. Communication with participants is often fragmented, leading to delays and inefficiencies in managing event updates and overall coordination.

The proposed CampusConnect - Event Registration System is a comprehensive, product-driven web-based platform designed to streamline event management for any educational institution that purchases and uses the product. Catering to administrators, organizers, and participants, it offers features like online registration, profile management, and event customization, making it highly adaptable to institutional needs. One of its key features is the Dynamic Form Builder, allowing organizers to create tailored forms to collect event-specific information, addressing the limitations of traditional systems. The interface is customizable, enabling institutions to align the platform with their branding and functional requirements. Participants benefit from simplified registration, profile updates, and access to event history, while organizers can efficiently manage schedules, categories, and participants. Additionally, the system supports secure fee payment integration, real-time communication via email, and ticket generation with QR codes to ensure seamless event operations.

The system is developed using modern web technologies, including HTML, CSS, and JavaScript for an interactive and responsive frontend, PHP for robust server-side logic, and MongoDB for efficient and flexible data storage and management.

2. SUPPORTING LITERATURE

2.1 Literature Review

Paper 1: J. R. V. Jeny, P. Sadhana, B. J. Kumar, S. L. Abhishek and T. Sai Chander, "A Web based-College Event Management System and Notification Sender," 2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2022, pp. 1434-1438, doi: 10.1109/ICIRCA54612.2022.9985774.

Event management systems (EMS) have evolved from manual processes to automated web-based solutions, improving efficiency, user experience, and security. Various studies have explored different approaches to enhance EMS functionality. Some researchers have introduced barcode-based validation for secure participant verification, while others have implemented background authentication codes to improve security. Automated scheduling, notification alerts, and real-time data management have also been incorporated to reduce the burden on organizers.

The Model-View-Controller (MVC) architecture is widely used in EMS for its modular design, improving maintainability and scalability. SQL-based databases such as MySQL and SQLyog ensure efficient data handling, while technologies like Java Server Pages (JSP) and Servlets enable dynamic web interactions.

Security remains a key focus, with researchers highlighting the importance of encryption techniques and authentication mechanisms. Performance optimization through real-time data processing methods has also been explored to enhance system responsiveness.

Future advancements in EMS may include AI-driven analytics for event forecasting, blockchain-based security for fraud prevention, and cloud integration for global accessibility. The continuous evolution of EMS is expected to further streamline event management, making it more efficient and user-friendly.

Paper 2: R. F. Olanrewaju, B. U. I. Khan, M. A. Morshidi, F. Anwar and M. L. B. M. Kiah, "A Frictionless and Secure User Authentication in Web-Based Premium Applications," in IEEE Access, vol. 9, pp. 129240-129255, 2021, doi: 10.1109/ACCESS.2021.3110310.

User authentication is a critical aspect of web-based applications, ensuring security while maintaining a seamless user experience. Traditional authentication mechanisms, such as username-password combinations, have proven vulnerable to attacks like phishing and brute force attempts. To address these challenges, researchers have explored advanced authentication solutions, integrating multi-factor authentication (MFA), biometric verification, and behavior-based authentication models.

Frictionless authentication has emerged as a promising approach to enhance security without disrupting the user experience. Studies highlight the effectiveness of behavior-based profiling, which leverages login patterns, device characteristics, and geolocation data to assess user authenticity. Risk-based authentication mechanisms dynamically adjust security measures based on contextual factors, reducing reliance on static credentials.

Several research efforts focus on multi-factor authentication systems, incorporating OTP verification, biometric data, and cryptographic tokens to strengthen access control. AI-driven authentication methods utilize machine learning algorithms to detect anomalies and unauthorized access attempts, enhancing system resilience against cyber threats.

Future advancements in authentication systems are expected to integrate blockchain-based identity management, federated authentication across platforms, and AI-driven fraud detection techniques. As web applications continue to evolve, secure and user-friendly authentication mechanisms will remain a key priority in ensuring data protection and accessibility.

Paper 3: K. T. Fathimath Hanan, T. Henna Sherin, U. Lulu Shadin, K. C. Dilshad and A. P. Sulthana Rinsy, "Streamlined Application for Managing College Events," 2024 International Conference on Innovations and Challenges in Emerging Technologies (ICICET), Nagpur, India, 2024, pp. 1-6, doi: 10.1109/ICICET59348.2024.10616323.

Event management in educational institutions has transitioned from traditional manual coordination to sophisticated web-based platforms that streamline planning, execution, and evaluation. Various research studies have explored how technological advancements have improved efficiency, security, and user experience in event management systems.

Several studies have focused on enhancing user accessibility and engagement by integrating modern web technologies. Many event management systems utilize Flutter for front-end development due to its cross-platform compatibility and user-friendly interface, while Python and Django are often employed for back-end functionalities to manage data flow and system logic. Real-time databases such as Firebase and MySQL enable seamless data storage, quick retrieval, and instant updates, ensuring efficient event registration, tracking, and notifications.

Security is a key aspect of event management systems. Researchers have explored different authentication techniques, including OTP-based login verification and encrypted data storage, to protect user information and prevent unauthorized access. These security measures enhance trust and reliability, ensuring data confidentiality and integrity.

Automation plays a crucial role in modern event management systems. Features such as automated scheduling, real-time notifications, and event evaluation modules have been developed to reduce manual effort and improve the overall event experience. Some studies have also introduced audience polling mechanisms, allowing participants to provide real-time feedback, thus enhancing engagement and decision-making.

Additionally, artificial intelligence (AI) and big data analytics are being leveraged to optimize event recommendations and attendance predictions. These technologies analyze past event data and user preferences to provide personalized event suggestions, helping organizers improve participation and resource allocation.

Future advancements in event management systems are expected to focus on integrating blockchain technology for credential verification, IoT-enabled real-time tracking for better logistics management, and AI-driven automation for smart decision-making. As educational institutions continue to embrace digital transformation, event management systems will further evolve to provide a more secure, efficient, and engaging platform for organizing college events.

2.2 Literature Review Summary

Web-based event registration and management systems have evolved with advancements in automation, security, and real-time data processing. Various studies have explored different methodologies to enhance efficiency, streamline user authentication, and improve event coordination. Several key features and functionalities in modern event registration systems are drawn from these studies, ensuring seamless event organization, secure user authentication, and enhanced user experience.

Jeny et al. (2022) present a web-based college event management system that automates event registration, scheduling, and notifications while incorporating authentication codes for security. The implementation of an automated event approval and rejection mechanism enables administrators to efficiently review and manage events. The study also highlights the use of the Model-View-Controller (MVC) architecture for a modular and scalable design, ensuring better system maintainability. Additionally, SQL-based databases are utilized for efficient data storage and retrieval, a principle that aligns with the structured storage and management of event-related information.

Fathimath Hanan et al. (2024) emphasize real-time updates, enhanced user engagement, and authentication in event management systems. The integration of real-time databases facilitates instant event tracking and notifications, ensuring up-to-date participant information. Dynamic event registration systems benefit from customizable form builders, allowing organizers to tailor event-specific registration forms based on requirements. Security measures such as OTP-based authentication enhance access control, providing a secure environment for users to verify their registration status and access event-related details.

Olanrewaju et al. (2021) focus on frictionless and secure authentication mechanisms, highlighting multi-factor authentication, AI-driven fraud detection, and behavior-based profiling. While AI-based authentication enhances security, essential security principles such as email OTP verification and role-based access control play a crucial role in preventing unauthorized access. These measures contribute to a secure event management system, ensuring that sensitive participant and event information is well protected.

The adoption of these research-backed features enables the development of an efficient and secure event registration and management system. The integration of automated approval processes, dynamic event creation, real-time updates, and secure authentication mechanisms enhances usability and operational efficiency, providing a streamlined experience for administrators, organizers, and participants

2.3 Findings and Proposals

The literature review on modern web-based event registration systems highlights key trends and research-backed findings to enhance the "CampusConnect" project. One major finding is the role of automation in improving event management efficiency. Studies by Jeny et al. (2022) emphasize automated event approval, rejection, and notifications, which streamline administrative tasks and reduce manual intervention. Implementing these features in "CampusConnect" would enhance operational efficiency and user experience.

Another key trend is adopting the Model-View-Controller (MVC) architecture, which improves system modularity, scalability, and maintainability. This architecture, as highlighted by Jeny et al. (2022), ensures seamless future upgrades and allows easy integration of new functionalities. To align with this, "CampusConnect" should adopt the MVC architecture for better scalability.

User authentication and security are also critical. Olanrewaju et al. (2021) emphasize multi-factor authentication (MFA) and role-based access control (RBAC) to prevent unauthorized access. Implementing OTP-based authentication and RBAC in "CampusConnect" would strengthen system security by restricting access based on predefined user roles (e.g., admins, organizers, participants).

Real-time data processing enhances user engagement, as highlighted by Fathimath Hanan et al. (2024). Incorporating real-time tracking and notifications through technologies like Firebase would improve participant interaction and keep users updated on event details.

Customizable registration forms are essential for flexibility, allowing organizers to tailor forms to event-specific needs. Integrating a jQuery-based form builder would enhance adaptability for diverse events, such as workshops, fests, and competitions.

Future enhancements for "CampusConnect" include AI-driven event recommendations to improve personalization, blockchain-based certificate generation for secure digital certificates, and advanced dashboards for event analytics. These features would further enhance usability, security, and data-driven decision-making, aligning "CampusConnect" with evolving trends in event management systems.

.

3. SYSTEM ANALYSIS

3.1 Module Description

Organizer Management Module:

- Organizers can self-register on the platform by providing necessary details such as name, email, contact number, and organization details.
- Registrations are subject to approval by institutional admins. Once approved, organizers gain access to event creation and management features.
- Login credentials are generated during registration, enabling organizers to log in securely.
- Organizers can update their profile information, including contact details, organization details, and password.
- Organizers can add and manage Co-organizers for each event, enabling collaborative event management.

Event Management Module:

- Approved organizers can create and manage events, including adding event titles, descriptions, schedules, and venue information.
- A Dynamic Form Builder allows organizers to create customizable registration forms tailored to each event's specific needs (e.g., optional fields).
- Organizers can manage participant registrations, track registration statuses (e.g., pending, approved), and generate event-related reports such as participant lists, and payment summaries.
- Enables organizers to send real-time email notifications and updates to registered participants.
- Organizers can define event-specific rules (e.g., eligibility criteria, registration limits) to streamline event participation.

Participants Management Module:

- Participants can register for single or multiple events via a user-friendly interface that displays event details, categories, and registration deadlines.
- Participants are notified about event-related updates in real time through email.
- Organizers can monitor and manage participant registrations, view registration trends, and track payment statuses.
- Participants can view event fees (if applicable) and make secure online payments through upi qr codes.
- Participants can also validate promo codes to get additional discounts.
- Participants can access and download event tickets or QR codes for seamless entry verification during event check-ins.

Customization Module:

- Institutional admins can customize the platform interface to align with their institution's branding by adding logos, banners, and colour schemes.

Feedback Management Module:

- Participants can provide feedback about events, event organizers, services, or any issues encountered during registration or participation.
- Feedback can be collected through feedback forms.
- Enables admins to review feedback-related issues, promoting a better user experience.

Admin Management Module

- Institutional admins manage platform users, including approving or rejecting organizer registrations and monitoring participant activities.
- Admins oversee event approval processes, ensuring that events adhere to institutional guidelines before being published on the platform.
- Admins can generate platform-wide reports, including event statistics, user engagement metrics, and payment summaries, to monitor overall system performance.
- Admins can handle announcements, such as event promotions using push notifications.

Report Management Module

- Admins and Organizers can generate Reports on Pariticipant Data as well as Income Report.
- Admins, Organizers and Co-Organizers can view a Quick Summary on all their Events in Dashboard.

3.2 Feasibility Analysis

A project feasibility study is an essential analysis that determines whether a proposed project is viable in terms of technical, economic, and operational factors. It helps assess whether the project can be completed within the defined scope, timeline, budget, and available resources. A feasibility study also highlights potential challenges and ensures that the project can be executed successfully before substantial resources are allocated.

3.2.1 Technical Feasibility

The CampusConnect - Event Registration System is technically feasible due to the use of a reliable and scalable technology stack that ensures efficiency, seamless integration, and real-time performance. The frontend is developed using HTML, CSS, JavaScript, and Bootstrap, providing a responsive, user-friendly, and mobile-first interface. These technologies enhance accessibility and user experience by enabling intuitive navigation, dynamic content rendering, and interactive form submissions for admins, organizers, participants, and co-organizers.

The backend is powered by PHP, a widely used server-side scripting language that efficiently handles core functionalities such as user authentication, event management, organizer approvals, and participant registration. PHP's flexibility and compatibility with various databases make it ideal for managing server-side logic, session handling, and real-time email notifications. The backend processes are structured to ensure secure data handling and quick response times, improving the overall system performance.

The database management is implemented using MongoDB, a highly scalable NoSQL database that effectively handles semi-structured and unstructured data, such as event details, participant registrations, and interface customization records. MongoDB's schema-less design allows for easy modification of data, supporting dynamic event configurations and future scalability. Its indexing and querying features enable fast retrieval of participant data, event registration limits, and approval statuses, ensuring real-time updates and efficient system functioning.

To further enhance functionality, the system integrates real-time email notifications using SMTP protocols. Notifications are automatically triggered for event approvals/rejections, participant registration updates, and critical system alerts, keeping all stakeholders promptly informed. The system also includes OTP-based authentication during critical processes, such as login and ticket verification, to enhance security and prevent unauthorized access.

A key technical feature of the CampusConnect system is the dynamic form builder powered by jQuery, which allows organizers to create event-specific registration forms tailored to their requirements. This flexibility improves usability and adaptability for various event types, including tech fests, workshops, and cultural programs.

By leveraging modern web technologies, a responsive design, and robust backend processing, the CampusConnect - Event Registration System ensures technical feasibility, scalability, and reliability, making it well-suited for real-world deployment in educational institutions. This well-integrated system is designed to handle large volumes of event registrations, streamline the approval process, and provide seamless interaction for users, ultimately delivering a smooth and efficient event management experience.

3.2.2 Economic Feasibility

The CampusConnect - Event Registration System is economically feasible due to its reliance on cost-effective, open-source technologies that minimize development and operational expenses. The system is built using widely adopted technologies, including HTML, CSS, JavaScript, PHP, jQuery, and MongoDB, all of which are open-source, eliminating the need for costly software licenses. This significantly reduces initial setup and development costs, making it an affordable solution for educational institutions seeking a comprehensive event management platform.

The economic feasibility is further enhanced by the availability of skilled developers familiar with PHP, MongoDB, and frontend technologies. These technologies are widely used in the web development community, ensuring that experienced developers can be hired at competitive rates. This availability of expertise accelerates the development process, reduces hiring costs, and facilitates long-term maintenance of the system.

Another major contributor to cost-effectiveness is automation. Key features such as automated event approvals/rejections, dynamic form creation, and email notifications reduce manual intervention, thereby minimizing administrative workload and associated labor costs. By automating participant registration, event updates, and ticket generation, the system optimizes resource utilization and improves overall efficiency, resulting in cost savings for administrators and event organizers.

Additionally, the system’s ability to support real-time data updates and notifications minimizes the need for manual follow-ups, streamlining event coordination and reducing communication-related expenses. The dynamic event management system allows organizers to customize registration forms and manage events without the need for constant technical support, further reducing operational costs.

Although the initial costs of development, hosting, and deployment may be present, the long-term savings and efficiency gains justify the investment. By enhancing workflow efficiency, reducing errors, and improving user satisfaction, the CampusConnect - Event Registration System proves to be a financially sustainable solution that benefits both institutions and users. The system’s scalability and adaptability ensure that it can continue to meet the growing needs of educational institutions without incurring excessive additional expenses, making it an economically sound investment in the long run.

3.2.3 Operational Feasibility

The CampusConnect - Event Registration System is highly feasible from an operational perspective, as it enhances event management efficiency, streamlines registration processes, and fosters smooth collaboration between administrators, organizers, participants, and co-organizers. The user-friendly frontend developed using HTML, CSS, JavaScript, and jQuery, ensures an intuitive interface that is easy to navigate for all user roles. Organizers can effortlessly create events, customize registration forms, and manage participants, while institutional admins can oversee the platform and approve or reject events in real-time.

The backend, built using PHP and MongoDB, handles core functionalities such as user authentication, event data management, and participant tracking. This robust infrastructure supports real-time data processing, allowing users to access up-to-date information about live events, registration statuses, and ticket generation. Automated approval mechanisms, dynamic notifications, and real-time event status updates minimize manual interventions, reducing delays and ensuring smooth operations for all stakeholders.

A key operational strength lies in the system's dynamic form builder, which allows organizers to create event-specific registration forms tailored to various event requirements, such as workshops, cultural fests, and technical competitions. This flexibility improves operational efficiency by allowing seamless customization without the need for technical support. Additionally, the integration of automated email notifications keeps participants informed about registration approvals, event reminders, and updates, further enhancing communication efficiency.

The system also ensures efficient participant verification and event entry management through QR code-based ticket generation. Participants can check their registration status, download tickets, and view event details directly from the platform, reducing the workload on organizers and eliminating bottlenecks at event venues.

3.3 System Environment

3.3.1 Software Environment

- Frontend:

  HTML, CSS, JavaScript, and jQuery are used for developing the web application, providing an intuitive user interface and smooth interaction for admins, organizers, participants, and co-organizers.

- Backend:

  PHP is utilized for efficient server-side logic, handling dynamic content, managing user requests, and ensuring secure communication with the database.

- Database:

  MongoDB is chosen for scalable and flexible NoSQL-based data management to store event details, user profiles, participant registrations, feedback, and customization configurations.

- Server OS:

  Apache Server running on Windows OS (via XAMPP) is used for local hosting and deployment, ensuring reliability and performance.

- Development Tools:
  - Visual Studio Code (VS Code) for frontend and backend development due to its lightweight and extensible features.
  - XAMPP for setting up a local development environment and running PHP-based server scripts.
  - Git/GitHub for version control and maintaining project files systematically.

3.3.2 Hardware Environment

- Client Devices:

  - Smartphones, tablets, and PCs with standard browsing capability.

  - Processor: Dual-core processor or higher.

  - RAM: 4GB or more to ensure smooth web application access.

  - Storage: 500MB minimum for storing temporary cache and application data.

- Server Requirements:

  - Processor: Quad-core processor or higher for handling multiple requests concurrently.

  - RAM: 8GB minimum (16GB recommended) to support real-time operations and user sessions.

  - Storage: 1TB storage with SSD to improve read-write speed and data retrieval.

  - Internet Connectivity: Reliable high-speed internet connection to support continuous system uptime and smooth data transfer between clients and the server.

- Networking:

  - A secure, high-speed network to facilitate communication between client devices and the server, ensuring minimal latency and secure data exchange.

3.4 Actors & Roles

Institution Administrator

- Oversees the entire system and manages user access for organizers.
- Sets up institution-specific branding and customization options.
- Monitors event schedules, registrations, and overall system performance.
- Generates detailed reports on event participation, and ticket sales.
- Send promotional notifications to bulk emails simultaneously.

Organizer

- Creates and manages event details, including schedules, and venues.
- Utilizes the Dynamic Form Builder to create tailored registration forms for specific events.
- Manages participant registrations and assigns roles to co-organizers.
- Sends notifications or updates to participants via email.
- Tracks registration payments and generates invoices for fee-based events.
- Oversees ticket generation and distribution with QR codes for entry verification.

Co-Organizer

- Assists organizers in managing event details and registrations.
- Handles participant queries and provides necessary support.
- Manages on-site event logistics, including attendance tracking and entry verification.
- Monitors ticket verification and ensures smooth participant entry at venues.

Participant

- Registers for single or multiple events through a user-friendly interface.
- Updates and manages their profile details.
- Receives event-related updates via email.
- Pays event fees securely through integrated payment methods.
- Accesses tickets with QR codes and tracks their event history.
- Provides feedback on events and services to improve the overall experience

4. SYSTEM DESIGN

4.1 Use Case Model

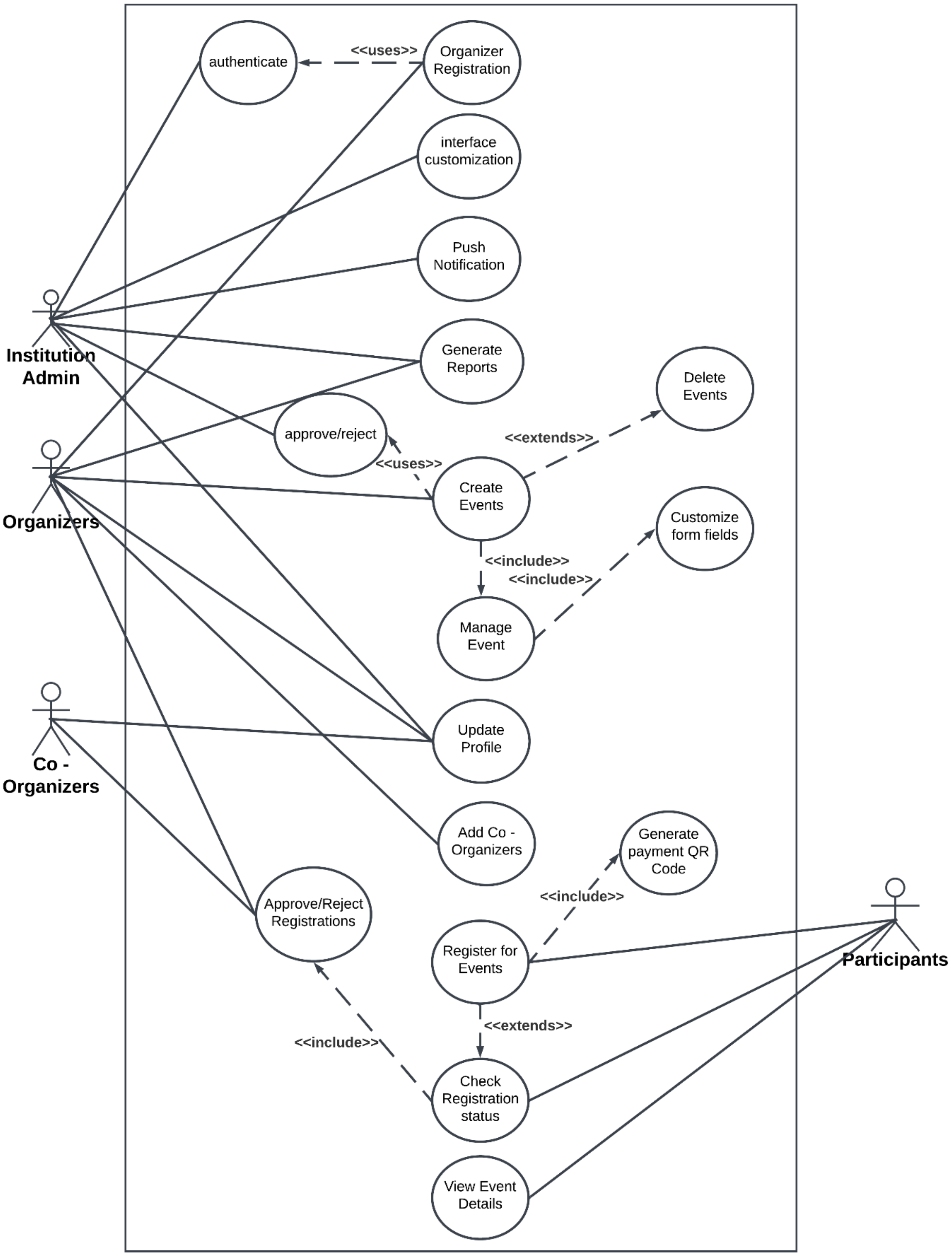


Figure 4.1 Use Case Diagram

The use case diagram illustrates the interactions between various actors and the specific actions they can perform within the CampusConnect - Event Registration System. Four key actors are identified: Institution Admin, Organizers, Co-Organizers, and Participants. Each actor plays a distinct role in managing and engaging with the system’s functionalities.

Within the diagram, several use cases are represented by ovals, each denoting a specific function that an actor can execute. These functionalities cover a range of essential operations, including authentication, organizer registration, event creation, interface customization, push notifications, and report generation. Additional functionalities, such as managing event details, checking registration status, and generating QR codes, are also highlighted to enhance user engagement and streamline event-related processes.

The diagram also showcases the connections between actors and their respective use cases through direct lines:

- Institution Admin: This actor can manage core system functionalities, including authenticating users, approving or rejecting organizers, generating reports, sending push notifications, and customizing the interface.

- Organizers: Responsible for creating events, managing event details, and adding co-organizers to assist in event operations. They can also approve or reject event registrations submitted by participants.

- Co-Organizers: Assigned by organizers to assist in event management and oversee specific event-related tasks.

- Participants: Able to register for events, view event details, check their registration status, and generate payment-related QR codes for easy transaction processing.

The diagram emphasizes the system’s streamlined workflow by showing how actions like event creation lead to subsequent processes such as event customization and participant registration. This interconnected flow enhances the system’s efficiency, minimizes operational delays, and ensures that users can perform their tasks seamlessly.

Overall, the use case diagram provides a comprehensive view of the CampusConnect - Event Registration System's functionalities, helping stakeholders understand the interactions between different users and actions within the system. This visual representation aids in effective communication, system design, and requirement analysis during the development process.

4.2 Activity Diagram

4.2.1 Event Creation

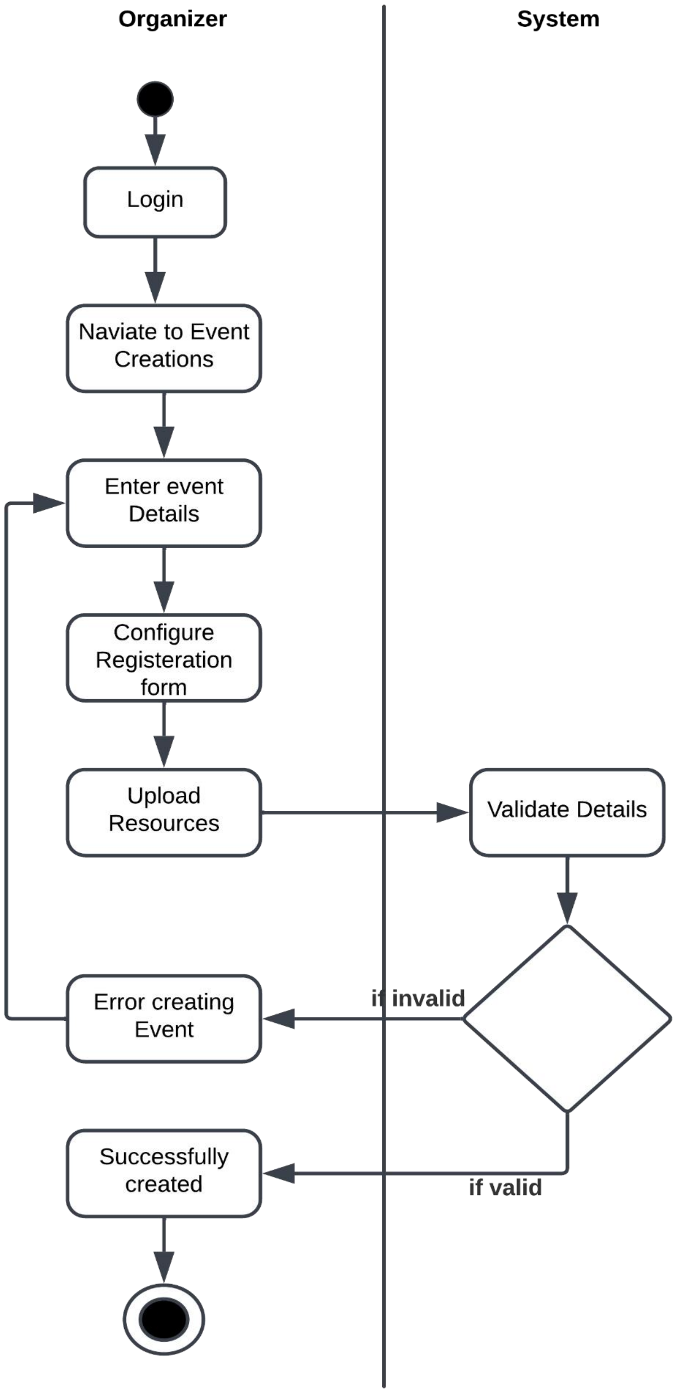


Figure 4.2 Activity Diagram for Event Creation

This activity diagram illustrates the event creation process in the CampusConnect - Event Registration System. It begins with the organizer logging in and navigating to the event creation section, where they input event details, configure the registration form, and upload necessary resources. The system then validates the entered information. If the input is valid, the event is successfully created; otherwise, the organizer is notified to address any errors before proceeding. This streamlined flow ensures efficient event setup with proper validation and error handling.

4.2.2 Participant Event Registration

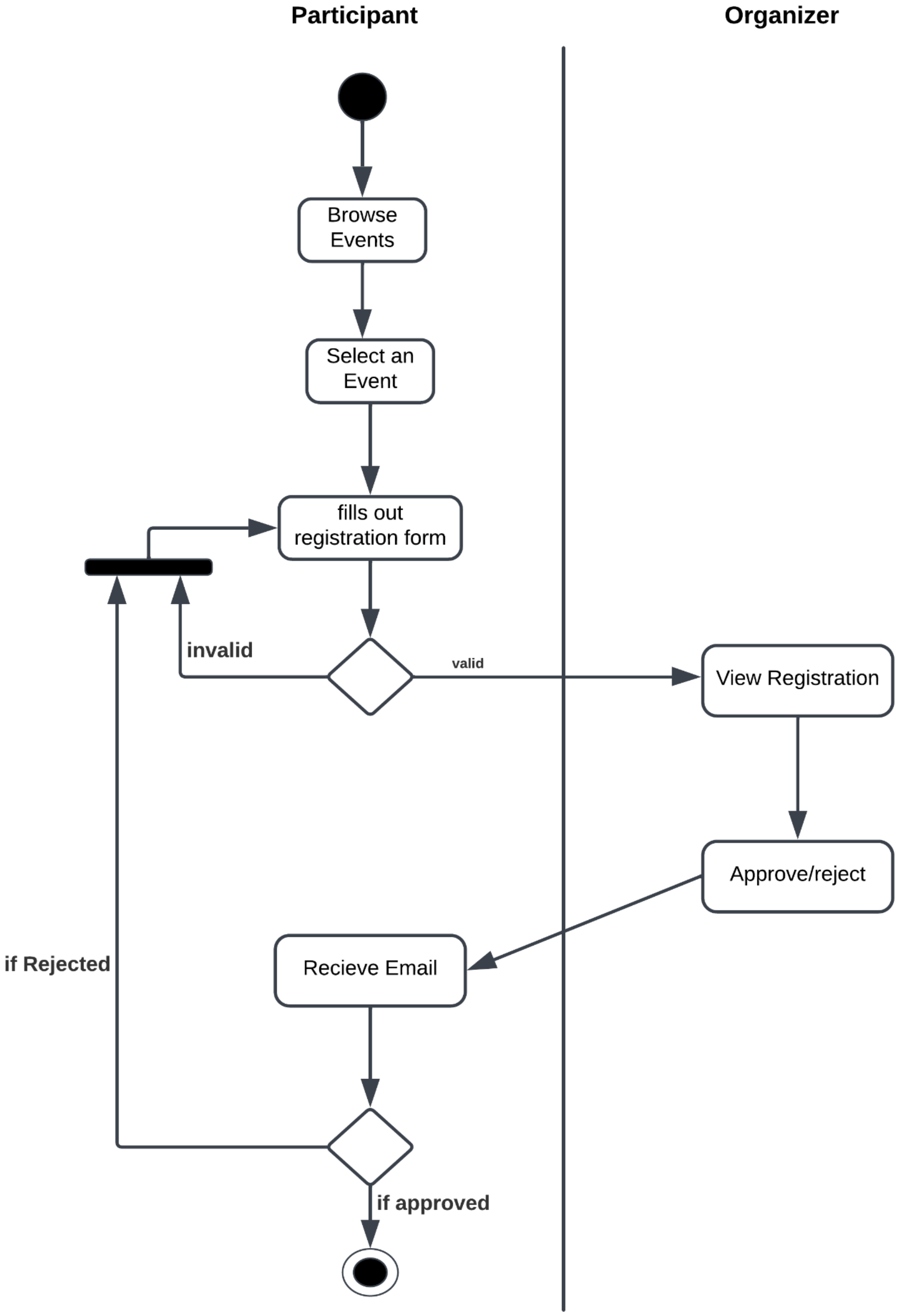


Figure 4.3 Activity Diagram for Event Registration

Event Registration System. It starts with the participant browsing available events and selecting a specific event. After selecting, the participant fills out the event registration form. The system then checks the validity of the form. If invalid, the user must resubmit it. If valid, the organizer views the registration and decides to approve or reject it. If approved, the participant receives a confirmation email and successfully completes the registration process. If rejected, the participant is notified accordingly.

4.3 Sequence Diagram

The sequence diagram serves as a visual representation of the chronological order of interactions among the system's components, providing a clear understanding of the process flow and communication pathways within the system.

4.3.1 Participant Event Registration

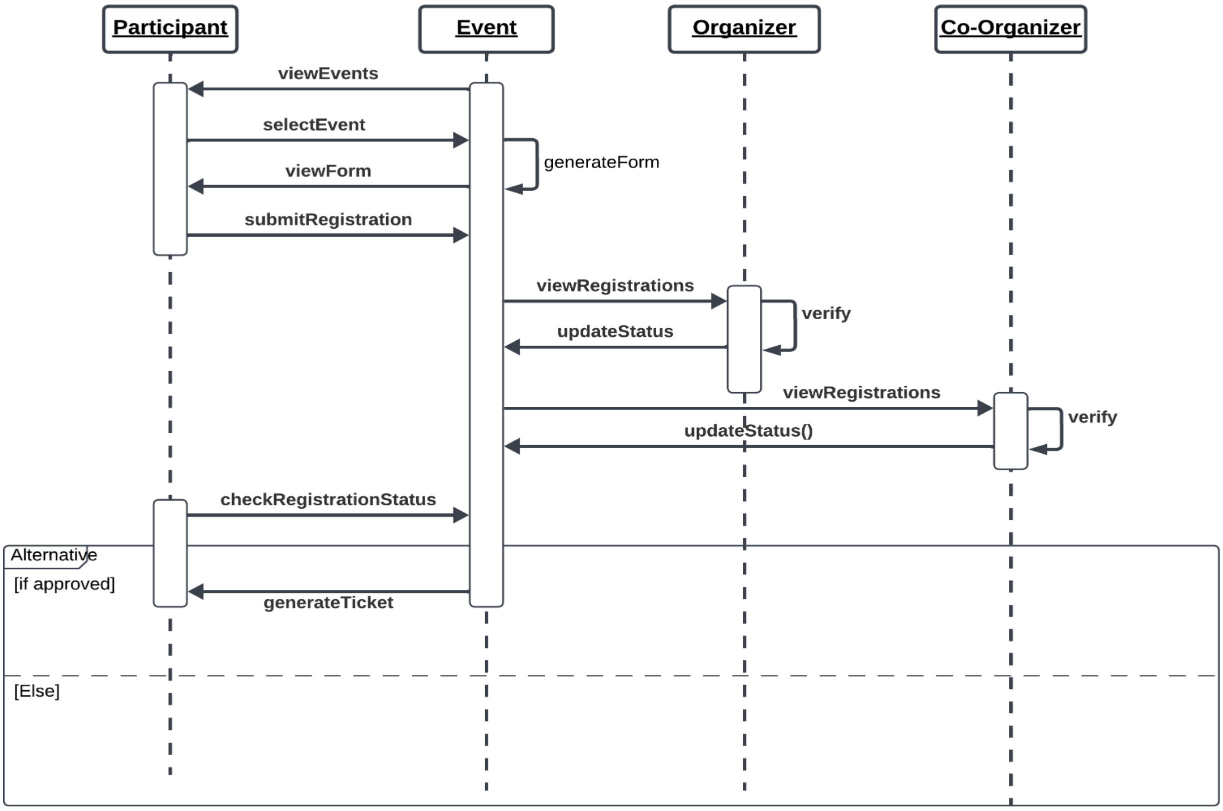


Figure 4.4 Sequence Diagram for Event Registration

This sequence diagram represents the event registration and approval workflow in the CampusConnect - Event Registration System. The process begins with a participant viewing events, selecting one, and submitting the registration form. The system verifies the form, and the event organizer reviews the registrations. The organizer can approve or reject the registration, and co-organizers may also verify and update the status. Once approved, the participant checks the registration status and receives a ticket for event entry. If not approved, alternative actions are triggered based on the status update.

4.3.2 Organizer Registration

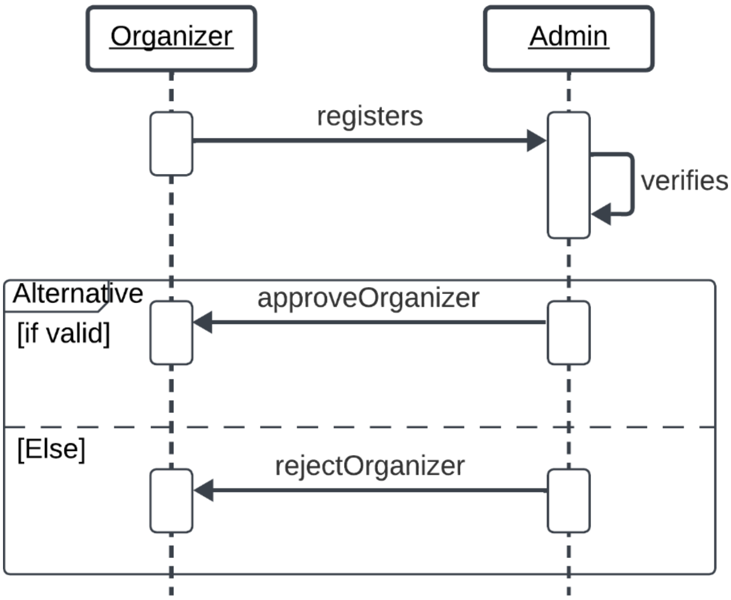


Figure 4.5 Sequence Diagram for Organizer Registration

This sequence diagram outlines the organizer approval process in the CampusConnect - Event Registration System. It begins with an organizer registering in the system. The admin receives the registration request and verifies the organizer's details. This flow ensures that only verified organizers can proceed to create and manage events in the system.

4.3.3 Interface Customization

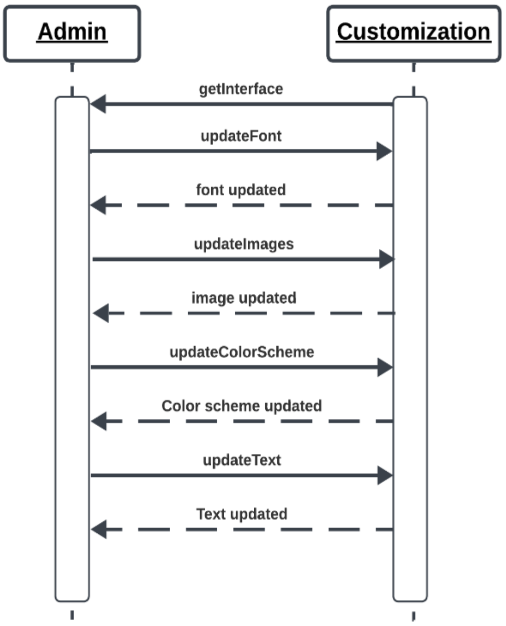


Figure 4.6 Sequence Diagram for Interface Customization

This sequence diagram outlines the customization functions for Institution Admin to meet their own needs and branding style. The Institution Admins can change the font, font size, text colour, and the text itself. Institution Admins can also view the changes made in real time.

4.4 List of Identified classes, attributes, and their relationships

Classes are an important mechanism for classifying objects. The primary role of a class is to define the attributes, methods, and applicability of its instances

4.4.1 Identified Classes

- Users
- Admin
- Organizers
- Co-Organizers
- Participants
- Events
- Customization
- Payments
- Feedback

4.4.2 Identified Attributes

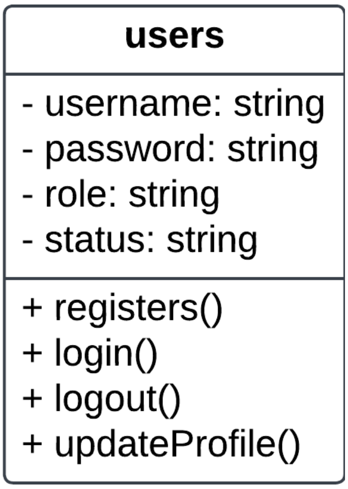


Figure 4.7 Snapshot of User Class

The User class serves as the foundation for various user types within the system, including Institutional Administrators, Organizers and Co-Organizers encapsulating common attributes such as username, password, role and status, along with essential methods like registers(), login(), logout() and updateProfile().

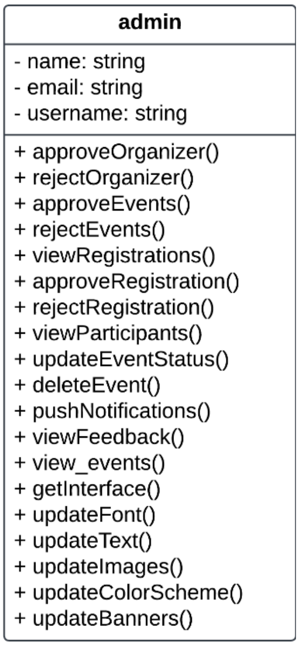


Figure 4.8 Snapshot of Admin Class

The Admin class represents the Institutional Administrator, having all the important methods like approve/reject Organizers, approve/reject Events, Customization methods to manage the overall functioning of the System.

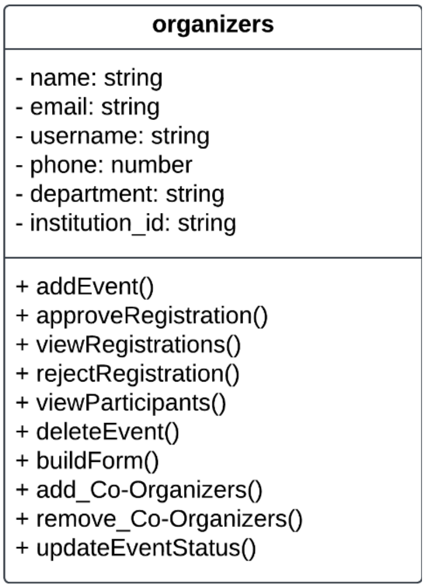


Figure 4.9 Snapshot of Organizer Class

The Organizers class is tailored for Organizers, offering functionalities such as add Event, View Registrations for each events, approve/reject registrations, update the Event Status, Delete Event, and Design the Registration forms for Events.

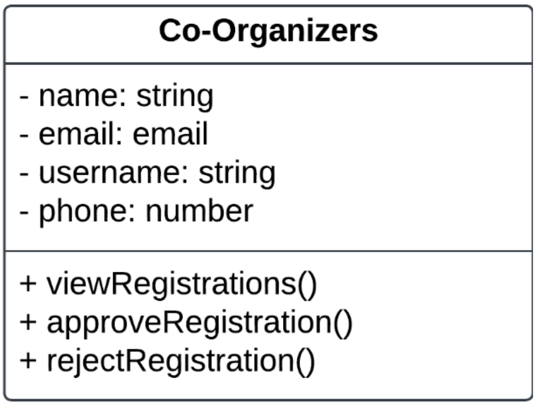


Figure 4.10 Snapshot of Co-Organizer Class

Co-Organizer class represents the Co-Organizers who are added by the Organizers for better Collaboration. Therefore Co-Organizers have only limited functionalities upon assigned events such as View Registrations and approve/reject Registrations.

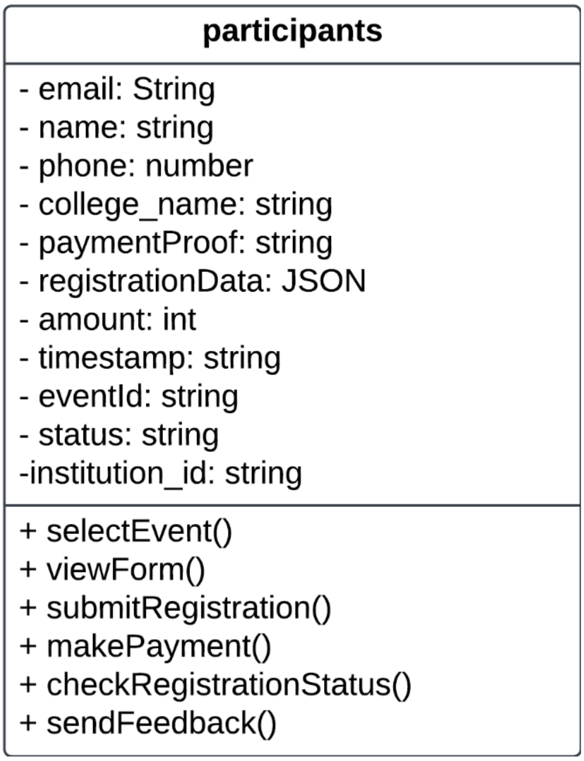
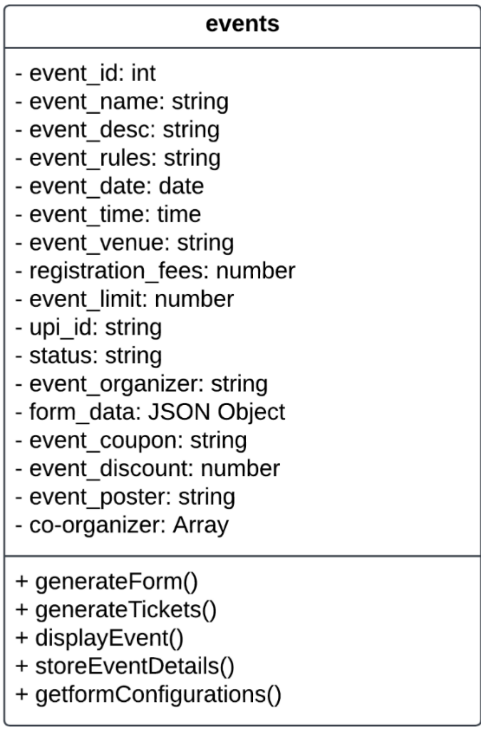


Figure 4.11 Snapshot of Participant Class

The Participants class represents the participants who can register for an event based on their preference. They can get instant, real-time updates about the registration via email and also can check for the status along with viewing and downloading tickets for successful event registrations. Additionally, they can also provide feedback to institution admins about the events.



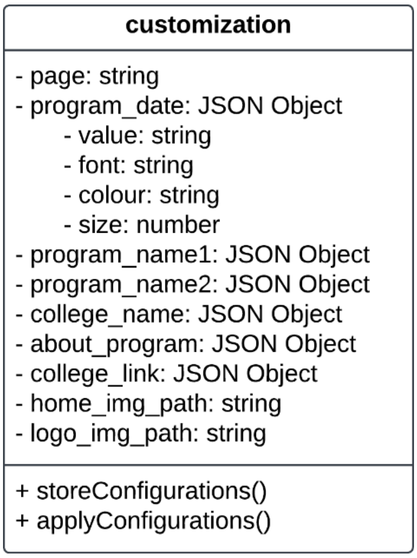
events

- event_id: int
- event_name: string
- event_desc: string
- event_rules: string
- event_date: date
- event_time: time
- event_venue: string
- registration_fees: number
- event_limit: number
- upi_id: string
- status: string
- event_organizer: string
- form_data: JSON Object
- event_coupon: string
- event_discount: number
- event_poster: string
- co-organizer: Array

+ generateForm()
+ generateTickets()
+ displayEvent()
+ storeEventDetails()
+ getformConfigurations()

Figure 4.12 Snapshot of Participant Class

The Events class serves as a container for information pertaining to various events created by Organizers within the system. It encompasses essential attributes such as event name, event date, event time, event limit, event fees, etc. The class offers a method to generate an event registration form from the form data stored as a JSON object for participants to register. The event details stored here are used to display on event pages so that participants can view and register for them.



customization

- page: string
- program_date: JSON Object
    - value: string
    - font: string
    - colour: string
    - size: number
- program_name1: JSON Object
- program_name2: JSON Object
- college_name: JSON Object
- about_program: JSON Object
- college_link: JSON Object
- home_img_path: string
- logo_img_path: string

+ storeConfigurations()
+ applyConfigurations()

Figure 4.13 Snapshot of Customization Class

The Customization Class serves as a container to store Website Interface Configuratuions such as the colour schema, text elements, font size and font family. These Stored Configuration allow Institutional Admins to Customize the Website to meet their own requirements and branding needs.

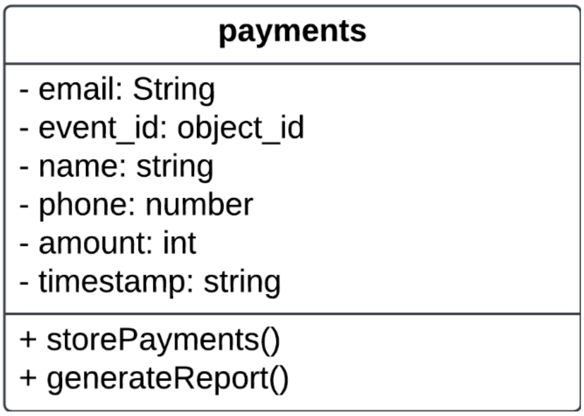


Figure 4.14 Snapshot of Payments Class

The Payments Class serves as a container to store payment information paid by participants at the time of event registration to keep track of the payment activities that happened behind an event. These stored details are also used to generate income reports.

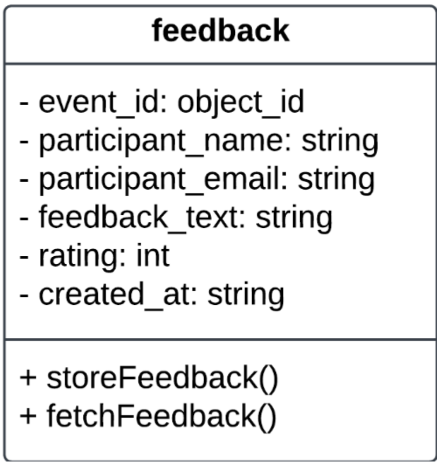


Figure 4.15 Snapshot of Feedback Class

The Feedback Class serves as a container to store feedback submitted by participants. Admins can view feedback to realize where to improve and take decisions using that. Also, this feedback can also be used to create an AI model for sentimental analysis, which gives summaries and decisions from the feedback.

4.4.3 Identified Relationships

Class	Related Class	Association Name	Cardinality
Admin	Organizers	Approves/Rejects	1 - *
Admin	Customization	Manage Customization	1 - *
Organizers	Co-Organizers	Add/Remove	1 - *
Organizers	Events	Creates/Manages	1 - *
Co-Organizers	Events	Creates/Manages	1 - *
Participants	Events	Registers	1 - *
Participants	Payments	Make Payments	1 - *
Participants	Feedback	Submit Feedbacks	1 - *
Admin	Feedback	View Feedbacks	1 - *

Table 4.1 Identified Relationships

4.5 CLASS DIAGRAM

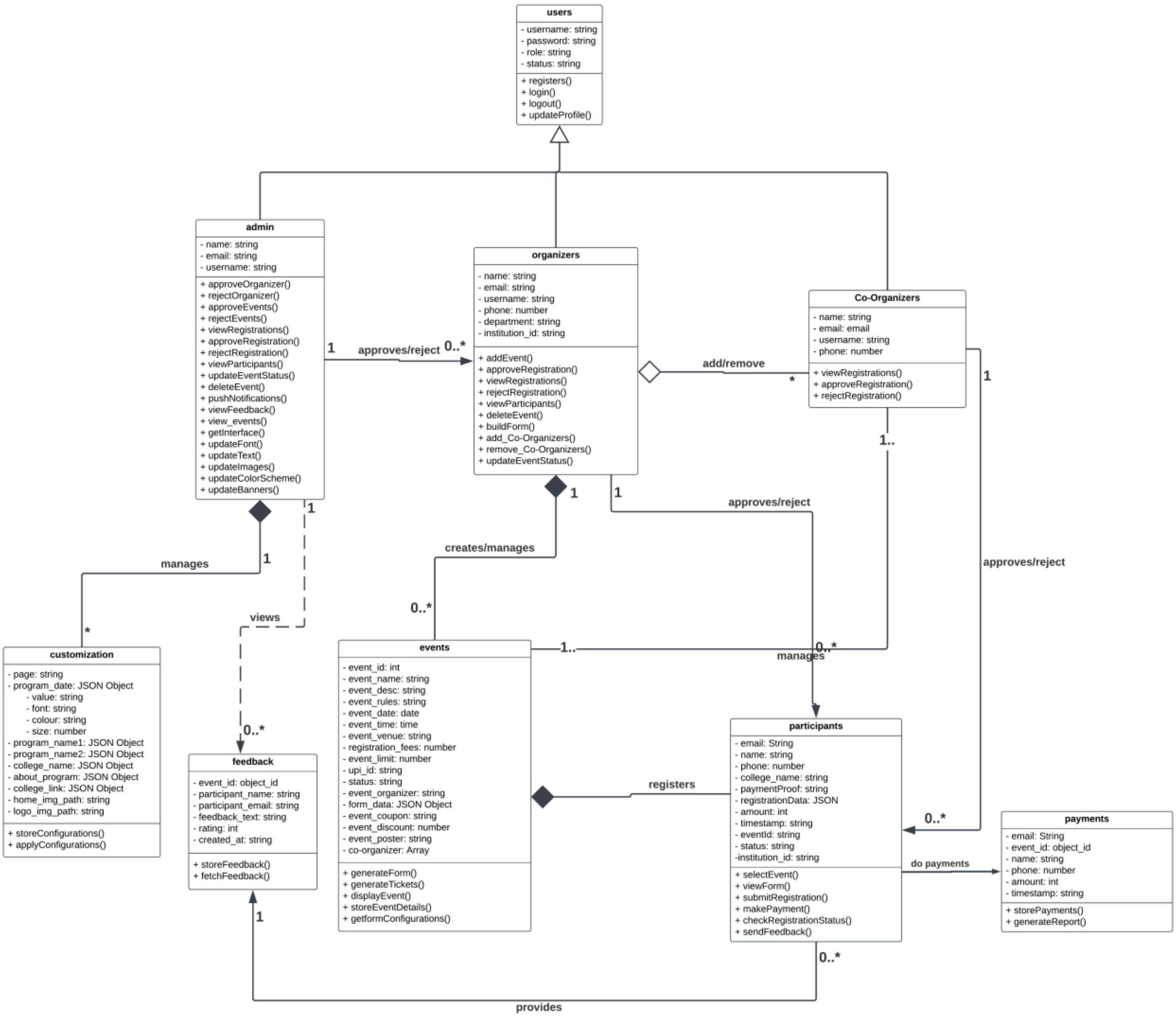


Figure 4.16 Class Diagram

4.6 DATABASE DESIGN

Table name: **User**

Primary Key: **_id, username**

Attributes	**Datatype**	**Constraints**	**Description**
_id	Object ID	Unique	Autogenerated ID
username	String	Unique	Username of the user
password	String		Password to login
role	String		Role of the user
status	String		Status of the user(approved/reject)

Table 4.2 User Table

Table name: **Admin**

Primary Key: **_id, email**

Attributes	**Datatype**	**Constraints**	**Description**
_id	Object ID	Unique	Autogenerated ID
name	String		Name of the Admin
username	String	Ref (Users: username)	Username of the admin
email	String	Unique	Email of admin

Table 4.3 Admin Table

Table name: **Organizers**

Primary Key: **_id, email**

Attributes	Datatype	Constraints	Description
_id	Object ID	Unique	Autogenerated ID
username	String	Ref (Users: username)	Username of Organizer
name	String		Name of the Organizer
email	String	Unique	Email of Organizer
phone	Number		Phone number of Organizer
department	String		Department of Organizer
institution_id	String		Institution Id image path

Table 4.4 Organizers Table

Table name: **Co-Organizer**

Primary Key: **_id, email**

Attributes	Datatype	Constraints	Description
_id	Object ID	Unique	Autogenerated ID
username	String	Ref (Users: _id)	Username of the Co-organizer
name	String		Name of the Co-Organizer
email	String	Unique	Email of Co-Organizer
Phone	Number		Phone number of Co-Organizer

Table 4.5 Co-Organizers Table

Table name: **Events**

Primary Key: **_id**

Attributes	Datatype	Constraints	Description
_id	Object ID	Unique	Autogenerated ID
event_name	String		Name of the event
event_desc	String		Short Description about event
event_rules	String		Rules & Information about event
event_date	Date		Date of event
event_time	Time		Event starting time
event_venue	String		Venue of Event
registration_fees	Number		Registration fees of the event
event_limit	Number		Max no of participants allowed
upi_id	String		Upi ID to receive event fees
status	String		Status of the event(live/hold)
event_organizer	String	Ref (Organizers: username)	Username of Organizer who created the event
form_data	JSON Object		Event Registration form fields
event_coupon	String		Promo Code of the event
event_discount	Number		Discount percentage of event for successful promo code validation
event_poster	String		Img path of event poster
co-organizers	Array	Ref (Co_organizers: username)	Collection of Co Organizer’s username for the event

Table 4.6 Events Table

Table name: **Participants**

Primary Key: **_id, email**

Attributes	**Datatype**	**Constraints**	**Description**
_id	Object ID	Unique	Autogenerated ID
event_id	Object ID	Ref (Events: _id)	Event ID of participant registered event
email	String	Unique	Email of Participant
College_name	String		College Name of the Participant
phone	Number		Phone number of Participant
registration_data			Event Specific Registration data
institution_id	String		Institution Id image path
timestamp	String		Time at which the participant registered
payment_proof	String		Payment Proof image path
status	String		Status of the Event Registration
amount	int		Amount which participant paid

Table 4.7 Participants Table

Table name: **Payments**

Primary Key: **_id**

Attributes	**Datatype**	**Constraints**	**Description**
_id	Object ID	Unique	Autogenerated ID
email	String	Ref (Participants: email)	Email Id of the Payer
event_id	String	Ref (Events: event_name)	Event id for which the payment has done
name	String	Ref (Participants: name)	Name of the Payer
phone	Number	Ref (Participants: phone)	Phone no of the Payer
amount	Int	Ref (Participants: amount)	Amount paid
timestamp	String	Ref (Participants:timestamp)	TimeStamp at which the payment is done

Table 4.8 Payments Table

Table name: **Feedback**

Primary Key: **_id**

Attributes	**Datatype**	**Constraints**	**Description**
_id	Object ID	Unique	Autogenerated ID
event_name	String	Ref (Events: event_name)	Event Name for which the feedback is submitted
participant_name	String	Ref (Participants: name)	Name of the participant
participant_email	String	Ref (Participants: email)	Email id of the participant
feedback_text	String		Feedback
rating	int		Rating provided for the event
submitted_at	String		TimeStamp at which the feedback is submitted

Table 4.9 Feedback Table

Table name: **Customization**

Primary Key: **_id, page**

Attributes	Datatype	Constraints	Description
_id	Object ID	Unique	Autogenerated ID
page	String	Unique	Name of the Customizable page
program_date	JSON Object		Style Attributes of Program Date
program_name1	JSON Object		Style Attributes of Program Name 1
program_name2	JSON Object		Style Attributes of Program Name 2
college_name	JSON Object		Style Attributes of College Name
about_program	JSON Object		Style Attributes of About Program
college_link	JSON Object		Style Attributes of College Link
home_img_path	String		Home Background Image
logo_img_path	String		Logo Image Path

Table 4.10 Customization Table

Attributes of JSON Object datatype which consists of Style attributes such as Font Size, Font Family, color of the element and value of the element.

4.7 UI DESIGN

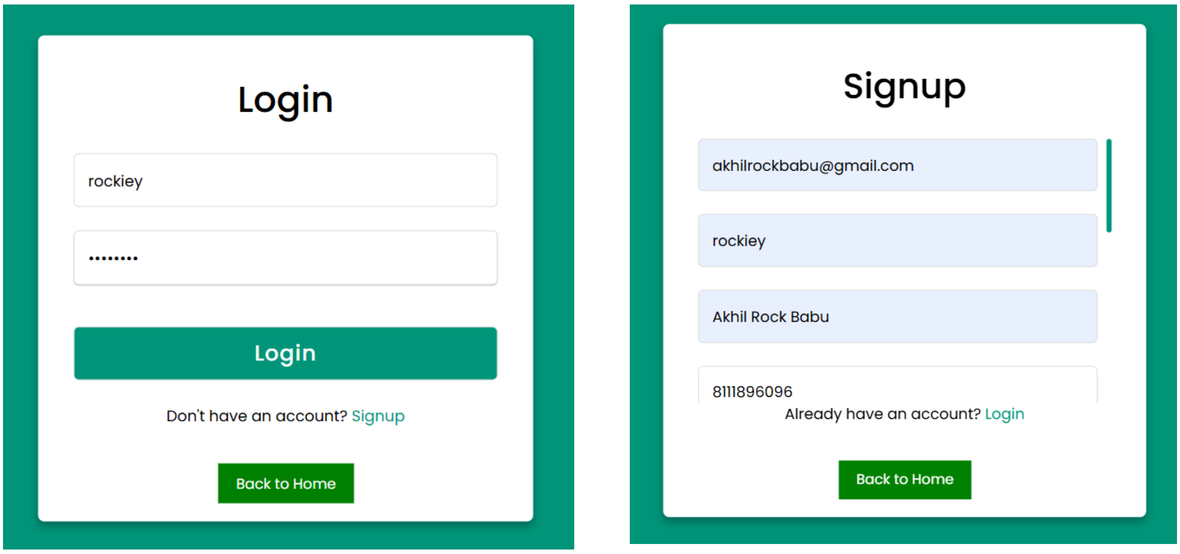
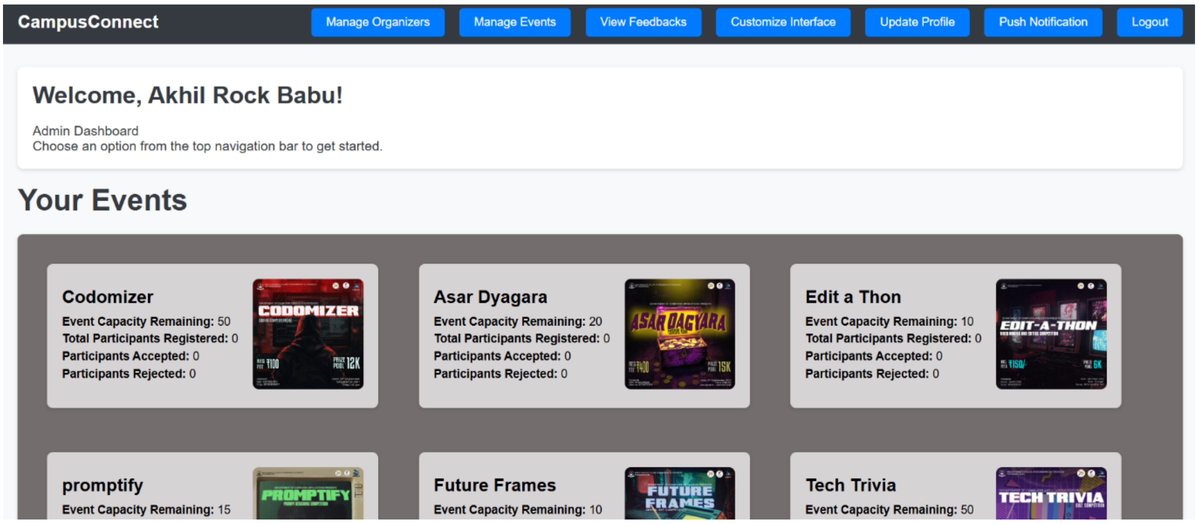


Figure 4.17 Login/Registration

The image shows the login and registration interfaces for user authentication and account creation. The login page (left) enables users to enter their username and masked password, with a green “Login” button for access and a “Signup” link for new users. A “Back to Home” button at the bottom provides easy navigation.

The signup page (right) allows new users to register by entering their email address, username, full name, phone number etc. It also includes a login link for those who already have an account and a “Back to Home” button to navigate to the homepage. Both pages maintain a simple, user-friendly design for seamless interaction.



Welcome, Akhil Rock Babu!

Admin Dashboard
Choose an option from the top navigation bar to get started.

Your Events

Codomizer
Event Capacity Remaining: 50
Total Participants Registered: 0
Participants Accepted: 0
Participants Rejected: 0

Asar Dyagara
Event Capacity Remaining: 20
Total Participants Registered: 0
Participants Accepted: 0
Participants Rejected: 0

Edit a Thon
Event Capacity Remaining: 10
Total Participants Registered: 0
Participants Accepted: 0
Participants Rejected: 0

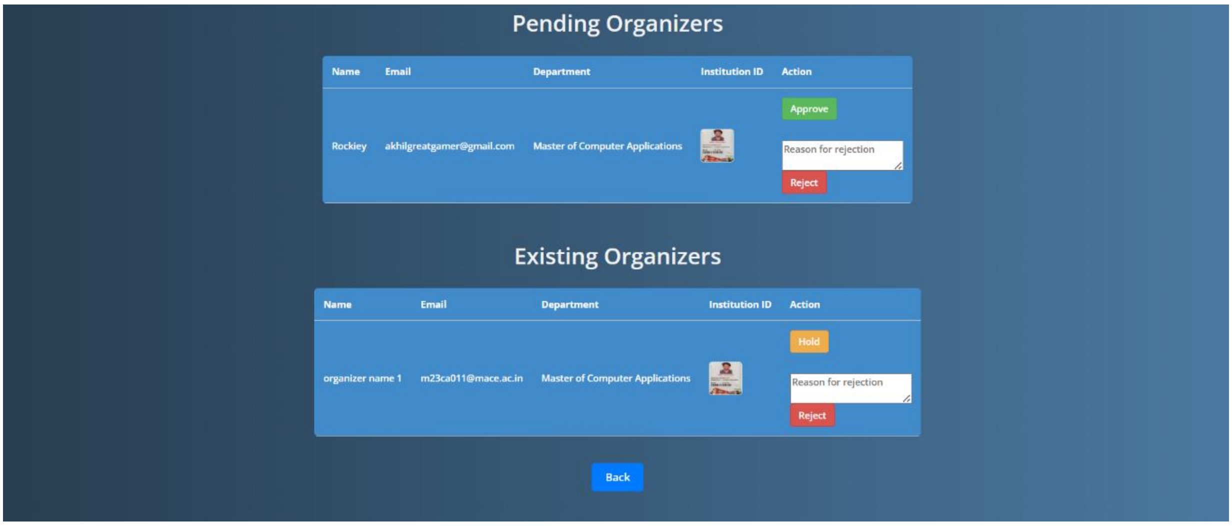
promptify
Event Capacity Remaining: 15

Future Frames
Event Capacity Remaining: 10

Tech Trivia
Event Capacity Remaining: 50

Figure 4.18 Institution Admin Dashboard

The image shows an Institution Admin Dashboard with functionalities for managing event details and tracking participant information. It provides options to manage organizers, events, view feedback, customize the interface, update profiles, and send push notifications. Below these options, a list of event cards is displayed, each containing event-specific details such as remaining capacity, total registered participants, and counts for accepted and rejected participants. This overview helps admins monitor and manage event registrations efficiently.



Pending Organizers

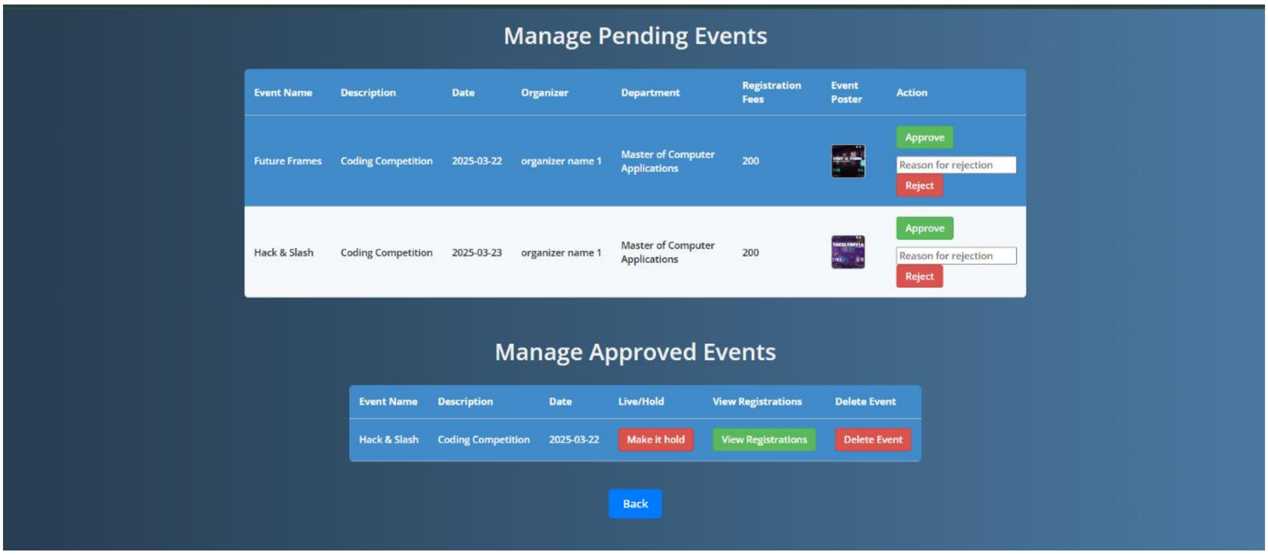
Name	Email	Department	Institution ID	Action
Rockiey	akhilgreatgamer@gmail.com	Master of Computer Applications		Approve Reason for rejection Reject

Existing Organizers

Name	Email	Department	Institution ID	Action
organizer name 1	m23ca011@mace.ac.in	Master of Computer Applications		Hold Reason for rejection Reject

Figure 4.19 Organizer Management

The interface facilitates organizer management by displaying two sections: pending organizers and existing organizers. In the "Pending Organizers" section, admins can approve or reject new organizer requests, with an option to provide a reason for rejection. The "Existing Organizers" section lists approved organizers and includes actions such as placing an organizer on hold or rejecting them, along with a field to specify a rejection reason. This setup streamlines the approval process and enhances control over organizer access.



Manage Pending Events

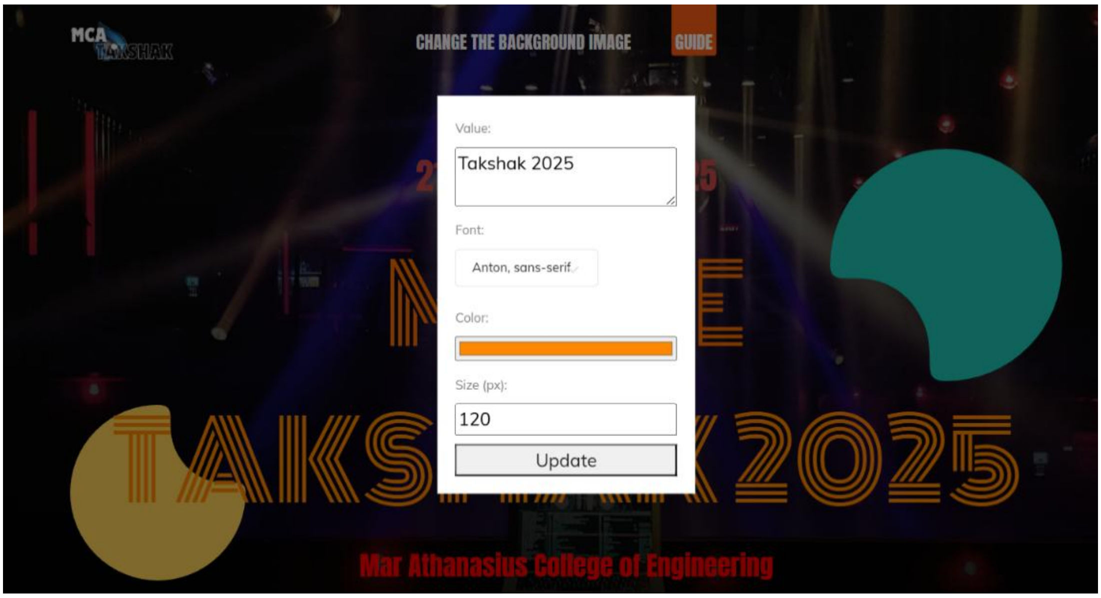
Event Name	Description	Date	Organizer	Department	Registration Fees	Event Poster	Action
Future Frames	Coding Competition	2025-03-22	organizer name 1	Master of Computer Applications	200		Approve Reason for rejection Reject
Hack & Slash	Coding Competition	2025-03-23	organizer name 1	Master of Computer Applications	200		Approve Reason for rejection Reject

Manage Approved Events

Event Name	Description	Date	Live/Hold	View Registrations	Delete Event
Hack & Slash	Coding Competition	2025-03-22	Make it hold	View Registrations	Delete Event

Figure 4.20 Event Management

The interface provides event management functionality by dividing it into "Manage Pending Events" and "Manage Approved Events" sections. In the pending events section, admins can review event details such as name, description, date, organizer, department, registration fees, and event poster, with options to approve or reject events and specify a rejection reason if applicable. The approved events section allows admins to manage live events by placing them on hold, viewing participant registrations, or deleting events. This structure enhances control over event approvals and ongoing event management.


MCA
TAKSHAK
CHANGE THE BACKGROUND IMAGE
GUIDE
TAKSHAK 2025
Mar Athanasius College of Engineering
Update

Value: Takshak 2025

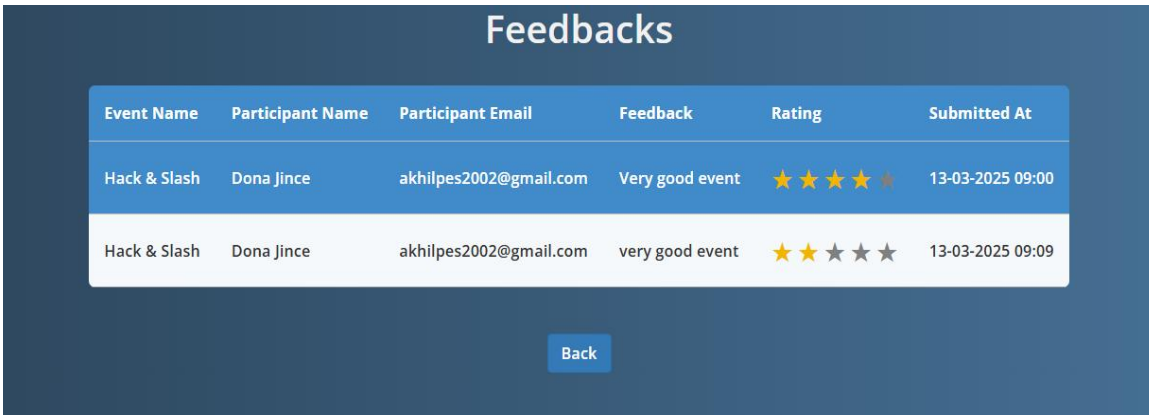
Font: Anton, sans-serif

Color:

Size (px): 120

Figure 4.21 Interface Customization

This interface customization feature allows Institution Admins to modify key visual elements of the event page. Admins can input a custom text value, select a preferred font from a dropdown, choose a color for the text, and adjust the text size in pixels. The "Update" button applies the changes in real-time, enabling Admins to preview and refine the appearance of the event title and related elements. Admins can also update the logo and background image by clicking and upload new images. This feature provides flexibility in personalizing the event's design to suit specific themes or branding requirements.

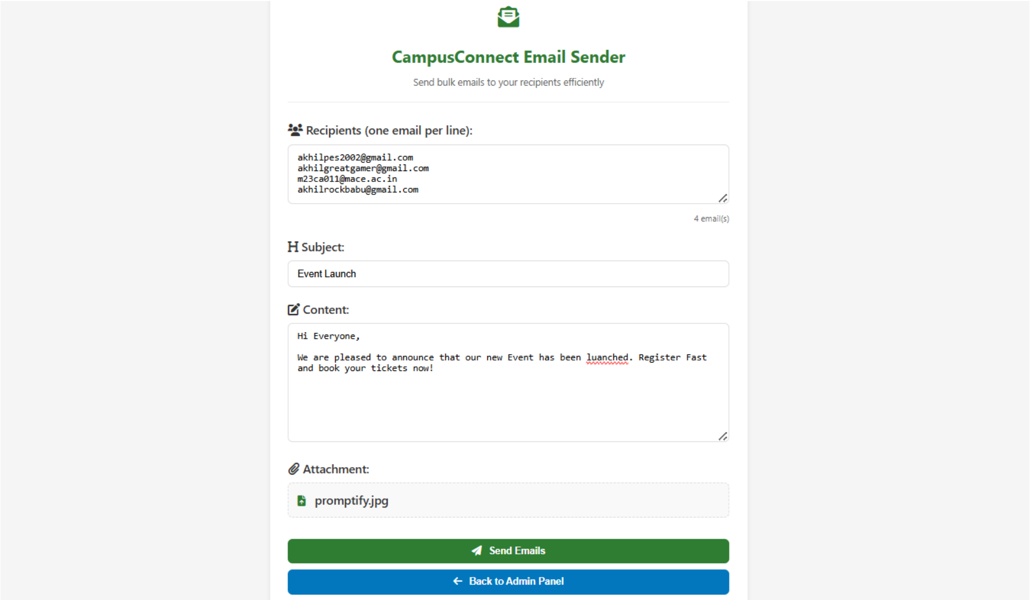


Feedbacks

Event Name	Participant Name	Participant Email	Feedback	Rating	Submitted At
Hack & Slash	Dona Jince	akhilpes2002@gmail.com	Very good event	★★★★☆	13-03-2025 09:00
Hack & Slash	Dona Jince	akhilpes2002@gmail.com	very good event	★★☆☆☆	13-03-2025 09:09

Figure 4.22 View Feedback

This feedback management interface displays a list of feedback entries submitted by participants for specific events. Each entry includes details such as the event name, participant name, participant email, feedback message, rating, and the timestamp indicating when the feedback was submitted. The institution admin can review these entries to evaluate event performance and gather insights for improving future events. The "Back" button enables the admin to navigate to the previous page after viewing the feedback.



CampusConnect Email Sender

Send bulk emails to your recipients efficiently

Recipients (one email per line):

```
akhilpes2002@gmail.com
akhilgreatgamer@gmail.com
m23ca011@mace.ac.in
akhilrockbabu@gmail.com
```

4 email(s)

Subject: Event Launch

Content:

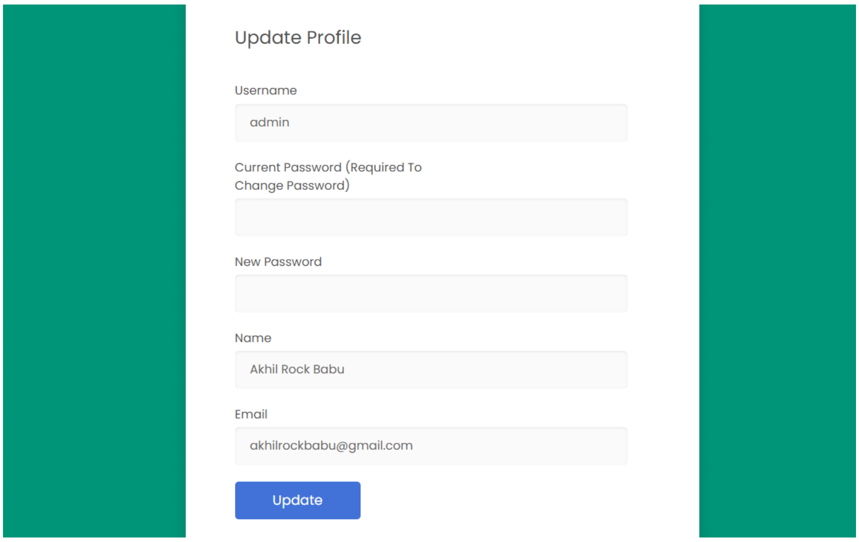
```
Hi Everyone,

We are pleased to announce that our new Event has been luanched. Register Fast
and book your tickets now!
```

Attachment: promptify.jpg

Figure 4.23 Push Notification

This interface shows the bulk email sender feature of the CampusConnect platform, enabling the institutional admin to send push notifications efficiently. The admin can input multiple recipient email addresses (one per line) and compose a subject line and email content. There is also an option to attach files, as shown with the file "promptify.jpg." After composing the email, the admin can click the "Send Emails" button to distribute the notification to all listed recipients. The "Back to Admin Panel" button provides a quick return to the main admin interface. This tool enhances event communication by simplifying bulk email dissemination.



Update Profile

Username: admin

Current Password (Required To Change Password):

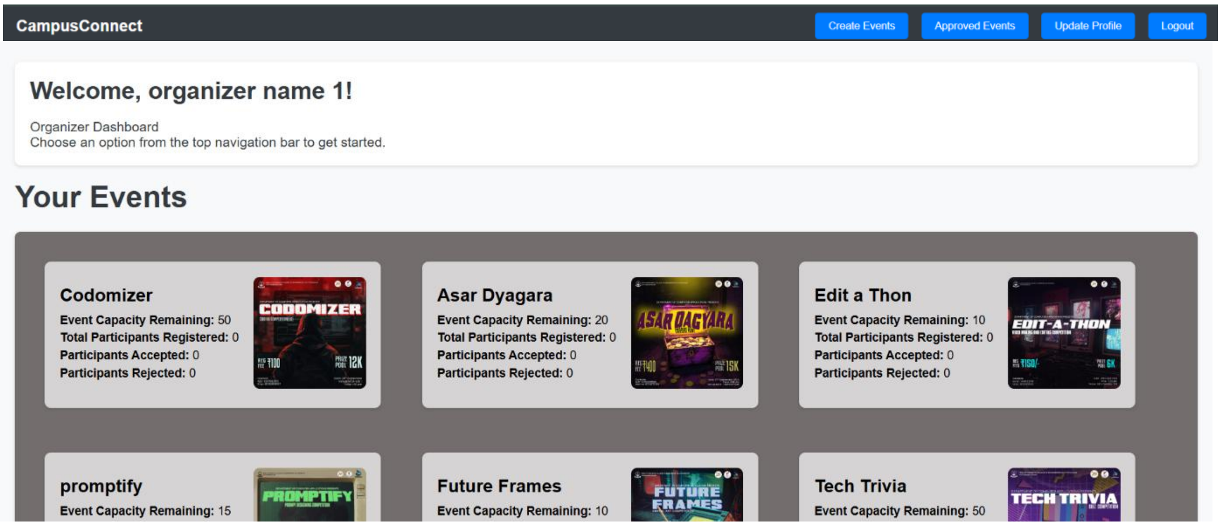
New Password:

Name: Akhil Rock Babu

Email: akhilrockbabu@gmail.com

Figure 4.24 Profile Management

This interface shows a form to update the profile. Users can update their profile by editing their details, such as username, password, name, and email. A password can only be modified if the current password is validated. There is an Update button to go back to the previous pag



CampusConnect

Welcome, organizer name 1!

Organizer Dashboard
Choose an option from the top navigation bar to get started.

Your Events

Codomizer
Event Capacity Remaining: 50
Total Participants Registered: 0
Participants Accepted: 0
Participants Rejected: 0

Asar Dyagara
Event Capacity Remaining: 20
Total Participants Registered: 0
Participants Accepted: 0
Participants Rejected: 0

Edit a Thon
Event Capacity Remaining: 10
Total Participants Registered: 0
Participants Accepted: 0
Participants Rejected: 0

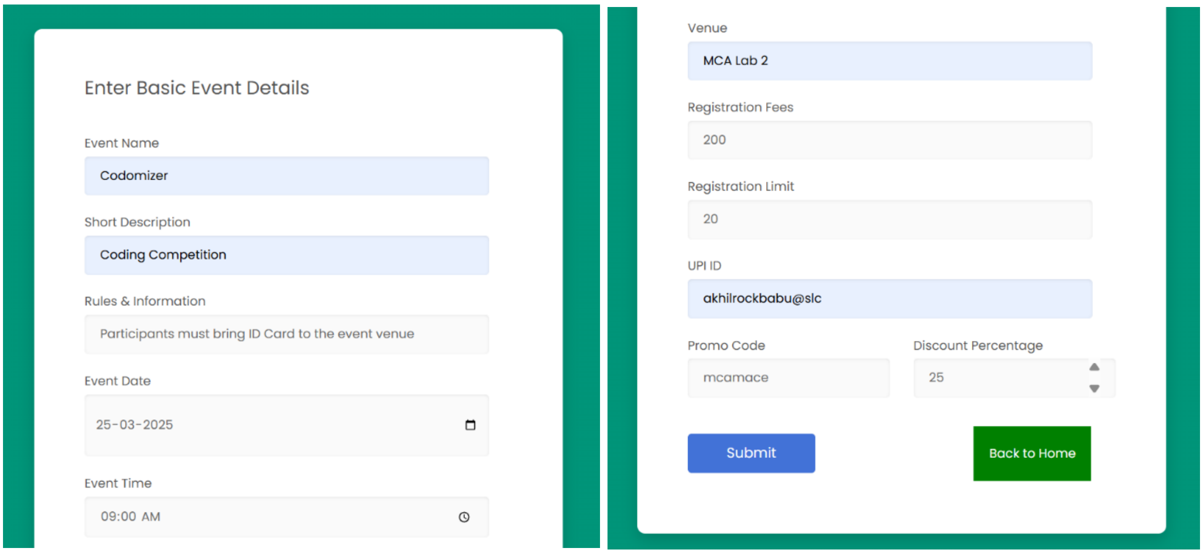
promptify
Event Capacity Remaining: 15

Future Frames
Event Capacity Remaining: 10

Tech Trivia
Event Capacity Remaining: 50

Figure 4.25 Organizer Dashboard

The image shows an Organizer Dashboard with functionalities for creating events and tracking participant information for approved events. It provides options to create events, manage events, and update profiles. Below these options, a list of event cards is displayed, each containing event-specific details such as remaining capacity, total registered participants, and counts for accepted and rejected participants. This overview helps Organizers monitor and manage event registrations efficiently.



Enter Basic Event Details

Event Name
Codomizer

Short Description
Coding Competition

Rules & Information
Participants must bring ID Card to the event venue

Event Date
25-03-2025

Event Time
09:00 AM

Venue
MCA Lab 2

Registration Fees
200

Registration Limit
20

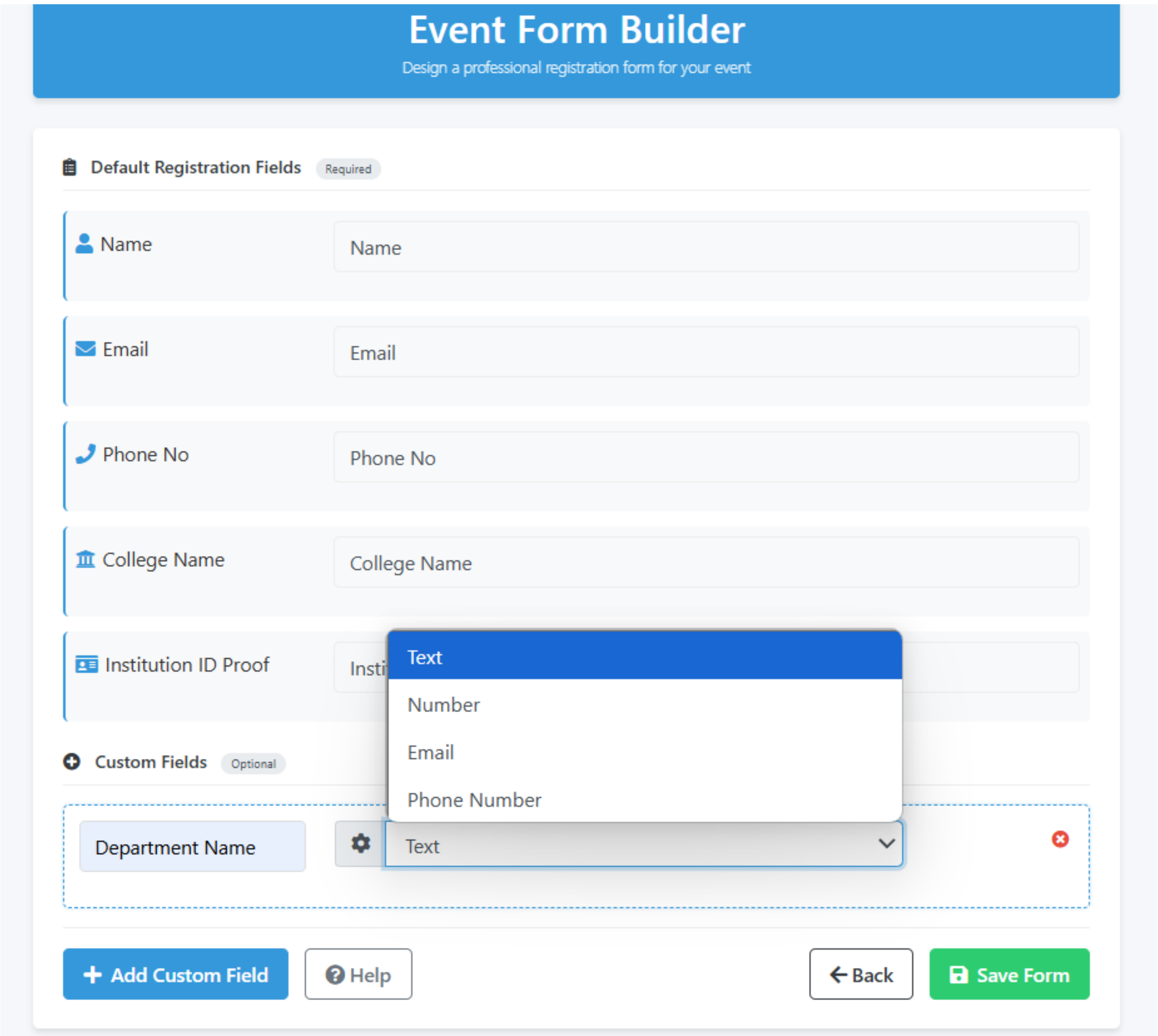
UPI ID
akhilrockbabu@slc

Promo Code
mcamace

Discount Percentage
25

Figure 4.26 Event Creation Basic Form

This image shows the basic form to collect the event details for creating a new event. The details, such as event name, event data, event time, registration fees, event limit, UPI ID to collect event fees, and also promo codes and their discount percentage.



Event Form Builder

Design a professional registration form for your event

Default Registration Fields Required

Name: Name

Email: Email

Phone No: Phone No

College Name: College Name

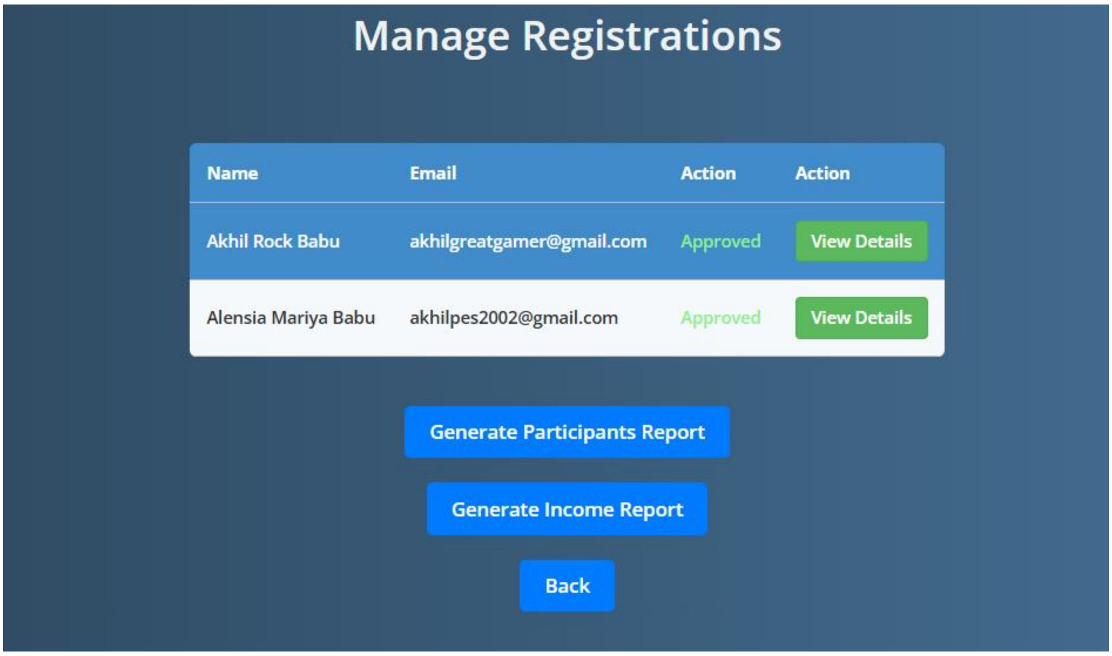
Institution ID Proof: Insti

Custom Fields Optional

Department Name — Text

Figure 4.27 Event Creation Form Builder

This Event Form Builder interface allows organizers to create customized registration forms for their events. The tool features a clean blue header and displays both required default fields (Name, Email, Phone No, College Name, and Institution ID Proof) and a section for optional custom fields. Users can add new custom fields via a dedicated button, with the option to select different field types (Text, Number, Email, Phone Number) from a dropdown menu. A custom "Department Name" field is currently being configured. The interface includes helpful navigation buttons at the bottom: "Add Custom Field," "Help," "Back," and a green "Save Form" button to finalize changes. This dynamic form builder streamlines the process of creating tailored registration forms that meet specific event requirements.

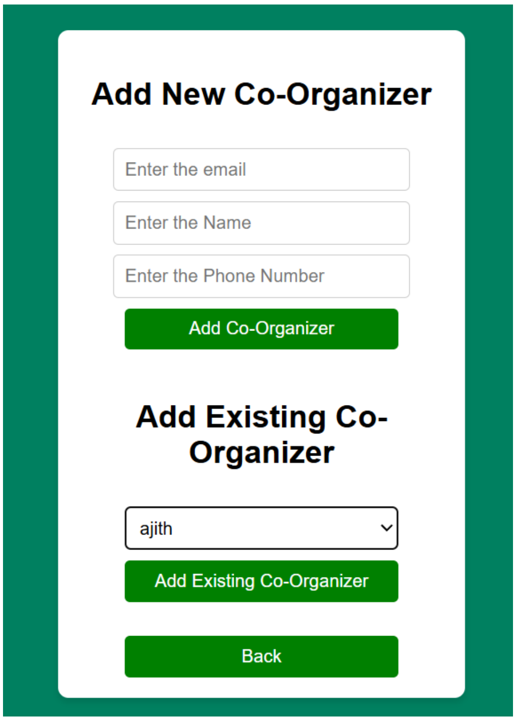


Manage Registrations

Name	Email	Action	Action
Akhil Rock Babu	akhilgreatgamer@gmail.com	Approved	View Details
Alensia Mariya Babu	akhilpes2002@gmail.com	Approved	View Details

Figure 4.28 Event Registrations Management

The Event Registrations Management interface provides tools to oversee participant registrations for their events. The page displays a tabular list of registrants showing essential information including names and email addresses, along with their approval status. Each entry includes a "View Details" option to access more comprehensive participant information. Below the registration list, the interface offers reporting functionality with options to "Generate Participants Report" and "Generate Income Report," allowing organizers to create summaries of attendance and financial data. A "Back" button provides navigation to previous screens. This management tool centralizes registration oversight and reporting capabilities in one streamlined dashboard.



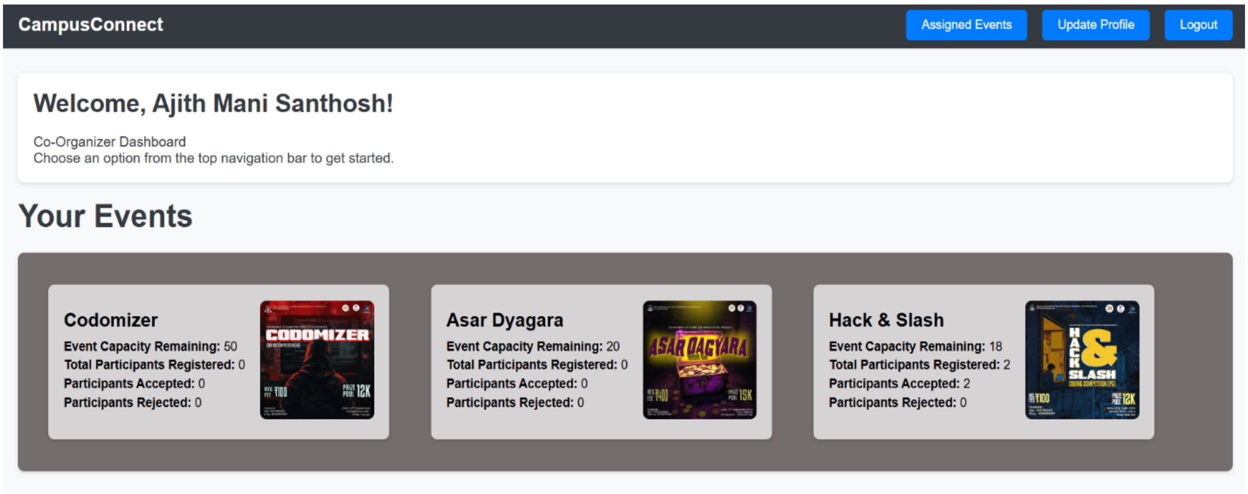
Add New Co-Organizer

Add Existing Co-Organizer

ajith

Figure 4.29 Add Co-Organizers

This interface provides two options for adding co-organizers to an event. The top section allows users to add new co-organizers by entering their email, name, and phone number in the provided fields, then clicking the "Add Co-Organizer" button. The bottom section offers a way to add existing users as co-organizers through a dropdown menu followed by an "Add Existing Co-Organizer" button to confirm the selection. This dual approach accommodates both new collaborators who aren't yet in the system and existing users who can be quickly assigned co-organizer privileges. A "Back" button at the bottom provides navigation to the previous screen. The functionality streamlines the process of building an event organization team.



Welcome, Ajith Mani Santhosh!

Co-Organizer Dashboard
Choose an option from the top navigation bar to get started.

Your Events

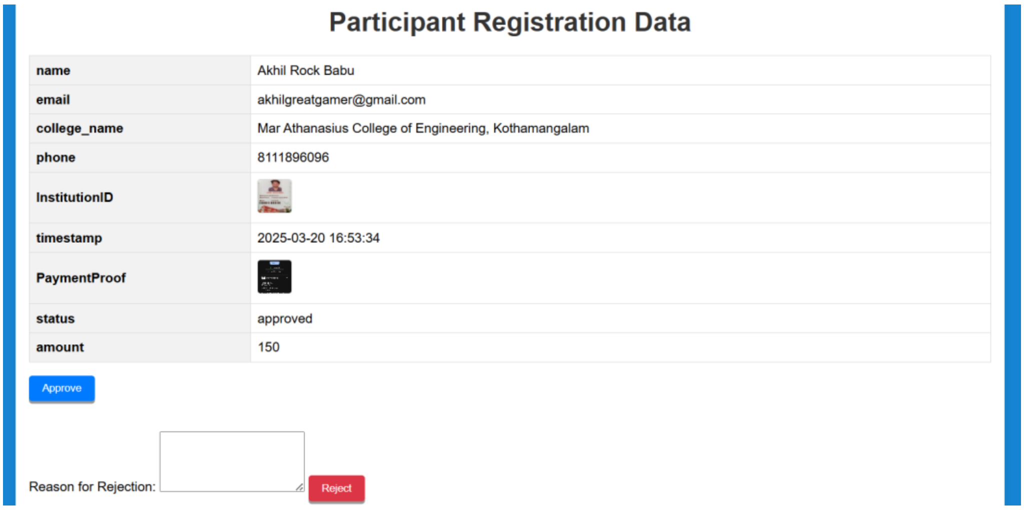
Codomizer
Event Capacity Remaining: 50
Total Participants Registered: 0
Participants Accepted: 0
Participants Rejected: 0

Asar Dyagara
Event Capacity Remaining: 20
Total Participants Registered: 0
Participants Accepted: 0
Participants Rejected: 0

Hack & Slash
Event Capacity Remaining: 18
Total Participants Registered: 2
Participants Accepted: 2
Participants Rejected: 0

Figure 4.30 Co-Organizers Dashboard

It provides clear navigation options in the header, including Assigned Events, Update Profile, and Logout. Below the welcome message, the dashboard displays a section titled "Your Events" which showcases the events that the user is co-organizing. Each event card features an event poster image and key metrics including event capacity remaining, total participants registered, and the number of participants accepted or rejected. This dashboard provides co-organizers with a quick overview of their event portfolio and registration status, allowing them to monitor participation metrics immediately.



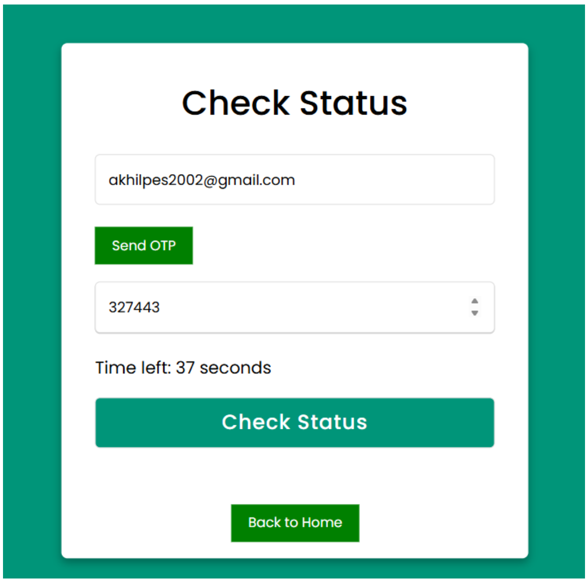
Participant Registration Data

name	Akhil Rock Babu
email	akhilgreatgamer@gmail.com
college_name	Mar Athanasius College of Engineering, Kothamangalam
phone	8111896096
InstitutionID	
timestamp	2025-03-20 16:53:34
PaymentProof	
status	approved
amount	150

Reason for Rejection:

Figure 4.31 View Registration Data

This interface displays detailed participant registration information for event organizers to review. The page presents a comprehensive overview of an individual registrant's information in a structured format. Fields include basic contact details (name, email, phone), institutional information (college name, institution ID with thumbnail image), registration metadata (timestamp), payment information (payment proof with thumbnail image, amount), and current approval status. The page provides action buttons at the bottom allowing organizers to "Approve" the registration or "Reject" it with an optional text field to provide a reason for rejection. This detailed view enables organizers to verify participant information, confirm payment, and make informed decisions about registration approval.



Check Status

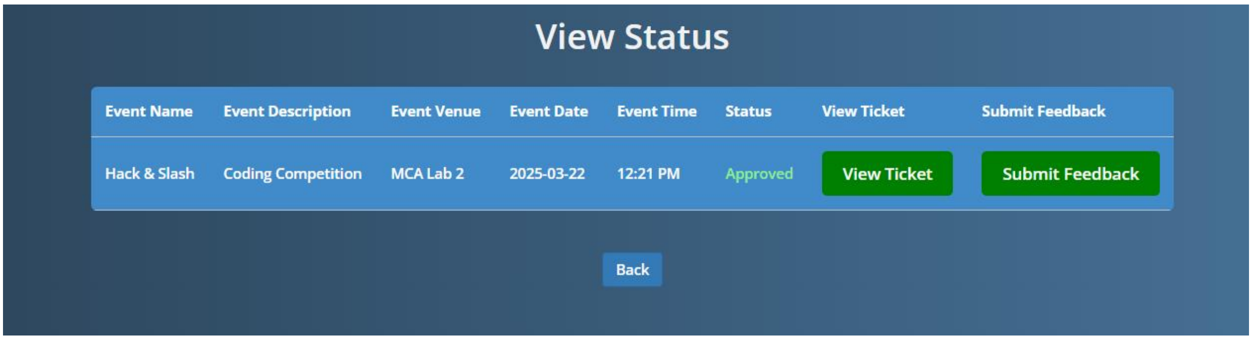
akhilpes2002@gmail.com

327443

Time left: 37 seconds

Figure 4.32 Check Status

This Check Status interface allows participants to verify their event registration status through a secure verification process. The screen features an email input field where users enter their registered email address, followed by a Send OTP button that triggers a one-time password delivery. Below this, users can input the received OTP in a dedicated field, with a countdown timer showing the remaining validity period of the code. Once the OTP is entered, users can click the prominent Check Status button to view their registration status and event tickets. A Back to Home option at the bottom provides easy navigation to the main page. This verification system ensures that only registered participants can access their specific registration information while maintaining security through email verification.

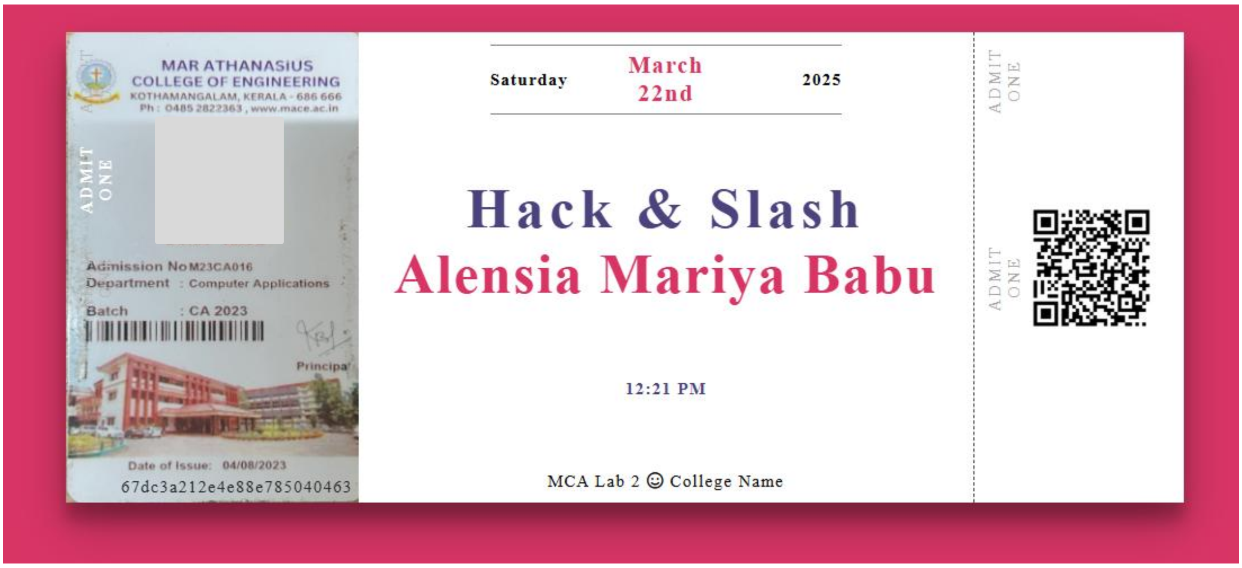


View Status

Event Name	Event Description	Event Venue	Event Date	Event Time	Status	View Ticket	Submit Feedback
Hack & Slash	Coding Competition	MCA Lab 2	2025-03-22	12:21 PM	Approved	View Ticket	Submit Feedback

Figure 4.33 View Status

The View Status page displays a participant's event registration status in a tabular format. The interface presents comprehensive details about the registered event including its name, description, venue, date, and time. A status column clearly indicates whether the registration has been approved. Two action buttons provide key functionality: View Ticket allows participants to access their digital ticket for the event, while Submit Feedback enables attendees to share their experience after the event. A Back button at the bottom of the page facilitates easy navigation to previous screens. This status view serves as a central reference point for participants to verify their registration status and access important event-related functions.



MAR ATHANASIUS
COLLEGE OF ENGINEERING
KOTHAMANGALAM, KERALA - 686 666
Ph : 0485 2822363 , www.mace.ac.in

ADMIT ONE

Admission No M23CA016
Department : Computer Applications
Batch : CA 2023

Principal

Date of Issue: 04/08/2023
67dc3a212e4e88e785040463

Saturday March 22nd 2025

Hack & Slash
Alensia Mariya Babu

12:21 PM

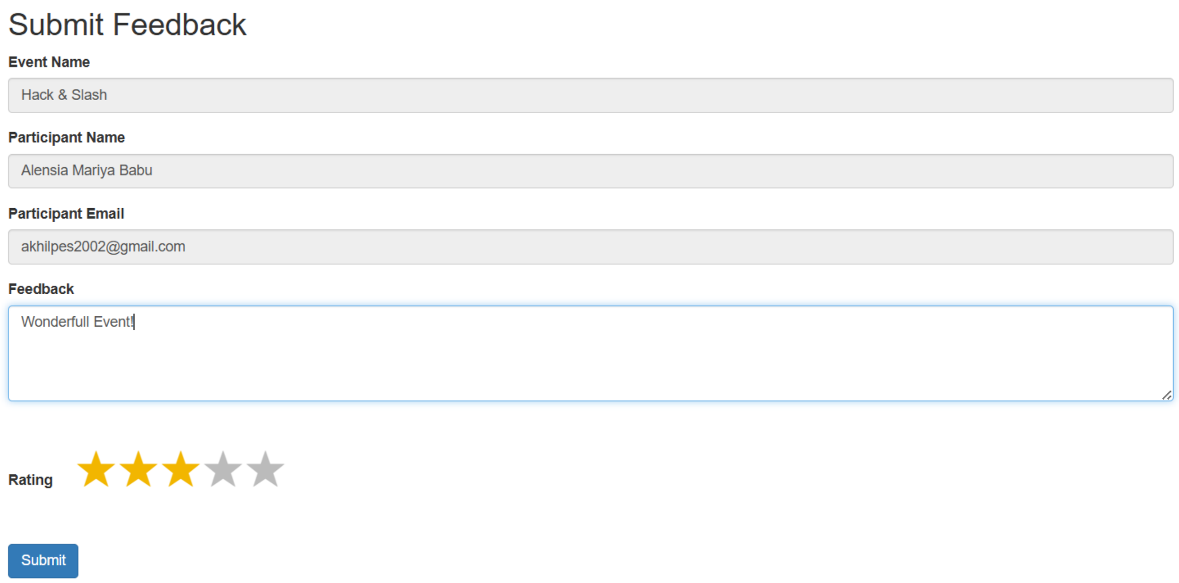
MCA Lab 2 ☺ College Name

ADMIT ONE

ADMIT ONE

Figure 4.34 Ticket Generation

The Ticket Generation feature creates personalized digital admission credentials for event participants. This example shows a ticket for the "Hack & Slash" event, combining the participant's institutional ID card on the left with essential event details in the center (date, event name, participant name, time, and venue). A QR code on the right enables quick verification during check-in. This digital ticket serves as the official admission document, merging identification and event information in a professional format.



Submit Feedback

Event Name

Hack & Slash

Participant Name

Alensia Mariya Babu

Participant Email

akhilpes2002@gmail.com

Feedback

Wonderfull Event!

Rating ★★★☆☆

Figure 4.35 Feedback Form

The Feedback Form interface allows event participants to share their experience after attending an event. The form displays pre-filled fields showing the event name and participant information (name and email) to ensure feedback is properly associated with the correct attendee and event. Participants can provide written comments in a text area and rate their overall experience using a five-star rating system. After completing their feedback, users can finalize their submission with the Submit button. This feedback collection mechanism helps organizers gather valuable insights about event quality and participant satisfaction to improve future offerings.

5. TESTING

Software testing is a critical element of software quality assurance and represents the ultimate review of the specifications, design, and code generation for CampusConnect - Event Registration System. Once the source code is generated, the system must be systematically tested before being delivered to colleges and educational institutions. The objective is to identify and remove all possible errors to ensure a seamless experience for institutional admins, organizers, co-organizers, and participants.

To maximize efficiency and accuracy, testing in CampusConnect - Event Registration System is conducted using well-structured techniques and disciplined methodologies. The importance of software testing and its implications for delivering a high-quality, error-free product cannot be overstated. The goal is to conduct a comprehensive and systematic evaluation of the platform's specifications, design, and code generation.

Given that the possible test cases for even simple modules of CampusConnect - Event Registration System can be practically infinite, a testing strategy is employed to optimize for the available time and resources. Thus, software testing aims not just to uncover software bugs but also to validate that the system meets user requirements and functions efficiently across a range of use cases, including handling exceptional cases and boundary conditions.

The testing process also plays a key role in improving CampusConnect - Event Registration System in terms of functionality, usability, and performance. It helps identify any gaps, errors, or missing requirements compared to user expectations and actual system behavior. Testing can be conducted manually or using automated tools, and it encompasses both White Box Testing and Black Box Testing methodologies.

The software testing process for CampusConnect - Event Registration System can be divided into two key components:

i Verification: This includes tasks that ensure that the platform correctly implements all specific functions, including organizer management, event creation, and participant registrations.

ii Validation: This involves tasks that ensure that the system is traceable to customer and user requirements, delivering the expected functionalities and features.

The testing process for CampusConnect - Event Registration System follows three phases:

- Unit Testing
- Integration Testing
- System/Final Testing

5.1 Unit Testing

Unit testing focuses on verifying the smallest units or modules of CampusConnect - Event Registration System, such as the login and registration module, event creation functionality, and user profile management. Each module is tested individually to validate that it functions as expected and that all conditions and validations are properly handled.

During unit testing, control paths are tested to ensure that data flows correctly into and out of the tested modules. Boundary conditions are tested to confirm that the modules function correctly even at input extremes (e.g., large participant lists, special character handling in form inputs). This ensures that all statements or conditions within a module are executed at least once.

5.1.1 Unit Test Cases

SI.No	Procedure	Expected Output	Actual Output	Status
1.	Validate Organizer Registration with valid Inputs	Organizer Registered Successfully	Organizer Registered Successfully	Pass
2.	Validate Organizer Registration with invalid Inputs	Organizer Registration Cancelled.	Error message indicating invalid field.	Pass
3.	Approve Organizer Registration by Admin	Registration Approved	Registration Approved	Pass
4.	Reject Organizer Registration by Admin	Registration Rejected	Registration Rejected	Pass
5.	Approved Organizer Login	Login Successful	Login Successful	Pass
6.	Pending Organizer Login	Login Failed	Error message indicating "wait for approval"	Pass
7.	Organizers create events by valid inputs	Event Creation Successful	Event Creation Successful	Pass
8.	Organizers create events by invalid inputs	Event Creation Unsuccessful	Event Creation Unsuccessful	Pass
9.	Admin Approves Events	Event Approved	Event Approved	Pass

10.	Admin Rejects an Events	Event Rejected	Event Rejected	Pass
11.	Organizer approves event Registrations	Registration Approved	Registration Approved	Pass
12.	Organizer rejects event Registrations	Registration Rejected	Registration Rejected	Pass
13.	Organizers assign Co-Organizers	Co-Organizers Assigned	Co-Organizers Assigned	Pass
14.	Participants View Generated Event Tickets	Generated Tickets Available	Generated Tickets Available	Pass
15.	Participants Submit Feedback	Feedback Submitted	Feedback Submitted	Pass

Table 5.1 Unit Test Cases

5.2 Integration Testing

Integration testing focuses on ensuring that the various modules of CampusConnect - Event Registration System work correctly when integrated together. Errors can arise during the interaction between modules, such as data loss across module interfaces or sub-functions not performing as expected when combined.

For this project, top-down integration testing was employed, starting with the main modules such as admin dashboard, organizer dashboard, and event management. As each module was integrated, it was tested to ensure smooth data flow and seamless functionality.

5.2.1 Integration Test Cases

SI.No	Procedure	Test Case Description	Expected Output	Actual Output	Status
1.	Integration of Organizer Management Module & Admin Management Module	Register a Organizer & Approve the same Organizer	Organizer Management Module Seamlessly integrates with Admin Management Module	Registered Organizers are correctly visible in Admin Management Module	Pass
2.	Integration of Organizer Management Module	Login as a Organizer & Create an Event	Event is Created under	Event is Created correctly and owned by the	Pass

	& Event Management Module		created Organizer	Organizer who created it.	
3.	Integration of Participant Management Module & Event Management Module	Register for and Event as a Participant	The Participant is registered for the same Event	Participants are Registered for the same event registered for.	Pass
4.	Integration of Admin Management Module & Customization Management Module	Login as an Admin and customize the website Interface	All the Customization changes are applied correctly	Website Interface is Customized by Admin to meet special branding needs	Pass
5.	Integration of Participant Management Module & Feedback Management Module	Submitting a Feedback as a Participant	Feedback is submitted successfully from the same Participant	Feedback is submitted successfully from the same Participant	Pass
6.	Integration of Admin Management Module & Feedback Management Module	Check for the submitted Feedbacks and it correctly displays all.	All the Submitted Feedbacks are visible for the Admin	All the Submitted Feedbacks are visible for the Admin	Pass
7.	Integration of Admin Management Module & Report Management Module	Generate Reports by login as an Admin	Reports are successfully Generated	Reports are successfully Generated	Pass
8.	Integration of Organizer Management Module & Report Management Module	Generate Reports by login as an Organizer	Reports are successfully Generated	Reports are successfully Generated	Pass

Table 5.2 Integration Test Cases

After each successful integration, the modules were tested, and any errors identified were fixed. Upon final integration, the system was observed to work without any uncovered errors, ensuring a fully functional event management process.

5.3 System Testing

System testing is the final phase in the testing process, where CampusConnect - Event Registration System is tested as a complete product to ensure it meets all functional and performance requirements. This phase involves testing the entire platform with prepared test data in a structured and planned manner.

Errors can arise at any stage of the software development process, and system testing aims to detect and eliminate those errors before delivery. This phase includes both verification and validation, which are classified into two methods:

- Static Testing: This involves reviewing the system's code and documentation without executing it, to identify any inconsistencies, missing requirements, or logical errors.
- Dynamic Testing: This involves executing the system to observe its behavior and validate that it meets the specified requirements.

During system testing, all the modules of CampusConnect - Event Registration System are tested together to ensure smooth overall functionality. Tests are conducted to check:

- Login and access management for different user roles (admin, organizers, participants).
- Event creation, dynamic form customization, and participant registration workflows.
- Email notifications and updates sent to participants in bulk.
- Secure payment handling and fee management.
- Platform customization options for institutional admins.

The testing process ensures that CampusConnect - Event Registration System operates efficiently, reliably, and in alignment with user expectations, delivering a seamless experience for all stakeholders.

5.4 Backend Testing

Backend testing, also known as database testing, plays a vital role in ensuring the reliability, functionality, and integrity of the CampusConnect - Event Registration System. This testing phase focuses on verifying server-side operations, including CRUD (Create, Read, Update, Delete) functionalities, and interactions with the database. Proper backend testing helps prevent issues such as data inconsistencies, data corruption, and transaction failures, ensuring that the system functions smoothly and efficiently.

In the CampusConnect - Event Registration System, backend testing is primarily conducted by testing SQL queries, stored procedures, and database triggers to validate data storage and retrieval. Various test cases are created to check the correctness of data insertion, deletion, and updates within the database. The test cases also validate the relationships between different database tables, ensuring that key functionalities such as user registration, event creation, approval processes, and ticket generation work seamlessly.

By testing backend functionalities rigorously, the system ensures accurate data handling, enhances performance, and maintains overall system quality. This contributes to a smooth event registration experience for all users, including admins, organizers, and participants.

5.4.1 Backend Test Cases

SI.No	Procedure	Expected Output	Actual Output	Status
1.	Admin approves a new organizer's registration request.	Organizer details are stored in the database.	Organizer details are stored in the database.	Pass
2.	Admin rejects a new organizer's registration request.	Delete the corresponding organizer record from the database.	Delete the corresponding organizer record from the database.	Pass
3.	Participants view their event registration details and ticket information.	Display the registration details and ticket.	Registration details and ticket successfully displayed.	Pass
4.	Organizer adds a new event and submits event details for admin approval.	Event details are stored and marked as "Pending"	Event details successfully stored and status set to pending	Pass
5.	Admin updates the approval status of an event (Approved/Rejected).	Update the event status and store the updated status in the DB.	Event status successfully updated in the database.	Pass

Table 5.3 Backend Test Cases

This backend testing approach ensures that the CampusConnect - Event Registration System remains robust, functional, and capable of handling various backend operations crucial for efficient event management.

5.5 GUI Testing

For the CampusConnect - Event Registration System, GUI testing focuses on ensuring that all visual interface components function correctly and deliver a smooth, user-friendly experience for admins, organizers, co-organizers, and participants. The testing is structured to validate various aspects of the graphical user interface (GUI) for different modules of the application, such as login, registration, event creation, and dashboard functionalities.

Key Areas of GUI Testing:

- Input Fields Testing: Verify that input fields (username, password, email, event name, etc.) accept data in the correct format and trigger validation messages for invalid entries.
- Button Testing: Ensure that buttons like 'Login', 'Register', 'Submit', and 'Create Event' perform the intended actions and redirect users to appropriate pages upon valid input.
- Error Messaging: Validate that error messages are clear, informative, and displayed when users enter incorrect or incomplete information.
- Visual and Alignment Checks: Confirm that images, text, and controls are properly aligned, readable, and visually appealing.
- Navigation: Verify that users can smoothly navigate between different pages without any glitches.

SI.No	Procedure	Expected Output	Actual Output	Status
1.	Check whether input fields (username, password, event name, etc.) accept user input	Able to enter data into input fields.	Able to enter data into input fields.	Pass
2.	Verify that the images (event banners, icons) have good clarity.	Images have good clarity.	Images have good clarity.	Pass
3.	Check the validation of input fields (email, password, radio buttons, etc.).	Validation successful.	Validation successful.	Pass
4.	Check whether error messages appear for incorrect login/registration data.	Error messages are displayed appropriately.	Error messages are displayed appropriately.	Pass
5.	Verify that fonts used in the interface are readable and consistent.	Fonts used are readable and consistent.	Fonts used are readable and consistent.	Pass
6.	Check the alignment of text, buttons, and other elements on the pages.	All elements are properly aligned.	All elements are properly aligned.	Pass
7.	Check spelling and grammar in the interface text and labels.	Spelling and grammar are correct.	Spelling and grammar are correct.	Pass

Table 5.4 GUI Test Cases

This structured GUI testing ensures that the CampusConnect - Event Registration System provides users with a visually appealing, intuitive, and error-free interface, contributing to a positive and seamless user experience.

6. IMPLEMENTATION

The implementation phase of the CampusConnect - Event Registration System involves transforming the conceptual design into a fully functional web-based platform that colleges and educational institutions can use to streamline event registration and management. This phase marks the culmination of planning, designing, developing, and testing efforts, bringing the system to life and ensuring it is ready for real-world usage.

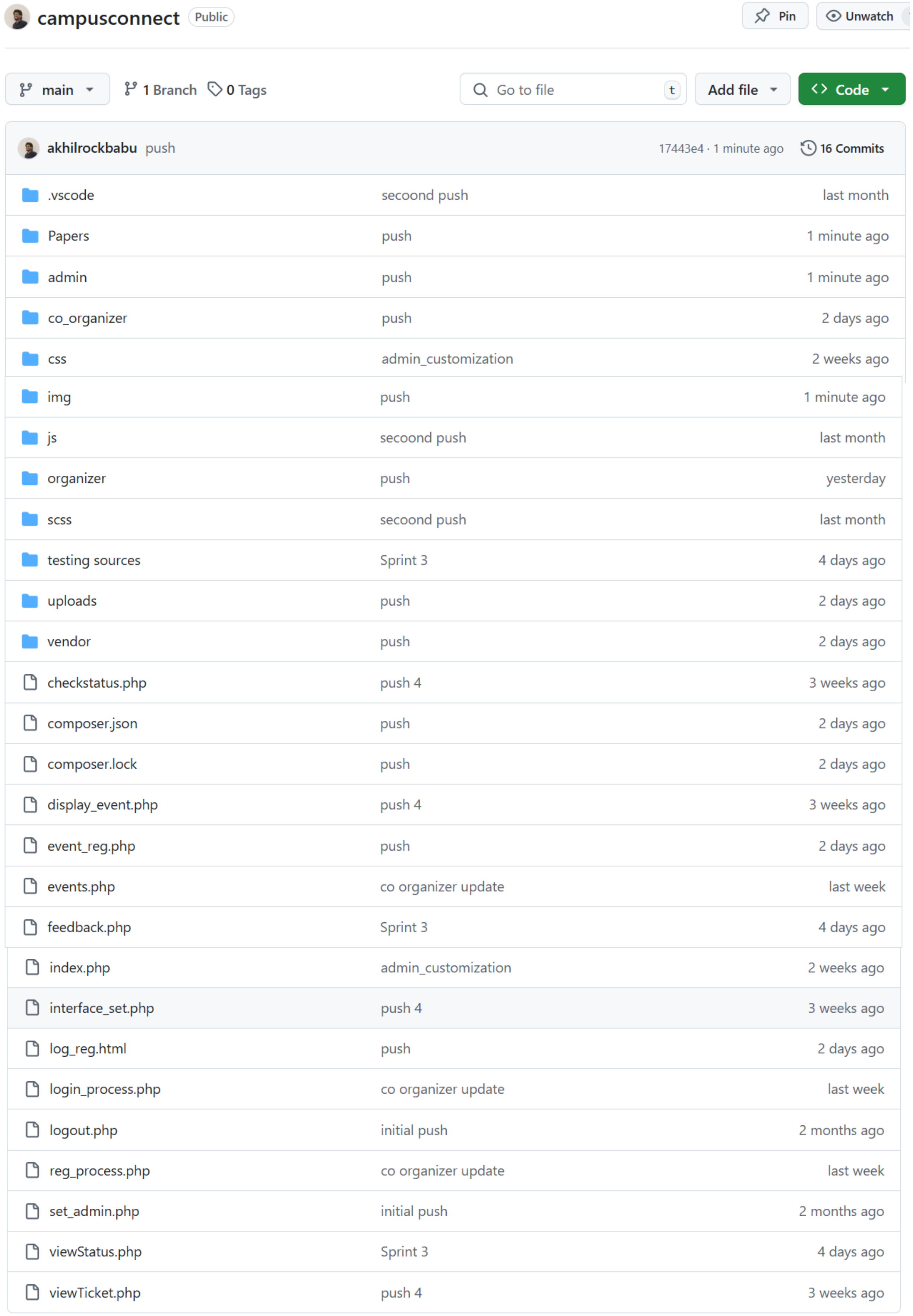
One of the key challenges during the implementation phase is ensuring seamless integration of various modules, including user registration, event creation, approval workflows, and ticket generation, while maintaining overall system stability. Additionally, compatibility with different user roles (admin, organizers, co-organizers, and participants) and ensuring efficient database interactions require careful consideration.

To address these challenges, the implementation process of CampusConnect follows a structured and well-coordinated approach. This includes:

- Modular Integration: Ensuring that all system modules function harmoniously and exchanging data correctly to prevent any disruptions in functionality.
- Database Optimization: Conducting thorough compatibility checks to ensure that database operations, such as storing and retrieving user and event details, perform efficiently.
- Code Reusability: Reusing and adapting existing functions and libraries wherever feasible to improve efficiency while maintaining system reliability and avoiding redundancy.
- User Training and Documentation: To facilitate the smooth adoption of the system, training sessions are tailored to each user group’s specific needs. This includes training admins on managing event approvals, guiding organizers on event creation and tracking, and assisting participants in registering for events and downloading tickets.

By addressing integration challenges, optimizing code reusability, and offering comprehensive user training, the implementation phase ensures that the CampusConnect - Event Registration System is robust, user-friendly, and easy to deploy. This effective implementation paves the way for seamless event management, empowering educational institutions to manage events efficiently, enhance user engagement, and simplify the entire event registration process. Ultimately, this contributes to the platform’s successful adoption and usability across diverse institutional settings.

7. GIT HISTORY



akhilrockbabu push 17443e4 · 1 minute ago

.vscode	secoond push	last month
Papers	push	1 minute ago
admin	push	1 minute ago
co_organizer	push	2 days ago
css	admin_customization	2 weeks ago
img	push	1 minute ago
js	secoond push	last month
organizer	push	yesterday
scss	secoond push	last month
testing sources	Sprint 3	4 days ago
uploads	push	2 days ago
vendor	push	2 days ago
checkstatus.php	push 4	3 weeks ago
composer.json	push	2 days ago
composer.lock	push	2 days ago
display_event.php	push 4	3 weeks ago
event_reg.php	push	2 days ago
events.php	co organizer update	last week
feedback.php	Sprint 3	4 days ago
index.php	admin_customization	2 weeks ago
interface_set.php	push 4	3 weeks ago
log_reg.html	push	2 days ago
login_process.php	co organizer update	last week
logout.php	initial push	2 months ago
reg_process.php	co organizer update	last week
set_admin.php	initial push	2 months ago
viewStatus.php	Sprint 3	4 days ago
viewTicket.php	push 4	3 weeks ago

Figure 7.1 Git History

8. CONCLUSION

In conclusion, the CampusConnect - Event Registration System has successfully achieved its primary objective of providing a streamlined, automated, and user-friendly solution for managing event registration processes in educational institutions. By integrating dynamic functionalities and advanced features, the system addresses the limitations of traditional manual and static event registration methods. Leveraging modern web technologies, including PHP, JavaScript, jQuery, and MongoDB, CampusConnect enhances the efficiency, flexibility, and security of event management. The modular design of the system, based on the Model-View-Controller (MVC) architecture, ensures scalability, maintainability, and future extensibility.

The platform supports multi-level user access, where institutional admins, organizers, co-organizers, and participants interact seamlessly through dedicated interfaces tailored to their specific roles. Organizers can register and create events, dynamically customize event-specific registration forms using the integrated jQuery Form Builder and manage participant registrations. Institutional admins oversee the approval/rejection process for organizers and events, customize the website interface to align with institutional branding, and generate detailed event analysis reports. Participants benefit from a simplified registration process, real-time event notifications, secure QR-code-based payment, and the ability to track their registration status, download tickets, and provide post-event feedback.

The system's security is reinforced through multiple layers of authentication and role-based access control. Features such as email-based OTP verification for participants and organizers prevent unauthorized access, while file validation ensures that only appropriate files, including institution ID uploads, are processed. Real-time data updates and automated notifications improve engagement by keeping users informed about approvals, and rejections.

The successful implementation of dynamic form building, live event status toggling, automated approval mechanisms, bulk email notifications, and customizable web pages demonstrates the system's transition from theoretical design to a practical, fully functional web-based product. Additionally, the QR-code integration for event ticket generation facilitates seamless entry verification at event venues, further enhancing user convenience and event security.

The CampusConnect project’s robust backend logic, efficient database structure, and interactive frontend ensure smooth data handling and system performance, even with large volumes of participant and event data. This comprehensive solution reduces administrative overhead by automating routine tasks, improving operational efficiency, and minimizing manual interventions. Moreover, the integration of analytics dashboards for event performance tracking provides valuable insights for future event planning and resource allocation.

Overall, CampusConnect stands as a scalable, secure, and adaptable platform designed to meet the evolving needs of educational institutions. The project's success underscores the importance of leveraging automation, customization, and real-time data processing in enhancing the event management experience.

9. FUTURE WORKS

The CampusConnect - Event Registration System has successfully implemented essential features to streamline event management for educational institutions. To enhance the system further, the following future works are proposed to improve scalability, user experience, and functionality:

AI-Driven Event Recommendations

Integrating machine learning algorithms can enhance user engagement by providing personalized event recommendations based on participants' past registrations and preferences. This would help increase event participation and improve overall user satisfaction.

Blockchain-Based Certificate Generation and Verification

Implementing blockchain technology will allow tamper-proof digital certificates for event participants, speakers, and winners. This feature enhances the authenticity, security, and transparency of certificate issuance and verification.

Advanced Analytics Dashboard for Organizers and Admins

Developing detailed analytics dashboards with interactive graphs, charts, and participant engagement trends will provide organizers and admins with actionable insights. This feature will enable data-driven decisions to optimize future event planning and participation.

By implementing these advanced features, CampusConnect can evolve into a more robust and innovative platform, providing an enhanced, secure, and user-friendly experience for admins, organizers, and participants alike.

10. APPENDIX

10.1 Minimum Software Requirements

PHP:

Version 7.4 or above is required for handling server-side logic, managing HTTP requests, and database interactions.

MongoDB:

MongoDB is required for storing and retrieving event-related data, participant information, and customization settings.

Apache Web Server (XAMPP/WAMP):
Necessary to host the web application locally and manage PHP scripts and database connections efficiently.

JavaScript and AJAX:

Required for enhancing interactivity in the frontend, handling dynamic form submissions, and updating content without reloading the page.

HTML5 and CSS3:
Essential for structuring and styling the web pages to ensure a responsive, user-friendly interface.

jQuery Form Builder:
Enables organizers to dynamically build event-specific registration forms.

SMTP or Mail Library:

Required to send email notifications, including approval/rejection alerts for organizers and participants.

10.2 Minimum Hardware Requirements

Processor (CPU):
A modern multi-core processor (e.g., Intel Core i5 or equivalent) is recommended to efficiently handle simultaneous user requests and backend processing.

RAM:
A minimum of 8 GB RAM is recommended to manage database queries, process dynamic content, and handle concurrent users without performance lags.

Storage:

Adequate storage space is needed for storing event details, participant information, uploaded files (e.g., event posters, payment proof), and system resources.

Internet Connection:
Required for handling online registrations, sending email notifications, and performing real-time database updates.

Display:
A monitor with at least a 1080p resolution is recommended for optimal viewing during system administration, event creation, and customization processes.

11. REFERENCES

1. J. R. V. Jeny, P. Sadhana, B. J. Kumar, S. L. Abhishek and T. Sai Chander, "A Web based-College Event Management System and Notification Sender," 2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2022, pp. 1434-1438, doi: 10.1109/ICIRCA54612.2022.9985774.

2. R. F. Olanrewaju, B. U. I. Khan, M. A. Morshidi, F. Anwar and M. L. B. M. Kiah, "A Frictionless and Secure User Authentication in Web-Based Premium Applications," in IEEE Access, vol. 9, pp. 129240-129255, 2021, doi: 10.1109/ACCESS.2021.3110310.

3. K. T. Fathimath Hanan, T. Henna Sherin, U. Lulu Shadin, K. C. Dilshad and A. P. Sulthana Rinsy, "Streamlined Application for Managing College Events," 2024 International Conference on Innovations and Challenges in Emerging Technologies (ICICET), Nagpur, India, 2024, pp. 1-6, doi: 10.1109/ICICET59348.2024.10616323..

4. PHP Official Documentation - https://www.php.net/manual/en/

5. MongoDB with PHP Official Documentation - https://www.mongodb.com/docs/languages/php/