CampusEventHub – Inter-College Event Management Platform

1. Abstract

CampusEventHub is a full-stack web application that helps digitalize and manage inter-college events. It acts as a central platform where colleges can host cultural fests, hackathons, workshops, and sports competitions, while students can explore, register, and participate easily.

The platform provides secure login with role-based access for students, college admins, and super admins. It includes event creation, registration management, QR-based ticket generation, email notifications, and analytics for better event control.

CampusEventHub reduces manual work, improves transparency, and makes event organization and participation more efficient and accessible.

2. Introduction

CampusEventHub serves as a centralized platform designed to streamline event management within colleges. It enables students to browse and register for upcoming events, while administrators can efficiently create and manage event details. The platform further incorporates registration management, analytical insights, and system-wide supervision by a Super Admin.

Milestones:

- **Milestone 1:** Implementation of secure registration and login functionalities for students and administrators, featuring role-based dashboards.
- **Milestone 2:** Development of event creation capabilities for administrators, along with event listing and filtering options for students.
- **Milestone 3:** Integration of a registration workflow with an approval system, a Super Admin dashboard, analytics tools, and QR-based ticket generation.
- **Milestone 4:** Development of a feedback and interaction system with user comment sections, event rating features, and an admin dashboard for analyzing feedback and engagement metrics.

Outputs: Registration and login interfaces, role-based dashboards, event creation form, event listing page, registration management panel, analytics dashboard and feedback and interaction system.

3. Project Scope

Included:

- **Student Module:** Secure registration and login, personalized dashboard, event browsing and filtering, registration submission, ticket download, notifications, feedback submission, and comment sections.
- **Admin Module:** Role-based secure login, event creation, listing and management, registration approval or rejection, analytics dashboard, participant tracking and feedback analysis.
- **Super Admin Module:** System-wide control with user and event management, analytics overview, monitoring of college activities, and supervision of event feedback and comment interactions.

Limitations:

CampusEventHub is currently deployed as a proof of concept using free-tier cloud hosting, effectively demonstrating its core functionalities. With MongoDB ensuring reliable data management, future development will focus on optimizing performance, enhancing scalability for large-scale deployments, and expanding the analytics module for deeper insights into user engagement and event performance.

4. Requirements

Functional Requirements:

- User registration and login with role-based access (Student, College Admin, Super Admin).
- Secure authentication using encrypted credentials.
- Event creation and management by college admins.
- Event browsing and filtering by students.
- Registration workflow with approval or rejection by admins.
- Automatic QR-based ticket generation for approved registrations.
- Email notifications for registration approvals and ticket delivery.
- Feedback and comment system for students to share event experiences.
- Event rating feature to evaluate overall event quality.

- Admin dashboard for analyzing feedback, ratings, and engagement metrics.
- Analytics dashboard for viewing event participation and statistics.
- Super Admin dashboard for system-wide monitoring and control.
- Activity logging for tracking administrative actions.

Non-Functional Requirements:

- Data security through authentication and authorization control.
- Simple, user-friendly, and responsive interface across devices.
- Consistent performance and reliability during concurrent usage.
- Scalable structure for integration of future features.
- Maintainable codebase with modular design and clear API endpoints.

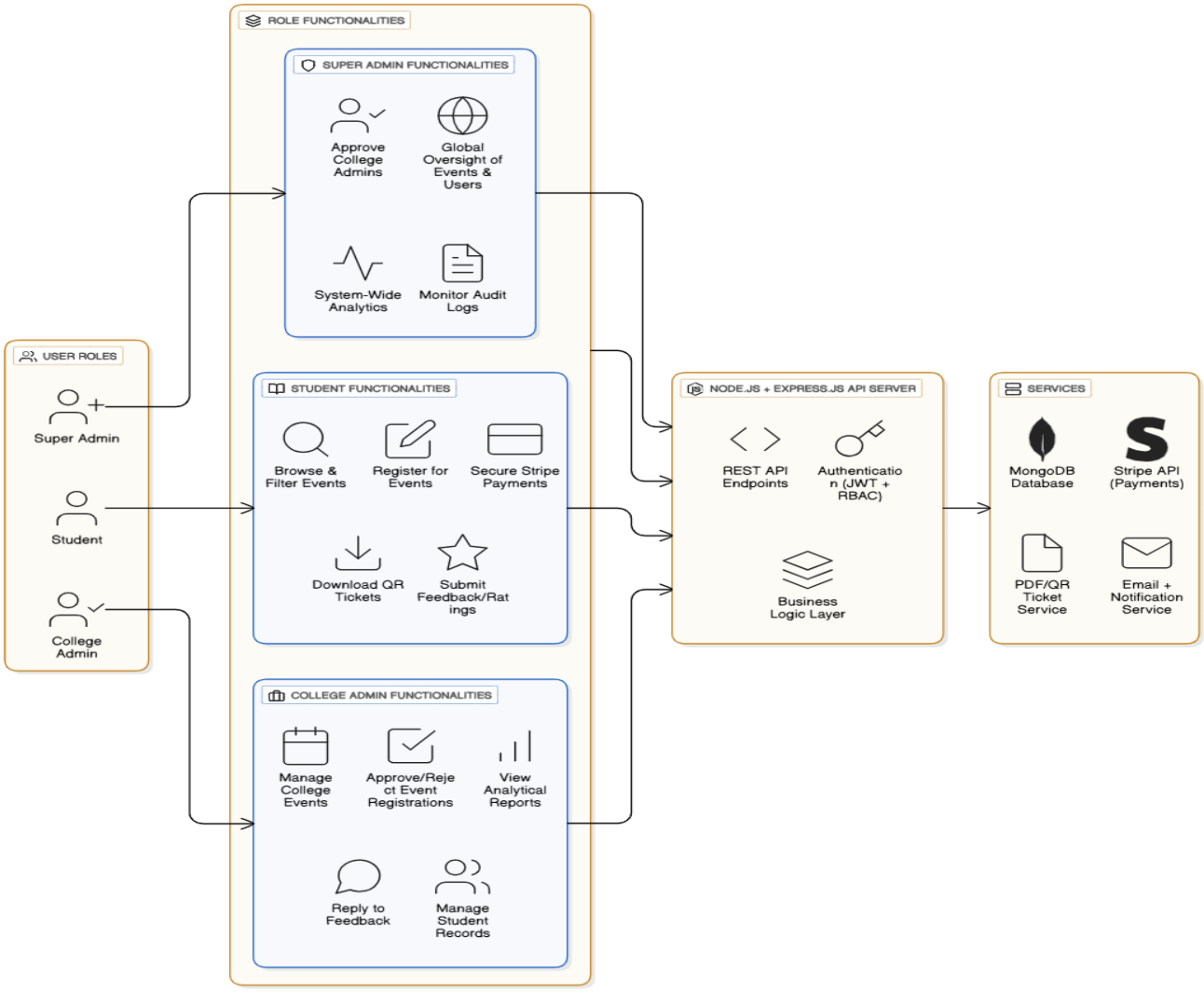
5. Technical Stack

- **Languages:** JavaScript (frontend & backend with Node.js).
- **Frameworks/Libraries:** Express.js (backend API development), React.js (frontend development), Tailwind CSS (responsive design), React Router DOM (client-side routing), Chart.js & React-Chartjs-2 (analytics and data visualization), Lucide React (iconography).
- **Database:** MongoDB.
- **Authentication and Security:** JWT (secure authentication), bcrypt/bcryptjs (password hashing), CORS (cross-origin request management).
- **File and Email Management:** Multer (file uploads), Nodemailer (automated emails), PDFKit and QRCode (ticket generation).
- **Payment Integration :** Stripe (secure online payment processing and transaction management).
- **Environment Management:** dotenv (secure handling of environment variables).
- **Development Tools:** GitHub (version control), VS Code (integrated development environment), Postman (API testing), Nodemon (server auto-restart during development).

- **Tools:** GitHub (version control), VS Code (IDE), Postman (API testing).
- **Deployment:** Free-tier-platform Netlify (frontend), Render (backend), MongoDB Atlas (database).

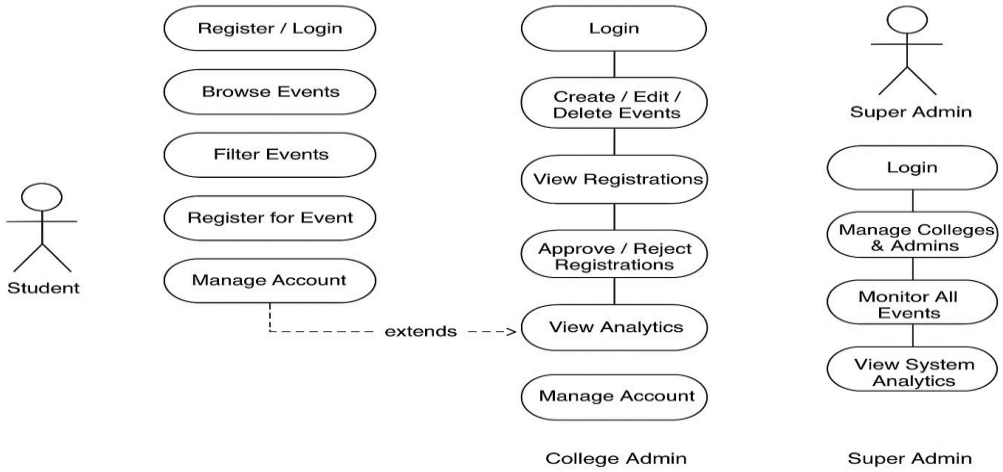
6. Architecture / Design

- **Backend:** Node.js + Express.js with REST APIs, MongoDB for data storage. Handles authentication, event management, registration workflows, analytics, and account management.
- **Frontend:** React.js with responsive, role-based dashboards, real-time event listing and filtering, and analytics visualization (Chart.js & React-Chartjs-2). Includes Settings page for all users.
- **Design Decisions:** RESTful APIs for simplicity, role-based access for security, cloud deployment for accessibility, and modular architecture for maintainability.

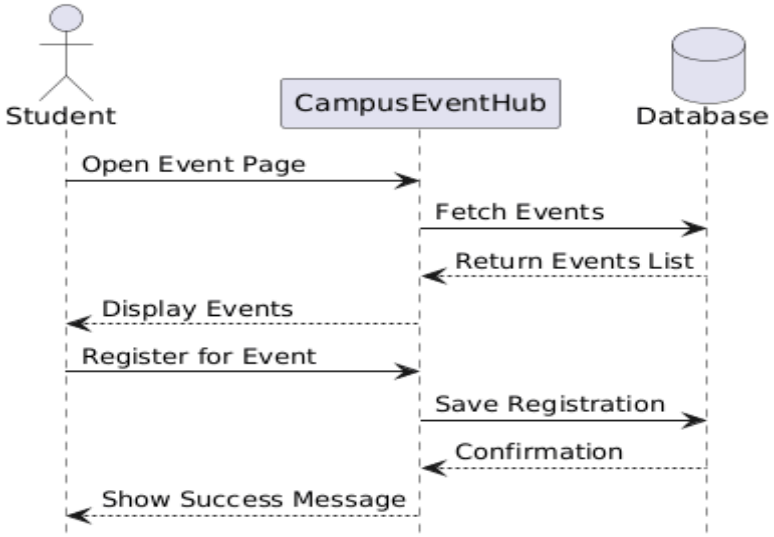


System Architecture

- **UML Diagrams**



Usecase Diagram



Sequence Diagram

7. Development

- **Backend:** Modular structure with routes, controllers, and models. Secure authentication for students, college admins, and Super Admin using JWT and bcrypt. RESTful APIs handle event creation, registration workflows, analytics, and account management.

- **Frontend:** Responsive React components with role-based dashboards. Integrated APIs for event listing, registration workflows, QR-based ticket generation, and account management via the Settings page. Analytics visualized using Chart.js and React-Chartjs-2.

- **Challenges & Solutions:**
    - **Role-based access:** Separate authentication flows and conditional rendering.
    - **Real-time updates:** Managed via React state and asynchronous API calls.
    - **Registration workflow & approvals:** Multi-step registration with admin approval and automated email notifications.
    - **Ticket generation:** PDFKit and QRCode for scannable event tickets.
    - **Account management:** Settings page implemented for all users to update name, email, and password securely.

8. Testing

- **API Testing:** Postman for authentication, event creation, registration workflows, and analytics APIs, and account management endpoints.
- **System Testing:** Comprehensive testing of registration/login, role-based dashboards, event creation and management, event browsing and filtering, registration approvals, QR-based ticket generation, and account updates via the Settings page for all users.
- **Bugs Fixed:** Alignment issues, filtering logic corrections, registration workflow errors, QR/PDF ticket inconsistencies, and account update functionality errors.

9. Deployment

- **Backend:** Node.js server with MongoDB hosted on free-tier cloud platforms. Environment variables managed securely using dotenv. APIs support authentication, event management, registration workflows, and analytics.
- **Frontend:** Built using npm run build and deployed on Netlify/Heroku. Supports responsive, role-based dashboards and integrates with backend APIs for real-time data.
- **Benefits:** Scalable, secure with role-based access, supports automated registration and ticketing workflows, and cost-effective through free-tier services.

10. User Guide

- **Students:** Register and log in, browse and filter events, view event details, register for events, access QR-based tickets, share feedback and comments on events and manage account information (name, email, password) via the Settings page.

- **College Admins:** Log in, create and manage events, approve or reject registrations, edit or delete event listings, view and respond to event feedback, access analytics, and manage account information via the Settings page.

- **Super Admin:** Manage system-wide operations, oversee users and events, monitor event feedback and comment activity, view analytics dashboards, and update account information through the Settings page.

- **GitHub Repository:** CampusEventHub Project – Team1

12. Result & Conclusion

CampusEventHub is a fully functional, secure, and user-friendly event management system that includes authentication, role-based dashboards, event creation and management, event browsing and filtering, registration workflows with admin approval, QR-based ticket generation, account management, and a feedback and comment section for improved user interaction and event evaluation.

The project provides a complete and scalable solution for managing campus events, offering valuable experience in API development, database modeling, responsive frontend design, secure payment integration with Stripe, and implementing feedback analysis for continuous improvement.