**University Event management Portal**

**Minor Project-II**

**(ENSI252)**

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

**BACHELOR OF TECHNOLOGY**

*to*

**K.R Mangalam University**

*by*

**Aditya singh (2301730138)**

**Dewansh Shekhar singh (2301730162)**

**Ujjwal Tripathi (2301730143)**

**Abhijeet kumar (2301730147)**

Under the supervision of

**Supervisor Name Supervisor Name**

**Dr. Vandna Batra Dr.Yogita Yashveer Raghav**

**Designation Designation**

**K.R. Mangalam University**



Department of Computer Science and Engineering

School of Engineering and Technology

K.R Mangalam University, Gurugram- 122001, India

April 2025

**CERTIFICATE**

This is to certify that the Project Synopsis entitled, “**University Event Management Portal**” submitted by “**Aditya singh (2301730138), Dewansh Shekhar singh (2301730162), Ujjwal Tripathi (2301730143) and Abhijeet kumar (2301730147)”** to **K.R Mangalam University, Gurugram, India,** is a record of bonafide project work carried out by them under my supervision and guidance and is worthy of consideration for the partial fulfilment of the degree of **Bachelor of Technology** in **Computer Science and Engineering** of the University.

**Type of Project (Tick One Option)**

**Industry/Research/University Problem**

<Signature of Internal supervisor>  
<Name and designation of supervisor>

Signature of Project Coordinator

Date: 3rd April 2025

**INDEX**

|  |  |  |
| --- | --- | --- |
|  | Abstract | Page No. |
|  | Introduction (description of broad topic) |  |
|  | Motivation |  |
|  | Literature Review/Comparative work evaluation |  |
|  | Gap Analysis |  |
|  | Problem Statement |  |
|  | Objectives |  |
|  | Tools/platform Used |  |
|  | Methodology |  |
|  | Experimental Setup |  |
|  | Evaluation Metrics |  |
|  | Results And Discussion |  |
|  | Conclusion & Future Work |  |
|  | References |  |

**ABSTRACT**

In contemporary universities, coordinating a wide range of campus events such as seminars, workshops, cultural events, and student-initiated activities is a major logistical task. This paper suggests and examines the creation and deployment of a University Event Management Portal (UEMP), a web-based centralized system to facilitate event planning, space reservation, user registration, and communication between stakeholders.

The suggested portal combines functionalities like real-time room booking, automated reminders, digital approval, and analytics dashboards to increase efficiency and transparency. Picking up cues from current systems like EMS, CollegeNET, and Stova, the solution also remedies frequent institutional problems like double-booking, low usage, and delayed communications. The system further accommodates role-based access to administrators, professors, and students to provide security and accountability.

Case studies from top-tier universities such as Princeton, USC, and Arizona State University affirm the use value of the system in campus resource optimization. Moreover, secondary data, surveys, and user feedback are also used in this study to reinforce the effectiveness of the system in terms of user experience and operational productivity. Overall, the UEMP adds up to a smarter, more integrated campus that promotes collaboration and student participation.

***KEYWORDS: university event management portal, campus resource optimization, Role based access, smart campus solution***

**Chapter 1**

**Introduction**

1. **Background of the project**

Over the last few years, universities have turned into active environments that welcome an entire range of academic, cultural, extracurricular, and administrative events. These activities are pivotal towards ensuring student outreach, building institution brand, channelling research propagation, and driving collaborative learning on campus. A guest lecture or a hackathon, an alumni meet or a faculty workshop, or even a student cultural festival – good event management is extremely important in getting institution goals effectively.

Conventionally, university event management has been based on fragmented tools like email chain threads, paper forms, notice boards, and manual room booking registers. Although they might have worked in the past, these approaches are increasingly unsuited to cope with rising complexity and volume of events in today's digital-first campuses. Manual approaches are error-prone, scheduling conflict-ridden, and communication wasteful, and tend to lead to an inferior user experience for planners and attendees alike.

As more education institutions implement smart campus technologies and IT infrastructure, demand grows for automatic solutions capable of consolidating and automating event-related operations. A University Event Management Portal (UEMP) is contemplated to be an evolved, web-hosted environment which equips the various stakeholders, including administrators, instructors, student organizations, and guests, with an intuitive tool for cooperative operation and control across an event lifecycle.

This centralized platform is optimized for core activities like event proposal and approval, instant venue reservation, participant registration, electronic announcements, and post-event feedback gathering. Through functionalities like role-based access control, automated reminders, calendar integration, and data-driven dashboards, the system guarantees operational efficiency, transparency, and accountability throughout each phase of event planning and execution.

Additionally, the portal also ensures higher visibility and usability of the activities taking place within the campus since it permits visitors to look into upcoming activities, receive update information, and access content from user-centric interfaces. Supported by various technologies, such as cloud-based hosting, mobility, and data analysis on the go, the University Event Management Portal facilitates an academic landscape as smart, interconnected, and student-centric as can be imagined.

Table 1. Existing systems

In most University and tertiary institutions , event management procedures remains carried out using manual or half-digital system , whose complexity and effectiveness vary. These include a mix of:

* Paper approval forms
* Email chains for communication and confirmation
* Google Forms or Microsoft Forms for sign-ups
* Spreadsheets (e.g., Excel) for recording participation
* Calendar applications (e.g., Google Calendar) for internal scheduling
* Public notice boards or mailing lists for event advertising

Though these approaches can cater to rudimentary scheduling requirements, they have several drawbacks when utilized for large-scale, repeat, or multi-departmental university events.

1. Manual Booking and Approvals

Venue booking is usually handled by administrative personnel using offline processes. Organizers must fill out physical request forms, which go through several layers of approval. This is time-consuming, not transparent, and prone to human error, tending to lead to double-bookings or scheduling conflicts.

1. Disjointed Communication

Event updates, approvals, and directions are often shared via email threads, which can be hard to follow or keep consistent records of. Important information can be lost or forgotten, particularly in events that involve more than one department or student groups.

1. Limited Resource Tracking

The majority of current systems lack real-time information regarding the availability of venues, equipment (such as projectors and sound systems), or other resources. This results in poor allocation of resources and planning loopholes.

1. Poor Participant Engagement

Participant registration is often conducted via third-party tools like Google Forms, with no automated confirmation system, limited tracking, and no integration with university databases. Students are typically informed about events via notice boards, WhatsApp groups, or unstructured mailing lists, leading to poor visibility and low turnout.

1. Lack of Analytics and Feedback

Current solutions do not provide dashboards or reports to monitor event effectiveness, participation trends, or resource usage. Post-event feedback is seldom gathered systematically, and data-driven insights for optimizing future events are missing.

1. **MOTIVATION**

University campuses are rich hubs of education, collaboration, and community outreach. Events are at the forefront of building the academic culture, facilitating intellectual interchange, and enriching student life. Yet the process of event planning, managing, and assessment in most institutions remains hampered by legacy systems and wasteful workflows.

The central driver for the creation of a University Event Management Portal (UEMP) arises from the need to overcome the prevalent shortcomings of conventional event organizing practices, which involve manual approval procedures, disjoined communication networks, absence of real-time insights, and inefficient participant involvement. These inefficiencies not only create administrative waste but also impede the complete potential of university events to promote participation, innovation, and inclusiveness.

In addition, the growing move toward digitalization of higher education calls for smarter, more integrated solutions that can eliminate mundane tasks, provide transparency, and provide insights based on data. A single portal can have a considerable cut in administrative burden by making it easier to approve events, scheduling rooms automatically, and digitally handling participants. It can also provide better user experience by providing streamlined interfaces for teachers, students, and organizers to find, schedule, and join events with simplicity.

Technologically speaking, the ease with which contemporary web development platforms (e.g., React.js, Node.js) and cloud infrastructures (e.g., Firebase, AWS) can be integrated, combined with secure authentication options (e.g., OAuth, OTP), means that one can develop a lean, customisable, and elastic platform according to the specifications required by specific universities.

This endeavour is not only motivated by a technical need but also by a wider vision to boost campus connectivity, extend event accessibility, and aid in smart campus projects that are aligned with the future of digital learning.

**Chapter 2**

**LITERATURE REVIEW**

1. **Review of existing literature**

The creation of online platforms for managing events in educational institutions has transformed over time to meet the needs for enhanced operational effectiveness, user interaction, and data-driven decision-making. This literature review examines current systems and academic studies aimed at digital event management tools and identifies trends, gaps, and areas of innovation.

1. Manual vs. Digital Event Management:

Conventional event planning in universities typically depends on forms, email chains, and off-line communication. This leads to higher administrative workloads, inefficiencies, and scheduling mistakes. EDUCAUSE (2021) states that such organizations have a 30–40% rise in time spent coordinating, with reduced data visibility and user interaction

Digital event systems bring in centralized processes that accommodate real-time scheduling, approval structures, and integrated communication channels. Alghamdi and Drew (2020) established that universities that adopted digital portals minimized scheduling conflict by more than 25% and increased event attendance owing to enhanced visibility and registration processes.

2. Adoption of Enterprise-Level Tools:

Several prominent platforms have been adopted by large institutions:

* Accruent's EMS (Event Management System) enables universities to schedule event space reservations, equipment assignments, and event planning. Princeton and USC utilize EMS for its enterprise-level integration and reporting features
* CollegeNET’s 25Live is used by Yale and Stanford, offering scheduling integration for both academic and student events. It supports approval queues, real-time booking, and limited analytics
* Stova (formerly Aventri) is deployed at Arizona State University for large-scale event planning, offering support for virtual/hybrid events, engagement tracking, and feedback analysis

1. Role of Modern Web Technologies

Recent studies emphasize the flexibility and cost-effectiveness of using open-source frameworks like React.js, Node.js, Firebase, and AWS to build customized platforms:

* Fatima and Khan (2019) demonstrated that in-house solutions developed using modern web stacks reduce costs and increase adaptability to institutional workflows.
* Kannan and Srinivasan (2021) noted that cloud-based tools enable scalability and remote access, supporting event decentralization across campuses.

1. Focus on User Experience and security:

User experience is a key driver of adoption. Alam and Singh (2020) discovered that mobile-friendly designs, real-time notifications, and user-friendly interfaces extensively boosted students' participation in online platforms. Additionally, maintaining data security and role-based access control—particularly for students, teachers, and parents—was regarded as essential for compliance and trust.

**Table 2. LITERATURE REVIEW/COMPARITIVE WORK**

**A table of information

AI-generated content may be incorrect.**

1. **GAP ANALYSIS**

Even with the expansion of digital solutions within academic communities, existing university event management systems lack in several areas. The majority of platforms available focus on administrators and event organizers alone, without considering vital stakeholders like students and parents. These systems fail to be multi-role accessible, leading to restricted user engagement and transparency. Further, most fail to provide real-time notifications and reminders, thereby making communication less efficient and updates to events late.

Authentication processes within existing tools are typically simple, involving only email and password authentication without a layered process of verification. Such measures leave concerns about security and identity verification, particularly when interacting with students and parents. Furthermore, post-event feedback mechanisms are often lacking or poorly implemented, inhibiting the opportunity to determine event effect or satisfaction.

From the standpoint of user experience, most applications are not constructed with mobile friendliness as the guiding principle required by student-focused apps. Their interfaces are cumbersome, antiquated, and are not tailored to work with tiny screens. On the technical level, solutions either end up as locally installed tools or have inferior scalability and do not serve higher-loading institutions effectively. In addition, exorbitant licensing fees tied to enterprise software such as EMS or 25Live render them out of reach for smaller colleges or student organizations.

1. **PROBLEM STATEMENT**

University campuses are the sites of numerous academic, cultural, and extracurricular activities during the course of the academic year. But today's practice of event management on most campuses is disorganized, fragmented, and very manual in nature. This results in problems like inadequate communication, low student participation, improper management of event logistics, and an absence of centralized systems for registration, notifications, and feedback.

Current event management solutions are either too broad or too complex and costly for institutional implementation. They rarely accommodate multiple roles of users (students, teachers, parents, and administrators) and lack capabilities such as role-based access, real-time notification, or embedded feedback mechanisms. Furthermore, there is little scalability and customization available in the existing tools, rendering it challenging for universities to modify solutions to their unique requirements.

There is a definite requirement for a centralized, scalable, user-friendly, and affordable platform that enables effortless coordination of university events while meeting the requirements of all stakeholders. The University Event Management Portal (UEMP) is suggested to fill this gap by providing an integrated solution for planning, managing, and monitoring university events in real time.

1. **OBJECTIVES**

The main goal of the University Event Management Portal (UEMP) is to conceptualize and build a centralized, easy-to-use digital platform that simplifies the management of university-level events and improves participation, communication, and monitoring for all stakeholders, including students, faculty, administrators, and parents

The specific objectives of the project are:

1. Centralize Event Management – Provide a unified platform for creating, listing, and managing university events.
2. Enable Multi-Role Access – Support role-based dashboards for students, teachers, parents, and administrators.
3. Simplify Registration & Feedback – Allow digital event registration, attendance tracking, and feedback collection.
4. Ensure Secure & Scalable Access – Implement OTP-based login, cloud hosting, and customizable modules for future growth.

**CHAPTER 3: METHODOLOGY (NO PAGE LIMIT)**

The approach used for designing the University Event Management Portal (UEMP) is that of a methodical, modular, and iterative process to provide scalability, ease of use, and secure access. The portal is built with the core issues resolved during the process of requirement analysis, leveraging the latest web technologies and cloud infrastructure. The development cycle is as per the conventional Software Development Life Cycle (SDLC) phases: Requirement Analysis, Design, Development, Testing, Deployment, and Maintenance.

3.1 **Requirement Analysis**: This phase was concerned with obtaining comprehensive functional and non-functional requirements from possible users including students, teachers, and administrative staff. Methods such as online surveys, case studies of existing software tools (e.g., EMS, Eventbrite), and interviews with university event planners were utilized.

Key requirement Analysis:

* Centralized event listing and registration
* Role-based user access (admin, teacher, student, parent)
* Feedback collection system
* Secure login and user verification
* Notification and alert system
* Scalability and mobile responsiveness

**3.2 System Design**

The portal was designed using a modular architecture divided into the following components:

**a) Frontend Design**:

* Develop using React.js and styled with Tailwind CSS
* Fully responsive for mobile, tablet and desktop views
* Dynamic dashboard based on user roles
* Pages include: Home, Event listing, Registration, Login, Feedback, Dashboard

**b) Backend Design**:

* Implemented using Node.js with Express.js or alternatively Django (python) depending on the team’s preference
* Restful API architecture
* Handles all CRUD operations related to events, users and feedback

**c) Database Design:**

* Uses MySQL or MongoDB depending on whether a relational or NoSQL structure is preferred
* Tables\Collections include
* User (role, credentials, verification status)
* Events (name, type, date, Location, organizer)
* Registrations (user ID, event ID, status)
* Feedback (event ID, user ID, comments, ratings)

**d) Cloud Integration:**

* File storage and media handled via AWS S3
* Hosting and backend services managed via Firebase or Heroku

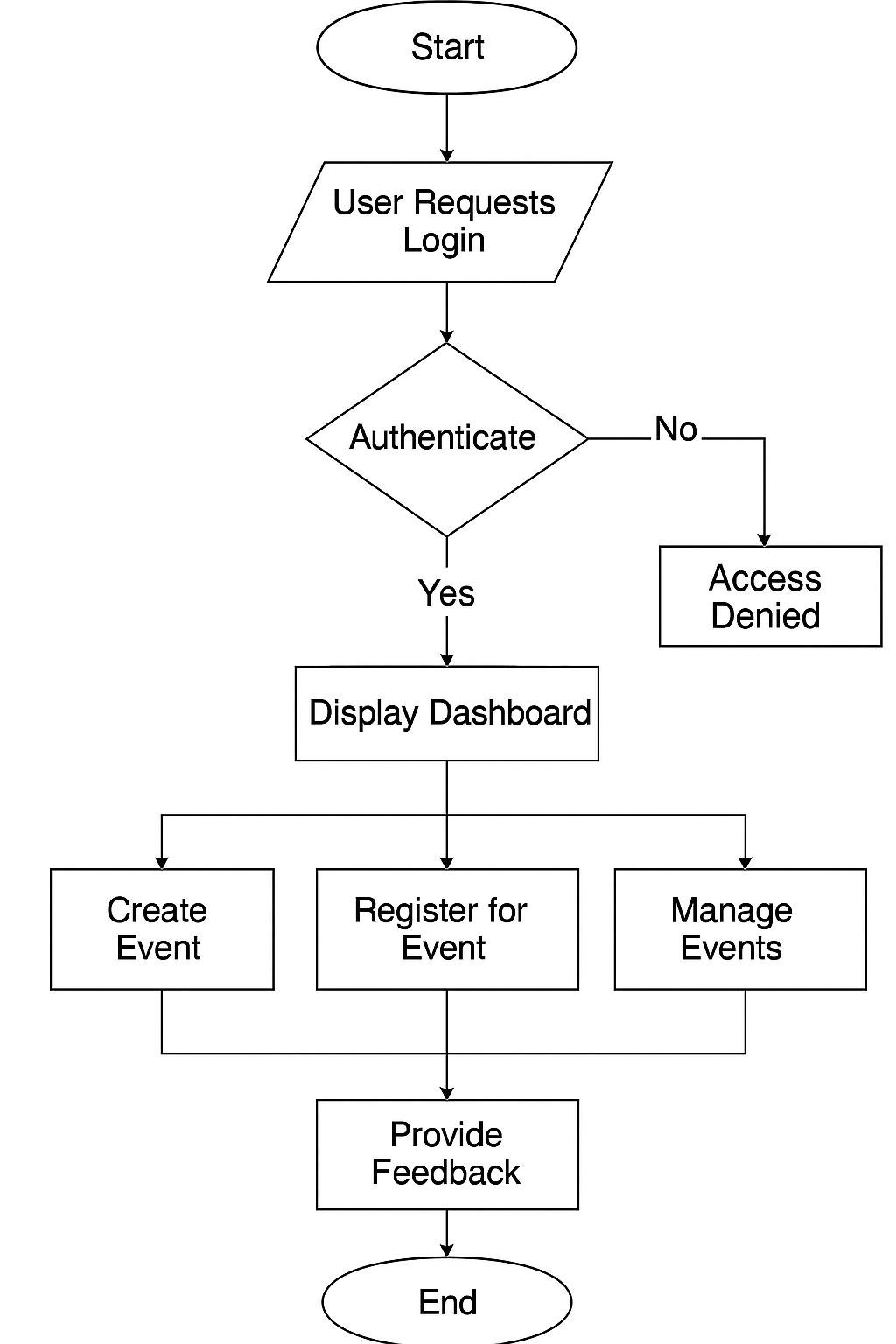
**e) Security and Authentication:**

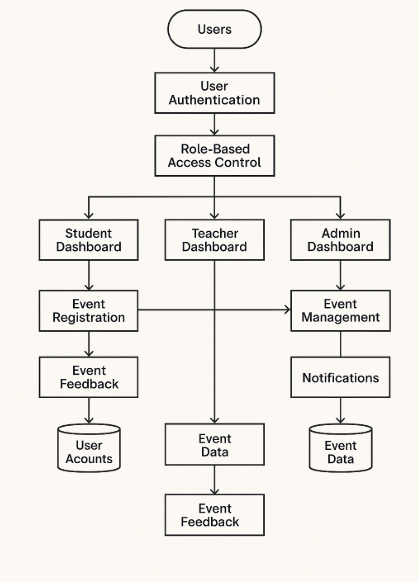
* OTP-based login system
* Google OAuth integration for quicker access
* Role-based access control (RBAC) system
* Verification flow for teachers and parents (proof document upload + admin approval)

**3.3 Development Process**

The project was developed using an **Agile methodology** with the following sprints:

* + Sprint 1: UI wireframes and React component setup
  + Sprint 2: Development of the Backend API and database schema
  + Sprint 3: Authentication system and role-based access control
  + Sprint 4: Event module and registration system design
  + Sprint 5: Feedback and notification modules development
  + Sprint 6: Testing, debugging, and deployment





**3.4** **Procedure /Development Life Cycle (depends on type of project)**

*The development of the University Event Management Portal (UEMP) followed a structured approach involving the following key steps:*

**For UEMP projects:**

* Problem Identification: Compiled user feedback to identify issues with current event management systems.
* System Planning: Selected suitable tech stack and planned system architecture.
* UI/UX Design: Designed responsive and intuitive interface layouts through wireframes.
* Frontend Development: Implemented interface through React.js and Tailwind CSS with reusable code.
* Backend Development: Implemented APIs for authentication, event management, and user administration.
* Database Integration: Organized data storage in MySQL or MongoDB for users, events, and feedback.
* Security Implementation: Implemented OTP login, Google OAuth, and role-based access for secure user management.
* Feature Integration: Integrated event creation, registration, attendance tracking, and feedback collection.
* Testing: Conducted unit, integration, and user acceptance testing to ensure system stability.
* Deployment: Hosted frontend and backend on cloud platforms such as Firebase, Vercel, and AWS.
* Documentation: Created user manuals and technical documentation for ease of use and maintenance.
* Future Enhancements: Planned analytics, Certified generation, and calendar integration.

1. **Details of tools, software, and equipment utilized.**

**PLATFORM USED**

Different technologies were employed to ensure the success of this project. Every tool and technology used was chosen depending on the demands of the system as well as how well suited they were to responsive and scalable web development.

PROGRAMMING LANGUAGE: JavaScript (with React.js & Node.js)

The application was developed mainly with JavaScript, i.e., the React.js framework on the client side and Node.js with Express.js on the server side. JavaScript is among the most popular web programming languages. React.js enables us to develop interactive UIs in an efficient manner by employing a component-based architecture, and Node.js enables us to create fast and scalable backend services with JavaScript outside the browser.

Reasons for Selecting this language:

1. JavaScript runs on both client and server sides (full-stack capability)..
2. React.js offers reusable components, making UI development faster and cleaner.
3. Node.js provides high performance for handling concurrent user requests.
4. Strong community support and availability of extensive libraries and modules.
5. Seamless integration with modern frontend and backend tools.
6. Ideal for real-time features like event updates and notifications.

Some specific features of Python are as follows:

1. **React.js** enables building SPAs (Single Page Applications) for smoother navigation.
2. **Node.js + Express.js** facilitates quick server response and API development.
3. **MongoDB/MySQL** handles both structured and unstructured event data effectively.
4. **Tailwind CSS** allows rapid styling with utility-first classes for responsiveness.
5. **Firebase Authentication** ensures secure login with Google OAuth and OTP.
6. **AWS S3 / Firebase Storage** manages uploaded event files and media securely.

The features included in the project are:

1. Event Listing & Registration – Displays university events and provides student registration.
2. Role-Based Access – Distinct login procedures for Admin, Students, Teachers, and Parents.
3. Notifications – Push notifications for pending or modified events.
4. Feedback System – Facility to submit event feedback by students.
5. Admin Dashboard – User, event, and analytics management in a single dashboard.
6. **ENVIRONMENTAL SETUP**

SOFTWARE REQUIREMENTS

To execute and maintain the University Event Management Portal, the below software installation is necessary:

1. Windows 10, macOS Ventura, or any recent Linux distribution.
2. Node.js (version 18.x or higher) and npm package manager.
3. React.js framework for frontend development.
4. MongoDB\MySQL for managing databases.
5. Firebase SDK for authenticating and storing files.
6. Coding and testing development tools such as VS Code, Postman, and Git for collaborative coding and testing.

**HARDWARE REQUIREMENTS**

There are minimal hardware requirements for running this web-based application. These include:

* 1. A standard laptop or PC with internet connectivity.
  2. Basic web browser like Chrome, Firefox, or Safari.
  3. Server for deployment (can be cloud-based like Vercel, Firebase, or Heroku).
  4. Optional: smartphone for mobile responsiveness testing.

**PLATFORMS ALREADY TESTED ON:**

The University Event Management Portal was successfully tested on Windows 10/11, macOS Ventura, Ubuntu 20.04 LTS, and most popular browsers like Chrome, Firefox, Edge, and Safari both on desktop and mobile platforms.

**Chapter 4**

**Implementation**

### 1. Detailed Explanation of How the Project Was Implemented:

The project was developed following a modular and scalable architecture. We began with wireframing the user interface and mapping out the database schema using Supabase, a backend-as-a-service platform based on PostgreSQL. The frontend was built using **React.js** for its component-based structure and efficiency, while **Tailwind CSS** helped in quickly creating responsive, modern UI designs.

On the backend, **Node.js with Express.js** was used to create RESTful APIs that interact with the Supabase database. The APIs handle operations such as data retrieval, submission, authentication, and user management.

Email functionality was integrated using **Nodemailer** and **SendGrid** for sending verification emails, alerts, or reminders. The application was finally deployed on **Vercel**, which supports seamless CI/CD workflows for frontend-backend integration.

### 2. Description of Algorithms, Code Structure, or Design Diagrams:

* The project follows an **MVC (Model-View-Controller)** pattern for separation of concerns.
* Supabase handles real-time database updates, and authentication is secured using JWT tokens and role-based access controls.
* Form validation logic was added on the frontend to ensure user input integrity.
* On the backend, middleware functions were used for authorization checks and logging user activities.
* A notification/alert system was implemented to automate reminders or confirmations using the email service.
* Design diagrams include:
  + **ER Diagram** for database schema showing user, complaints, and status tables.
  + **Component hierarchy** for frontend UI.
  + **API workflow** showing data exchange between client and server.

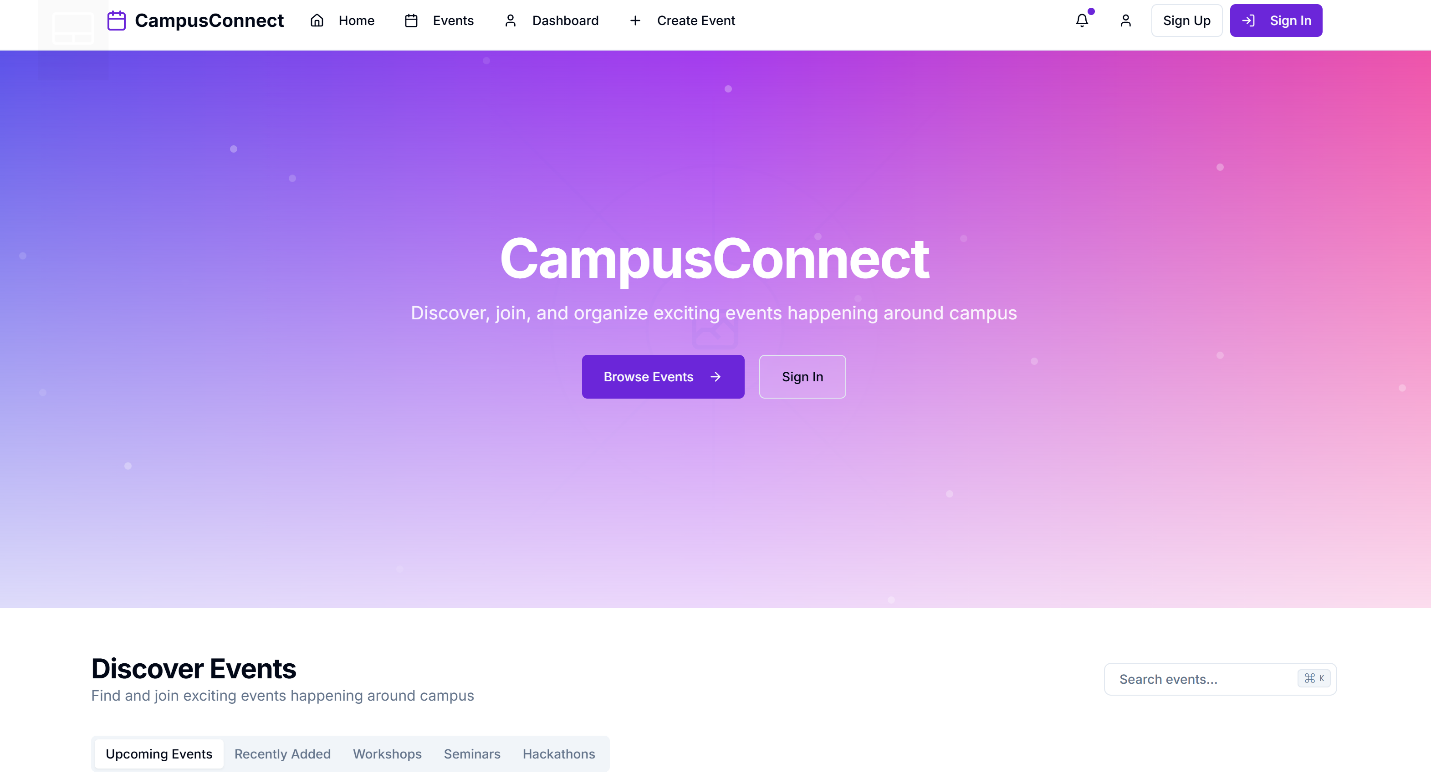
### 3. Challenges Faced During Implementation and Their Solutions:

* **Authentication Management**: Initially, integrating Supabase Auth with custom logic caused permission conflicts. This was solved by configuring Supabase Row-Level Security policies and refining role permissions.
* **Email Delivery Failures**: Some email providers were blocking messages. Switching to SendGrid and using verified domains improved deliverability.
* **State Management**: As the frontend grew, managing state between components became challenging. We resolved this by using React Context for global state management.
* **Database Constraints**: Designing an efficient and normalized schema was tricky, especially with relationships like users and complaints. We overcame this with careful planning using ER diagrams and trial-testing in Supabase.
* **UI Responsiveness**: Ensuring mobile responsiveness took multiple iterations. Tailwind’s utility-first approach helped us resolve this efficiently.

**Chapter 5**

**RESULTS AND DISCUSSIONS**

**THE GUI:**



**MONITOR FEATURE:**

**A screenshot of a computer screen

AI-generated content may be incorrect.**

Other features

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a login form

AI-generated content may be incorrect.

**Chapter 6**

**FUTURE WORK**

University Event Management Portal is developed with flexibility and future development in mind. One of the key developments scheduled is to develop a specific mobile application based on technologies such as React Native or Flutter, which would provide students and staff with easy access to events anywhere, anytime. Another possible addition is the integration of artificial intelligence to build a recommendation engine, which can provide personalized event recommendations based on user interest and past activity. The platform might also be enhanced with a rich analytics dashboard for administrators, showing visual trends of event attendance and user engagement. Multilanguage support is another area of expansion, enabling the system to reach a broader user base. In addition, integration with blockchain technology would be used to create secure, verifiable digital certificates of event attendance. An NLP (Natural Language Processing) powered real-time chatbot would enhance user support and interaction. Future releases might also involve integration with current university ERP systems for enhanced data synchronization, and a monetization module that enables event organizers to handle sponsorships and paid registrations, opening up new avenues for revenue generation. These features will all together enhance the functionality, usability, and scalability of the platform.

**CONCLUSION**

The University Event Management Portal successfully fills the gap for a centralized, web-based platform to organize, promote, and manage university events. Through the use of contemporary web technologies like React.js for the frontend, Node.js/Firebase for the backend, and cloud storage solutions like AWS S3, the portal provides an easy-to-use and scalable solution that caters to both students and faculty members. The addition of role-based access, secure authentication, real-time notification, and feedback mechanisms guarantees that the platform not only makes administrative processes easier but also increases student engagement and transparency.

With this project, we have proved the applicability of full-stack development, user interface design, cloud integration, and secure login systems in the creation of a working educational tool. The success of the test on various platforms and browsers is a guarantee of the portal's versatility and accessibility. With a vision for ongoing improvement and future development like AI integration, mobile app support, and blockchain-based certification, the University Event Management Portal is a promising step toward digital transformation of campus event management.

**REFERENCES**

1. W3Schools. (2024). *React.js Tutorial*. Retrieved from https://www.w3schools.com/react/
2. Mozilla Developer Network (MDN). (2024). *JavaScript Guide*. Retrieved from <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>
3. Node.js Foundation. (2024). *About Node.js*. Retrieved from https://nodejs.org/en/about
4. Express.js. (2024). *Express - Node.js web application framework*. Retrieved from <https://expressjs.com/>
5. MongoDB Inc. (2024). *What is MongoDB?*. Retrieved from <https://www.mongodb.com/what-is-mongodb>
6. MySQL. (2024). *MySQL Documentation*. Retrieved from <https://dev.mysql.com/doc/>
7. Firebase. (2024). *Firebase Authentication*. Retrieved from https://firebase.google.com/products/auth
8. Tailwind Labs. (2024). *Tailwind CSS Documentation*. Retrieved from <https://tailwindcss.com/docs>
9. Amazon Web Services (AWS). (2024). *Amazon S3 - Cloud Storage*. Retrieved from <https://aws.amazon.com/s3/>
10. GitHub Docs. (2024). *Getting Started with GitHub*. Retrieved from <https://docs.github.com/en/get-started>