**SYNOPSIS**

**Report on**

**EVENTIFY(Event Management System)**

**by**

Aakrti Sharma 2300290140001

Akshat Gautam 2300290140005

Anchal Tyagi 2300290140023

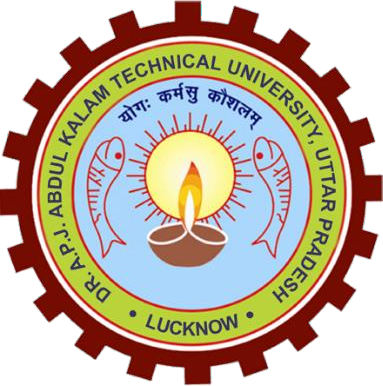
Anuj Singh 2300290140029

### Session:2024-2025 (III Semester)

Under the supervision of

### Prof. Mr.Amit Kumar Gupta

**KIET Group of Institutions, Delhi-NCR, Ghaziabad**

****

**Department Of Computer Applications**

**KIET GROUP OF INSTITUTIONS, DELHI-NCR, GHAZIABAD-201206**

(2023)

**ABSTRACT**

Eventify is a user-friendly web app designed to make organizing and managing college events seamless. From cultural fests and tech symposiums to workshops and club activities, Eventify enables students, faculty, and event coordinators to plan, manage, and execute events

effortlessly. The app’s intuitive features allow for collaboration, customization, and smooth communication, ensuring successful events every time.

# TABLE OF CONTENTS

Page Number

|  |  |
| --- | --- |
| 1. Introduction | 4 |
| 1. Literature Review 2. Project / Research Objective | 5  6 |
| 4. Hardware and Software Requirements | 9 |
| 5. Project Flow/ Research Methodology | 11 |
| 6. Project / Research Outcome | 13 |
| 7. Proposed Time Duration | 14 |
| References/ Bibliography | 16 |

**Introduction**

In a dynamic college environment, numerous events, workshops, and activities are organized regularly. However, students and faculty often face challenges in keeping track of these events, resulting in low participation and missed opportunities. *Eventify* is a web-based event management system designed to address this issue by consolidating all campus events into a single, user-friendly platform.

The goal of *Eventify* is to provide a centralized hub where users can easily explore, register for, and participate in college events. The platform allows event organizers to create and manage event listings, while students can browse events based on their interests, receive notifications, and keep track of their participation history. Through an intuitive and accessible interface, *Eventify* aims to increase student engagement and streamline the event management process for organizers.

## **Literature Review**

Traditional event management on college campuses has largely relied on manual methods like notice boards and social media, which often result in inefficiencies and low participation. Web- based platforms such as *Eventbrite* and *Meetup* have improved event organization through

features like online registration and event calendars, but these systems are often generalized and not tailored to the specific needs of college environments.

Some institutions have developed campus-specific event portals, such as *CampusGroups*, which allow better event management through features like attendance tracking and reminders.

However, these systems often lack customization, ease of use, or mobile accessibility, which are critical for student engagement. Despite these advancements, current solutions still face challenges, such as fragmented communication and lack of integrated feedback mechanisms.

**Eventify** fills these gaps by offering a comprehensive, college-specific platform that centralizes event discovery, registration, reminders, and feedback, providing a more effective and engaging experience for students and organizers alike.

## **Project / Research Objective**

**Eventify** app, focus on:

**1. Target Audience**

College Students**:**

Students looking to participate in a variety of events, ranging from cultural festivals to technical workshops and webinars.

Need a single platform to explore events, register easily, and track their schedules.

Organizers:

Event managers, faculty coordinators, or student committees who want a hassle-free way to manage registrations, payments, and participant lists.

Often face challenges like mismanagement of participant data, unclear communication, or delays in event preparation**.**

**2. Core Features**

Event Creation:

Organizers can create and publish events with custom details like event categories, venue, date/time, and participant capacity.

Event Registration:

Provide step-by-step registration for users, including fields for personal details and preferences.

Auto-generate a unique registration ID or QR code for participants to use during event check-ins

Notifications:

Enable real-time notifications for changes in schedule or venue.

Notify participants about event milestones, such as registration openings, payment deadlines, or event reminders**.**

User Roles:

***Admin*:** Manages overall platform settings and oversees all events.

***Organizer***: Creates and manages their own events, monitors participant activity.

***Participant***: Registers, pays, and receives updates for events.

**3. Tech Stack**

Frontend:

A clean and responsive design ensuring users can easily navigate event listings and register.

Backend**:**

Well-structured APIs to handle user data securely and efficiently, ensuring smooth event management.

Database:

Store event details, user data, and payment records with proper indexing to ensure quick query responses**.**

**4. Security**

Data Privacy:

Protect sensitive user data like phone numbers, emails, and payment details by limiting unnecessary exposure.

Authentication:

Include account recovery mechanisms like email or phone-based OTP verification to enhance user trust.

Error Handling:

Graceful handling of invalid inputs during registration.

**5. Scalability**

Efficient Resource Management:

Otimize database usage by archiving old event data while keeping recent recordsreadily accessible.

Smooth User Experience:

Design APIs to handle multiple concurrent requests, ensuring no slowdowns during peak traffic times (e.g., event registrations).

**6. User Experience (UX)**

Search and Filters:

Add filters based on event type (e.g., Technical, Cultural), date, or department to improve navigation**.**

Interactive Dashboards:

For participants:Show a personalized dashboard with their registered events, payment status, and upcoming deadlines**.**

For organizers: Offer analytics on participant count, revenue generated, and engagement metrics.

Feedback Integration**:**

Allow users to provide ratings and comments post-event to help organizers improve the quality of future events.

## **Hardware and Software Requirements**

For developing and running **Eventify**, the following hardware and software requirements are essential:

#### Hardware Requirements:

##### Development Machine:

* 1. **Processor**: Intel i5 or higher.
  2. **RAM**: Minimum 8 GB.
  3. **Storage**: SSD with at least 256 GB for faster operations
  4. **Internet**: Reliable internet connection for testing, version control (Git), and deployment

1. **Server** (if self-hosting):
   1. **Processor**: 4 cores or higher.
   2. **RAM**: Minimum 4 GB (8 GB for better performance)
   3. **Storage**: SSD with 50 GB or more for storing data and backups
   4. **Network**: High-speed connection with low latency for handling requests

#### Software Requirements:

##### Development Tools:

* 1. **Operating System**: Windows 10/11.
  2. **Code Editor**: Visual Studio Code.
  3. **Version Control**: Git with GitHub/GitLab for collaboration and source control
  4. **Package Manager**: npm (Node Package Manager) for managing project dependencies

#### Frontend:

* 1. **React.js**: JavaScript framework for building the user interface
  2. **HTML5/CSS3**: For structuring and styling the app
  3. **Browser**: Chrome, Firefox (for testing and debugging).

#### Backend:

* 1. **Node.js**: JavaScript runtime for backend development
  2. **Express.js**: Web framework for building APIs
  3. **Database**: MongoDB.

# Project Flow for Eventify

### Requirement Gathering:

* 1. Identify key features: event creation, registration, notifications.
  2. Gather user needs (college students, organizers, admins).

### Planning:

* 1. Define project scope and phases (basic MVP, feature expansion).
  2. Decide on the technology stack (React, Node.js, MySQL/MongoDB).
  3. Set up timelines and milestones.

### Design:

* 1. UI/UX design: Create wireframes and prototypes focused on intuitive navigation and mobile responsiveness.
  2. Database design: Define schemas for users, events, and payments.

### Development:

* 1. **Frontend**: Build user interfaces with React.
  2. **Backend**: Develop APIs using Node.js and Express, connect to MySQL/MongoDB.
  3. **Integrations**: Implement payment gateway (Stripe/Razorpay), email/SMS notifications.

##### Testing:

* 1. Perform unit and integration testing using tools like Jest or Postman.
  2. User acceptance testing (UAT) with real users for feedback.

##### Deployment:

* 1. Deploy on platforms like Heroku or AWS.
  2. Monitor app performance and security.

##### Feedback & Iteration:

* 1. Collect user feedback and refine the app with updates and bug fixes.

# Project Outcome

##### Functional Event Management Platform:

* 1. A fully functional web app allowing users (students, organizers) to create, manage, and participate in college events seamlessly.
  2. Key features include event creation, registration, notifications, and payment integration.

##### Improved Event Coordination:

* 1. Simplified event management for organizers with automated registrations, reminders, and participant tracking.
  2. Improved user experience for students with a clear and intuitive interface.

##### Secure and Scalable System:

* 1. Robust security features, including user authentication and data protection.
  2. Scalable infrastructure capable of handling multiple events and users simultaneously.

##### Mobile-Responsive Interface:

* 1. The app is designed to work smoothly across desktop and mobile platforms, improving accessibility for users on the go.

##### User Feedback Loop:

* 1. The app will have features like post-event feedback and ratings, allowing continuous improvement based on user input.

# Proposed Time Duration

###### Week 1: Planning, Research, and Design

* **Requirement Gathering** (1-2 days): Identify essential features (event creation, registration, basic notifications).
* **Tech Stack Finalization** (1 day): Choose technologies (React, Node.js, MySQL).
* **UI/UX Design** (2-3 days): Create simple wireframes and a basic prototype.
* **Database Design** (1 day): Define essential schemas for users and events.

###### Week 2: Frontend and Backend Development (MVP)

* **Frontend Development** (3 days):
  + Build basic UI with React: event listing, registration form, home page.
* **Backend Development** (3 days):
  + Set up Node.js/Express for handling events and user registration.
  + Connect to MySQL for data storage (event and user tables).
* **Basic User Authentication** (1 day): Implement a simple login and registration system.

###### Week 3: Feature Completion and Integration

* **Event Management** (3 days):
  + Implement event creation and management for organizers.
* **Basic Notifications** (2 days):
  + Set up email or SMS notifications for event registration (use basic email APIs like SendGrid).
* **Testing** (2 days):
  + Unit testing of core features using Postman or similar tools.

###### Week 4: Deployment and Final Adjustments

* **Deployment** (2 days):
  + Deploy the app on Heroku or AWS.
* **Bug Fixing and Testing** (2 days):
  + Final round of testing, fix critical bugs.
* **Feedback and Iteration** (2 days):
  + Gather initial feedback and make small improvements.

##### Scope Adjustments for Time Constraints:

1. **Postpone advanced features** like payment integration and detailed notifications.
2. **Focus on core functionality**: event creation, registration, and basic notifications.
3. **Rapid UI design**: Keep the interface simple and functional.

# References/ Bibliography

1. **W3Schools**. (n.d.). *React.js Tutorial*. W3Schools.com
2. **MDN Web Docs**. (n.d.). *Introduction to Express.js*. [MDN Web Docs](https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs)
3. **React.js Documentation**. (n.d.). Reactjs.org
4. **Node.js Documentation**. (n.d.). Nodejs.org
5. **MySQL Documentation**. (n.d.). [MySQL.com](https://dev.mysql.com/doc/)
6. **MongoDB Documentation**. (n.d.). [docs.mongodb.com](https://docs.mongodb.com/)