**ABSTARCT**

The College Event Organizer is a web-based application designed to streamline event management for college campuses. The platform connects event organizers with students, allowing them to easily create, view, and register for events. Organizers can post events, specifying details such as the event title, description, date, time, venue, and interest categories. Users can browse events based on their selected interests and register for them, receiving email confirmations and reminders about upcoming events. This project contains four key modules: 1. User Management User Registration & Authentication: Handles sign-up, login, and session management (using JWT). User Preferences: Allows users to select and manage their interests (e.g., AI, Sports). 2. Event Management Event Creation: Allows event organizers to post new events (title, description, date, location, category). Event Listings: Displays a list of upcoming events for users to browse. Event Registration: Allows users to register for events and stores their details. 3. Notifications Interest-Based Notifications: Sends email notifications to users based on their interests when matching events are posted. Event Reminders: Sends reminder emails (e.g., 1 day or 1 hour before the event). Email Confirmation: Sends email confirmations when a user registers for an event. 4. Google Maps Integration Event Location: Shows the event location on Google Maps. Navigation: Provides real-time directions from the user's location to the event venue.

1. **INTRODUCTION**

The **College Event Organizer** is an interactive web-based application developed to streamline the planning, promotion, and participation of events within a college campus. It addresses the common challenges faced in traditional event coordination, such as lack of centralized information, poor communication, and limited student outreach. By bringing together event organizers and students on a single platform, the system ensures a smoother and more engaging experience for all participants.

This application allows event organizers to efficiently create and manage events by specifying essential details including the event name, description, date, time, venue, and relevant categories. Once published, these events are made visible to students, who can browse upcoming activities and register based on their interests.

**Main Features and Modules:**

1. **User Management**
   * Allows users to sign up, log in, and manage their profiles securely.
   * Supports selection and customization of personal interests to tailor the event experience.
2. **Event Management**
   * Enables event organizers to create detailed event posts.
   * Displays a curated list of upcoming events for users to explore and register for.
   * Stores registration data for future reference and communication.
3. **Notification System**
   * Sends confirmation emails upon successful registration.
   * Delivers timely reminders before each registered event.
   * Notifies users when new events related to their selected interests are posted.
4. **Google Maps Integration**
   * Shows exact event locations through an embedded map view.
   * Offers live navigation guidance to help users reach the venue from their current location.

The College Event Organizer not only simplifies the logistics of event management but also fosters a stronger sense of community by ensuring students are aware of and engaged in campus activities that align with their passions and goals.

1. **PROPOSED METHOD AND IMPLEMENTATION**

The main objective of the **College Event Organizer** system is to provide a unified, user-friendly platform for managing and participating in college events. Traditional methods of event organization—such as notice boards, word-of-mouth communication, and manual registrations—are time-consuming, error-prone, and inefficient. Additionally, students often miss out on events that align with their interests due to lack of personalized communication or centralized access to information.

## **2.1 Traditional System**

In the traditional setup, event communication and coordination are carried out manually or through disjointed tools. Event information is typically shared via notice boards, classroom announcements, WhatsApp groups, or Telegram channels. Registration is often handled through Google Forms, printed sheets, or direct verbal confirmation.

**Limitations**

* Message Overload: In platforms like WhatsApp or Telegram, important event messages are easily lost among other discussions and notifications.
* No Automated Reminders: Students may forget about upcoming events due to the lack of confirmation emails or timely notifications.
* Venue Confusion: Without proper mapping or guidance, students may find it difficult to locate the event venue.

## **2.2 Proposed College Event Organizer System**

The College Event Organizer system provides a digital platform that connects event organizers with students, enabling seamless event creation, promotion, and participation.

* User Registration: Students register on the platform, create their profile, and select interest categories (e.g., AI, Sports, Cultural).
* Event Discovery: The home page displays a curated list of upcoming events filtered based on the user's selected interests.
* Event Registration: Users can register for events directly through the platform, with their participation details stored securely.
* Notifications: Students receive email confirmations upon registration and reminders before the event).
* Google Maps Integration: Each event page shows the venue on an interactive map and offers real-time navigation from the user’s current location.

**2.3 Software Requirements Specification**

Frontend : HTML5, CSS3  
Frontend Framework : React.js  
Backend : Node.js with Express.js  
Database : MongoDB  
Email Service : Nodemailer  
Maps Integration : Google Maps API  
Development Environment : Visual Studio Code

1. **SOFTWARE DESCRIPTION**
   1. **Integrated Development Environment(IDE):**Visual Studio Code (VS Code) is a lightweight, feature-rich code editor widely used for web development. It supports syntax highlighting, IntelliSense (smart code completion), debugging tools, and integrated terminal support. Its extensive library of extensions for JavaScript, React, Node.js, and MongoDB make it ideal for developing full-stack applications. With Git integration and support for custom snippets, it significantly boosts productivity and code quality in real-time application development.
   2. **React.js:**

React.js is a popular JavaScript library used for building user interfaces, especially single-page applications (SPAs). Its component-based architecture promotes modularity, reusability, and easy maintenance. React uses a virtual DOM for efficient UI rendering and state management, enabling seamless updates to the interface without full page reloads. JSX, the syntax extension in React, combines HTML-like structure within JavaScript, making UI development more intuitive and structured.

* 1. **HTML5 and CSS3:**

HTML5 is the core markup language used to define the structure and content of web pages. It introduces semantic elements like <header>, <section>, and <footer>, which help in creating accessible and well-organized layouts. HTML5 works seamlessly with React’s component-based approach to render structured and dynamic content.

Structure of HTML:

<html>

<head>

<title>Page Title</title>

<meta charset="UTF-8">

<link rel="stylesheet" href="styles.css">

<script src="script.js"></script>

</head>

<body></body>

</html>

CSS3 is used for styling and layout design, enhancing the visual appearance of web applications. It supports advanced features like transitions, animations, media queries, and flexbox/grid layouts, enabling responsive design. With CSS3, developers can create engaging, interactive user interfaces that adapt to various screen sizes and devices.

* 1. **Node.js and Express.js:**

Node.js is a server-side JavaScript runtime built on Chrome’s V8 engine. It allows JavaScript to be used for backend development, enabling full-stack development with a single language. Node.js is event-driven and non-blocking, making it ideal for handling concurrent connections and building scalable web applications.  
Express.js is a lightweight web framework for Node.js that simplifies server-side development. It provides a robust set of features for building RESTful APIs, including routing, middleware integration, and HTTP utilities. Express handles request parsing, error handling, and response generation efficiently, forming the backbone of the backend services.

* 1. **MongoDB:**

MongoDB is a NoSQL, document-oriented database designed for high performance and flexibility. It stores data in BSON (binary JSON) format, which aligns well with JavaScript and React-based applications. MongoDB enables dynamic schema design, allowing developers to make changes to the data structure without downtime. It supports powerful querying and indexing capabilities, making it suitable for storing and managing user data, event information, and notifications.

* 1. **Nodemailer:**

Nodemailer is a module for Node.js used to send emails from the server. It supports HTML content, attachments, and various email transport methods such as SMTP. In this application, Nodemailer is used for sending email notifications to users regarding event registrations, confirmations, and updates. This integration helps maintain effective communication between the system and users.

* 1. **Google Maps API:**

The Google Maps API is integrated to display event locations directly on the map. It allows embedding of interactive maps with features such as markers, info windows, and location search. Users can view venue locations and get directions seamlessly within the application. This enhances the event discovery experience and supports better navigation and planning.

* 1. **JWT (JSON Web Tokens):**

JWT is used for secure and stateless user authentication. When a user logs in, a token is generated and sent to the client. This token is then included in subsequent API requests to validate user identity without the need for server-side sessions. JWT improves security, scalability, and performance by reducing server load and ensuring data integrity during transmission.

1. **SYSTEM DESIGN**
   1. **User’s Flow**