

# **COLLEGE EVENT MANAGEMENT WEBSITE REPORT SUBMITTED TO**

**K. R. MANGALAM UNIVERSITY**



**K.R. MANGALAM UNIVERSITY**

**BACHELOR OF TECHNOLOGY  
IN  
COMPUTER SCIENCE AND ENGINEERING  
( CYBERSECURITY )**

**SUBMITTED BY : G161  
Uday Biswas ( 2401410038 )  
Aman Singh ( 2401410023 )  
Abhishek Maurya ( 2401410039 )  
Prem Sagar Mallik ( 2401410018 )**

**SCHOOL OF ENGINEERING & TECHNOLOGY  
K. R. MANGALAM UNIVERSITY  
SOHNA , HARYANA 122103 , INDIA**

# **INTRODUCTION:-**

The purpose of the application “College Event Management Website” is to assist in organizing and managing events in colleges. Students, organizers and the administrators can easily interact with the system as it provides a common place or platform. It consists of various functionalities such as event listing, user login, team member display and a contact form for communication.

The work of designing the website frontend was shared between HTML, CSS and JavaScript. While HTML provided structure to the content, CSS styled it and JavaScript was responsible for making it interactive. It also implemented a responsive design which resized the website according to the screen size, so it could be accessed on mobile devices too. Information of events and team members was also automatically grabbed and populated on the site with the help of JavaScript as well as login popups, event creation form and contact forms.

The website backend is supported by Flask, which is a Python web framework used for building light weight applications. The API endpoints provided by the server for fetching events, submitting contact forms, login user (dummy implementation) and creating new events, also stored files in a static manner. Stimulated databases were used through storing data temporarily in python lists. It CORS (Cross-Origin Resource Sharing) would then enable communication between the backend and second website.

Key features include:

- A login system with form validation.
- A user-friendly form to create and add new events.
- Dynamic display of upcoming events and team members.
- A "Contact Us" form that collects user messages.
- Mobile-friendly layout and smooth user experience.

Currently, the project uses in-memory storage without a real database, and user authentication is basic. Future enhancements could include integrating a database like MySQL, implementing secure user authentication, enabling event registration tracking, and sending email notifications.

In conclusion, this project successfully demonstrates the basic functionalities required for a college event management system. It lays a strong foundation for further development into a fully-featured web application.

## **Objectives:-**

The main goal of the "College Event Management Website" is to make it an efficient, centralized, & user friendly process of managing college related events. Traditionally managing college related events involves handing out posters, spreading information via social media, offline registrations and paperwork, etc. In order to rid society of all these manual processes, this project aims at creating a digital process that makes event management easy & convenient for everyone involved.

Organizers provide student easy access to find events coming up and detail information, register for/make an interest. The website allows event organizers create events themselves using a onetime online form and automatically send the list of events back up, so they don't have to manually edit the website.

Another important target is to strengthen communication between the students and the organizing team: There is a 'Contact Us' function integrated into the Portal which lets users easily contact the organizing team with questions, suggestions or complaints – thus increasing the responsiveness and participation.

In addition, the site should provide a platform to introduce the organizing team thus making it more transparent and building trust among the college community. We fully automate the site (configurable and extendable with accessing full authentication) by including login functionality (this will expand into full authentication in the future), mobile responsiveness and handling dynamic content.

Also the project is building a basis for scalability in future. Potentially additional features (database integration, authentication, Event registration tracking and notification systems) could be added so that the basic prototype can be turned into a specialized event management system.

In other words, the primary goal would be to simplify events creation and promotion, foster communications, foster student engagement and build a modern digital interface that represents the spirit of college life while also providing an efficient and organized way for event management.

- Students can view upcoming events and register.
- Organizers can create and manage events.
- The administration can oversee event operations.
- Team members and contact details are easily accessible.

## **Technology Used:-**

Technology	Purpose
HTML	Structuring Web Pages
CSS	Styling Web Pages
JavaScript	Frontend Interactivity
Flask ( Python )	Backend Server Framework
JSON	Data Exchange Format

## **System Requirements:-**

- Python 3.8+
- Flask (pip install flask flask-cors)
- Web Browser (Chrome, MS Edge)
- Text Editor (VS Code)

# System Components:-

## Frontend

- **HTML (`home.html`) :**
  - Defines the layout of the pages (login form, create event form, homepage, events, team section, contact form).
  - Includes headers, footers, and responsive design elements.
- **CSS (`mm.css`) :**
  - Provides an attractive and responsive design.
  - Styles different components like the login page, event cards, contact form, and team profiles.
- **JavaScript (`mm.js`) :**
  - Manages dynamic content such as:
    - Displaying/hiding login and create-event modals.
    - Populating events and team sections dynamically.
    - Handling form submissions and UI interactions like mobile menus and scroll effects.

## Backend

- **Python Flask Application (`app.py`) :**
  - Serves static files (HTML, CSS, JS).
  - Provides REST API endpoints for:
    - Fetching events and team members.
    - Creating new events.
    - Handling login attempts.
    - Receiving contact form messages.
  - Uses in-memory storage (Python lists) to simulate a database.

## Features Implemented:-

Feature	Description
<b>Login System</b>	Users can attempt to login (dummy implementation without authentication logic).
<b>Event Listing</b>	Users can view upcoming events with images, dates, and descriptions.
<b>Create Event</b>	Organizers can submit a new event through a modal form.
<b>Team Members</b>	Displays information about the organizing team.
<b>Contact Us Form</b>	Allows users to send messages to the organizers.
<b>Mobile Responsive Design</b>	Layout adjusts for mobile devices.
<b>Dynamic Content</b>	Events and team members are dynamically inserted into the page.

## Flow of the Application:-

1. **Homepage** loads with navigation links and an introduction.
2. **User** can:
  - Explore upcoming events.
  - View the organizing team.
  - Contact the organizers via the contact form.
  - Log in or create a new event using modal forms.
3. **Backend APIs** handle event creation, login submission, and contact form data.

## Deployment:-

- The application is designed to be run locally using Flask.
- To run:
  1. Install dependencies (Flask, Flask-CORS).
  2. Start the server by running `python app.py`.
  3. Visit `http://localhost:8080/` to access the website.

## Working Process:-

1. User visits the homepage and sees an introduction about college events.
2. Clicking "Login" brings up a login popup.
3. Organizers can click "Create Event" to add a new event via a form.
4. Events are displayed dynamically on the "Upcoming Events" section.
5. Students can explore, view event details, and register.
6. Visitors can contact the team through the contact form.
7. The backend handles form submissions and event creation logic.

## **Limitation:-**

- No persistent database connection; uses in-memory lists.
- No actual authentication or user management.
- No image uploads (only image URLs).
- No admin panel for event moderation.

## **Future Improvement:-**


- Integrate a real database (like SQLite ).
- Implement secure login/authentication with session management.
- Add user roles (Admin, Organizer, Student).
- Allow file uploads for event images.
- Enable event registration and tracking.
- Add email notifications for event registration and contact forms.





# Screenshots:-

## Contact Us

Have questions about an upcoming event? Want to get involved? Reach out to us and we'll get back to you as soon as possible.

 **Email**  
info@collegeevent.com

 **Phone**  
+91 62992 49967


 **Address**  
College Campus, City

Full Name

Email

Message

Send Message



## Welcome Back

Sign in to manage your college events

Email

Password

☐

Remember me

[Forgot password?](#)

Sign In


Don't have an account? [Sign up](#)

# Create New Event

Fill in the details to create a new event

Event Title

Event Date

Location

Description

Image URL

## **Conclusion:-**

The "College Event Management Website" effectively illustrates an operational prototype for streamlining the management of college events by means of digital technologies. Through the integration of a responsive frontend and a lean Flask backend, the project provides core features such as event creation, event browsing, team introduction, and communication by means of a contact form.

The platform provides a better overall experience for both students and event organizers through an organized and easily accessible solution. Even though it is utilizing in-memory data storage and simple login capabilities today, the project lays a good foundation for further improvements such as secure authentication, database support, real-time tracking of registrations, and email notifications.

This web project not only computerizes conventional event handling procedures but also promotes increased student involvement and activity. With improvement, it has the potential to grow into an all-encompassing event management system applicable for actual college operations. It accomplishes its purpose of uniting and coordinating the college community electronically.