

PROJECT REPORT

**SUBJECT:-
DATABASE MANAGMENT SYSTEM**

**TOPIC:-
COLLEGE EVENT
MANAGEMENT SYSTEM**



PREPARED BY:-

**GARIMA BHARDWAJ
(00517702723)**

**UJJWAL VERMA
(01217702723)**

**KOMAL SHARMA
(02617702723)**

INTRODUCTION:-

In every educational institution, organizing and managing events like seminars, fests, workshops, hackathons, and cultural programs is a regular activity. Traditionally, these tasks are handled manually through notice boards, paper-based registration forms, and verbal communication, which is often inefficient, error-prone, and time-consuming.

The College Event Management System is a web-based solution designed to automate the entire process of managing college events. This system provides a platform where administrators can create events, students can register for them, and organizers can manage participants and event details—all from a centralized interface.

This project aims to eliminate the drawbacks of manual systems by offering a more reliable, accessible, and scalable digital platform that simplifies the event management lifecycle from planning to execution.

OBJECTIVE:-

The main objectives of the College Event Management System are:

- ☒ To automate the creation and scheduling of college events.
- ☒ To provide a user-friendly portal for students to view, search, and register for events.
- ☒ To enable organizers and admins to manage registrations, track participation, and generate event reports.
- ☒ To reduce paperwork, manual errors, and redundant data entry.
- ☒ To ensure secure and centralized event data storage.
- ☒ To improve communication and coordination between students, organizers, and administration.

CODE USED:-

```
CREATE DATABASE college_event_mgmt;  
USE college_event_mgmt;
```

```
CREATE TABLE students (  
    student_id INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(100) NOT NULL,  
    email VARCHAR(100) UNIQUE NOT NULL,  
    branch VARCHAR(50),  
    year INT,  
    phone VARCHAR(15)  
);
```

```
CREATE TABLE events (  
    event_id INT AUTO_INCREMENT PRIMARY KEY,  
    event_name VARCHAR(100) NOT NULL,  
    event_date DATE NOT NULL,  
    location VARCHAR(100),  
    seat_limit INT NOT NULL,  
    available_seats INT NOT NULL  
);
```

```
CREATE TABLE registrations (  
    reg_id INT AUTO_INCREMENT PRIMARY KEY,  
    student_id INT,  
    event_id INT,  
    reg_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    FOREIGN KEY (student_id) REFERENCES students(student_id) ON DELETE CASCADE,  
    FOREIGN KEY (event_id) REFERENCES events(event_id) ON DELETE CASCADE,  
    CONSTRAINT unique_registration UNIQUE(student_id, event_id)  
);
```

```
CREATE TABLE faculty (  
    faculty_id INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(100),  
    department VARCHAR(100),  
    email VARCHAR(100) UNIQUE  
);
```

```
CREATE TABLE event_coordinators (  
    coordinator_id INT AUTO_INCREMENT PRIMARY KEY,  
    event_id INT,  
    faculty_id INT,  
    FOREIGN KEY (event_id) REFERENCES events(event_id),  
    FOREIGN KEY (faculty_id) REFERENCES faculty(faculty_id),  
    CONSTRAINT unique_coordinator UNIQUE(event_id, faculty_id)  
);
```

```

INSERT INTO students (name, email, branch, year, phone) VALUES
('Ananya Roy', 'ananya.roy@college.edu', 'CSE', 2, '9876543210'),
('Rohit Sharma', 'rohit.sharma@college.edu', 'ECE', 3, '9876543211'),
('Sneha Kapoor', 'sneha.kapoor@college.edu', 'IT', 1, '9876543212'),
('Vikram Singh', 'vikram.singh@college.edu', 'ME', 4, '9876543213'),
('Tanya Jain', 'tanya.jain@college.edu', 'CSE', 3, '9876543214');

```

```

INSERT INTO faculty (name, department, email) VALUES
('Dr. Nidhi Sharma', 'CSE', 'nidhi.sharma@college.edu'),
('Dr. Suresh Mehta', 'ECE', 'suresh.mehta@college.edu'),
('Prof. Reena Das', 'IT', 'reena.das@college.edu'),
('Dr. Kunal Bose', 'ME', 'kunal.bose@college.edu');

```

```

INSERT INTO events (event_name, event_date, location, seat_limit, available_seats) VALUES
('AI & ML Workshop', '2025-06-05', 'Auditorium A', 100, 100),
('Robotics Hackathon', '2025-06-08', 'Innovation Lab', 50, 50),
('Tech Quiz', '2025-06-10', 'Room 301', 30, 30),
('Web Dev Bootcamp', '2025-06-12', 'Lab 2', 60, 60);

```

```

INSERT INTO event_coordinators (event_id, faculty_id) VALUES
(1, 1), -- AI & ML Workshop coordinated by Dr. Nidhi Sharma
(2, 2), -- Robotics Hackathon by Dr. Suresh Mehta
(3, 3), -- Tech Quiz by Prof. Reena Das
(4, 1), -- Web Dev Bootcamp also coordinated by Dr. Nidhi Sharma
(4, 3); -- And also by Prof. Reena Das

```

```

DELIMITER $$

```

```

CREATE TRIGGER after_registration_insert
AFTER INSERT ON registrations
FOR EACH ROW
BEGIN
    UPDATE events
    SET available_seats = available_seats - 1
    WHERE event_id = NEW.event_id;
END $$
DELIMITER ;

```

```

DELIMITER $$
CREATE PROCEDURE RegisterStudentForEvent(
    IN p_student_id INT,
    IN p_event_id INT
)
BEGIN
    DECLARE seats_left INT;
    SELECT available_seats INTO seats_left FROM events WHERE event_id = p_event_id;
    IF seats_left > 0 THEN
        INSERT INTO registrations(student_id, event_id)
        VALUES (p_student_id, p_event_id);
    ELSE

```

```
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'No available seats for this event';
END IF;
END $$
```

```
DELIMITER ;
CREATE VIEW student_event_view AS
SELECT
    s.name AS student_name,
    e.event_name,
    e.event_date,
    r.reg_date
FROM
    students s
JOIN
    registrations r ON s.student_id = r.student_id
JOIN
    events e ON r.event_id = e.event_id;
```

```
CALL RegisterStudentForEvent(1, 1); -- Ananya for AI Workshop
CALL RegisterStudentForEvent(2, 2); -- Rohit for Robotics Hackathon
CALL RegisterStudentForEvent(3, 3); -- Sneha for Tech Quiz
CALL RegisterStudentForEvent(4, 1); -- Vikram for AI Workshop
CALL RegisterStudentForEvent(5, 4); -- Tanya for Web Dev Bootcamp
```




```
SELECT event_name, event_date, available_seats
FROM events
WHERE available_seats > 0;
SELECT s.name, s.branch, s.year
FROM students s
JOIN registrations r ON s.student_id = r.student_id
WHERE r.event_id = 1;
```

```
CALL RegisterStudentForEvent(1, 2);
SELECT event_name, event_date, location
FROM events
WHERE event_date > '2025-06-07'
ORDER BY event_date ASC;
```



```
SELECT s.name AS student_name, s.branch, s.year
FROM students s
JOIN registrations r ON s.student_id = r.student_id
JOIN events e ON r.event_id = e.event_id
WHERE e.event_name = 'AI & ML Workshop';
```




```
SELECT e.event_name, COUNT(r.reg_id) AS total_registrations
FROM events e
LEFT JOIN registrations r ON e.event_id = r.event_id
GROUP BY e.event_id;
SELECT * FROM student_event_view;
```




OUTPUT:-

Result Grid			Filter Rows:	Export:		Wrap Cell Cont
	student_name	event_name	event_date	reg_date		
▶	Ananya Roy	AI & ML Workshop	2025-06-05	2025-05-13 19:44:52		
	Vikram Singh	AI & ML Workshop	2025-06-05	2025-05-13 19:45:10		
	Rohit Sharma	Robotics Hackathon	2025-06-08	2025-05-13 19:45:01		
	Ananya Roy	Robotics Hackathon	2025-06-08	2025-05-13 19:45:24		
	Sneha Kapoor	Tech Quiz	2025-06-10	2025-05-13 19:45:05		
	Tanya Jain	Web Dev Bootcamp	2025-06-12	2025-05-13 19:45:14		

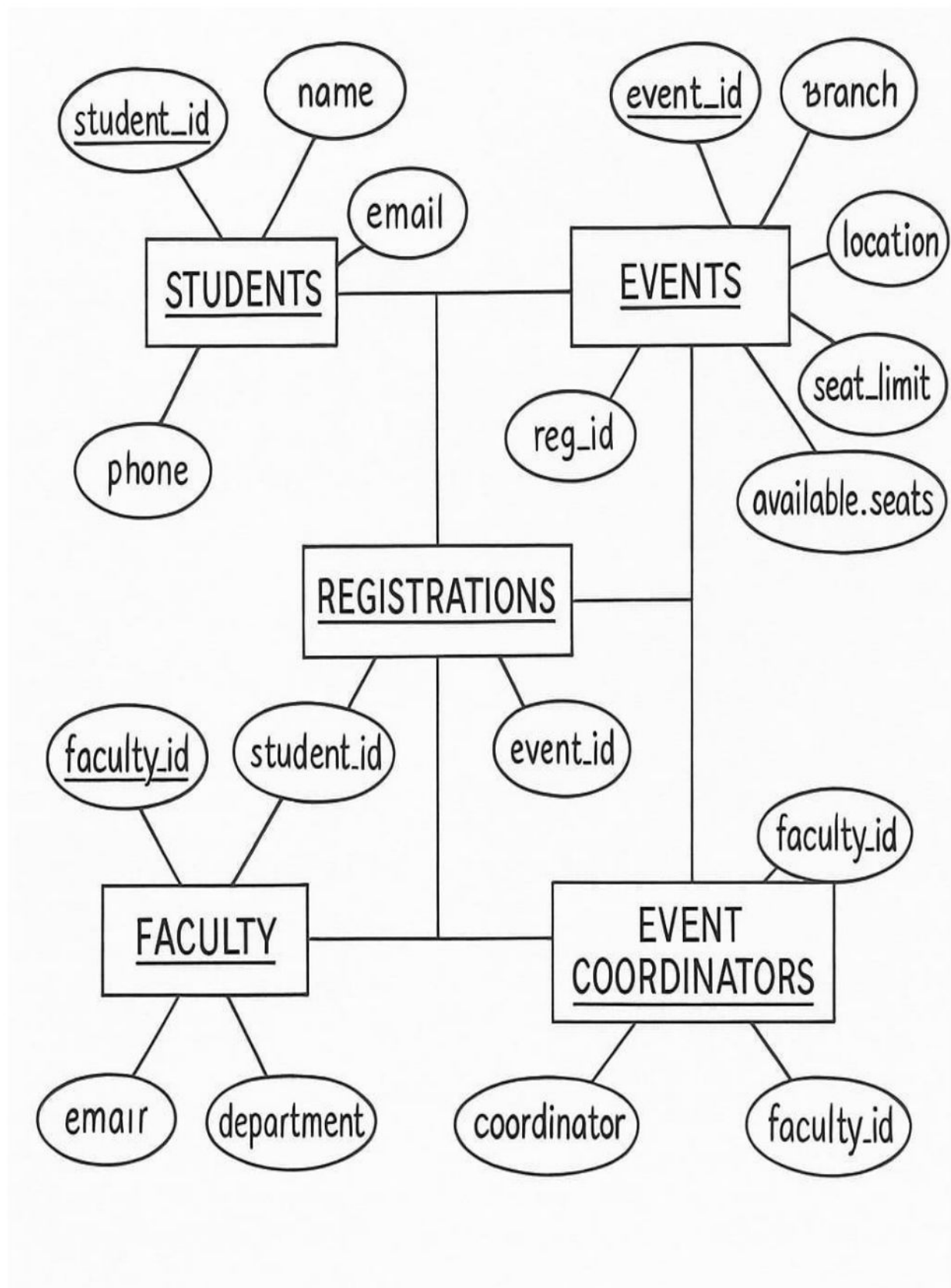
Result Grid			Filter Rows:	Export:	
	event_name	total_registrations			
▶	AI & ML Workshop	2			
	Robotics Hackathon	2			
	Tech Quiz	1			
	Web Dev Bootcamp	1			

Result Grid			Filter Rows:	
	student_name	branch	year	
▶	Ananya Roy	CSE	2	
	Vikram Singh	ME	4	

Result Grid			Filter Rows:	Export:		V
	event_name	event_date	location			
▶	Robotics Hackathon	2025-06-08	Innovation Lab			
	Tech Quiz	2025-06-10	Room 301			
	Web Dev Bootcamp	2025-06-12	Lab 2			

Result Grid			Filter Rows:	Export:	
	event_name	event_date	available_seats		
▶	AI & ML Workshop	2025-06-05	98		
	Robotics Hackathon	2025-06-08	48		
	Tech Quiz	2025-06-10	29		
	Web Dev Bootcamp	2025-06-12	59		

ER DIAGRAM:-



TECHNOLOGIES USED :-

The development of the College Event Management System involved a variety of tools, programming languages, and platforms to create a functional and user-friendly web application.

Category	Technology / Tool
Frontend	HTML5, CSS3, JavaScript
Backend	PHP (Hypertext Preprocessor)
Database	MySQL
Server	Apache (XAMPP/LAMP stack)
IDE / Editor	Visual Studio Code / Sublime Text
Design & Modeling	Draw.io / Lucidchart (for ER diagram)
Browser	Chrome / Firefox (for testing)

SYSTEM DESIGN :-

The system is designed in modular format, divided into three main roles:



1. Admin Module

Create, update, and delete event listings
Approve or reject event proposals by organizers



2. Organizer Module

Propose new events with details like name, date, venue
Edit event information after approval



3. Student Module

Browse upcoming events by category or date
Register online for any event

EXISTING MANUAL SYSTEM – PROBLEMS & LIMITATIONS

In many colleges, event management is still done manually, which includes handwritten notices, registration via paper forms, verbal announcements, and physical coordination. This method has several drawbacks:

- ◆ **Time-consuming:** Event registration, approvals, and communication take a lot of time.
- ◆ **Lack of transparency:** Students often miss important updates due to ineffective communication.
- ◆ **Prone to errors:** Manual entries can result in inaccurate records and lost data.
- ◆ **Limited accessibility:** Information is not accessible remotely or after college hours.
- ◆ **No centralized database:** No single place to store and manage event-related data efficiently.
- ◆ **Difficult reporting:** Generating reports and participation data manually is tedious.

PROPOSED AUTOMATED SYSTEM – BENEFITS

The **College Event Management System** addresses the above limitations through a web-based automated solution. Key benefits include:

- ✓ **Centralized System:** One platform for students, organizers, and administrators.
 - ✓ **Real-Time Access:** Students can register and access event details 24/7.
 - ✓ **Data Accuracy:** Information is stored digitally, reducing human error.
 - ✓ **Efficient Communication:** Event updates and status are instantly available.
 - ✓ **Secure and Organized:** All records are stored in a secure, structured database.
-



CONCLUSION:-

The College Event Management System project successfully demonstrates how digital automation can improve the planning, coordination, and execution of college events. By replacing the traditional manual methods with a centralized online platform, the system ensures better communication, efficient registration, secure data handling, and smoother overall operations.

This web-based application not only saves time and reduces paperwork but also enhances the experience for students, organizers, and administrators alike. The modular design ensures that each user role—Admin, Organizer, and Student—can interact with the system efficiently based on their needs.

This project has also provided a practical understanding of how to apply web development tools and database management in real-world scenarios. It integrates both front-end and back-end technologies to create a functional, user-friendly system.



FUTURE SCOPE:-

- Integration with college email and SMS systems for notifications
- Mobile app development for Android/iOS platforms
- QR code-based event check-in for attendance
- Feedback collection and analysis system post-event
- Admin dashboard analytics with charts and participation trends