RAJALAKSHMI ENGINEERING COLLEGE

DATABASE MANAGEMENT SYSTEM-CS23332

EVENT MANAGEMENT SYSTEM

MINI PROJECT REPORT

Submitted By,

Devyesh C 241801049
Dhinesh J 241801054
Ambrish Kumar A 241801017

Department: Artificial Intelligence And Data Science
Semester: 3rd Semester
Year: 2nd Year

OCTOBER 2025

BONAFIDE CERTIFICATE

Certified that this project **"EVENT MANAGEMENT SYSTEM"** is the bonafide work of **"DEVYESH C, DHINESH J, AMBRISH KUMAR A"** who carried out the project work under my supervision.

SIGNATURE

SAVITHRI R
ASSISTANT PROFESSOR,
DEPARTMENT OF
ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

This mini project report is submitted for the viva voce examination to be held on _____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

INDEX

1. INTRODUCTION:

The Event Management System is a web-based application developed to simplify the process of registering and managing college events. In most educational institutions, event registration is still handled manually through physical forms or spreadsheets, which often leads to issues such as inefficient data collection, lack of transparency, difficulty in maintaining participant details, and delays in coordination. This system aims to overcome these limitations by providing a centralized online platform where students can register for events and administrators can manage them easily from a single interface.

This project primarily focuses on designing a clean and user-friendly interface for both students and administrators, ensuring ease of access and smooth navigation. The application allows students to sign up, log in, and register for various events conducted within the college. On the administrative side, the system enables authorized users to view participant details, update event-related information, and monitor registrations in an organized manner. The overall approach is to simplify event handling and enhance communication between participants and organizers.

Although the current version of the system uses only front-end technologies to demonstrate the working flow of the application, it has been structured in a way that can be easily extended with a real back-end and database in the future. This provides a scalable foundation for future development, enabling the project to evolve from a demonstration model into a complete functional portal.

2. ABSTRACT:

The Event Management System is a web-based application designed to digitalize and simplify the process of managing college-level events. The system provides a structured interface where students can register for events online, eliminating the need for manual paperwork. It also supports administrators by allowing them to view and manage participant information efficiently. The primary focus of this project is on the front-end development to demonstrate the workflow of a complete event management platform, including user registration, login authentication (UI-based), dashboard access, and event-related interactions.

This project highlights the importance of user-friendly design, responsive layout, and modular structure, which collectively improve usability and future scalability. Although it currently functions without a real-time database or server-side implementation, its architecture is prepared for easy integration with back-end technologies in later stages. The system serves as a prototype model that can be further expanded into a fully functional application for real-world deployment within educational institutions.

3. TOOLS USED:

3.1 HTML, CSS:

For Developing the Front-End, HTML and CSS is used.

3.2 Java:

The Back-End is Developed using the Java Programming Language.

3.3 Visual Studio Code:

The whole Development is carried out with the help of VS Code. It is used for writing and organising the project files.

3.4 Data Base:

MySQL Database is used to store the Tables of the system.

3.5 Connectivity:

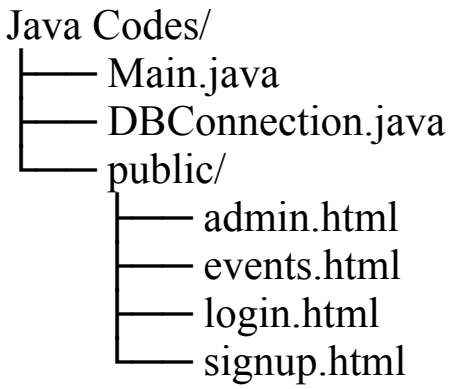
The JDBC Driver is used the link the Database with the Back-End.

3.6 Browser:

Safari browser is used for testing and running the web-pages.

4. STRUCTURE OF FILES IN THE PROJECT FOLDER:

4.1 Flow Chart:



4.2 Contents of each File / Folder:

Main.java:

Acts as the entry point of the application and sets up the server, routes, and handlers.

DBConnection.java:

Contains utility methods to establish a connection with the MySQL database.

public/ folder:

Contains all frontend pages and assets that are served to users.

admin.html:

Contains all frontend pages and assets that are served to users.

events.html:

Page intended to display available events to users for registration.

login.html:

Interface for users to log in to the system.

signup.html:

Interface for users to register for an account.

5. FRONT-END:

The front-end of the Event Management System is built using **HTML**,and **CSS** to provide a clean and user-friendly interface. Since this project is intended for a college environment, the design focuses on simplicity, quick access, and ease of navigation rather than using large UI frameworks.

The application contains separate interfaces for users and administrators. The **login** and **signup** pages allow users to register and sign in to the system, while the **admin dashboard** provides additional features for event management. A **collapsible sidebar** layout is used in the admin interface to give a modern, application-style feel and to efficiently utilize screen space. Through this sidebar, the admin can navigate between viewing the dashboard, events, user registrations, and profile/logout options.

The user interface is styled using CSS to maintain proper spacing, typography, and consistency in layout. JavaScript is used in the sidebar to control the toggling animation and improve interactivity. The front-end is lightweight and responsive, which ensures that the pages load quickly and can be accessed comfortably on laptops, desktops, or college lab machines without depending on advanced hardware or external frameworks.

5.1 admin.html:

It contains a collapsible sidebar with navigation options such as Home, Profile, Events, and Logout. The main content area displays admin-level functions like viewing and managing events. This layout gives a web-application feel rather than a static page.

Purpose:

- **Allows admin to manage events and system data**
- **Provides a centralized control panel**
- **Modern UI with collapsible sidebar**

5.2 events.html:

The Events page of the Event Management System is a front-end web interface where all upcoming events are displayed to the registered users in a clean and responsive card layout. It allows users to view basic details of each event such as name, date, venue, registration fee, and closing date.

Purpose:

- **To enable students to easily browse available events and register for them with a single click.**
- **Acts as a bridge between the front-end and the backend server.**
- **The slide-out side panel provides navigation and logout functionality, improving user experience.**

5.3 login.html:

The Login page is the authentication interface of the Event Management System where users enter their registered email and password to access their account. It validates the user credentials and ensures that only authorized users can access the internal features of the system.

Purpose:

- **Secure access control by verifying the identity of the user.**
- **First point of interaction after signup and prevents unauthorized usage of the portal.**

5.4 signup.html:

The Signup page is the user registration interface of the Event Management System where new users provide their personal and academic details such as name, roll number, email, phone number, department, and year to create an account in the system.

Purpose:

- **To collect and store the necessary user information.**
- **It enables new users to register themselves.**

6. BACK-END:

The backend of the Event Management System is built using Java and MySQL, operating as the core business logic layer of the application. It is responsible for processing requests coming from the front-end, interacting with the database, and returning the relevant responses back to the user. The backend manages key functionalities such as user registration, login authentication, secure data retrieval, and event creation by the administrator. It also performs validation, routing, and ensures proper communication between the user interface and the database. By separating the application logic from the presentation layer, the system becomes more scalable, maintainable, and secure.

6.1 DBConnection.java:

The DBConnection.java file acts as the central utility for connecting the application to the MySQL database. Instead of creating database connections repeatedly across different parts of the code, this class provides a single reusable method getConnection() that returns an active database connection whenever required. It loads the MySQL JDBC driver dynamically and ensures that the server can securely establish a link with the database using the configured URL, username, and password. This decoupling of connection logic promotes cleaner code and easier configuration management, making changes in database credentials or connection parameters much simpler without modifying every handler file.

6.2 Main.java:

The Main.java file serves as the entry point of the backend and functions as the embedded web server controller for the application. It initializes an HttpServer instance, assigns a listening port, and registers URL endpoints with their respective handler classes. Each handler corresponds to a specific operation such as signup, login, event creation, or fetching event data. Additionally, the StaticHandler included in this file enables the server to serve HTML, CSS, and JavaScript files directly from the public folder. In essence, this file acts as a routing and configuration hub that determines how each incoming request is processed and which component should handle it. Without this file, none of the backend services would be reachable by the client-side interface.

7. DATABASE:

The database used for this project is **eventdb**, which is designed to efficiently store and manage all the information related to users, events, and event registrations. The database ensures secure storage of credentials, structured representation of event details, and a proper mapping of which user has registered for which event. It follows a relational model using MySQL, which provides reliability, scalability, and easy integration with the Java backend application.
The database consists of **three main tables**: users, events, and registrations. Each table is created with a clear purpose and a primary key to uniquely identify each record. Proper relationships are also established through foreign keys to maintain referential integrity.

7.1 Tables:

1. **users**
2. **events**
3. **registrations**

7.2 Table Description:

user Table:

Stores the details of all registered users and administrators of the system, including personal and login information.

events Table:

Contains all event-related information such as event name, date, venue, registration fee, deadline, and participant limit.

registrations Table:

Maintains details of which user has registered for which event. This acts as a bridge between the users and events tables.

7.3 Table Structure Summary:

users

- Fields: name, roll_no, email, phone_number, department, year, password, role
- Primary Key: email
- Purpose: Stores user and admin information for authentication and access control.

events

- Fields: id, event_name, event_date, venue, reg_fee, reg_close_date, max_participants
- Primary Key: id
- Purpose: Contains complete information about each event created by the admin.

registrations

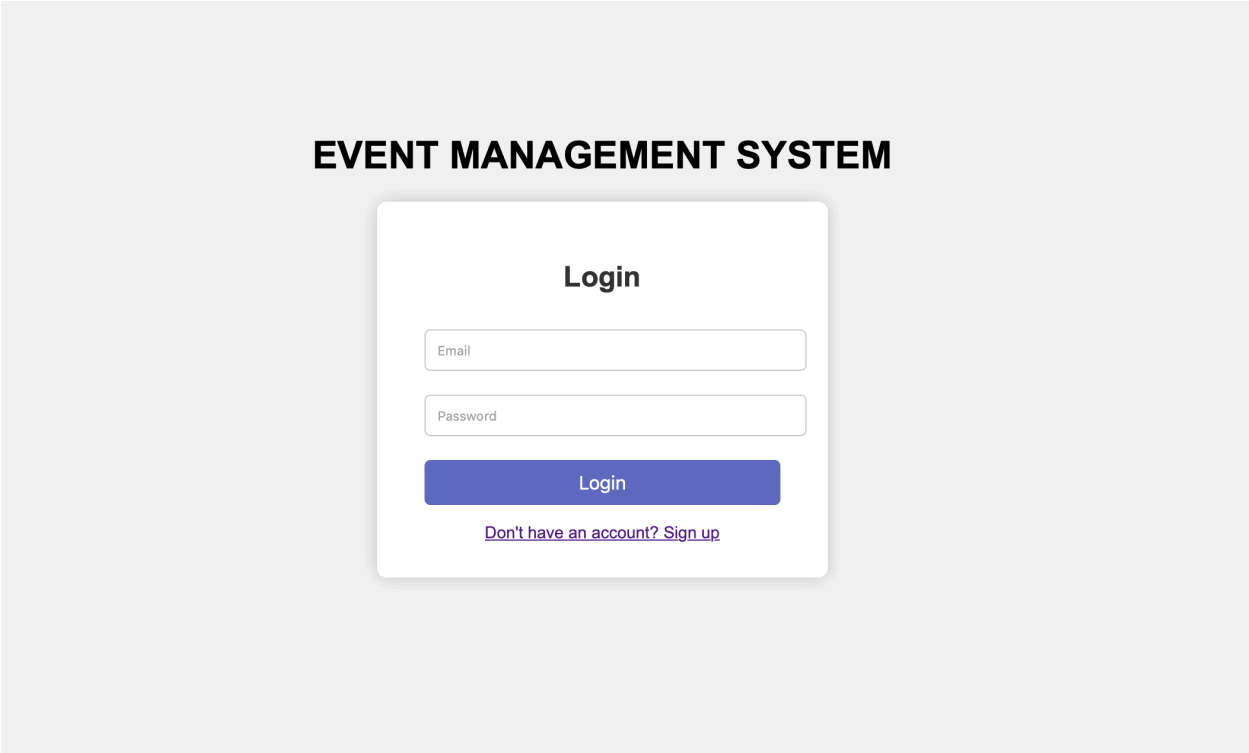
- Fields: id, event_id, event_name, event_date, user_name
- Primary Key: id
- Foreign Key: event_id (references events.id)
- Purpose: Keeps track of which user has registered for which event.

7.4 Relationship:

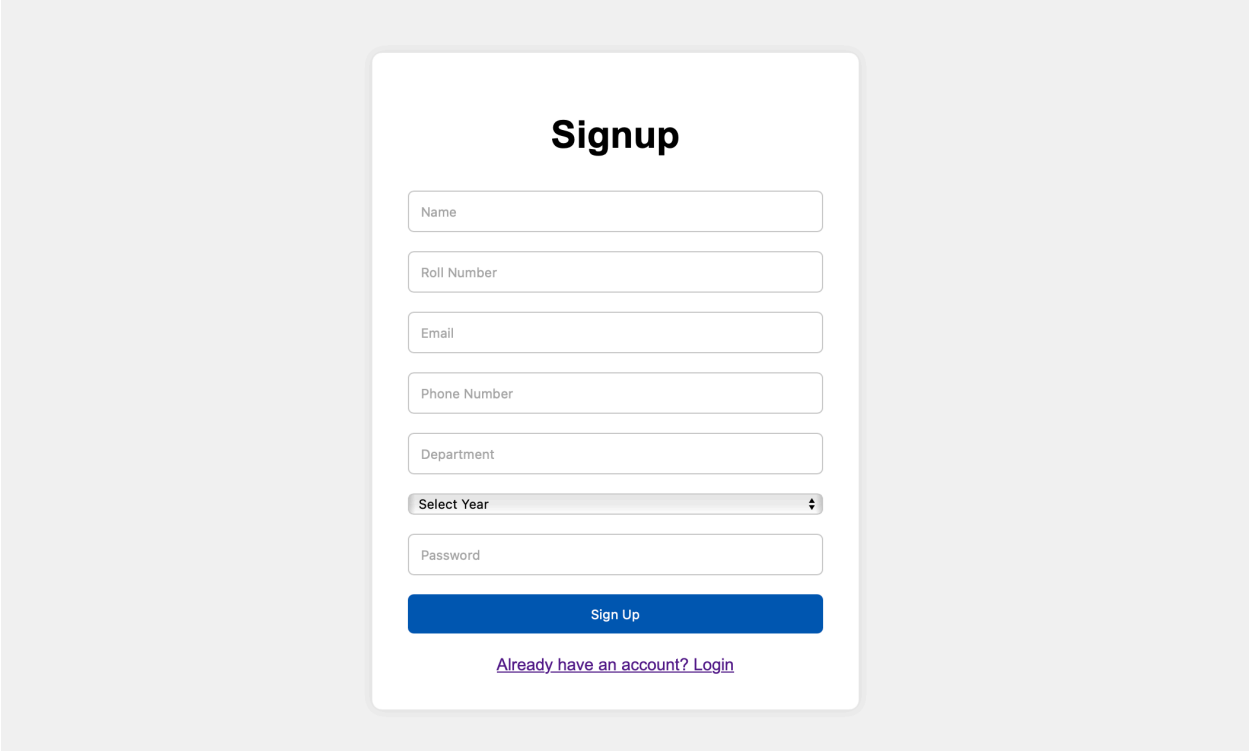
The database follows a one-to-many relationship between the events table and the registrations table. A single eventcan have multiple registrations, but each registration refers to only one event, achieved using the foreign key event_id.

8. SCREENSHOTS OF WEB PAGES:

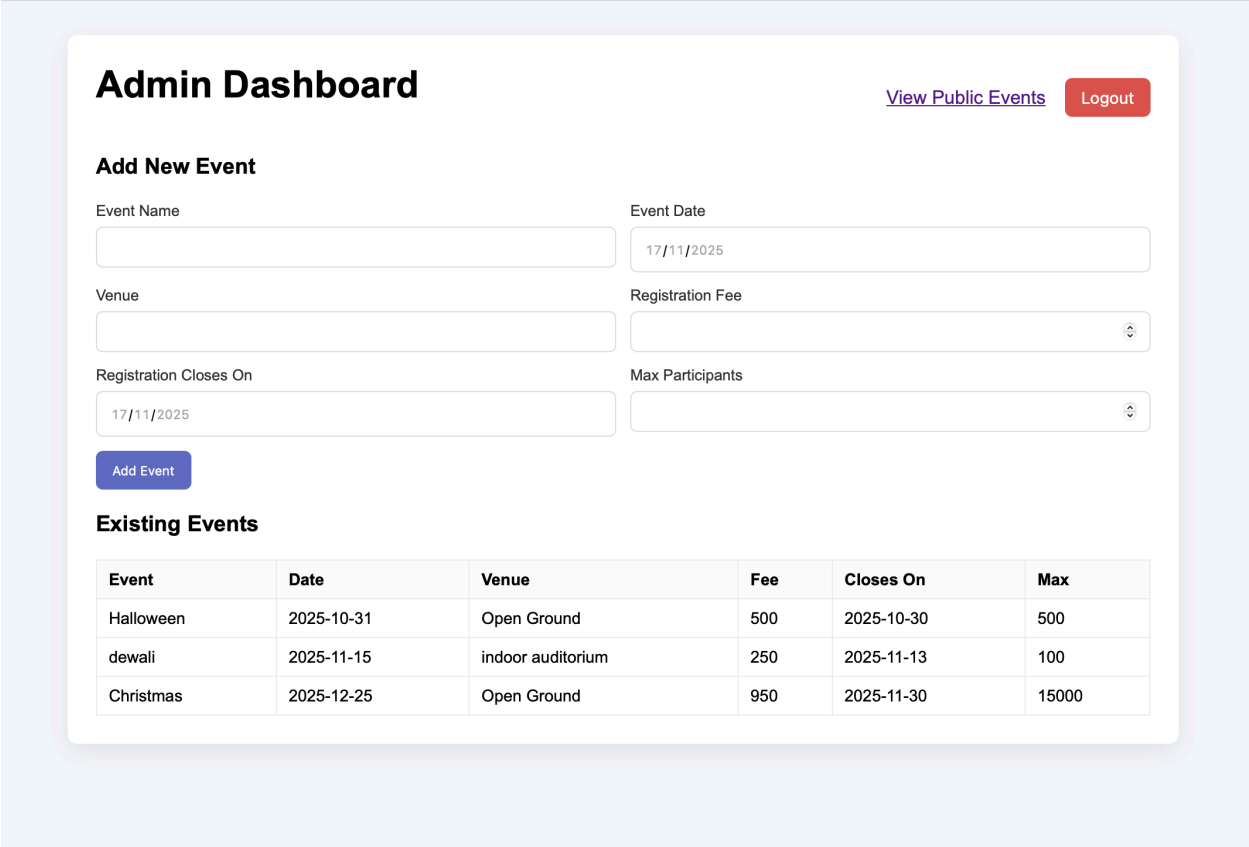
8.1 Login Page:



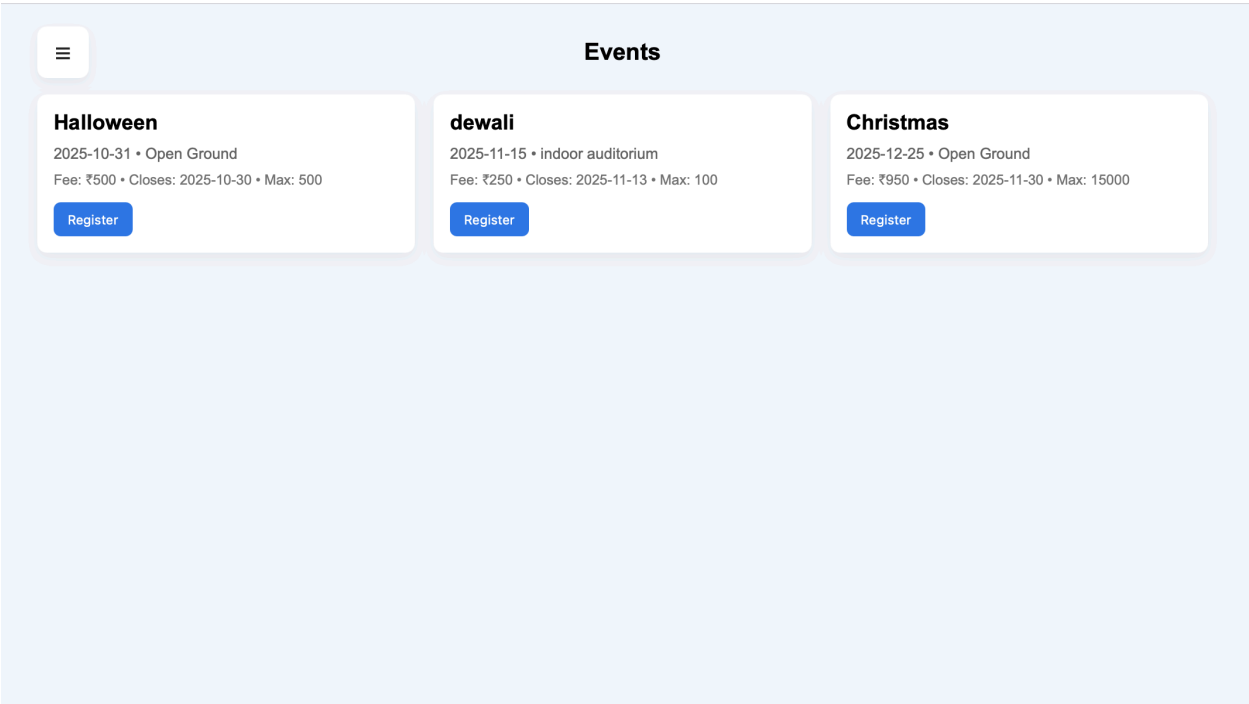
8.2 Sign-up Page:



8.3 Admin Home Page:



8.4 User Home Page:



8.5 DataBase Relations:

8.5.1 Events:

```
mysql> select * from events;
+----+------------+------------+------------------+---------+---------------+------------------+
| id | event_name | event_date | venue            | reg_fee | reg_close_date | max_participants |
+----+------------+------------+------------------+---------+---------------+------------------+
|  1 | Christmas  | 2025-12-25 | Open Ground      |  950.00 | 2025-11-30     |            15000 |
|  2 | Halloween  | 2025-10-31 | Open Ground      |  500.00 | 2025-10-30     |              500 |
|  3 | dewali     | 2025-11-15 | indoor auditorium |  250.00 | 2025-11-13     |              100 |
+----+------------+------------+------------------+---------+---------------+------------------+
3 rows in set (0.001 sec)
```

8.5.2 Registrations:

```
mysql> select * from registrations;
+----+----------+------------+------------+-----------+
| id | event_id | event_name | event_date | user_name |
+----+----------+------------+------------+-----------+
|  1 |        1 | Christmas  | 2025-12-25 | Antony    |
|  2 |        2 | Halloween  | 2025-10-31 | Dhyanesh  |
|  3 |        3 | dewali     | 2025-11-15 | Dhyanesh  |
+----+----------+------------+------------+-----------+
3 rows in set (0.019 sec)
```

8.5.3 Users:

```
mysql> select * from registrations;
+----+----------+------------+------------+-----------+
| id | event_id | event_name | event_date | user_name |
+----+----------+------------+------------+-----------+
|  1 |        1 | Christmas  | 2025-12-25 | Antony    |
|  2 |        2 | Halloween  | 2025-10-31 | Dhyanesh  |
|  3 |        3 | dewali     | 2025-11-15 | Dhyanesh  |
+----+----------+------------+------------+-----------+
3 rows in set (0.001 sec)
```

9. FUTURE ENHANCEMENTS:

Although the current version of the **Event Management System** successfully handles essential operations such as user registration, login authentication, event creation, and participant registration, there are several opportunities to enhance its functionality and user experience in future versions.

1. Role-Based Access Control (RBAC):
Implementing a more advanced user role system where admins, event organizers, and participants have different access levels. This would improve security and streamline management.

2. **Email and SMS Notifications:**
Adding automated email or SMS notifications to inform users about event confirmations, reminders, and updates would enhance communication and engagement.

3. **Event Analytics Dashboard:**
A graphical dashboard can be developed to display statistics such as total registrations, most popular events, and participant demographics. This would provide administrators with better insights.

4. Online Payment Integration:
Integration with online payment gateways (like Razorpay or PayPal) would allow participants to pay registration fees securely during the event sign-up process.

5. **User Profile Management:**
Adding profile management features where users can update their information, view event history, and download participation certificates.

6. Feedback and Rating System:
Introducing a feedback system after each event would help organizers evaluate success and improve future events based on participant responses.

7. Cloud Deployment:
Hosting the system on cloud platforms (AWS, Azure, or Google Cloud) to allow remote access and scalability for larger institutions or organizations.

8. Mobile Application Support:
Developing a mobile app version of the system for Android and iOS platforms to make event registration and management more accessible and user-friendly.

10. SUMMARY:

The **Event Management System** project was developed with the goal of simplifying and automating the process of managing events within an institution or organization. It successfully integrates multiple functionalities such as user registration, login authentication, event creation by administrators, and event participation by students — all within a secure and user-friendly interface.

By combining **HTML, CSS, and Java** for frontend and backend development, and using **MySQL** as the database, the system ensures smooth interaction between users and the server. The modular design of the application enhances maintainability and scalability, while the database structure guarantees data integrity and consistency.

This project demonstrates how modern web technologies can be used to reduce manual work, improve data management, and make event organization more efficient. It not only benefits administrators by centralizing event control but also provides convenience to participants by allowing them to view and register for events easily.

Overall, the Event Management System lays a strong foundation for a digital event management solution. With future enhancements like analytics, notifications, and mobile accessibility, it has the potential to evolve into a complete and professional-grade event management platform.