COLLEGE EVENT MANAGEMENT SYSTEM

A MINI PROJECT REPORT

Submitted by

SARANRAJ A 220701510

KAVIARASU M 220701517

In partial Fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE

(AUTONOMOUS) THANDALAM

CHENNAI – 602105

2023-2024

BONAFIDE CERTIFICATE

Certified that this project report " **COLLEGE EVENT MANAGEMENT SYSTEM** " is the bonafide work of "**SARANRAJ A (220701510), KAVIARASU M (220701517)**"

Who carried out the project work under my supervision.

Submitted for the Practical Examination held on _____

SIGNATURE

Dr. R. Sabitha,
Professor and II Year Academic Head,
Computer Science and Engineering,
Rajalakshmi Engineering College
(Autonomous),
Thandalam, Chennai – 602105

SIGNATURE

Mrs. Vinothini,,
Assistant Professor ,
Computer Science and Engineering,
Rajalakshmi Engineering College
(Autonomous),
Thandalam, Chennai – 602105

INTERNAL EXAMINER

EXTERNAL EXAMINER

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1. INTRODUCTION

The College Cultural Event Management System is a comprehensive web-based solution aimed at enhancing the efficiency and effectiveness of organizing, managing, and participating in cultural events within a college environment. Cultural events are an integral part of college life, providing students with opportunities to showcase their talents, engage in creative pursuits, and foster a sense of community. However, managing these events involves numerous logistical challenges, including coordinating event details, handling participant registrations, and maintaining accurate records.

Traditional methods of event management, which often rely on manual processes and paper-based systems, can be time-consuming and prone to errors. To address these challenges, the College Cultural Event Management System offers a digital platform that streamlines the entire event management process. This system allows administrators to easily create, update, and delete events, while providing students with a user-friendly interface to view event details and register online.

The system leverages modern web technologies, including PHP for server-side scripting, MySQL for database management, and JavaScript and CSS for creating a dynamic and responsive user interface. Key features include the ability to manage event details, track participant registrations, and export data for reporting purposes. By automating and digitizing these processes, the College Cultural Event Management System reduces administrative burdens and enhances the overall experience for both event organizers and participants.

This project not only aims to improve the operational efficiency of cultural event management but also to provide a scalable and secure solution that can be adapted to the unique needs of any educational institution. Through this system, colleges can ensure that their cultural events are organized more effectively, allowing for greater student participation and engagement.

1.2. OBJECTIVES

The primary objective of the College Cultural Event Management System is to streamline the process of organizing and managing cultural events. This includes:

1. **Efficient Event Creation and Management**: Enabling administrators to easily create, update, and delete events, along with detailed information such as event name, date, time, venue, description, and participants.
2. **User-Friendly Registration**: Providing students with an intuitive interface to view event details and register online, thereby enhancing student participation.
3. **Accurate Record-Keeping**: Maintaining a centralized database of all events and registrations to ensure data accuracy and ease of access.
4. **Data Export and Reporting**: Allowing administrators to export event and participant data for reporting and analysis purposes.

Features

1. **Admin Panel**: The system includes a secure admin panel where administrators can manage event details. Features include adding new events, updating existing events, and deleting events that are no longer needed.
2. **Student Interface**: A dedicated interface for students to view upcoming events, register for their preferred events, and receive confirmation of their registrations.
3. **Database Management**: Utilizing MySQL to store all event-related data securely. This includes event details, participant information, and registration records.
4. **Export Functionality**: The ability to export participant lists and other event data to Excel, facilitating easy reporting and record-keeping.
5. **Responsive Design**: Ensuring the platform is accessible on various devices, including desktops, tablets, and smartphones, to cater to the needs of all users.
6. **Interactive Elements**: Incorporating JavaScript and CSS to enhance the user experience with interactive features such as hover effects, popups for confirmation messages, and real-time data updates.

Technologies Used

1. **PHP**: For server-side scripting and handling business logic.
2. **MySQL**: As the relational database management system to store event and participant data.
3. **HTML/CSS**: To create a structured and visually appealing user interface.
4. **JavaScript**: For adding interactivity and improving user experience.
5. **Libraries and Tools**: Utilizing external libraries like PhpSpreadsheet for exporting data to Excel.

Workflow

1. **Event Creation**: Administrators log into the admin panel and fill out a form to create a new event, specifying details such as event name, date, time, venue, and description.
2. **Registration**: Students access the event listing on the platform, select the events they are interested in, and complete the registration process online.
3. **Data Management**: All event and participant data is stored in the MySQL database, ensuring centralized and secure data management.
4. **Exporting Data**: Administrators can export participant lists and event data to Excel for further analysis and record-keeping.
5. **Feedback and Updates**: Post-event, administrators can update records, gather feedback, and plan future events more effectively based on the data collected.

1.3. MODULES

1.User Authentication Module

- **Purpose**: To ensure secure access to the system.
- **Features**:
    - Login and logout functionality for administrators.
    - Password encryption and validation.
    - Session management to prevent unauthorized access.

The purpose of implementing secure access to the college event management system is to ensure that only authorized users can perform administrative tasks, thereby protecting sensitive data and maintaining the integrity of the system. Key features to achieve this include login and logout functionality for administrators, which provides a controlled entry and exit point to the system, ensuring that only verified users can access administrative functions. Password encryption and validation are crucial for safeguarding user credentials, as encrypted passwords make it significantly harder for attackers to decipher and misuse them. Additionally, robust session management mechanisms are employed to prevent unauthorized access; by tracking active sessions and automatically logging out users after a period of inactivity or when they manually log out, the system mitigates the risk of unauthorized access. Together, these features form a comprehensive security framework that upholds the confidentiality, integrity, and availability of the system.

2. Event Management Module

- **Purpose**: To facilitate the creation, updating, and deletion of events.
- **Features**:
    - Form for adding new events with fields for event name, date, time, venue, description, and participants.
    - Edit functionality to update existing event details.
    - Delete functionality to remove events from the system.
    - Validation to ensure all necessary event details are provided.

The purpose of facilitating the creation, updating, and deletion of events within the college event management system is to enable efficient and effective event management. Key features supporting this purpose include a form for adding new events, which includes fields for essential details such as event name, date, time, venue, description, and participants, ensuring comprehensive event information is captured. The edit functionality allows administrators to update existing event details, providing flexibility to accommodate changes and corrections as needed. The delete functionality enables the removal of events from the system, helping to maintain an accurate and current event schedule. Additionally, validation mechanisms are implemented to ensure that all necessary event details are provided, preventing incomplete or incorrect data entries. This comprehensive approach to event management streamlines the process for administrators, ensuring that events are accurately represented and efficiently handled within the system.

3. Student Registration Module

- **Purpose**: To allow students to register for events.
- **Features**:
    - List view of all upcoming events.
    - Registration form for students to sign up for events.
    - Confirmation messages to acknowledge successful registration.
    - Cancellation option for students to withdraw from events.

The Student Registration Module of the college event management system is designed to streamline the process for students to register for events. The primary purpose of this module is to facilitate student engagement by providing a straightforward and efficient method for event registration. The module features a list view of all upcoming events, allowing students to easily browse and select events of interest based on detailed information such as event name, date, time, and venue. Once a student chooses an event, they can use the registration form to sign up by providing necessary personal details. Upon successful registration, students receive confirmation messages that acknowledge their enrollment and provide any additional event information or instructions. Additionally, the module includes a cancellation option, enabling students to withdraw from events if their plans change, thus keeping the event organizers informed about the actual number of participants. This comprehensive approach ensures a user-

friendly experience for students while allowing organizers to efficiently manage and update event participation records.

4. Database Management Module

- **Purpose**: To handle all data storage and retrieval operations.
- **Features**:
    - Integration with MySQL to store event and participant details.
    - Queries for inserting, updating, deleting, and fetching data.
    - Backup and restore functionality to protect data integrity.

The Database Management Module of the college event management system is crucial for handling all data storage and retrieval operations, ensuring that the system functions efficiently and reliably. The primary purpose of this module is to manage the backend database operations seamlessly. It integrates with MySQL to store essential details about events and participants, providing a robust and scalable solution for data management. The module supports various database operations, including queries for inserting new records, updating existing data, deleting records, and fetching data as needed. These operations are fundamental for maintaining accurate and up-to-date information within the system. Additionally, the module incorporates backup and restore functionality, which is vital for protecting data integrity and ensuring that data can be recovered in case of accidental loss or corruption. By facilitating secure and efficient data management, the Database Management Module underpins the overall functionality and reliability of the college event management system.

5. Data Export Module

- **Purpose**: To provide an option to export event and participant data for reporting purposes.
- **Features**:
    - Export data to Excel format using PhpSpreadsheet or similar libraries.
    - Option to download participant lists and event details.
    - Filters to export data for specific events or date ranges.

The purpose of the data export feature in the college event management system is to enable administrators to generate and utilize reports by exporting event and participant data. This feature is essential for facilitating comprehensive reporting and data analysis. It allows for the export of data to Excel format using PhpSpreadsheet or similar libraries, making it easy to handle and share the information in a widely-used format. Administrators have the option to download participant lists and detailed information about events, providing valuable insights for event planning and evaluation. Additionally, the feature includes filters that enable the export of data for specific events or within certain date ranges, allowing for targeted and relevant reporting. This flexibility ensures that administrators can efficiently generate the necessary reports to support decision-making and event management processes.

6. User Interface Module

- **Purpose**: To provide a responsive and interactive interface for both administrators and students.
- **Features**:
    - User-friendly forms and tables for data entry and display.
    - CSS and JavaScript for styling and interactivity.
    - Responsive design to ensure compatibility with various devices.

The purpose of providing a responsive and interactive interface in the college event management system is to enhance the user experience for both administrators and students, ensuring that the system is accessible and easy to use across various devices. This module features user-friendly forms and tables for efficient data entry and display, making it simple for administrators to manage events and for students to register for them. By incorporating CSS for styling, the interface is visually appealing and consistent, while JavaScript adds interactivity, enabling dynamic elements such as real-time form validation and responsive feedback. The responsive design ensures that the interface adapts seamlessly to different screen sizes and devices, from desktops to smartphones, providing a smooth and intuitive experience regardless of how the system is accessed. This combination of user-friendly design, interactivity, and responsiveness makes the event management system efficient and accessible, significantly improving the usability and satisfaction of its users.

7. Reporting Module

- **Purpose**: To generate reports on event participation and other metrics.
- **Features**:
    - Generate summary reports of all events.
    - Detailed reports on individual events, including participant statistics.
    - Graphical representation of data for better visualization.

8. Admin Dashboard Module

- **Purpose**: To provide a centralized control panel for administrators.
- **Features**:
    - Overview of all events and registrations.
    - Quick access to event management and data export options.
    - Activity logs to track changes and updates.

The purpose of the centralized control panel in the college event management system is to provide administrators with a comprehensive and efficient tool for managing all aspects of events and registrations. This control panel offers an overview of all events and participant registrations, giving administrators a clear and organized snapshot of current and upcoming activities. With quick access to essential functions such as event management and data export options, administrators can efficiently create, update, and delete events, as well as export participant and event data for reporting purposes. Additionally, the control panel includes activity logs that track all changes and updates made within the system, ensuring transparency and accountability. This centralized interface streamlines administrative tasks, enhances oversight, and improves the overall efficiency of managing college events.

CHAPTER 2

SURVEY OF TECHNOLOGIES

2.1 SOFTWARE DESCRIPTION

The College Cultural Event Management System is a web-based application designed to streamline the organization and management of cultural events within a college or university. This system provides a comprehensive solution for event creation, student registration, data management, and reporting, ensuring a seamless experience for both administrators and participants.

2.1.1 Functional Requirements

Functional requirements are detailed statements that specify the functions and behaviors a system must have to meet the needs of its users and stakeholders. These requirements describe the tasks the system should perform, the operations it must support, and the ways users will interact with it. Here is a basic explanation of functional requirements using an example of a cultural event management system:

- **User Authentication:**
  - Secure login and logout for administrators.
  - Password encryption and validation.
- **Event Management:**
  - Add, update, and delete events.
  - Validate event details to ensure completeness.
- **Student Registration:**
  - Display list of events for students to view.
  - Register students for selected events.
  - Provide confirmation and cancellation options.
- **Data Export:**
  - Export event and participant data to Excel.

- Filter data based on specific criteria.

- **Reporting:**
    - Generate summary and detailed reports.

2.1.2 Non-Functional Requirements

Non-functional requirements specify the quality attributes, performance, and constraints of a system. They define how the system performs and operates rather than what it should do. Following subtopic are NFR of this project:

- **Performance:**
    - Efficient handling of multiple user requests.
    - Quick data retrieval and processing.
- **Scalability:**
    - Ability to handle an increasing number of events and participants.
- **Security:**
    - Protection against SQL injection and cross-site scripting (XSS).
    - Secure data transmission using HTTPS.
- **Usability:**
    - User-friendly interface with intuitive navigation.
    - Responsive design for compatibility with various devices.
- **Maintainability:**
    - Well-documented code for easy maintenance and updates.
    - Modular design to facilitate future enhancements.

2.1.3 Database Design

- **Tables:**
    - **Events:** Stores event details (event ID, name, date, time, venue, description, participants).
    - **Users:** Stores user credentials and roles.
    - **Registrations:** Tracks student registrations for events.
- **Relationships:**
    - One-to-many relationship between events and registrations.
    - Users can have roles (e.g., administrator).

2.1.4 User Interface Design

- **Admin Dashboard:**
    - Overview of events and registrations.
    - Links to add, update, and delete events.
    - Export data and generate reports.
- **Event List:**
    - Table displaying all events with options to edit or delete.
    - Registration form for students.
- **Registration Form:**
    - Form fields for entering student details and selecting events.
    - Confirmation message on successful registration.

2.1.5 Implementation Details

- **Authentication:**
    - Sessions for managing logged-in users.
    - Password hashing using PHP's password_hash function.

- **Data Export:**
    - PhpSpreadsheet library for creating and downloading Excel files.
    - Button to trigger export functionality.
- **Reporting:**
    - SQL queries to generate reports.
    - Integration with chart libraries for data visualization.

2.1.6 Testing and Deployment

- **Testing:**
    - Unit testing for individual components.
    - Integration testing for overall system functionality.
    - User acceptance testing (UAT) with feedback from stakeholders.
- **Deployment:**
    - Hosting on a web server (e.g., Apache).
    - Database setup and configuration.
    - Regular backups and maintenance.

2.2 LANGUAGES

- **Frontend:**
    - HTML5, CSS3: For structuring and styling the web pages.
    - JavaScript: For interactive features and dynamic content.
- **Backend:**
    - PHP: For server-side scripting and handling business logic.
    - MySQL: For database management and storage of event-related data.

- **Libraries and Frameworks:**
  - PhpSpreadsheet: For exporting data to Excel.
- **Tools:**
  - XAMPP/WAMP: For local development environment setup.
  - PhpMyAdmin: For database management.
- **IDE:** Visual Studio Code or any preferred IDE for development.

CHAPTER 3

REQUIREMENTS AND ANALYSIS

3.1 REQUIREMENT SPECIFICATION

Requirements refer to the conditions or capabilities that must be met or possessed by a system, component, or software product to satisfy a contract, standard, specification, or other formally imposed documents. Requirements are the foundation for the development of any project and are critical for ensuring that the final product meets the needs and expectations of stakeholders. Requirements can be classified into different types:

1. User Requirements

1. User Authentication

- Login/Logout: Secure authentication for both administrators and students.
- Role-Based Access: Different functionalities accessible based on user roles.

2. Event Management

- Create Events: Administrators can add new events with details like event name, date, venue, description, and participant limit.
- Update Events: Administrators can modify existing event details.
- Delete Events: Administrators can remove events from the system.
- View Events: Both administrators and students can view a list of all events with relevant details.

3. Student Registration

- Register for Events: Students can register for events and receive confirmation.
- Cancel Registration: Students can cancel their registration for events.
- View Registered Events: Students can see the events they are registered for.

5. Data Export

- **Export to Excel:** Administrators can export event details and participant lists to Excel files for record-keeping and reporting.

  System Requirements

2. Functional Requirements

- **Database Management**
  - Store user information, event details, and registration data securely.
  - Provide efficient data retrieval for display and reporting.
- **Event CRUD Operations**
  - Enable administrators to create, read, update, and delete event records in the database.
  - Ensure data integrity and validation during CRUD operations.
- **Registration Management**
  - Allow students to register and cancel registration for events.
  - Update the database with registration changes in real-time.
- **Data Export**
  - Implement functionality to export event and registration data to Excel format.
  - Ensure exported data is accurate and formatted properly.

Non-Functional Requirements

- **Performance**
  - The system should handle multiple concurrent users efficiently.
  - Response times for data retrieval and updates should be minimal.
- **Scalability**
  - The system should be able to accommodate an increasing number of users and events.
  - Design the database and application logic to handle growth without significant performance degradation.
- **Security**
  - Implement strong password policies and encryption for sensitive data.
  - Protect against common vulnerabilities such as SQL injection and cross-site scripting (XSS).
- **Usability**
  - The user interface should be intuitive and easy to navigate.
  - Provide clear instructions and feedback to users for all actions.

- **Maintainability**
    - The system code should be modular and well-documented.
    - Use industry-standard frameworks and libraries to facilitate maintenance and updates.

3.2 HARDWARE AND SOFTWARE REQUIREMENTS

Hardware Requirements

Hardware requirements specify the physical components needed to support the system's development, deployment, and operation. For a college event management system, the hardware requirements can be divided into server-side and client-side components

- **Server Requirements:**
    - Processor: Intel Core i5 or equivalent
    - RAM: 8 GB or higher
    - Storage: 100 GB HDD or SSD
    - Network: Reliable network connection with at least 100 Mbps speed
- **Client Requirements:**
    - Processor: Intel Core i3 or equivalent
    - RAM: 4 GB or higher
    - Storage: 20 GB free space
    - Network: Reliable internet connection

Software Requirements

- **Server-Side:**
    - Operating System: Windows Server 2012 or later, or any Linux distribution such as Ubuntu 18.04 or later
    - Web Server: Apache 2.4 or Nginx
    - Database Server: MySQL 5.7 or later

- Scripting Language: PHP 7.4 or later
- Additional Libraries: Composer for PHP dependency management, PHPExcel library for Excel export functionality

➢ **Client-Side:**

- Operating System: Windows 10, macOS, or any Linux distribution
- Web Browser: Latest versions of Google Chrome, Mozilla Firefox, Microsoft Edge, or Safari
- Additional Software: Microsoft Excel or compatible spreadsheet software for viewing exported Excel files

➢ **Development Environment:**

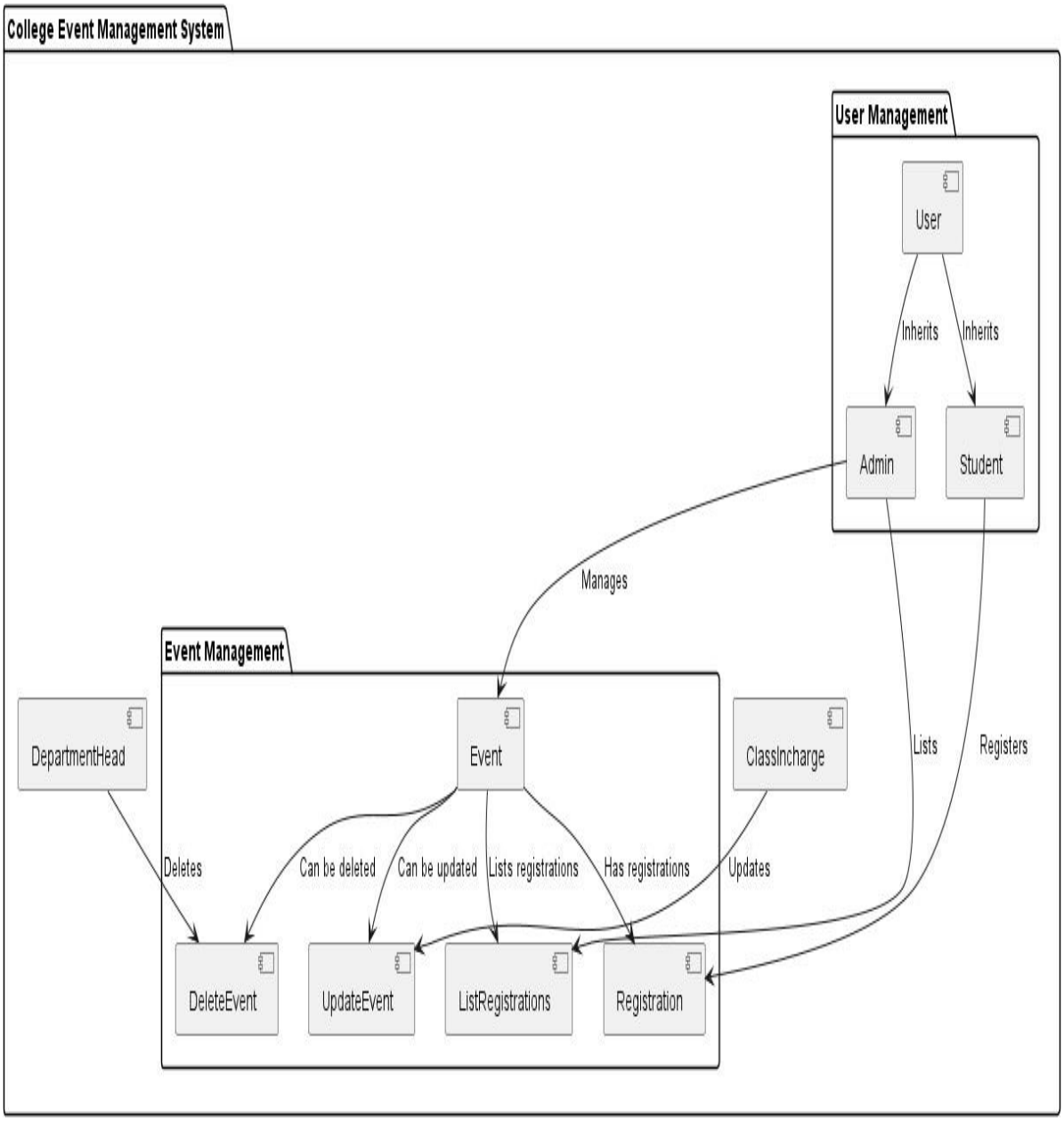
- IDE/Editor: Visual Studio Code, PHPStorm, or any other preferred IDE with PHP support
- Version Control: Git with GitHub or GitLab for repository management
- Local Server Setup: XAMPP, WAMP, or MAMP for local development and testing

➢ **Frameworks and Libraries:**

- Front-End: jQuery for enhanced JavaScript functionality, Bootstrap for responsive design (optional)
- Back-End: PHP for server-side scripting
- Excel Export: PHPExcel library for generating Excel files

3.2 ARCHITECTURE DIAGRAM

An architectural diagram is a visual representation that outlines the structure and organization of a system, showing the relationships and interactions between its components. It provides a high-level overview of how the system is designed, focusing on components, modules, and their interactions. Architectural diagrams help stakeholders understand the system's layout and facilitate communication among developers, designers, and other project participants.



Event Management Component:

1. **Entities:**

    - **DepartmentHead**: Represents department heads or administrators who oversee events. They have the authority to manage events.
    - **Event:** Represents various events organized by the college. Events have attributes such as EventName, EventDate, and EventLocation. The EventID serves as the primary key.
    - **Registration:** Represents user registrations for events. It includes information like RegistrationDate. The RegistrationID is the primary key.

2. **Actions and Relationships:**

    - **DeleteEvent:** Indicates that an event can be deleted by the DepartmentHead.
    - **Can be deleted:** Connects the DeleteEvent action to the Event entity.
    - **Can be updated:** Indicates that an event's details can be modified.
    - **Lists registrations:** Represents events that display their registrations.
    - **Has registrations:** Connects the Event entity to the Registration entity.

User Management Component:

1. **Entities:**

    - **User:** Represents individuals who interact with the system. Users can be categorized into different roles:
        - **Admins:** Administrators who manage the system and its users.
        - **Inheritants:** Users who inherit certain permissions or roles.
        - **Student:** Regular students who participate in events.

2. **Actions and Relationships:**

    - **Manages:** Indicates that Admins manage users (including DepartmentHeads and Students).

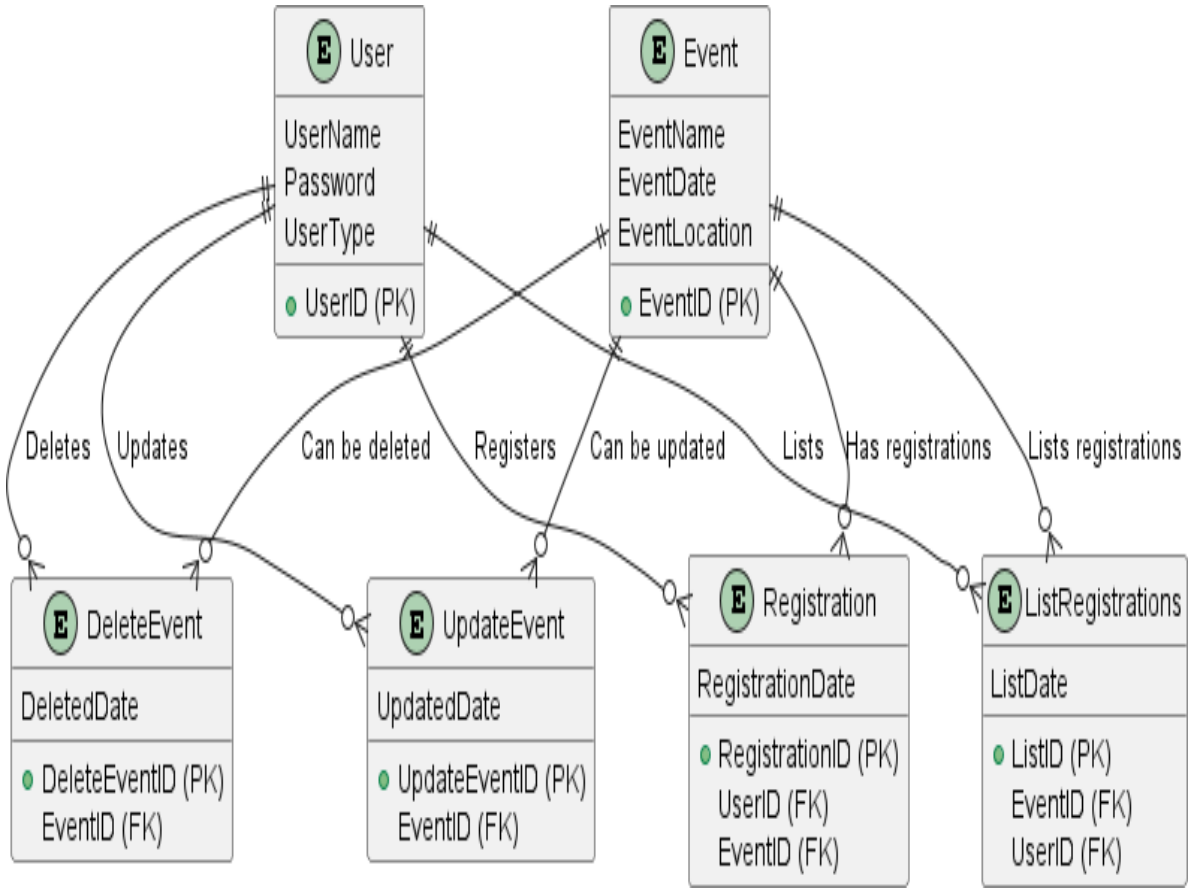
- **Inherits:** Represents the inheritance of roles or permissions from Inheritants to Students.
- **Lists:** Indicates that Students can view event listings.
- **Registers for an event:** Students can register for events.
- **Updates:** Students can update event details.

Cross-Component Interaction:

- The User Management component interacts with the Event Management component:
  - Users (including Admins and Students) participate in events managed by the DepartmentHead.
  - Students can register for events and view event details.

3.3 ER DIAGRAM

An Entity-Relationship (ER) Diagram is a type of flowchart that illustrates how entities such as people, objects, or concepts relate to each other within a system. ER diagrams are widely used in database design and modeling to visually represent the data structures and relationships among data entities.



1. Entities:

- **User:** Represents individuals who interact with the system. Attributes include UserName, Password, UserType, and UserID (which serves as the primary key).
- **Event:** Represents various events organized by the college. Attributes include EventName, EventDate, and EventLocation. The EventID is the primary key.

- **DeleteEvent:** Indicates events that can be deleted. It has a relationship with the Event entity.
- **UpdateEvent:** Represents events that can be updated. It also relates to the Event entity.
- **Registration:** Represents user registrations for events. Attributes include RegistrationDate. The RegistrationID serves as the primary key.
- **ListRegistrations:** Represents events that have registrations. It has a relationship with both the Event and Registration entities.

2. Relationships:

- **Deletes:** Indicates that an event can be deleted.
- **Updates:** Indicates that an event can be updated.
- **Can be deleted:** Connects the DeleteEvent entity to the Event entity.
- **Registers:** Represents the registration of users for events.
- **Can be updated:** Connects the UpdateEvent entity to the Event entity.
- **Lists:** Indicates that an event can be listed.
- **Has registrations:** Connects the Event entity to the Registration entity.
- **Lists registrations:** Indicates that an event can list its registrations.

3. Primary Keys (PK):

- UserID in the User entity.
- EventID in the Event entity.
- RegistrationID in the Registration entity.
- ListID in the ListRegistrations entity.

4. Foreign Keys (FK):

- EventID in the DeleteEvent, UpdateEvent, and ListRegistrations entities.
- UserID in the ListRegistrations entity.

3.5 NORMALIZATION

we identify the main entities and their attributes based on the functions of the college event management system:

1. **User**
    - UserID (Primary Key)
    - UserName
    - Password
    - UserType
2. **Event**
    - EventID (Primary Key)
    - EventName
    - EventDate
    - EventLocation
3. **Registration**
    - RegistrationID (Primary Key)
    - UserID (Foreign Key referencing User.UserID)
    - EventID (Foreign Key referencing Event.EventID)
    - RegistrationDate
4. **UpdateEvent**
    - UpdateEventID (Primary Key)
    - EventID (Foreign Key referencing Event.EventID)
    - UpdatedDate
5. **DeleteEvent**
    - DeleteEventID (Primary Key)
    - EventID (Foreign Key referencing Event.EventID)
    - DeletedDate
6. **ListRegistrations**
    - ListID (Primary Key)
    - EventID (Foreign Key referencing Event.EventID)
    - UserID (Foreign Key referencing User.UserID)
    - ListDate

Step 2: Apply First Normal Form (1NF)

In 1NF, we ensure that each table has atomic values and no repeating groups or arrays. This is usually achieved by breaking down composite attributes into separate columns.

Step 3: Apply Second Normal Form (2NF)

In 2NF, we ensure that all attributes are fully functionally dependent on the entire primary key. If any attribute is functionally dependent on only part of the primary key, we create separate tables to resolve this.

- User and Event tables are already in 2NF as all attributes depend on their respective primary keys.
- Registration, UpdateEvent, DeleteEvent, and ListRegistrations tables each have a composite primary key, so we need to check for partial dependencies.

Step 4: Apply Third Normal Form (3NF)

In 3NF, we eliminate transitive dependencies. If an attribute depends on another non-key attribute, it should be moved to its own table.

- **Registration table**: UserID and EventID depend on RegistrationID (Primary Key). No transitive dependencies.
- **UpdateEvent table:** EventID depends on UpdateEventID (Primary Key). No transitive dependencies.
- **DeleteEvent table:** EventID depends on DeleteEventID (Primary Key). No transitive dependencies.
- **ListRegistrations table:** EventID and UserID depend on ListID (Primary Key). No transitive dependencies.

Final Normalized Schema

Based on the normalization process, we have a set of tables that are in 3NF, ensuring data integrity and minimizing redundancy:

- User
- Event
- Registration
- UpdateEvent
- DeleteEvent
- ListRegistrations

CHAPTER 4

PROGRAM CODE

4.1 Admin page:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Admin Page</title>
<style>
body {
    font-family: Arial, sans-serif;
}

.container {
    max-width: 800px;
    margin: 0 auto;
    padding: 20px;
    background-color: #f5f5f5;
    border-radius: 10px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

.header {
    text-align: center;
    margin-bottom: 20px;
}

.navbar {
    display: flex;
```

```
    justify-content: space-between;
    background-color: #333;
    border-radius: 5px;
    padding: 10px;
}

.navbar a {
    color: #fff;
    text-decoration: none;
    padding: 8px 15px;
    border-radius: 5px;
}

.navbar a:hover {
    background-color: #555;
}

.content {
    background-color: #fff;
    padding: 20px;
    border-radius: 5px;
}

.footer {
    text-align:  center;
    margin-top: 20px;
    color: #777;
}
li{
    list-style: none;
```

```
    padding-top: 10px;
    font-size: 20px;
}
i{
    padding-right: 10px;
    font-size: 30px;
}
a{
    color: black;
}
</style>
</head>
<body>
<div class="container">
    <div class="header">
        <h1>Welcome Admin</h1>
    </div>
    <div class="navbar">
    <a href="#" class="menu-bar" onclick="showhome()">Home</a>
    <a href="#" class="menu-bar" onclick="showevent()">Add Event</a>
    <a href="#" class="menu-bar" onclick="showaddedevent()">Added Events</a>
    <a href="#" class="menu-bar" onclick="showstdregister()">Register Student List</a>
    <a href="#" class="menu-bar" onclick="showcontact()">contact us</a>
    <a href="adminlogin.php">logout</a>

    </div>
    <div id="home">
        <h2>Home</h2>
        <p>The College Event Management System is a comprehensive platform designed to streamline and enhance the planning, organization, and execution of various events within a college or university setting. It encompasses a range of features and functionalities aimed at
```

simplifying event management tasks, improving communication, and ensuring successful events. .

© SK College Event Management System

<h2>Add Events</h2>
<ul class="accordion">

<input type="radio" name="accordion" id="first">
<label for="first">Cultural Events</label>

These include music concerts, dance performances, drama plays, fashion shows, and talent competitions
<img src="adminimage/cultural.jpg">
Add culturals

<input type="radio" name="accordion" id="second">
<label for="second">Sports Events</label>

```
            <p>Colleges often organize sports tournaments, athletic meets, and intercollegiate sports competitions in various disciplines like football, basketball, cricket, athletics, etc.</p>
            <img src="adminimage/sports.jpg">
            <a href="sports.php">Add sports event</a>
        </div>
    </li>

    <li>
        <input type="radio" name="accordion" id="third">
        <label for="third">Academic Events</label>
        <div class="content">
            <p>These include seminars, workshops, conferences, symposiums, and guest lectures by experts in different fields.</p>
            <img src="adminimage/academic.jpg">
            <a href="academic.php">Add academic event</a>
        </div>
    </li>

    <li>
        <input type="radio" name="accordion" id="fourth">
        <label for="fourth">Alumni Events</label>
        <div class="content">
            <p>Colleges may organize alumni reunions, networking events, and alumni mentorship programs.</p>
            <img src="adminimage/alumini.jpg">
            <a href="alumni.php">Add alumni event</a>
        </div>
    </li>

    <li>
        <input type="radio" name="accordion" id="fifth">
```

```
            <label for="fifth">Placement and Training Events</label>
            <div class="content">
                <p>These include job fairs, campus recruitment drives, career counseling sessions, and resume-building workshops.</p>
                <img src="adminimage/placement.jpg">
                <a href="placement.php">Add Placement and Training event</a>
            </div>
        </li>
    </ul>

</div>

<div id="addedevent">
    <h2>Added Events</h2>
    <ul class="accordion">
        <li>
            <input type="radio" name="accordion" id="sixth">
            <label for="sixth">Cultural Events</label>
            <div class="content">
                <p>These include music concerts, dance performances, drama plays, fashion shows, and talent competitions</p>
                <img src="adminimage/cultural.jpg">
                <a href="addedculturalsevent.php">Added cultural events</a>
            </div>
        </li>

        <li>
            <input type="radio" name="accordion" id="seventh">
            <label for="seventh">Sports Events</label>
```

```
            <div class="content">
                <p>Colleges often organize sports tournaments, athletic meets, and intercollegiate sports competitions in various disciplines like football, basketball, cricket, athletics, etc.</p>
                <img src="adminimage/sports.jpg">
                <a href="addedsportevent.php">Added sports event</a>
            </div>
        </li>

        <li>
            <input type="radio" name="accordion" id="eighth">
            <label for="eighth">Academic Events</label>
            <div class="content">
                <p>These include seminars, workshops, conferences, symposiums, and guest lectures by experts in different fields.</p>
                <img src="adminimage/academic.jpg">
                <a href="addedacademicevent.php">Added academic event</a>
            </div>
        </li>

        <li>
            <input type="radio" name="accordion" id="ninth">
            <label for="ninth">Alumni Events</label>
            <div class="content">
                <p>Colleges may organize alumni reunions, networking events, and alumni mentorship programs.</p>
                <img src="adminimage/alumini.jpg">
                <a href="addedalumnievent.php">Added alumni event</a>
            </div>
        </li>

        <li>
```

```
            <input type="radio" name="accordion" id="tenth">
            <label for="tenth">Placement and Training Events</label>
            <div class="content">
              <p>These include job fairs, campus recruitment drives, career counseling sessions, and resume-building workshops.</p>
              <img src="adminimage/placement.jpg">
              <a href="addedplacementevent.php">Added Placement and Training event</a>
            </div>
          </li>
        </ul>

    </div>

    <div id="studentregisterevent">
        <h2>Student Register Events</h2>
        <ul class="accordion">
          <li>
            <input type="radio" name="accordion" id=" Eleven">
            <label for=" Eleven">Cultural Events</label>
            <div class="content">
              <p>These include music concerts, dance performances, drama plays, fashion shows, and talent competitions</p>
              <img src="adminimage/cultural.jpg">
              <a href="culturalsregisterstudents.php">Registered list of cultural events</a>
            </div>
          </li>

          <li>
            <input type="radio" name="accordion" id="Twelve">
            <label for="Twelve">Sports Events</label>
```

```
            <div class="content">
                <p>Colleges often organize sports tournaments, athletic meets, and intercollegiate sports competitions in various disciplines like football, basketball, cricket, athletics, etc.</p>
                <img src="adminimage/sports.jpg">
                <a href="sportsregisterstudent.php">Registered list of sports events</a>
            </div>
        </li>

        <li>
            <input type="radio" name="accordion" id="Thirteen">
            <label for="Thirteen">Academic Events</label>
            <div class="content">
                <p>These include seminars, workshops, conferences, symposiums, and guest lectures by experts in different fields.</p>
                <img src="adminimage/academic.jpg">
                <a href="academicregisterstudent.php">Registered list of academic events</a>
            </div>
        </li>

        <li>
            <input type="radio" name="accordion" id="Fourteen">
            <label for="Fourteen">Alumni Events</label>
            <div class="content">
                <p>Colleges may organize alumni reunions, networking events, and alumni mentorship programs.</p>
                <img src="adminimage/alumini.jpg">
                <a href="alumniregisterstudent.php">Registered list of alumni events</a>
            </div>
        </li>

        <li>
```

```
            <input type="radio" name="accordion" id=" Fifteen">
            <label for=" Fifteen">Placement and Training Events</label>
            <div class="content">
              <p>These include job fairs, campus recruitment drives, career counseling sessions, and resume-building workshops.</p>
              <img src="adminimage/placement.jpg">
              <a href="placementregisterstudent.php">Registered list of Placement events</a>
            </div>
          </li>
        </ul>

    </div>

    <div id="contactus">
        <h1> contact</h1>
        <ul>
            <li><i class="fa-solid fa-user"></i> Kaviarasu.M</li>
            <li><i class="fa-solid fa-phone"></i> 9360108857</li>
            <li><i class="fa-solid fa-envelope"></i> 220701517@rajalakshmi.edu.in</li>

            <li><i class="fa-solid fa-user"></i> Saranraj.A</li>
            <li><i class="fa-solid fa-phone"></i> 8248823072</li>
            <li><i class="fa-solid fa-envelope"></i> 220701510@rajalakshmi.edu.in</li>
        </ul>
    </div>


</div>


<style>
```

```
    a{
        color: white;
        text-align: center;
        cursor: pointer;
    }
    #addevent{
        display: none;
    }
    #addedevent{
        display: none;
    }
    #studentregisterevent{
        display: none;
    }
    #contactus{
        display: none;
    }
    img{
        width: 35%;
    }
    #addevent{
    padding: 10px 12%;
    text-align: center;
    font-size: 35px;
    font-weight: 600;
}

#addedevent{
    padding: 10px 12%;
    text-align: center;
```

```
    font-size: 35px;
    font-weight: 600;
}
#studentregisterevent{
    padding: 10px 12%;
    text-align: center;
    font-size: 35px;
    font-weight: 600;
}
.accordion{
    margin: 60px auto;
    width: 100%;
    max-width: 750px;
}
.accordion li{
    list-style: none;
    width: 100%;
    padding: 5px;
}
.accordion p{
    font-size: 23px;
    font-weight: 500;
}
.accordion li label{
    display: flex;
    align-items: center;
    padding: 20px;
    font-size: 23px;
    font-weight: 500;
    background: #303030;
```

```
    margin-bottom: 2px;
    cursor: pointer;
    position: relative;
}
label::after{
    content: '+';
    font-size: 34px;
    position: absolute;
    right: inherit;
    right: 20px;
    transition: transform 0.5s;
}
input[type="radio"]{
    display: none;
}
.accordion .content{
    background: #303030;
    text-align: left;
    padding: 0 20px;
    max-height: 0;
    overflow: hidden;
    transition: max-height 0.5s, padding 0.5s;
}
.accordion input[type="radio"]:checked + label +.content{
    max-height: 600px;
    padding: 30px;

}
.accordion input[type="radio"]:checked + label::after{
    transform: rotate(135deg);
```

```
}

#addevent{
    font-size: 23px;
    color: white;
}
#addevent h2{
    color: black;
}

#addedevent{
    font-size: 23px;
    color: white;
}
#addedevent h2{
    color: black;
}

#studentregisterevent{
    font-size: 23px;
    color: white;
}
#studentregisterevent h2{
    color: black;
}

body {
        background-size: cover;
        background-position: center;
        background-repeat: no-repeat;
```

```
        background-image: url('adminimage/adminhomebg.jpg');
}
</style>
<script>
    function showhome(){
    document.getElementById('home').style.display='block';
    document.getElementById('addevent').style.display='none';
    document.getElementById('addedevent').style.display='none';
    document.getElementById('studentregisterevent').style.display='none';
    document.getElementById('contactus').style.display='none';
}

function showevent(){
    document.getElementById('home').style.display='none';
    document.getElementById('addevent').style.display='block';
    document.getElementById('addedevent').style.display='none';
    document.getElementById('studentregisterevent').style.display='none';
    document.getElementById('contactus').style.display='none';
}

function showaddedevent(){
    document.getElementById('home').style.display='none';
    document.getElementById('addevent').style.display='none';
    document.getElementById('addedevent').style.display='block';
    document.getElementById('studentregisterevent').style.display='none';
    document.getElementById('contactus').style.display='none';
}

function showstdregister(){
    document.getElementById('home').style.display='none';
```

```
    document.getElementById('addevent').style.display='none';
    document.getElementById('addedevent').style.display='none';
    document.getElementById('studentregisterevent').style.display='block';
    document.getElementById('contactus').style.display='none';
}

function showcontact(){
    document.getElementById('home').style.display='none';
    document.getElementById('addevent').style.display='none';
    document.getElementById('addedevent').style.display='none';
    document.getElementById('studentregisterevent').style.display='none';
    document.getElementById('contactus').style.display='block';
}
</script>

<script src="https://kit.fontawesome.com/47326d26e9.js"
crossorigin="anonymous"></script>

</body>
</html>
```

4.2 Student page:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Student Page</title>
<style>
body {
    font-family: Arial, sans-serif;
}

.container {
    max-width: 800px;
    margin: 0 auto;
    padding: 20px;
    background-color: #f5f5f5;
    border-radius: 10px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

.header {
    text-align: center;
    margin-bottom: 20px;
}

.navbar {
    display: flex;
    justify-content: space-between;
    background-color: #333;
```

```
    border-radius: 5px;
    padding: 10px;
}

.navbar a {
    color: #fff;
    text-decoration: none;
    padding: 8px 15px;
    border-radius: 5px;
}

.navbar a:hover {
    background-color: #555;
}

.content {
    background-color: #fff;
    padding: 20px;
    border-radius: 5px;
}

.footer {
    text-align:  center;
    margin-top: 20px;
    color: #777;
}
li{
    list-style: none;
    padding-top: 10px;
    font-size: 20px;
```

```
}
i{
    padding-right: 10px;
    font-size: 30px;
}
</style>
</head>
<body>
<div class="container">
    <div class="header">
        <h1>Welcome Student</h1>
    </div>
    <div class="navbar">
    <a href="#" class="menu-bar" onclick="showhome()">Home</a>
    <a href="#" class="menu-bar" onclick="showregister()">Register for Events</a>
    <a href="#" class="menu-bar" onclick="showcontact()">contact us</a>
    <a href="login.php">logout</a>

    </div>
    <div id="home">
        <h2>Home</h2>
        <p>The College Event Management System is a comprehensive platform designed to streamline and enhance the planning, organization, and execution of various events within a college or university setting. It encompasses a range of features and functionalities aimed at simplifying event management tasks, improving communication, and ensuring successful events. .</p>
        <div class="footer">
        <p>&copy; SK College Event Management System</p>
    </div>
    </div>
```

```
<div id="addevent">
    <h2>Register for Events</h2>
    <ul class="accordion">
        <li>
            <input type="radio" name="accordion" id="first">
            <label for="first">Cultural Events</label>
            <div class="content">
                <p>These include music concerts, dance performances, drama plays, fashion shows, and talent competitions</p>
                <img src="studentimage/cultural.jpg">
                <a href="culturalevents.php">Register for Cultural event</a>
            </div>
        </li>

        <li>
            <input type="radio" name="accordion" id="second">
            <label for="second">Sports Events</label>
            <div class="content">
                <p>Colleges often organize sports tournaments, athletic meets, and intercollegiate sports competitions in various disciplines like football, basketball, cricket, athletics, etc.</p>
                <img src="studentimage/sports.jpg">
                <a href="sportsevent.php">Register for sports event</a>
            </div>
        </li>

        <li>
            <input type="radio" name="accordion" id="third">
            <label for="third">Academic Events</label>
            <div class="content">
```

```
            <p>These include seminars, workshops, conferences, symposiums, and guest lectures by experts in different fields.</p>
            <img src="studentimage/academic.jpg">
            <a href="academicevent.php">Register for academic event</a>
          </div>
        </li>

        <li>
          <input type="radio" name="accordion" id="fourth">
          <label for="fourth">Alumni Events</label>
          <div class="content">
            <p>Colleges may organize alumni reunions, networking events, and alumni mentorship programs.</p>
            <img src="studentimage/alumini.jpg">
            <a href="alumnievent.php">Register for alumni event</a>
          </div>
        </li>

        <li>
          <input type="radio" name="accordion" id="fifth">
          <label for="fifth">Placement and Training Events</label>
          <div class="content">
            <p>These include job fairs, campus recruitment drives, career counseling sessions, and resume-building workshops.</p>
            <img src="studentimage/placement.jpg">
            <a href="placementevent.php">Register for Placement and Training event</a>
          </div>
        </li>
      </ul>

    </div>
```

```
    <div id="contactus">
        <h1> contact</h1>
        <ul>
            <li><i class="fa-solid fa-user"></i> Kaviarasu.M</li>
            <li><i class="fa-solid fa-phone"></i> 9360108857</li>
            <li><i class="fa-solid fa-envelope"></i> 220701517@rajalakshmi.edu.in</li>

            <li><i class="fa-solid fa-user"></i> Saranraj.A</li>
            <li><i class="fa-solid fa-phone"></i> 8248823072</li>
            <li><i class="fa-solid fa-envelope"></i> 220701510@rajalakshmi.edu.in</li>
        </ul>
    </div>
</div>
<style>
    a{
        color: white;
        text-align: center;
        cursor: pointer;
    }
    #addevent{
        display: none;
    }
    #addedevent{
        display: none;
    }
    #contactus{
        display: none;
    }
    img{
        width: 35%;
```

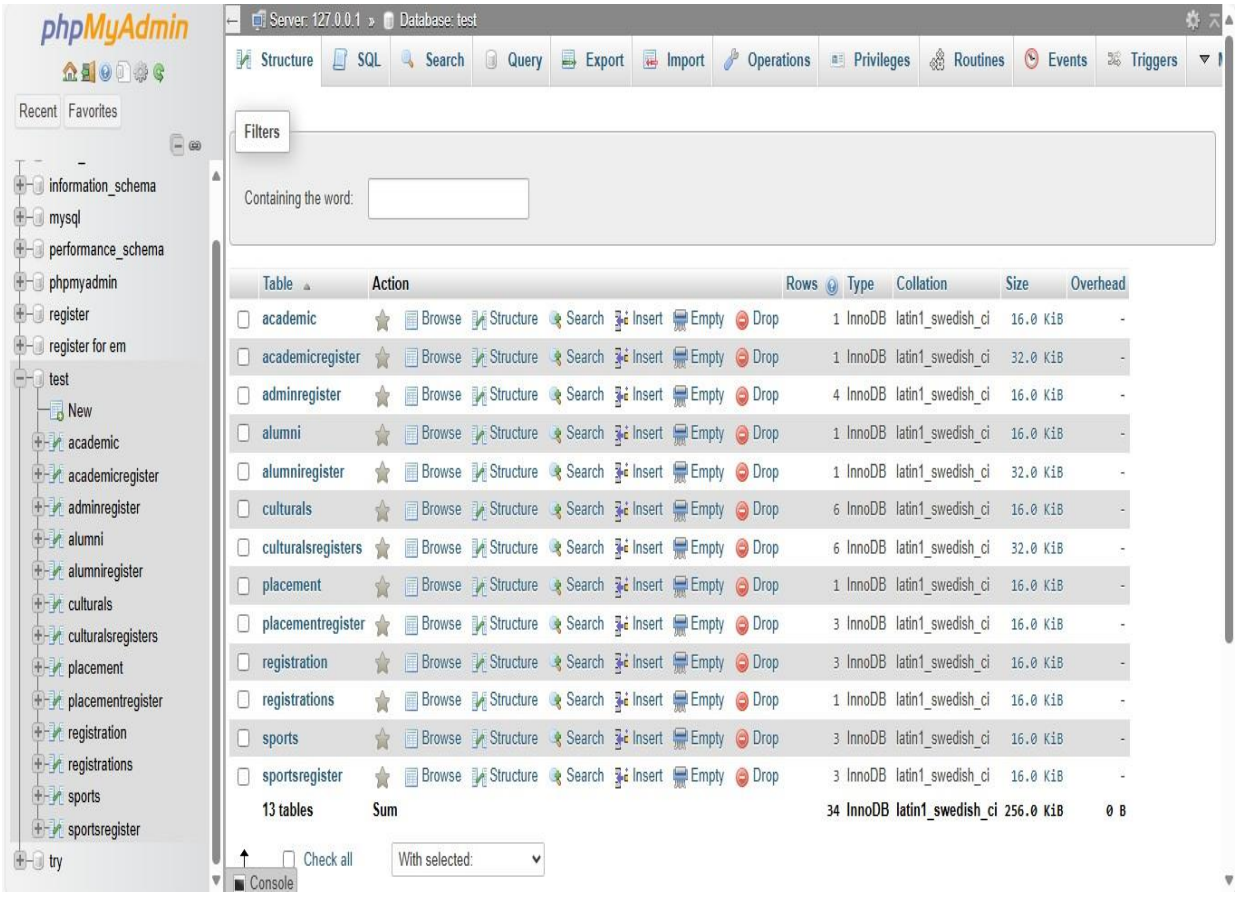
```
    }
    #addevent{
    padding: 10px 12%;
    text-align: center;
    font-size: 35px;
    font-weight: 600;
}

#addedevent{
    padding: 10px 12%;
    text-align: center;
    font-size: 35px;
    font-weight: 600;
}
.accordion{
    margin: 60px auto;
    width: 100%;
    max-width: 750px;
}
.accordion li{
    list-style: none;
    width: 100%;
    padding: 5px;
}
.accordion p{
    font-size: 23px;
    font-weight: 500;
}
.accordion li label{
    display: flex;
```

```
    align-items: center;
    padding: 20px;
    font-size: 23px;
    font-weight: 500;
    background: #303030;
    margin-bottom: 2px;
    cursor: pointer;
    position: relative;
}
label::after{
    content: '+';
    font-size: 34px;
    position: absolute;
    right: inherit;
    right: 20px;
    transition: transform 0.5s;
}
input[type="radio"]{
    display: none;
}
.accordion .content{
    background: #303030;
    text-align: left;
    padding: 0 20px;
    max-height: 0;
    overflow: hidden;
    transition: max-height 0.5s, padding 0.5s;
}
.accordion input[type="radio"]:checked + label +.content{
    max-height: 600px;
```

```
    padding: 30px;

}
.accordion input[type="radio"]:checked + label::after{
    transform: rotate(135deg);
}

#addevent{
    font-size: 23px;
    color: white;
}
#addevent h2{
    color: black;
}

#addedevent{
    font-size: 23px;
    color: white;
}
#addedevent h2{
    color: black;
}

body {
        background-size: cover;
        background-position: center;
        background-repeat: no-repeat;
        background-image: url('studentimage/adminhomebg.jpg');
}

</style>
```

```
<script>
    function showhome(){
    document.getElementById('home').style.display='block';
    document.getElementById('addevent').style.display='none';
    document.getElementById('contactus').style.display='none';
}

function showregister(){
    document.getElementById('home').style.display='none';
    document.getElementById('addevent').style.display='block';
    document.getElementById('contactus').style.display='none';
}

function showcontact(){
    document.getElementById('home').style.display='none';
    document.getElementById('addevent').style.display='none';
    document.getElementById('contactus').style.display='block';
}
</script>

<script src="https://kit.fontawesome.com/47326d26e9.js"
crossorigin="anonymous"></script>

</body>
</html>
```

4.3 PHP Database:



CHAPTER 5

RESULTS AND DISCUSSION

5.1 Results:

1. **Functionalities Implemented:**
    - User registration and login functionality are implemented.
    - Event creation, update, deletion, and listing functionalities are available.
    - User registration data is stored in the database.
    - Event data is stored and managed in the database.
    - Registration for events is recorded in the database.
2. **Database Interaction:**
    - The system connects to a MySQL database using PHP's MySQLi extension.
    - SQL queries are used to insert, update, delete, and retrieve data from the database.
    - User authentication and event management operations are performed securely through prepared statements to prevent SQL injection.
3. **User Interface:**
    - Basic HTML templates are used for the user interface, including header and footer sections for navigation.
    - Registration and login forms are provided for users.
    - Event listing is displayed in a structured format.

5.2 Discussion:

1. **System Architecture:**
    - The project follows a basic MVC (Model-View-Controller) architecture, where models represent data structures, controllers handle business logic and database operations, and views manage user interface presentation.

- However, in the provided single-page code, the separation of concerns is limited due to the code's compact nature. In a real-world application, it's recommended to separate concerns more clearly for better maintainability.

2. **Database Design and Normalization:**
    - The project demonstrates basic database design principles with tables for users, events, and registrations.
    - Further normalization could be applied to optimize the database structure, such as breaking down user roles into separate tables or ensuring data integrity with foreign key constraints.
3. **Security Considerations:**
    - The project uses prepared statements to mitigate SQL injection attacks.
    - However, additional security measures such as password hashing and session management could enhance overall system security.
4. **Scalability and Extensibility:**
    - The project provides a foundation for an event management system but may require enhancements for scalability and extensibility.
    - For a larger system, features like role-based access control, event categories, notifications, and reporting could be added.
5. **User Experience:**
    - **The user interface is basic and functional. Improvements such as a more intuitive event registration process, event search/filter options, and event details pages could enhance user experience.**
6. **Deployment and Maintenance:**
    - Deployment considerations include configuring the database server, ensuring PHP compatibility, and securing the application environment.
    - Maintenance tasks involve regular updates, bug fixes, and monitoring for security vulnerabilities.

CHAPTER 6

PROJECT OUTCOME

Figure 6.1 Login page:

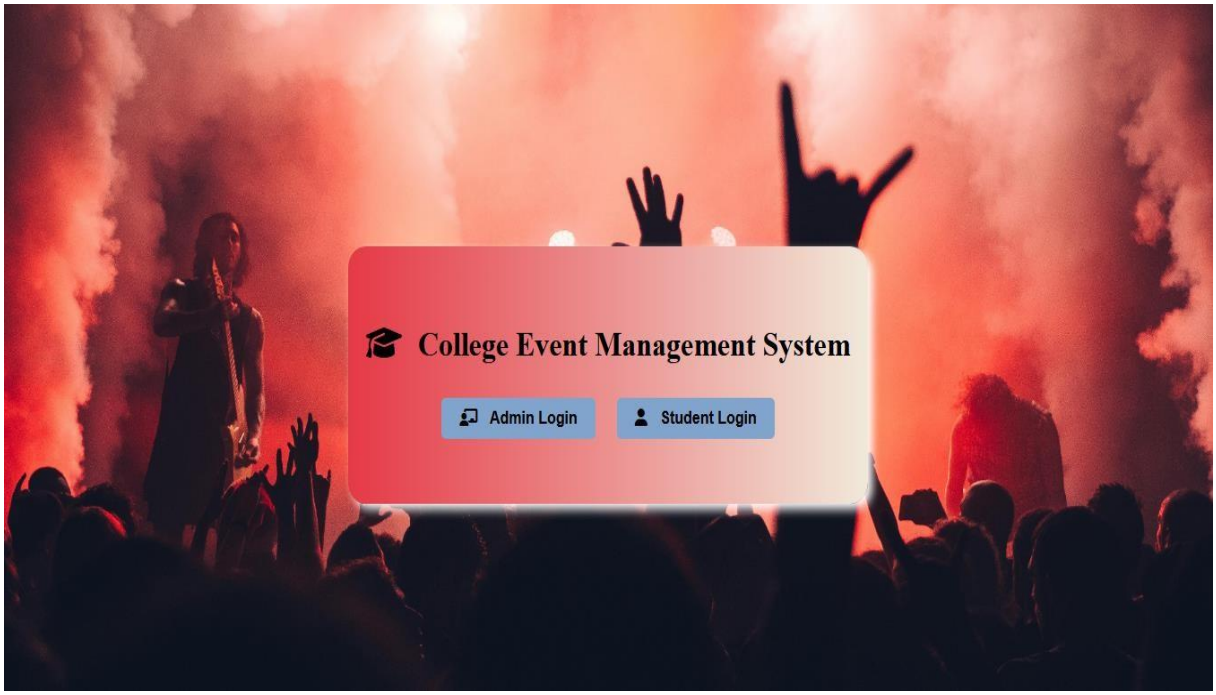


Figure 6.2 Admin Dashboard:

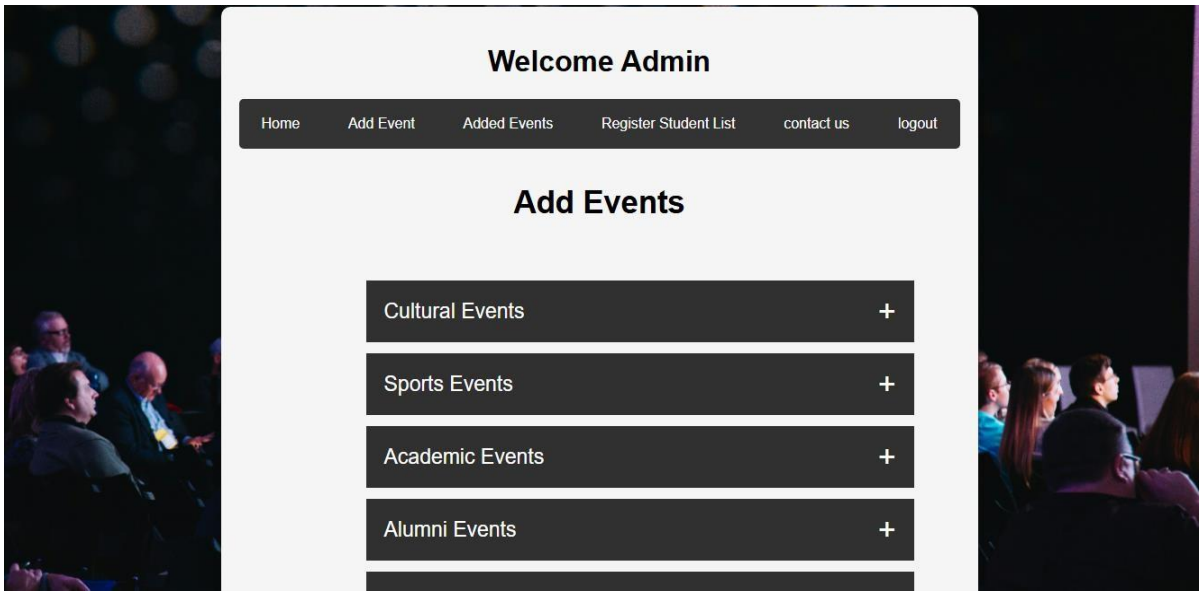


Figure 6.3 Student Dashboard:

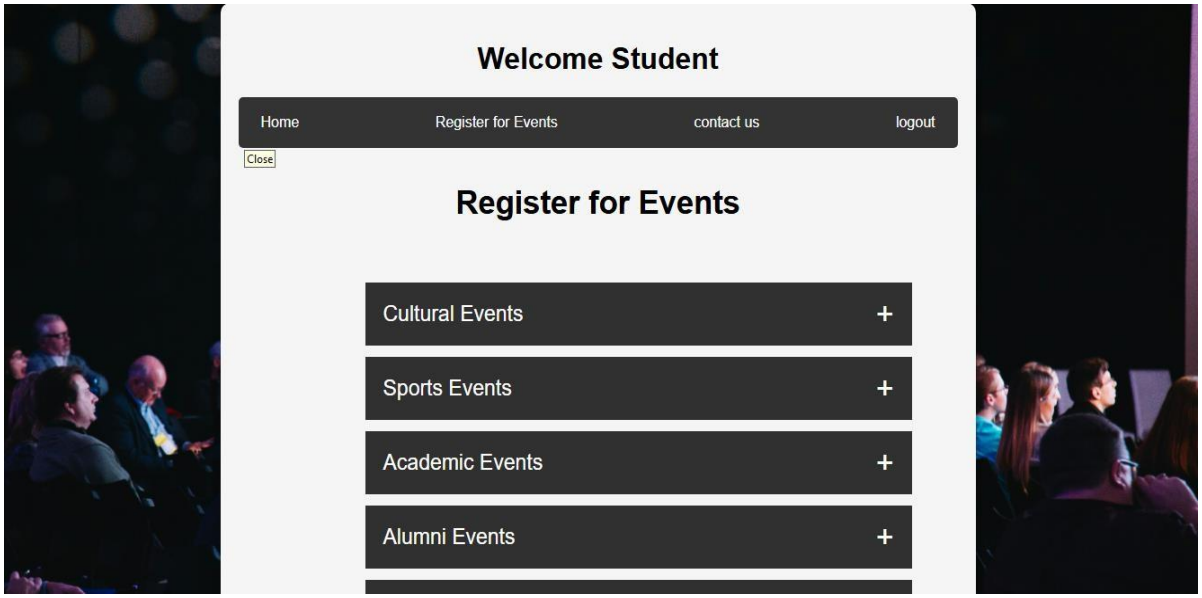


Figure 6.4 Admin Events Update Options:

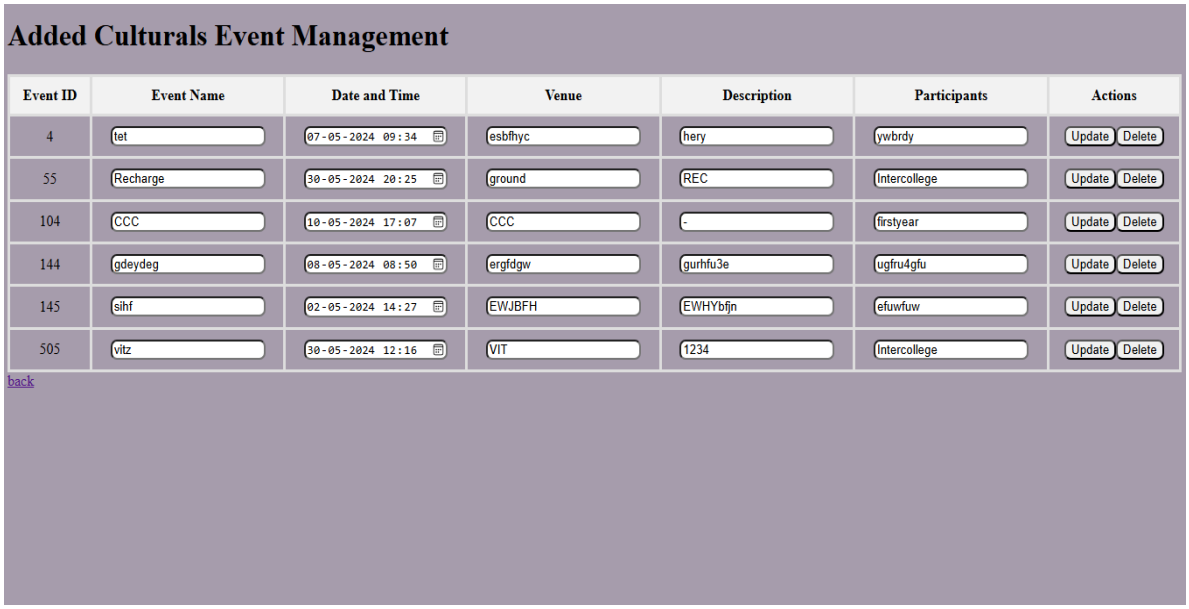
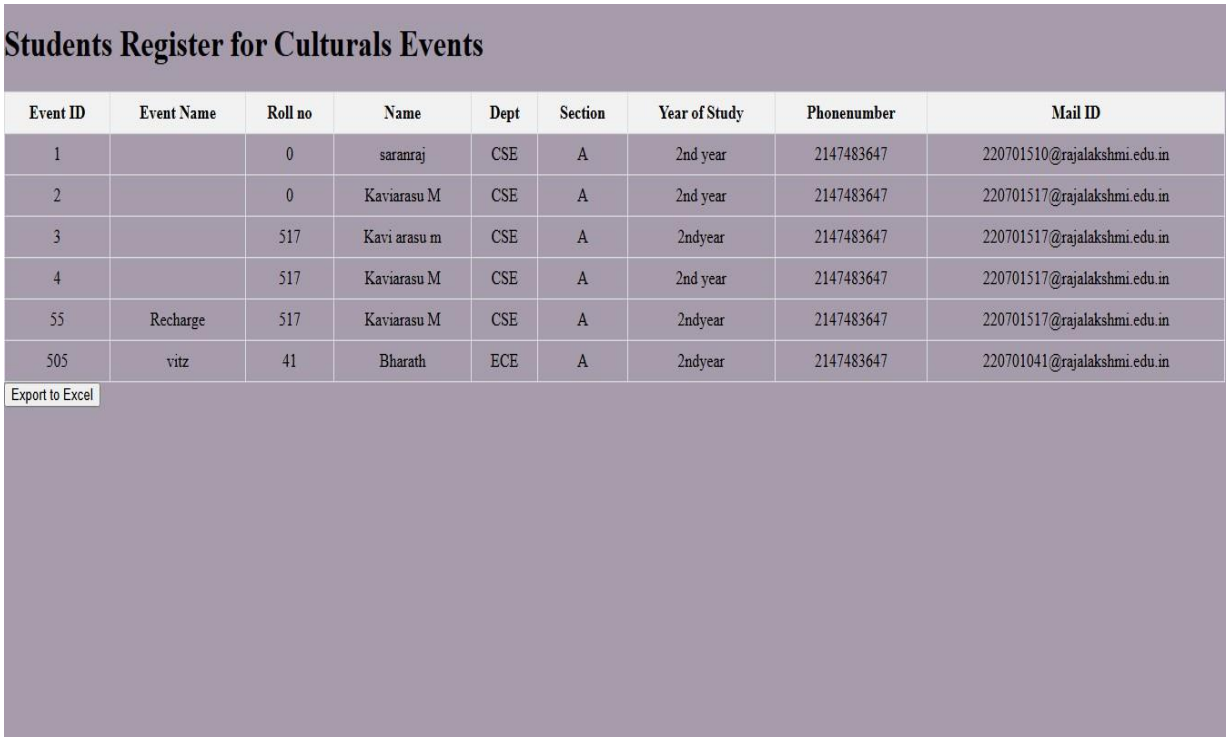


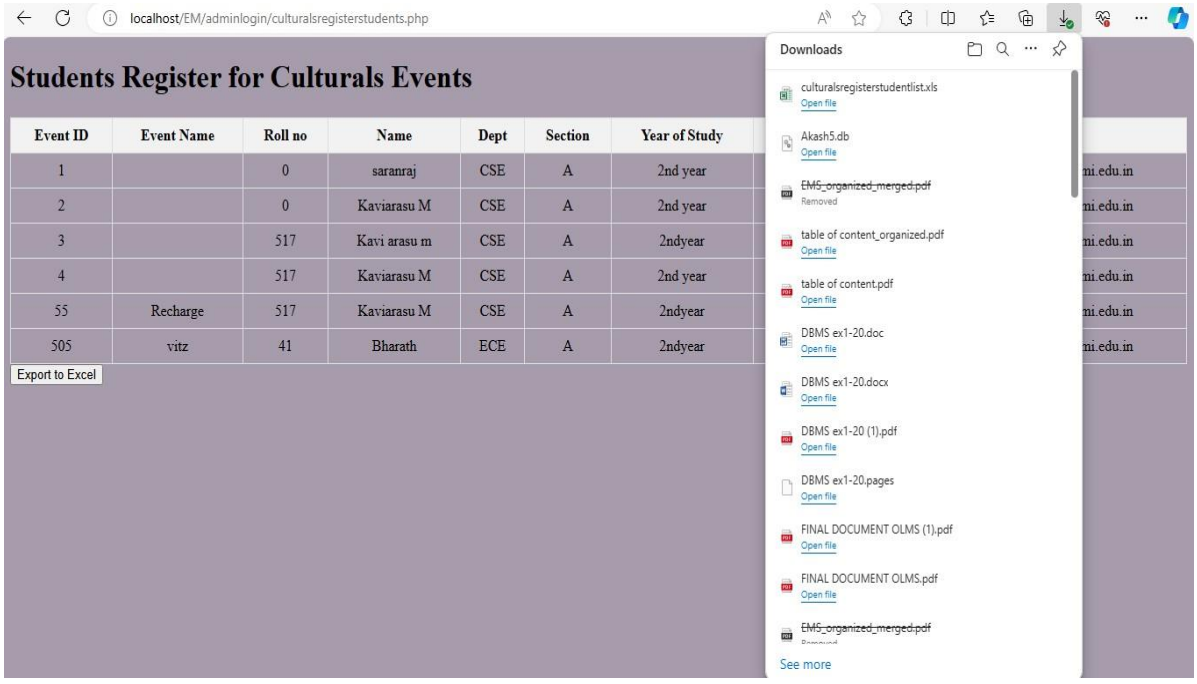
Figure 6.5 Excel Report Generation:



Students Register for Culturals Events

Event ID	Event Name	Roll no	Name	Dept	Section	Year of Study	Phonenumber	Mail ID
1		0	saranraj	CSE	A	2nd year	2147483647	220701510@rajalakshmi.edu.in
2		0	Kaviarasu M	CSE	A	2nd year	2147483647	220701517@rajalakshmi.edu.in
3		517	Kavi arasu m	CSE	A	2ndyear	2147483647	220701517@rajalakshmi.edu.in
4		517	Kaviarasu M	CSE	A	2nd year	2147483647	220701517@rajalakshmi.edu.in
55	Recharge	517	Kaviarasu M	CSE	A	2ndyear	2147483647	220701517@rajalakshmi.edu.in
505	vitz	41	Bharath	ECE	A	2ndyear	2147483647	220701041@rajalakshmi.edu.in

Export to Excel



CHAPTER 7

CONCLUSION

The college event management system project, developed using PHP, represents a functional foundation for handling event-related activities within a college setting. Through features like user registration, event creation, updating, and deletion, as well as event registration for users, the system demonstrates core functionalities crucial for managing college events efficiently. The project's adherence to security practices, including the use of prepared statements to prevent SQL injection, reflects a commitment to data integrity and user protection. However, for real-world deployment, considerations such as password hashing, session management, and enhanced input validation would further bolster the system's security posture. The project's basic MVC architecture, albeit in a single-page format, provides a structured approach to code organization, promoting better maintainability and scalability. Moving forward, enhancements in user experience, such as improved event listing and registration processes, along with advanced features like role-based access control and event categorization, could elevate the system's functionality and user satisfaction. Overall, while the project serves as a functional prototype, continuous refinement and expansion are essential to meet the evolving needs of a comprehensive college event management solution.

CHAPTER 8

REFERENCES

1. Welling, Luke, and Laura Thomson. "PHP and MySQL Web Development." Addison-Wesley Professional, 2016.
2. Ullman, Larry. "PHP and MySQL for Dynamic Web Sites." Peachpit Press, 2017.
3. Freeman, Eric. "Head First Design Patterns." O'Reilly Media, 2004.
4. Fowler, Martin. "Patterns of Enterprise Application Architecture." Addison-Wesley Professional, 2002.
5. Schach, Stephen R. "Object-Oriented and Classical Software Engineering." McGraw-Hill Education, 2016.
6. IEEE Computer Society. "IEEE Xplore Digital Library." https://ieeexplore.ieee.org/.
7. McConnell, Steve. "Code Complete: A Practical Handbook of Software Construction." Microsoft Press, 2004.
8. Martin, Robert C. "Clean Code: A Handbook of Agile Software Craftsmanship." Prentice Hall, 2008.
9. Gamma, Erich, et al. "Design Patterns: Elements of Reusable Object-Oriented Software." Addison-Wesley Professional, 1994.
10. Larman, Craig. "Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design." Pearson, 2004.
11. Martin, Robert C. "Clean Architecture: A Craftsman's Guide to Software Structure and Design." Prentice Hall, 2017.
12. Fowler, Martin. "Refactoring: Improving the Design of Existing Code." Addison-Wesley Professional, 2018.
13. Sommerville, Ian. "Software Engineering." Pearson Education, 2015.
14. Hunt, Andrew, and David Thomas. "The Pragmatic Programmer: Your Journey to Mastery." Addison-Wesley Professional, 2019.

15. McConnell, Steve. "Software Estimation: Demystifying the Black Art." Microsoft Press, 2006.

16. Beck, Kent. "Test Driven Development: By Example." Addison-Wesley Professional, 2002.

17. Fowler, Martin. "Domain-Specific Languages." Addison-Wesley Professional, 2010.

18. Ambler, Scott W. "Agile Modeling: Effective Practices for Extreme Programming and the Unified Process." Wiley, 2002.

19. Shalloway, Alan, and James R. Trott. "Design Patterns Explained: A New Perspective on Object-Oriented Design." Addison-Wesley Professional, 2004.

20. Crockford, Douglas. "JavaScript: The Good Parts." O'Reilly Media, 2008.

21. Richardson, Leonard. "RESTful Web Services: Web services for the real world." O'Reilly Media, 2007.

22. Brooks Jr., Frederick P. "The Mythical Man-Month: Essays on Software Engineering." Addison-Wesley Professional, 1995.

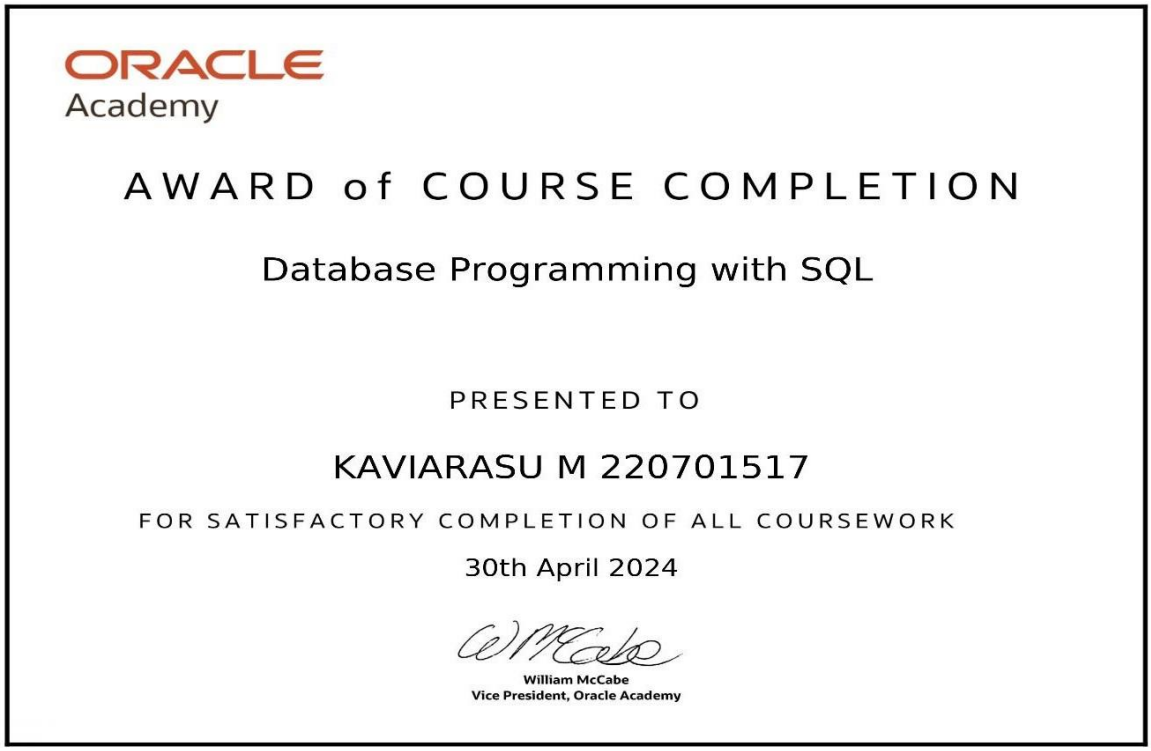
23. Deitel, Paul, et al. "Internet & World Wide Web: How to Program." Prentice Hall, 2007.

24. ISO/IEC/IEEE. "IEEE 12207-2008 - Systems and software engineering - Software life cycle processes." IEEE Standards Association, 2008.

25. Shaw, Mary, and David Garlan. "Software Architecture: Perspectives on an Emerging Discipline." Prentice Hall, 1996.

CHAPTER 9

CERTIFICATES



ORACLE
Academy

AWARD of COURSE COMPLETION

Database Programming with SQL

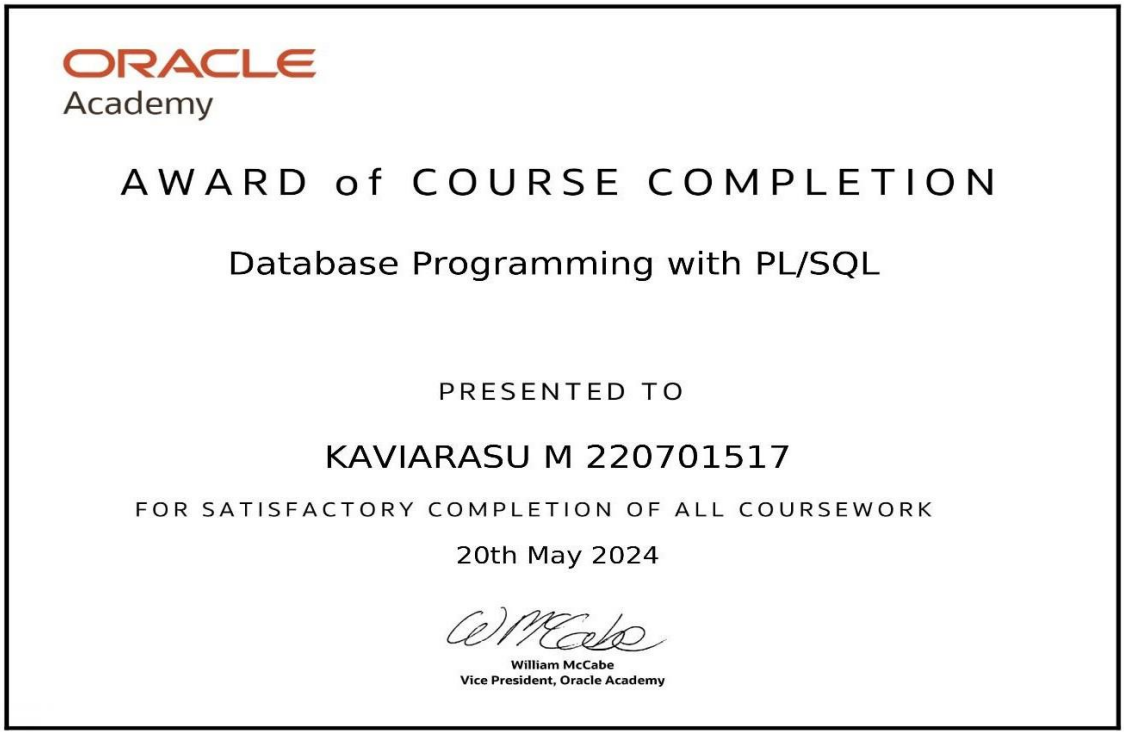
PRESENTED TO

KAVIARASU M 220701517

FOR SATISFACTORY COMPLETION OF ALL COURSEWORK

30th April 2024

William McCabe
Vice President, Oracle Academy



ORACLE
Academy

AWARD of COURSE COMPLETION

Database Programming with PL/SQL

PRESENTED TO

KAVIARASU M 220701517

FOR SATISFACTORY COMPLETION OF ALL COURSEWORK

20th May 2024

William McCabe
Vice President, Oracle Academy

ORACLE
Academy

AWARD of COURSE COMPLETION

Database Programming with SQL

PRESENTED TO

SARAN RAJ A 220701510

FOR SATISFACTORY COMPLETION OF ALL COURSEWORK

6th May 2024

William McCabe
Vice President, Oracle Academy

ORACLE
Academy

AWARD of COURSE COMPLETION

Database Programming with PL/SQL

PRESENTED TO

SARAN RAJ A 220701510

FOR SATISFACTORY COMPLETION OF ALL COURSEWORK

27th May 2024

William McCabe
Vice President, Oracle Academy


Thank you!