EVENT REGISTRATION SYSTEM IN JAVA

A PROJECT REPORT

Submitted by

Divya (23BCS10824)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE ENGINEERING

Chandigarh University

NOV - 2025

BONAFIDE CERTIFICATE

Certified that this project report **"EVENT REGISTRATION SYSTEM IN JAVA"** is the bonafide work of "**Divya (23BCS10824)"** who carried out the project work under my supervision.

SIGNATURE

Kamal Kumar (E13941)

Submitted for the project viva-voce examination held on

INTERNAL EXAMINER EXTERNAL EXAMINER

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my supervisor Kamal Kumar(E13491), Department of Computer Science and Engineering, Chandigarh University, for his/her guidance, motivation, and valuable suggestions during the course of this project.

I am also grateful to Head of the Department of Computer Science and Engineering, for providing the necessary support and encouragement.

My heartfelt thanks to my classmates and friends for their cooperation and feedback, and to my family for their constant support and inspiration.

Lastly, I thank Chandigarh University for providing me the opportunity to work on this project that strengthened my technical and analytical skills.

Divya

23BCS10824

ABSTRACT

The Event Registration System in Java is a comprehensive desktop-based software solution developed to automate and optimize the process of managing events and participant registrations within educational institutions. In many universities and organizations, event management is often carried out manually using spreadsheets or paper records, which can result in inefficiencies, data duplication, and human errors. This project addresses these challenges by developing a user-friendly and reliable system that simplifies event creation, student registration, and data handling through an integrated digital platform.

The application is developed using Java Swing for building an interactive graphical user interface and MySQL as the backend database for secure and efficient data storage. Java Database Connectivity (JDBC) acts as the bridge between the application and database, enabling seamless data operations such as insertion, deletion, updating, and retrieval of records. The system consists of multiple modules including event management, student management, and registration management all accessible from a unified interface.

Through this project, essential software engineering principles such as modular programming, database connectivity, and GUI design have been implemented. The system provides real-time data consistency, improved accessibility, and reduced administrative workload. Testing and validation confirmed that all functionalities perform as intended, ensuring accuracy and reliability of data operations.

Overall, the Event Registration System in Java offers a practical approach to institutional event management, delivering an automated, scalable, and maintainable solution that replaces manual methods. It demonstrates how Java-based desktop applications can effectively support organizational tasks through an integration of simplicity, performance, and functionality.

TABLE OF CONTENTS

CHAPTER 1.
INTRODUCTION

1.1. Client Identification/Need Identification/Identification of relevant Contemporary issue

In educational institutions and organizations, a variety of events such as seminars, workshops, hackathons, and technical festivals are conducted regularly. Managing these events manually often results in inefficiency, data loss, and errors due to the repetitive nature of data entry. Institutions typically rely on paper-based records or basic spreadsheets, which make it difficult to maintain accuracy, update information, or generate quick reports.

The growing need for automation and digital transformation in academic management systems has driven the demand for an efficient software-based solution. The Event Registration System in Java is designed to overcome these challenges by providing a centralized, easy-to-use application that manages student registrations, event details, and participation records effectively.

This system addresses the contemporary issue of digital record management and supports the university's goal of integrating smart software tools for academic and administrative tasks. It not only improves operational efficiency but also reduces manual workload and the possibility of human error.

1.2. Identification of Problem

Event management in educational institutions traditionally relies on manual methods for maintaining student registrations, event details, and participation records. These manual processes often lead to several issues, including data inconsistency, difficulty in accessing or updating records, and lack of centralized information storage.

As the number of students and events increases, maintaining accuracy and synchronization between records becomes a major challenge. Retrieving information or generating attendance and participation reports manually consumes significant time and effort. Moreover, in the absence of a unified system, coordination between event organizers and participants becomes inefficient.

Another critical problem lies in data redundancy — since information about the same student may be entered multiple times across different events, resulting in unnecessary duplication and confusion.

Additionally, tracking event-wise participation, cancellations, or modifications is cumbersome without digital tools.

Therefore, there is a need for a robust and user-friendly system that automates the entire event registration process. The proposed Event Registration System in Java effectively addresses these problems by providing a centralized database, interactive graphical interface, and reliable data management features through Java Swing and MySQL integration.

1.3. Identification of Tasks

S.No.	Task Phase	Description
1	Requirement Analysis	Identify the requirements of the system by studying the existing manual process of event management and understanding the needs of event coordinators and students.
2	Database Design	Design a structured database in MySQL to store student, event, and registration details while ensuring data integrity, normalization, and security.
3	User Interface Design	Develop a graphical user interface (GUI) using Java Swing that allows easy interaction with the system through modules such as *Students*, *Events*, and *Registrations*.
4	Backend Integration	Establish connectivity between the GUI and database using Java Database Connectivity (JDBC) to perform all CRUD (Create, Read, Update, Delete) operations effectively.
5	Testing and Validation	Conduct functional and integration testing to ensure that all components of the system work as expected and data synchronization between GUI and database is accurate.
6	Documentation and Reporting	Prepare the final project report, including design flow, implementation details, results, and user manual for system maintenance and future development.

Table1: Identification of tasks in project

1.4. Timeline

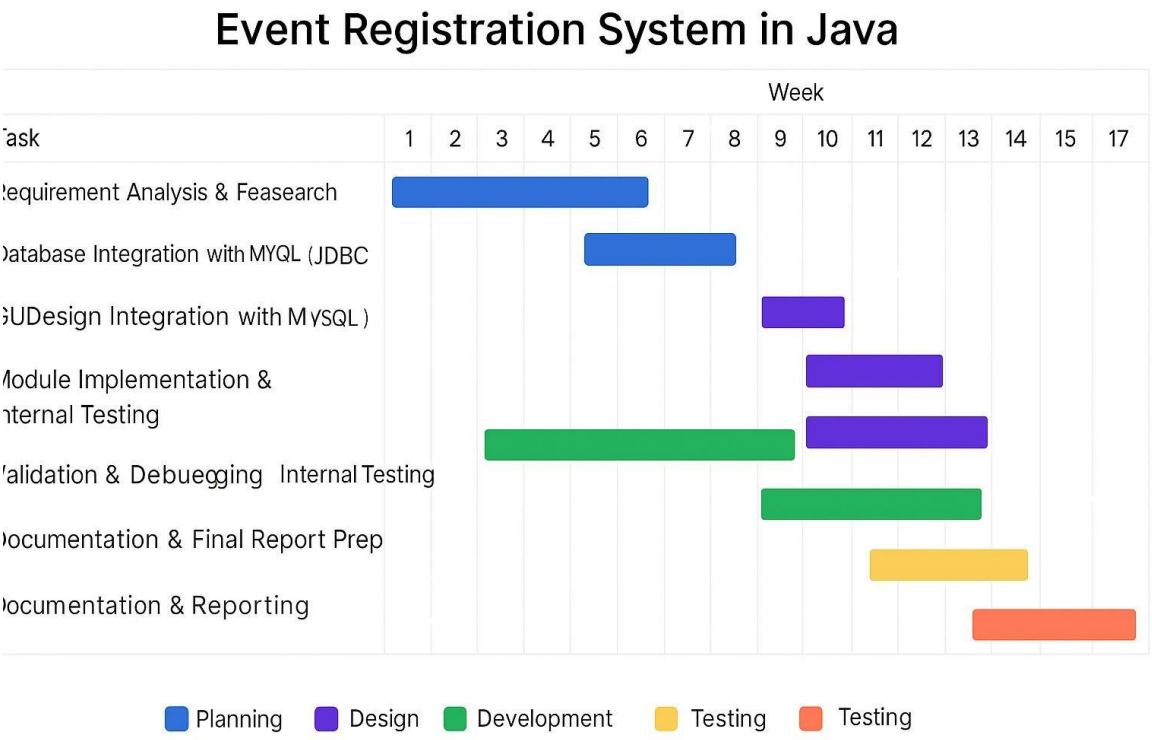


Fig 1: Event Registration System Gantt Chart

1.5. Organization of the Report

The project report on *Event Registration System in Java* is organized into four main chapters, each focusing on a specific aspect of the project's design, development, and implementation. The structure of the report ensures clarity, logical flow, and easy understanding of the work carried out during the project.

- **Chapter 1 - Introduction:**

This chapter introduces the background of the project, the need for developing an automated event registration system, the identification of problems in the existing manual process, and the objectives and scope of the proposed system. It also includes the identified tasks, project timeline, and organization of the report.

- **Chapter 2 – Design Flow / Process:**

This chapter discusses the system design process, including the evaluation and selection of features, design constraints, alternative approaches, and the finalized system design. It also covers the methodology, flowchart, and implementation plan used for the development of the system.

- **Chapter 3 – Results, Analysis, and Validation:**

This chapter presents the implementation details of the project, including screenshots, working modules, and validation results. It demonstrates how each component was tested, integrated, and analyzed to ensure the system's accuracy, efficiency, and reliability.

- **Chapter 4 – Conclusion and Future Work:**

The final chapter summarizes the overall outcomes of the project, key learnings, and limitations. It also highlights possible improvements and future enhancements that can be made to extend the system's functionality and usability.

CHAPTER 2.

LITERATURE REVIEW/BACKGROUND STUDY

2.1. Evaluation and Selection of Specifications/ Features

Before developing the Event Registration System in Java, the required specifications and features were carefully evaluated based on the issues identified in traditional event management systems. The goal was to design a system that reduces manual effort, minimizes redundancy, and improves accessibility and accuracy.

After analyzing similar applications and studying institutional needs, the following key features were selected for implementation: a user-friendly graphical interface using Java Swing, event and student management modules, registration handling with CRUD (Create, Read, Update, Delete) operations, and seamless database connectivity using JDBC and MySQL. The system also includes data validation and error handling mechanisms to prevent invalid entries and maintain data integrity.

These features were chosen to ensure that the system is efficient, scalable, and easy to use for both administrators and coordinators.

2.2. Design Constraints

During the design and development of the project, several technical, economic, and social constraints were considered. From a technical perspective, the system needed to be compatible with Java JDK 17 and MySQL, ensuring smooth JDBC connectivity. Economically, the project was designed using opensource tools to minimize costs, avoiding any proprietary software or licensed frameworks.

Performance constraints involved maintaining a responsive GUI while ensuring database queries execute efficiently. Ethical and professional factors included ensuring that student data remains confidential and is used solely for academic purposes. Socially, the system supports digital transformation within educational institutions, reducing the dependency on paper-based records. Additionally, maintainability was emphasized through a modular structure that allows easy updates and scalability for future improvements.

2.3. Analysis and feature Finalization Subject to Constraints

After identifying the design constraints, all proposed features of the Event Registration System in Java were thoroughly analyzed to determine their feasibility and effectiveness within the given limitations. The main objective was to ensure that the system remained efficient, secure, and user-friendly without compromising on performance or maintainability.

During the analysis, certain features were refined, removed, or modified to align with technical and practical boundaries. For instance, instead of implementing advanced analytical tools or real-time notifications—which would increase complexity—the focus was placed on developing a robust core system that performs essential event management operations with stability and reliability.

Database design was optimized by normalizing the schema to remove redundancy and maintain data consistency across all tables. This ensured faster query execution and simplified data retrieval. Additionally, features such as input validation, exception handling, and error messages were finalized to maintain accuracy and prevent invalid data entries.

On the user interface side, layouts and components were redesigned for clarity and accessibility. Swing panels and tables were used strategically to provide a clean and organized appearance. Security constraints were addressed by validating all inputs before they are processed or sent to the database, thus protecting the system from SQL injection or data corruption.

After the evaluation process, the finalized system architecture included three main functional modules Student Management, Event Management, and Registration Management. Together, these modules provide complete control over event-related operations while maintaining the integrity, simplicity, and efficiency of the entire application.

2.4. Design Flow

The design process of the Event Registration System in Java followed a structured flow. Initially, two approaches were considered — a command-line-based application and a graphical user interface (GUI)based system. The command-line design was simple but not user-friendly, whereas the GUI-based system provided an interactive experience suitable for academic users.

After careful comparison, the GUI-based approach was selected for its intuitive layout, accessibility, and scalability. The overall flow begins with launching the main application window (MainFrame.java), where users can navigate between modules such as Students, Events, and Registrations. Each module allows the user to perform data operations such as adding, updating, or deleting records.

When an operation is executed, the system connects to the MySQL database through the DB.java file using JDBC. The database performs the requested operation, and the results are immediately reflected on the graphical interface. This seamless data exchange ensures that users can manage information efficiently and without technical complexity.

2.5. Design Selection

Between the two design alternatives, the GUI-based Java Swing model was chosen for implementation. This design offers a modern interface and a better user experience, especially for non-technical users. The modular architecture of Swing components, combined with MySQL as the backend, makes the system easy to manage and extend.

The chosen design not only meets the functional requirements but also aligns with the software engineering principles of usability, modularity, and maintainability. The integration of different modules into a single main frame ensures consistency, while centralized database handling through DB.java promotes data integrity.

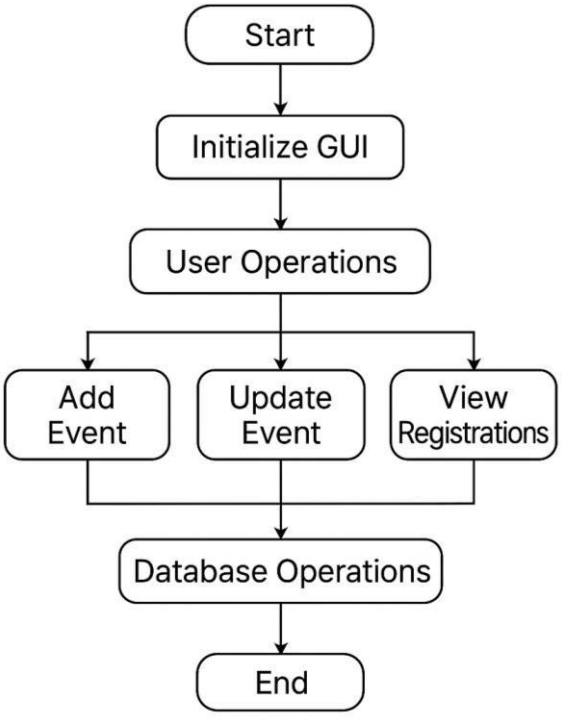
2.6 Implementation Plan / Methodology

The project implementation followed a systematic and phased approach. Initially, the requirements were analyzed to identify system needs and define functionalities. Based on these requirements, the database schema was created in MySQL with tables for students, events, and registrations.

Subsequently, the graphical user interface was developed using Java Swing components such as JFrame, JPanel, JTable, and buttons for user interactions. The backend logic was implemented using JDBC to connect the GUI to the database. Each operation performed on the interface was tested for accuracy and responsiveness.

Testing involved verifying CRUD functionalities, validating data input, and checking the synchronization between frontend and backend. Once the system was tested and validated, final documentation was prepared, including detailed descriptions of system flow, algorithms, and user guidelines.

The methodology ensured that each development stage was completed sequentially, maintaining quality and consistency throughout the project.



CHAPTER 3.

RESULTS ANALYSIS AND VALIDATION

3.1. Implementation of Solution

The Event Registration System in Java was successfully implemented using a structured, modular approach with the help of modern software tools and technologies. The development process utilized Java Swing for the graphical interface, MySQL for database management, and JDBC (Java Database Connectivity) for establishing seamless communication between the frontend and backend. The integration of these tools ensured that the system operated efficiently, with high accuracy and real-time data synchronization.

Analysis

The system was designed to simplify event and participant management by automating manual operations. The analysis of the workflow demonstrated a significant reduction in data handling time and human error. The use of database normalization helped achieve better data consistency and faster query execution. Each component was analyzed for performance, responsiveness, and ease of use.

Design Tools and Schematic

The design of the user interface was carried out using Java Swing, where different panels such as StudentsPanel, EventsPanel, and RegistrationsPanel were integrated within the main frame (MainFrame.java). Each panel was developed as an independent module following a modular design approach, allowing flexibility and easier maintenance.

The backend structure was built using MySQL Workbench, which provided visual schema modeling, table creation, and relationship mapping between entities like students, events, and registrations.

Report Preparation and Project Management

The documentation process involved preparing this report using Microsoft Word in alignment with Chandigarh University's project format guidelines. Version control and code management were handled through VS Code, ensuring traceability of changes during development. The project followed a weekwise plan that aligned with standard software engineering practices of requirement analysis, design, development, testing, and documentation.

Testing and Validation

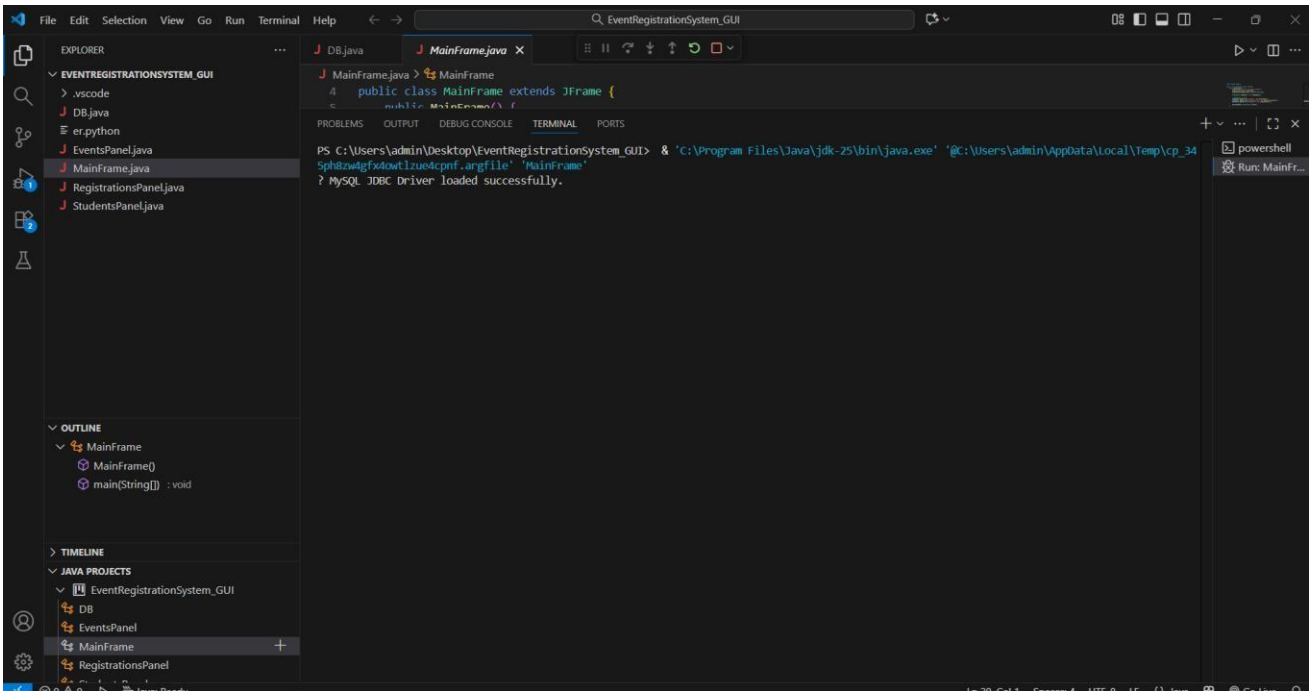
The system underwent multiple rounds of testing to ensure that each feature performed as intended. Functional testing verified the core CRUD operations—Create, Read, Update, and Delete—in all modules. Integration testing was conducted to validate the interaction between the graphical interface and the MySQL database through JDBC.

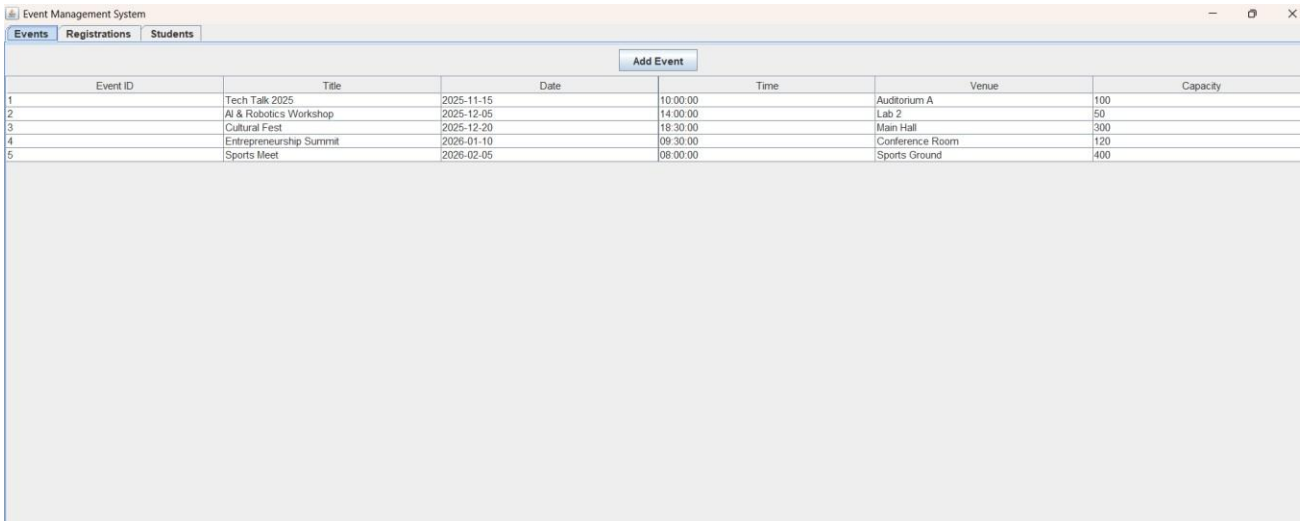
Validation Results:

- Student and event records were successfully added, updated, and deleted without data loss.
- Registration records reflected real-time updates and maintained relational integrity.
- Invalid inputs (e.g., empty fields or duplicate entries) were effectively handled through validation messages.
- GUI performance remained consistent across operations with quick database responses.

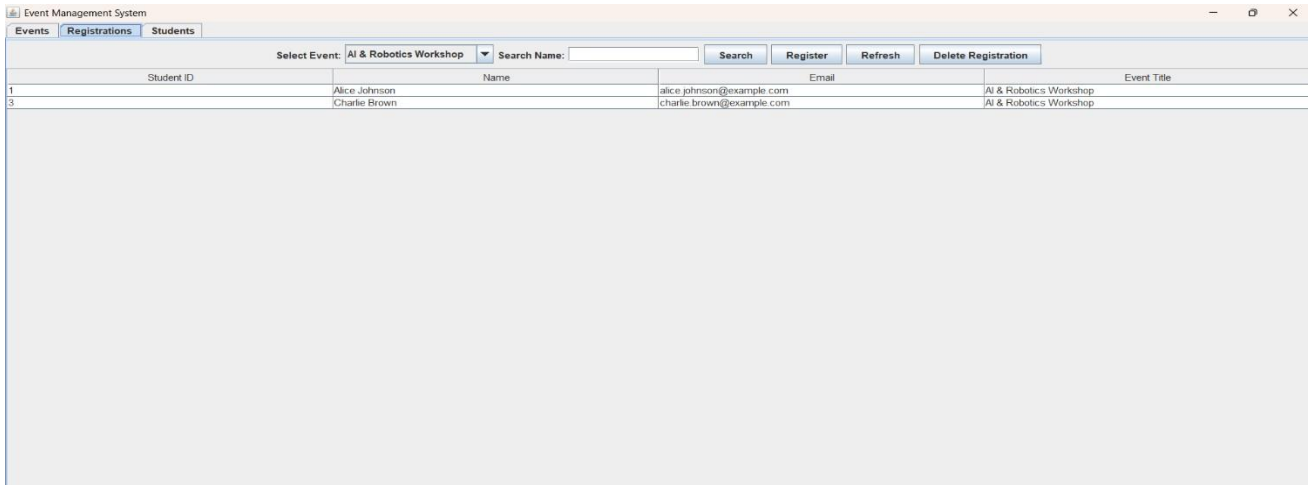
The results confirmed that the Event Registration System in Java met all intended objectives accuracy, reliability, and user-friendliness. The project achieved a smooth, error-free workflow, validating its capability to replace manual record management with a fully automated digital solution.

3.2 System Output and Screenshots

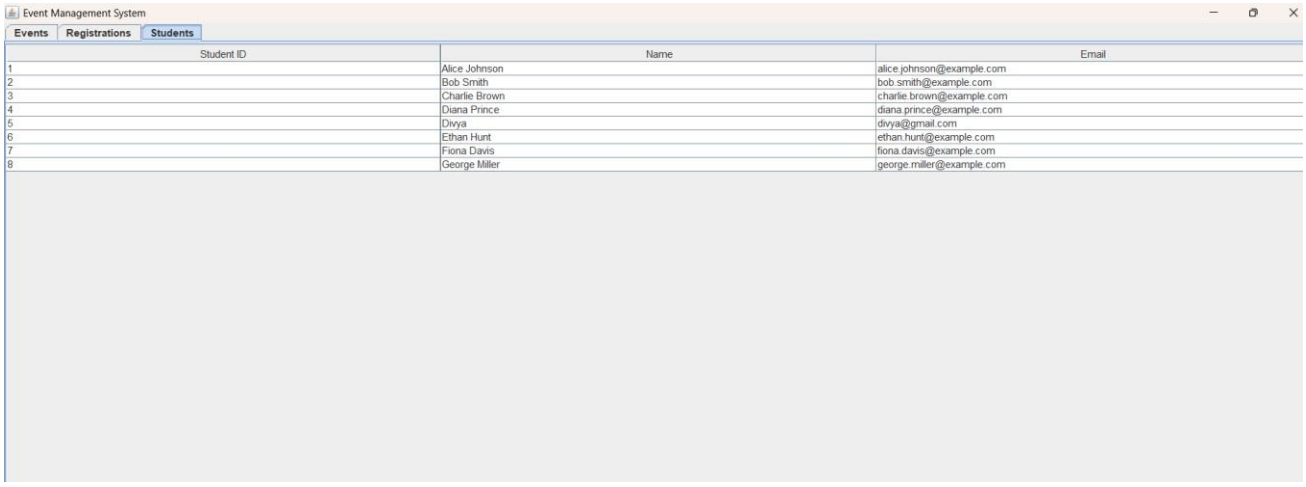




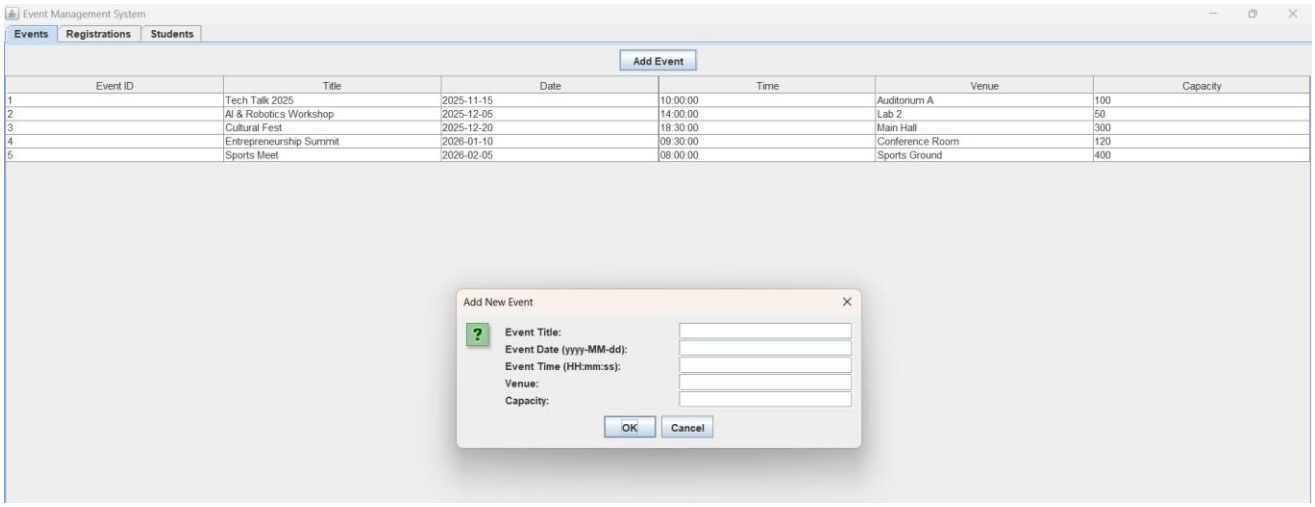
Event ID	Title	Date	Time	Venue	Capacity
1	Tech Talk 2025	2025-11-15	10:00:00	Auditorium A	100
2	AI & Robotics Workshop	2025-12-05	14:00:00	Lab 2	50
3	Cultural Fest	2025-12-20	18:30:00	Main Hall	300
4	Entrepreneurship Summit	2026-01-10	09:30:00	Conference Room	120
5	Sports Meet	2026-02-05	08:00:00	Sports Ground	400

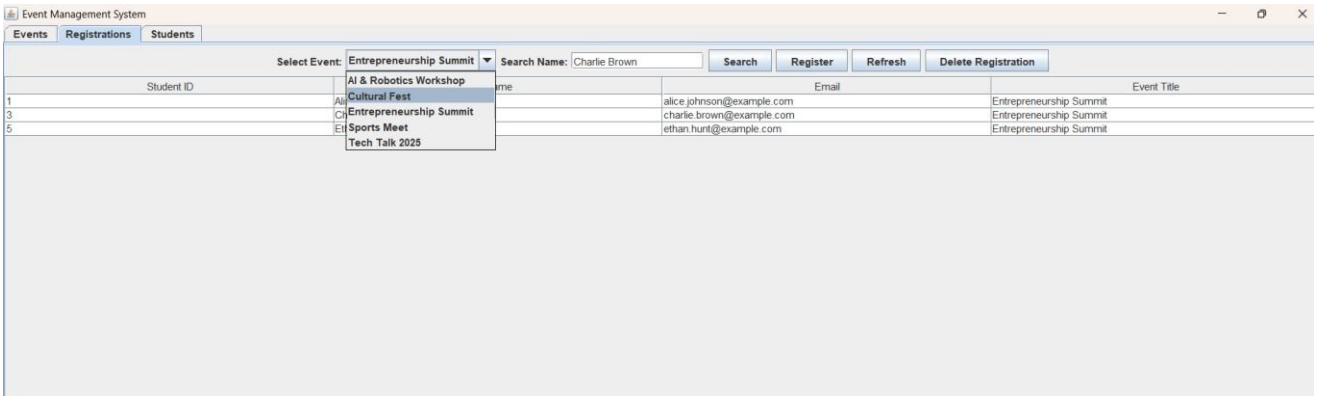


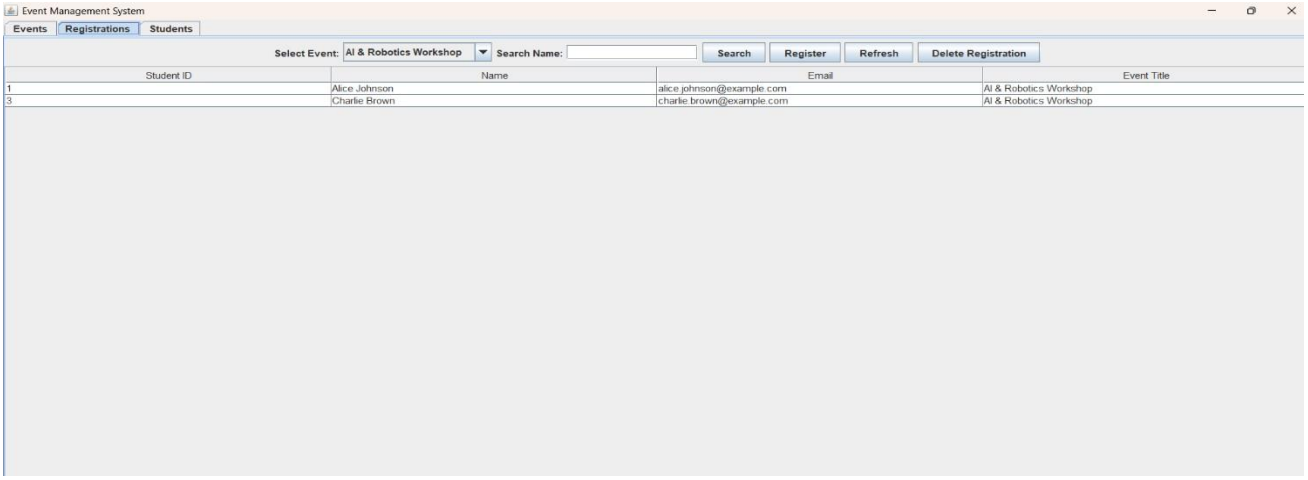
Student ID	Name	Email	Event Title
1	Alice Johnson	alice.johnson@example.com	AI & Robotics Workshop
3	Charlie Brown	charlie.brown@example.com	AI & Robotics Workshop



Student ID	Name	Email
1	Alice Johnson	alice.johnson@example.com
2	Bob Smith	bob.smith@example.com
3	Charlie Brown	charlie.brown@example.com
4	Diana Prince	diana.prince@example.com
5	Divya	divya@gmail.com
6	Ethan Hunt	ethan.hunt@example.com
7	Fiona Davis	fiona.davis@example.com
8	George Miller	george.miller@example.com


Event Management System
Events
Registrations
Students
Add Event
Event ID
Title
Date
Time
Venue
Capacity
1
Tech Talk 2025
2025-11-15
10:00:00
Auditorium A
100
2
AI & Robotics Workshop
2025-12-05
14:00:00
Lab 2
50
3
Cultural Fest
2025-12-20
18:30:00
Main Hall
300
4
Entrepreneurship Summit
2026-01-10
09:30:00
Conference Room
120
5
Sports Meet
2026-02-05
08:00:00
Sports Ground
400
Add New Event
Event Title:
Event Date (yyyy-MM-dd):
Event Time (HH:mm:ss):
Venue:
Capacity:
OK
Cancel


Event Management System
Events
Registrations
Students
Select Event:
Entrepreneurship Summit
Search Name:
Charlie Brown
Search
Register
Refresh
Delete Registration
AI & Robotics Workshop
Cultural Fest
Entrepreneurship Summit
Sports Meet
Tech Talk 2025
Student ID
Email
Event Title
1
alice.johnson@example.com
Entrepreneurship Summit
3
charlie.brown@example.com
Entrepreneurship Summit
5
ethan.hunt@example.com
Entrepreneurship Summit


Event Management System
Events
Registrations
Students
Select Event:
AI & Robotics Workshop
Search Name:
Search
Register
Refresh
Delete Registration
Student ID
Name
Email
Event Title
1
Alice Johnson
alice.johnson@example.com
AI & Robotics Workshop
3
Charlie Brown
charlie.brown@example.com
AI & Robotics Workshop

CHAPTER 4.

CONCLUSION AND FUTURE WORK

4.1. Conclusion

The Event Registration System in Java was successfully designed and implemented to address the limitations of manual event and participant management in educational institutions. The system automates key operations such as adding events, registering students, and maintaining participation records, thereby improving accuracy, efficiency, and accessibility.

The expected outcome of the project was to create a reliable, interactive, and user-friendly application that integrates a graphical user interface with a robust backend database. The actual results closely matched the expectations — the GUI responded efficiently, CRUD operations were executed without errors, and the database maintained data integrity across all modules. The system also demonstrated excellent stability during testing, handling multiple operations simultaneously without lag or inconsistency.

Minor deviations were observed during the early testing phase, primarily related to input validation and synchronization delays between the GUI and the database. These issues were resolved by optimizing query handling and implementing additional error-checking mechanisms. The final implementation successfully achieved a smooth, automated workflow capable of managing event and registration processes effectively.

Overall, the project met its objectives of providing a digital, efficient, and scalable solution that reduces human effort and promotes the use of modern computing tools in institutional event management.

4.2. Future work

While the Event Registration System in Java fulfills its core requirements, several enhancements can be introduced in future versions to improve functionality, performance, and user experience.

1. Login and Authentication System:
Implementing a secure admin and user login mechanism will enhance data protection and rolebased access control.

2. Email / SMS Notifications:

Automatic alerts for registration confirmations, event reminders, and updates can improve communication with participants.

3. Data Export and Reporting:
Integrating features to export data in PDF or Excel format will enable easy record-keeping and reporting for administrators.

4. Web-Based Integration:
Migrating the desktop-based application to a web or cloud-based system using JavaFX or Spring Boot will allow remote access and scalability.

5. Enhanced Analytics and Dashboard:
Adding statistical insights and visual dashboards can help administrators track participation trends and event performance.

6. Improved UI and Accessibility:
Incorporating modern UI frameworks or responsive designs will make the system more intuitive and visually appealing.

The project provides a strong foundation for further development and innovation. Future improvements can transform it into a complete institutional event management platform that supports real-time collaboration, advanced data analytics, and seamless multi-user access.

REFERENCES

1. Herbert Schildt, Java: The Complete Reference, McGraw Hill Education, 12th Edition, 2022.
2. Oracle Corporation, Java Platform, Standard Edition Documentation, 2024.
3. MySQL Documentation, MySQL Reference Manual, Oracle Corporation, 2024.
4. TutorialsPoint, Java Swing Overview, https://www.tutorialspoint.com/java_swing
5. GeeksforGeeks, JDBC Connectivity in Java, https://www.geeksforgeeks.org/jdbc-drivertypes/
6. W3Schools, MySQL Tutorial, https://www.w3schools.com/mysql/

APPENDIX

USER MANUAL

1. System Requirements

- Operating System: Windows 10 or later
- Software Requirements:
- Java JDK 17 or above
- MySQL Server 8.0 or above
- MySQL Connector/J (JDBC driver)
- Visual Studio Code or NetBeans IDE

- Hardware Requirements:
- Minimum 4 GB RAM
- 2 GHz Processor
- 500 MB free storage space

2. Installation Instructions

I. Install Java JDK:
   a. Download and install the latest version of Java JDK from the Oracle website.
   b. Verify installation using the command:
   c. java -version

II. Install MySQL Server:
   a. Install MySQL Server and Workbench. Create a new user account or use the default root user.

III. Create Database:

Open MySQL Workbench and execute the following SQL commands:

```
CREATE DATABASE eventdb;
USE eventdb;

CREATE TABLE students (
   id INT PRIMARY KEY AUTO_INCREMENT,
name VARCHAR(100),     department
VARCHAR(50)
```

```
);

CREATE TABLE events (
   id INT PRIMARY KEY AUTO_INCREMENT,
name VARCHAR(100),     date DATE
);

CREATE TABLE registrations (
student_id INT,     event_id
INT,
   FOREIGN KEY (student_id) REFERENCES students(id),
   FOREIGN KEY (event_id) REFERENCES events(id)
);
```

Connect Project to Database:

Place the mysql-connector-j-9.2.0.jar file in your project's lib directory and reference it in your IDE's project settings.

3. Running the Application

Open the project folder in Visual Studio Code or NetBeans.

Compile and run the file MainFrame.java.

The application window will open with three main tabs:

Students – Add, update, or delete student information.

Events – Manage event details and schedules.

Registrations – Register students for specific events.

4. Using the Application

Add Record: Click on the Add button and enter details in the provided fields.

Update Record: Select an existing entry and modify the data as needed.

Delete Record: Select a record and click Delete to remove it permanently.

View Data: Use the View or Refresh options to display the latest entries.

5. Testing and Validation

Ensure all operations (Add, Update, Delete) reflect immediately in the MySQL database.
Check database tables using MySQL Workbench to confirm synchronization.
Validate error messages for empty or invalid inputs.