ALUMNI MANAGEMENT SYSTEM

A PROJECT REPORT

Submitted by

VISHALINI V (2303811710422180)

in partial fulfillment of requirements for the award of the course

CGB1201 - JAVA PROGRAMMING

In

COMPUTER SCIENCE AND ENGINEERING

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM – 621 112

NOVEMBER- 2024

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY (AUTONOMOUS)

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report on **“ALUMNI MANAGEMENT SYSTEM”** is the bonafide work of **VISHALINI V (2303811710422180)** who carried out the project work during the academic year 2024 - 2025 under my supervision.

CGB1201-JAVA PROGRAMMING Dr.A.DELPHIN CAROLINA RANI, M.E.,Ph.D., HEAD OF THE DEPARTMENT PROFESSOR	CGB1201-JAVA PROGRAMMING Mr.MALARMANNAN A, M.E., SUPERVISOR ASSISTANTP ROFESSOR
SIGNATURE	**SIGNATURE**
Mrs.A.Delphin Carolina Rani, M.E.,Ph.D.,	Mr. A. Malarmannan, M.E.,
HEAD OF THE DEPARTMENT	**SUPERVISOR**
PROFESSOR	ASSISTANT PROFESSOR
Department of CSE	Department of CSE
K.Ramakrishnan College of Technology (Autonomous)	K.Ramakrishnan College of Technology (Autonomous)
Samayapuram–621112.	Samayapuram–621112.

Submitted for the viva-voce examination held on 06-12-2024

CGB1201-JAVA PROGRAMMING Mr. M. SARAVANAN, M.E., INTERNAL EXAMINER ASSISTANT PROFESSOR	CGB1201-JAVA PROGRAMMING Mr.R. KARTHIK, M.E., EXTERNAL EXAMINER ASSISTANT PROFESSOR 8138-SCE, TRICHY. 06/12/2024
INTERNAL EXAMINER	EXTERNAL EXAMINER

DECLARATION

I declare that the project report on **“ALUMNI MANAGEMENT SYSTEM”** is the result of original work done by us and best of our knowledge, similar work has not been submitted to **“ANNA UNIVERSITY CHENNAI”** for the requirement of Degree of **BACHELOR OF ENGINEERING**. This project report is submitted on the partial fulfilment of the requirement of the completion of the course **CGB1201 - JAVA PROGRAMMING.**

.

Signature

V. Vish

VISHALINI V

Place: Samayapuram

Date: 06-12-2024

ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and in-debt to our institution "**K.Ramakrishnan College of Technology (Autonomous)**", for providing us with the opportunity to do this project.

I glad to credit honourable chairman **Dr. K. RAMAKRISHNAN**, **B.E.,** for having provided for the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director **Dr. S. KUPPUSAMY, MBA, Ph.D.,** for forwarding to our project and offering adequate duration in completing our project.

I would like to thank **Dr. N. VASUDEVAN, M.Tech., Ph.D.,** Principal, who gave opportunity to frame the project the full satisfaction.

I whole heartily thanks to **Dr. A. DELPHIN CAROLINA RANI, M.E.,Ph.D.,** Head of the department, **COMPUTER SCIENCE AND ENGINEERING** for providing her encourage pursuing this project.

I express our deep expression and sincere gratitude to our project supervisor **MR. A. MALARMANNAN, M.E.,** Department of **COMPUTER SCIENCE AND ENGINEERING,** for his incalculable suggestions, creativity, assistance and patience which motivated us to carry out this project.

I render our sincere thanks to Course Coordinator and other staff members for providing valuable information during the course.

I wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

VISION OF THE INSTITUTION

To serve the society by offering top-notch technical education on par with global standards

MISSION OF THE INSTITUTION

- Be a center of excellence for technical education in emerging technologies by exceeding the needs of the industry and society.
- Be an institute with world class research facilities
- Be an institute nurturing talent and enhancing the competency of students to transform them as all-round personality respecting moral and ethical values

VISION OF DEPARTMENT

To be a center of eminence in creating competent software professionals with research and innovative skills.

MISSION OF DEPARTMENT

M1: Industry Specific: To nurture students in working with various hardware and software platforms inclined with the best practices of industry.

M2: Research: To prepare students for research-oriented activities.

M3: Society: To empower students with the required skills to solve complex technological problems of society.

PROGRAM EDUCATIONAL OBJECTIVES

1. PEO1: Domain Knowledge

To produce graduates who have strong foundation of knowledge and skills in the field of Computer Science and Engineering.

2. PEO2: Employability Skills and Research

To produce graduates who are employable in industries/public sector/research organizations or work as an entrepreneur.

3. PEO3: Ethics and Values

To develop leadership skills and ethically collaborate with society to tackle real-world challenges.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO 1: Domain Knowledge

To analyze, design and develop computing solutions by applying foundational concepts of Computer Science and Engineering.

PSO 2: Quality Software

To apply software engineering principles and practices for developing quality software for scientific and business applications.

PSO 3: Innovation Ideas

To adapt to emerging Information and Communication Technologies (ICT) to innovate ideas and solutions to existing/novel problems

PROGRAM OUTCOMES (POs)

Engineering students will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences

3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations

4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions

5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations

6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice

7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development

8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

ABSTRACT

The Alumni Management System is a software application developed to strengthen the relationship between educational institutions and their alumni. It serves as a centralized platform to maintain and manage alumni records, facilitate communication, and encourage active alumni participation in institutional activities. By enabling institutions to keep track of their alumni, the system promotes engagement and fosters networking opportunities for both alumni and current students.

The system offers functionalities such as alumni registration, where users can store essential details like name, batch, email, and year of graduation. This ensures institutions have a comprehensive database for future references. Additionally, it features search capabilities to filter alumni based on specific criteria, like graduation year, making it easier to locate individuals or groups. The system also supports event management, enabling institutions to create, store, and display upcoming events such as reunions, seminars, or mentorship programs. These events serve as opportunities for alumni to connect with their peers and contribute to institutional growth.

Designed with an intuitive user interface using Java AWT, the system provides real-time feedback on actions such as registration and event creation, ensuring a seamless user experience. The application uses ArrayLists to manage data efficiently, offering a simple yet powerful solution for alumni tracking and event coordination.

ABSTRACT WITH POs AND PSOs MAPPING

CO 5 : BUILD JAVA APPLICATIONS FOR SOLVING REAL-TIME PROBLEMS.

ABSTRACT	POs MAPPED	PSOs MAPPED
The Alumni Management System is a software application developed to strengthen the relationship between educational institutions and their alumni. It serves as a centralized platform to maintain and manage alumni records, facilitate communication, and encourage active alumni participation in institutional activities. By enabling institutions to keep track of their alumni, the system promotes engagement and fosters networking opportunities for both alumni and current students.	**PO1 -3** **PO2 -3** **PO3 -3** **PO4 -3** **PO5 -3** **PO6 -3** **PO7 -3** **PO8 -3** **PO9 -3** **PO10 -3** **PO11-3** **PO12 -3**	**PSO1 -3** **PSO2 -3** **PSO3 -3**

Note: 1- Low, 2-Medium, 3- High

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Objective

The Alumni Management System is designed to help educational institutions maintain lasting connections with their alumni and foster a thriving community. The primary objective of this system is to create a centralized platform that facilitates communication, networking, and event management between alumni and their alma mater. By providing an intuitive interface for alumni registration, the system will store essential data such as contact information, year of graduation, and current professional status. This data will allow the institution to manage and engage with alumni effectively, ensuring they are kept informed about important events, institutional developments, and networking opportunities.

Furthermore, the system seeks to simplify administrative tasks related to alumni management, ensuring a user-friendly interface for administrators and alumni alike. It strives to serve as a bridge for alumni to stay connected with their alma mater, contribute to institutional growth, and leverage the alumni network for career development, mentorship, and knowledge sharing. Ultimately, the system is a step towards creating a vibrant, engaged, and supportive alumni ecosystem.

1.2 Overview

The Alumni Management System is a platform designed to help educational institutions stay connected with their alumni. It allows alumni to register, update their profiles, and stay informed about events, job opportunities, and other institutional activities. The system enables alumni to network with each other, attend events, and contribute to the institution through mentoring or offering career guidance. It also helps institutions manage alumni data, organize reunions and events, and track alumni engagement. The goal of the system is to strengthen the relationship between alumni and their alma mater, fostering long-term support and involvement in the institution's growth.

This system aims to bridge the gap between alumni and institutions by fostering communication, networking, and engagement. Through its functionalities, it simplifies administrative tasks such as maintaining alumni records, organizing events, and tracking alumni participation. The platform also offers tools for searching alumni based on graduation years, providing a structured way to locate.

1.3 Java Programming Concepts

1. Object-Oriented Programming (OOP)

- **Encapsulation:** Keeps alumni data safe and only allows it to be changed in specific ways.
- **Abstraction:** Hides the complicated parts of the system and only shows what the user needs to see.
- **Inheritance:** Allows different parts of the system to share common features (like name or contact info).
- **Inheritance:** Allows different parts of the system to share common features (like name or contact info).

2. Classes and Objects

- Class: A blueprint for creating things (like a list of alumni or events).
- Object: An actual instance of a class. For example, an actual alumnus or an event.

3. Variables and Data Types

- Variables store data (like name, email, or graduation year) and each piece of data has a specific type, like text (String) or number (int).

4. Methods and Functions

- Methods are actions that the system can perform, like registering an alumnus or creating an event.

5. Control Structures

- The if-else, switch-case, and looping constructs like for and while are used to control the flow of the program. For instance, in the event creation process, the system might use an if-else statement to check whether an event already exists before creating a new one. Loops can be used to display lists of alumni or events to the user.

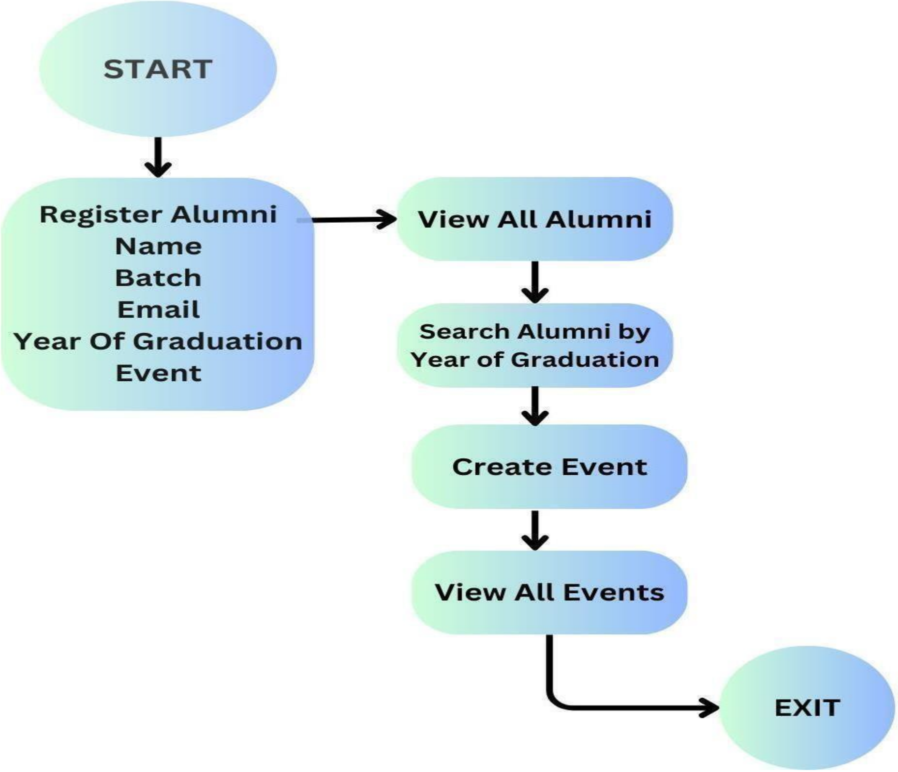
CHAPTER 2

PROJECT METHODOLOGY

2.1 Proposed Work

The Alumni Management System aims to create a platform that connects alumni with their educational institution, facilitating communication, networking, and event management. The proposed work includes designing a user-friendly interface using Java's AWT package, where both alumni and administrators can easily interact with the system. Key features will include alumni registration, the ability to update personal details, and search functionalities based on criteria like year of graduation. The system will also allow administrators to create and manage events, track participant details, and send notifications to alumni about upcoming events or institutional updates.

2.2 Block Diagram



CHAPTER 3

MODULE DESCRIPTION

1.1 User Interface Module:

The User Interface (UI) Module is a critical component of the Alumni Management System, as it provides the means through which users interact with the system. It serves as the bridge between the system’s functionality and the user, allowing for seamless interaction with the underlying processes. The goal of the UI module is to create an intuitive, user-friendly, and responsive interface that ensures an efficient user experience for administrators and other system users.

1.2 Alumni Management Module:

The Alumni Management Module is the core component of the Alumni Management System. Its primary purpose is to manage and store essential information about alumni, facilitate communication, and provide functionalities for managing alumni records. This module allows educational institutions to easily register, update, search, and view alumni details, ensuring a comprehensive and organized database of past students.

1.3 Search Module:

The Search Module in the Alumni Management System is a crucial component that allows users to efficiently search and retrieve specific alumni or event information from the database. This module simplifies the process of finding relevant alumni details or events based on various search criteria, ensuring that administrators can manage records more efficiently.

1.4 Event Management Module:

The Event Management Module in the Alumni Management System is a key component that allows the system to manage, organize, and track alumni events effectively.

1.5 Data Display Module:

The Data Display Module is an integral component of the Alumni Management System that facilitates the efficient visualization and presentation of stored data. This module is responsible for displaying various types of information related to alumni and events, ensuring that users can easily access and review the data they need.

CHAPTER 4

CONCLSION & FUTURE SCOPE

2. 1 CONCLUSION

The **Alumni Management System** successfully fulfils its purpose of creating and maintaining strong, ongoing connections between educational institutions and their alumni. By integrating features like alumni registration, event management, and data visualization, the system helps institutions engage alumni in meaningful ways, facilitating communication, networking, and event participation.

The alumni registration feature, in particular, is a core functionality that ensures the institution keeps accurate and updated records of its alumni, including personal information, batches, and years of graduation. This data not only helps in organizing alumni events but also plays a vital role in fostering long-term relationships between alumni and the institution. While the system works well for smaller datasets, it is scalable, with future enhancements such as database integration and improved security measures suggested to accommodate growing user needs. Feedback from users indicates that the system is easy to use, with intuitive interfaces for both alumni and administrators.

4.2 FUTURE SCOPE

The **Alumni Management System** has the potential to grow into a more comprehensive and dynamic platform to cater to the evolving needs of educational institutions and their alumni. A significant improvement could be the integration of a robust database system like MySQL or PostgreSQL to ensure secure, persistent, and scalable data management. Moving the system to a web-based platform using technologies such as Java Servlets, JSP, or Spring Boot would make it accessible from anywhere. Additionally, a mobile application for Android and iOS would enhance the system’s reach and convenience for users.

The platform could also include a payment gateway for event registrations, alumni association fees, and donations, making financial transactions seamless. Introducing an alumni achievements section to highlight milestones and awards would foster pride and motivation within the alumni network.

APPENDIX A
(SOURCE CODE)

```
import java.awt.*; import java.awt.event.*;
import java.util.ArrayList;

public class AlumniManagementSystem extends Frame {

    // Components
    Label titleLabel, nameLabel, batchLabel, emailLabel, yearLabel, eventLabel, messageLabel;
    TextField nameField, batchField, emailField, yearField, eventField;
    Button registerAlumniButton, viewAlumniButton, searchAlumniButton, createEventButton, viewEventsButton, exitButton;
    TextArea outputArea;

    // Data Storage
    ArrayList<String> alumniList = new ArrayList<>(); ArrayList<String> eventList = new ArrayList<>();

    public AlumniManagementSystem() {
        // Frame settings
        setTitle("Alumni Management System"); setSize(600, 600);
        setLayout(null); setBackground(Color.LIGHT_GRAY);

        // Title
        titleLabel = new Label("Alumni Management System");
        titleLabel.setBounds(200, 30, 300, 30); titleLabel.setFont(new Font("Arial", Font.BOLD, 18)); add(titleLabel);

        // Labels and Input Fields
```

```
nameLabel = new Label("Name:"); nameLabel.setBounds(50, 80, 100, 20); add(nameLabel);

nameField = new TextField(); nameField.setBounds(150, 80, 200, 20); add(nameField);

batchLabel = new Label("Batch:"); batchLabel.setBounds(50, 110, 100, 20); add(batchLabel);

batchField = new TextField(); batchField.setBounds(150, 110, 200, 20); add(batchField);

emailLabel = new Label("Email:"); emailLabel.setBounds(50, 140, 100, 20); add(emailLabel);

emailField = new TextField(); emailField.setBounds(150, 140, 200, 20); add(emailField);

yearLabel = new Label("Year of Graduation:");
yearLabel.setBounds(50, 170, 150, 20);
add(yearLabel);

yearField = new TextField(); yearField.setBounds(200, 170, 150, 20); add(yearField);

eventLabel = new Label("Event:");
eventLabel.setBounds(50, 200, 100, 20);
add(eventLabel);
```

```
eventField = new TextField(); eventField.setBounds(150, 200, 200, 20); add(eventField);

// Buttons
registerAlumniButton = new Button("Register Alumni");
registerAlumniButton.setBounds(50, 240, 150, 30);
add(registerAlumniButton);

viewAlumniButton = new Button("View All Alumni");
viewAlumniButton.setBounds(210, 240, 150, 30);
add(viewAlumniButton);

searchAlumniButton = new Button("Search Alumni by Year");
searchAlumniButton.setBounds(370, 240, 170, 30);
add(searchAlumniButton);

createEventButton = new Button("Create Event"); createEventButton.setBounds(50, 280, 150, 30); add(createEventButton);

viewEventsButton = new Button("View All Events");
viewEventsButton.setBounds(210, 280, 150, 30);
add(viewEventsButton);

exitButton = new Button("Exit"); exitButton.setBounds(370, 280, 150, 30); add(exitButton);

// Output Area
outputArea = new TextArea(); outputArea.setBounds(50, 330, 500, 200); outputArea.setEditable(false); add(outputArea);
```

```
        // Message Label messageLabel
        = new Label("");
        messageLabel.setBounds(50, 540, 500, 20); add(messageLabel);

        // Event Listeners
        registerAlumniButton.addActionListener(e -> registerAlumni()); viewAlumniButton.addActionListener(e -> viewAllAlumni()); searchAlumniButton.addActionListener(e ->
        searchAlumniByYear()); createEventButton.addActionListener(e -> createEvent());
        viewEventsButton.addActionListener(e -> viewAllEvents()); exitButton.addActionListener(e -> System.exit(0));

        // Window Listener to Close addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) { dispose();
            }
        });
    }

    // Register Alumni
    private void registerAlumni() {
        String name = nameField.getText();
        String batch = batchField.getText();
        String email = emailField.getText();
        String year = yearField.getText();

        if (!name.isEmpty() && !batch.isEmpty() && !email.isEmpty() && !year.isEmpty()) {
            alumniList.add("Name: " + name + ", Batch: " + batch + ", Email: " + email + ", Year: " +
year);
            messageLabel.setText("Alumni registered successfully!"); clearFields();
        } else {
```

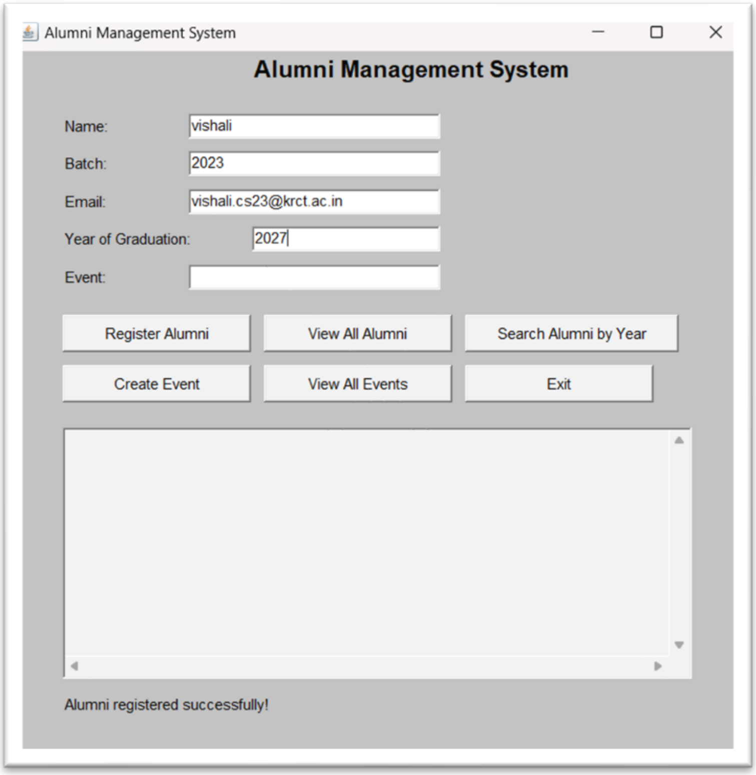
```
        messageLabel.setText("All fields are required for alumni registration!");
    }
}

// View All Alumni
private void viewAllAlumni() { outputArea.setText("Registered Alumni:\n"); for
    (String alumni : alumniList) {
        outputArea.append(alumni + "\n");
    }
}

// Search Alumni by Year of Graduation private void searchAlumniByYear() {
    String year = yearField.getText();
    outputArea.setText("Alumni Graduated in Year " + year + ":\n");
    for (String alumni : alumniList) {
        if (alumni.contains("Year: " + year)) { outputArea.append(alumni + "\n");
        }
    }
}

// Create Event
private void createEvent() {
    String event = eventField.getText();
    if (!event.isEmpty()) {
        eventList.add(event);
        messageLabel.setText("Event created successfully!");
        eventField.setText("");
    } else {
        messageLabel.setText("Event field cannot be empty!");
    }
}
```

```
    // View All Events
    private void viewAllEvents() { outputA-
        rea.setText("Upcoming Events:\n"); for
        (String event : eventList) {
            outputArea.append(event + "\n");
        }
    }

    // Clear Input Fields private
    void clearFields() {
        nameField.setText("");
        batchField.setText("");
        emailField.setText("");
        yearField.setText("");
    }

    // Main Method
    public static void main(String[] args) {
        new AlumniManagementSystem().setVisible(true);
    }
}
```

APPENDIX B

(SCREENSHOTS)

2.1 Register Alumni



Name: vishali

Batch: 2023

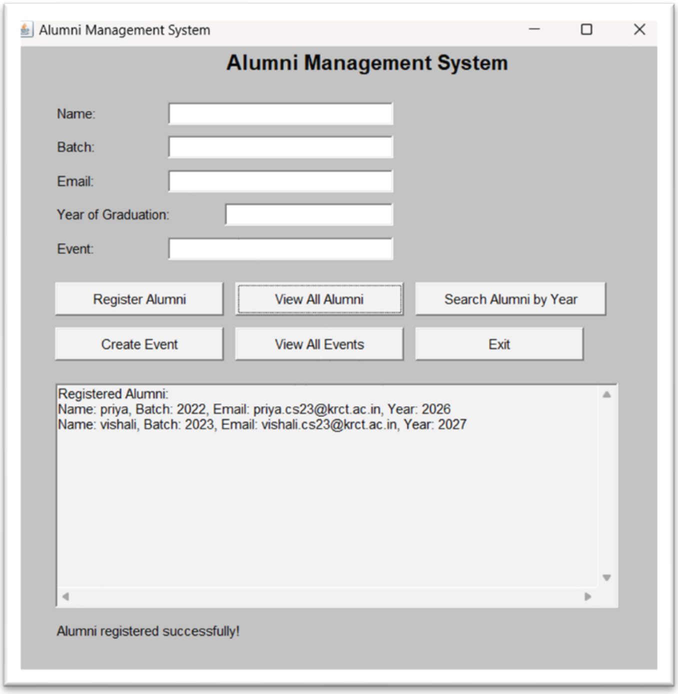
Email: vishali.cs23@krct.ac.in

Year of Graduation: 2027

Event:

Alumni registered successfully!

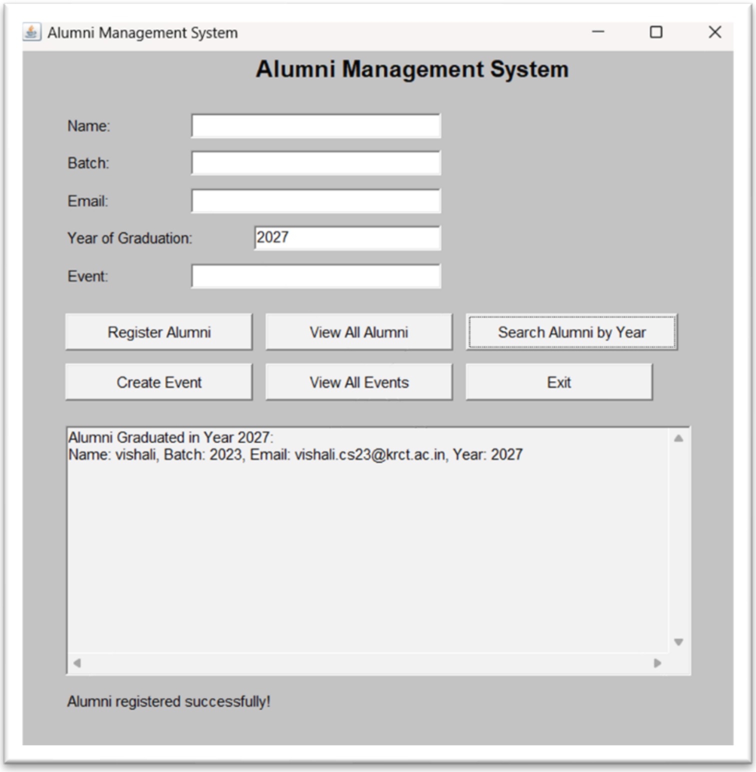
2.2 View All Alumni



```
Registered Alumni:
Name: priya, Batch: 2022, Email: priya.cs23@krct.ac.in, Year: 2026
Name: vishali, Batch: 2023, Email: vishali.cs23@krct.ac.in, Year: 2027
```

Alumni registered successfully!

2.3 Search Alumni by Year



Name:

Batch:

Email:

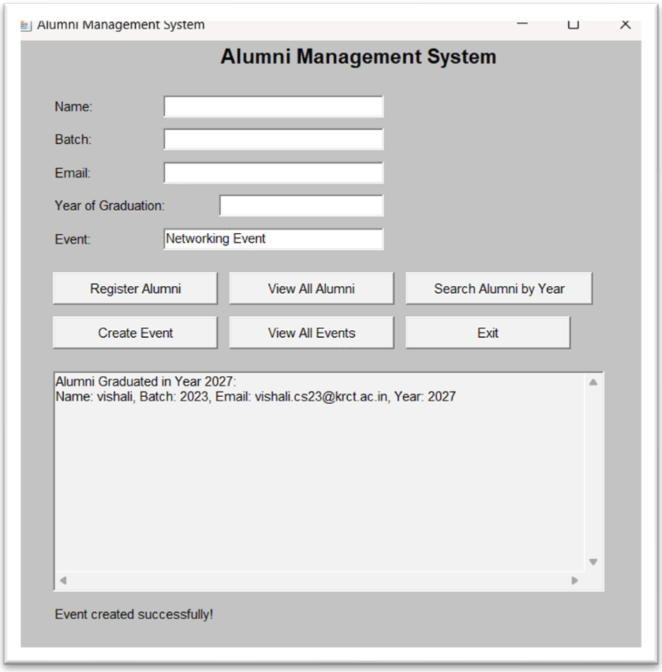
Year of Graduation: 2027

Event:

```
Alumni Graduated in Year 2027:
Name: vishali, Batch: 2023, Email: vishali.cs23@krct.ac.in, Year: 2027
```

Alumni registered successfully!

2.4 Create Event



Name:

Batch:

Email:

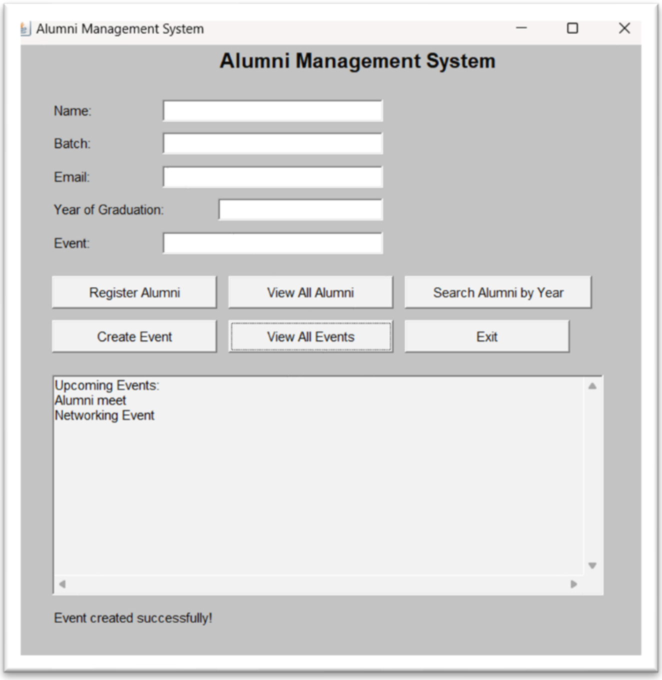
Year of Graduation:

Event: Networking Event

```
Alumni Graduated in Year 2027:
Name: vishali, Batch: 2023, Email: vishali.cs23@krct.ac.in, Year: 2027
```

Event created successfully!

2.5 View All Events



```
Upcoming Events:
Alumni meet
Networking Event
```

Event created successfully!

REFERENCES

1. “Java: The Complete Reference" by Herbert Schildt (2021 Edition) Provides detailed information on Java programming, including GUI development with AWT and event handling.

2. "Java: The Complete Reference" (2021, McGraw-Hill Education) – A comprehensive guide covering all Java topics, including GUI and event handling.

3. "Core Java Volume I – Fundamentals" (2018, Pearson Education) – Focuses on Java basics and building simple applications.

4. "Java Programming for Beginners" (2017, Learntoprogram.tv) – A beginner’s guide to learning Java programming and building simple applications.

5. Oracle Java Documentation https://docs.oracle.com/javase/ Official Java documentation for understanding AWT, event handling, and GUI components.