ALUMNI MANAGEMENT SYSTEM

A PROJECT REPORT

Submitted by

NANDHINI K (2303811724322075)

in partial fulfillment of requirements for the award of the course

CGB1201 – JAVA PROGRAMMING

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

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

SAMAYAPURAM – 621 112

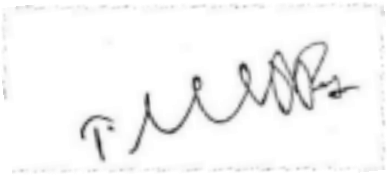
DECEMBER, 2024

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 **NANDHINI K (2303811724322075)** who carried out the project work during the academic year 2024 - 2025 under my supervision.



Signature

Dr. T. AVUDAIAPPAN M.E.,Ph.D.,

HEAD OF THE DEPARTMENT,

Department of Artificial Intelligence,

K. Ramakrishnan College of Technology,

Samayapuram, Trichy -621 112.

Signature

Mrs. S. GEETHA M.E.,

SUPERVISOR,

Department of Artificial Intelligence,

K. Ramakrishnan College of Technology,

Samayapuram, Trichy -621 112.

SSubmitted for the viva-voce examination held on 3.12.24

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

I declare that the project report on "**ALUMNI MANAGEMENT SYSTEM** " is the result of original work done by me and best of my knowledge, similar work has not been submitted to "**ANNA UNIVERSITY CHENNAI**" for the requirement of Degree of **BACHELOR OF TECHNOLOGY**. This project report is submitted on the partial fulfillment of the requirement of the award of the **CGB1201 – JAVA PROGRAMMING.**

NANDHINI K

Place: Samayapuram

Date: 3/12/2024

ACKNOWLEDGEMENT

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

I extend our sincere acknowledgement and appreciation to the esteemed and honourable Chairman, **Dr. K. RAMAKRISHNAN, B.E.,** for having provided 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 our project and offering an adequate duration to complete it.

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

I thank **Dr.T.AVUDAIAPPAN, M.E.,Ph.D**., Head of the Department of **ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**, for providing her encouragement in pursuing this project.

I wish to convey our profound and heartfelt gratitude to our esteemed project guide **Mrs.S. GEETHA M.E.,** Department of **ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**, for her incalculable suggestions, creativity, assistance and patience which motivated us to carry out this project

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

VISION AND MISSION OF THE DEPARTMENT

To excel in education, innovation and research in Artificial Intelligence and Data Science to fulfill industrial demands and societal expectations.

Mission 1: To educate future engineers with solid fundamentals, continually improving teaching methods using modern tools.

Mission 2: To collaborate with industry and offer top-notch facilities in a conductive learning environment.

Mission 3: To foster skilled engineers and ethical innovation in AI and Data Science for global recognition and impactful research.

Mission 4: To tackle the societal challenge of producing capable professionals by instilling employability skills and human values.

PROGRAM EDUCATIONAL OBJECTIVES (PEOS)

PEO 1: Compete on a global scale for a professional career in Artificial Intelligence and Data Science.

PEO 2: Provide industry-specific solutions for the society with effective communication and ethics.

PEO 3: Hone their professional skills through research and lifelong learning initiatives.

PROGRAM OUTCOMES

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.

PROGRAM SPECIFIC OUTCOMES (PSOs)

- **PSO 1:** Capable of working on data-related methodologies and providing industry-focussed solutions.
- **PSO2:** Capable of analysing and providing a solution to a given real-world problem by designing an effective program.

ABSTRACT

The Alumni Management System is a comprehensive platform developed to enhance the relationship between educational institutions and their alumni, fostering continuous engagement and collaboration. This system serves as a centralized hub for facilitating effective communication, networking, and event organization. It empowers alumni to stay connected with their alma mater while providing avenues for career growth, mentorship opportunities, and professional networking. The system incorporates features such as personalized profiles, event notifications, discussion forums, job boards, and feedback mechanisms, ensuring a seamless and interactive experience for users. It not only strengthens the alumni network but also creates opportunities for knowledge sharing and collaboration, benefiting current students through mentorship and career guidance. By supporting career development initiatives and encouraging active alumni participation, the platform contributes to institutional development. Its goal is to foster a lifelong connection, enabling alumni to contribute to the growth and success of the institution while benefiting from the network for their personal and professional development. This system ultimately transforms the alumni-institution relationship into a mutually beneficial partnership.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The Alumni Management System is a sophisticated platform designed to streamline the relationship between educational institutions and their alumni through a structured and efficient approach. Built on a layered architecture, it ensures seamless data management and smooth event organization by dividing functionalities into distinct layers. The system provides an intuitive user interface for collecting alumni and event details, while its controller efficiently validates inputs and manages their flow to the service layer, which handles core business logic. With a robust data layer ensuring secure and scalable information storage, this system empowers institutions to foster meaningful alumni engagement and organize events effectively, enhancing connectivity and collaboration for long-term benefits.

1.2 OBJECTIVE

The objective of the Alumni Management System is to strengthen the bond between educational institutions and their alumni by providing a centralized platform for continuous engagement and collaboration. It aims to enhance communication, foster professional networking, and support career development through features such as personalized profiles, job boards, and mentorship programs. The system seeks to encourage active alumni participation in institutional initiatives, facilitate knowledge sharing through forums and discussions, and organize events to promote collaboration.

CHAPTER 2
PROJECT METHODOLOGY

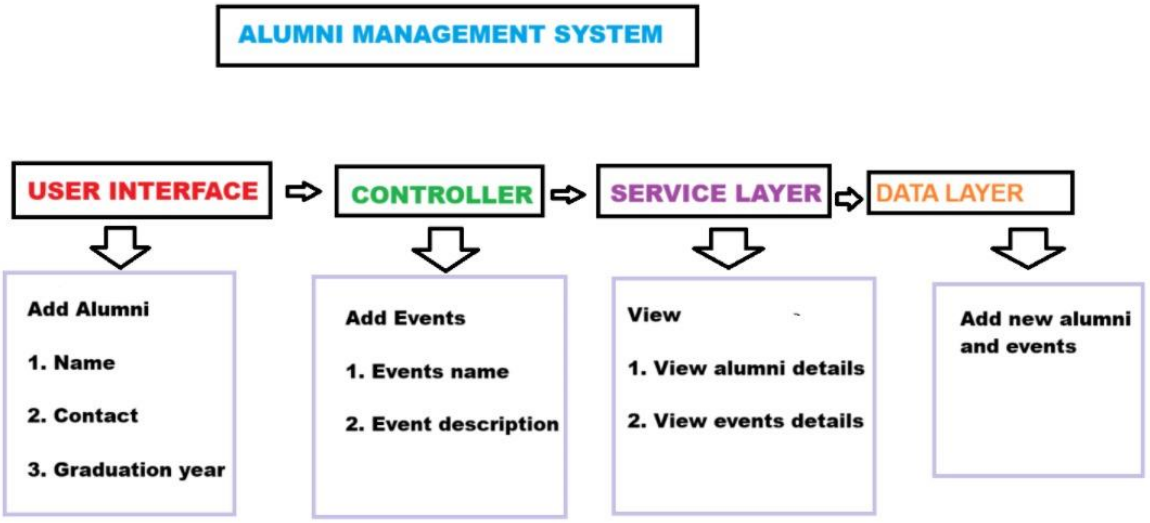
2.1 PROPOSED WORK

The proposed Alumni Management System is built on a robust and layered architecture designed to ensure efficient data handling, event management, and scalability. The architecture comprises multiple tiers that work in unison to streamline operations and enhance user experience. At the forefront is the User Interface (UI), which serves as the primary point of interaction for users, allowing them to input alumni details, register for events, and access system features in a user-friendly manner. Behind the scenes, the Controller Layer plays a pivotal role in managing the flow of information by validating user inputs, ensuring data integrity, and directing the processed requests to the appropriate modules.

The Service Layer forms the core of the system, encapsulating the business logic required for essential operations such as adding, updating, retrieving, and managing alumni records and event details. This layer ensures that the platform functions seamlessly, catering to the diverse needs of alumni and administrators alike. The Data Layer, on the other hand, is responsible for securely storing and managing information, utilizing robust in-memory structures or scalable databases to handle large volumes of data efficiently.

This layered approach not only ensures smooth interaction and effective data management but also facilitates adaptability and scalability, making the system capable of handling the growing needs of alumni engagement and event organization. By integrating these components cohesively, the Alumni Management System delivers a comprehensive and reliable solution for fostering strong alumni relationships and streamlining institutional activities.

2.2 BLOCK DIAGRAM



CHAPTER 3
JAVA PROGRAMMING CONCEPTS

JAVA PROGRAMMING CONCEPTS:

This project utilizes a range of Java programming concepts, including:

3.1 OBJECT-ORIENTED PROGRAMMING (OOP):

1. The system was built using OOP principles, ensuring modularity, reusability, and maintainability. Classes and objects represent various components like alumni, events, and communication features, promoting clear organization and easy updates.

3.2 CORE JAVA FEATURES:

1. Exception Handling: To manage errors like invalid user input or database connection failures.
2. File Handling: To read and write reports or logs when required.

Input/Output Streams:

1. To manage data transfer between the program and external sources, such as files or user input. Input streams allow the system to read data, such as alumni records or event details, from files, while output streams are used to write updated information back to those files.

CHAPTER 4

MODULE DESCRIPTION

4.1 ALUMNI MODULE

- The module is responsible for managing individual alumni records, including storing their name, contact details, graduation year, and other relevant information. It enables creating, viewing, updating, and deleting alumni data.

4.2 EVENT MODULE

- The module focuses on managing event-related data, such as event names, descriptions, dates, and locations. It allows for creating new events, updating event details, and retrieving event information for display or notification purposes.

4.3 ALUMNI MANAGEMENT MODULE

- The module connects alumni to specific events or initiatives. It ensures effective communication by managing invitations, tracking alumni participation, and maintaining a record of their involvement in events or institutional activities.

4.4 EVENT MANAGEMENT MODULE

- The module oversees the organization of events, including scheduling, managing attendee lists, sending notifications, and tracking the success of events. It streamlines event planning and ensures seamless execution.

4.5 USER INTERFACE MODULE

- This module provides the front-end interface for users to interact with the system. It includes menus, input fields, and display options for alumni and event data. The interface ensures ease of navigation and smooth user interaction with the system

CHAPTER 5
CONCLUSION

In conclusion, the Alumni Management System plays a crucial role in building and maintaining a strong connection between educational institutions and their alumni. By implementing efficient modules such as alumni profile management, event organization, and invitation handling, the system provides a centralized platform for communication and engagement. It enables alumni to stay informed about institutional events, access career resources, and participate in professional networking opportunities. Additionally, the system fosters a sense of community, encouraging alumni to give back through mentorship or contributions. Overall, the system not only streamlines administrative tasks but also empowers the institution to leverage alumni involvement for growth and development, ultimately strengthening the long-term relationship between the institution and its alumni.

REFERENCES:

Books:

- ***"Java: A Beginner's Guide"*** **by Herbert Schildt** – This book covers Java basics and advanced topics, including AWT/Swing. It's ideal for understanding how to design interactive applications like your Alumni Management System.

Websites:

- **GeeksforGeeks - Java AWT & Swing** – Offers detailed tutorials and code examples for working with AWT and Swing to create interactive user interfaces.

APPENDICES

APPENDIX A – SOURCE CODE

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.util.ArrayList;
import java.util.List;

class Alumni {
    private int id;
    private String name;
    private String email;
    private String year;
    private String occupation;

    public Alumni(int id, String name, String email, String year, String occupation)
{
        this.id = id;
        this.name = name;
        this.email = email;
        this.year = year;
        this.occupation = occupation;
    }

    public int getId() {
        return id;
    }
```

```
    public String getName() {
        return name;
    }

    public String getEmail() {
        return email;
    }

    public String getYear() {
        return year;
    }

    public String getOccupation() {
        return occupation;
    }

    @Override
    public String toString() {
        return "ID: " + id + ", Name: " + name + ", Email: " + email +
                ", Year: " + year + ", Occupation: " + occupation;
    }
}

class Event {
    private int id;
    private String name;
    private String date;
    private String description;
```

```
private List<Alumni> invitedAlumni;

public Event(int id, String name, String date, String description) {
    this.id = id;
    this.name = name;
    this.date = date;
    this.description = description;
    this.invitedAlumni = new ArrayList<>();
}

public int getId() {
    return id;
}

public String getName() {
    return name;
}

public String getDate() {
    return date;
}

public String getDescription() {
    return description;
}

public void inviteAlumni(Alumni alumni) {
    invitedAlumni.add(alumni);
}
```

```
    public List<Alumni> getInvitedAlumni() {
        return invitedAlumni;
    }

    @Override
    public String toString() {
        return "ID: " + id + ", Name: " + name + ", Date: " + date +
                ", Description: " + description;
    }
}

public class AlumniManagementSystem extends JFrame {
    private List<Alumni> alumniList;
    private List<Event> eventList;
    private int nextAlumniId;
    private int nextEventId;

    private JTextArea displayArea;

    public AlumniManagementSystem() {
        alumniList = new ArrayList<>();
        eventList = new ArrayList<>();
        nextAlumniId = 1;
        nextEventId = 1;

        setupUI();
    }
```

```
private void setupUI() {
    setTitle("Alumni Management System");
    setSize(800, 600);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setLayout(new BorderLayout());

    displayArea = new JTextArea();
    displayArea.setEditable(false);
    JScrollPane scrollPane = new JScrollPane(displayArea);

    JPanel buttonPanel = new JPanel();
    buttonPanel.setLayout(new GridLayout(3, 2, 10, 10));

    JButton addAlumniButton = new JButton("Add Alumni");
    addAlumniButton.addActionListener(e -> addAlumni());

    JButton viewAlumniButton = new JButton("View Alumni");
    viewAlumniButton.addActionListener(e -> viewAlumni());

    JButton addEventButton = new JButton("Add Event");
    addEventButton.addActionListener(e -> addEvent());

    JButton viewEventsButton = new JButton("View Events");
    viewEventsButton.addActionListener(e -> viewEvents());

    JButton inviteAlumniButton = new JButton("Invite Alumni to Event");
    inviteAlumniButton.addActionListener(e -> inviteAlumniToEvent());

    JButton viewInvitesButton = new JButton("View Event Invites");
```

```
        viewInvitesButton.addActionListener(e -> viewEventInvites());

        buttonPanel.add(addAlumniButton);
        buttonPanel.add(viewAlumniButton);
        buttonPanel.add(addEventButton);
        buttonPanel.add(viewEventsButton);
        buttonPanel.add(inviteAlumniButton);
        buttonPanel.add(viewInvitesButton);

        add(scrollPane, BorderLayout.CENTER);
        add(buttonPanel, BorderLayout.SOUTH);
    }

    private void addAlumni() {
        String name = JOptionPane.showInputDialog(this, "Enter Alumni Name:");
        String email = JOptionPane.showInputDialog(this, "Enter Alumni Email:");
        String year = JOptionPane.showInputDialog(this, "Enter Graduation Year:");
        String occupation = JOptionPane.showInputDialog(this, "Enter
Occupation:");

        if (name != null && email != null && year != null && occupation != null) {
            Alumni alumni = new Alumni(nextAlumniId++, name, email, year,
occupation);
            alumniList.add(alumni);
            displayMessage("Alumni added successfully!");
        } else {
            displayMessage("Action canceled.");
        }
    }
```

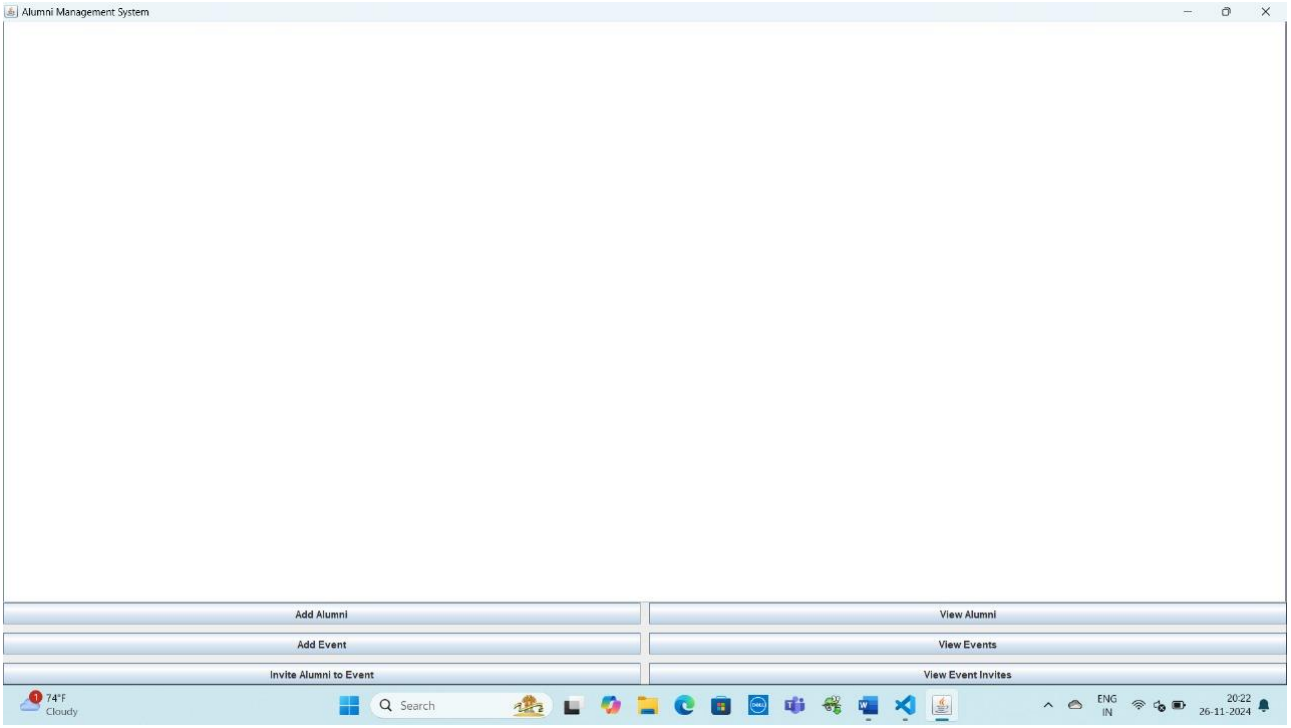
```
    private void viewAlumni() {
        if (alumniList.isEmpty()) {
            displayMessage("No alumni records available.");
            return;
        }

        StringBuilder sb = new StringBuilder("Alumni List:\n");
        for (Alumni alumni : alumniList) {
            sb.append(alumni).append("\n");
        }
        displayMessage(sb.toString());
    }

    private void addEvent() {
        String name = JOptionPane.showInputDialog(this, "Enter Event Name:");
        String date = JOptionPane.showInputDialog(this, "Enter Event Date
(DD/MM/YYYY):");
        String description = JOptionPane.showInputDialog(this, "Enter Event
Description:");

        if (name != null && date != null && description != null) {
            Event event = new Event(nextEventId++, name, date, description);
            eventList.add(event);
            displayMessage("Event added successfully!");
        } else {
            displayMessage("Action canceled.");
        }
    }
```

```
    private void viewEvents() {
        if (eventList.isEmpty()) {
            displayMessage("No events available.");
            return;
        }

        StringBuilder sb = new StringBuilder("Event List:\n");
        for (Event event : eventList) {
            sb.append(event).append("\n");
        }
        displayMessage(sb.toString());
    }

    private void inviteAlumniToEvent() {
        int eventId = Integer.parseInt(JOptionPane.showInputDialog(this, "Enter
Event ID:"));
        int alumniId = Integer.parseInt(JOptionPane.showInputDialog(this, "Enter
Alumni ID:"));

        Event event = null;
        Alumni alumni = null;

        for (Event e : eventList) {
            if (e.getId() == eventId) {
                event = e;
                break;
            }
        }
```

```
        for (Alumni a : alumniList) {
            if (a.getId() == alumniId) {
                alumni = a;
                break;
            }
        }

        if (event != null && alumni != null) {
            event.inviteAlumni(alumni);
            displayMessage("Alumni " + alumni.getName() + " invited to event " +
event.getName() + " successfully!");
        } else {
            displayMessage("Invalid Event ID or Alumni ID.");
        }
    }

    private void viewEventInvites() {
        int eventId = Integer.parseInt(JOptionPane.showInputDialog(this, "Enter
Event ID to View Invites:"));

        for (Event event : eventList) {
            if (event.getId() == eventId) {
                StringBuilder sb = new StringBuilder("Invited Alumni for Event: " +
event.getName() + "\n");
                for (Alumni alumni : event.getInvitedAlumni()) {
                    sb.append(alumni).append("\n");
                }
                displayMessage(sb.toString());
```

```
                return;
            }
        }
        displayMessage("Event with ID " + eventId + " not found.");
    }

    private void displayMessage(String message) {
        displayArea.setText(message);
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> {
            AlumniManagementSystem system = new AlumniManagementSystem();
            system.setVisible(true);
        });
    }
}
```

APPENDIX B - SCREENSHOTS



Alumni List:
ID: 1, Name: K.NANDHINI, Email: nandhini@gmail.com, Year: 2012, Occupation: Working at zoho
ID: 2, Name: A.REXCIA, Email: rexcia@gmail.com, Year: 2016, Occupation: Working at Wipro
ID: 3, Name: R.M. MANSA SHREE, Email: manasa@gmsail.com, Year: 2014, Occupation: Working at Cognizant

Alumni Management System

Event List:
ID: 1, Name: CODING BATTLE, Date: 20/12/2024, Description: Problem Solving
ID: 2, Name: SOLO DANCE, Date: 20/12/2024, Description: Only one must be perform a dance

Alumni Management System

Invited Alumni for Event: Coding Battle
ID: 1, Name: K.NANDHINI, Email: nandhini@gmail.com, Year: 2012, Occupation: Working at zoho