

EVENT SCHEDULING SYSTEM



A PROJECT REPORT

Submitted by

HARINI R (2303811710422054)

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 “**EVENT SCHEDULING SYSTEM**” is the bonafide work of **HARINI R (2303811710422054)** 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

SIGNATURE

Dr.A.Delphin Carolina Rani, M.E., Ph.D.,

HEAD OF THE DEPARTMENT

PROFESSOR

Department of CSE

K.Ramakrishnan College of Technology
(Autonomous)

Samayapuram-621112.

CGB1201-JAVA PROGRAMMING
Mr. M. SARAVANAN, M.E.,
SUPERVISOR
ASSISTANT PROFESSOR

SIGNATURE

Mr. M. Saravanan, M.E.,

SUPERVISOR

ASSISTANT PROFESSOR

Department of CSE

K.Ramakrishnan College of Technology
(Autonomous)

Samayapuram-621112.

Submitted for the viva-voce examination held on 02.12.2024

CGB1201-JAVA PROGRAMMING
Mr. M. ARMANAN A, M.E.,
INTERNAL EXAMINER
ASSISTANT PROFESSOR

INTERNAL EXAMINER

CGB1201-JAVA PROGRAMMING
Dr. R. SETHAMILSELVI, M.E., Ph.D.,
EXTERNAL EXAMINER
PROFESSOR
8138-SCE, TRICHY.

EXTERNAL EXAMINER

DECLARATION

I declare that the project report on “**EVENT SCHEDULING 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**.



HARINI R

Place: Samayapuram

Date:02.12.2024

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. M. SARAVANAN, 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 **Event Scheduling System** is a robust and user-friendly application designed to simplify the organization and management of events for individuals and viewing, editing, and searching events based on details like name, date, and time. It enhances productivity by supporting features like event categorization, recurring event options, and reminders to ensure timely notifications. The system's modular design ensures scalability, allowing for future integration of advanced capabilities such as multi-user support, cloud synchronization, and third-party calendar integration. Its intuitive interface promotes ease of use, catering to users of varying technical expertise. Data validation and secure storage mechanisms ensure accuracy and reliability. By automating repetitive scheduling tasks, the system reduces manual effort and minimizes errors. It serves as an efficient tool for individuals managing personal schedules and organizations coordinating team events. With potential for further expansion, the system is a practical solution for improving event planning and productivity.

ABSTRACT WITH POs AND PSOs MAPPING

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

ABSTRACT	POs MAPPED	PSOs MAPPED
Efficient scheduling and management of events are critical in today's fast-paced environment, where organizations and individuals need tools to streamline planning, coordination, and communication. The Event Scheduling System is a comprehensive platform designed to facilitate the organization and execution of events ranging from meetings and conferences to social gatherings and project timelines. This system leverages advanced algorithms to optimize scheduling by accommodating constraints such as participant availability, venue resources, and time zones.	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 NO.	TITLE	PAGE NO.
	ABSTRACT	viii
1	INTRODUCTION	1
	1.1 Objective	1
	1.2 Overview	1
	1.3 Java Programming concepts	2
2	PROJECT METHODOLOGY	3
	2.1 Proposed Work	3
	2.2 Block Diagram	3
3	MODULE DESCRIPTION	4
	3.1Event management module	4
	3.2Add event module	4
	3.3Search event module	4
	3.4Exit module	4
4	CONCLUSION & FUTURE SCOPE	5
	4.1 Conclusion	5
	4.2 Future Scope	5
	REFERENCES	11
	APPENDIX A (SOURCE CODE)	6
	APPENDIX B (SCREENSHOTS)	10

CHAPTER 1

INTRODUCTION

1.1 Objective

The objective of the Event Scheduling System is to streamline the planning and scheduling of events, ensuring efficiency and accuracy. The system aims to minimize scheduling conflicts by considering participant availability, venue constraints, and other relevant factors. It provides an intuitive, user-friendly interface for easy event creation and management, while enabling real-time updates and notifications to keep all stakeholders informed. By seamlessly integrating with calendars, email systems, and third-party tools, the system enhances usability and accessibility. Additionally, it supports multi-time zone scheduling, automating repetitive tasks to save time and reduce errors. The system also offers analytics and reporting features, providing valuable insights for future event planning. By facilitating collaboration among participants and offering a reliable, scalable solution, it meets the diverse needs of businesses, educational institutions, and individuals alike.

1.2 Overview

The Event Scheduling System is a comprehensive solution designed to simplify and optimize the process of planning, managing, and coordinating events. Whether for corporate meetings, conferences, social gatherings, or personal events, the system aims to eliminate the complexities of manual scheduling by automating key tasks. It allows users to create, update, and manage events efficiently, considering factors such as participant availability, venue resources, and time zone differences. The system features a user-friendly interface that makes event creation accessible to all users, while offering real-time updates, reminders, and notifications to keep participants informed. Seamless integration with popular calendar applications and third-party tools ensures that the scheduling process is streamlined across different platforms.

Additionally, the system provides analytical tools for monitoring event performance,

offering insights that help optimize future events.

1.3 Java Programming Concepts

Basic concepts of OOPS:

- ✓ **Class and Object:** A class is a blueprint, and an object is an instance of the class.
- ✓ **Encapsulation:** Bundles data and methods into a single unit (class) while restricting direct access to data.
- ✓ **Inheritance:** Enables a class (child) to inherit properties and methods from another class (parent), promoting code reuse.
- ✓ **Polymorphism:** Allows methods to perform differently based on the object context (e.g., method overloading and overriding).
- ✓ **Abstraction:** Hides implementation details and exposes only essential features, simplifying system design.

Project related Concepts

1.Event Management:

The system allows users to create events by specifying their name, date, and time. It stores this information for later viewing and searching.

2.Data Structures:

Array List: The Array List <Event> is used to store events dynamically. It allows for easy addition and retrieval of events.

3. User Interaction:

The system operates via a simple console interface, offering a menu-driven approach for users to interact with the system. The menu provides options to add, view, search events, and exit the program.

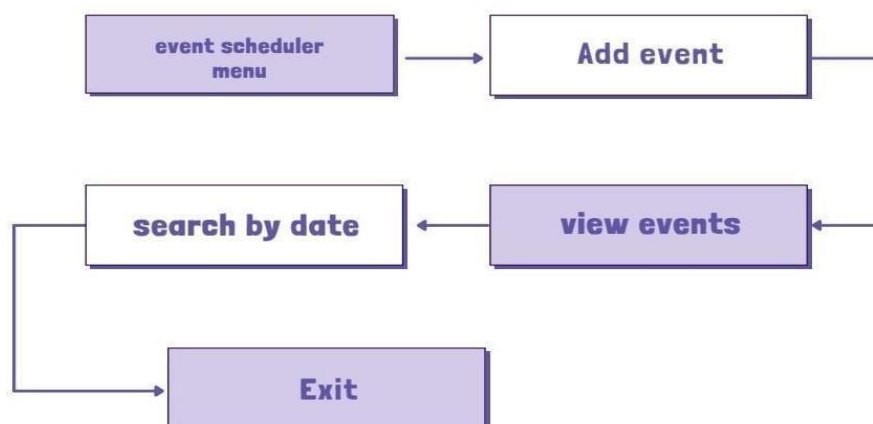
CHAPTER 2

PROJECT METHODOLOGY

2.1 Proposed Work

The proposed work for the Event Scheduling System focuses on enhancing its functionality, user experience, and scalability. The first goal is to improve event management by introducing features like event categories, recurring events, and reminders, allowing users to organize and automate their schedules more effectively. User authentication and authorization will be added to provide personalized access, with role-based permissions and secure password storage to protect user data. Additionally, the system will transition from in-memory data storage to a database for persistent event storage, ensuring data durability. Multi-user support will be introduced with shared calendars, event invitations, and RSVP tracking, enabling collaboration. The system will also include reporting and analytics features, allowing users to track their event history and export data in various formats. Integration with external tools like Google Calendar and weather services will further extend its capabilities.

2.2 Block Diagram



CHAPTER 3

MODULE DESCRIPTION

3.1 Event Management Module:

The Event Management Module is the core component of the system, enabling users to schedule, view, edit, and delete events. It supports features like recurring events and event categorization to improve organization and accessibility

3.2 Add Event Module:

The Add Event Module allows users to input event details such as the name, date, and time, with built-in validation to ensure data accuracy before storing it in the system.

3.3 Search Event Module:

The Search Events Module provides users with the ability to search and filter events based on specific criteria such as date, name, or category, making it easier to find relevant events.

3.4 Exit Module:

Exit Module ensures a smooth and secure shutdown of the application, prompting the user for confirmation and ensuring all data is saved before the system closes. These modules collectively offer a comprehensive and user-friendly event scheduling system.

CHAPTER4

CONCLUSION & FUTURE SCOPE

4.1 CONCLUSION

The Event Scheduling System provides a comprehensive solution for organizing and managing events efficiently. By incorporating key modules like event management, event addition, search functionalities, and a secure exit process, the system offers users a streamlined and user-friendly interface. The flexibility to add, view, edit, and search events ensures that users can easily manage their schedules. The integration of data validation and persistence ensures that the system remains reliable and accurate. Additionally, the system is designed to accommodate future enhancements, such as recurring events and notifications. With its modular structure, the system can easily be expanded to include new features based on user needs. Overall, it offers an effective tool for both individuals and organizations to handle event scheduling in a systematic and organized manner.

4.2 FUTURE SCOPE

The future scope of the Event Scheduling System includes integrating advanced features like automated reminders and email notifications, enhancing user engagement. Adding mobile app support and cloud synchronization would enable users to access their events across different devices seamlessly. The system can be expanded to support collaborative event management, allowing multiple users to share and manage events in real time. Integrating with external platforms like Google Calendar or Outlook would provide greater flexibility. Additionally, implementing machine learning could help suggest optimal event times based on user preferences and history. With these advancements, the system can evolve into a more powerful tool for personal and professional event management.

APPENDIX A

(SOURCE CODE)

```
import
javax.swing.*;
import
java.awt.*;
import
java.awt.event.ActionEvent
t;          import
java.awt.event.ActionListe
ner;        import
java.util.ArrayList;

class Event {
    private    String
    eventName;
    private    String
    eventDate; private
    String eventTime;

    public Event(String eventName, String eventDate, String
    eventTime) { this.eventName = eventName;
    this.eventDate =
    eventDate;
    this.eventTime =
    eventTime;
}
```

```
public String  
    getEventName() {  
        return eventName;  
    }
```

```
public String  
    getEventDate() { return  
        eventDate;  
    }
```

```
public String  
    getEventTime() {  
        return eventTime;  
    }
```

```
@Override  
public String toString() {  
    return "Event Name: " + eventName + ", Date: " + eventDate + ", Time: " +  
        eventTime;  
    }  
}
```

```
public class EventSchedulerSwing {
```

```
private static ArrayList<Event> eventList = new ArrayList<>();
```

```
public static void main(String[] args) {  
    // Create the main frame  
    JFrame frame = new JFrame("Event  
        Scheduler");  
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOS  
E); frame.setSize(400, 300);  
    frame.setLayout(new GridLayout(4, 1));  
  
    // Create buttons for the main menu  
    JButton addEventButton = new  
    JButton("Add Event"); JButton  
    viewEventsButton = new JButton("View  
    Events");  
    JButton searchEventButton = new  
    JButton("Search Event by Date"); JButton  
    exitButton = new JButton("Exit");  
  
    // Add buttons to the  
    frame  
    frame.add(addEventBut  
ton);  
    frame.add(viewEventsB  
utton);  
    frame.add(searchEvent  
Button);  
    frame.add(exitButton);  
  
    // Add event listeners for each button  
    addEventButton.addActionListener(e ->
```

```

showAddEventDialog(frame));
viewEventsButton.addActionListener(e ->
showViewEventsDialog(frame));
searchEventButton.addActionListener(e ->
showSearchEventDialog(frame));
exitButton.addActionListener(e -> System.exit(0));

// Display the frame
frame.setVisible(true);
}

private static void showAddEventDialog(JFrame frame) {
    // Create a panel for the input fields
    JPanel panel = new JPanel(new
    GridLayout(4, 2));    JTextField
    nameField = new JTextField();
    JTextField    dateField    = new
    JTextField(); JTextField timeField =
    new JTextField();

    // Add input fields to the
    panel panel.add(new
    JLabel("Event Name:"));

```

```

panel.add(nameField);
panel.add(new JLabel("Event Date (YYYY-MM-
DD:")); panel.add(dateField);
panel.add(new JLabel("Event Time
(HH:MM:")); panel.add(timeField);

// Show the dialog and get user input
int result = JOptionPane.showConfirmDialog(frame,
panel, "Add Event",
JOptionPane.OK_CANCEL_OPTION);
if (result ==
JOptionPane.OK_OPTION)
{
String name =
nameField.getText(); String
date = dateField.getText();
String time =
timeField.getText();

if (!name.isEmpty() && !date.isEmpty() &&
!time.isEmpty()) {
eventList.add(new
Event(name, date,
time));

JOptionPane.showMessageDialog(frame, "Event
added successfully!");
} else {
JOptionPane.showMessageDialog(frame, "All fields are
required.", "Error",
JOptionPane.ERROR_MESSAGE);
}
}
}

```

```

private static void
    showViewEventsDialog(JFrame frame) {
    if (eventList.isEmpty()) {
        JOptionPane.showMessageDialog(frame, "No events scheduled.");
    } else {
        StringBuilder events = new
        StringBuilder("Scheduled Events:\n"); for (Event
        event : eventList) {
            events.append(event).append("\n");
        }
        JOptionPane.showMessageDialog(frame,  events.toString());
    }
}

private static void showSearchEventDialog(JFrame frame) {
    String date = JOptionPane.showInputDialog(frame, "Enter date to
    search (YYYY-MM-DD:"); if (date != null && !date.isEmpty()) {
        StringBuilder results = new StringBuilder("Events on " + date + ":\n");
    }
}

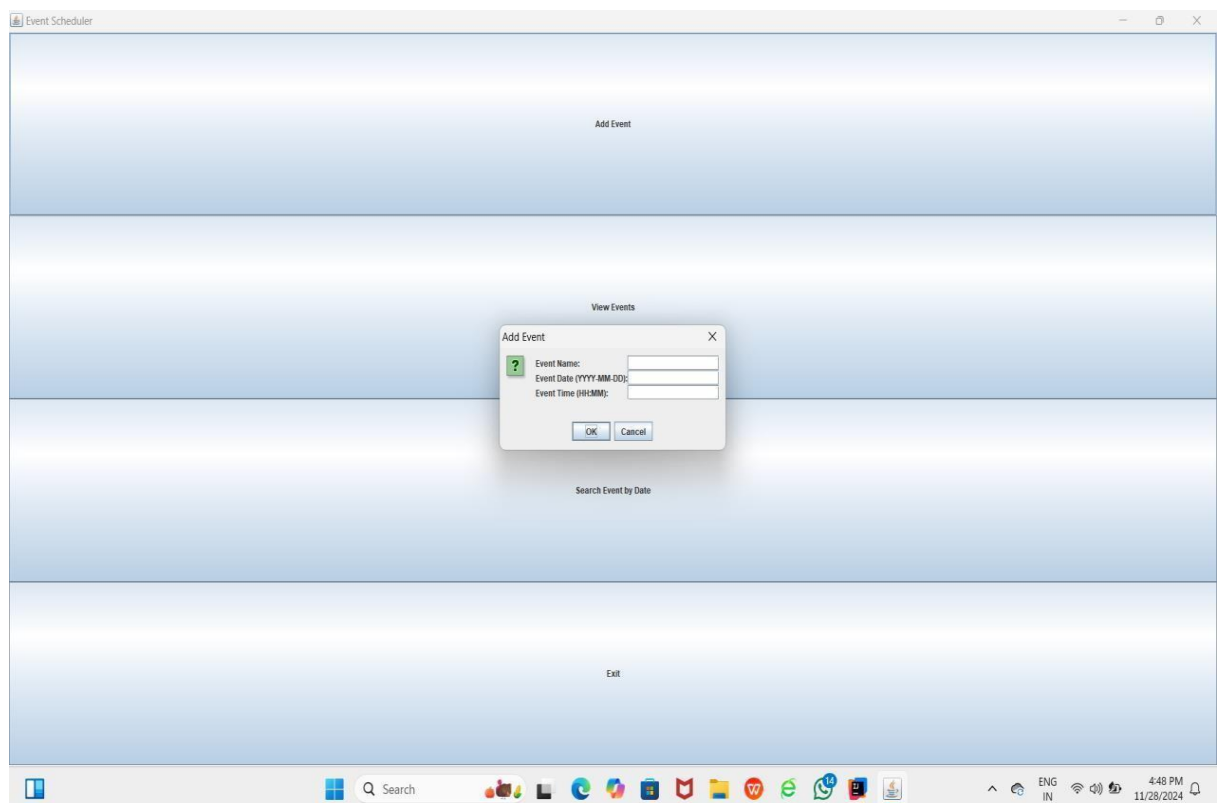
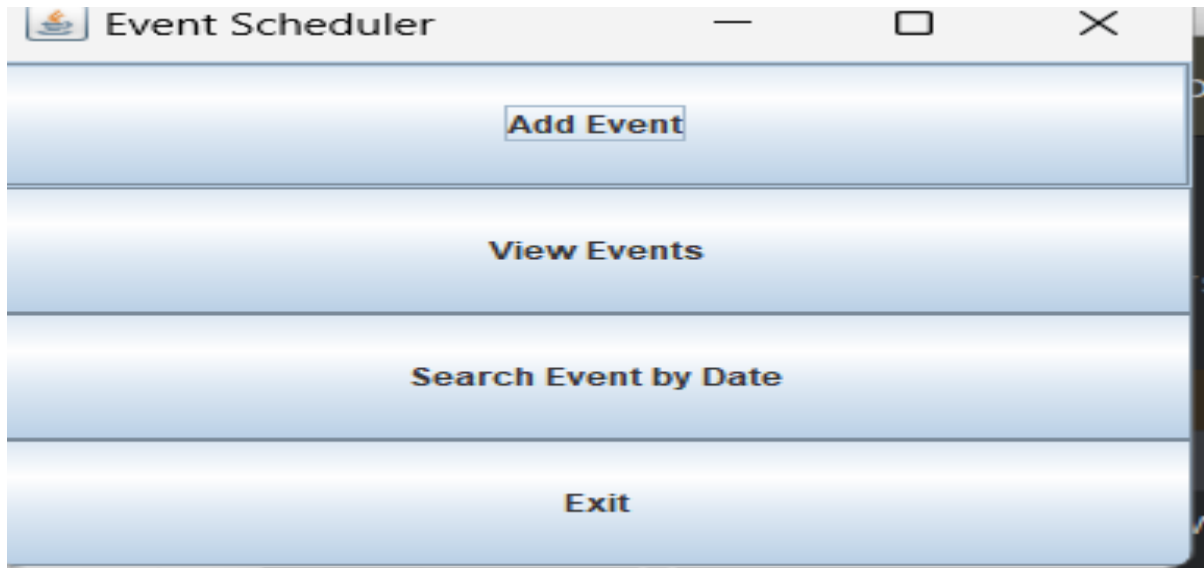
```

```

boolean found = false;
for (Event event : eventList) {
    if
        (event.getEventDate().equals(date)) {
            results.append(event).append
                ("\n"); found = true;
        }
    }
    if (found) {
        JOptionPane.showMessageDialog(frame, results.toString());
    } else {
        JOptionPane.showMessageDialog(frame, "No events found on this date.");
    }
}
}
}

```

APPENDIX B (SCREENSHOTS)



REFERENCES

JAVA BOOKS:

1. "Head First Java" by Kathy Sierra and Bert Bates

- This book is a great resource for beginners and intermediate Java developers. It provides a clear introduction to Java programming concepts, object-oriented principles, and real-world examples.

2. "Java: The Complete Reference" by Herbert Schildt

- A comprehensive guide to Java, covering everything from basic syntax to advanced topics. It is a helpful reference for understanding Java libraries and their usage for building applications like the Event Scheduling System.

WEBSITES:

1. Oracle Java Documentation

- [Oracle Java Documentation](#)
- The official documentation for Java, including all standard libraries, frameworks, and APIs. Essential for understanding Java classes and methods needed for your event scheduling system.

2. GeeksforGeeks - Java Programming Language

- GeeksforGeeks Java
- A popular site with tutorials, problem-solving articles, and code examples. It's especially helpful for learning Java fundamentals and advanced concepts like collections and event handling.

LINKS:

- Stack Overflow: <https://stackoverflow.com/questions/tagged/java>
A valuable platform for seeking help from a large community of Java developers when you encounter specific challenges or need advice on best practices.
- GitHub: <https://github.com/>
You can find open-source Java projects related to event scheduling and gain insights into how other developers have structured their systems. You can also upload and share your project here.