



**TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
THAPATHALI CAMPUS**

**C-Programming Project Report
On
Event Management System**

Submitted By:

Mandip Chhetri (THA081BEI018)
Kiran Paudel (THA081BEI013)
Purushottam Neupane (THA081BEI033)

Submitted To:

Department of Electronics and Computer Engineering
Thapathali Campus
Kathmandu, Nepal

March, 2025



**TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
THAPATHALI CAMPUS**

**C-Programming Project Report
On
Event Management System**

Submitted By:

Mandip Chhetri (THA081BEI018)
Kiran Paudel (THA081BEI013)
Purushottam Neupane (THA081BEI033)

Submitted To:

Department of Electronics and Computer Engineering
Thapathali Campus
Kathmandu, Nepal

In partial fulfillment for the award of a Bachelor's degree in Electronics, Communication
and Information Engineering

Under the Supervision of
Prajwol Pakka

March, 2025

DECLARATION

We hereby declare that the report of the project entitled “Event Management System” which is being submitted to the Department of Electronics and Computer Engineering, IOE, Thapathali Campus, in the partial fulfillment of the requirements for the award of the Degree of Bachelor of Engineering in Electronics, Communication and Information/Computer Engineering, is a bonafide report of the work carried out by us. The materials contained in this report have not been submitted to any University or Institution for the award of any degree and we are the only author of this complete work and no sources other than the listed here have been used in this work

Mandip Chhetri	(THA081BEI018)	_____
----------------	----------------	-------

Kiran Paudel	(THA081BEI013)	_____
--------------	----------------	-------

Purushottam Neupane	(THA081BEI033)	_____
---------------------	----------------	-------

DATE: March, 2025

CERTIFICATE OF APPROVAL

The undersigned certify that they have read, and recommended to the Institute of Engineering for acceptance, a project report entitled '**Event Management System**' submitted by **Mandip Chhetri, Kiran Paudel, Purushottam Neupane** in partial fulfillment of the requirements for the Bachelor's Degree in (Electronics, Communication and Information/-Computer) Engineering.

Supervisor: **Prajwol Pakka**

Department of Electronics and Computer Engineering

Institute of Engineering, Thapathali Campus

External Examiner:

Project Co-ordinator: **Er. Umesh Kanta Ghimire**

Department of Electronics and Computer Engineering

Institute of Engineering, Thapathali Campus

Head of Department: **Er. Kiran Chandra Dahal**

Department of Electronics and Computer Engineering

Institute of Engineering, Thapathali Campus

DATE OF APPROVAL: March, 2025

COPYRIGHT

The authors have agreed that the Library, Department of Electronics and Computer Engineering, Institute of Engineering, Thapathali Campus may make this report freely available for inspection. Moreover, the authors have agreed that permission for extensive copying of this project report for scholarly purpose may be granted by the supervisors who supervised the project work recorded herein or in their absence, by the Head of the Department wherein the project report was done. It is understood that the recognition will be given to the authors of this project and to the Department of Electronics and Computer Engineering, Thapathali Campus, Institute of Engineering in any use of the material of this report. Copying or publication or the other use of this report for financial gain without approval of the Department of Electronics and Computer Engineering, Institute of Engineering, Thapathali Campus and authors' written permission is strictly prohibited.

Request for permission to copy or to make any use of the material in this project in whole or part should be addressed to department of Electronics and Computer Engineering, IOE, Thapathali Campus.

ACKNOWLEDGEMENT

This project, **Event Management System**, is prepared in partial fulfilment of the requirements for the Bachelor's degree in Electronics, Communication and Information Engineering. We express our deepest gratitude to the Department of Electronics and Computer Engineering, IOE, Thapathali Campus, for providing us with the opportunity to work on this project as part of our curriculum.

We extend our heartfelt gratitude to our C Programming instructor, **Prajwal Pakka**, for his invaluable guidance and unwavering support throughout the development of this project. His expert insights, particularly in areas such as file handling and data structures, played a crucial role in helping us refine our work and strengthen our programming skills. His encouragement and constructive feedback have been instrumental in making this project a success.

We also appreciate the motivation and collaboration of our peers and colleagues, whose suggestions and insights greatly contributed to improving the functionality and efficiency of our project. Additionally, we acknowledge the various online resources, textbooks, and tutorials that served as valuable references, particularly in areas such as dynamic memory allocation and the implementation of menu-driven interfaces.

The experience of working on this project has significantly enriched our technical knowledge and provided us with practical exposure to project development. In addition, it has strengthened our teamwork skills and reinforced the importance of collaboration in problem solving.

The authors are grateful to everyone who contributed, directly or indirectly, to the successful completion of this project.

Mandip Chhetri (THA081BEI018)

Kiran Paudel (THA081BEI013)

Purushottam Neupane (THA081BEI033)

ABSTRACT

This project focuses on the development of an **Event Management System** using C programming, aimed at simplifying the management and participation in events. The system enables users to create, register, cancel registrations, and view event details, with data stored persistently through file handling. Key functionalities include user authentication, dynamic memory allocation, and a menu-driven interface for seamless interaction.

The project showcases the application of advanced C programming techniques, such as data structures, file handling, and memory management, while reinforcing core programming concepts like loops, conditionals, and input/output operations. Designed for offline use, the system is well-suited for small organizations, educational institutions, and local communities, offering an efficient solution for managing small-scale events.

Keywords: Event Management, C Programming, File Handling, Dynamic Memory Allocation, Menu-Driven Interface

TABLE OF CONTENTS

DECLARATION	i
CERTIFICATE OF APPROVAL	ii
COPYRIGHT	iii
ACKNOWLEDGEMENT	iv
ABSTRACT	v
LIST OF FIGURES	viii
LIST OF ABBREVIATIONS	ix
1 INTRODUCTION	1
1.1 Motivation	1
1.2 Problem Defination	1
1.3 Objectives	2
1.4 Project Applications	2
1.5 Scope of the Project	2
1.6 Report of Organization	3
2 LITERATURE REVIEW	4
2.1 Historical Evolution	4
3 REQUIREMENT ANALYSIS	5
3.1 Hardware Requirements	5
3.2 Software Requirements	5
3.3 Instruments Required	6
4 SYSTEM ARCHITECTURE AND METHODOLOGY	7
4.1 System Architecture	7
4.1.1 Block Diagram of the System	7
4.1.2 Dataflow Diagram	8
4.1.3 Modules of the Project	8
4.1.4 Tools and Environment	9
4.2 Methodology	9
4.2.1 Header Files	9
4.2.2 Functions	10
4.2.3 Conditional Statements	10

4.2.4	Loops	10
4.2.5	Data Structures	10
4.2.6	File Handling	11
5	IMPLEMENTATION DETAILS	12
5.1	User Authentication/Login Module	12
5.2	Main Menu Module	12
5.3	Event Management Module (Admin)	12
5.4	Event Management Module (Organizer)	13
5.5	Event Registration Module (Participant)	13
5.6	Data Verification and Storage	13
5.7	Data Security	14
6	Results and Analysis	15
6.1	System Outputs	15
6.1.1	Login Module	15
6.1.2	Admin Module	15
6.1.3	Organizer Module	16
6.1.4	Participant Module	16
6.1.5	Show Event Option	16
6.1.6	Create Event Option	17
6.1.7	Delete Event Option	17
6.2	Analysis and Discussion	17
6.2.1	Tabular Representation of Results	17
6.3	Error Analysis	18
6.4	Sources of Errors	18
7	FUTURE ENHANCEMENT	19
8	CONCLUSION	20
9	TIME ESTIMATION	21
	REFERENCES	22

List of Figures

4.1	Block Diagram of Event Management System	7
4.2	Dataflow Diagram of Event Management System	8
6.1	Login Module Interface	15
6.2	Admin Module Interface	15
6.3	Organizer Module Interface	16
6.4	Participant Module Interface	16
6.5	Show Event Option	16
6.6	Create Event Option	17
6.7	Delete Event Option	17
9.1	Gantt Chart	21

LIST OF ABBREVIATIONS

CLI	Command Line Interface
GUI	Graphical User Interface
GCC	GNU Compiler Collection
GNU	GNU's Not Unix
IDE	Integrated Development Environment
GDB	GNU Debugger

1. INTRODUCTION

The Event Management System is a C-based application designed to streamline the organization and participation in such events. This project aims to simplify the event management process by providing a user-friendly platform where users can create, register for, and manage events efficiently. Event organizers can specify essential details like the event's date, time, location, and description, while participants can browse available events, register, and track their registrations.

1.1. Motivation

The motivation behind developing the Event Management System stems from the challenges faced by both participants and organizers. Traditional methods of event management, such as notice board announcements, often lead to miscommunication and inefficiencies. In many institutions, event information is scattered across various channels, making it difficult for students to stay updated and for organizers to manage events smoothly.

This system aims to address these issues by centralizing event information digitally, allowing students to easily discover and register for events without the need for manual paperwork. By developing this system, we seek to create a more organized and efficient approach to event management, where both participants and organizers can engage more effectively in college events.

1.2. Problem Defination

Many students are often unaware of upcoming events and hackathons that are essential for their academic and personal growth. In many educational institutions, event information is poorly communicated, often limited to notice boards or word of mouth, leading to confusion and missed opportunities. This lack of awareness and the difficulty in efficiently managing event registrations make it challenging for both students and event organizers to engage effectively.

The primary challenge addressed by this project is the lack of a centralized system that provides up-to-date event information, registration details, rules, and criteria for participation. Students struggle to find relevant event details in a timely manner, and organizers face issues in tracking registrations and communicating event specifics.

1.3. Objectives

The project aims to achieve the following objectives:

- To minimize paperwork and manual record-keeping, promoting a more efficient system.
- To simplify event registration, offering a quick and straightforward process.
- To help participants easily discover events that align with their interests.

1.4. Project Applications

There are many applications of this system, particularly in educational institutions, corporate environments, and community programs. Some of its key applications include:

- **Educational Institutions:** Universities and colleges can utilize the system to manage academic events such as seminars, hackathons, workshops, and cultural programs.
- **Corporate Events:** Companies can use the system to coordinate employee training sessions, meetings, and team-building activities.
- **Community and Social Events:** Local event organizers can efficiently manage social gatherings, charity programs, and public competitions.
- **Academic Competitions:** The system is ideal for managing quizzes, debates, coding contests, and other academic competitions.

1.5. Scope of the Project

The scope of the Event Management System includes streamlining event coordination by automating registration and participation processes. The main aspects of its scope are:

- **Centralized Event Management:** The system eliminates reliance on notice boards and paperwork by providing a structured event management platform.
- **Automated Registration and Tracking:** Participants can register for events digitally, and organizers can efficiently track registrations.

- **Role-Based Access:** Organizers can create and manage events, while participants can browse, register, and cancel their participation.
- **Scalability:** The system is designed to be scalable, allowing future enhancements such as online integration, notifications, and user analytics.

1.6. Report of Organization

This report is divided into Eight sections, each covering different aspects of the Event Management System. The organization of the report is as follows:

- **Chapter 1: Introduction** – This chapter provides an overview of the Event Management System, including its Intro, motivation, problem definition, objectives, applications, and scope.
- **Chapter 2: Literature Review** – This section explores the history and evolution of event management techniques, reviewing existing systems and their limitations.
- **Chapter 3: System Architecture** – Discusses the overall structure of the system, including the system architecture and data flow diagrams.
- **Chapter 4: Methodology** – Describes the working principles of the system, including file handling, data structures, functions, and logic implementation.
- **Chapter 5: Implementation details** – Provides details about the implementation of the system, including coding techniques, development environment, and integration of various modules.
- **Chapter 6: Results and Analysis** – Illustrates the outputs of the system, demonstrating its functionality through sample cases and test results.
- **Chapter 7: Future Enhancements** – Discusses potential improvements and extensions that can be added to the system in future versions.
- **Chapter 8: Conclusion and References** – Summarizes key findings and insights gained from the development of the project, followed by a list of references used.

2. LITERATURE REVIEW

The Event Management System represents an innovative approach to streamline the organization and coordination of events, particularly within academic environments. Digital event management platforms are designed to centralize event information, automate registration processes, and enhance communication between organizers and participants. This literature review examines the evolution, underlying methodologies, and recent advancements in digital event management systems, with an emphasis on implementations using the C programming language.

2.1. Historical Evolution

Early methods of event management were predominantly manual, relying on physical notice boards, printed forms, and word-of-mouth to disseminate information. These traditional approaches were not only inefficient but also prone to errors, leading to missed opportunities and poor event turnout. With the advent of digital technologies, rudimentary computer-based systems were developed to address these challenges. These initial systems employed basic file handling techniques to store and manage event data, marking a significant improvement over manual methods. However, they were limited by the technology of their time—lacking scalability, robust user interfaces, and integration with other digital platforms.

Over the years, enhancements in data structures and dynamic memory allocation have enabled the creation of more efficient and scalable platforms. These advancements have significantly improved the speed and accuracy of data retrieval and storage, ensuring that event-related information is accessible and up-to-date.

Modern event management systems integrate a variety of methodologies to deliver a seamless user experience. They incorporate modular programming approaches to break down complex functionalities into manageable components, ensuring ease of maintenance and scalability. Whether implemented with a GUI or CLI model, these systems focus on centralizing event information, automating registration processes, and providing real-time updates to users. The core objective is to empower organizers to manage events efficiently while allowing participants to quickly discover and register for events without the hassles of traditional manual processes.

3. REQUIREMENT ANALYSIS

The following details the hardware, instruments, and software essential for both the development and operation of the Event Management System.

3.1. Hardware Requirements

The following hardware components are essential for the development and operation of the system:

- **Processor:** A minimum of an Intel Core i3 or AMD Ryzen 3 processor is required to ensure smooth execution of the application.
- **RAM:** At least 4 GB of RAM is recommended for efficient compilation and execution of the C-based program.
- **Storage:** A minimum of 100 MB of free disk space is needed to store source files, event data, and compiled binaries.
- **Input Devices:** A standard keyboard and mouse are necessary for user interaction within the CLI.
- **Display:** A screen with a resolution of 1024×768 or higher is required for clear visibility of the CLI interface.

3.2. Software Requirements

The system requires the following software components for development and execution:

- **Operating System:** The system is compatible with Windows, Linux, and macOS.
- **Compiler:** GCC is used to compile and execute the C program.
- **IDE:** VS Code or Code::Blocks is recommended for writing and debugging the program.
- **Libraries:** Standard C libraries such as `stdio.h`, `stdlib.h`, `string.h`, and `conio.h` are used for input/output operations, memory management and data conversion, and file handling.

3.3. Instruments Required

Although no specialized instruments are required, the following resources are useful for system testing and debugging:

- **Text Editor:** A lightweight text editor like Notepad++ for quick code modifications.
- **Debugging Tools:** Built-in debugging tools in IDEs such as GDB for error detection.
- **Version Control System:** GitHub or Git for version tracking and collaborative development.

4. SYSTEM ARCHITECTURE AND METHODOLOGY

This section outlines the theoretical background, system architecture, and methodologies used in the development of this Project. It describes the architecture, algorithms, and implementation techniques that ensure efficient event management.

4.1. System Architecture

The architecture of the Event Management System defines how various modules interact to provide a seamless event management experience. It consists of different layers responsible for handling user interactions, event data storage, and system functionalities.

4.1.1. Block Diagram of the System

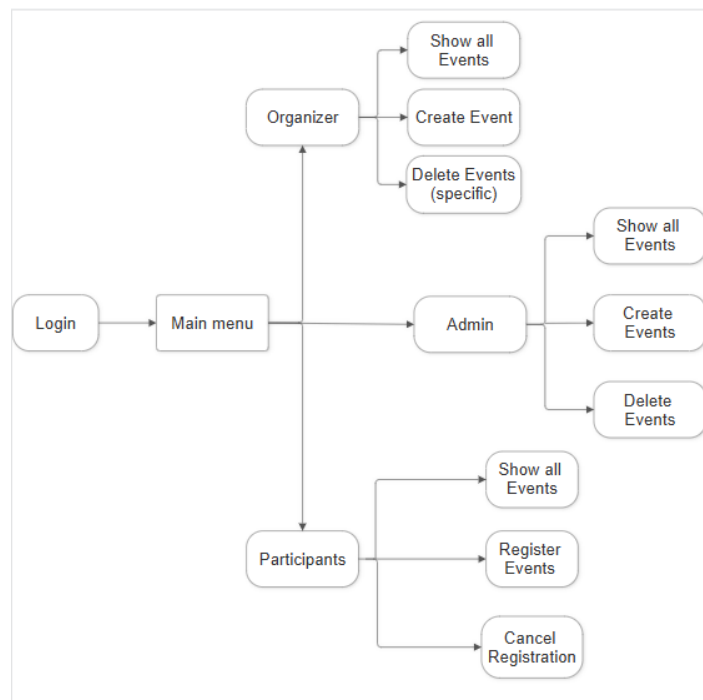


Figure 4.1: Block Diagram of Event Management System

4.1.2. Dataflow Diagram

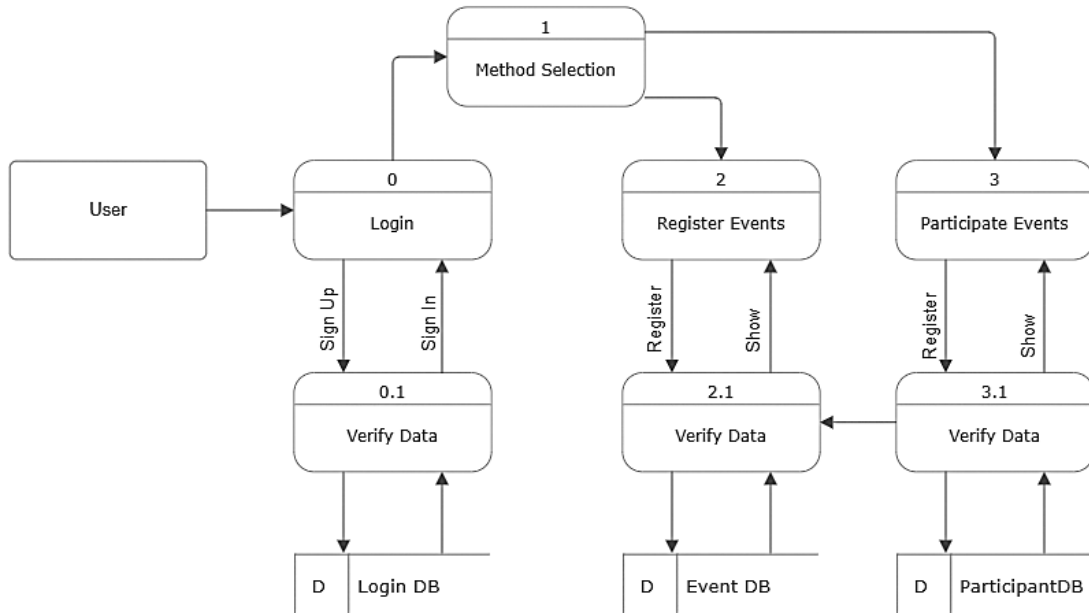


Figure 4.2: Dataflow Diagram of Event Management System

4.1.3. Modules of the Project

The system consists of several key modules that work together to provide event management functionalities:

- **User Interface:** The interface follows a menu-driven structure that allows users to navigate different functionalities after login. It provides options for event organizers and participants.
- **Login Module:** The system includes a login mechanism to authenticate users and ensure secure access. It directs users to the appropriate sections based on their roles.
- **Admin Module:** The Admin module provides the following functionalities:
 - Show All Events: Displays all events available in the system.
 - Create Events: Allows Admin to add new events with relevant details.
 - Delete Events: Enables Admin to remove all existing events.
- **Organizer Module:** The organizer module provides the following functionalities:
 - Show Events: Displays all events available in the system.

- Create Events: Allows organizers to add new events with relevant details.
- Delete Events: Enables organizers to remove existing events created by them.
- **Participant Module:** The participant module allows users to:
 - Show Events: View all available events.
 - Register for Events: Sign up for selected events.
 - Cancel Registration: Withdraw from registered events.
- **Error Handling Mechanism:** The system includes error handling for invalid selections, duplicate registrations, and invalid event cancellations, ensuring proper system operation.
- **Exit Function:** This safely terminates the program while saving event and registration data.

4.1.4. Tools and Environment

The system is developed using C programming and compiled using GCC. Development is carried out using VS Code, ensuring efficient debugging and testing. The system is cross-platform and compatible with both Windows and Linux.

4.2. Methodology

The development of the Event Management System follows a structured approach utilizing key programming concepts such as file handling, data structures, and modular programming.

4.2.1. Header Files

The following header files are used for different functionalities:

- `stdio.h` - Standard input/output operations.
- `stdlib.h` - Dynamic memory allocation and Data Conversion.
- `string.h` - String manipulation functions.
- `conio.h` - Console input/output functions (mainly `getch()` function).

4.2.2. Functions

Functions play a crucial role in modularizing the system and improving code readability. Key functions include:

- `createEvent()` - Creates and stores event details.
- `registerParticipant()` - Registers a participant for an event.
- `viewEvents()` - Displays all available events.
- `cancelRegistration()` - Cancels a participant's registration.
- `login()` - Authenticates users.

4.2.3. Conditional Statements

Conditional statements are employed for:

- **User Authentication:** Verifying credentials and granting appropriate access.
- **Event Validation:** Ensuring correct input formats and preventing duplicate events.
- **Participant Management:** Checking for duplicate registrations and handling errors effectively.

4.2.4. Loops

Loops are used for iterative operations such as:

- Displaying event details from files.
- Searching for registered participants.
- Keeping the system running until exit is selected.

4.2.5. Data Structures

Data structures such as arrays and structs are used to store event and participant information before writing to files.

4.2.6. File Handling

File handling ensures data persistence by storing and retrieving event and registration details from text files.

5. IMPLEMENTATION DETAILS

This section details the implementation of core modules. It covers the functionality and specific implementation steps for user authentication and login, the role-based main menu, and the administrative event management features. The explanation emphasizes data handling, access control, and navigation techniques employed within each module.

5.1. User Authentication/Login Module

Functionality: Users can log in as Admin (default), Organizer, or Participant. Organizers and Participants have the option to sign up and sign in.

Implementation:

- Users enter credentials for authentication.
- The system verifies the credentials against the `users.dat`.
- Successful authentication grants access to the main menu accordingly.

5.2. Main Menu Module

Functionality: Provides role-based access to organizers and participants.

Implementation:

- While login, users get their role (Admin, Organizer or Participant) according to the Username and Password that they provided.
- The system directs them to the appropriate dashboard.
- Implemented using C switch-case statements for menu navigation.

5.3. Event Management Module (Admin)

Functionality: Allows Admin to create, manage, and delete all events.

Implementation:

- Admin can create events by entering details such as name, date, and description. Also able to delete all events which are registered in the system.
- Events are stored in a file-based `events.dat`.

5.4. Event Management Module (Organizer)

Functionality: Allows organizers to create, manage, and delete events.

Implementation:

- Organizers can create events by entering details such as name, date, and description.
- Events are stored in a file-based `event.dat`.
- Organizers can delete events that they registered, updating the file accordingly.

5.5. Event Registration Module (Participant)

Functionality: Allows participants to register for events.

Implementation:

- Participants view available events from the `event.dat`.
- They register for events, and the system updates the `registrations.dat`.

Technologies Used:

- Programming Language: C
- Data Storage: File Handling (for user credentials)
- Security: Hashing for password protection in binary

5.6. Data Verification and Storage

Functionality: Ensures the integrity of user and event data.

Implementation:

- Login credentials are validated before storing in the `users.dat`.
- Event details are checked for completeness before adding to the `events.dat`.
- Participant registrations are verified before updating the `registrations.dat`.

5.7. Data Security

- Encrypted storage of sensitive user data in Binary Files.
- Implemented password masking with asterisks during input.
- Input validation to prevent buffer overflow and invalid entries.

6. Results and Analysis

This section presents the results and outputs of the Event Management System (EMS). The outputs are displayed in the form of tables, graphs, and figures. Additionally, an analysis of the results, error sources, and validation techniques is included.

6.1. System Outputs

The following images illustrate the outputs of different modules in the system:

6.1.1. Login Module

```
===== LOGIN MENU =====  
1. Admin  
2. Organizer  
3. Participant  
4. Exit  
Enter your choice: █
```

Figure 6.1: Login Module Interface

6.1.2. Admin Module

```
===== ADMIN MENU =====  
1. Show All Events  
2. Create Event  
3. Delete Event  
4. Logout  
Enter your choice: █
```

Figure 6.2: Admin Module Interface

6.1.3. Organizer Module

```
===== ORGANIZER MENU =====
1. Show All Events
2. Show My Events
3. Create My Event
4. Delete My Event
5. Logout
Enter your choice: █
```

Figure 6.3: Organizer Module Interface

6.1.4. Participant Module

```
===== PARTICIPANT MENU =====
1. Show Available Events
2. Register for Event
3. Show Registered Events
4. Cancel Registration
5. LogOut
Enter your choice: █
```

Figure 6.4: Participant Module Interface

6.1.5. Show Event Option

```
===== AVAILABLE EVENTS =====
```

ID	Name	Location	Date	Time	Registered/Max
1	Yathartha	Thapathali	2/12/2024	10:00	1/99 (Registered)
2	Locus	Pulchowk	2/4/2025	10:00	0/99
3	DevFest	KU	4/2/2025	12:00	0/25
4	Tech-EX	Softwarica	5/6/2025	10:00	0/49
5	WordCamp	NICT	5/5/2025	10:00	0/60
6	COC	Thapathali	3/4/2025	11:00	1/99 (Registered)

```
Press any key to continue...█
```

Figure 6.5: Show Event Option

6.1.6. Create Event Option

```
===== CREATE NEW EVENT =====
Event Name: Yathartha
Event Description: this is yathartha 2.0
Location: Thapathali
Date (DD/MM/YYYY): 2/12/2024
Time (HH:MM): 10:00
Maximum Participants: 99

Event created successfully with ID: 1

Press any key to continue...|
```

Figure 6.6: Create Event Option

6.1.7. Delete Event Option

```
===== DELETE EVENT =====
Available events:
ID      Name      Location      Date      Time      Registered/Max
-----
1       Yathartha    Thapathali    2/12/2024  10:00     1/99
2       Locus        Pulchowk      2/4/2025   10:00     0/99
3       DevFest      KU            4/2/2025   12:00     0/25
4       Tech-EX      Softwarica    5/6/2025   10:00     0/49
5       WordCamp     NICT          5/5/2025   10:00     0/60
6       COC          Thapathali    3/4/2025   11:00     1/99

Enter event ID to delete(0 for cancel): |
```

Figure 6.7: Delete Event Option

6.2. Analysis and Discussion

6.2.1. Tabular Representation of Results

The following table summarizes the system performance metrics based on different test cases:

Test Case	Expected Output	Actual Output
Login with valid credentials	Successful login	Successful login
Login with invalid credentials	Access denied	Access denied
Event creation by organizer	Event added	Event added
Event deletion by organizer	Event removed	Event removed
Participant event registration	Registration successful	Registration successful

Table 6.1: System Test Cases and Results

6.3. Error Analysis

The following factors contributed to minor deviations from expected outputs:

- Incorrect user input leading to failed authentication attempts.
- Minor delays in file handling operations due to large data storage.
- Edge cases where simultaneous event registrations caused conflicts.

6.4. Sources of Errors

Potential sources of errors in the system include:

- User input errors (e.g., incorrect format or missing details).
- File handling limitations (e.g., improper data synchronization).
- Buffer overflows in case of extremely large inputs.

7. FUTURE ENHANCEMENT

This Event Management System can be improved in several ways to enhance functionality, security, and user experience. Future enhancements include:

- **Web and Mobile Application:** Expand the system to a web-based and mobile platform using technologies like Django, React, Flutter, or React Native.
- **Database Integration:** Replace file handling with a relational database like MySQL or PostgreSQL for better performance, structured data storage, and faster retrieval.
- **Graphical User Interface (GUI):** Develop a GUI to improve user interaction and navigation.
- **Automated Notifications:** Implement email, SMS, and push notifications to inform users about event updates, reminders, and registrations.
- **Payment Gateway Integration:** Enable online payment methods like PayPal, Stripe, or Razorpay for paid events, allowing secure transactions.
- **AI-Based Event Recommendations:** Use artificial intelligence to suggest events based on user interests, past participation, and engagement patterns.
- **Cloud-Based Storage:** Host the system on cloud platforms like AWS, Google Cloud, or Azure for scalability, data security, and remote accessibility.
- **Enhanced Reporting and Analytics:** Provide detailed reports, statistics, and graphical analysis of event participation and trends.
- **User Role Expansion:** Introduce additional roles such as event sponsors, vendors, and moderators to broaden the system's functionality.

8. CONCLUSION

Overall, this project was an exciting and rewarding experience that allowed us to collaborate and apply our knowledge in C programming to solve a real-world problem. Working together on the Event Management System not only strengthened our technical skills but also improved our ability to tackle challenges as a team.

Throughout the development process, we encountered various hurdles, from implementing efficient file handling to ensuring seamless user interactions. Overcoming these obstacles required creativity, persistence, and teamwork, which ultimately helped us grow both as programmers and problem solvers.

This project has broadened our understanding of event management systems and reinforced the importance of structured programming, modular design, and user-centric development. It has also ignited our curiosity about future enhancements, motivating us to explore advanced technologies such as databases, web development, and artificial intelligence for similar applications.

In the end, this project was more than just writing code—it was an experience that pushed us beyond our comfort zones and showcased the possibilities of what we can achieve through dedication and teamwork.

9. TIME ESTIMATION

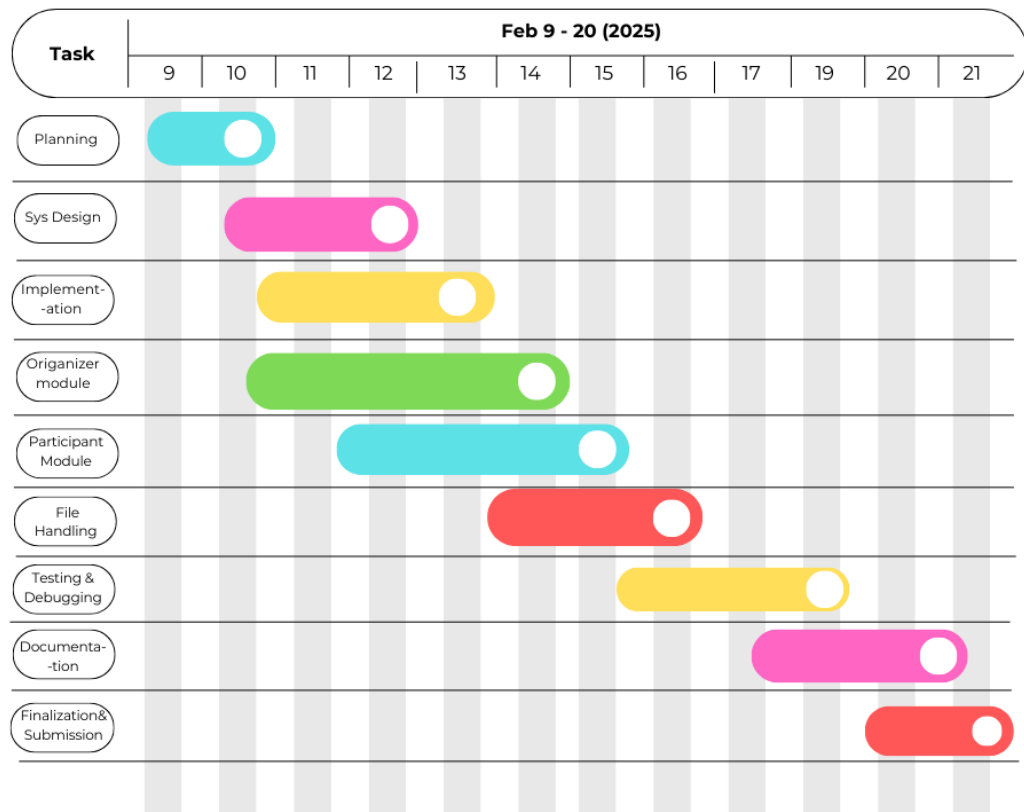


Figure 9.1: Gantt Chart

References

- [1] MeetingBox, "A Brief History of Event Management," October 2024. Available: <https://www.meetingbox.com/post/a-brief-history-of-eventmanagement-event-planning-from-the-1990s-to-present>. [Accessed: Feb. 8, 2025].
- [2] eLeapSoftware, "The Evolution of Event Management in the Digital Age," n.d. Available: <https://www.eleapsoftware.com/glossary/the-evolution-of-event-management-in-the-digital-age/>. [Accessed: Feb. 8, 2025].
- [3] Ananya U, Shetty Kavya Umesh, Shraddha Harish Mendon, Priya Poojary, Joseph Michael Jerard V, "Event Management System for Educational Institutions," *International Journal of Creative Research Thoughts*, 2022. Available: <https://ijcrt.org/papers/IJCRT22A6460.pdf>. [Accessed: Feb. 8, 2025].
- [4] Gaurav Thombare, Pavan Jadhav, Sachin Dhere, Prasad Jadhav, K.N. Honwadkar, "College Event Management: A Survey of Analytics and Personalization," *International Research Journal of Modern Engineering and Technology*, 2022. DOI: <https://www.doi.org/10.56726/IRJMETS45827>. [Accessed: Feb. 8, 2025].