TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
THAPATHALI CAMPUS

C-Programming Project Proposal
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

Feb, 2025

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

List of Figures

LIST OF ABBREVIATIONS

CLI	Command Line Interface
GUI	Graphical User Interface
GCC	GNU Compiler Collection
GNU	GNU's Not Unix

1. INTRODUCTION

1.1. Background

Many events are organized on a daily basis, ranging from cultural programs to exhibitions and competitions, offering opportunities for participants to showcase their skills and knowledge. These events are essential for fostering learning, collaboration, and personal growth. However, managing such events, coordinating registrations, and ensuring smooth participation can often become a cumbersome task.

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.

Using file handling and data structures in C, the system ensures efficient data storage and retrieval, making it easier for organizers to manage event details and participant registrations. The application provides a centralized platform that supports a wide range of events, from cultural and academic programs to competitions like ”LOCUS” and ”YATHARTHA.”

This project demonstrates key C programming concepts such as file handling, data structures, and functions, providing an efficient solution for managing events and ensuring seamless participation for both organizers and participants.

1.2. 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.

1.3. 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.

1.4. 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.

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. PROPOSED SYSTEM ARCHITECTURE

3.1. Block Diagram or System Architecture

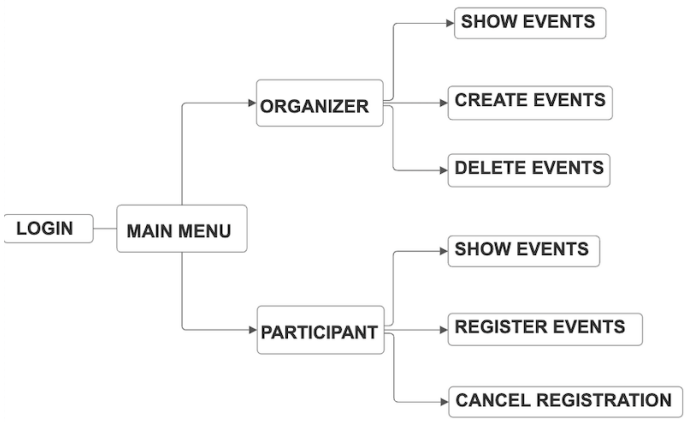

Figure 3.1: Block Diagram of Event Management System

3.2. Dataflow Diagram

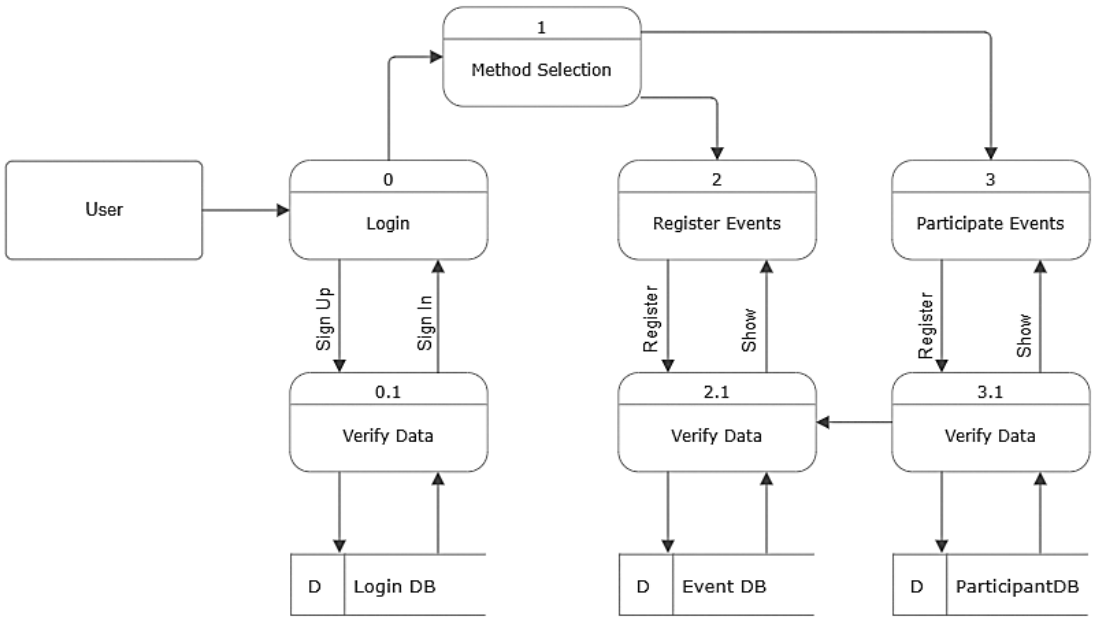

Figure 3.2: Dataflow Diagram of Event Management System

3.3. Modules of Project

- **User Interface** The interface of the event management system includes a menu-driven structure where users can access different modules after login. The interface provides a main menu that branches into two primary user roles: Organizer and Participant.
- **Login Module** The system begins with a login interface that authenticates users and directs them to the main menu. This ensures secure access and proper role assignment for system functionality.
- **Organizer Module** The organizer module contains three primary functions:
  - Show Events: Displays all events in the system
  - Create Events: Allows organizers to add new events with relevant details
  - Delete Events: Enables organizers to remove existing events from the system
- **Participant Module** The participant module provides three essential functions:
  - Show Events: Lists available events for registration
  - Register Events: Allows participants to sign up for selected events
  - Cancel Registration: Enables participants to withdraw from registered events
- **Error Handling Mechanism** The system implements comprehensive error handling to manage scenarios such as invalid selections, duplicate registrations, or attempting to cancel non-existent registrations. Appropriate error messages guide users through proper system usage.
- **Exit** The exit function safely terminates the program, ensuring all event data and registration information is properly saved before closing the application.

3.4. Tools and Environment

This Event Management System is developed using the C programming language, utilizing the GCC compiler for code compilation and execution. Visual Studio Code (VS Code) is used as the primary development environment, providing an efficient workspace with integrated debugging tools and extensions for C programming. The system is designed to run on both Windows and Linux, ensuring cross-platform compatibility and accessibility.

4. METHODOLOGY

The various methodologies that will be used in the development of the Event Management project include different header files, functions, conditional statements, loops, data structures, and file handling, which are explained below.

4.1. Header Files

In C language, a header file is a text file that contains pieces of code written in the C programming language. The name of a header file, by convention, ends with the `.h` extension. It is inserted inside a program by coding the `#include` preprocessor directive. The program will use various header files to perform tasks such as input/output operations, string manipulation, and file handling. The header files that will be used include:

- `stdio.h`: For standard input/output operations.
- `stdlib.h`: For dynamic memory allocation.(`if necessary`).
- `string.h`: For string manipulation functions.
- `conio.h`: For console input/output functions.

4.2. Functions

Functions will be an essential part of the development of our project. The use of functions helps to decrease the repeatability of code and improves modularity. Functions will allow us to define reusable blocks of code that can be used whenever a specific task needs to be performed. This will improve code readability as well as efficiency. Key functions in the project include:

- `createEvent()`: To create a new event and store it in a file.
- `registerParticipant()`: To register a participant for an event.
- `viewEvents()`: To display all available events or registered events.
- `cancelRegistration()`: To cancel a participant's registration for an event.
- `login()`: To authenticate users (organizers and participants).

4.3. Conditional Statements

Conditional statements will be employed throughout the program to handle user inputs, validate data, and manage program flow. They will serve the following purposes:

- **User Authentication:** Conditional statements will check if the entered username and password match the stored credentials. Based on the user type (organizer or participant), the system will display the appropriate menu options.
- **Event Creation:** Conditional statements will validate event details (e.g., date format, duplicate event titles). If the input is invalid, the system will display an error message and prompt the user to re-enter the details.
- **Participant Registration:** Conditional statements will check if the participant is already registered for the event. If the participant is already registered, the system will display a message and prevent duplicate registrations.
- **Error Handling:** Conditional statements will handle errors such as invalid event IDs during registration, attempts to cancel a registration that does not exist, and file access errors (e.g., if `events.txt` or `participants.txt` cannot be opened).

4.4. Loops

Loops will be used to iterate through data structures and files, enabling efficient data processing. Key uses of loops include:

- **Displaying Events:** A loop will iterate through the `events.txt` file to display all available events.
- **Searching for Registrations:** A loop will search through the `participants.txt` file to find a participant's registered events.
- **Menu Navigation:** A loop will keep the program running until the user chooses to exit.

4.5. Data Structures

Data structures such as structs and arrays will be used to organize and store event and participant data.

Arrays of structs will provide temporary storage for events and participants before writing to files.

4.6. File Handling

File handling will be used to store event and participant data persistently.

5. SCOPE AND APPLICATIONS

This C-based Event Management System is designed to streamline the organization and participation in college events, providing a user-friendly platform for both event organizers and attendees. The system allows organizers to create new events by specifying essential details—such as date, time, location, and a brief description—to inform and attract potential participants. At the same time, it facilitates a smooth registration process, enabling students to browse available events, view detailed information, and manage their own registrations with ease.

By leveraging file handling and data structures in C, the system ensures efficient storage and rapid retrieval of event data, including both event details and participant registrations. This robust functionality establishes a solid foundation for managing a wide range of college events, from cultural programs and exhibitions to competitive events like ”LOCUS”, ”YATHARTHA”, etc.

Applications:

- **College Events Management:** The system is ideal for managing all types of college events, ranging from academic seminars and workshops to sports events and cultural programs. It allows event organizers to create, manage, and monitor events with ease.
- **Student Participation and Registration:** Students can quickly browse through a variety of events, register for those that interest them, and manage their registrations efficiently. The system helps reduce the administrative workload involved in event sign-ups and cancellations.
- **Event Information Dissemination:** The system serves as a central hub for distributing information about upcoming events, making it easy for both organizers and participants to stay updated on event schedules, locations, and other essential details.
- **Error-Free Event Management:** The application ensures that events are properly managed and prevents common issues such as double registrations or invalid event entries, making it a reliable tool for organizers and participants alike.

6. TIME ESTIMATION

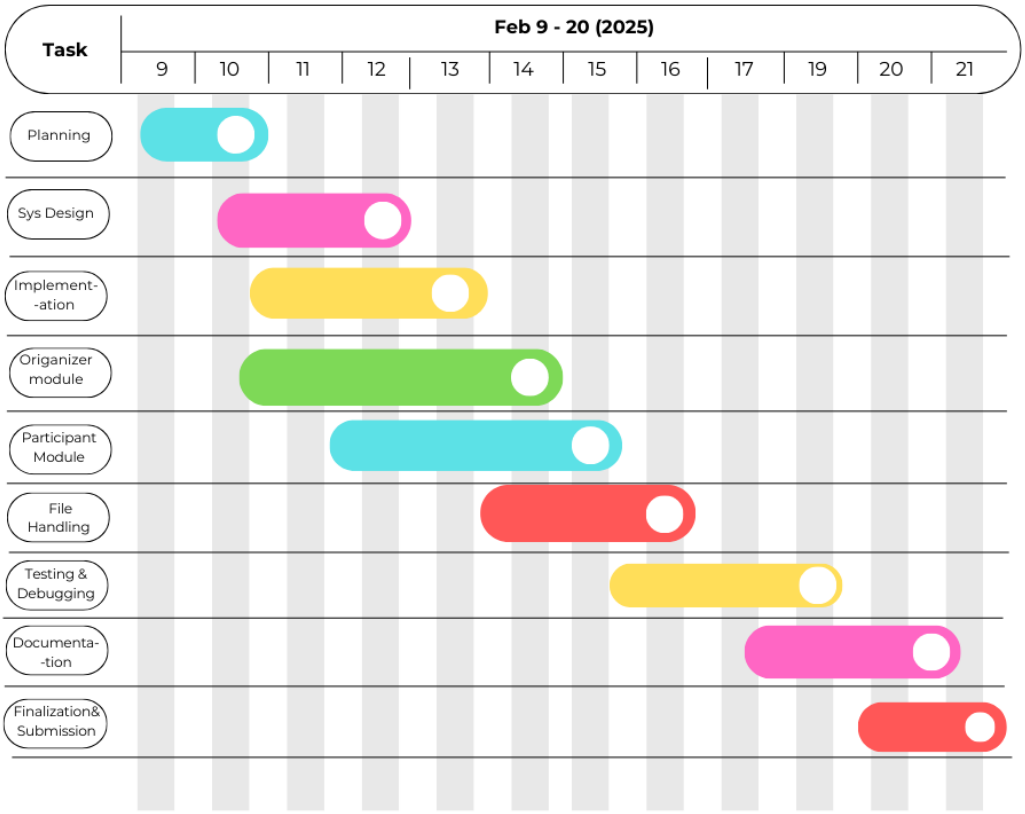

Figure 6.1: Gantt Chart

7. FEASIBILITY ANALYSIS

Technical Feasibility:

The Event Management System is technically feasible and well-suited for the requirements of our project. C programming is an ideal choice for handling the event-related data, as it offers efficient memory management and strong support for file handling and data structures. The system has been designed in a modular way, breaking down tasks into smaller, manageable components, making development and maintenance easier. C is also widely used, meaning there is a wealth of online resources, tutorials, and community support to assist in solving any challenges that may arise. Additionally,it is technically viable and achievable within the given timeframe.

Legal Feasibility:

From a legal perspective, the Event Management System adheres to basic data privacy guidelines. Since the system will involve handling student information, it is crucial to implement proper consent mechanisms and ensure that data storage and retrieval are done in compliance with relevant data privacy laws and institutional regulations. The system should have clear terms of use, privacy policies, and security measures to safeguard sensitive data and prevent unauthorized access. Adhering to these principles ensures that the project remains legally sound and compliant with data protection practices.

Economical Feasibility:

Economically, the system presents significant savings for the organizers in terms of time and resources. The automated process of event registration and management will reduce administrative costs associated with manual processes such as paper forms and event tracking. The initial development cost is minimal, as it primarily involves programming and system setup, both of which are manageable within the scope of a student project. Once developed, the system will have very low maintenance costs, making it a cost-effective solution for the long term. Overall, this system is a valuable investment that will improve efficiency, save money, and streamline event management processes.

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].