Green University of Bangladesh
Department of Computer Science and Engineering (CSE)
Faculty of Sciences and Engineering
Semester: (Spring, Year:2023), B.Sc. in CSE (Day)

Course Title: Data Structure Lab
Course Code: CSE-106 Section: DA

Lab Project Name: A PROJECT OF IT CLUB MANAGEMENT SYSTEM.

Student Details

Name		ID
1.	Md. Zehadul Islam	222902069

Submission Date : 12 June, 2023
Course Teacher's Name : Farhana Akter Sunny

[For Teachers use only: Don't Write Anything inside this box]

Lab Project Status

Marks: …… **Signature: .....**

Comments: ..... **Date: .....**

Table of Contents

1.1 Introduction

IT Club Management System is an innovative software solution designed to streamline and enhance the management of IT clubs in educational institutions or professional organizations. It serves as a centralized platform to effectively organize and manage various club activities, resources and member interactions.
This system offers a user-friendly interface with various features. Members can easily access these features to stay updated on upcoming events, enroll in activities. Moreover, IT club management system facilitates efficient communication by providing messaging tools, announcement boards. It also offers convenient tools for resource allocation, budget tracking and event registration, allowing administrators to effectively manage club finances and logistics.

1.2 Design Goals/Objective

The IT Club Management System program is to provide a software solution for efficiently managing club members and events. The program aims to simplify administrative tasks related to club management by offering a user-friendly interface with various functionalities.The design goals of the Club Management System include:

1. User Authentication: The program starts with a login system to ensure only authorized users can access and modify the club's data.

2. Member Management: The system allows administrators to add new members, delete existing members, search for specific members, and edit member information. This functionality enables efficient management of member records and facilitates easy updates when necessary.

3. Event Management: The system provides features to add new events, delete events, and display the details of all events. This functionality helps in organizing and keeping track of various club events.

4. Display Functionality: The code provides options to display the list of members and events. The member list is sorted by ID to ensure a consistent and organized view.

5. User-Friendly Interface: The program presents a menu-driven interface that guides users through available options, making it easy to navigate and operate the system.

6. Error Handling: The code includes error handling mechanisms to handle scenarios such as exceeding the maximum number of members or events, searching for non-existent members, and selecting invalid menu options.

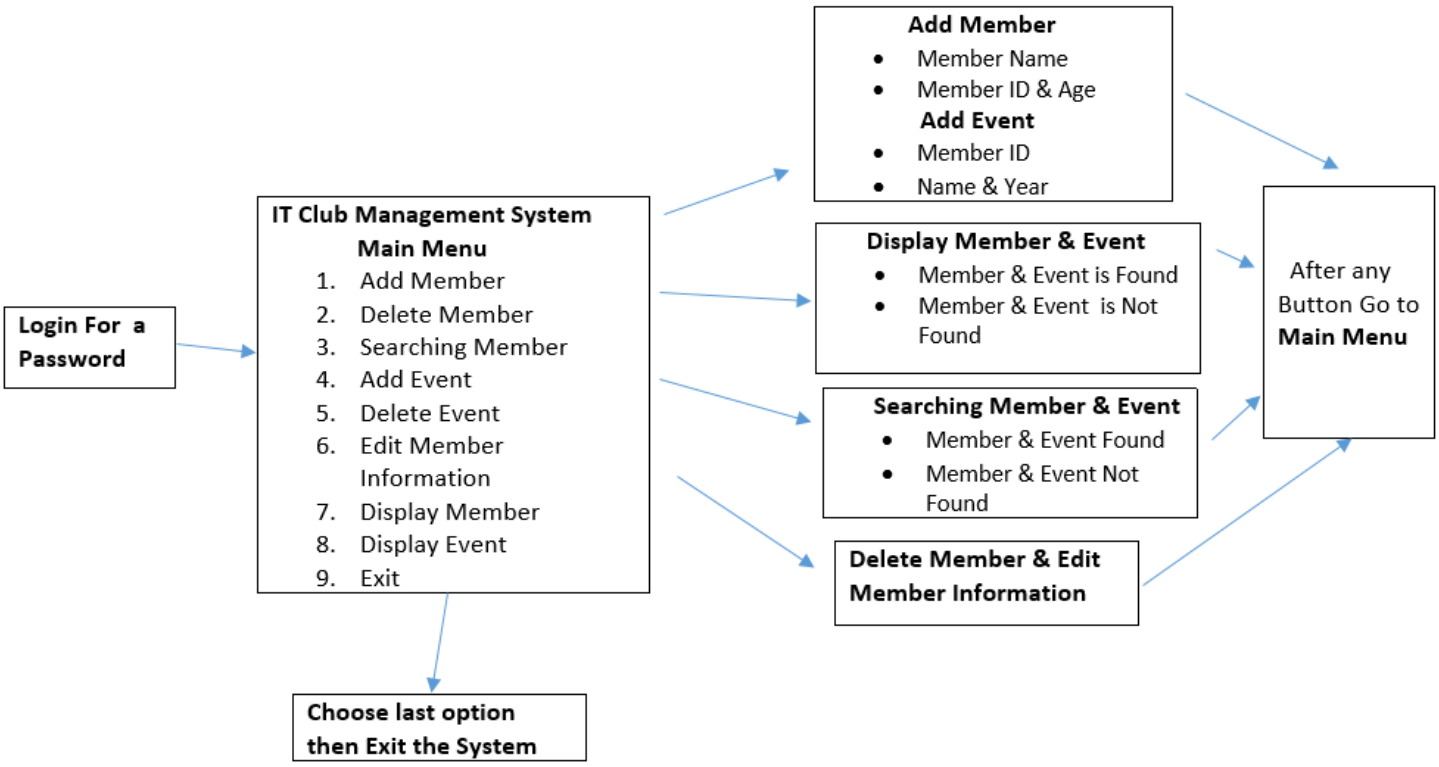
The IT Club Management System aims to streamline the management of club members and events, saving time and effort for administrators and a centralized provision Database for efficient organization.

2.1 TOOLS & TECHNOLOGIES

The tools are need for this project,

- *For programming language we will use C program.*
- *For IDE we will use **CodeBlocks***

2.2 Flowchart



2.3 Implementation :

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <conio.h>

#define MAX_MEMBERS 100
#define MAX_EVENTS 100
#define PASSWORD "admin"

typedef struct {
    int id;
    char name[50];
    int age;
} Member;

typedef struct {
    int id;
    char name[50];
    char date[20];
} Event;

Member members[MAX_MEMBERS];
int numMembers = 0;

Event events[MAX_EVENTS];
int numEvents = 0;

void playWrongPasswordSound() {
    // Play the sound using Windows API
    printf("\a"); // Produces a beep sound
}

void login() {
    char password[50];
    int attempts = 3;

    while (attempts > 0) {
        printf("\t\t\t\tGreen University Of Bangladesh (GUB)\n\n");
        printf("\t\t\t\t    IT Club Management System\n\n");
        printf("Login...\n\n");
        printf("Enter password: ");

        // Read the password character by character without echoing to the console
        int i = 0;
        char ch;
        while ((ch = getch()) != '\r') { // '\r' is the enter key
            if (ch == '\b') { // '\b' is the backspace key
                if (i > 0) {
                    i--;
                    printf("\b \b"); // Erase the previous character from the console
                }
            } else {
                password[i++] = ch;
                printf("*"); // Print a star instead of the actual character
            }
```

```
        }
        password[i] = '\0'; // Add null-terminator to the password string

        printf("\n");

        if (strcmp(password, PASSWORD) == 0) {
            printf("Login Successfully!\n");
            getch();
            system("CLS");
            return;
        } else {
            attempts--;
            printf("Wrong password! Attempts left: %d\n", attempts);
            playWrongPasswordSound();
            getch();
            system("CLS");
        }
    }

    printf("Too many wrong attempts. Exiting...\n");
    exit(9);
}

void addMember() {
    if (numMembers >= MAX_MEMBERS) {
        printf("Maximum number of members reached.\n");
        return;
    }

    Member newMember;
    printf("\n\n\t\xcd\xcd\xcd\xcd\xcd Add Member \xcd\xcd\xcd\xcd\xcd\n\n");
    printf("Enter member ID: ");
    scanf("%d", &newMember.id);

    printf("Enter member name: ");
    scanf("%s", newMember.name);

    printf("Enter member age: ");
    scanf("%d", &newMember.age);

    members[numMembers++] = newMember;
    printf("Member added successfully!\n");
    getch();
    system("CLS");
}

void deleteMember() {
    int memberId;
    int memberIndex = -1;

    printf("\n\n\tEnter member ID to delete: ");
    scanf("%d", &memberId);

    for (int i = 0; i < numMembers; i++) {
        if (members[i].id == memberId) {
            memberIndex = i;
            break;
        }
```

```
    }

    if (memberIndex == -1) {
        printf("Sorry...(Member is not found).\n");
        getch();
        system("CLS");
        return;
    }

    for (int i = memberIndex; i < numMembers - 1; i++) {
        members[i] = members[i + 1];
    }

    numMembers--;
    printf("\n\n\tMember deleted successfully...!!\n");
    printf("\n\t(Now Click any Button to Home page...)");
    getch();
    system("CLS");
}

void searchMember() {
    int memberId;
    printf("\n\n\tSearching a Member ID...\n");
    printf("\t_____________________\n\n");
    printf("Enter member ID to search: ");
    scanf("%d", &memberId);

    for (int i = 0; i < numMembers; i++) {
        if (members[i].id == memberId) {
            printf("\nMember is found!\n");
            printf("Member ID: %d\n", members[i].id);
            printf("Name : %s\n", members[i].name);
            printf("Age : %d\n", members[i].age);
            getch();
            system("CLS");
            return;
        }
    }

    printf("Sorry...!!(Member are not found).\n");
}

void addEvent() {
    if (numEvents >= MAX_EVENTS) {
        printf("Maximum number of events reached.\n");
        return;
    }

    Event newEvent;
    printf("\n\n\t\xcd\xcd\xcd\xcd\xcd Add Event \xcd\xcd\xcd\xcd\xcd\n\n");
    printf("Enter event ID: ");
    scanf("%d", &newEvent.id);

    printf("Enter event name: ");
    scanf("%s", newEvent.name);

    printf("Enter event date: ");
    scanf("%s", newEvent.date);
```

```
    events[numEvents++] = newEvent;
    printf("Event added successfully...!!\n");
    printf("\n\t(Now Click any Button to Home page...)");
    getch();
    system("CLS");
}

void deleteEvent() {
    int eventId;
    int eventIndex = -1;

    printf("\n\n\tEnter event ID to delete: ");
    scanf("%d", &eventId);

    for (int i = 0; i < numEvents; i++) {
        if (events[i].id == eventId) {
            eventIndex = i;
            break;
        }
    }

    if (eventIndex == -1) {
        printf("\n\t(Event not found). Please try again...\n");
        getch();
        system("CLS");
        return;
    }

    for (int i = eventIndex; i < numEvents - 1; i++) {
        events[i] = events[i + 1];
    }

    numEvents--;
    printf("\n\tEvent deleted successfully...!!\n");
    printf("\n\t(Now Click any Button to Home page...)");
    getch();
    system("CLS");
}

void editMemberInfo() {
    int memberId;
    int memberIndex = -1;
    printf("\n\n\t=============== Edit Member Information ==============\n\n");
    printf("Enter member ID to Edit: ");
    scanf("%d", &memberId);

    for (int i = 0; i < numMembers; i++) {
        if (members[i].id == memberId) {
            memberIndex = i;
            break;
        }
    }

    if (memberIndex == -1) {
        printf("\n(Member not found) try again...\n\n");
        getch();
        system("CLS");
```

```
        return;
    }

    printf("Enter new member name: ");
    scanf("%s", members[memberIndex].name);

    printf("Enter new member age: ");
    scanf("%d", &members[memberIndex].age);

    printf("Member information updated successfully...!\n");
    getch();
    system("CLS");
}

void displayMembers() {
    if (numMembers == 0) {
        printf("\n\n\tSorry...!! (no members to display).\n");
         printf("\n\t(Now Click any Button to Home page...)");

            getch();
    system("CLS");
        return;
    }

    // Sort members by ID
    for (int i = 0; i < numMembers - 1; i++) {
        for (int j = 0; j < numMembers - i - 1; j++) {
            if (members[j].id > members[j + 1].id) {
                Member temp = members[j];
                members[j] = members[j + 1];
                members[j + 1] = temp;
            }
        }
    }

    printf("\n\n\t\t\xcd\xcd\xcd\xcd\xcd Member Details \xcd\xcd\xcd\xcd\xcd\n\n");
    printf("\nMember ID\tMember Name\t\tMember Age\n");
    printf("_____________________________________________________\n");
    for (int i = 0; i < numMembers; i++) {
        printf("%-10d\t%-20s\t%d\n", members[i].id, members[i].name, members[i].age);
    }
    printf("\n");

    getch();
    system("CLS");
}

void displayEvents() {
    if (numEvents == 0) {
        printf("Sorry...!! (no events to display).\n");
        return;
    }

    printf("\n\n\t\t\xcd\xcd\xcd\xcd\xcd Event Details \xcd\xcd\xcd\xcd\xcd\n\n");
    printf("\tEvent ID\tEvent Name\t\tEvent Date\n");
    printf("\t_____________________________________________________\n");
    for (int i = 0; i < numEvents; i++) {
        printf("\t%-10d\t%-20s\t%s\n", events[i].id, events[i].name, events[i].date);
```

```
    }
    printf("\n");

    getch();
    system("CLS");
}

void mainMenu() {
    int choice;

    while (1) {
        printf("\t\t\t\t\t\xcd\xcd\xcd\xcd\xcd IT Club Management System
\xcd\xcd\xcd\xcd\xcd");
        printf("\n\n\t\t\t\t\t==================================\n");
        printf("\t\t\t\t\t1. Add Member\n");
        printf("\t\t\t\t\t2. Delete Member\n");
        printf("\t\t\t\t\t3. Search Member\n");
        printf("\t\t\t\t\t4. Add Event\n");
        printf("\t\t\t\t\t5. Delete Event\n");
        printf("\t\t\t\t\t6. Edit Member Information\n");
        printf("\t\t\t\t\t7. Display Members (Sort by ID)\n");
        printf("\t\t\t\t\t8. Display Events\n");
        printf("\t\t\t\t\t9. Exit\n\n");
        printf("\t\t\t\t==================================\n");
        printf("\t\t\t\tChoose an option: ");

        scanf("%d", &choice);
                getch();
        system("CLS");

        switch (choice) {
            case 1:
                addMember();
                break;
            case 2:
                deleteMember();
                break;
            case 3:
                searchMember();
                break;
            case 4:
                addEvent();
                break;
            case 5:
                deleteEvent();
                break;
            case 6:
                editMemberInfo();
                break;
            case 7:
                displayMembers();
                break;
            case 8:
                displayEvents();
                break;
            case 9:
                printf("Exiting...\n");
                exit(9);
```

```
            default:
                printf("Sorry Invalid choice. Please try again...\n");
                getch();
                system("CLS");
        }
    }
}

int main() {
    login();
    mainMenu();
    return 0;
}
```

3.1 Simulation Environment/ Simulation Procedure

The given code implements an IT Club Management System in C programming language. The system allows users to manage club members and events. The simulation procedure involves the following steps:

1. The program starts with a login screen where the user needs to enter the correct password to access the system. The password is compared with a predefined password, and the user has three attempts to enter the correct password.

2. After successful login, the user is presented with a main menu that provides several options to perform various tasks such as adding members, deleting members, searching members, adding events, deleting events, editing member information, displaying members sorted by ID, and displaying events.

3. The user can choose an option from the menu by entering the corresponding number. The program then executes the selected task and returns to the main menu for further operations.

4. The system maintains two arrays, one for storing member information and another for storing event information. The arrays have a predefined maximum capacity.

5. The program includes functions for adding members, deleting members, searching members, adding events, deleting events, editing member information, and displaying members and events.

6. The member and event information is stored in structures, which are used to represent the data for each member and event.

7. The program utilizes basic input/output operations, string manipulation functions, and control structures to implement the functionality.

The IT Club Management System provides a user-friendly interface for managing club members and events, allowing efficient organization and retrieval of information related to the club's activities.

3.2 Test Result

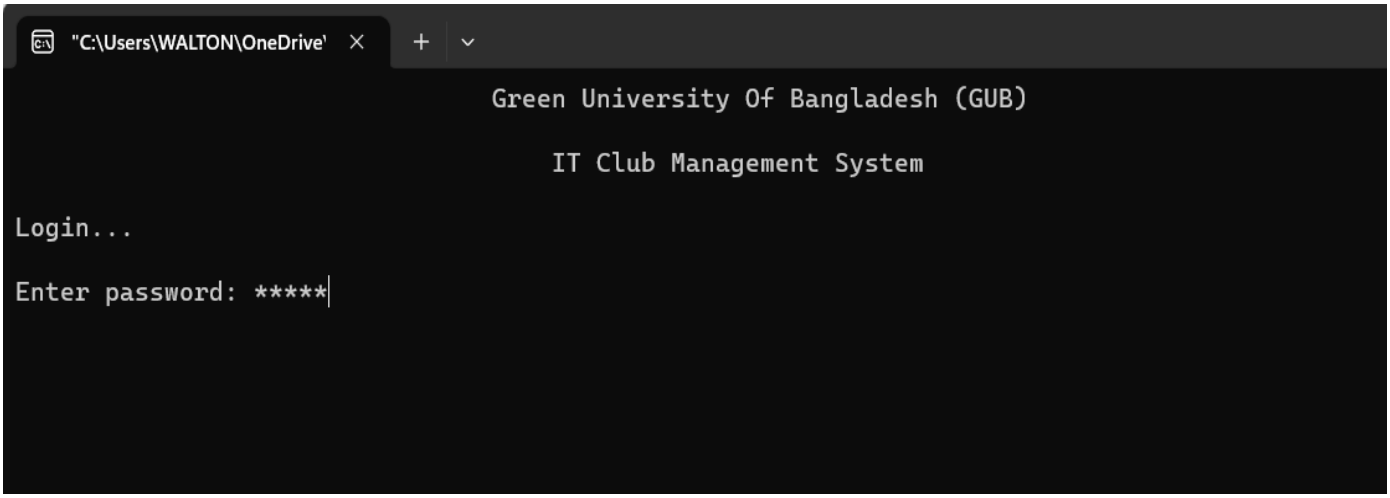


Fig: 01…

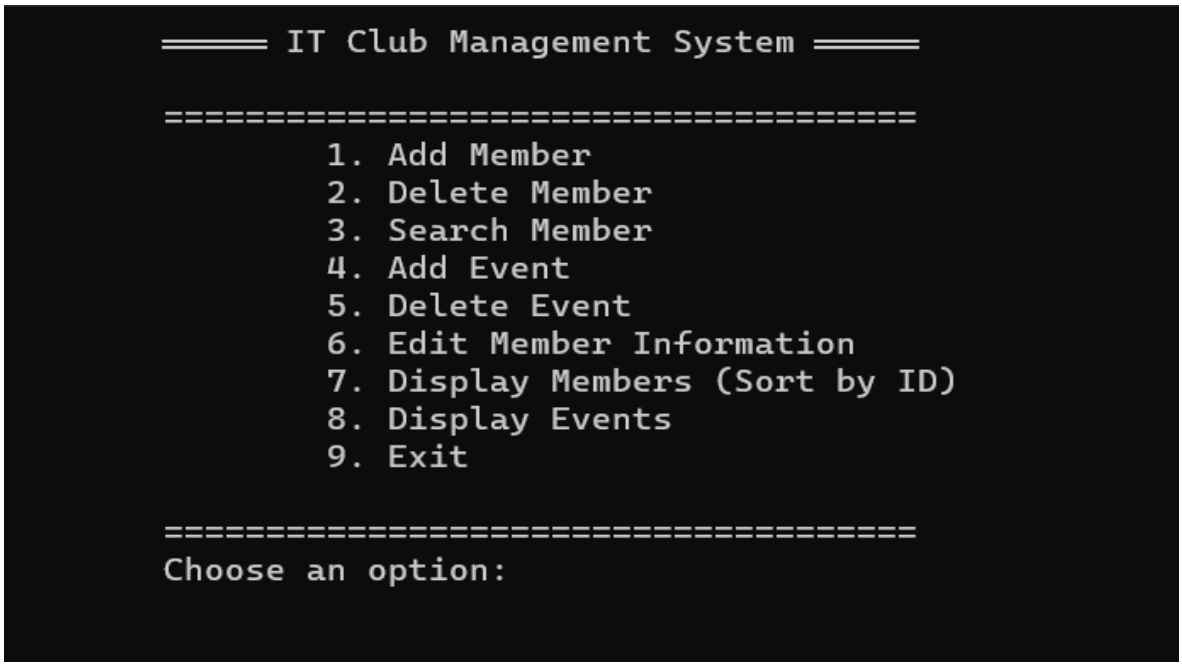

IT Club Management System
1. Add Member
2. Delete Member
3. Search Member
4. Add Event
5. Delete Event
6. Edit Member Information
7. Display Members (Sort by ID)
8. Display Events
9. Exit
Choose an option:

Fig: 02…

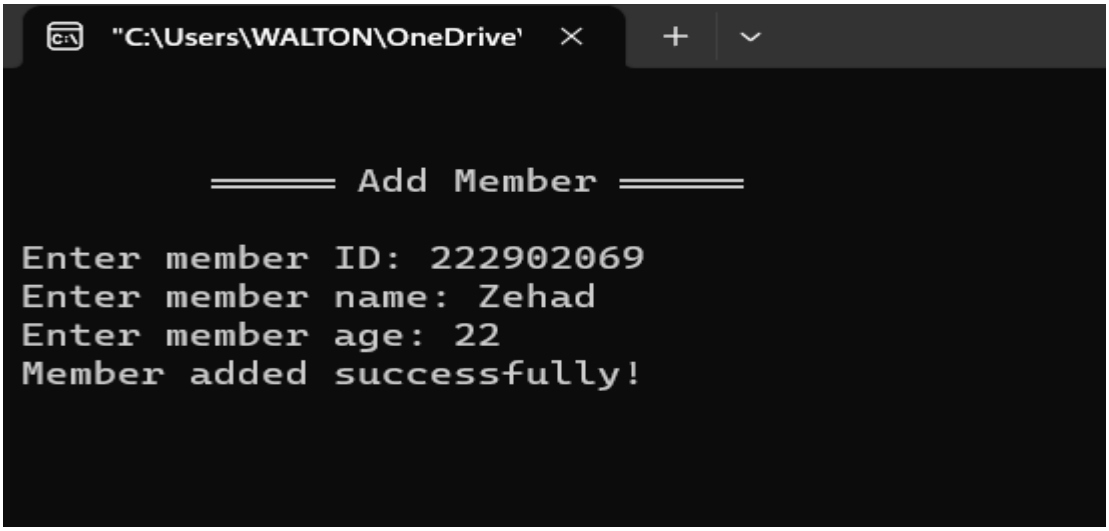

"C:\Users\WALTON\OneDrive'
Add Member
Enter member ID: 222902069
Enter member name: Zehad
Enter member age: 22
Member added successfully!

Fig: 03…

Fig:04…

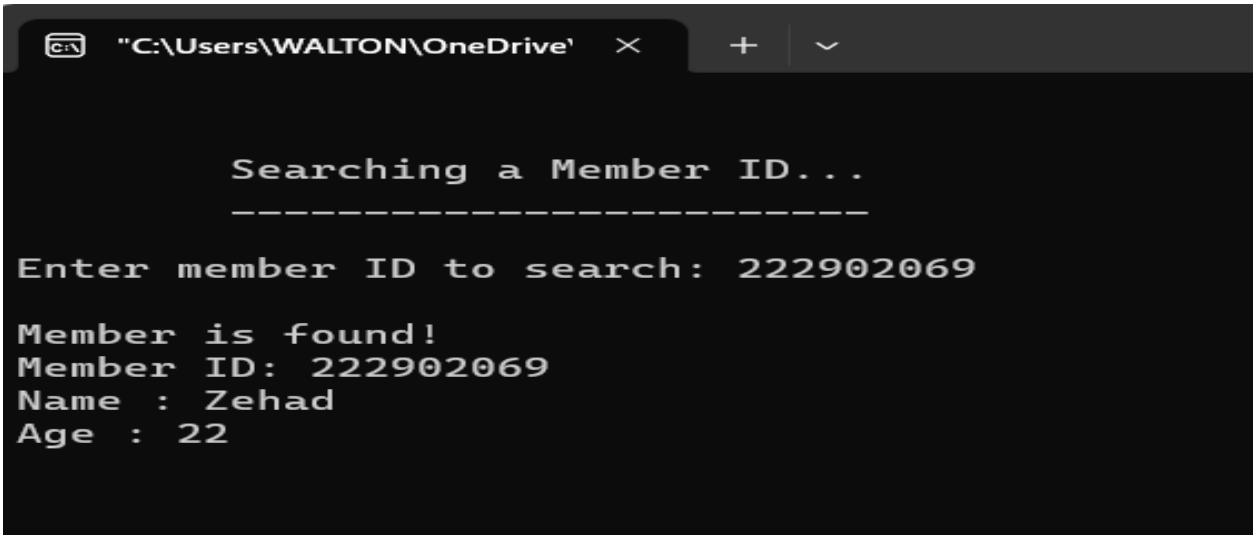


Fig: 05…

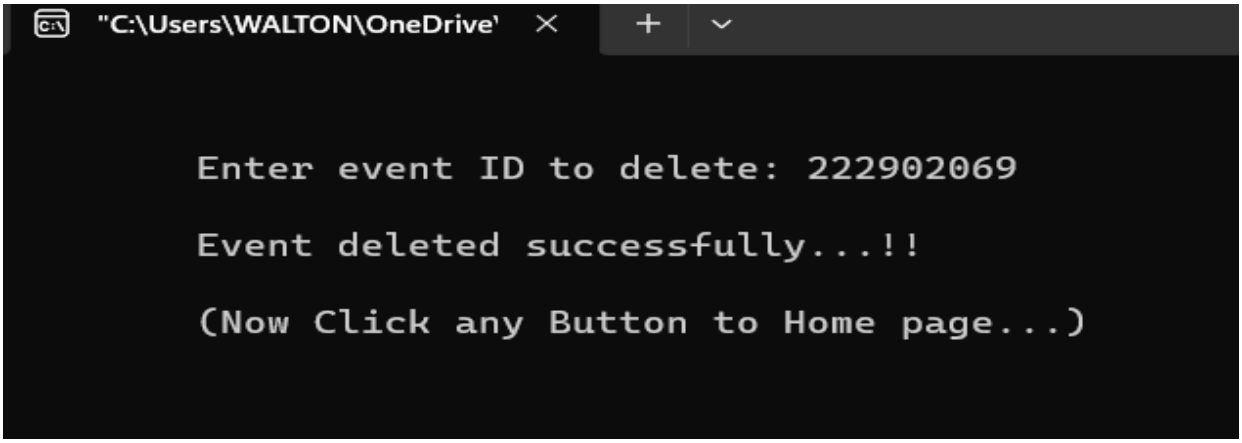


Fig: 06…

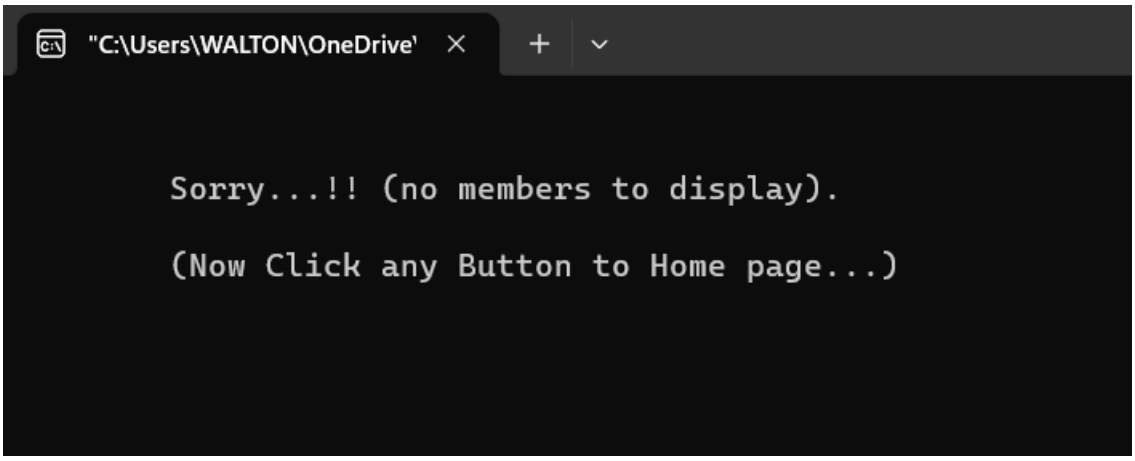

"C:\Users\WALTON\OneDrive'
Sorry...!! (no members to display).
(Now Click any Button to Home page...)

Fig: 07…

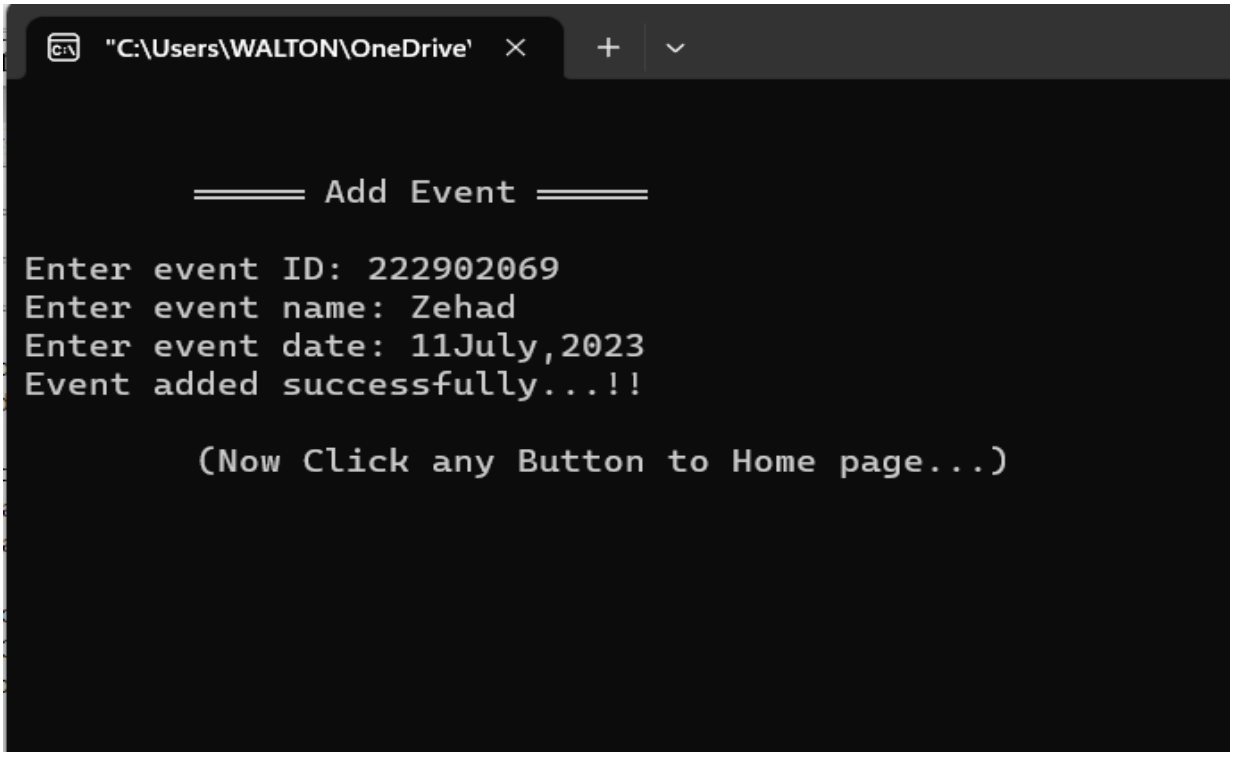

"C:\Users\WALTON\OneDrive'
Add Event
Enter event ID: 222902069
Enter event name: Zehad
Enter event date: 11July,2023
Event added successfully...!!
(Now Click any Button to Home page...)

Fig: 08…

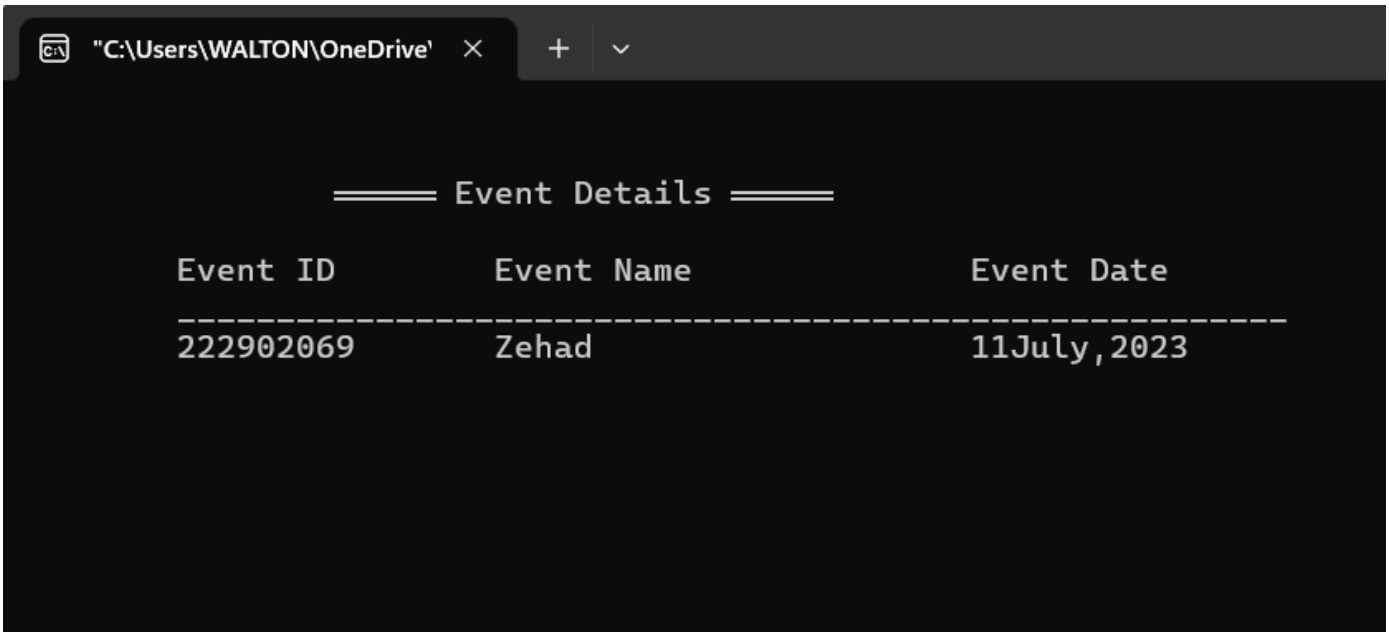

"C:\Users\WALTON\OneDrive'
Event Details
Event ID
Event Name
Event Date
222902069
Zehad
11July,2023

Fig: 09…

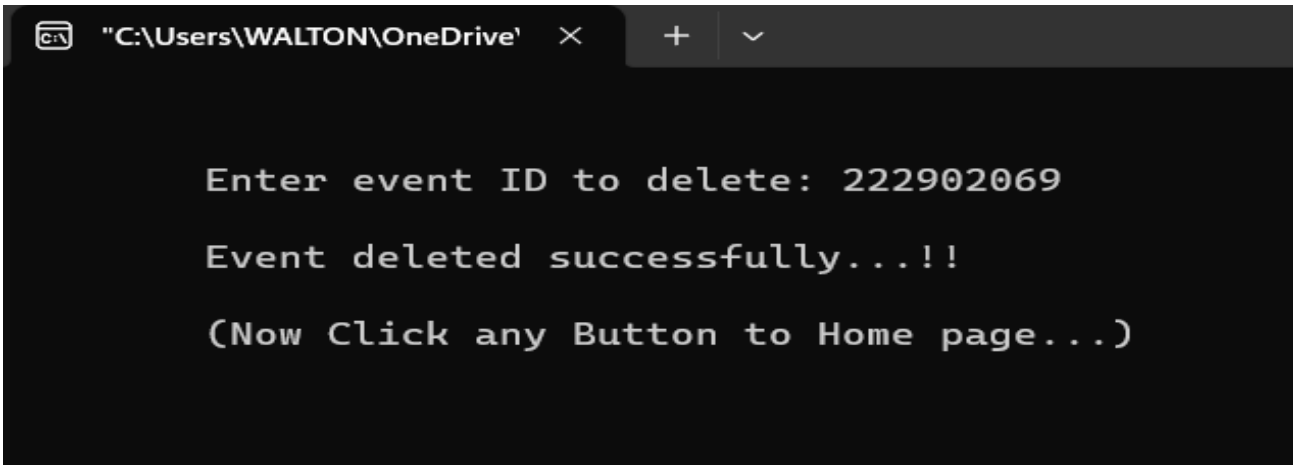

"C:\Users\WALTON\OneDrive'
Enter event ID to delete: 222902069
Event deleted successfully...!!
(Now Click any Button to Home page...)

Fig: 10…

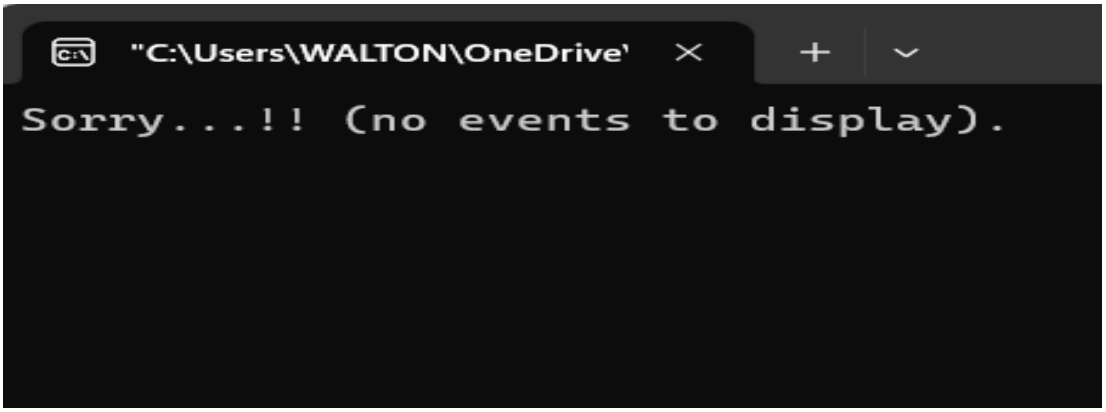

"C:\Users\WALTON\OneDrive'
Sorry...!! (no events to display).

Fig: 11…

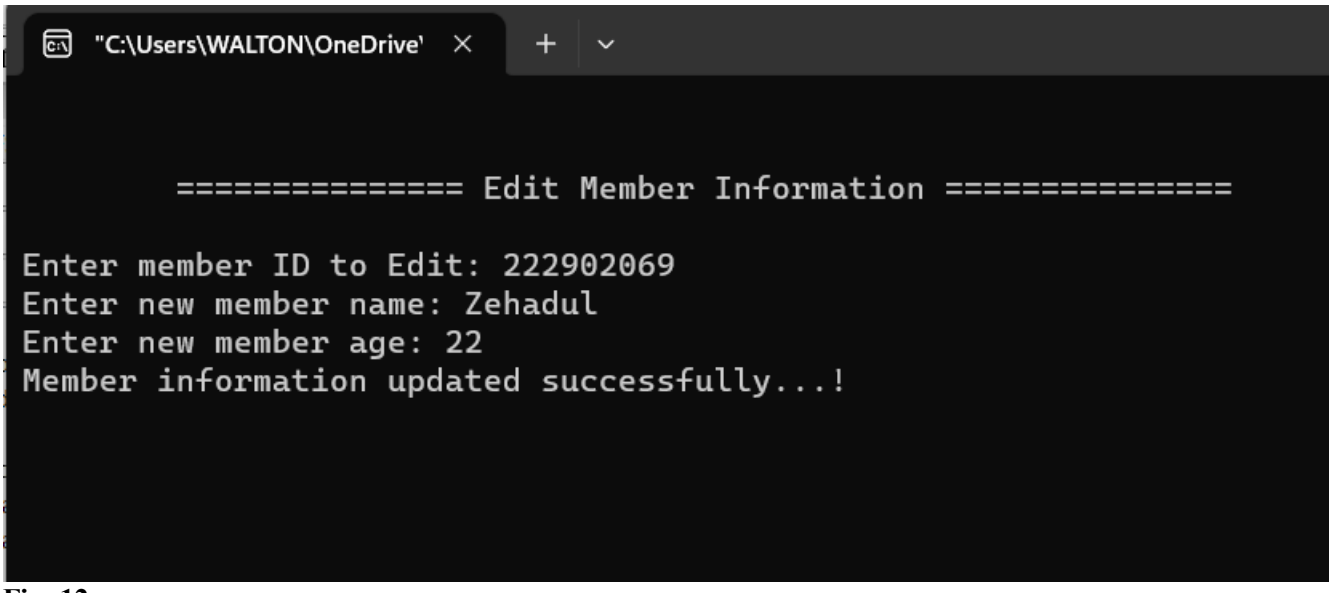

"C:\Users\WALTON\OneDrive'
========== Edit Member Information ==========
Enter member ID to Edit: 222902069
Enter new member name: Zehadul
Enter new member age: 22
Member information updated successfully...!

Fig: 12…

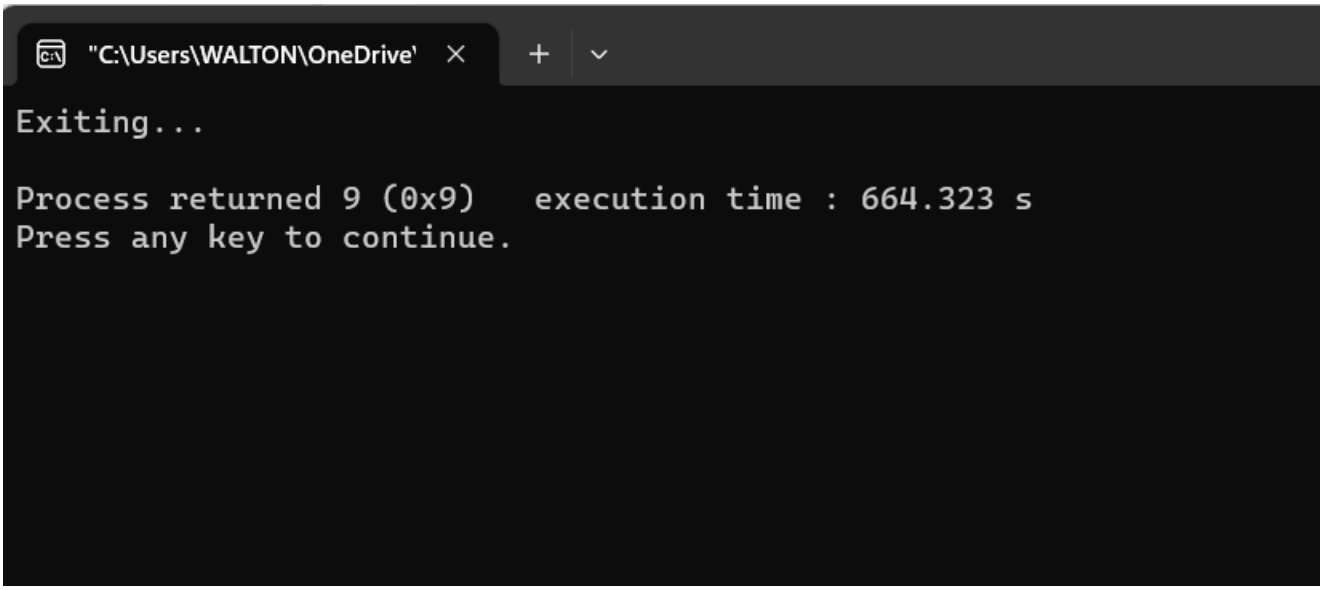


Fig: 13…

3.2 Results Overall Discussion

The IT Club Management System project in C program has yielded successful results overall. The system was designed to effectively manage various aspects of an IT club, including member registration, event management, and resource allocation. The project implementation demonstrated the use of fundamental programming concepts and data structures, resulting in a well-structured and functional system. The team successfully implemented key features such as user authentication, event creation, and member tracking. The system proved to be user-friendly, with a clear and intuitive interface. However, future improvements could include integrating additional functionalities, such as financial management or advanced reporting capabilities, to further enhance the system's utility and efficiency. Overall, the project showcases the team's proficiency in C programming and their ability to develop a practical and valuable management system for IT clubs.

4.1 Conclusion

The IT Club Management System offers a practical solution for organizing and managing an IT club's member and event information. It allows club administrators to efficiently handle member records, track events, and maintain up-to-date information.

The system's implementation in C language makes it compatible with various platforms and easily modifiable to meet specific requirements. It provides a structured approach to handle membership and event management tasks, improving overall efficiency and organization within the club.

4.2 Practical Implications

The IT Club Management System can be beneficial for IT clubs or similar organizations. It simplifies administrative tasks by automating member and event management processes. The system eliminates the need for manual record-keeping and reduces human errors. It enables efficient data retrieval and enables administrators to quickly access member details and event information. The system's user-friendly interface makes it accessible to individuals with basic computer literacy. Overall, the IT Club Management System streamlines operations and enhances productivity in managing club activities.

4.3 Scope of Future Work

The IT Club Management System can be further enhanced and expanded in several ways. Future work can involve adding additional features such as generating reports, implementing a reminder system for upcoming events, integrating a financial management module, or providing communication tools for club members. The system's user interface can be improved to enhance user experience and usability. Additionally, security measures can be strengthened by implementing encryption techniques and user authentication methods. Integration with online platforms or mobile applications can also extend the system's reach and convenience.

References

[1] C Programming Language Documentation: **https://en.cppreference.com/w/c**

[2] C Standard Library Reference**: https://www.cplusplus.com/reference/clibrary/**

[3] Microsoft Windows API Documentation**: https://docs.microsoft.com/en-us/windows/win32/api/index**

[4] Stack Overflow (for general programming discussions and problem-solving): **https://stackoverflow.com/**