|  |  |
| --- | --- |
|  |  |

**IT CLUB Management System PROJECT**

**A PROJECT REPORT**

*Submitted in partial fulfilment of the requirements for the degree of*

**BACHELORS OF COMPUTER APPLICATION**

|  |  |
| --- | --- |
| ***Submitted By:***  **[Adarsh Sharma]**  **[24AEH0403]** | ***Submitted To:***  **[Ms. Nisha Chaudhary]** |

**Department of CS & IT**

**Avviare Educational Hub**

**Noida, UP – 201301, India**

**ABSTRACT**

**----------About Project---------**

**IT Club Management System is based on the concept of managing records and details of members in an IT club or office before being entered into the dashboard system. To enter the system, the users must first pass through a login system to access the features. This mini-project consists of limited features- however, features that are necessary.**

**Keywords: - IT department, Management System, C Programming, Management of Employees**

**ACKNOWLEDGEMENT**

I are highly grateful to the Ms. Kanika Singh, Director of Operations, Avviare Educational Hub, Noida for providing this opportunity to carry out the major project work. The constant guidance and encouragement received from Mr. Ashutosh Rathore H.O.D. IT Department, Avviare Educational Hub, Noida has been of great help in carrying out the project work and is acknowledged with reverential thanks. I would like to express a deep sense of gratitude and thanks profusely to Ms. Nisha Choudhary, without her wise counsel and able guidance, it would have been impossible to complete the project in this manner. I express gratitude to other faculty members of IT department of Avviare Educational Hub, Noida for their intellectual support throughout the course of this work. Finally, I are indebted to all whosoever have contributed in this report work.

**Name: Adarsh Sharma**

**Date: 20/11/2024**

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **F. No.** | **Figure Description** | **Page No.** |
| **1.** | **IT Club Management system** | **6** |
| **2.** | **Output** | **16** |
| **3.** | **Output** | **17** |
| **4.** | **Output** | **18** |
| **5.** | **Flow Chart** | **19** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| **Content** | **Page No.** |
|  | **2** |
|  | **3** |
|  | **4** |
|  | **5** |
|  | **6** |
|  | **7** |
|  | **8** |
|  | **9** |
|  | **10 - 15** |
|  | **16 - 18** |
|  | **19** |
|  | **20** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**The system creates an external file to store the user’s data permanently. IT Club Management system is developed using C Programming Language and different variables, strings have been used for the development of it.  
IT Club Management System in C Programming with source code is free to download. Use for educational purposes only! For the project demo, have a look at the video below.**

**The IT Club Management System, lets the user add member details by entering the name, member ID, address, contact number, and department name. By the member ID, the user can easily remove, search for, or edit the member information. The user may view the list of all available members with details and department names. The last feature of this project deals with managing Events: here, the user can view, delete, and also add an event with some event date, name, and information.**

**Features:**

1. **Login System**
2. **Manage IT members**
3. **Manage Events**

**Code Structure:**

1. **Structured Club Data Type**
2. **Main Menu Interface**
3. **Functions of Edit, Add, and Delete the members in IT Department**

****

**Project Requirement**

**Function Requirements:**

1. **User Management** 
   * Login/Logout Functionality
   * User Role (Manager, Administrator)
   * Password Protection
2. **Club Management**
   * Add new member
   * Display all the members
   * Search the member
   * Delete existing member

**3. Club Management System**

- Club members details (their Names, IDs, Age, Email addresses)

- Track Member availability (available, deleted, or not found in system)

1. **Reporting** 
   * Generate a report of club member availability and their information history

**Technical Requirements:**

1. **Programming Language:** - C
2. **Database Management:** - File based or relational data base (e.g. MySQL)
3. **Operating System:** - Windows/ Linux
4. **Hardware requirement: -** 
   * 1. **Processor** - intel Core i3 or equivalent
     2. **RAM**  - 4GB or more
     3. **Storage** - 500GB or more

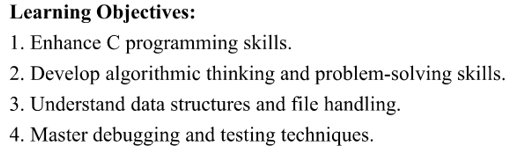
**OBJECTIVE**

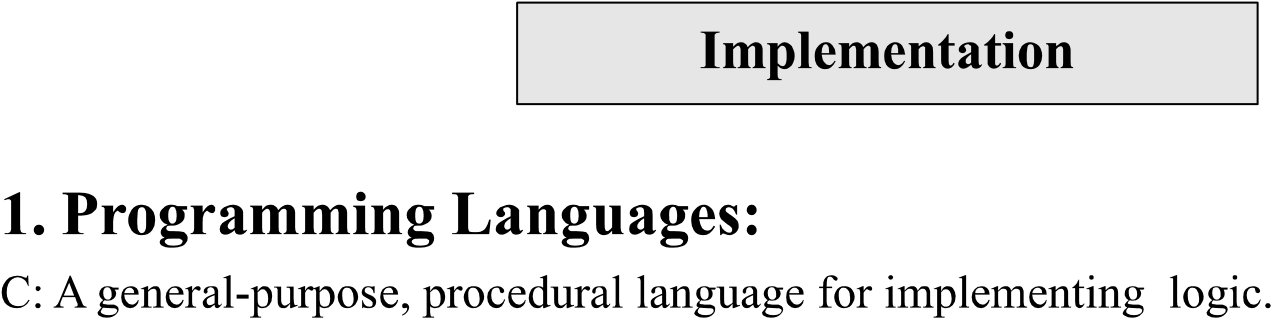
* + **Main Goal:** Create an IT Club Management System to practice C programming skill.
  + **Use Concepts:** Like array, Functions, and user input/Output.
  + **Learning Outcome:** Reinforce of foundational knowledge of C programming

1. Develop and implement an online information

2. Generate more systematic and effective activities

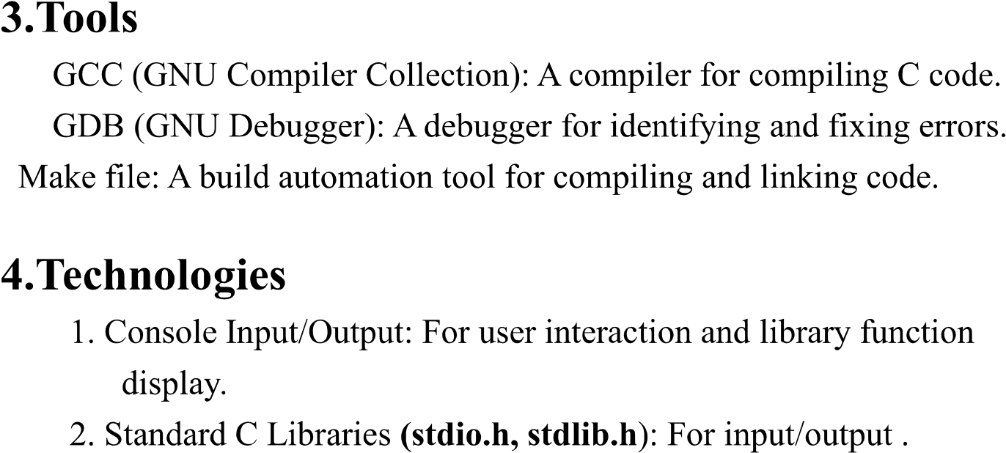
3. Simplify the entire member management process

****



**2. Integrated Development Environment (IDEs):**

Code Blocks: A free, open-source IDE for C/C++ Development



**#include <stdio.h>**

**#include <string.h>**

**// Define the maximum number of members the club can have**

**#define MAX\_MEMBERS 100**

**// Define the structure for a club member**

**struct Member {**

**int id;**

**char name[50];**

**int age;**

**char email[50];**

**};**

**// Declare the array of members**

**struct Member members[MAX\_MEMBERS];**

**int memberCount = 0;**

**// Function to add a new member**

**void addMember() {**

**if (memberCount >= MAX\_MEMBERS) {**

**printf("Sorry, the club is full. Cannot add more members.\n");**

**return;**

**}**

**printf("Enter Member ID: ");**

**scanf("%d", &members[memberCount].id);**

**getchar(); // To consume the newline character left by scanf**

**printf("Enter Member Name: ");**

**fgets(members[memberCount].name, sizeof(members[memberCount].name), stdin);**

**members[memberCount].name[strcspn(members[memberCount].name, "\n")] = '\0'; // Remove the newline at the end of the name**

**printf("Enter Member Age: ");**

**scanf("%d", &members[memberCount].age);**

**getchar(); // To consume the newline character left by scanf**

**printf("Enter Member Email: ");**

**fgets(members[memberCount].email, sizeof(members[memberCount].email), stdin);**

**members[memberCount].email[strcspn(members[memberCount].email, "\n")] = '\0'; // Remove the newline at the end of the email**

**memberCount++;**

**printf("Member added successfully!\n");**

**}**

**// Function to display all members**

**void displayMembers() {**

**if (memberCount == 0) {**

**printf("No members in the club.\n");**

**return;**

**}**

**printf("Club Members List:\n");**

**for (int i = 0; i < memberCount; i++) {**

**printf("ID: %d\n", members[i].id);**

**printf("Name: %s\n", members[i].name);**

**printf("Age: %d\n", members[i].age);**

**printf("Email: %s\n", members[i].email);**

**printf("-------------------------\n");**

**}**

**}**

**// Function to search for a member by ID**

**void searchMember() {**

**int id;**

**printf("Enter Member ID to search: ");**

**scanf("%d", &id);**

**for (int i = 0; i < memberCount; i++) {**

**if (members[i].id == id) {**

**printf("Member found!\n");**

**printf("ID: %d\n", members[i].id);**

**printf("Name: %s\n", members[i].name);**

**printf("Age: %d\n", members[i].age);**

**printf("Email: %s\n", members[i].email);**

**return;**

**}**

**}**

**printf("No member found with ID: %d\n", id);**

**}**

**// Function to delete a member by ID**

**void deleteMember() {**

**int id;**

**printf("Enter Member ID to delete: ");**

**scanf("%d", &id);**

**for (int i = 0; i < memberCount; i++) {**

**if (members[i].id == id) {**

**for (int j = i; j < memberCount - 1; j++) {**

**members[j] = members[j + 1];**

**}**

**memberCount--;**

**printf("Member with ID %d has been deleted.\n", id);**

**return;**

**}**

**}**

**printf("No member found with ID: %d\n", id);**

**}**

**// Function to display the menu**

**void displayMenu() {**

**printf("\nClub Management System\n");**

**printf("1. Add Member\n");**

**printf("2. Display All Members\n");**

**printf("3. Search Member\n");**

**printf("4. Delete Member\n");**

**printf("5. Exit\n");**

**}**

**int main() {**

**int choice;**

**// Main loop to keep showing the menu until the user chooses to exit**

**do {**

**displayMenu();**

**printf("Enter your choice: ");**

**scanf("%d", &choice);**

**switch (choice) {**

**case 1:**

**addMember();**

**break;**

**case 2:**

**displayMembers();**

**break;**

**case 3:**

**searchMember();**

**break;**

**case 4:**

**deleteMember();**

**break;**

**case 5:**

**printf("Exiting the system...\n");**

**break;**

**default:**

**printf("Invalid choice. Please try again.\n");**

**}**

**} while (choice != 5);**

**return 0;**

**}**

**How it works:**

1. Add Member: Adds a new member into the club by capturing the member's ID, name, age, and email.

2. Display All Members: List all the details of the club members.

3. Search Member: It allows the user to search for a member by his/her ID and shows the member's details.

4. Delete Member: It allows the user to delete a member by his ID.

5. Exit: This ends the program.

**About:**

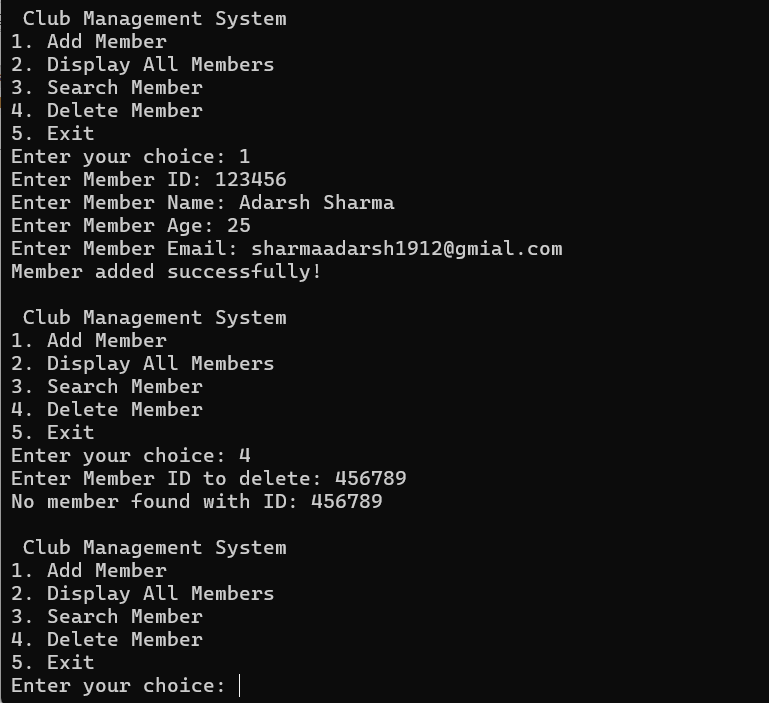
It can support a maximum number of MAX\_MEMBERS that is 100 in this case.

The function fgets is used for reading strings like name and email to avoid the problem of white spaces in names.

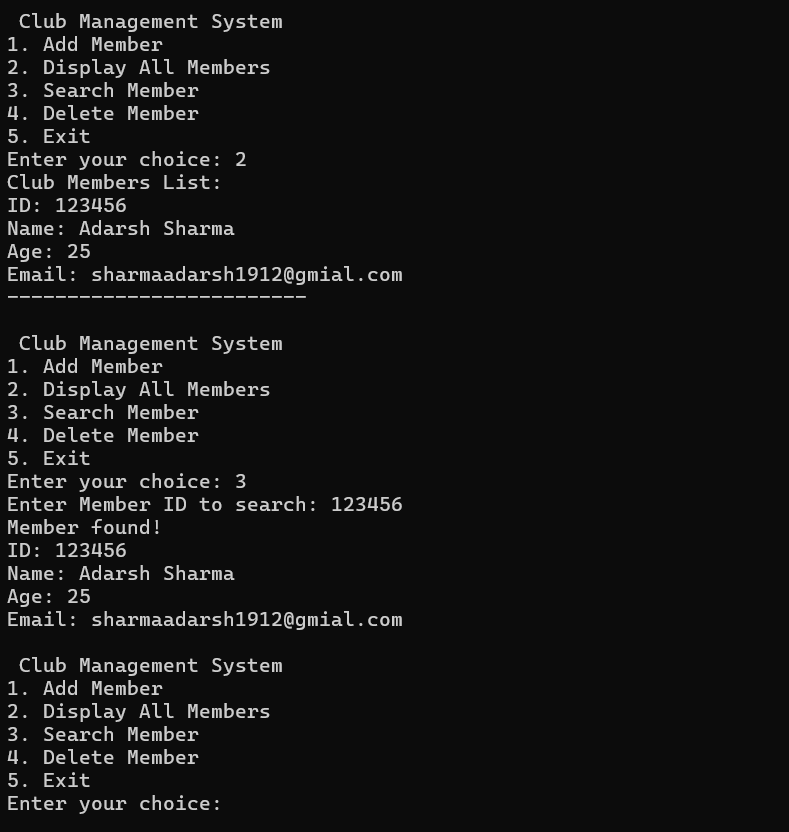
Using basic printf and scanf to communicate with the user, the system is short and easy to understand:

You can increase this code by adding features like updating member details or adding activities for members.

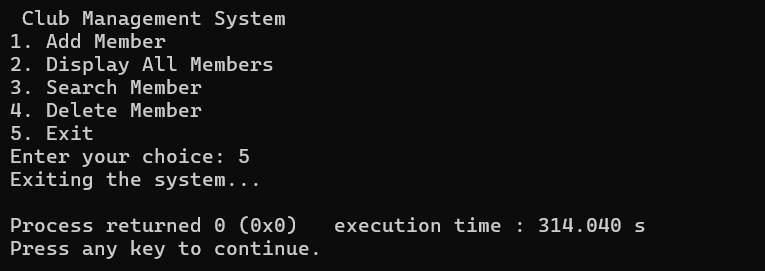
OUTPUT

****

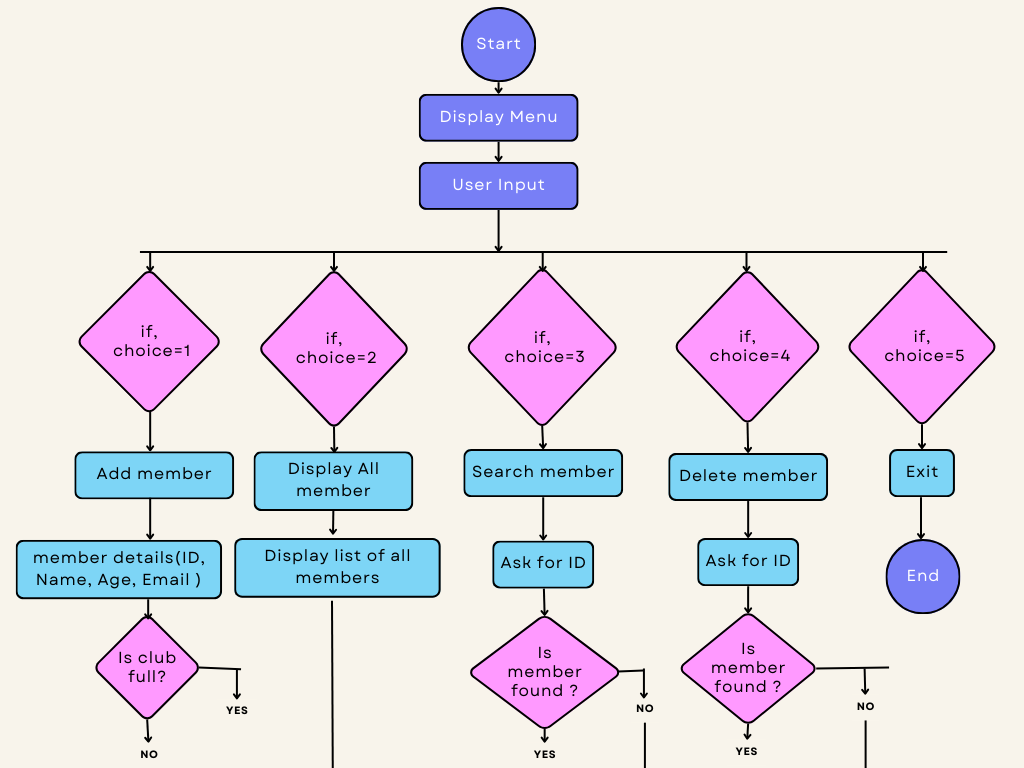
**Image(i)**

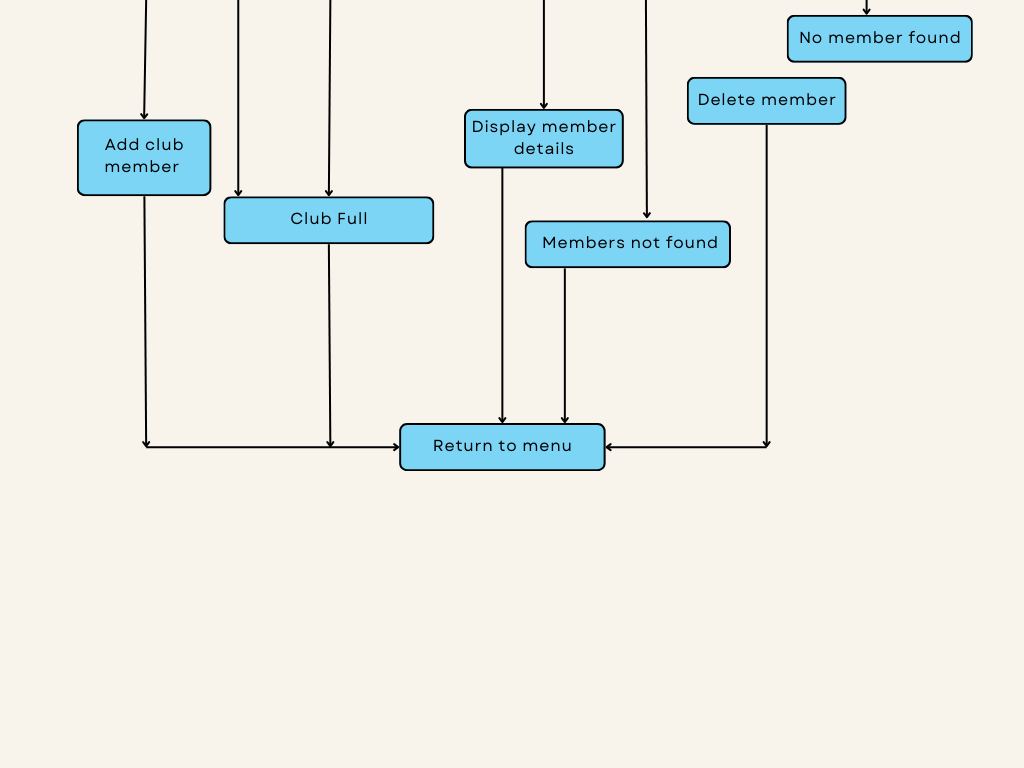
****

**Image(ii)**

****

**Image(iii)**

****

****

**CONCLUSION**

In conclusion, following the expectations and the stipulated budget and time frames, it is essential to make a note that the XYZ Project proved to be successful. By means of effective resource management and stakeholder engagement, the team was able to deliver a high-quality product that met their client's needs. Important milestones were reached; these included the completion of the initial phases ahead of schedule, which allowed for additional testing and refinement.

They encountered various challenges, chiefly with regard to communication and risk management. Those issues brought home the need for clearer protocols and regular check-ins between team members and stakeholders.

It is recommended to implement an altogether more robust risk assessment framework and enhance communication strategies so as to make it easier for all team members to juggle between controlled delivery and being informed. Those lessons will be a guide to better practices on the next project and are likely to bake in quite a number of lessons for future projects.

**Future Scope:**

Future Enhancement

* Collaboration and Communication Tools
* Personalized User Experience
* Cybersecurity Enhancements
* IT Management System Architect
* Cybersecurity Specialist

Overall, the future of IT management systems is expected to be highly dynamic, with a strong focus on leveraging advanced technologies to deliver enhanced operational efficiency, data-driven decision making, and improved overall business agility.