SRM GYM MANAGEMENT SYSTEM

Project submitted to the SRM University–AP, Andhra Pradesh

for the partial fulfillment of the requirements to award the degree of

**Bachelor of Technology**

In

**Computer Science and Engineering School of Engineering and Sciences**



Submitted by

**G Sai Kartheek (AP21110011447)**

**Y Sai Sravani (AP21110011432)**

**P G V L Kundanaa (AP21110011459)**

**M Manoj Kumar (AP21110011472)**

Under the Guidance of

**Bhaskara Santhosh**

**SRM University–AP**

**Neerukonda, Mangalgiri, Guntur Andhra Pradesh – 522 240**

**[December, 2022]**

# Certificate

This is to certify that the work present in this Project entitled “**SRM GYM MANAGEMENT SYSTEM”** has been carried out by **Group-12**(**Sai Kartheek, Sai Sravani, PGVL Kundanaa, Manoj Kumar)** under my/our supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor of Technology/Master of Technology in **School of Engineering and Sciences**.

**Supervisor**

(Signature)

Mr. E Bhaskara Santhosh

Assistant Professor, CSE Department, SRM UNIVERSITY, AP.

# Acknowledgements

I'd like to thank our teacher, Bhaskara Santhosh sir, for giving us the opportunity to work on this project. This project taught us a lot about different core concepts of Object-Oriented Programming, such as data encapsulation and abstraction. Inheritance, objects and classes etc. We are grateful to him for his assistance and guidance in completing this project. Finally, I'd like to thank all of my dear teammates for being a part of the team and for assisting me in completing this project.

Group 12(CSE-U)

# Table of Contents:

[Certificate 3](#_TOC_250009)

[Acknowledgements 5](#_TOC_250008)

[Table of Contents 7](#_TOC_250007)

[Abstract 9](#_TOC_250006)

[Statement of Contributions 11](#_TOC_250005)

1. [Introduction 13](#_TOC_250004)
2. [Methodology 15](#_TOC_250003)
   1. [DESIGN 15](#_TOC_250002)
   2. IMPLEMENTATION 16
3. [Results 24](#_TOC_250001)
4. [Concluding Remarks 32](#_TOC_250000)

# Abstract

SRM GYM MANAGEMENT SYSTEM deals with the development of a Gym Management System for SRM University. This software will handle the entire data of the Gym. It makes the work of a Fitness Manager very easy to keep track of each and every record. The gym management system is a simple console application with the system strictly protected by password information. In this project, the manager can do a lot of tasks in the system that he can add a new member, see member list, and remove permanent member details. The system has all the Member functions to list down the details of gym members. That can help you manage your business efficiency. Using this system even the member can join himself in the gym and the trainers gets their list in their console. The SRM GYM MANAGEMENT SYSTEM makes the work of members, trainers and managers easier.

# Statement of Contributions

**Work contributed by team members:**

Sai Kartheek - Login page Code, Report

Sai Sravani - Admin Code, files

Manoj Kumar - Member Code, Report

PGVL Kundanaa - Trainer Code, files

# Introduction

Our SRM Gym Management System is a gym management system. You can keep records on your members, their memberships, and have quick and easy communication between you and your members. Gym Management also includes a booking system, point of sale, banking, concessions and has a range of reports that help in the management of your gym. Our Gym Management System is a complete gym and recreation facility system program which looks after all of your members, memberships and activities. It is designed for gyms. Our gym management System provides lots of functions such data entry of member, keeping records of all the things about member’s fees, plan, and physical fitness which helps to provide good quality of services to customer from Gym managers. In this proposed system also provide the total information about machinery and data of trainers is also stored in it. Services provided by Gym are also handled by this system.

**2.1 DESIGN**

The Main Menu has 4 operations:

1) User

2) Admin

3) Trainer

4) Exit

The User Menu has 4 operations:

1)Join gym

2)Quit gym

3)Edit your profile

4)Exit

The Admin Menu has 6 operations:

1) Create a member

2) Display all records

3) Search a particular record

4) Edit a record

5) Delete a member

6) Exit

The trainer Menu has 3 operations:

1) Trainer 1

2) Trainer 2

3) Exit

The user, admin, trainer has separate logins where admin is password protected.

EXIT operation exits from the program and displays group members names.

Few hexadecimal codes are used for better appearance of output.

Classes, Files and Functions are used for the implementation of Code.

**2. Implementation**

**Source code:**

#include <fstream>

#include <iomanip>

#include <iostream>

#include <cstring>

#include <cstdlib>

using namespace std;

void exitfun(){

    system("cls");

    cout<<"\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb  Thanks for Using SRM GYM MANAGEMENT SYSTEM  \xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"<<endl

        <<"\n\n\t\t\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2 Team Members \xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2 "

        <<"\n\n\t\t\t\x4  Sai Kartheek"

        <<"\n\t\t\t\x4  Sai Sravani"

        <<"\n\t\t\t\x4  Manoj Kumar"

        <<"\n\t\t\t\x4  Lakshmi kundanaa\n\n\n\n\n\n\n\n\n\n";

}

class alogin

{

    string admin;

public:

    int adlogin()

    {

        cout << "Enter Admin Code:" << endl;

        cin >> admin;

        if (admin == "admin")

        {

            int p = getadpassword();

            if (p == 1)

            {

                cout << "Welcome !!!!" << endl;

                return 1;

            }

        }

        else

        {

            cout << "Incorrect Admin Code !!! " << endl;

            adlogin();

            cout << "Login Failed" << endl;

            return 0;

        }

        return 0;

    }

    int getadpassword();

};

int alogin::getadpassword()

{

    string pw;

    cout << "\nEnter Password:" << endl;

    cin >> pw;

    if (pw == "SSMK")

    {

        cout << "Login Successful" << endl;

        return 1;

    }

    else

    {

        cout << "Entered Password is Incorrect" << endl;

        getadpassword();

        return 0;

    }

}

class member

{

    int member\_number, j;

    char mem\_name[50], classs[50], timings[50], mem\_phonenum[11];

    float fee;

    long long int contact;

public:

    int time\_slots()

    {

        int k;

        cout << "\nplease select your preferred timings\npress 1 for: morining 6-7\npress 2 for: morning 7-8\npress 3 for:morning 8-9\npress 4 for: evening 4-5\npress 5 for:evening 5-6\npress 6 for:evening 6-7\n";

        cin >> k;

        switch (k)

        {

        case 1:

            strcpy(timings, "6AM-7AM");

            break;

        case 2:

            strcpy(timings, "7AM-8AM");

            break;

        case 3:

            strcpy(timings, "8AM-9AM");

            break;

        case 4:

            strcpy(timings, "4PM-5PM");

            break;

        case 5:

            strcpy(timings, "5PM-6PM");

            break;

        case 6:

            strcpy(timings, "6PM-7PM");

            break;

        }

        return 0;

    }

    void create\_mem()

    {

        int k, l, j;

        cout << endl

             << "Please Enter The member Number: ";

        cin >> member\_number;

        cout << endl

             << "Please Enter The Name of The member: ";

        getchar();

        cin.getline(mem\_name, 50);

        cout << endl

             << "Please Enter The contact number: ";

        cin >> contact;

        cout << "\n 1.Gold class\n\tGold Class \n\t1.Access to all equipments\n\t2.Personal Training\n\n 2.Silver class\n\tSilver Class features\n\t1.Access to limited equipments\n\t2.No Personal Training\n\n";

        cout << "Enter your Choice:";

        cin >> k;

        if(k==1)

        {

            cout<<"Trainer 1 alloted";

        }

        if(k==2)

        {

            cout<<"Trainer 2 alloted";

        }

        ofstream t1;

        if(k==1)

        {

        t1.open("d://trainer1.txt",ios::app);

        t1<<mem\_name<<endl;

        }

        ofstream t2;

        if (k==2){

            t2.open("d://trainer2.txt",ios::app);

            t2<<mem\_name<<endl;

        }

        switch (k)

        {

        case 1:

        {

            strcpy(classs, "gold");

            fee = 3000;

            cout << "\nyour monthly fee would be: " << fee << endl;

            //l = trainee\_allotment(classs);

        }

        break;

        case 2:

        {

            strcpy(classs, "silver");

            fee = 2000;

            cout << "\nyour monthly fee would be: " << fee << endl;

            //l = trainee\_allotment(classs);

        }

        break;

        }

        time\_slots();

    }

    void show\_mem()

    {

        cout << endl

             << "member code: " << member\_number;

        cout << endl

             << "Name: " << mem\_name;

        cout << endl

             << "category: " << classs;

        cout << endl

             << "fee: " << fee;

        cout << endl

             << "contact: " << contact;

        cout << endl

             << "timings: " << timings << endl;

    }

    int getmem()

    {

        return member\_number;

    }

    float getfee()

    {

        return fee;

    }

    char \*getName()

    {

        return mem\_name;

    }

    long long int getcontact()

    {

        return contact;

    }

};

fstream fp;

member m1;

void save\_member()

{

    fp.open("d://list\_of\_members.txt", ios::app);

    m1.create\_mem();

    fp.write((char \*)&m1, sizeof(m1));

    fp.close();

    cout << endl

         << endl

         << "The member has been successfully added\n\n ";

    getchar();

}

void show\_all()

{

    system("cls");

    cout << endl

         << "\t\t...RECORDS...\n";

    fp.open("d://list\_of\_members.txt", ios::in);

    while (fp.read((char \*)&m1, sizeof(m1)))

    {

        m1.show\_mem();

        getchar();

    }

    fp.close();

}

void display\_record(int num)

{

    bool found = false;

    fp.open("d://list\_of\_members.txt", ios::in);

    while (fp.read((char \*)&m1, sizeof(m1)))

    {

        if (m1.getmem() == num)

        {

            system("cls");

            m1.show\_mem();

            found = true;

        }

    }

    fp.close();

    if (found == false)

        cout << "\n\nNo record found\n\n";

    getchar();

}

void edit\_member()

{

    int num;

    bool found = false;

    // system("cls");

    cout << endl

         << endl

         << "\tPlease Enter The member number: ";

    cin >> num;

    fp.open("d://list\_of\_members.txt", ios::in | ios::out);

    while (fp.read((char \*)&m1, sizeof(m1)) && found == false)

    {

        if (m1.getmem() == num)

        {

            m1.show\_mem();

            cout << "\nPlease Enter The New details of the member: " << endl;

            m1.create\_mem();

            int pos = 0;

            fp.seekp(pos, ios::cur);

            fp.write((char \*)&m1, sizeof(m1));

            cout << endl

                 << endl

                 << "\t Record Successfully Updated...\n\n";

            found = true;

        }

    }

    fp.close();

    if (found == false)

        cout << endl

             << endl

             << "Record Not Found...\n\n";

    getchar();

}

void delete\_member()

{

    int num;

    system("cls");

    cout << endl

         << endl

         << "Please Enter The member number: ";

    cin >> num;

    fp.open("d://list\_of\_members.txt", ios::in | ios::out);

    fstream fp2;

    fp2.open("d://Temp.txt", ios::out);

    fp.seekg(0, ios::beg);

    while (fp.read((char \*)&m1, sizeof(m1)))

    {

        if (m1.getmem() != num)

        {

            fp2.write((char \*)&m1, sizeof(m1));

        }

    }

    fp2.close();

    fp.close();

    remove("d://list\_of\_members.txt");

    rename("d://Temp.txt", "d://list\_of\_members.txt");

    cout << endl

         << endl

         << "\tRecord Deleted...\n\n";

    getchar();

}

void fnmanage()

{

    for (;;)

    {

        // system("cls");

        int option;

        cout << "\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

        cout << "\n\tPress 1 to CREATE MEMBER";

        cout << "\n\tPress 2 to DISPLAY ALL RECORDS";

        cout << "\n\tPress 3 to SEARCH FOR A PARTICULAR RECORD ";

        cout << "\n\tPress 4 to EDIT MEMBER DETAILS";

        cout << "\n\tPress 5 to DELETE MEMBER";

        cout << "\n\tPress 6 to Exit";

        cout << "\n\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

        cout << "\n\n\tOption: ";

        cin >> option;

        switch (option)

        {

        case 1:

            system("cls");

            save\_member();

            break;

        case 2:

            system("cls");

            show\_all();

            break;

        case 3:

            int num;

            system("cls");

            cout << "\n\n\tPlease Enter The member Number: ";

            cin >> num;

            display\_record(num);

            break;

        case 4:

            system("cls");

            edit\_member();

            break;

        case 5:

            system("cls");

            delete\_member();

            break;

        case 6:

            system("cls");

            exitfun();

            exit(0);

        default:

            system("cls");

            fnmanage();

        }

    }

}

void fnuser()

{

    for (;;)

    {

        int m;

        cout << "\n1.join gym\n2.quit gym\n3.edit your profile\n4.Exit";

        cout << endl

             << "enter your choice" << endl;

        cin >> m;

        switch (m)

        {

        case 1:

            system("cls");

            save\_member();

            break;

        case 2:

            delete\_member();

            break;

        case 3:

            edit\_member();

            break;

        case 4:

            exitfun();

            exit(0);

        }

    }

}

void fntrainer()

{

    ifstream t1;

    ifstream t2;

    for (;;)

    {

        int tom;

        cout << endl<<"1.trainer1\n2.trainer2\n3.Exit\n";

        cout << endl

             << "enter your trainer number" << endl;

        cin >> tom;

        switch(tom)

        {

        case 1: cout<<"You are alloted to the ones who chosen gold membership"<<endl;

                   t1.open("d://trainer1.txt",ios::in);

                   if(t1.is\_open()){

                string tr;

                while(getline(t1,tr)){

                    cout<<tr<<"\n";

                }

            }

                  // cout<<mem\_name;

                   break;

        case 2: cout<<"You are alloted to the ones who chosen silver membership"<<endl;

            t2.open("d://trainer2.txt",ios::in);

            if(t2.is\_open()){

                string tr2;

                while(getline(t2,tr2)){

                    cout<<tr2<<"\n";

                }

            }

            //cout<<mem\_name;

            break;

            t1.close();

            t2.close();

       case 3:

           exitfun();

           exit(0);

        }

    }

}

int main()

{

    system("color F");

    int i, k;

    string name, code;

    cout << "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2 Welcome To SRM Gym Management System \xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2" << endl;

    cout << endl

         << "SELECT MODE\n" << endl

         << "1.User" << endl

         << "2.Admin" << endl

         << "3.Trainer" << endl

         << "4.Exit\n\n"<<endl

         << "Enter Your Choice:\n";

    cin >> i;

    if (i == 1)

    {

        system("cls");

        system("color F");

        cout << endl

             << "you're in user mode" << endl;

        fnuser();

    }

    if (i == 2)

    {

        system("cls");

        system("color F");

        cout << endl

             << "you're in admin mode" << endl;

        alogin ad;

        k = ad.adlogin();

        if (k == 1)

        {

            fnmanage();

        }

        else

        {

            exit(0);

        }

    }

    if (i == 3)

    {

        system("cls");

        system("color F");

        cout << endl

             << "you're in trainers mode" << endl;

        fntrainer();

    }

    if (i == 4)

    {

        system("cls");

        system("color F");

        exitfun();

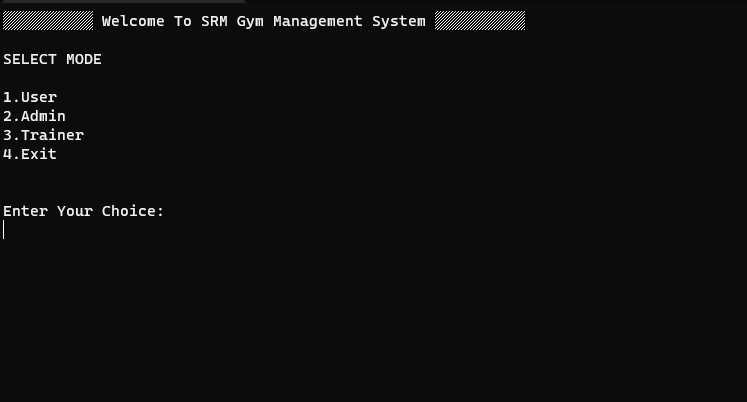
    }

    return 0;

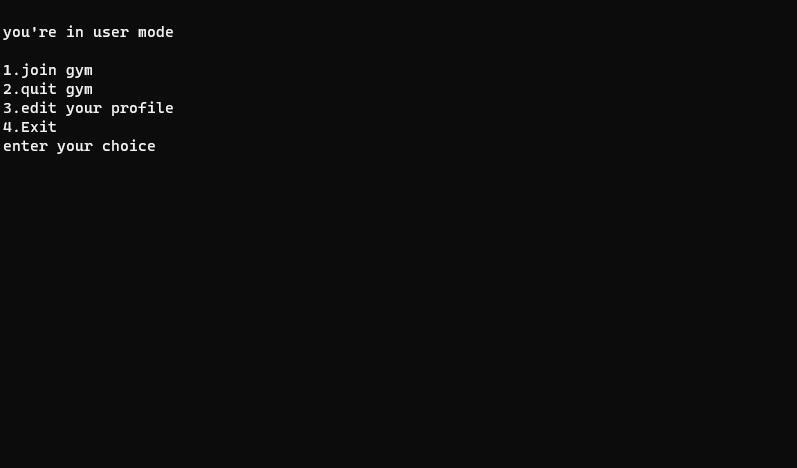
}

**3.Results**

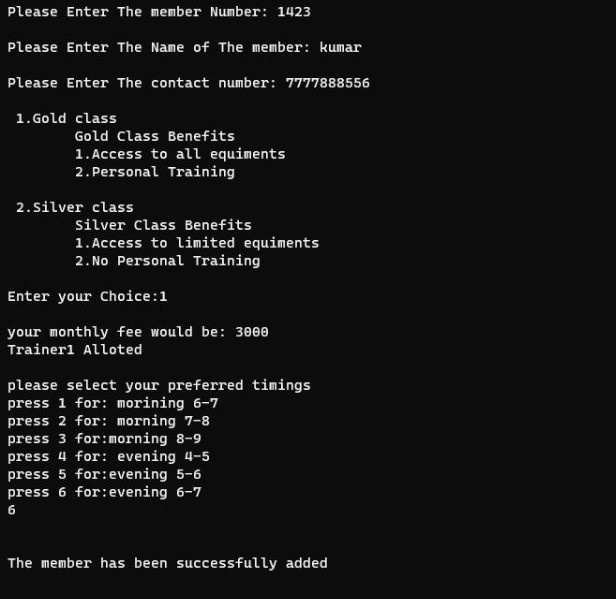
* The Main Menu is displayed as below.



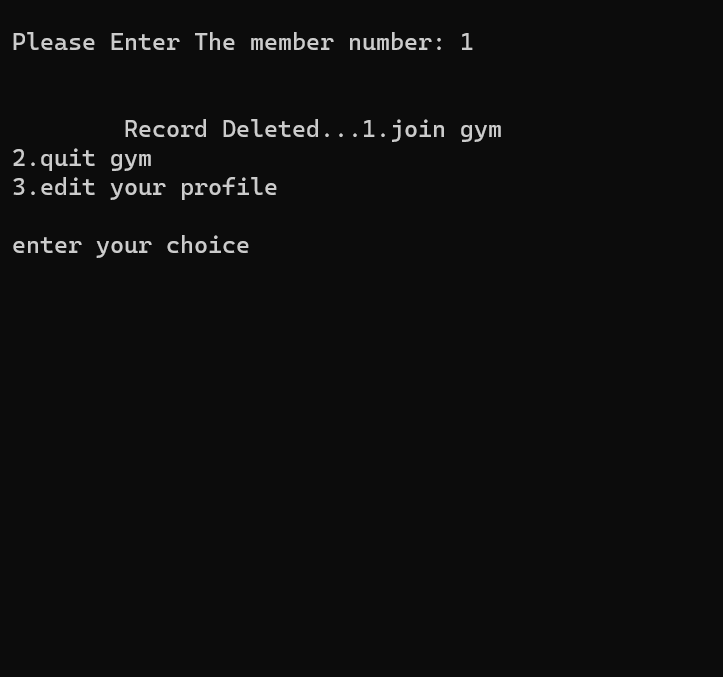
* After entering Choice as 1 the User Menu is displayed.

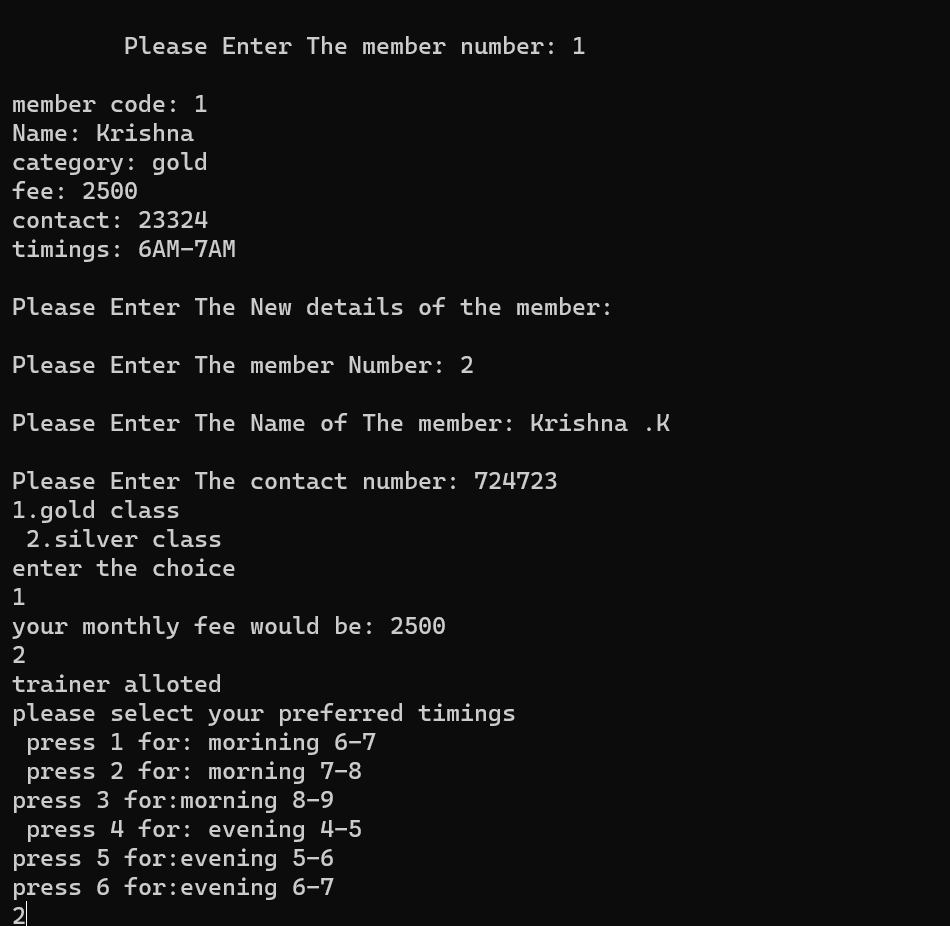


**JOIN GYM**



**QUIT GYM**

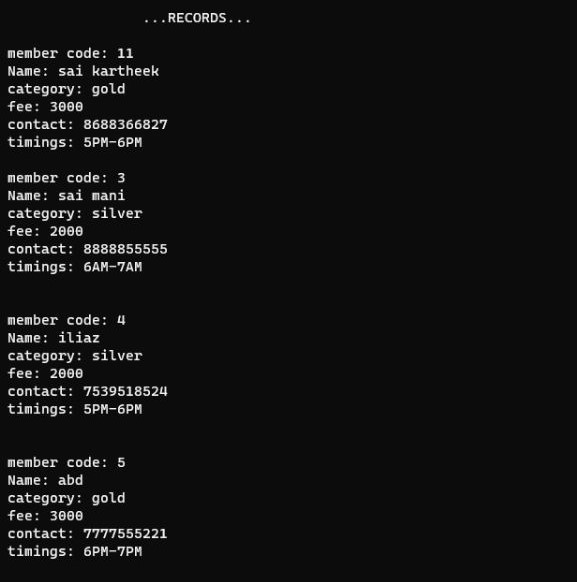


**3.EDIT PROFILE**

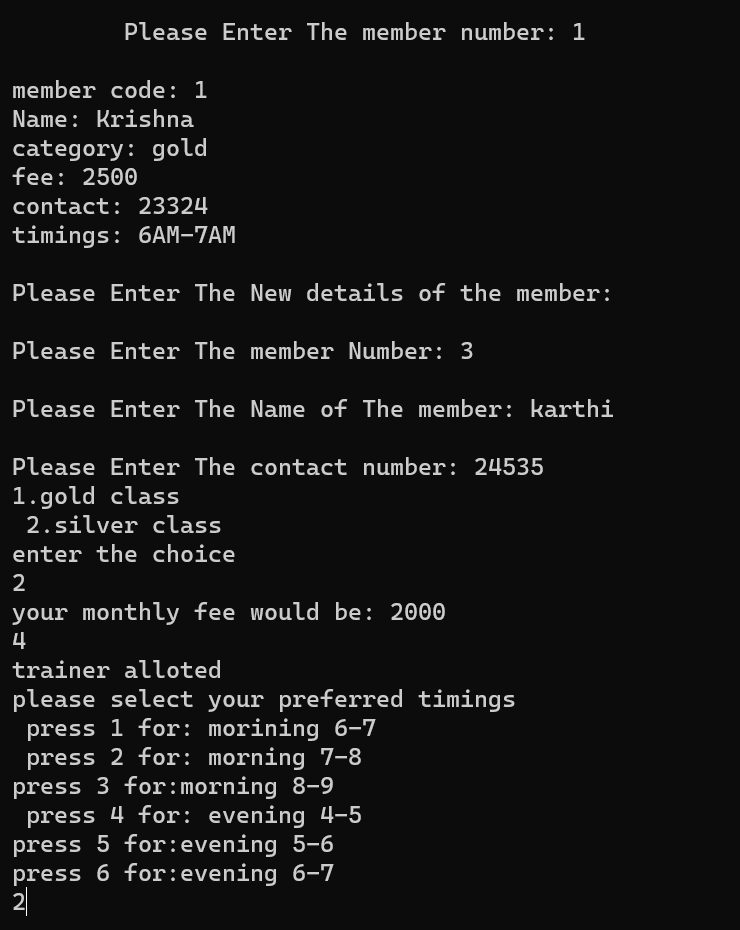
**ADMIN MODE**



**DISPLAY RECORDS**



**CREATE MEMBER**

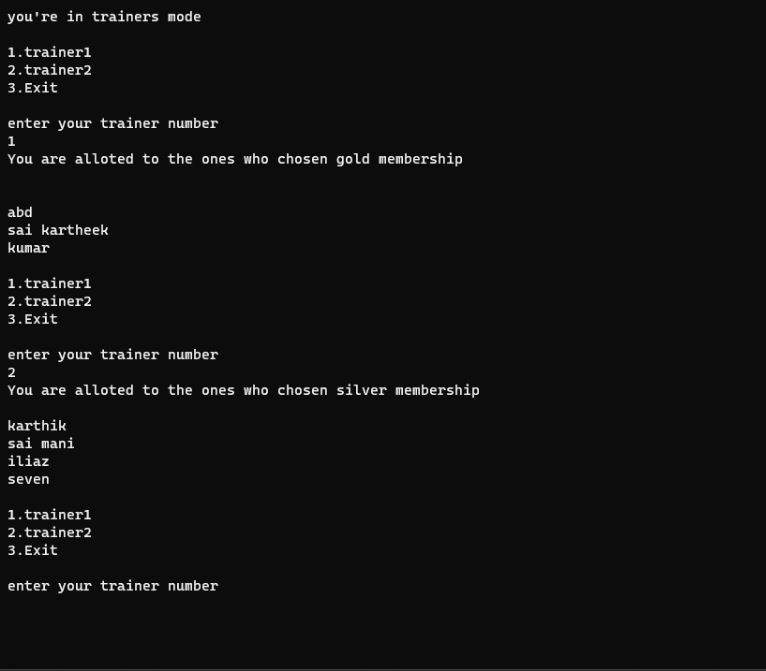


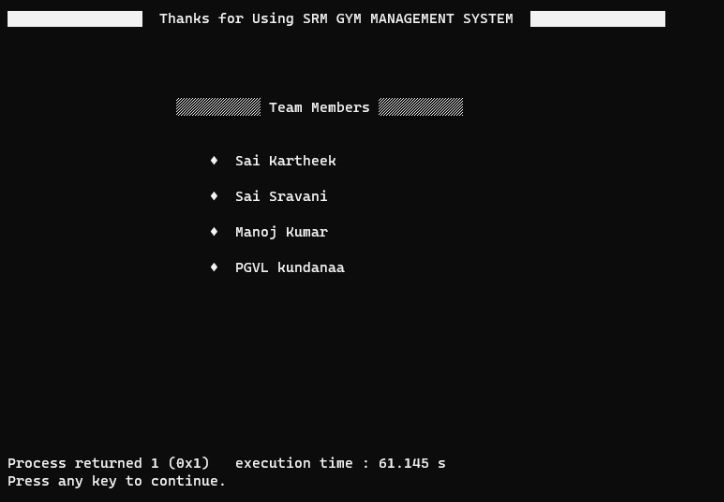
**SEARCH FOR A PARTICULAR RECORD**

****

****

**TRAINER MODE**



**EXIT OPTION**

# Concluding Remarks:

In the process of completing this project, we have understood the oops concept and learned more about classes and inheritance. By this project they can easily get membership in gym and even the gym manager can access all the members and trainer's details. At last, we conclude that due to this project user can easily keep records of gym and can build and manage the gym management system successfully….