MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE

(A Govt Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

Soft Skill Project Report

on

Gym Management System

Submitted By:
Shanky bhilwar
(0901CA211056)

Mentor:
Dr. Anshu Chaturvedi
(Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE
Gwalior – 474005(MP) est.1957

July – December 2021

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE

(A Govt Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

<u>CERTIFICATE</u>

This is certified that **Shanky bhilwar(0901CA211056)** has submitted the project report titled **Gym Management System** under the mentorship of **Dr. Anshu Chaturvedi** (Professor), as the skills based mini project in 1st year of Master of Computer Application in Computer Science and Engineering from **Madhav Institute of Technology and Science, Gwalior**.

Dr. Anshu Chaturvedi
(Professor)
Computer Science and Engineering

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE
(A Govt Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

DECLARATION

I hereby declare that the work being presented in this project report, for the fulfilment of partial requirement of the skills based mini project in 1st year of Master of Computer Application in Computer Science and Engineering at **Madhav Institute of Technology & Science, Gwalior** is an authenticated and original record of my work under the mentorship of **Dr. Anshu Chaturvedi**, (Professor), MITS Gwalior.

I declare that I have not submitted the matter embodied in this report anywhere else.

.

Shanky Bhilwar
0901CA211056
2021-2023 Year,
Master of Computer Application,
Computer Science and Engineering

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE
(A Govt Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

ACKNOWLEDGEMENT

The full semester project has proved to be pivotal to my career. I am thankful to my institute, **Madhav Institute of Technology and Science** to allow me to continue my disciplinary project. I extend my gratitude to the Director of the institute, **Dr. R. K. Pandit** and Dean Academics, **Dr. Manjaree Pandit** for this.

I would sincerely like to thank my department, **Department of Computer Science and Engineering**, for allowing me to explore this project. I humbly thank **Dr. Manish Dixit**, Professor and Head, Department of Computer Science and Engineering, for his continued support during the course of this engagement, which eased the process and formalities involved.

I am sincerely thankful to my faculty coordinator. I am grateful to the guidance of **Dr. Anshu Chaturvedi**, (Professor), Computer Science and Engineering, for her continued support and guidance throughout the project. I am also very thankful to the faculty and staff of the department.

Shanky Bhilwar
0901CA211056
2021-2023 Year,
Master of Computer Application,
Computer Science and Engineering

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE

ABSTRACT

In many Gyms, It is hard to manage all the records of all the customers on pen and paper mode. By using this application, we can save all details of customers and their plans. It is also easy for Gym owners to get all details of all customers in single place.

So this project can be helpful for both gym owners as well as for gym members. In this application all receipts are store in a digital format, so there are no issues of losing any confidential receipts. This application will also notify the user (gym members) about their fees and also notifies the gym owner about the payment clearance. This application in future can be elaborated by providing supplement store, diet information, personal training etc.

CONTENTS

INTRODUCTION

Gym Management System is a application that handles the entire data of the Gym. It makes the work of a Fitness Manager very easy instead of writing data in a notebook. In past, the gym managers were using notebooks to write the data along with customers' names who purchased the gym memberships. So it was very difficult to keep track of each and every record.

If a Gym Manager wants to search for a particular gym record then that task was very time-consuming. We have created separate logins for the users and the gym managers i.e. admin, in which the gym manager is password protected.

In this project, the gym manager can add, update, delete and create gym memberships and can assign them to a particular user.The user can quit the gym anytime, whenever he or she wanted.The software gives facility for the users to edit their profiles.Only the admin can delete and edit the membership.During the purchase of membership, the trainer is allotted automatically.This project uses the concept of file handling.

Problem Defination

Existing system was manual. Time consuming as data entry which include calculations took lot of time. Searching was very complex as there could be 100's of entry every year.

The proposed system is expected to be faster than the existing system. The Project was made in order to effectively and efficiently cater to requirements of the fitness center. Very frequently the person who generally holds the tasks to manage the center needs to keep records of all the transactions as well as data manually.

Generally, In order to structure these tasks, Separate Registers are maintained. This whole process thus becomes quite cumbersome for them to control manually. Moreover, Any wrong data entered mistakenly can brings serious results. This Manually Managed system of the store was also heavily proned to data loss due to certain causes Misplacement of Registers, Destruction of Registers, Unauthorized access to registers etc. which can bring in disastrous Consequences.

Code:

```cpp
#include<fstream>
#include<cstring>
#include<iomanip>
#include<iostream>
#include<cstring>
#include<cstdlib>
usingnamespace std;
class alogin
{
    string admin;
    public:
    int adlogin()
    {
        cout<<"enter the admin code"<<endl;
        cin>>admin;
        if(admin=="admin")
        {
            int p=getadpassword();
            if (p==1)
            {
                cout<<"login succesful"<<endl;
                return1;
            }
        }
        else{
            cout<<"user name not found"<<endl;
            adlogin();
            cout<<"login failed"<<endl;
            return0;
        }
    }
    int getadpassword();
};
int alogin::getadpassword()
{
    string pw;
    cout<<"enter the password"<<endl;
    cin>>pw;
    if(pw=="shanky")
    {
        cout<<"WELCOME!!"<<endl;
        return1;
    }
    else{
        cout<<"entered password is incorrect"<<endl;
        getadpassword();
        return0;
    }
}
void allotment_gold(int gc,int sc)
{
    if(gc!=0)
    {
        cout<<gc<<endl;
        cout<<"trainer alloted\n";
    }
     else
     {
         cout<<"no trainer available\n";
     }
}
void allotment_silver(int gc,int sc)
{
    if(sc!=0)
```

```cpp
        {
            cout<<sc<<endl;
            cout<<"trainer alloted\n";
        }
         else
         {
            cout<<"\nno trainer available\n";
            if(gc!=0)
            {
                cout<<"enter  another class\n";
                allotment_gold(gc,sc);
            }
         }
}
int trainee_allotment(char* c)
{    char class_type[10];
     int i,gold_class=3,silver_class=5;
     strcpy(class_type,c);
                if((gold_class!=0) || (silver_class)!=0)
                    {
                        if(strcmp(class_type,"gold")==0)
                        {
                            gold_class--;
                            allotment_gold(gold_class,silver_class);
                        }
                        elseif(strcmp(class_type,"silver")==0)
                        {
                            silver_class--;
                            allotment_silver(gold_class,silver_class);
                        }
                        else
                        {
                            cout<<"no traianer availbale\n";
                            return1;
                        }
                    }
}
class member
    {
        int member_number,j;
        char mem_name[50], classs[50],timings[50];
        float fee;

    longint contact;
        public:
     int time_slots()
       {
            int k;
            cout<<"please select your preferred timings\n press 1 for: morining 6-7\n press 2 for:
morning 7-8\npress 3 for:morning 8-9\n";
            cout<<" press 4 for: evening 4-5\npress 5 for:evening 5-6\npress 6 for:evening 6-7\n";
            cin>>k;
            switch(k)
            {
                case1:strcpy(timings,"6AM-7AM");break;
                case2:strcpy(timings,"7AM-8AM");break;
                case3:strcpy(timings,"8AM-9AM");break;
                case4:strcpy(timings,"4PM-5PM");break;
                case5:strcpy(timings,"5PM-6PM");break;
                case6:strcpy(timings,"6PM-7PM");break;
            }
}
        void create_mem()
        {   int k,l,j;
            cout<<endl<<"Please Enter The member Number: ";
                cin>>member_number;
```

```cpp
                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<<"1.gold class\n 2.silver class\n";
        cout<<"enter the choice\n";
        cin>>k;
        switch (k)
        {
            case1:{strcpy(classs,"gold");
                fee=2500;
                cout<<"your monthly fee would be: "<<fee<<endl;
                l=trainee_allotment(classs);
                }
                break;
            case2:{strcpy(classs,"silver");
                fee=2000;
                cout<<"your 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;
        }
        float getcontact()
        {
            return contact;
        }
};
fstream fp;
member m1;
void save_member()
{
    fp.open("newdata2.dat",ios::out|ios::app);
    m1.create_mem();
    fp.write((char*)&m1,sizeof(m1));
    fp.close();
    cout<<endl<<endl<<"the member has been succesfully added...";
    getchar();
}
void show_all()
{
    system("cls");
    cout<<endl<<"\t\tRECORDS...";
    fp.open("newdata2.dat",ios::in);
```

```cpp
        while(fp.read((char*)&m1,sizeof(m1)))
        {
            m1.show_mem();
            getchar();
        }
        fp.close();

    }
     void display_record(int num)
    {
        bool found=false;
        fp.open("newdata2.dat",ios::in);
        while(fp.read((char*)&m1,sizeof(m1)))
        {
            if(m1.getmem()==num)
            {
                system("cls");
                m1.show_mem();
                found=true;
            }
        }
        fp.close();
        if(found == true)
        cout<<"\n\nNo record found";
        getchar();
    }
     void edit_member()
    {
        int num;
        bool found=false;
        system("cls");
        cout<<endl<<endl<<"\tPlease Enter The member number: ";
        cin>>num;
        fp.open("newdata2.dat",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=1*sizeof(m1);
                fp.seekp(pos,ios::cur);
                fp.write((char*)&m1,sizeof(m1));
                cout<<endl<<endl<<"\t Record Successfully Updated...";
                found=true;
            }
        }
        fp.close();
        if(found==false)
            cout<<endl<<endl<<"Record Not Found...";
        getchar();
    }
    void delete_member()
    {
        int num;
        system("cls");
        cout<<endl<<endl<<"Please Enter The member number: ";
        cin>>num;
        fp.open("newdata2.dat",ios::in|ios::out);
        fstream fp2;
        fp2.open("Temp2.dat",ios::out);
        fp.seekg(0,ios::beg);
        while(fp.read((char*)&m1,sizeof(m1)))
        {
            if(m1.getmem()!=num)
            {
```

```cpp
                fp2.write((char*)&m1,sizeof(m1));
            }
        }
        fp2.close();
        fp.close();
        remove("newdata2.dat");
        rename("Temp2.dat","newdata2.dat");
        cout<<endl<<endl<<"\tRecord Deleted...";
        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 GO BACK TO MAIN MENU";
        cout<<"\n\t*******************************************";
        cout<<"\n\n\tOption: ";
        cin>>option;
        switch(option)
        {
            case1: system("cls");
                    save_member();
                    break;

            case2: show_all();
                    break;

            case3:
                    int num;
                system("cls");
                    cout<<"\n\n\tPlease Enter The member Number: ";
                    cin>>num;
                    display_record(num);
                    break;

            case4: edit_member();
                break;

            case5: delete_member();
                    break;

            case6: system("cls");
                    break;

            default:fnmanage();
            }
        }
    }
void fitness()
{
    {cout<<"displaying fitness tips..."<<endl;
        std::ifstream f("tips.txt");

        if (f.is_open())
            {
            std::cout << f.rdbuf();
            }}
}
void fnuser()
```

```cpp
{
    for(;;){
    int m;
    cout<<"1.join gym\n2.quit gym\n3.edit your profile\n";
    cout<<endl<<"enter your choice"<<endl;
    cin>>m;
    switch(m)
    {
        case1:system("cls");
                    save_member();
                    break;
            case2:delete_member();
                    break;
            //case 3:fitness();
            //  break;
            case3:edit_member();
                    break;
    }
}
}
int main()
{
        int i,k;
        string name,code;
        cout<<"Gym Management System Project in C++"<<endl;
        cout<<endl<<"SELECT MODE"<<endl<<"1.User"<<endl<<"2.Admin"<<endl;
        cin>>i;
            if(i==1){cout<<endl<<"you're in user mode"<<endl;
                    fnuser();
                    }
            if(i==2){cout<<endl<<"you're in admin mode"<<endl;
                    alogin ad;
                    k=ad.adlogin();
                    if(k==1)
                    {
                        fnmanage();
                    }
                    else
                    {
                        cout<<"you cannot access manageral details!";
                    }
                    }
        return0;
}
```

Output:

Main Screen:

```
Gym Management System Project in C++

SELECT MODE
1.User
2.Admin
```

When the user choice as 1 then the following screen will appear:

```
Gym Management System Project in C++

SELECT MODE
1.User
2.Admin
1

you're in user mode
1.join gym
2.quit gym
3.edit your profile

enter your choice
```

Admin Screen:

```
*********************************************
Press 1 to CREATE MEMBER
Press 2 to DISPLAY ALL RECORDS
Press 3 to SEARCH FOR A PARTICULAR RECORD
Press 4 to EDIT MEMBER DETAILS
Press 5 to DELETE MEMBER
Press 6 to GO BACK TO MAIN MENU
*********************************************

Option:
```

Displaying all record:

```
                RECORDS...
member code: 56
Name: Shankk
category: gold
fee: 2500
contact: 56655665
timings: 8AM-9AM

member code: 56
Name: Shanky Bhilwar
category: gold
fee: 2500
contact: 895565365
timings: 7AM-8AM
```

Conclusion:

Since we are entering details of the customer's electronically in the "Gym Management System", data will be secured. Using this application, we can retrieve a member's history with a single click. Thus, processing information will be faster. It guarantees accurate maintenance of member's details. It easily reduces the book keeping task and thus reduces the human effort and increases accuracy speed. It reduces complexity.

Biblography:

https://www.academia.edu/26750275/Project_report_on_gym_management_system_project

https://studylib.net/doc/25661677/gym-management-system-project-report--1-