**PURBANCHAL UNIVERSITY**

**Biratnagar, Nepal**

****

A Project report on

**“GYM MANAGEMENT SYSTEM”**

In the partial fulfillment for the requirement of the 2nd Semester Project-II (subject code-BIT156C0) in the completion of **Bachelor of Information Technology (BIT)** degree at **KIST college** **of Information Technology**, under **Purbanchal University.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Submitted By:-** | | |  | **Submitted To:-** | |
| **1)** |  | **Manish Dhungana** |  |  |  |  |
| **2)** |  | **Rohan Mishra** | | **Purbanchal University** | | |

1. **Santosh Yadav**
2. **Suyog Rai**

**Under the Guidance of**

**Mr. Deepak Khadka**

**Lecturer, BIT**

**KIST COLLEGE OF INFORMATION AND TECHNOLOGY KAMALPOKHARI, KATHMANDU NEPAL**

**KIST COLLEGE OF INFORMATION AND TECHNOLOGY KAMALPOKHARI, KATHMANDU NEPAL**

****

**CERTIFICATE**

This is to certify that the project work entitled **“GYM MANAGEMENT SYSTEM”** is carried out by **MANISH DHUNGANA (5400), ROHAN MISHRA(5423), SANTOSH YADAV(5438), SUYOG RAI (5401),** bona fide students of **KIST COLLEGE OF INFORMATION AND TECHNOLOGY** in partial fulfillment for the award of **BACHELOR IN INFORMATION AND TECHNOLOGY** of the **PURBANCHAL UNIVERSITY, BIRATNAGAR NEPAL**, during the year **2021-2022**. It is certified that all corrections indicatedfor internal assessment have been incorporated in the report submitted in the department library. The project report has been approved, as it satisfied the academic requirements in respect of the project work prescribed for the said degree.

The details of the students are as follows:-

|  |  |  |
| --- | --- | --- |
| **NAME** | **REGISTRATION NO.** | **SYMBOL NO.** |
| Manish Dhungana | 058-3-2-04723-2020 | 324620 |
| Rohan Mishra | 058-3-2-04734-2020 | 324631 |
| Santosh Yadav | 058-3-2-04740-2020 | 313437 |
| Suyog Rai | 058-3-2-04748-2020 | 324645 |

Course Semester: - 2nd Semester

Subject: - Project-II

Subject Code: - BIT 156 C0



Mr. Deepak Khadka

Program Coordinator, BIT



**KIST COLLEGE OF INFORMATION AND TECHNOLOGY KAMALPOKHARI, KATHMANDU**

**Examiner’s Certification**

The Project Report

On

**“GYM MANAGEMENT SYSTEM”**

**Developed by**

**Manish Dhungana**

**Rohan Mishra**

**Santosh Yadav**

**Suyog Rai**

Is approved and is acceptable in qualify form.

**Internal Examiner**

Name:

Designation:

**External Examiner**

Name:

Designation:

**ACKNOWLEDGEMENT**

It is with greatest satisfaction and euphoria that we are submitting our project report entitled **“GYM MANAGEMENT SYSTEM”.** We have completed it as a part of the curriculum of **PURBANCHAL UNIVERSITY.**

We also take this opportunity to express a deep sense of gratefulness to our **BIT Coordinator** and **Project teacher MR. Deepak Khadka** for his amiable support, valuableinformation and guidance which helped us in completing this task throughout its various stages. We are indebted to all members of **KIST College,** for the valuable support and suggestion provided by them using their specific fields’ knowledge. We are grateful for their cooperation during the period of our project.

Finally, we would also like to express our gratefulness towards **Purbanchal University** for designing such a wonderful course structure. It will help us to get more knowledge in the field of Information Technology & help us to have a bright future in the field of technology.

We hope our university will accept this attempt as a successful project.

Last but not the least, our sincere thanks to our parents, teaching and non-teaching staffs of our college and also my friends.

**MANISH DHUNGANA (5400)**

**ROHAN MISHRA (5423)**

**SANTOSH YADAV (5438)**

**SUYOG RAI (5401)**

**STUDENT’S DECLARATION**

We hereby declare that the project report entitled “**GYM MANAGEMENT SYSTEM**” is a result of our own work. If we are found guilty of copying any other report or published information and showing as our original work, we understand that we shall be liable and punishable by **Purbanchal University**.

We further certify that this Project submitted in partial fulfillment of the requirement for the award of Bachelor in Information Technology (**BIT**) of the **Purbanchal University** is our original work and has not been submitted for award of any other degree or other similar title or prize.

|  |  |  |  |
| --- | --- | --- | --- |
| S.N. | Name | Registration No. | Symbol No. |
| 1 | Manish Dhungana | 058-3-2-04723-2020 | 324620 |
| 2 | Rohan Mishra | 058-3-2-04734-2020 | 324631 |
| 3 | Santosh Yadav | 058-3-2-04740-2020 | 313437 |
| 4 | Suyog Rai | 058-3-2-04748-2020 | 324645 |

**TO WHOM IT MAY CONCERN**

This is to certify that **Mr. Manish Dhungana, Mr. Rohan Mishra, Mr. Santosh Yadav and Mr. Suyog Rai of Bachelor in Information Technology (BIT)** has studied as per the curriculum of **BIT 2nd Semester** and completed the project entitled **“GYM MANAGEMENT SYSTEM”**.This project is the original work of **Mr. Manish Dhungana, Mr. Rohan Mishra, Mr. Santosh Yadav and Mr. Suyog Rai** and was carried out under the supervision of **Mr.** **Deepak Khadka** as per the guidelines provided by **Purbanchal University** and certified as per the student’s declaration that project “**Gym Management System**” has not been presented anywhere as a part of any other academic work.

The detail of the student is as follows:

Name of Students : Manish Dhungana

Rohan Mishra

Santosh Yadav

Suyog Rai

Semester : 2nd

Subject Code : BIT 156 C0

Project Title : **Gym management System**

…………………………….

Mr. Deepak Khadka  
Program Coordinator, BIT  
KIST College of Information Technology

# ABSTRACT

**Gym Management System** is a project which aims in developing a computerized system to maintain all the daily work of gym with more efficiency. It can improve management of the files and data in the gym. It also has a facility of admin login through which the admin can monitor the whole system. The gym management system is the 32bit windows application to help users to maintain and organize the gym.

This Gym Management System is mainly used by gym manager and admin. Normal gym manager is able to manage the members of the gym details. The system is mainly used to design the user interface and keep database records. The facility of adding new members and searching the information of the members, etc. is there in the software. It is also easy to use for both beginners and advanced users.

Overall, this project of ours is being developed to help the gym owners to maintain the gym centers in the best way possible and also reduce the human efforts.

**TABLE OF CONTENTS**

[ABSTRACT 7](#_Toc111923195)

[CHAPTER 1 1](#_Toc111923196)

[INTRODUCTION 1](#_Toc111923197)

[1.1. INTRODUCTION 1](#_Toc111923198)

[1.2. PROBLEM STATEMENT 2](#_Toc111923199)

[1.3. OBJECTIVES 2](#_Toc111923200)

[1.4. SCOPE 3](#_Toc111923201)

[1.5. ADVANTAGES 3](#_Toc111923202)

[CHAPTER 2 4](#_Toc111923203)

[SYSTEM DESIGN 4](#_Toc111923204)

[2.1. ALGORITHM 4](#_Toc111923205)

[2.2. FLOWCHART 5](#_Toc111923206)

[2.3 E-R DIAGRAM 6](#_Toc111923207)

[CHAPTER 3 7](#_Toc111923209)

[REQUIREMENT ANALYSIS AND IMPLEMENTATION 7](#_Toc111923210)

[3.1. SYSTEM REQUIREMENTS 7](#_Toc111923211)

[3.2. SYSTEM METHODOLOGY 8](#_Toc111923212)

[CHAPTER 4 12](#_Toc111923214)

[CONCLUSION AND FUTURE SCOPE 12](#_Toc111923215)

[4.1. CONCLUSION 12](#_Toc111923216)

[4.2. FUTURE SCOPE 12](#_Toc111923217)

[CHAPTER 5 13](#_Toc111923218)

[5. APPENDICES 13](#_Toc111923219)

[5.1 SCREENSHOTS 13](#_Toc111923220)

[5.2 SOURCE CODE 20](#_Toc111923221)

# CHAPTER 1

## INTRODUCTION

### 1.1. INTRODUCTION

The project titled “Gym management system” is gym management software for managing, monitoring and controlling transaction gym. The project “Gym management system” is developed in C++ programming. It is specially designed to manage the overall aspect of gym. Having gym management software will make it easy and straightforward to use and execute business tasks and ultimately give the members the quality they expect out of the gym. Gym management software can also be referred as gym scheduling software as you can get an information update about employee, members and the trainer.

This is a simple gym management system. The system elaborates the basic concept of the storing and retrieving the information of the members. Although, in future the improvised version can be written in the other programming language as well. Our software is easy to use for beginners.

Gym Management Software is a gym and health club membership management system. You can keep records of your members, and their memberships, and have quick and easy communication between you and your members. It also includes a booking system and has a range of reports that help in the management of your club.

Therefore, Gym Management Software is a complete gym and recreation facility system program that looks after all of your members, memberships, and activities. It is designed for gyms, recreation centers, and health clubs. The Software provides lots of functions such as customer data entry, keeping records of all the customer’s fees, plan, and physical fitness which help to provide good quality of services to customer from Gym managers.

### 1.2. PROBLEM STATEMENT

The currently existing gym record system has been long outdated and lacks a lot of features. The traditional method of keeping records in registers manually has a lot lacking in comparison to the new technological methods. The problems of traditional Gym management system are such as the existing system is manual. It was very time consuming as data entry and calculations took a lot of time. Also, the searching of data was very difficult as one had to go through hundreds of pages of data. The chances of redundancy were also very high because checking and analyzing the data is very difficult. The method of managing data also is not very cost effective.

Furthermore, the chances of data loss are also coming in concern as there is only one register in which data is kept and if anything is to happen to that register all data is lost and taking about the data security, backup is also a very tedious task, as one would have to manually write hundreds of pages of data.

### 1.3. OBJECTIVES

The main objective of this whole Gym Management System Project is to eliminate the whole traditional management system and to eliminate its problem as well. This Project aims to do address the problems and solve them effectively. The objectives of this project are:

• To store, retrieve and manipulate the data.

• To reduce the work load because data entry would only require typing for a bit.

• To store data of the members and other information in computer database in spite of using longer bulky registers and physical hard copies

• To search of data with only a few commands.

• To reduce the data loss.

• To reduce data redundancy.

### 1.4. SCOPE

Here are some of the scopes of gym management system:

• Gym management system is best option for users for automated attendance.

• This system helps the owner and admin to maintain large data about users and their daily transactions in the gymnasium.

• These benefits to engage members smoothly, you can cut hours spent on admin, collect fees, send out invoices, automate communications, and enhance member retention.

• The need for a well-organized, computer base gymnasium management system has become the need of the society and gymnasium industry.

### 1.5. ADVANTAGES

* Managing Bookings and Scheduling Classes
* Cost Reduction Through Data
* Decision-Making Support
* A More Unified Workflow
* Gain Insight into Your Financial Reporting

.

# CHAPTER 2

## SYSTEM DESIGN

### 2.1. ALGORITHM

Step 1: Start.

Step 2: Program ask for Admin Login.

Step 3: The program asks for a choice based on the menu given bellow:

1. Membership Details

2. Trainer Details

3. Monetary Details

4. Exit

Step 4: Case 1: Membership Details.

Step 4.1: Asks for a choice based on the menu given bellow:

1. Add a new gym member

2. List details of all the members of gym

3. Find member and asks to edit or delete a membership of a gym members

4. Go back to Step 3

Case 2: Trainer Details.

Step 4.2: Asks for a choice based on the menu given bellow:

1. Add a new gym trainer

2. List details of all the trainers of gym

3. Find trainer and asks to edit or delete the detail of trainer.

4. Go back to Step 3

Case 3: Monetary Details.

Step 4.3: Asks for a choice based on the menu given bellow:

1.Show total income of the gym

2.Lists all the monetary transaction of the gym(All time)

3.Find specific transaction

4.Go back to Step 3

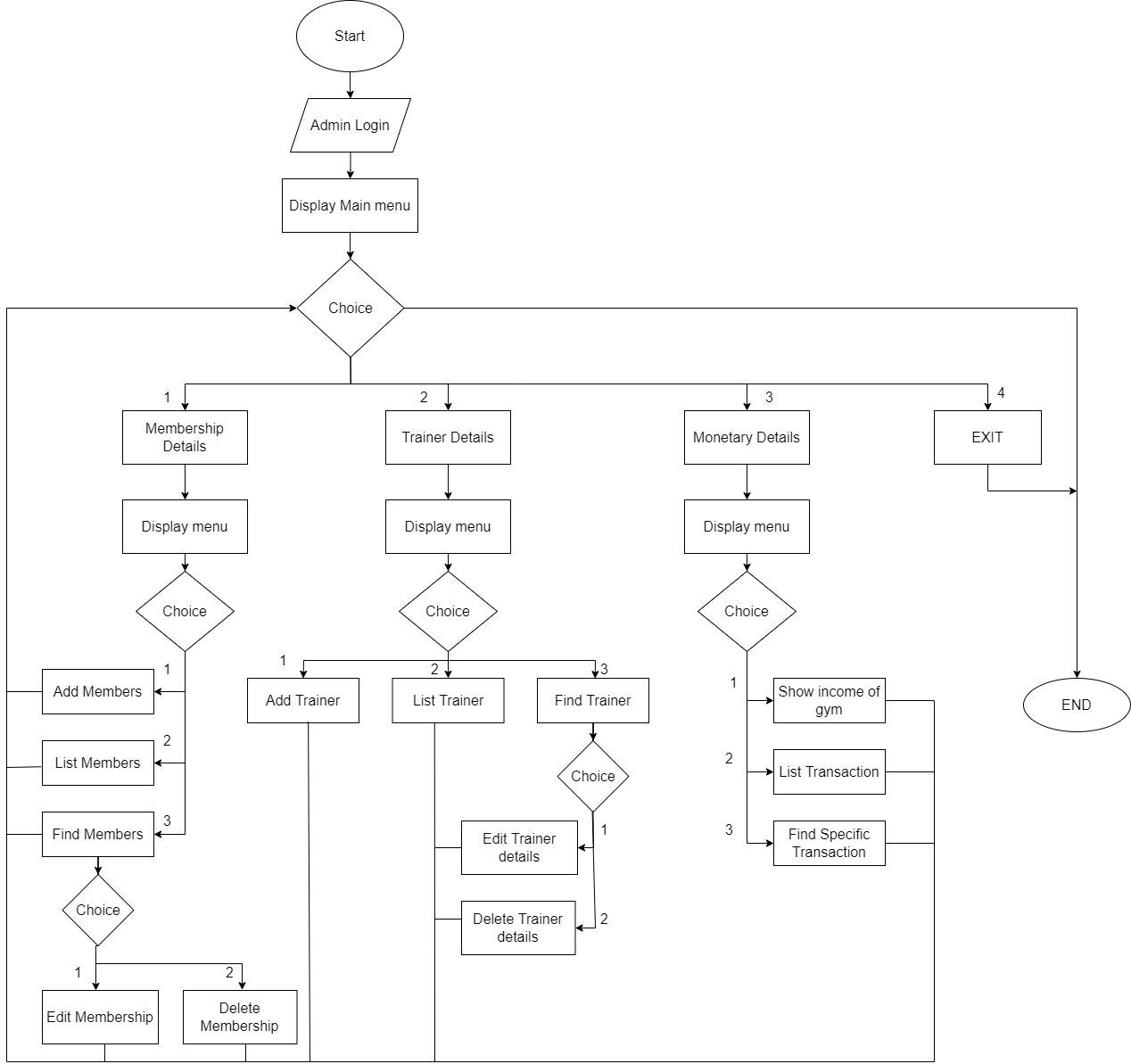
Case 4: Exits the program.

Step 5: Display output according to the case.

Step 6: Stop.

### 2.2. FLOWCHART

Flowchart is a diagram that represents workflow or process.



**Figure 1: Flowchart**

### 2.3 E-R DIAGRAM

# 

# CHAPTER 3

## REQUIREMENT ANALYSIS AND IMPLEMENTATION

### 3.1. SYSTEM REQUIREMENTS

Following hardware and software requirement should be met for flawless running of this system:

**Hardware:** Hardware is the collection of physical parts of a computer system. This includes the computer case, monitor, keyboard, and mouse. It also includes all the parts inside the computer case, such as the hard disk drive, motherboard, video card, and many others. Computer hardware is what you can physically touch.

**Minimum System Requirements**

OS: Windows 95 or higher.

RAM: 256 MB

Hard Disk: 200MB of free space available

Processor:  Intel Pentium III or later.

**Recommended System Requirements**

OS: Windows 7 or Higher

RAM: 4GB

Hard Disk: 500MB or above free space available

Processor: Intel I3 or Higher

**Software:** Software is a set of instructions, data or programs used to operate computers and

execute specific tasks. It is the opposite of hardware, which describes the physical aspects of a computer.

**OPERATING SYSTEM**: WINDOWS XP, 2000 Professional

**COMPILER**: DEV C++, TURBO C, VISUAL STUDIO

### 3.2. SYSTEM METHODOLOGY

#### WATERFALL MODEL

**The waterfall model is a linear, sequential approach to the software development life cycle (SDLC) that is popular in software engineering and product development. The waterfall model emphasizes a logical progression of steps.**

**The sequential phases described in the Waterfall model are:**

Deployment and Maintenance

Integration and Testing

Implementation

System Design

Requirement Analysis

Figure 3 :Waterfall model

#### 3.2.1. REQUIREMENT ANALYSIS

#### FUNCTIONAL REQUIREMENT

In software and system engineering, a functional requirement defines a function of a system or its component, where a function is described as a specification of behavior between input and outputs

### 

Figure 2:Use Case Diagram

#### 3.2.2. SYSTEM DESIGN

System design is the process of defining the components, modules, interfaces, and data for a system to satisfy specified requirements. System development is the process of creating or altering systems, along with the processes, practices, models, and methodologies used to develop them.

Importance:

* If any pre-existing code need to be understood, organized, and pieced together.
* It is common for the project team to have to write some code and produce original programs that support the application logic of the system.

#### 3.2.3. IMPLEMENTATION

This phase is initiated after the designing is compete. In this phase, programmers code based on project requirements and specifications, with some testing and implementation taking place as well. We used C programming to implement our project.

* File Handling was used for the data and records.
* Functions for sub modules.
* The system is first developed in small programs called units, which are integrated in the next phase. The testing of each developed unit individually is referred as unit testing.

#### 3.2.4. INTEGRATION AND TESTING

The systems integration test function is to ensure that the developed systems meet all the technical requirements with the components and subsystems integrated. All the modules/functions are tested. Individual functions are provided and output is generated. The code is tested through the unit testing.

* **Unit Testing:** A testing technique using which individual modules are tested to determine if there are any issues to be fixed. It is concerned with functional correctness of the standalone modules. The main aim is to isolate each unit of the system to identify, analyze and fix the defects.

**Unit Testing Technique:**

* + **Black box Testing:** Black -box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings.
  + **White box Testing:** White-box testing is a method of software testing that tests internal structures or workings of an application, as opposed to its functionality. In white box testing an internal perspective of the system, as well as programming skills, are used to design test cases.

#### 3.2.5. DEPLOYMENT AND MAINTENANCE

Once the software has been deployed for customer uses the maintenance phase begins and any defects found and change requests are then taken care of with new updates. The tests of the product and the product passes each testing phase, the product is ready to go live. This means the product is ready to be used in a real environment by all end users of the product.

* Once the functional and non-functional testing is done, the product us deployed in the customer environment or released into the market.
* After the product is deployed to the user’s market from there the maintenance phase starts
* Once the product or the system is in use there will be many patches to be fixed.
* The user might ask for new features and enhancements. It is the responsibility of the maintenance team to attend to these requests and to fix the bugs that are found.
* The maintenance effort revisits all the other stages of the software life cycle.
* Each modification requires planning, specification, design, coding, testing, installation.

# CHAPTER 4

## CONCLUSION AND FUTURE SCOPE

### 4.1. CONCLUSION

The project entitled “Gym Management System”, which was developed and had solved the problems that were identified in the current system as follows:

* The development of gym management system hopefully could solve the problem of member registration by performing the procedure on device, PCs which is more efficient and less expensive.
* The system could also solve the problem of membership renewal by renewing membership dates faster and searching process is done more effectively.
* The system of hopefully could solve the problem of non-member transaction by payment records be done quickly and reduce errors.
* This system could solve the problem of Attendance by recording date and exact time of every gym user automatically from the time of entry to the time of exit.
* The development of gym management system hopefully could solve the problem of financial reports by generating proof of output produced per period in the form of a report from the processing of transactions that can facilitate the process of checking the income of member registration, membership renewal and non-member transactions.

### 4.2. FUTURE SCOPE

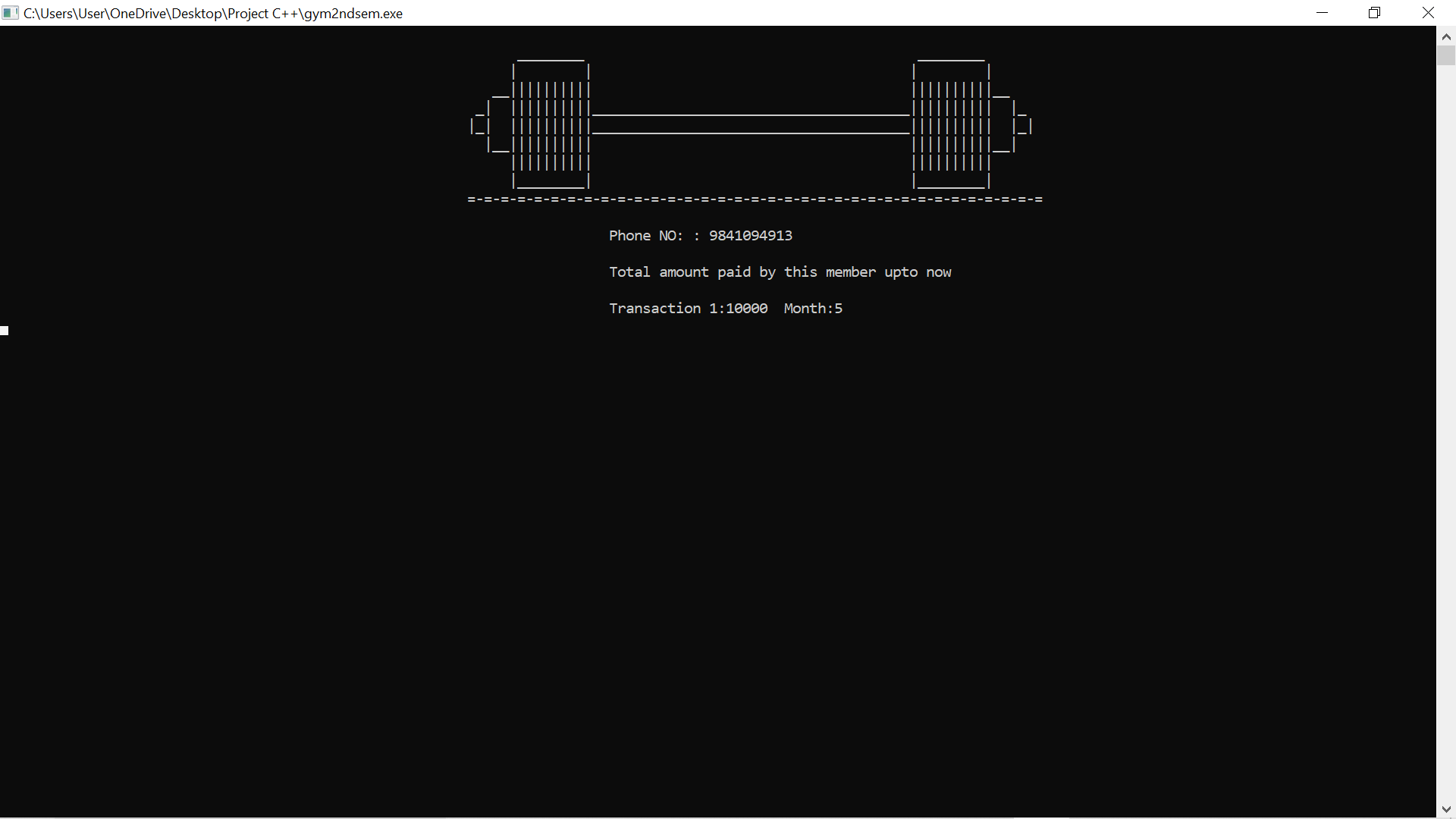
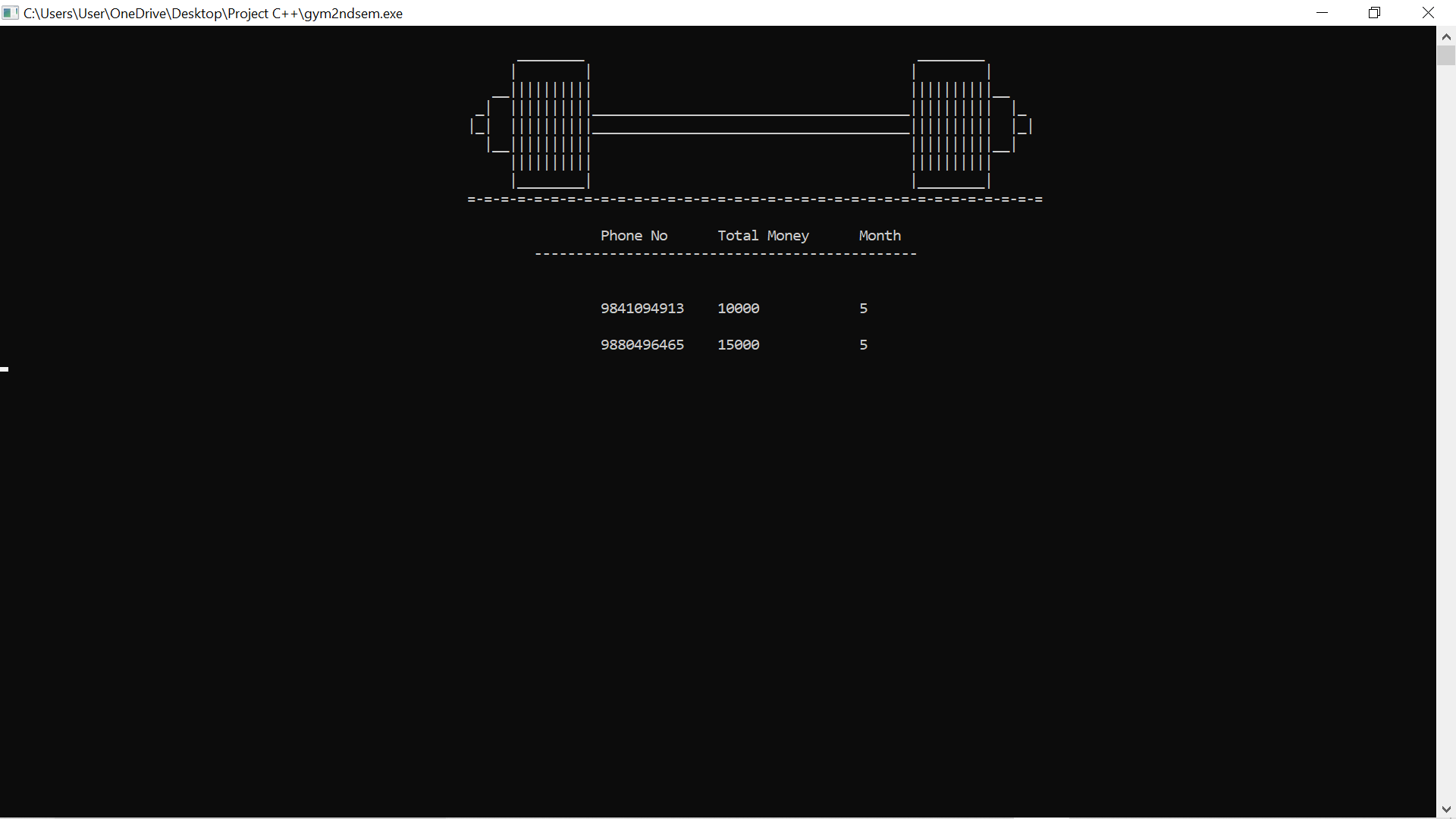
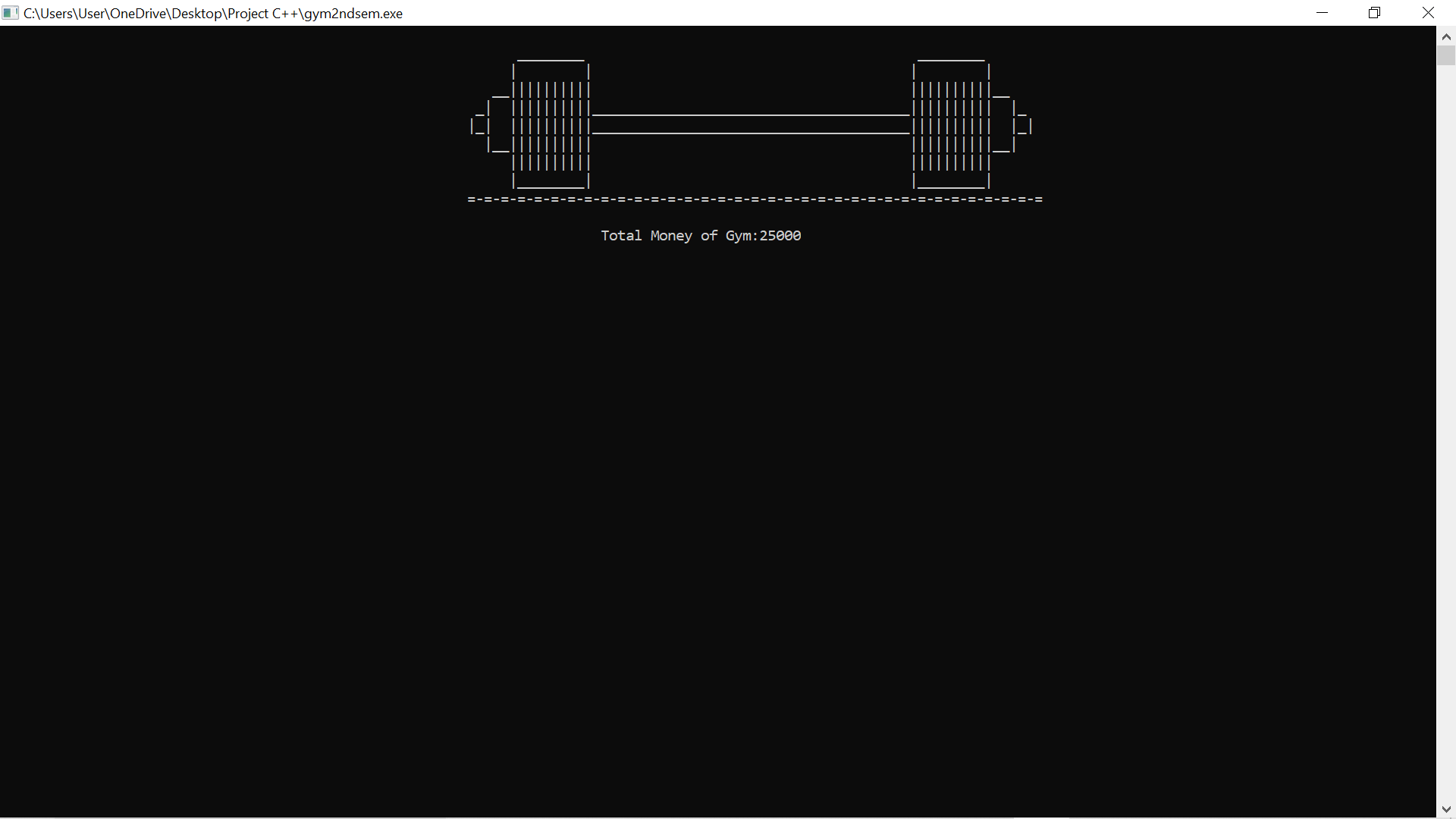
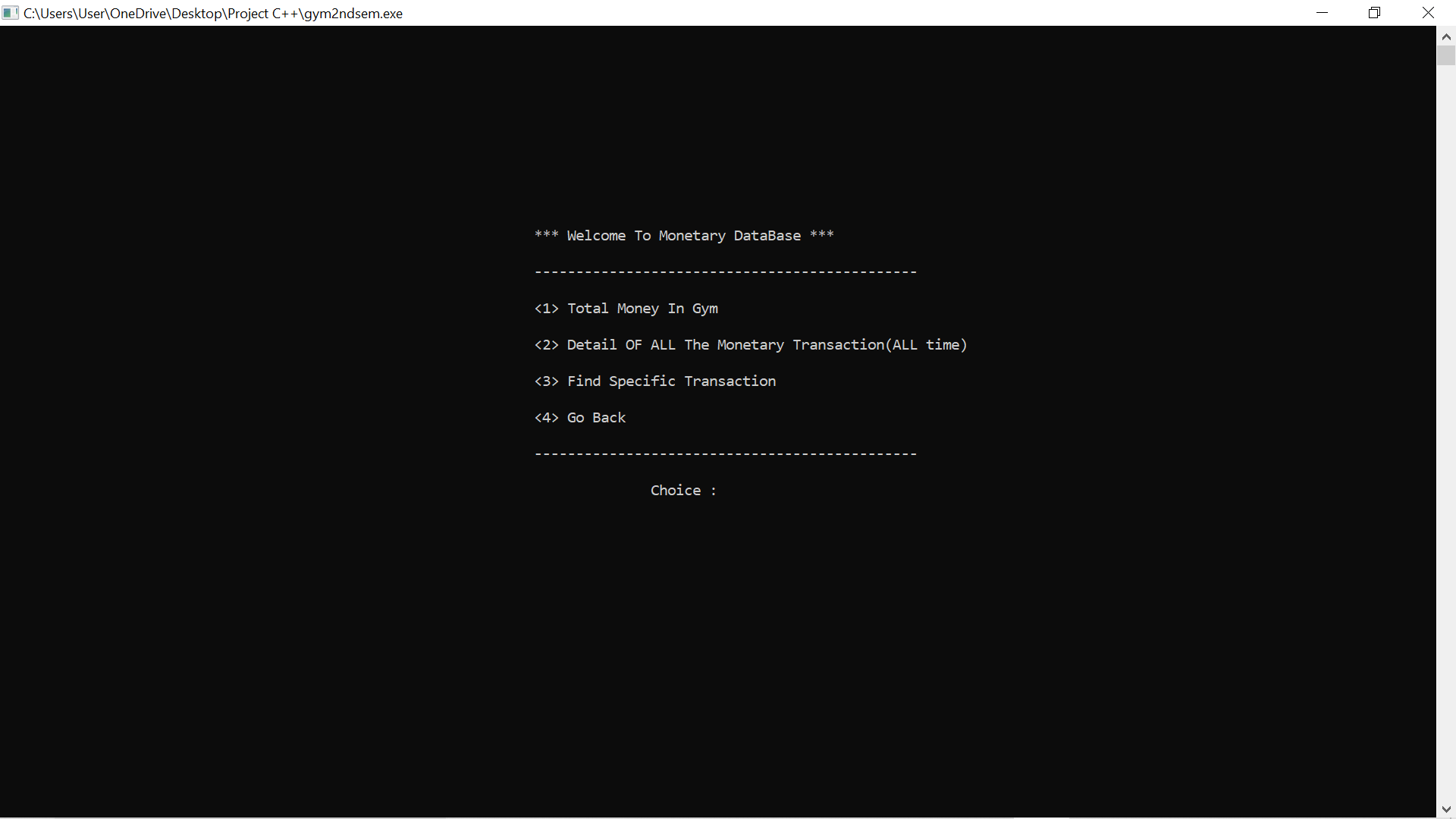
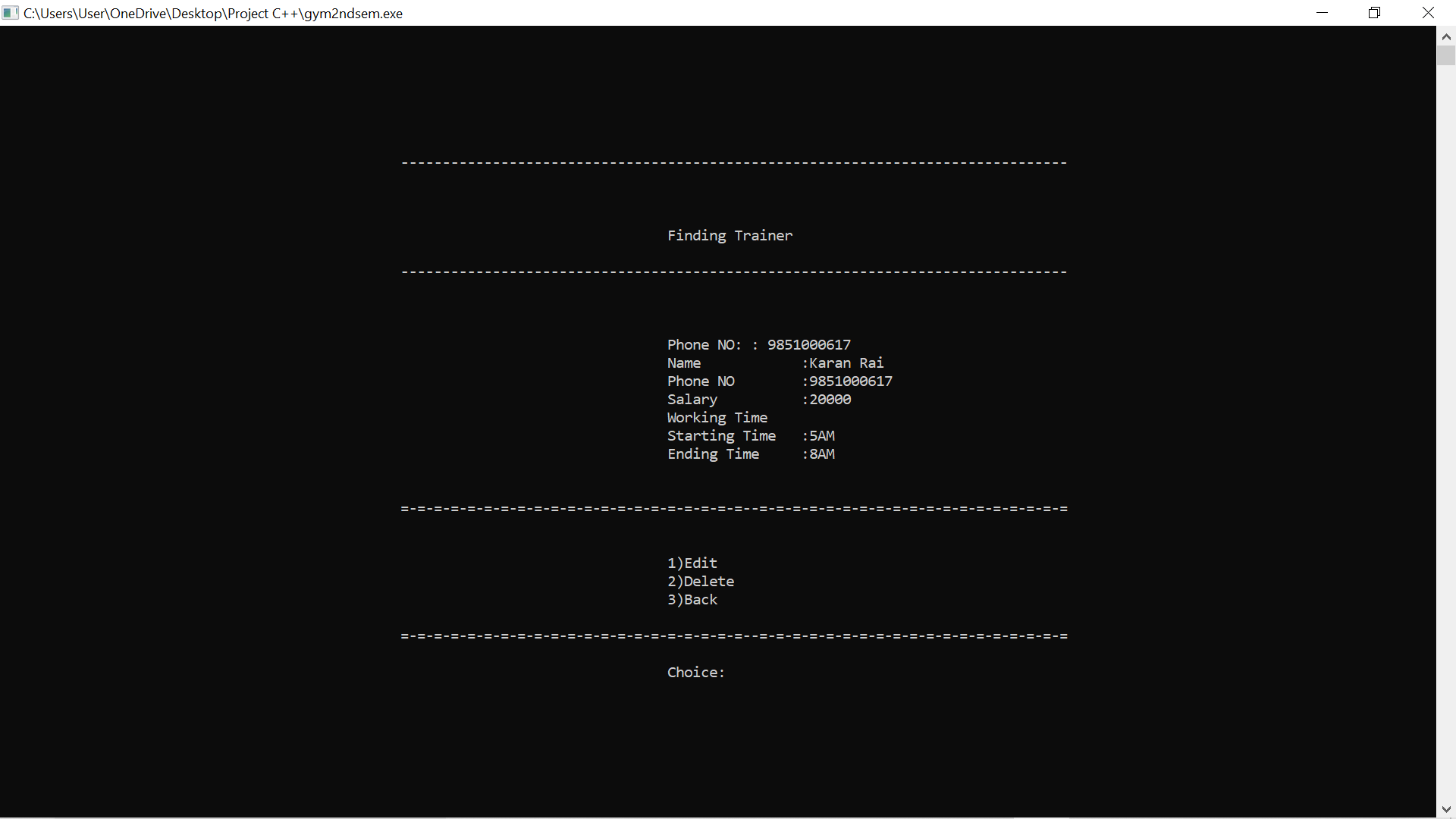
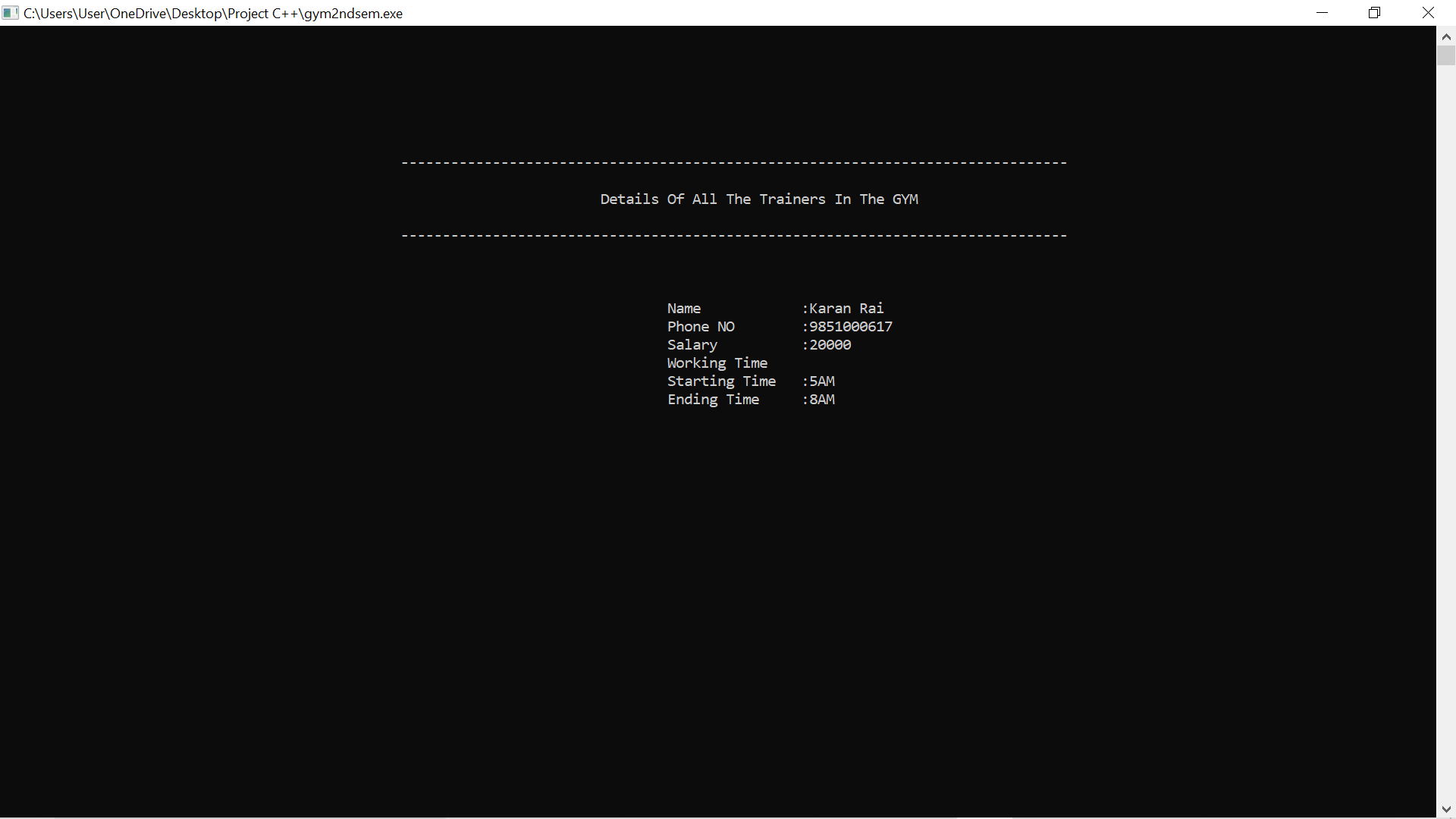
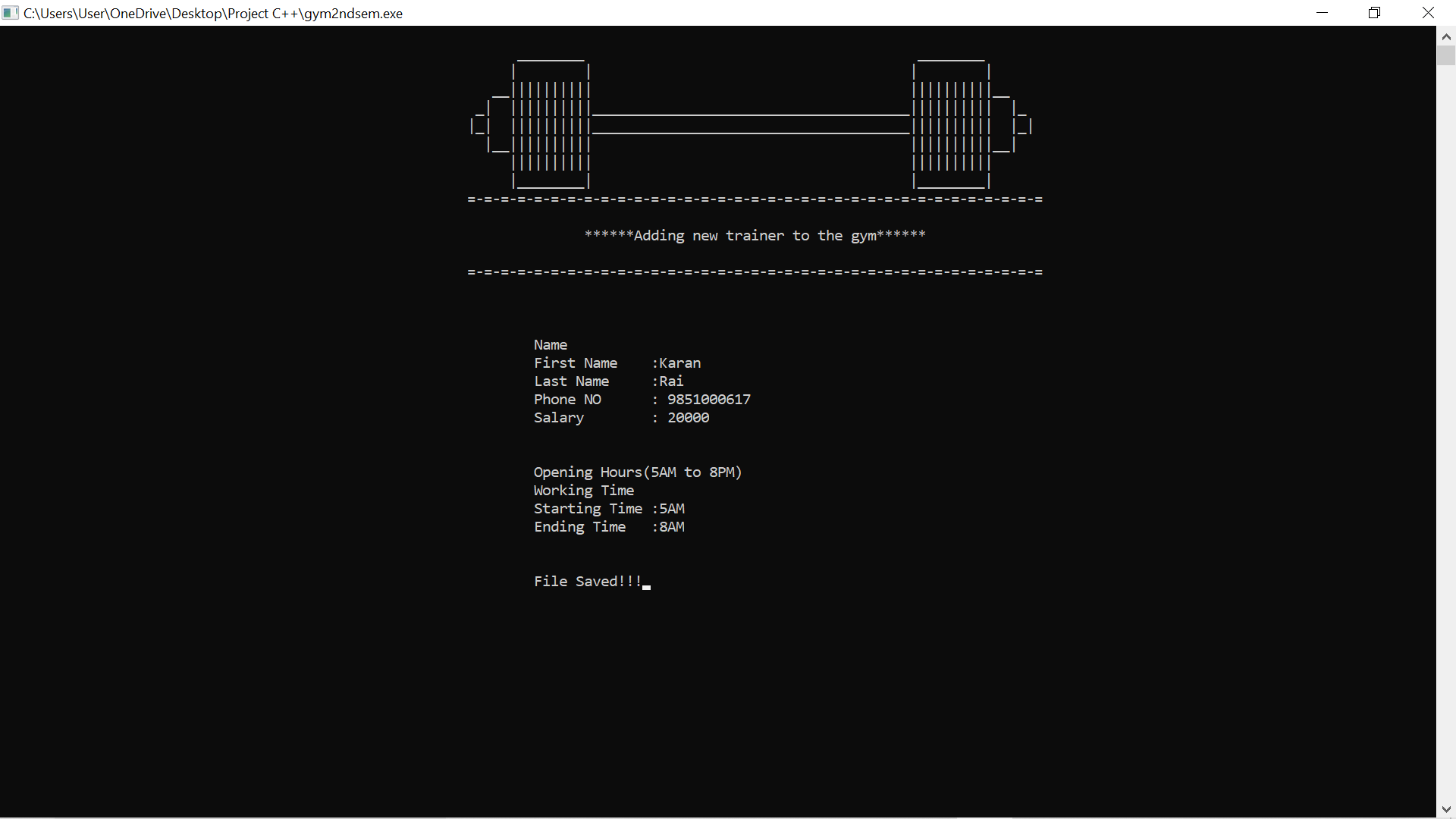
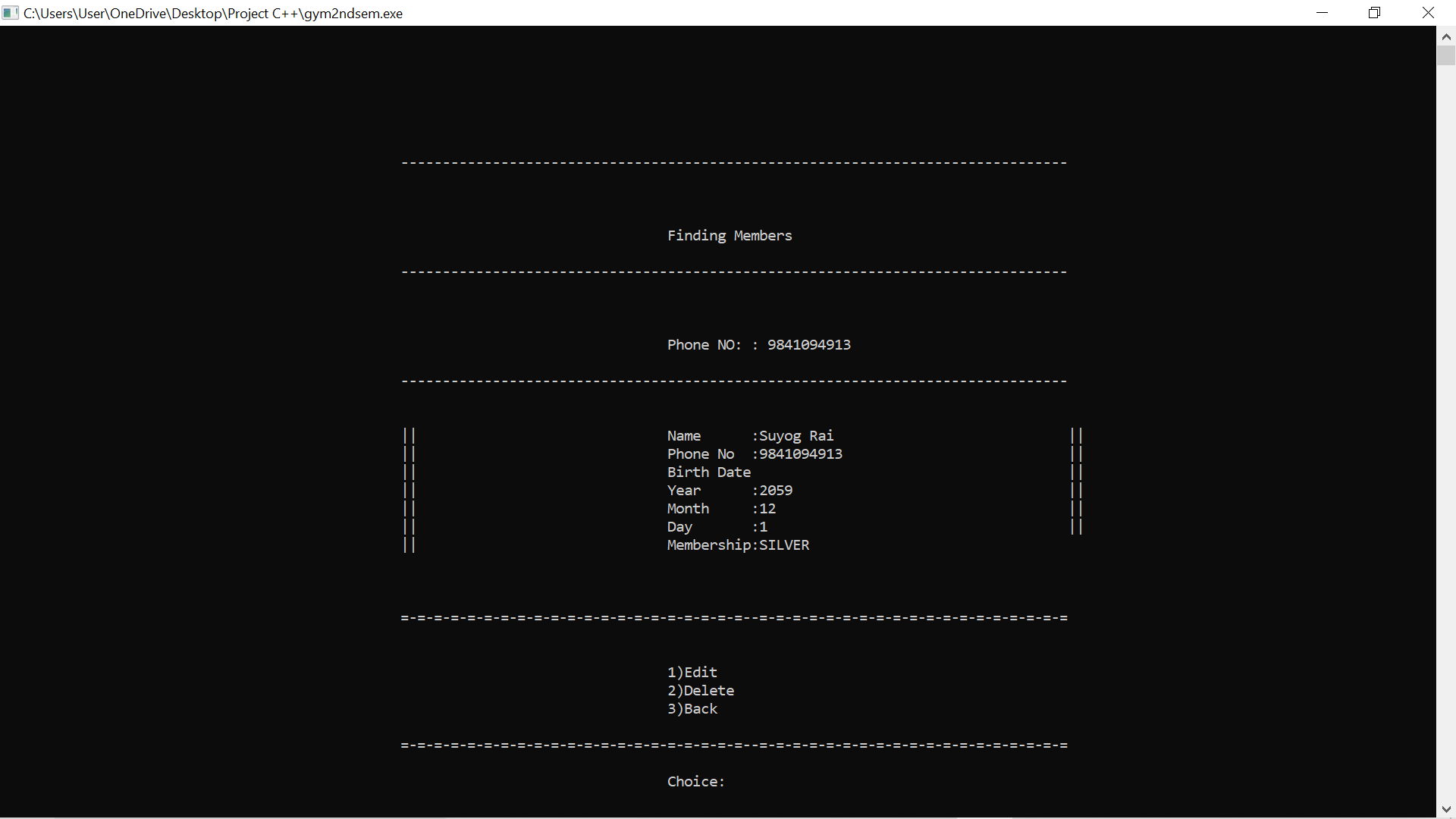
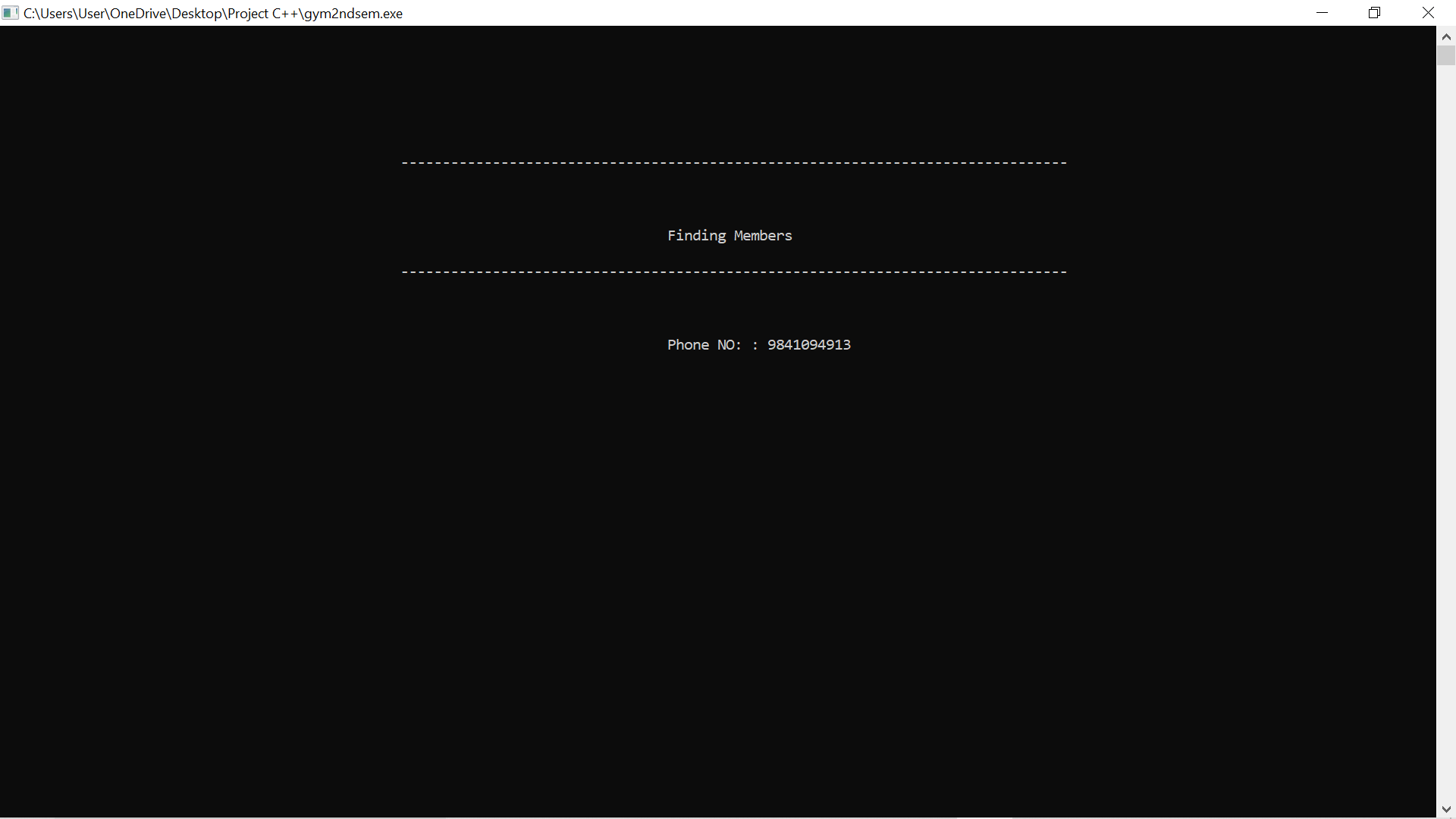
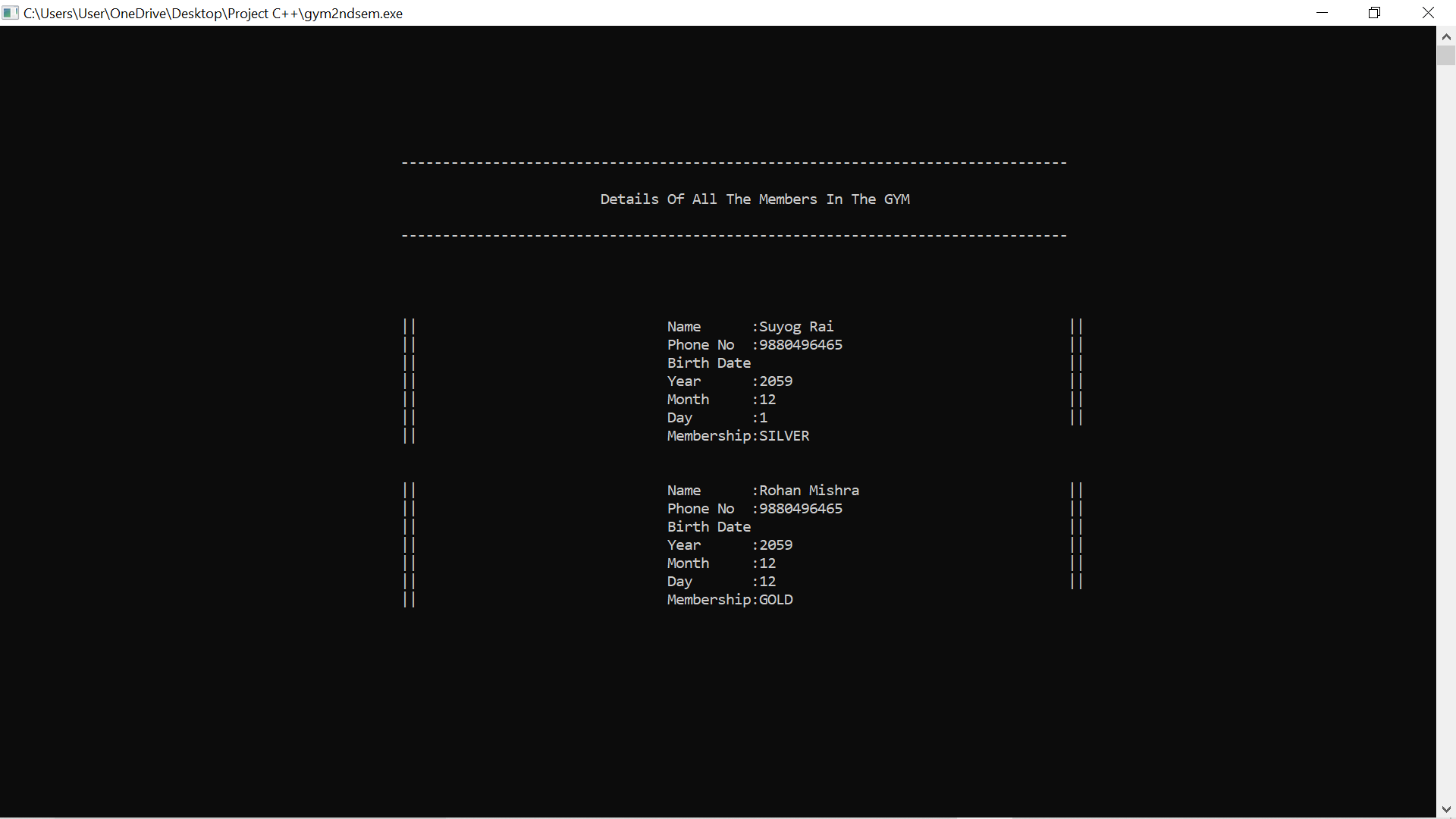
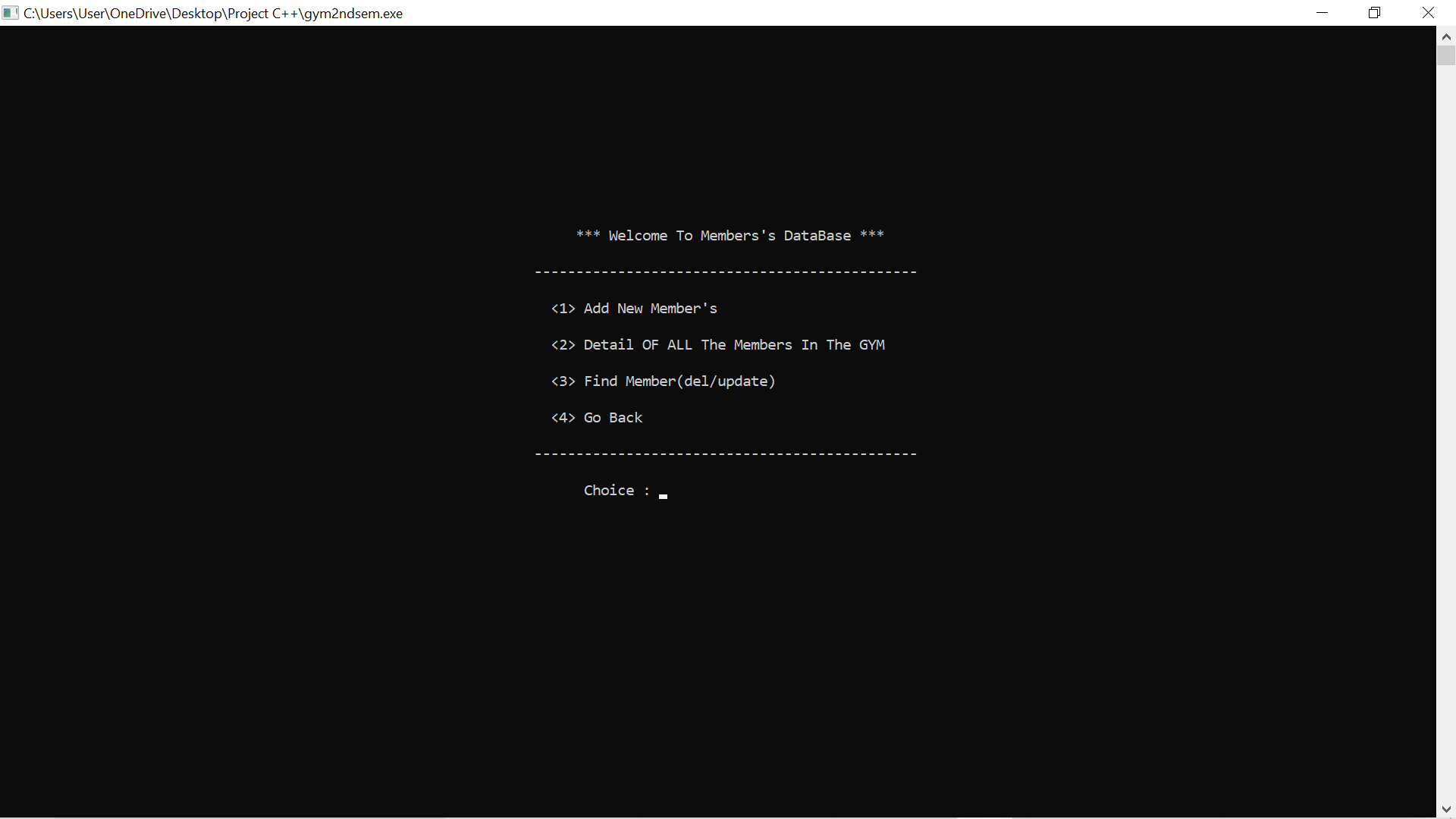
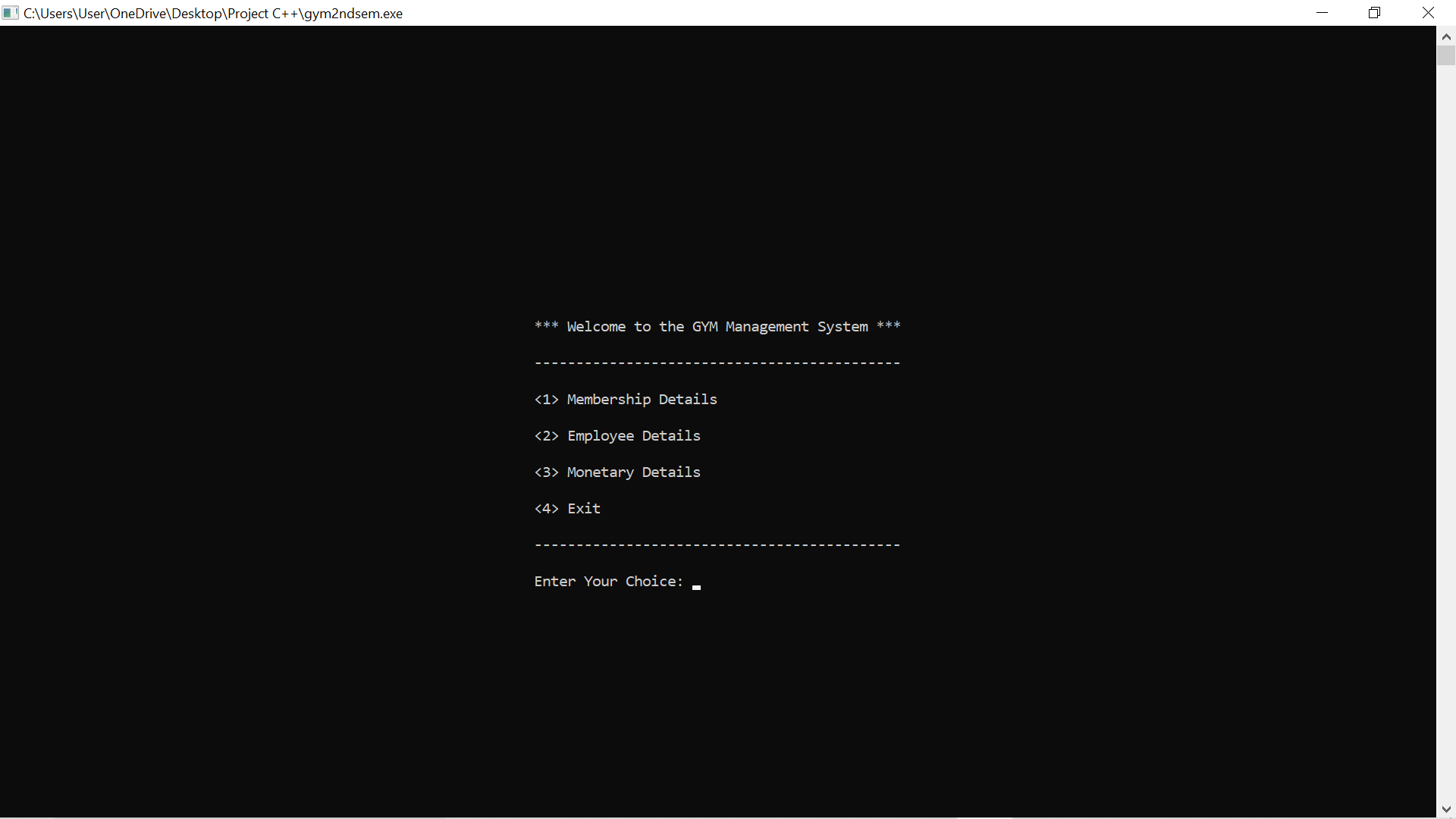
Therefore, the software is one of the most effective ways for modern gyms to achieve success and optimize their business potential and it has great future scope as well. This technology has progressed well beyond checking members into the gym and processing dues.

* Member Administration
* Software upgrade for better work efficient
* Organizing billing management effectively

# CHAPTER 5

## 5. APPENDICES

### 5.1 SCREENSHOTS



## 5.2 SOURCE CODE

#include<iostream>

#include<iomanip>

#include<conio.h>

#include<stdlib.h>

#include<fstream>

#include<string>

#include<unistd.h>

#include<windows.h>

using namespace std;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*FunctionDeclaration\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void adminlogin();

void mainmenu();

void editmenu();

void edittrainermenu();

void graphics();

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ClassCreationForMember\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

class Member

{

public:

int ch;

void input();

void add\_members();

void show\_members();

void find\_members();

void update\_member();

void del\_members();

};

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ClassCreationForMTrainer\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

class Trainer

{

public:

void Tinput();

void add\_Trainer();

void show\_Trainer();

void find\_Trainer();

void update\_tainer();

void del\_trainer();

};

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ClassCreationForMonetary\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

class Transaction

{

public:

void Minput();

void show\_Total();

void find\_Transaction();

void show\_all();

};

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*MemberFunctionDefinition\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void Member::input()

{

system("cls");

cout << "\n\n";

cout << "\n\n\n\n\n\n\n\n\n\t\t\t\t\t\t\t\t \*\*\* Welcome To Members's DataBase \*\*\*" << endl << endl;

cout << "\t\t\t\t\t\t\t\t----------------------------------------------" << endl << endl;

cout << "\t\t\t\t\t\t\t\t <1> Add New Member's " << endl << endl;

cout << "\t\t\t\t\t\t\t\t <2> Detail OF ALL The Members In The GYM " << endl << endl;

cout << "\t\t\t\t\t\t\t\t <3> Find Member(del/update) " << endl << endl;

cout << "\t\t\t\t\t\t\t\t <4> Go Back " << endl << endl;

cout << "\t\t\t\t\t\t\t\t----------------------------------------------" << endl << endl;

cout<<"\t\t\t\t\t\t\t\t Choice : ";

}

void Member::add\_members()

{

label0:

int package,month,total;

system("cls");

cout<<endl<<endl;

fstream file\_obj0;

fstream file\_obj1;

string MByear,MBmonth,MBday,Mphone,MFname,MLname,Membership;

fflush(stdin);

graphics();

cout<<"\n\t\t\t\t\t\t\t\t \*\*\*\*\*\*Adding new member to the gym\*\*\*\*\*\* "<<endl;

cout<<"\n\t\t\t\t\t\t\t=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=\n";

file\_obj0.open("E://Member2nd.txt",ios::app|ios::in);

cout << endl<<endl;

cout << "\t\t\t\t\t\t\t\t\t\tName ";

cout << "\n\t\t\t\t\t\t\t\t\t\tFirst Name :";

getline(cin, MFname);

fflush(stdin);

cout << "\t\t\t\t\t\t\t\t\t\tLast Name :";

getline(cin, MLname);

fflush(stdin);

cout << endl;

cout << "\t\t\t\t\t\t\t\t\t\tPhone No. : ";

getline(cin,Mphone);

fflush(stdin);

cout << "\t\t\t\t\t\t\t\t\t\tBirth Date"<<endl;

cout << "\t\t\t\t\t\t\t\t\t\tYear :";

getline(cin,MByear);

fflush(stdin);

cout << "\t\t\t\t\t\t\t\t\t\tMonth : ";

getline(cin,MBmonth);

fflush(stdin);

cout << "\t\t\t\t\t\t\t\t\t\tDay : ";

getline(cin, MBday);

fflush(stdin);

flag1:

cout<<endl;

cout<<"\n\t\t\t\t\t\t\t=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=\n";

cout<<endl;

cout << "\t\t\t\t\t\t\t\t\tSelect your Membership Package: \n";

cout << "\t\t\t\t\t\t\t\t1) SILVER 2) GOLD 3) Details of packages\n";

cout << "\n\t\t\t\t\t\t\t\t\t\t Choice: ";

cin>>package;

if(package == 1)

{

cout<< "\n\t\t\t\t\t\t\t\t\tEnter total month of Subscription : ";

cin >> month;

Membership="SILVER";

total = month \* 2000;

file\_obj1.open("E://Monetary.txt",ios::out|ios::app);

file\_obj1<<" "<<Mphone<<" "<<total<<" "<<month<<" "<<"\n";

file\_obj1.close();

}

else if(package == 2)

{

cout<< "\n\t\t\t\t\t\t\t\t\tEnter total month of Subscription : ";

cin >> month;

Membership="GOLD";

total = month \* 3000;

file\_obj1.open("E://Monetary.txt",ios::out|ios::app);

file\_obj1<<" "<<Mphone<<" "<<total<<" "<<month<<" "<<"\n";

file\_obj1.close();

}

else if(package == 3)

{

cout<<"\n\t\t\t\t\t\t\t\t\t Silver Gold";

cout<<"\n\t\t\t\t\t\t\t\t\t-Gym Equipment access -Gym Equipment access ";

cout<<"\n\t\t\t\t\t\t\t\t\t-Gym Trainer access -Gym Trainer access";

cout<<"\n\t\t\t\t\t\t\t\t\t-Paid access to massage -One on One trainer interaction";

cout<<"\n\t\t\t\t\t\t\t\t\t and spa sessions";

cout<<"\n\t\t\t\t\t\t\t\t\t-Paid entry to sauna -Online Sessions";

cout<<"\n\t\t\t\t\t\t\t\t\t -Free massage and spa sessions";

cout<<"\n\t\t\t\t\t\t\t\t\t -Free sauna entry";

getchar();

goto flag1;

}

else

{

goto flag1;

}

file\_obj0.close();

file\_obj0.open("E://Member2nd.txt",ios::out|ios::app);

file\_obj0<<" "<<MFname<<" "<<MLname;

file\_obj0<<" "<<Mphone<<" "<<MByear<<" "<<MBmonth<<" "<<MBday<<" "<<Membership<<"\n";

file\_obj0.close();

cout<<"\n\t\t\t\t\t\t\t\t\tMember addded sucessfully!!!";

system("cls");

}

void Member::show\_members()

{

label1:

system("cls");

string MByear,MBmonth,MBday,Mphone,MFname,MLname,Membership;

cout << "\n\n\n\n\n\n\n\t\t\t\t\t\t--------------------------------------------------------------------------------" << endl << endl;

cout << "\t\t\t\t\t\t\t\t\tDetails Of All The Members In The GYM" << endl << endl;

cout << "\t\t\t\t\t\t--------------------------------------------------------------------------------" << endl << endl;

cout << endl << endl;

fstream file\_obj0;

file\_obj0.open("E://Member2nd.txt",ios::in);

if(!file\_obj0)

cout<<"\n\n File Openning Error...";

else

{

file\_obj0>>MFname>>MLname;

file\_obj0>>Mphone>>MByear>>MBmonth>>MBday>>Membership;

while(!file\_obj0.eof())

{

cout<<"\n\t\t\t\t\t\t||\t\t\t\tName :"<<MFname<<" "<<MLname<<"\t\t\t\t||";

cout<<"\n\t\t\t\t\t\t||\t\t\t\tPhone No :"<<Mphone<<"\t\t\t\t||";

cout<<"\n\t\t\t\t\t\t||\t\t\t\tBirth Date "<<"\t\t\t\t\t||";

cout<<"\n\t\t\t\t\t\t||\t\t\t\tYear :"<<MByear<<"\t\t\t\t\t||";

cout<<"\n\t\t\t\t\t\t||\t\t\t\tMonth :"<<MBmonth<<"\t\t\t\t\t||";

cout<<"\n\t\t\t\t\t\t||\t\t\t\tDay :"<<MBday<<"\t\t\t\t\t||";

cout<<"\n\t\t\t\t\t\t||\t\t\t\tMembership:"<<Membership;

cout<<endl<<endl;

file\_obj0>>MFname>>MLname>>Mphone>>MByear>>MBmonth>>MBday>>Membership;

}

file\_obj0.close();

}

getch();

}

void Member::find\_members()

{

label2:

system("cls");

cout << "\n\n\n\n\n\n\n\t\t\t\t\t\t--------------------------------------------------------------------------------" << endl << endl;

cout<<endl<<endl;

cout << "\t\t\t\t\t\t\t\t\t\tFinding Members" << endl << endl;

cout << "\t\t\t\t\t\t--------------------------------------------------------------------------------" << endl << endl;

fstream file\_obj0;

string MByear,MBmonth,MBday,Mphone,MFname,MLname,Membership,FID;

file\_obj0.open("E://Member2nd.txt",ios::app|ios::in);

if(!file\_obj0)

cout<<"\n\n File Openning Error...";

else

{

cout<<"\n\n\t\t\t\t\t\t\t\t\t\tPhone NO: : ";

cin>>FID;

cout << "\n\t\t\t\t\t\t--------------------------------------------------------------------------------" << endl;

cout<<endl;

file\_obj0>>MFname>>MLname;

file\_obj0>>Mphone>>MByear>>MBmonth>>MBday>>Membership;

while(!file\_obj0.eof())

{

if(FID==Mphone)

{

cout<<"\n\t\t\t\t\t\t||\t\t\t\tName :"<<MFname<<" "<<MLname<<"\t\t\t\t||";

cout<<"\n\t\t\t\t\t\t||\t\t\t\tPhone No :"<<Mphone<<"\t\t\t\t||";

cout<<"\n\t\t\t\t\t\t||\t\t\t\tBirth Date "<<"\t\t\t\t\t||";

cout<<"\n\t\t\t\t\t\t||\t\t\t\tYear :"<<MByear<<"\t\t\t\t\t||";

cout<<"\n\t\t\t\t\t\t||\t\t\t\tMonth :"<<MBmonth<<"\t\t\t\t\t||";

cout<<"\n\t\t\t\t\t\t||\t\t\t\tDay :"<<MBday<<"\t\t\t\t\t||";

cout<<"\n\t\t\t\t\t\t||\t\t\t\tMembership:"<<Membership;

cout<<endl<<endl;

break;

}

file\_obj0>>MFname>>MLname>>Mphone>>MByear>>MBmonth>>MBday>>Membership;

}

file\_obj0.close();

}

editmenu();

}

void Member::update\_member()

{

system("cls");

int package,month,total;

fstream file\_obj0,file\_obj1,temp\_file;

string MByear,MBmonth,MBday,Mphone,MFname,MLname,Membership,FID;

cout<<"\n\n\t\t\t\t\t\t\t\t\t\tUpdate Member Record";

file\_obj0.open("E://Member2nd.txt",ios::in);

temp\_file.open("E://temp\_file.txt",ios::app|ios::out);

if(!file\_obj0)

cout<<"\n\n\t\t\t\t\t\t\t\t\t\t File Openning Error...";

else

{

cout<<"\n\n\t\t\t\t\t\t\t\t\t\t\t Phone NO: : ";

cin>>FID;

fflush(stdin);

file\_obj0>>MFname>>MLname;

file\_obj0>>Mphone>>MByear>>MBmonth>>MBday>>Membership;

while(!file\_obj0.eof())

{

if(FID==Mphone)

{

cout << "\t\t\t\t\t\t\t\t\t\tName ";

cout << "\n\t\t\t\t\t\t\t\t\t\tFirst Name :";

getline(cin, MFname);

fflush(stdin);

cout << "\t\t\t\t\t\t\t\t\t\tLast Name :";

getline(cin, MLname);

fflush(stdin);

cout << endl;

cout << "\t\t\t\t\t\t\t\t\t\tPhone No. : ";

getline(cin,Mphone);

fflush(stdin);

cout << "\t\t\t\t\t\t\t\t\t\tBirth Date"<<endl;

cout << "\t\t\t\t\t\t\t\t\t\tYear :";

getline(cin,MByear);

fflush(stdin);

cout << "\t\t\t\t\t\t\t\t\t\tMonth : ";

getline(cin,MBmonth);

fflush(stdin);

cout << "\t\t\t\t\t\t\t\t\t\tDay : ";

getline(cin, MBday);

fflush(stdin);

flag1:

cout<<endl;

cout<<"\n\t\t\t\t\t\t\t=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=\n";

cout<<endl;

cout << "\t\t\t\t\t\t\t\t\tSelect your Membership Package: \n";

cout << "\t\t\t\t\t\t\t\t\t1) SILVER 2) GOLD\n";

cout << "\n\t\t\t\t\t\t\t\t\t\t Choice: ";

cin>>package;

cout<< "\n\t\t\t\t\t\t\t\t\tEnter total month of Subscription : ";

cin >> month;

if(package == 1)

{

Membership="SILVER";

total = month \* 2000;

file\_obj1.open("C://Monetary.txt",ios::out|ios::app);

file\_obj1<<" "<<Mphone<<" "<<total<<"\n";

file\_obj1.close();

}

else if(package == 2)

{

Membership="GOLD";

total = month \* 3000;

file\_obj1.open("C://Monetary.txt",ios::out|ios::app);

file\_obj1<<" "<<Mphone<<" "<<total<<"\n";

file\_obj1.close();

}

else

{

goto flag1;

}

temp\_file<<" "<<MFname<<" "<<MLname<<" "<<Mphone<<" "<<MByear<<" "<<MBmonth<<" "<<MBday<<" "<<Membership<<"\n";

}

else

temp\_file<<" "<<MFname<<" "<<MLname<<" "<<Mphone<<" "<<MByear<<" "<<MBmonth<<" "<<MBday<<" "<<Membership<<"\n";

file\_obj0>>MFname>>MLname>>Mphone>>MByear>>MBmonth>>MBday>>Membership;

}

file\_obj0.close();

temp\_file.close();

remove("E://Member2nd.txt");

rename("E://temp\_file.txt","E://Member2nd.txt");

}

}

void Member::del\_members()

{

system("cls");

fstream file\_obj0,temp\_file;

string MByear,MBmonth,MBday,Mphone,MFname,MLname,Membership,FID;

cout<<"\n\n\t\t\t\tDelete Member Record";

file\_obj0.open("E://Member2nd.txt",ios::in);

temp\_file.open("E://temp\_file.txt",ios::app|ios::out);

if(!file\_obj0)

cout<<"\n\n File Openning Error...";

else

{

cout<<"\n\n Phone NO: : ";

cin>>FID;

fflush(stdin);

file\_obj0>>MFname>>MLname;

file\_obj0>>Mphone>>MByear>>MBmonth>>MBday>>Membership;

while(!file\_obj0.eof())

{

if(FID==Mphone)

{

system("cls");

cout<<"\n\n\t\t\t\tDeleted Member Record";

cout<<"\n\n One Member is Deleted Successfully...";

getchar();

}

else

temp\_file<<" "<<MFname<<" "<<MLname<<" "<<Mphone<<" "<<MByear<<" "<<MBmonth<<" "<<MBday<<" "<<Membership<<"\n";

file\_obj0>>MFname>>MLname>>Mphone>>MByear>>MBmonth>>MBday>>Membership;

}

file\_obj0.close();

temp\_file.close();

remove("E://Member2nd.txt");

rename("E://temp\_file.txt","E://Member2nd.txt");

}

}

void Trainer::Tinput()

{

system("cls");

cout << "\n\n";

cout << "\n\n\n\n\n\n\n\n\n\t\t\t\t\t\t\t\t\*\*\* Welcome To Trainers's DataBase \*\*\*" << endl << endl;

cout << "\t\t\t\t\t\t\t\t----------------------------------------------" << endl << endl;

cout << "\t\t\t\t\t\t\t\t<1> Add New Trainer " << endl << endl;

cout << "\t\t\t\t\t\t\t\t<2> Detail OF ALL The Trainers In The GYM " << endl << endl;

cout << "\t\t\t\t\t\t\t\t<3> Find Trainer(del/update) " << endl << endl;

cout << "\t\t\t\t\t\t\t\t<4> Go Back " << endl << endl;

cout << "\t\t\t\t\t\t\t\t----------------------------------------------" << endl << endl;

cout<<"\t\t\t\t\t\t\t\t\t Choice : ";

}

void Trainer::add\_Trainer()

{

label0:

system("cls");

graphics();

cout<<"\n\t\t\t\t\t\t\t\t \*\*\*\*\*\*Adding new trainer to the gym\*\*\*\*\*\* "<<endl;

cout<<"\n\t\t\t\t\t\t\t=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=\n";

fstream file\_obj1;

string Tsalary,Tphone,TFname,TLname,temp,Stime,Etime;

fflush(stdin);

cout << endl << endl;

cout << "\n\t\t\t\t\t\t\t\tName ";

cout << "\n\t\t\t\t\t\t\t\tFirst Name :";

getline(cin, TFname);

fflush(stdin);

cout << "\t\t\t\t\t\t\t\tLast Name :";

getline(cin, TLname);

fflush(stdin);

cout << "\t\t\t\t\t\t\t\tPhone NO : ";

getline(cin,Tphone);

fflush(stdin);

cout << "\t\t\t\t\t\t\t\tSalary : ";

getline(cin,Tsalary);

fflush(stdin);

cout << endl;

cout << "\n\t\t\t\t\t\t\t\tOpening Hours(5AM to 8PM) ";

cout << "\n\t\t\t\t\t\t\t\tWorking Time ";

cout << "\n\t\t\t\t\t\t\t\tStarting Time :";

getline(cin, Stime);

fflush(stdin);

cout << "\t\t\t\t\t\t\t\tEnding Time :";

getline(cin, Etime);

fflush(stdin);

cout << endl << endl;

temp=".";

file\_obj1.open("E://Trainer2nd.txt",ios::out|ios::app);

file\_obj1<<" "<<TFname<<" "<<TLname<<" "<<Tphone<<" "<<Tsalary<<" "<<Stime<<" "<<Etime<<"\n";

file\_obj1.close();

cout<<"\t\t\t\t\t\t\t\tFile Saved!!!";

getchar();

system("cls");

}

void Trainer::show\_Trainer()

{

label1:

system("cls");

fstream file\_obj1;

string Tsalary,Tphone,TFname,TLname,temp,Stime,Etime;

cout << "\n\n\n\n\n\n\n\t\t\t\t\t\t--------------------------------------------------------------------------------" << endl << endl;

cout << "\t\t\t\t\t\t\t\t\tDetails Of All The Trainers In The GYM" << endl << endl;

cout << "\t\t\t\t\t\t--------------------------------------------------------------------------------" << endl << endl;

cout << endl << endl;

file\_obj1.open("E://Trainer2nd.txt",ios::in);

if(!file\_obj1)

cout<<"\n\n File Openning Error...";

else

{

file\_obj1>>TFname>>TLname;

file\_obj1>>Tphone>>Tsalary>>Stime>>Etime;

while(!file\_obj1.eof())

{

cout<<"\t\t\t\t\t\t\t\t\t\tName :"<<TFname<<" "<<TLname<<endl;

cout<<"\t\t\t\t\t\t\t\t\t\tPhone NO :"<<Tphone<<endl;

cout<<"\t\t\t\t\t\t\t\t\t\tSalary :"<<Tsalary<<endl;

cout<<"\t\t\t\t\t\t\t\t\t\tWorking Time"<<endl;

cout<<"\t\t\t\t\t\t\t\t\t\tStarting Time :"<<Stime<<endl;

cout<<"\t\t\t\t\t\t\t\t\t\tEnding Time :"<<Etime<<endl;

cout<<endl<<endl;

file\_obj1>>TFname>>TLname;

file\_obj1>>Tphone>>Tsalary>>Stime>>Etime;

}

file\_obj1.close();

}

getchar();

}

void Trainer::find\_Trainer()

{

label2:

system("cls");

cout << "\n\n\n\n\n\n\n\t\t\t\t\t\t--------------------------------------------------------------------------------" << endl << endl;

cout<<endl<<endl;

cout << "\t\t\t\t\t\t\t\t\t\tFinding Trainer" << endl << endl;

cout << "\t\t\t\t\t\t--------------------------------------------------------------------------------" << endl << endl;

fstream file\_obj1;

string TFid;

string Tsalary,Tphone,TFname,TLname,temp,Stime,Etime;

file\_obj1.open("E://Trainer2nd.txt",ios::in);

if(!file\_obj1)

cout<<"\n\n File Openning Error...";

else

{

cout<<"\n\n\t\t\t\t\t\t\t\t\t\tPhone NO: : ";

cin>>TFid;

file\_obj1>>TFname>>TLname;

file\_obj1>>Tphone>>Tsalary>>Stime>>Etime;

while(!file\_obj1.eof())

{

if(TFid==Tphone)

{

cout<<"\t\t\t\t\t\t\t\t\t\tName :"<<TFname<<" "<<TLname<<endl;

cout<<"\t\t\t\t\t\t\t\t\t\tPhone NO :"<<Tphone<<endl;

cout<<"\t\t\t\t\t\t\t\t\t\tSalary :"<<Tsalary<<endl;

cout<<"\t\t\t\t\t\t\t\t\t\tWorking Time"<<endl;

cout<<"\t\t\t\t\t\t\t\t\t\tStarting Time :"<<Stime<<endl;

cout<<"\t\t\t\t\t\t\t\t\t\tEnding Time :"<<Etime<<endl;

cout<<endl<<endl;

break;

}

file\_obj1>>TFname>>TLname;

file\_obj1>>Tphone>>Tsalary>>Stime>>Etime;

}

file\_obj1.close();

}

edittrainermenu();

}

void Trainer::update\_tainer()

{

system("cls");

fstream file\_obj1,temp\_file1;

string TFid;

string Tsalary,Tphone,TFname,TLname,temp,Stime,Etime;

cout<<"\n\n\t\t\t\t\t\t\t\t\t\tUpdate Trainer Record";

file\_obj1.open("E://Trainer2nd.txt",ios::in);

temp\_file1.open("E://temp\_file2nd.txt",ios::app|ios::out);

if(!file\_obj1)

cout<<"\n\n File Openning Error...";

else

{

cout<<"\n\n Phone NO: : ";

cin>>TFid;

fflush(stdin);

file\_obj1>>TFname>>TLname;

file\_obj1>>Tphone>>Tsalary>>Stime>>Etime;

while(!file\_obj1.eof())

{

if(TFid==Tphone)

{

cout << "\t\t\t\t\t\t\t\tName ";

cout << "\n\t\t\t\t\t\t\t\tFirst Name :";

getline(cin, TFname);

fflush(stdin);

cout << "\t\t\t\t\t\t\t\tLast Name :";

getline(cin, TLname);

fflush(stdin);

cout << endl << endl;

cout << "\t\t\t\t\t\t\t\tPhone NO : ";

getline(cin,Tphone);

fflush(stdin);

cout << endl << endl;

cout << "\t\t\t\t\t\t\t\tSalary : ";

getline(cin,Tsalary);

fflush(stdin);

cout << endl << endl;

cout << "\t\t\t\t\t\t\t\tWorking Time ";

cout << "\n\t\t\t\t\t\t\t\tStarting Time :";

getline(cin, Stime);

fflush(stdin);

cout << "\t\t\t\t\t\t\t\tEnding Time :";

getline(cin, Etime);

fflush(stdin);

cout << endl << endl;

temp\_file1<<" "<<TFname<<" "<<TLname<<" "<<Tphone<<" "<<Tsalary<<" "<<Stime<<" "<<Etime<<"\n";

}

else

temp\_file1<<" "<<TFname<<" "<<TLname<<" "<<Tphone<<" "<<Tsalary<<" "<<Stime<<" "<<Etime<<"\n";

file\_obj1>>TFname>>TLname;

file\_obj1>>Tphone>>Tsalary>>Stime>>Etime;

}

file\_obj1.close();

temp\_file1.close();

remove("E://Trainer2nd.txt");

rename("E://temp\_file2nd.txt","E://Trainer2nd.txt");

}

}

void Trainer::del\_trainer()

{

system("cls");

fstream file\_obj1,temp\_file1;

string TFid;

string Tsalary,Tphone,TFname,TLname,temp,Stime,Etime;

cout<<"\n\n\t\t\t\tDelete Member Record";

file\_obj1.open("E://Trainer2nd.txt",ios::in);

temp\_file1.open("E://temp\_file2nd.txt",ios::app|ios::out);

if(!file\_obj1)

cout<<"\n\n File Openning Error...";

else

{

cout<<"\n\n Phone NO: : ";

cin>>TFid;

fflush(stdin);

file\_obj1>>TFname>>TLname;

file\_obj1>>Tphone>>Tsalary>>Stime>>Etime;

while(!file\_obj1.eof())

{

if(TFid==Tphone)

{

system("cls");

cout<<"\n\n\t\t\t\tDeleted Member Record";

cout<<"\n\n One Member is Deleted Successfully...";

}

else

temp\_file1<<" "<<TFname<<" "<<TLname<<" "<<Tphone<<" "<<Tsalary<<" "<<Stime<<" "<<Etime<<"\n";

file\_obj1>>TFname>>TLname;

file\_obj1>>Tphone>>Tsalary>>Stime>>Etime;

}

file\_obj1.close();

temp\_file1.close();

remove("E://Trainer2nd.txt");

rename("E://temp\_file2nd.txt","E://Trainer2nd.txt");

}

}

void Transaction::Minput()

{

system("cls");

cout << "\n\n";

cout << "\n\n\n\n\n\n\n\n\n\t\t\t\t\t\t\t\t\*\*\* Welcome To Monetary DataBase \*\*\*" << endl << endl;

cout << "\t\t\t\t\t\t\t\t----------------------------------------------" << endl << endl;

cout << "\t\t\t\t\t\t\t\t<1> Total Money In Gym " << endl << endl;

cout << "\t\t\t\t\t\t\t\t<2> Detail OF ALL The Monetary Transaction(ALL time) " << endl << endl;

cout << "\t\t\t\t\t\t\t\t<3> Find Specific Transaction " << endl << endl;

cout << "\t\t\t\t\t\t\t\t<4> Go Back " << endl << endl;

cout << "\t\t\t\t\t\t\t\t----------------------------------------------" << endl << endl;

cout<<"\t\t\t\t\t\t\t\t\t Choice : ";

}

void Transaction::show\_Total()

{

fstream file\_obj1,temp\_money,main\_money;

int total,month;

string Mphone;

int grand\_total=0;

file\_obj1.open("E://Monetary.txt",ios::in);

main\_money.open("E://total.txt",ios::app|ios::in);

temp\_money.open("E://temp\_money.txt",ios::app|ios::out);

if(!file\_obj1)

cout<<"\n\n File Openning Error...";

else

{

file\_obj1>>Mphone>>total>>month;

while(!file\_obj1.eof())

{

grand\_total=grand\_total+total;

file\_obj1>>Mphone>>total>>month;

}

system("cls");

graphics();

cout<<"\n\t\t\t\t\t\t\t\t\tTotal Money of Gym:"<<grand\_total;

temp\_money<<" "<<grand\_total;

file\_obj1.close();

}

main\_money.close();

temp\_money.close();

remove("E://total.txt");

rename("E://temp\_money.txt","E://total.txt");

getchar();

}

void Transaction::show\_all()

{

system("cls");

fstream file\_obj1,temp\_money,main\_money;

int total,month;

string Mphone,phno;

file\_obj1.open("E://Monetary.txt",ios::in);

if(!file\_obj1)

cout<<"\n\n File Openning Error...";

else

{

graphics();

cout<<"\n\t\t\t\t\t\t\t\t\tPhone No Total Money Month";

cout << "\n\t\t\t\t\t\t\t\t----------------------------------------------" << endl << endl;

file\_obj1>>Mphone>>total>>month;

while(!file\_obj1.eof())

{

cout<<"\n\t\t\t\t\t\t\t\t\t"<<Mphone<<" "<<total<<" "<<month<<endl;

file\_obj1>>Mphone>>total>>month;

}

file\_obj1.close();

getchar();

}

getchar();

}

void Transaction::find\_Transaction()

{

system("cls");

fstream file\_obj1,temp\_money,main\_money;

int total,i=0,month;

string Mphone,phno;

file\_obj1.open("E://Monetary.txt",ios::in);

if(!file\_obj1)

cout<<"\n\n File Openning Error...";

else

{

graphics();

file\_obj1>>Mphone>>total>>month;

cout<<"\n\t\t\t\t\t\t\t\t\t Phone NO: : ";

cin>>phno;

cout<<"\n\t\t\t\t\t\t\t\t\t Total amount paid by this member upto now\n";

while(!file\_obj1.eof())

{

if(phno==Mphone)

{

++i;

cout<<"\n\t\t\t\t\t\t\t\t\t Transaction "<<i<<":"<<total<<" Month:"<<month<<endl;

}

file\_obj1>>Mphone>>total>>month;

}

getchar();

file\_obj1.close();

}

getchar();

}

int main()

{

adminlogin();

}

void adminlogin()

{

Member member;

Trainer trainer;

Transaction tran;

char choice,mchoice,tchoice,money;

char x,YN;

int grand\_total;

string pass ="";

char ch;

system("cls");

cout << "\n\n\n\n\n\n\n\n\t\t\t\t\t@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@\n";

cout << "\t\t\t\t\t@@ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ @@\n";

cout << "\t\t\t\t\t@@| |@@\n";

cout << "\t\t\t\t\t@@| |@@\n";

cout << "\t\t\t\t\t@@| |@@\n";

cout << "\t\t\t\t\t@@| |@@\n";

cout << "\t\t\t\t\t@@| |@@\n";

cout << "\t\t\t\t\t@@| |@@\n";

cout << "\t\t\t\t\t@@| WELCOME TO |@@\n";

cout << "\t\t\t\t\t@@| |@@\n";

cout << "\t\t\t\t\t@@| GYM MANAGEMENT SYSTEM |@@\n";

cout << "\t\t\t\t\t@@| |@@\n";

cout << "\t\t\t\t\t@@| |@@\n";

cout << "\t\t\t\t\t@@| |@@\n";

cout << "\t\t\t\t\t@@| |@@\n";

cout << "\t\t\t\t\t@@| |@@\n";

cout << "\t\t\t\t\t@@|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|@@\n";

cout << "\t\t\t\t\t@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@\n\n\n\n\t\t\t\t\t";

cout<<"\n\t\t\t\t\t\t\t\t\t\xdb| Admin Login |\xdb";

cout<<"\n\n\t\t\t\t\t\t\t\t\tEnter the password:";

ch = getch();

while(ch != 13)

{

pass.push\_back(ch);

cout << '\*';

ch = getch();

}

if(pass == "admin" || pass =="ADMIN" || pass == "Admin")

{

cout << "\n\n\n\t\t\t\t\t\t\t\t\t\xdb| Access Granted! |\xdb\n";

Sleep(1000);

label:

mainmenu();

cin>>choice;

fflush(stdin);

switch(choice)

{

case '1':

flag9:

system("cls");

member.input();

cin>>mchoice;

fflush(stdin);

switch(mchoice)

{

case '1':

do

{

member.add\_members();

cout<<"\n\t\t\t\t\t\t\t\t\tADD another member(Y/N):";

cin>>YN;

fflush(stdin);

}while(YN=='y'||YN=='Y');

break;

case '2':

member.show\_members();

break;

case '3':

member.find\_members();

break;

case '4':

break;

default:

cout<<"Wrong Choice!!";

goto flag9;

break;

}

break;

case '2':

flag8:

system("cls");

trainer.Tinput();

cin>>tchoice;

fflush(stdin);

switch(tchoice)

{

case '1':

do

{

trainer.add\_Trainer();

cout<<"\n\t\t\t\t\t\t\t\t\tADD another member(Y/N):";

cin>>YN;

fflush(stdin);

}while(YN=='y'||YN=='Y');

break;

case '2':

trainer.show\_Trainer();

break;

case '3':

trainer.find\_Trainer();

break;

case '4':

break;

default:

cout<<"Wrong Choice!!";

goto flag8;

}

break;

case '3':

tran.Minput();

cin>>money;

fflush(stdin);

switch(money)

{

case '1':

tran.show\_Total();

break;

case '2':

tran.show\_all();

break;

case '3':

tran.find\_Transaction();

break;

case '4':

break;

default:

cout<<"Wrong Choice!!";

goto flag8;

}

break;

case '4':

system("cls");

exit(0);

break;

default:

cout<<"Wrong Choice!!";

break;

}

goto label;

}

else

{

cout <<"\n\n\t\t\t\t\t\t\t\t\xdb Access Aborted...\n\t\t\t\t\t\t\t\tPlease Try Again\n\n\xdb";

Sleep(2000);

exit(0);

}

}

void mainmenu()

{

system("cls");

cout<<endl<<endl<<endl<<endl;

cout << "\n\n\n\n\n\n\n\n\n\n\n\n\t\t\t\t\t\t\t\t\*\*\* Welcome to the GYM Management System \*\*\*" << endl << endl;

cout << "\t\t\t\t\t\t\t\t--------------------------------------------" << endl << endl;

cout << "\t\t\t\t\t\t\t\t<1> Membership Details" << endl << endl;

cout << "\t\t\t\t\t\t\t\t<2> Employee Details" << endl << endl;

cout << "\t\t\t\t\t\t\t\t<3> Monetary Details" << endl << endl;

cout << "\t\t\t\t\t\t\t\t<4> Exit" << endl << endl;

cout << "\t\t\t\t\t\t\t\t--------------------------------------------" << endl << endl;

cout << "\t\t\t\t\t\t\t\tEnter Your Choice: ";

}

void editmenu()

{

int choiceED;

Member m1;

cout <<endl<<endl;

cout <<"\t\t\t\t\t\t=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=--=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=" << endl << endl;

cout<<"\n\t\t\t\t\t\t\t\t\t\t1)Edit ";

cout<<"\n\t\t\t\t\t\t\t\t\t\t2)Delete ";

cout<<"\n\t\t\t\t\t\t\t\t\t\t3)Back ";

cout<<endl<<endl<<"\t\t\t\t\t\t=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=--=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=" << endl << endl;

cout<<"\t\t\t\t\t\t\t\t\t\tChoice:";

cin>>choiceED;

fflush(stdin);

switch(choiceED)

{

case 1:

m1.update\_member();

break;

case 2:

m1.del\_members();

break;

case 3:

break;

default:

break;

}

}

void edittrainermenu()

{

int choiceTrainerED;

Trainer t1;

cout <<"\t\t\t\t\t\t=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=--=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=" << endl << endl;

cout<<"\n\t\t\t\t\t\t\t\t\t\t1)Edit ";

cout<<"\n\t\t\t\t\t\t\t\t\t\t2)Delete ";

cout<<"\n\t\t\t\t\t\t\t\t\t\t3)Back ";

cout<<endl<<endl<<"\t\t\t\t\t\t=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=--=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=" << endl << endl;

cout<<"\t\t\t\t\t\t\t\t\t\tChoice:";

cin>>choiceTrainerED;

fflush(stdin);

switch(choiceTrainerED)

{

case 1:

t1.update\_tainer();

break;

case 2:

t1.del\_trainer();

break;

case 3:

break;

default:

break;

}

}

void graphics()

{

cout<<"\n\t\t\t\t\t\t\t \_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_";

cout<<"\n\t\t\t\t\t\t\t | | | |";

cout<<"\n\t\t\t\t\t\t\t \_\_|||||||||| ||||||||||\_\_";

cout<<"\n\t\t\t\t\t\t\t \_| ||||||||||\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|||||||||| |\_";

cout<<"\n\t\t\t\t\t\t\t|\_| ||||||||||\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|||||||||| |\_|";

cout<<"\n\t\t\t\t\t\t\t |\_\_|||||||||| ||||||||||\_\_|";

cout<<"\n\t\t\t\t\t\t\t |||||||||| ||||||||||";

cout<<"\n\t\t\t\t\t\t\t |\_\_\_\_\_\_\_\_| |\_\_\_\_\_\_\_\_|";

cout<<"\n\t\t\t\t\t\t\t=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=\n";

}