

مشروع متكامل - Gym Management System

معلومات المشروع

اسم المشروع: نظام إدارة نادي رياضي (Gym Management System)

اللغة المستخدمة: C++

المستوى: متوسط

المدة المتوقعة: 3-4 أسابيع

نوع المشروع: Console Application with File Handling

الأهداف التعليمية

بنهاية هذا المشروع، ستكون قادراً على:

1. استخدام Structures بشكل احترافي
2. التعامل مع File Handling (Read/Write)
3. تطبيق CRUD Operations (Create, Read, Update, Delete)
4. استخدام Vectors لإدارة البيانات
5. بناء Menu System متكامل
6. معالجة الأخطاء (Error Handling)
7. التحقق من صحة البيانات (Data Validation)
8. إنشاء تقارير وإحصائيات

المطلوب (Functional Requirements)

Module 1: إدارة الأعضاء (Members Management)

1.1 إضافة عضو جديد (Add New Member)

المطلوب:

- إدخال جميع بيانات العضو
- التحقق من عدم تكرار رقم العضو
- حفظ البيانات في ملف members.txt
- عرض رسالة نجاح

البيانات المطلوبة:



- Member ID (GYM001, GYM002, etc.)
- Full Name
- Phone Number (10 digits)
- Email
- Age (18-70)
- Gender (Male/Female)
- Join Date (DD/MM/YYYY)
- Status (Active/Inactive)

Validation Rules:

- متبوعاً بآرقام GYM يجب أن يكون فريداً، بينما بـ: أرقام فقط
- Phone: 10 أرقام فقط
- Age: 70 بين 18 و
- Email: @ يجب أن يحتوي على

1.2 عرض جميع الأعضاء (Display All Members)

المطلوب:

- قراءة البيانات من الملف
- عرضها في جدول منسق
- عرض عدد الأعضاء الكلي
- التمييز بين الأعضاء النشطين وغير النشطين

شكل العرض:



MEMBERS LIST

ID	Name	Phone	Age	Status
GYM001	Ahmad Hassan	0599123456	25	Active
GYM002	Sara Mohammed	0598765432	30	Active
GYM003	Khaled Ali	0597654321	28	Inactive

Total Members: 3 | Active: 2 | Inactive: 1

1.3 البحث عن عضو (Search Member)

المطلوب:

- البحث بواسطة Member ID
- البحث بواسطة الاسم (جزئي)
- البحث بواسطة رقم الهاتف
- عرض النتائج بشكل واضح

1.4 تعديل بيانات عضو (Update Member)

المطلوب:

- البحث عن العضو بواسطة ID
- عرض البيانات الحالية
- السماح بتعديل أي حقل
- حفظ التعديلات في الملف
- عرض رسالة تأكيد

1.5 حذف عضو (Delete Member)

المطلوب:

- البحث عن العضو
- عرض بيانته للتأكد
- طلب تأكيد الحذف (Are you sure? Y/N)
- حذف العضو من الملف
- عرض رسالة نجاح

1.6 تفعيل/تعطيل عضو (Activate/Deactivate)

المطلوب:

- والعكس Inactive إلى Active تغيير حالة العضو من
- حفظ التغيير

Module 2: إدارة الاشتراكات (Memberships Management)

2.1 أنواع الاشتراكات

المطلوب تعريف:



1. Monthly Membership

- Duration: 30 days
- Price: 150 SR

2. Quarterly Membership

- Duration: 90 days
- Price: 400 SR
- Discount: 11%

3. Half-Yearly Membership

- Duration: 180 days
- Price: 750 SR
- Discount: 17%

4. Yearly Membership

- Duration: 365 days
- Price: 1400 SR
- Discount: 22%

2.2 إضافة اشتراك جديد (Add New Membership)

المطلوب:

- التحقق من وجود العضو
- اختيار نوع الاشتراك
- حساب تاريخ الانتهاء تلقائياً
- حساب السعر
- حفظ في memberships.txt

البيانات المطلوبة:



- Membership ID (MEM001, MEM002, etc.)
- Member ID
- Membership Type
- Start Date
- End Date (calculated automatically)
- Price
- Status (Active/Expired)

2.3 عرض جميع الاشتراكات (Display All Memberships)

المطلوب:

- عرض الاشتراكات النشطة
- عرض الاشتراكات المنتهية
- حساب الأيام المتبقية للاشتراكات النشطة

2.4 تجديد اشتراك (Renew Membership)

المطلوب:

- البحث عن الاشتراك
- اختيار نوع الاشتراك الجديد
- حساب تاريخ البداية من تاريخ انتهاء الاشتراك السابق
- حفظ البيانات

2.5 فحص صلاحية الاشتراك (Check Membership Validity)

المطلوب:

- إدخال Member ID
- عرض حالة الاشتراك
- عرض الأيام المتبقية
- تنبيه إذا كان الاشتراك قرب على الانتهاء (أقل من 7 أيام)

Module 3: تسجيل الحضور (Attendance Tracking) 17

3.1 تسجيل حضور (Check-in)

المطلوب:

- إدخال Member ID
- التحقق من صلاحية الاشتراك
- تسجيل التاريخ والوقت
- حفظ في attendance.txt
- عرض رسالة ترحيب

البيانات المطلوبة:



- Attendance ID (ATT001, ATT002, etc.)
- Member ID
- Date
- Check-in Time
- Check-out Time (optional)

3.2 تسجيل خروج (Check-out)

المطلوب:

- إدخال Member ID
- تحديث وقت الخروج
- حساب مدة التمرين

3.3 عرض سجل الحضور (View Attendance Records)

المطلوب:

- عرض سجل حضور عضو معين
- عرض حضور يوم معين
- عرض جميع السجلات

3.4 إحصائيات الحضور (Attendance Statistics)

المطلوب:

- عدد مرات حضور عضو معين خلال فترة
- أكثر الأعضاء حضوراً
- متوسط وقت التمرين
- الأيام الأكثر ازدحاماً

Module 4: إدارة المدفوعات (Payments Management) 💰

4.1 تسجيل دفعه (Record Payment)

المطلوب:

- ربط الدفعه بعضو معين
- تحديد المبلغ
- تحديد طريقة الدفع (Cash/Card/Online)
- تسجيل التاريخ

- حفظ في payments.txt

البيانات المطلوبة:



- Payment ID (PAY001, PAY002, etc.)
- Member ID
- Amount
- Payment Date
- Payment Method
- Description (Membership Fee, etc.)

4.2 عرض سجل المدفوعات (View Payment History)

المطلوب:

- عرض مدفوعات عضو معين
- عرض مدفوعات يوم/شهر معين
- عرض جميع المدفوعات

4.3 تقارير المدفوعات (Payment Reports)

المطلوب:

- إجمالي الإيرادات الشهرية
- إجمالي الإيرادات السنوية
- تصنيف حسب طريقة الدفع
- أعلى المدفوعات

Module 5: التقارير والإحصائيات (Reports & Statistics)

5.1 تقرير عام (General Report)

المطلوب عرض:



- Total Members: XXX
- Active Members: XXX
- Inactive Members: XXX
- Active Memberships: XXX
- Expired Memberships: XXX
- Today's Attendance: XXX
- Monthly Revenue: XXX SR
- Yearly Revenue: XXX SR

5.2 تقرير الأعضاء (Members Report)

المطلوب:

- الأعضاء الجدد هذا الشهر
- الأعضاء حسب الفئة العمرية
- الأعضاء حسب الجنس
- معدل عمر الأعضاء

5.3 تقرير الاشتراكات (Memberships Report)

المطلوب:

- الاشتراكات المنتهية هذا الشهر
- الاشتراكات القريبة من الانتهاء (أقل من 7 أيام)
- أكثر أنواع الاشتراكات مبيعاً
- معدل تجديد الاشتراكات

5.4 تقرير الحضور (Attendance Report)

المطلوب:

- معدل الحضور اليومي
- أكثر الأعضاء التزاماً
- الأيام الأكثر ازدحاماً
- الساعات الأكثر ازدحاماً

5.5 التنبيهات (Alerts)

المطلوب عرض:



! Expiring Soon (< 7 days): XX memberships

! Expired Memberships: XX

! Inactive Members (> 30 days): XX

! Pending Payments: XX

البنية التقنية (Technical Structure)

المطلوبة الـ 1. Structures



```
struct Member {  
    string memberID;  
    string fullName;  
    string phone;  
    string email;  
    int age;  
    string gender;    // "Male" or "Female"  
    string joinDate; // DD/MM/YYYY  
    bool isActive;  
    bool markForDelete; // حذف  
};
```

```
struct Membership {  
    string membershipID;  
    string memberID;  
    string type;      // "Monthly", "Quarterly", "Half-Yearly", "Yearly"  
    string startDate; // DD/MM/YYYY  
    string endDate;  // DD/MM/YYYY  
    double price;  
    bool isActive;  
    bool markForDelete;  
};
```

```
struct Attendance {  
    string attendanceID;  
    string memberID;  
    string date;      // DD/MM/YYYY  
    string checkInTime; // HH:MM  
    string checkOutTime; // HH:MM  
    int duration;    // minutes  
    bool markForDelete;  
};
```

```
struct Payment {  
    string paymentID;  
    string memberID;  
    double amount;  
    string paymentDate; // DD/MM/YYYY  
    string paymentMethod; // "Cash", "Card", "Online"  
    string description;
```

```
    bool markForDelete;  
};
```

الملفات المطلوبة 2.



Project Folder/

```
└── main.cpp      الملف الرئيسي //  
└── members.txt   بيانات الأعضاء //  
└── memberships.txt بيانات الاشتراكات //  
└── attendance.txt سجل الحضور //  
└── payments.txt سجل المدفوعات //  
└── README.md     توثيق المشروع //
```

صيغة حفظ البيانات في الملفات 3.

members.txt:



GYM001###Ahmad Hassan###0599123456###ahmad@email.com###25###Male###01/01/2025###1
GYM002###Sara Mohammed###0598765432###sara@email.com###30###Female###05/01/2025###1

memberships.txt:



MEM001###GYM001###Monthly###01/01/2025###31/01/2025###150.00###1
MEM002###GYM002###Yearly###05/01/2025###05/01/2026###1400.00###1

attendance.txt:



ATT001##GYM001##15/01/2025##08:30##10:00##90

ATT002##GYM002##15/01/2025##09:00##10:30##90

payments.txt:



PAY001##GYM001##150.00##01/01/2025##Cash##Monthly Membership Fee

PAY002##GYM002##1400.00##05/01/2025##Card##Yearly Membership Fee



المطلوب تطويرها Functions

الدوال المساعدة (Utility Functions)



cpp

// 1. تقطيع النص.

```
vector<string> SplitString(string text, string delimiter);
```

// 2. تحويل السطر إلى Structure

```
Member ConvertLineToMember(string line);
Membership ConvertLineToMembership(string line);
Attendance ConvertLineToAttendance(string line);
Payment ConvertLineToPayment(string line);
```

// 3. تحويل إلى سطر Structure

```
string ConvertMemberToLine(Member member);
string ConvertMembershipToLine(Membership membership);
string ConvertAttendanceToLine(Attendance attendance);
string ConvertPaymentToLine(Payment payment);
```

// 4. التحقق من صحة البيانات.

```
bool IsValidPhone(string phone);
bool IsValidEmail(string email);
bool IsValidAge(int age);
bool IsValidDate(string date);
```

// 5. التتحقق من وجود العنصر.

```
bool IsMemberExists(string memberID);
bool IsMembershipExists(string membershipID);
```

// 6. توليد ID تلقائي.

```
string GenerateMemberID();
string GenerateMembershipID();
string GenerateAttendanceID();
string GeneratePaymentID();
```

// 7. حساب الأيام بين تاريخين.

```
int CalculateDaysBetween(string date1, string date2);
```

// 8. الحصول على التاريخ الحالي.

```
string GetCurrentDate();
string GetCurrentTime();
```

// 9. حساب تاريخ الانتهاء.

```
string CalculateEndDate(string startDate, int days);
```

// 10. تنظيف الشاشة والانتظار.

```
void ClearScreen();
void PauseScreen();
void PrintLine(int length = 70);
void PrintHeader(string title);
```

Group 2: File Operations (عمليات الملفات)



cpp

// قراءة البيانات من الملفات . 1.

```
vector<Member> LoadMembersFromFile();
vector<Membership> LoadMembershipsFromFile();
vector<Attendance> LoadAttendanceFromFile();
vector<Payment> LoadPaymentsFromFile();
```

// حفظ البيانات في الملفات . 2.

```
void SaveMembersToFile(vector<Member> members);
void SaveMembershipsToFile(vector<Membership> memberships);
void SaveAttendanceToFile(vector<Attendance> attendanceList);
void SavePaymentsToFile(vector<Payment> payments);
```

// إضافة سطر جديد للملف . 3.

```
void AddLineToFile(string filename, string line);
```

// تحديث البيانات في الملف . 4.

```
void UpdateMemberInFile(Member member);
void UpdateMembershipInFile(Membership membership);
```

Group 3: Members Functions (دوال الأعضاء)



cpp

// 1. قراءة بيانات عضو جديد.

Member ReadNewMember();

// إضافة عضو جديد. 2.

void AddNewMember();

// عرض جميع الأعضاء. 3.

void ShowAllMembers();

// طباعة بيانات عضو في سطر جدول. 4.

void PrintMemberLine(Member member);

// طباعة بطاقة عضو كاملة. 5.

void PrintMemberCard(Member member);

// البحث عن عضو. 6.

Member FindMemberByID(string memberID);
vector<Member> FindMembersByName(string name);
Member FindMemberByPhone(string phone);

// تغيير بيانات عضو. 7.

void UpdateMember();

// حذف عضو. 8.

void DeleteMember();

// تفعيل/تعطيل عضو. 9.

void ToggleMemberStatus();

// 10. شاشات العرض.

void ShowAddMemberScreen();
void ShowAllMembersScreen();
void ShowSearchMemberScreen();
void ShowUpdateMemberScreen();
void ShowDeleteMemberScreen();

Group 4: Memberships Functions (دوال الاشتراكات)



cpp

// 1. عرض أنواع الاشتراكات.

void ShowMembershipTypes();

// الحصول على سعر الاشتراك . 2.

double GetMembershipPrice(string type);

// الحصول على مدة الاشتراك . 3.

int GetMembershipDuration(string type);

قراءة بيانات اشتراك جديد . 4.

Membership ReadNewMembership();

// إضافة اشتراك جديد . 5.

void AddNewMembership();

// عرض جميع الاشتراكات . 6.

void ShowAllMemberships();

// طباعة بيانات اشتراك . 7.

void PrintMembershipLine(Membership membership);

void PrintMembershipCard(Membership membership);

// البحث عن اشتراك . 8.

Membership FindMembershipByID(string membershipID);

vector<Membership> FindMembershipsByMemberID(string memberID);

// فحص صلاحية الاشتراك . 9.

bool IsMembershipValid(string memberID);

int GetRemainingDays(Membership membership);

// تجديد اشتراك . 10.

void RenewMembership();

// عرض الاشتراكات القريبة من الانتهاء . 11.

void ShowExpiringSoonMemberships();

// تحديث حالة الاشتراكات (منتهية/نشطة) . 12.

void UpdateMembershipsStatus();

// شاشات العرض . 13.

void ShowAddMembershipScreen();

```
void ShowAllMembershipsScreen();  
void ShowCheckMembershipScreen();  
void ShowRenewMembershipScreen();
```

Group 5: Attendance Functions (دوال الحضور)



cpp

```
// تسجيل حضور .1
void CheckIn();

// تسجيل خروج .2
void CheckOut();

// عرض سجل الحضور .3
void ShowAttendanceRecords();

// طباعة سجل حضور .4
void PrintAttendanceLine(Attendance attendance);

// البحث عن سجلات حضور عضو .5
vector<Attendance> GetMemberAttendance(string memberID);

// البحث عن سجلات حضور يوم معين .6
vector<Attendance> GetDayAttendance(string date);

// حساب عدد مرات الحضور .7
int CountMemberAttendance(string memberID, string fromDate, string toDate);

// حساب متوسط مدة التمرين .8
double CalculateAverageWorkoutDuration(string memberID);

// شاشات العرض .9
void ShowCheckInScreen();
void ShowCheckOutScreen();
void ShowAttendanceHistoryScreen();
void ShowAttendanceStatisticsScreen();
```

Group 6: Payments Functions (دوال المدفوعات)



cpp

```

// 1. قراءة بيانات دفعه جديدة.
Payment ReadNewPayment();

// 2. تسجيل دفعه.
void RecordPayment();

// 3. عرض سجل المدفوعات.
void ShowAllPayments();

// 4. طباعة بيانات دفعه.
void PrintPaymentLine(Payment payment);
void PrintPaymentReceipt(Payment payment);

// 5. البحث عن مدفوعات عضو.
vector<Payment> GetMemberPayments(string memberID);

// 6. حساب إجمالي مدفوعات عضو.
double CalculateTotalPayments(string memberID);

// 7. حساب الإيرادات الشهرية.
double CalculateMonthlyRevenue(int month, int year);

// 8. حساب الإيرادات السنوية.
double CalculateYearlyRevenue(int year);

// 9. شاشات العرض.
void ShowRecordPaymentScreen();
void ShowPaymentsHistoryScreen();
void ShowPaymentsReportScreen();

```

Group 7: Reports Functions (دوال التقارير)



cpp

```

// 1. تقرير عام
void ShowGeneralReport();

// 2. تقرير الأعضاء
void ShowMembersReport();

// 3. تقرير الاشتراكات
void ShowMembershipsReport();

// 4. تقرير الحضور
void ShowAttendanceReport();

// 5. تقرير المدفوعات
void ShowPaymentsReport();

// 6. إحصائيات الأعضاء
int CountActiveMembers();
int CountInactiveMembers();
double CalculateAverageAge();

// 7. إحصائيات الاشتراكات
int CountActiveMemberships();
int CountExpiredMemberships();
int CountExpiringSoonMemberships();

// 8. أكثر نوع اشتراك مبيعاً
string GetMostPopularMembershipType();

// 9. أكثر الأعضاء حضوراً
vector<pair<string, int>> GetTopAttendingMembers(int count);

// 10. عرض التنبيهات
void ShowAlerts();

```

Group 8: Menu Functions (دوال القوائم)



cpp

```
// 1. القائمة الرئيسية.  
void ShowMainMenu();  
  
// قائمة إدارة الأعضاء. 2.  
void ShowMembersMenu();  
  
// قائمة إدارة الاشتراكات. 3.  
void ShowMembershipsMenu();  
  
// قائمة الحضور. 4.  
void ShowAttendanceMenu();  
  
// 5. قائمة المدفوعات.  
void ShowPaymentsMenu();  
  
// 6. قائمة التقارير.  
void ShowReportsMenu();  
  
// قراءة خيار القائمة. 7.  
int ReadMenuItemOption(int min, int max);  
  
// 8. تنفيذ خيار القائمة.  
void PerformMainMenuOption(int option);  
void PerformMembersMenuOption(int option);  
void PerformMembershipsMenuOption(int option);  
void PerformAttendanceMenuOption(int option);  
void PerformPaymentsMenuOption(int option);  
void PerformReportsMenuOption(int option);  
  
// العودة للقائمة الرئيسية. 9.  
void GoBackToMainMenu();  
void GoBackToMenu();
```

🎨 تصميم واجهات العرض (UI Design)

1. القائمة الرئيسية.



GYM MANAGEMENT SYSTEM

Welcome, Admin!

Date: 29/10/2025

- [1] Members Management
- [2] Memberships Management
- [3] Attendance Tracking
- [4] Payments Management
- [5] Reports & Statistics
- [6] Settings
- [0] Exit System

Choose your option [0-6]: _

قائمة إدارة الأعضاء .2





MEMBERS MANAGEMENT MODULE

- [1] Add New Member
- [2] Display All Members
- [3] Search for Member
- [4] Update Member Information
- [5] Delete Member
- [6] Activate/Deactivate Member
- [0] Back to Main Menu

Choose your option [0-6]: _

شاشة إضافة عضو جديد. 3.



+ ADD NEW MEMBER

Enter Member Information:

Full Name : _____
 Phone Number : _____
 Email Address : _____
 Age : _____
 Gender (M/F) : _____

[S] Save Member [C] Cancel [R] Reset

Your choice: _

عرض جميع الأعضاء .4.



ALL MEMBERS LIST

ID	Full Name	Phone	Age	Gender	Status	Join
GYM001	Ahmad Hassan	0599123456	25	Male	<input checked="" type="checkbox"/> Active	01/01
GYM002	Sara Mohammed	0598765432	30	Female	<input checked="" type="checkbox"/> Active	05/01
GYM003	Khaled Ali	0597654321	28	Male	<input checked="" type="checkbox"/> Inactive	10/12
GYM004	Layla Ahmed	0596543210	22	Female	<input checked="" type="checkbox"/> Active	15/01
GYM005	Omar Youssef	0595432109	35	Male	<input checked="" type="checkbox"/> Active	20/01

Statistics:

Total Members: 5 | Active: 4 (80%) | Inactive: 1 (20%)

Press any key to return to menu...

بطاقة عضو .5. (Member Card)



MEMBER CARD

Member ID : GYM001
Full Name : Ahmad Hassan
Phone : 0599123456
Email : ahmad@email.com
Age : 25 years
Gender : Male
Join Date : 01/01/2025
Status : Active

Member Statistics:

— Total Attendance : 45 times
— Current Membership : Yearly (Active)
— Days Remaining : 335 days
— Total Payments : 1,400 SR

شاشة التقرير العام .6



GENERAL SYSTEM REPORT

Date: 29/10/2025

MEMBERS OVERVIEW

- Total Members : 150
- Active Members : 135 (90%)
- Inactive Members : 15 (10%)
- New Members This Month : 12
- Average Age : 28 years

MEMBERSHIPS OVERVIEW

- Active Memberships : 128
- Expired Memberships : 22
- Expiring Soon (< 7 days) : 8 
- Most Popular Type : Yearly (45%)
- Renewal Rate : 78%

ATTENDANCE OVERVIEW

- Today's Attendance : 45 members
- This Week's Average : 52 members/day
- This Month's Total : 1,247 check-ins
- Average Workout Time : 85 minutes

FINANCIAL OVERVIEW

- Today's Revenue : 450 SR
- This Month's Revenue : 18,750 SR
- This Year's Revenue : 225,000 SR
- Average Revenue/Member : 1,500 SR

ALERTS & NOTIFICATIONS

- 8 memberships expiring in the next 7 days
- 22 expired memberships need renewal
- 15 members inactive for more than 30 days

[R] Refresh Report [E] Export to File [B] Back to Menu

17 خطة التنفيذ التفصيلية (4 أسابيع)

الأسبوع الأول: البنية الأساسية والأعضاء

اليوم 1-2: الإعداد والهيكل الأساسي

المهام:

1. إنشاء المشروع وملف main.cpp

2. تعريف جميع الـ Structures

3. كتابة Utility Functions:

- SplitString()
- ConvertLineToMember()
- ConvertMemberToLine()
- IsValidPhone()
- IsValidEmail()
- IsValidAge()

4. اختبار الدوال المساعدة

الكود المطلوب:



cpp

```

#include <iostream>
#include <fstream>
#include <string>
#include <vector>
#include <iomanip>
using namespace std;

// Constants
const string MEMBERS_FILE = "members.txt";
const string SEPARATOR = "#/#/";

// Structures
struct Member {
    string memberID;
    string fullName;
    string phone;
    string email;
    int age;
    string gender;
    string joinDate;
    bool isActive;
    bool markForDelete;
};

// Utility Functions
vector<string> SplitString(string text, string delimiter) {
    // Implementation here
}

// باقي الدول ...

```

اليوم 3-4: إدارة الأعضاء - الجزء الأول

المهام:

1. LoadMembersFromFile()
2. SaveMembersToFile()
3. GenerateMemberID()
4. ReadNewMember()
5. AddNewMember()

6. اختبار إضافة الأعضاء

اليوم 5-6: إدارة الأعضاء - الجزء الثاني

المهام:

1. ShowAllMembers()

2. PrintMemberLine()

3. FindMemberByID()

4. UpdateMember()

5. DeleteMember()

6. اختبار جميع عمليات الأعضاء

اليوم 7: القوائم والواجهات

المهام:

1. تصميم ShowMainMenu()

2. تصميم ShowMembersMenu()

3. ربط جميع دوال الأعضاء بالقوائم

4. الأول اختبار شامل للـ Module

Deliverable: للأسبوع الأول

- نظام إدارة أعضاء كامل وعامل

- للأعضاء CRUD جميع عمليات

- واجهات مستخدم منسقة

الأسبوع الثاني: الاشتراكات والمدفوعات

اليوم 8-9: إدارة الاشتراكات - الجزء الأول

المهام:

1. تعریف أنواع الاشتراکات والأسعار

2. ConvertLineToMembership()

3. ConvertMembershipToLine()

4. LoadMembershipsFromFile()

5. SaveMembershipsToFile()

6. GenerateMembershipID()

7. GetMembershipPrice()

8. GetMembershipDuration()

اليوم 10-11: إدارة الاشتراكات - الجزء الثاني

المهام:

1. CalculateEndDate()
2. ReadNewMembership()
3. AddNewMembership()
4. ShowAllMemberships()
5. IsMembershipValid()
6. GetRemainingDays()
7. اختبار إضافة الاشتراكات

اليوم 12-13: المدفوعات

المهام:

1. ConvertLineToPayment()
2. ConvertPaymentToLine()
3. LoadPaymentsFromFile()
4. SavePaymentsToFile()
5. GeneratePaymentID()
6. RecordPayment()
7. ShowAllPayments()
8. CalculateTotalPayments()

اليوم 14: القوائم والاختبار

المهام:

1. ShowMembershipsMenu()
2. ShowPaymentsMenu()
3. ربط جميع الدوال
4. اختبار شامل

Deliverable: للأسبوع الثاني

- نظام اشتراكات كامل
- نظام مدفوعات كامل
- ربط الاشتراكات بالأعضاء

الأسبوع الثالث: الحضور والتقارير

اليوم 15-16: نظام الحضور

المهام:

1. ConvertLineToAttendance()
2. ConvertAttendanceToLine()
3. LoadAttendanceFromFile()
4. SaveAttendanceToFile()
5. GetCurrentTime()
6. CheckIn()
7. CheckOut()
8. ShowAttendanceRecords()

اليوم 17-18: إحصائيات الحضور

المهام:

1. CountMemberAttendance()
2. CalculateAverageWorkoutDuration()
3. GetTopAttendingMembers()
4. ShowAttendanceStatistics()
5. اختبار نظام الحضور

اليوم 19-20: التقارير الأساسية

المهام:

1. CountActiveMembers()
2. CountInactiveMembers()
3. CountActiveMemberships()
4. CountExpiredMemberships()
5. CalculateMonthlyRevenue()
6. ShowGeneralReport()
7. ShowMembersReport()

اليوم 21: القوائم والاختبار

المهام:

1. ShowAttendanceMenu()
2. ShowReportsMenu()

3. اختبار شامل

للأسبوع الثالث Deliverable:

- نظام حضور كامل
- تقارير وإحصائيات
- متعلقة Modules جميع الـ

الأسبوع الرابع: التحسينات والإنتهاء

اليوم 22-23: ميزات متقدمة

المهام:

1. RenewMembership()
2. ShowExpiringSoonMemberships()
3. ShowAlerts()
4. UpdateMembershipsStatus() (تلقائي)
5. تحسين Input Validation
6. تحسين Error Handling

اليوم 24-25: تحسين الواجهات

المهام:

1. تحسين جميع القوائم
2. إضافة ألوان ورموز
3. تحسين رسائل الخطأ
4. إضافة رسائل النجاح
5. تحسين التنسيق

اليوم 26-27: الاختبار الشامل

المهام:

1. اختبار جميع الـ Modules
2. اختبار الحالات الخاصة
3. إصلاح الأخطاء (Bugs)
4. اختبار الأداء
5. اختبار سيناريوهات الاستخدام

اليوم 28: التوثيق والإنتهاء

المهام:

1. شامل README.md كتابة
2. إضافة تعليقات للكود
3. إنشاء دليل المستخدم
4. أخذ Screenshots
5. رفع المشروع على GitHub

Deliverable: النهائي

- مشروع كامل 100%
- ت العمل Features جميع الـ
- توثيق شامل
- جاهز للعرض

✓ معايير التقييم (Grading Criteria)

الوظائف الأساسية (40 نقطة) 1.

المعيار	النقط
كاملة CRUD (CRUD) إدارة الأعضاء	10
إدارة الاشتراكات كاملة	10
نظام الحضور يعمل	8
نظام المدفوعات يعمل	7
File Handling صحيح	5

جودة الكود (25 نقطة) 2.

المعيار	النقط
تنظيم الكود والـ Functions	8
استخدام Structures بشكل صحيح	5
التعليقات والتوثيق	5
تسمية المتغيرات واضحة	4
عدم وجود Code Duplication	3

التحقق من البيانات (15 نقطة) 3.

المعيار	النقط
لأعضاء Input Validation	5
للاشتراكات Input Validation	4
جيـد Error Handling	4
رسائل خطأ واضحة	2

التقارير والإحصائيات (10 نقاط) .4.

الميزة	النقط
التقرير العام	3
تقارير الأعضاء والاشتراكات	3
إحصائيات الحضور	2
التنبيهات	2

واجهة المستخدم (10 نقاط) .5.

المعيار	النقط
تصميم القوائم	4
سهولة الاستخدام	3
التنسيق والعرض	3

نصائح مهمة للنجاح

✓ DO (افعل):

- والدوال المساعدة أولاً **Structures** - اكتب الـ 1.
- بعد كتابتها **Function** - اختبر كل 2.
- احفظ التقدم بانتظام - **Git** استخدم 3.
- اكتب تعليقات - وثّق الكود أثناء الكتابة 4.
- اسأل عند الحاجة - لا تتردد في طلب المساعدة 5.

✗ DON'T (لا تفعل):

- لأنسخ كود جاهز - اكتب بنفسك 1.
- لاتترك الاختبار للنهاية - اختبر باستمرار 2.
- لاتهمل التوثيق - مهم جداً 3.
- لاتكتب كل الكود دفعة واحدة - خطوة بخطوة 4.
- لاتؤجل المهام - التزم بالجدول 5.

ملاحظات إضافية

حول File Handling:



cpp

// عند الكتابة، استخدم //
ios::out // الكتابة (overwrite)
ios::app // للإضافة (append)
ios::in // للقراءة

// مثال:
fstream file;
file.open("members.txt", ios::app); // إضافة للنهاية

حول Date Handling:



cpp

// استخدم string للتاريخ بصيغة DD/MM/YYYY
// مثال:
string date = "29/10/2025";

// للمقارنة، قد تحتاج تحويلها لأرقام
// أو استخدام مكتبة <ctime>

حول Input Validation:



cpp

```
// دائمًاً تحقق من المدخلات //
while (true) {
    cout << "Enter age: ";
    cin >> age;

    if (cin.fail() || age < 18 || age > 70) {
        cout << "✖ Invalid age! Try again.\n";
        cin.clear();
        cin.ignore(10000, '\n');
    } else {
        break;
    }
}
```

المطلوب محتوى ملف README.md

يحتوي على README.md عند إنتهاء المشروع، يجب إنشاء ملف



markdown

🏋️‍♀️ Gym Management System

📖 Description

A comprehensive console-based gym management system built with C++. Manages members, memberships, attendance tracking, and payments.

✨ Features

- Complete member management (CRUD operations)
- Membership plans (Monthly, Quarterly, Half-Yearly, Yearly)
- Attendance tracking with check-in/check-out
- Payment management and revenue tracking
- Comprehensive reports and statistics
- Alerts for expiring memberships

💻 Technologies

- Language: C++
- Data Storage: File Handling (Text Files)
- Concepts: Structures, Vectors, File I/O

📋 Requirements

- C++ Compiler (g++, MinGW, or Visual Studio)
- No external libraries required

🚀 How to Run

1. Clone the repository
2. Compile: `g++ main.cpp -o gym`
3. Run: `./gym`

📂 File Structure

- `main.cpp` - Main program file
- `members.txt` - Members data
- `memberships.txt` - Memberships data
- `attendance.txt` - Attendance records
- `payments.txt` - Payment records

💬 Author

[Your Name]

📅 Project Timeline

Duration: 4 weeks

Completed: [Date]

🎉 Learning Outcomes

- Mastered file handling in C++
 - Implemented complex data structures
 - Built real-world application
 - Improved problem-solving skills
-

🎉 الخلاصة

هذا المشروع سيعطيك:

- ✅ خبرة عملية قوية
- ✅ مشروع محترم لـ Portfolio
- ✅ فهم عميق لـ C++
- ✅ مهارات حل المشكلات
- ✅ شيء تفتخر فيه!

🚀 استمتع بالبرمجة وبالتفويف

الدعم والمساعدة 🔍

إذا واجهت أي مشكلة:

1. راجع الكود خطوة بخطوة.
2. استخدم cout لـ debugging
3. اسأل زملاءك.
4. ابحث عن الخطأ على الإنترنت.
5. اسأل الدكتور.

Remember:

"The only way to learn programming is by writing programs."

- Dennis Ritchie

👉 !الآن، ابدأ واتكتب أول سطر كود