#include <stdio.h>

#include <stdlib.h>

// Structure definitions

typedef struct {

    int id;

    char name[100];

    int age;

    char gender;

    char disease[100];

} Patient;

typedef struct {

    int id;

    char name[100];

    char specialty[100];

    int experience;

} Doctor;

typedef struct {

    int id;

    int patientId;

    int doctorId;

    char date[20];

} Appointment;

// Function prototypes

void mainMenu();

void patientManagement();

void doctorManagement();

void appointmentScheduling();

void billingSystem();

void addPatient();

void viewPatients();

void editPatient();

void deletePatient();

void addDoctor();

void viewDoctors();

void editDoctor();

void deleteDoctor();

void scheduleAppointment();

void viewAppointments();

void cancelAppointment();

void generateBill();

void viewBills();

// Global variables

Patient patients[100];

Doctor doctors[50];

Appointment appointments[100];

int patientCount = 0;

int doctorCount = 0;

int appointmentCount = 0;

int main() {

    int choice;

    do {

        mainMenu();

        scanf("%d", &choice);

        switch (choice) {

            case 1: patientManagement(); break;

            case 2: doctorManagement(); break;

            case 3: appointmentScheduling(); break;

            case 4: billingSystem(); break;

            case 5: printf("Exiting...\n"); break;

            default: printf("Invalid choice!\n"); break;

        }

    } while (choice != 5);

    return 0;

}

void mainMenu() {

    printf("\n====================================\n");

    printf("|      Hospital Management System   |\n");

    printf("====================================\n");

    printf("|  1. Patient Management            |\n");

    printf("|  2. Doctor Management             |\n");

    printf("|  3. Appointment Scheduling        |\n");

    printf("|  4. Billing                       |\n");

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

    printf("====================================\n");

    printf("Enter your choice: ");

}

// Patient Management Functions

void patientManagement() {

    int choice;

    printf("\n--- Patient Management ---\n");

    printf("1. Add Patient\n");

    printf("2. View Patients\n");

    printf("3. Edit Patient\n");

    printf("4. Delete Patient\n");

    printf("5. Back to Main Menu\n");

    printf("Enter your choice: ");

    scanf("%d", &choice);

    switch (choice) {

        case 1: addPatient(); break;

        case 2: viewPatients(); break;

        case 3: editPatient(); break;

        case 4: deletePatient(); break;

        case 5: return;

        default: printf("Invalid choice!\n"); break;

    }

}

void addPatient() {

    printf("Enter Patient ID: ");

    scanf("%d", &patients[patientCount].id);

    printf("Enter Patient Name: ");

    scanf("%s", patients[patientCount].name);

    printf("Enter Age: ");

    scanf("%d", &patients[patientCount].age);

    printf("Enter Gender (M/F): ");

    scanf(" %c", &patients[patientCount].gender);

    printf("Enter Disease: ");

    scanf("%s", patients[patientCount].disease);

    patientCount++;

    printf("Patient added successfully!\n");

}

void viewPatients() {

    for (int i = 0; i < patientCount; i++) {

        printf("\nID: %d\n", patients[i].id);

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

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

        printf("Gender: %c\n", patients[i].gender);

        printf("Disease: %s\n", patients[i].disease);

    }

}

void editPatient() {

    int id, found = -1;

    printf("Enter Patient ID to edit: ");

    scanf("%d", &id);

    for (int i = 0; i < patientCount; i++) {

        if (patients[i].id == id) {

            found = i;

            break;

        }

    }

    if (found != -1) {

        printf("Enter new Name: ");

        scanf("%s", patients[found].name);

        printf("Enter new Age: ");

        scanf("%d", &patients[found].age);

        printf("Enter new Gender (M/F): ");

        scanf(" %c", &patients[found].gender);

        printf("Enter new Disease: ");

        scanf("%s", patients[found].disease);

        printf("Patient updated successfully!\n");

    } else {

        printf("Patient not found!\n");

    }

}

void deletePatient() {

    int id, found = -1;

    printf("Enter Patient ID to delete: ");

    scanf("%d", &id);

    for (int i = 0; i < patientCount; i++) {

        if (patients[i].id == id) {

            found = i;

            break;

        }

    }

    if (found != -1) {

        for (int i = found; i < patientCount - 1; i++) {

            patients[i] = patients[i + 1];

        }

        patientCount--;

        printf("Patient deleted successfully!\n");

    } else {

        printf("Patient not found!\n");

    }

}

// Doctor Management Functions

void doctorManagement() {

    int choice;

    printf("\n--- Doctor Management ---\n");

    printf("1. Add Doctor\n");

    printf("2. View Doctors\n");

    printf("3. Edit Doctor\n");

    printf("4. Delete Doctor\n");

    printf("5. Back to Main Menu\n");

    printf("Enter your choice: ");

    scanf("%d", &choice);

    switch (choice) {

        case 1: addDoctor(); break;

        case 2: viewDoctors(); break;

        case 3: editDoctor(); break;

        case 4: deleteDoctor(); break;

        case 5: return;

        default: printf("Invalid choice!\n"); break;

    }

}

void addDoctor() {

    printf("Enter Doctor ID: ");

    scanf("%d", &doctors[doctorCount].id);

    printf("Enter Doctor Name: ");

    scanf("%s", doctors[doctorCount].name);

    printf("Enter Specialty: ");

    scanf("%s", doctors[doctorCount].specialty);

    printf("Enter Experience (in years): ");

    scanf("%d", &doctors[doctorCount].experience);

    doctorCount++;

    printf("Doctor added successfully!\n");

}

void viewDoctors() {

    for (int i = 0; i < doctorCount; i++) {

        printf("\nID: %d\n", doctors[i].id);

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

        printf("Specialty: %s\n", doctors[i].specialty);

        printf("Experience: %d years\n", doctors[i].experience);

    }

}

void editDoctor() {

    int id, found = -1;

    printf("Enter Doctor ID to edit: ");

    scanf("%d", &id);

    for (int i = 0; i < doctorCount; i++) {

        if (doctors[i].id == id) {

            found = i;

            break;

        }

    }

    if (found != -1) {

        printf("Enter new Name: ");

        scanf("%s", doctors[found].name);

        printf("Enter new Specialty: ");

        scanf("%s", doctors[found].specialty);

        printf("Enter new Experience (in years): ");

        scanf("%d", &doctors[found].experience);

        printf("Doctor updated successfully!\n");

    } else {

        printf("Doctor not found!\n");

    }

}

void deleteDoctor() {

    int id, found = -1;

    printf("Enter Doctor ID to delete: ");

    scanf("%d", &id);

    for (int i = 0; i < doctorCount; i++) {

        if (doctors[i].id == id) {

            found = i;

            break;

        }

    }

    if (found != -1) {

        for (int i = found; i < doctorCount - 1; i++) {

            doctors[i] = doctors[i + 1];

        }

        doctorCount--;

        printf("Doctor deleted successfully!\n");

    } else {

        printf("Doctor not found!\n");

    }

}

// Appointment Scheduling Functions

void appointmentScheduling() {

    int choice;

    printf("\n--- Appointment Scheduling ---\n");

    printf("1. Schedule Appointment\n");

    printf("2. View Appointments\n");

    printf("3. Cancel Appointment\n");

    printf("4. Back to Main Menu\n");

    printf("Enter your choice: ");

    scanf("%d", &choice);

    switch (choice) {

        case 1: scheduleAppointment(); break;

        case 2: viewAppointments(); break;

        case 3: cancelAppointment(); break;

        case 4: return;

        default: printf("Invalid choice!\n"); break;

    }

}

void scheduleAppointment() {

    printf("Enter Appointment ID: ");

    scanf("%d", &appointments[appointmentCount].id);

    printf("Enter Patient ID: ");

    scanf("%d", &appointments[appointmentCount].patientId);

    printf("Enter Doctor ID: ");

    scanf("%d", &appointments[appointmentCount].doctorId);

    printf("Enter Date (DD/MM/YYYY): ");

    scanf("%s", appointments[appointmentCount].date);

    appointmentCount++;

    printf("Appointment scheduled successfully!\n");

}

void viewAppointments() {

    for (int i = 0; i < appointmentCount; i++) {

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

        printf("Patient ID: %d\n", appointments[i].patientId);

        printf("Doctor ID: %d\n", appointments[i].doctorId);

        printf("Date: %s\n", appointments[i].date);

    }

}

void cancelAppointment() {

    int id, found = -1;

    printf("Enter Appointment ID to cancel: ");

    scanf("%d", &id);

    for (int i = 0; i < appointmentCount; i++) {

        if (appointments[i].id == id) {

            found = i;

            break;

        }

    }

    if (found != -1) {

        for (int i = found; i < appointmentCount - 1; i++) {

            appointments[i] = appointments[i + 1];

        }

        appointmentCount--;

        printf("Appointment canceled successfully!\n");

    } else {

        printf("Appointment not found!\n");

    }

}

// Billing System Functions

void billingSystem() {

    int choice;

    printf("\n--- Billing System ---\n");

    printf("1. Generate Bill\n");

    printf("2. View Bills\n");

    printf("3. Back to Main Menu\n");

    printf("Enter your choice: ");

    scanf("%d", &choice);

    switch (choice) {

        case 1: generateBill(); break;

        case 2: viewBills(); break;

        case 3: return;

        default: printf("Invalid choice!\n"); break;

    }

}

void generateBill() {

    int patientId, found = -1;

    float consultationFee, treatmentCharges, totalAmount;

    printf("Enter Patient ID for billing: ");

    scanf("%d", &patientId);

    for (int i = 0; i < patientCount; i++) {

        if (patients[i].id == patientId) {

            found = i;

            break;

        }

    }

    if (found != -1) {

        printf("Enter Consultation Fee: ");

        scanf("%f", &consultationFee);

        printf("Enter Treatment Charges: ");

        scanf("%f", &treatmentCharges);

        totalAmount = consultationFee + treatmentCharges;

        printf("\n--- Bill Details ---\n");

        printf("Patient ID: %d\n", patients[found].id);

        printf("Patient Name: %s\n", patients[found].name);

        printf("Consultation Fee: %.2f\n", consultationFee);

        printf("Treatment Charges: %.2f\n", treatmentCharges);

        printf("Total Amount: %.2f\n", totalAmount);

    } else {

        printf("Patient not found!\n");

    }

}

void viewBills() {

    printf("Feature not implemented yet.\n");

}