#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX\_PATIENTS 100

struct Patient {

int id;

char name[100];

int age;

char gender[10];

};

struct Appointment {

int patientId;

char doctorName[100];

char date[20];

struct Appointment\* next;

};

struct Patient patients[MAX\_PATIENTS];

int patientCount = 0;

struct Appointment\* appointmentHead = NULL;

void addPatient();

void displayPatients();

void searchPatient();

void addAppointment();

void displayAppointments();

void deleteAppointment();

void addPatient() {

if (patientCount >= MAX\_PATIENTS) {

printf("Maximum patient limit reached.\n");

return;

}

printf("Enter patient name: ");

scanf(" %[^\n]", patients[patientCount].name);

printf("Enter age: ");

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

printf("Enter gender: ");

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

patients[patientCount].id = patientCount + 1;

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

patientCount++;

}

void displayPatients() {

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

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

printf("ID: %d, Name: %s, Age: %d, Gender: %s\n",

patients[i].id, patients[i].name,

patients[i].age, patients[i].gender);

}

}

void searchPatient() {

int choice, id;

char name[100];

printf("Search by:\n1. ID\n2. Name\nEnter choice: ");

scanf("%d", &choice);

if (choice == 1) {

printf("Enter patient ID: ");

scanf("%d", &id);

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

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

printf("Found: %s, Age: %d, Gender: %s\n",

patients[i].name, patients[i].age, patients[i].gender);

return;

}

}

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

} else if (choice == 2) {

printf("Enter patient name: ");

scanf(" %[^\n]", name);

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

if (strcmp(patients[i].name, name) == 0) {

printf("Found: ID: %d, Age: %d, Gender: %s\n",

patients[i].id, patients[i].age, patients[i].gender);

return;

}

}

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

} else {

printf("Invalid choice.\n");

}

}

void addAppointment() {

int patientId;

printf("Enter patient ID for appointment: ");

scanf("%d", &patientId);

if (patientId < 1 || patientId > patientCount) {

printf("Invalid patient ID.\n");

return;

}

struct Appointment\* newAppt = (struct Appointment\*)malloc(sizeof(struct Appointment));

newAppt->patientId = patientId;

printf("Enter doctor's name: ");

scanf(" %[^\n]", newAppt->doctorName);

printf("Enter appointment date (dd-mm-yyyy): ");

scanf(" %s", newAppt->date);

newAppt->next = appointmentHead;

appointmentHead = newAppt;

printf("Appointment scheduled.\n");

}

void displayAppointments() {

struct Appointment\* temp = appointmentHead;

printf("\n--- Appointments List ---\n");

if(temp !=NULL){

while (temp != NULL) {

printf("Patient ID: %d, Doctor: %s, Date: %s\n",

temp->patientId, temp->doctorName, temp->date);

temp = temp->next;

}

}

else{

printf("No Appointments Scheduled.");

}

}

void deleteAppointment() {

struct Appointment\* temp = appointmentHead;

struct Appointment\* prev = NULL;

if(temp !=NULL){

int patientId;

printf("Enter patient ID to delete appointment: ");

scanf("%d", &patientId);

while (temp != NULL) {

if (temp->patientId == patientId) {

if (prev == NULL) {

appointmentHead = temp->next;

} else {

prev->next = temp->next;

}

free(temp);

printf("Appointment deleted.\n");

return;

}

prev = temp;

temp = temp->next;

}

}

else{

printf("No appointments found.\n");

}

}

int main() {

int choice;

while (1) {

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

printf("1. Add Patient\n");

printf("2. Display Patients\n");

printf("3. Search Patient\n");

printf("4. Add Appointment\n");

printf("5. Display Appointments\n");

printf("6. Delete Appointment\n");

printf("7. Exit\n");

printf("Enter choice: ");

scanf("%d", &choice);

switch (choice) {

case 1: addPatient(); break;

case 2: displayPatients(); break;

case 3: searchPatient(); break;

case 4: addAppointment(); break;

case 5: displayAppointments(); break;

case 6: deleteAppointment(); break;

case 7: printf("Exiting Program...");

exit(0);

default: printf("Invalid choice.\n");

}

}

return 0;

}