#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

#include <windows.h>

int slots[5][5] = {

{3, 5, 4, 2, 6},

{4, 3, 5, 2, 1},

{6, 4, 3, 2, 5},

{2, 3, 5, 6, 4},

{5, 6, 4, 3, 2}};

typedef struct Patient {

int id;

char name[50];

int age;

char gender[10];

char severity[20];

char disease[50];

struct Patient\* next;

} Patient;

Patient\* patientHead = NULL;

typedef struct Doctor {

int id;

char name[50];

char specialty[50];

int availableSlots;

struct Doctor\* next;

} Doctor;

Doctor\* doctorHead = NULL;

typedef struct Appointment {

char patientNam[50];

int doctorID;

struct Appointment\* next;

} Appointment;

void menu();

void login();

void firstinterface();

void initializeDoctors();

void returnlanding();

void slowTxt();

void addPatient();

void displayPatients();

void inputPatientData();

void updatePatient();

void deletePatientByID();

void makeAppointment();

void browseDoctorsBySpecialty();

int main() {

system("cls");

initializeDoctors();

firstinterface();

login();

system("cls");

return 0;

}

void slowTxt(char\* str) {

system("CLS");

printf("\n\n");

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

Sleep(20);

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

Sleep(20);

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

Sleep(20);

printf("\n\n\t");

int x = strlen(str);

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

printf("%c", str[i]);

Sleep(20);

}

printf("\n\n");

}

void firstinterface() {

char ab[] = "||||||||||||||||||||||||||||||||||||||||||||";

char ar[] = "|||||||| Welcome to our project ||||||||";

printf("\n\n\t%s\n", ab);

printf("\t");

for (int i = 0; i < sizeof(ar) - 1; i++) {

Sleep(25);

printf("\033[1m%c", ar[i]);

}

printf("\033[0m\n");

printf("\t%s\n", ab);

printf("\n\n");

}

void menu() {

int choice, id;

printf("\n\n");

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

printf("\t[1] Add New Patient\n");

printf("\t[2] Display Patients\n");

printf("\t[3] Search Patient by ID\n");

printf("\t[4] Update Patient Information\n");

printf("\t[5] Delete Patient by ID\n");

printf("\t[6] Return to Home\n");

printf("\t[7] Exit\n\n");

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

printf("\tEnter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

inputPatientData();

break;

case 2:

displayPatients();

break;

case 3:

printf("\tEnter Patient ID to search: ");

scanf("%d", &id);

Patient\* patient = searchPatientByID(id);

if (patient) {

printf("\tPatient Found: \n\t\tID : %d\n\t\tName : %s\n\t\tAge : %d\n\t\tGender : %s\n\t\tDisease : %s\n", patient->id, patient->name, patient->age, patient->gender, patient->disease);

} else {

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

}

break;

case 4:

printf("\tEnter Patient ID to update: ");

scanf("%d", &id);

updatePatient(id);

break;

case 5:

printf("\tEnter Patient ID to delete: ");

scanf("%d", &id);

deletePatientByID(id);

break;

case 6:

system("cls");

login();

break;

case 7:

printf("\tExiting...\n");

return;

default:

printf("\tInvalid choice! Please try again.\n");

}

returnlanding();

}

void login() {

int j;

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

printf("\n\t\t\t1. Admin Login\n\n");

printf("\t\t\t2. For Patient\n\n");

int x;

printf("\tEnter Your Choice : ");

scanf("%d", &x);

if (x == 1) {

int pass = 1234, pas;

printf(" \n Username : Admin");

printf(" \n ENTER PASSWORD : ");

scanf("%d", &pas);

if (pass == pas) {

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

char str[] = " WELCOME !!!! LOGIN IS SUCCESSFUL";

int x = strlen(str);

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

printf("%c", str[i]);

Sleep(20);

}

Sleep(1000);

system("cls");

// system("color 8f");

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

printf(" \n");

printf(" \n");

printf(" \t Please Wait...\n\n\n\n\n");

printf(" \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_ \n");

printf(" / \\ \n");

printf(" | Loading............. | \n");

printf("\t\t |\t");

for (j = 0; j < 24; j++) {

printf("%c", 219);

Sleep(50);

}

printf(" |\n \\\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_/ \n");

printf(" \n");

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

// system("pause");

Sleep(1000);

system("cls");

system("color 0f");

menu();

} else {

printf("Invalid Password !\n\n");

Sleep(40);

login();

}

}

if (x == 2) {

system("cls");

printf("\n");

printf("\t\t\t1. Make Appointment.\n\n");

int a;

printf("\tEnter your Choice: ");

scanf("%d", &a);

if (a == 1) {

browseDoctorsBySpecialty();

}

}

}

void returnlanding() {

printf("\n\tTo return Home[H]\n\tTo return to Main Menu[M]\n\tTo Close the Programme[0]\n\tEnter your choice: ");

char x;

scanf(" %c", &x);

if (x == '0') {

return;

} else if (x == 'M' || x == 'm') {

menu();

return;

} else {

login();

return;

}

}

void inputPatientData() {

int id, age;

char name[50], gender[10], disease[50], severity[20];

char s[] = "You wanted to add a new Patient. \n\tPlease enter his/her detailed information";

slowTxt(s);

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

printf("\n");

printf("\tEnter Patient ID: ");

scanf("%d", &id);

printf("\tEnter Patient Name: ");

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

printf("\tEnter Patient Age: ");

scanf("%d", &age);

printf("\tEnter Patient Gender: ");

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

printf("\tEnter Patient Disease: ");

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

printf("\tEnter Severity (e.g., Mild, Moderate, Severe): ");

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

addPatient(id, name, age, gender, disease, severity);

printf("\tPatient details added successfully!\n\n");

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

}

void addPatient(int id, const char\* name, int age, const char\* gender, const char\* disease, const char\* severity) {

Patient\* newPatient = (Patient\*)malloc(sizeof(Patient));

newPatient->id = id;

strcpy(newPatient->name, name);

newPatient->age = age;

strcpy(newPatient->gender, gender);

strcpy(newPatient->disease, disease);

strcpy(newPatient->severity, severity);

newPatient->next = NULL;

if (patientHead == NULL) {

patientHead = newPatient;

} else {

Patient\* temp = patientHead;

while (temp->next != NULL) {

temp = temp->next;

}

temp->next = newPatient;

}

}

Patient\* searchPatientByID(int id) {

Patient\* temp = patientHead;

while (temp != NULL) {

if (temp->id == id) {

return temp;

}

temp = temp->next;

}

return NULL; // Return NULL if the patient is not found

}

void deletePatientByID(int id) {

if (patientHead == NULL) {

printf("\tNo patients to delete.\n");

return;

}

Patient \*temp = patientHead, \*prev = NULL;

if (temp->id == id) {

patientHead = temp->next;

free(temp);

printf("\tPatient with ID %d deleted successfully.\n", id);

return;

}

while (temp != NULL && temp->id != id) {

prev = temp;

temp = temp->next;

}

if (temp == NULL) {

printf("\tPatient with ID %d not found.\n", id);

return;

}

prev->next = temp->next;

free(temp);

printf("\tPatient with ID %d deleted successfully.\n", id);

}

void displayPatients() {

Patient\* temp = patientHead;

char ar[] = "Patient List:";

printf("\t");

for (int i = 0; i < sizeof(ar) - 1; i++) {

Sleep(25);

printf("%c", ar[i]);

}

printf("\n");

while (temp != NULL) {

printf("\t\tID : %d\n\t\tName : %s\n\t\tAge : %d\n\t\tGender : %s\n\t\tDisease : %s\n", temp->id, temp->name, temp->age, temp->gender, temp->disease);

temp = temp->next;

printf("\n");

}

printf("\n");

}

void updatePatient(int id) {

Patient\* patient = searchPatientByID(id);

if (patient == NULL) {

printf("Patient with ID %d not found.\n", id);

return;

}

printf("\tEnter New Patient Name: ");

scanf(" %[^\n]s", patient->name);

printf("\tEnter New Patient Age: ");

scanf("%d", &patient->age);

printf("\tEnter New Patient Gender: ");

scanf(" %[^\n]s", patient->gender);

printf("\tEnter New Patient Disease: ");

scanf(" %[^\n]s", patient->disease);

printf("\tEnter Severity (e.g., Mild, Moderate, Severe): ");

scanf(" %[^\n]s", patient->severity);

printf("\tPatient information updated successfully!\n");

}

void initializeDoctors() {

doctorHead = NULL;

char specialties[5][50] = {

"Cardiology", "Orthopedics", "Dermatology", "Pediatrics", "Neurology"};

char doctorNames[5][5][100] = {

{"Assoc. Prof. Dr. Bijoy Dutta", "Prof. Dr. Md. Sahabuddin Khan", "Prof. Dr. Toufiqur Rahman Faruque", "Dr. AKS Zahid Mahmud Khan", "Prof. Dr. Ashok Kumar Dutta"},

{"Asst. Prof. Dr. Md. Nazmul Huda", "Dr. Md. Mizanur Rahman", "Dr. M A Mamun", "Dr. K M Shorfuddin Ashik", "Prof. Dr. Md. Kamrul Ahsan"},

{"Dr. Asif Imran Siddiqui", "Dr. Farzana Rahman Shathi", "Prof. Dr. M.N. Huda", "Prof. Lt. Col. Dr. Md. Abdul Wahab", "Prof. Dr. M. U. Kabir Chowdhury"},

{"Dr. Mithun Sarker", "Dr. Chowdhury Md. Niazuzzaman", "Dr. Hasan Mahmud Abdullah", "Dr. Md. Zahidul Islam", "Dr. Md. Waliur Rahman"},

{"Dr. Shamim Rashid", "Dr. Md. Shuktarul Islam (Tamim)", "Dr. Mohiuddin Ahmed", "Dr. Rakib Hasan Mohammad", "Prof. Dr. Subash Kanti Dey"}};

// int slots[5][5] = {

// {3, 5, 4, 2, 6},

// {4, 3, 5, 2, 1},

// {6, 4, 3, 2, 5},

// {2, 3, 5, 6, 4},

// {5, 6, 4, 3, 2}};

Doctor\* temp = NULL;

for (int i = 0; i < 5; i++) {

for (int j = 0; j < 5; j++) {

Doctor\* newDoctor = (Doctor\*)malloc(sizeof(Doctor));

newDoctor->id = i \* 5 + j + 1;

strcpy(newDoctor->name, doctorNames[i][j]);

strcpy(newDoctor->specialty, specialties[i]);

newDoctor->availableSlots = slots[i][j];

newDoctor->next = NULL;

if (doctorHead == NULL) {

doctorHead = newDoctor;

temp = doctorHead;

} else {

temp->next = newDoctor;

temp = newDoctor;

}

}

}

}

void browseDoctorsBySpecialty() {

char specialties[5][50] = {

"Cardiology", "Orthopedics", "Dermatology", "Pediatrics", "Neurology"};

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

printf("\tAvailable Specialties:\n");

for (int i = 0; i < 5; i++) {

printf("\t%d. %s\n", i + 1, specialties[i]);

}

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

printf("\n\tEnter the number of the specialty to browse: ");

int choice;

scanf("%d", &choice);

if (choice < 1 || choice > 5) {

printf("\tInvalid choice! Returning to main menu.\n");

return;

}

char selectedSpecialty[50];

strcpy(selectedSpecialty, specialties[choice - 1]);

Doctor\* temp = doctorHead;

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

printf("\tDoctors in %s:\n", selectedSpecialty);

int doctorFound = 0;

while (temp != NULL) {

if (strcmp(temp->specialty, selectedSpecialty) == 0) {

printf("\t\033[1mDoctor ID: %d\033[0m\n\tName: %s\n\tAvailable Slots: %d\n\n",

temp->id, temp->name, temp->availableSlots);

doctorFound = 1;

}

temp = temp->next;

}

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

if (!doctorFound) {

printf("\tNo doctors found in this specialty.\n");

return;

}

// Proceed to make an appointment

char patientName[50];

int doctorID;

printf("\tEnter your name: ");

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

printf("\tEnter the Doctor ID to make an appointment: ");

scanf("%d", &doctorID);

makeAppointment(patientName, doctorID);

}

Doctor\* searchDoctorByID(int id) {

Doctor\* temp = doctorHead;

while (temp != NULL) {

if (temp->id == id) {

return temp;

}

temp = temp->next;

}

return NULL;

}

void makeAppointment(char patientName[], int doctorID) {

Doctor\* doctor = searchDoctorByID(doctorID);

if (doctor == NULL) {

printf("\tDoctor with ID %d not found.\n", doctorID);

return;

}

if (doctor->availableSlots <= 0) {

printf("\tNo slots available for Doctor ID %d (%s).\n", doctorID, doctor->name);

return;

}

doctor->availableSlots--;

printf("\tAppointment confirmed for Patient: %s with Doctor ID %d (%s).\n", patientName, doctorID, doctor->name);

printf("\tYour serial number is %d.\n", slots[(doctorID - 1) / 5][(doctorID - 1) % 5] - doctor->availableSlots);

returnlanding();

}