Final Project Report — Programming Fundamentals

University Name: Fast National University Of Emerging Sciences

Department: Department of Software Engneering

Course: Programming Fundamentals

Project Title: Hospital Management System

Submitted By: Wajiha Nazir (Roll No. 25K-3021)

Emaan Binte Shams (Roll No. 25K-3082)

Submitted To: Miss Izzah Salam

Semester: Fall 2025

Date: 25-11-2025

Abstract

The Hospital Management System is a console-based C program that allows efficient management of patient appointments in hospitals. The system stores data such as patient information, doctors, appointment timings, and reason for visit. It ensures that no doctor is overbooked by checking the availability of time slots on a given date. The program also generates unique patient IDs automatically. Data is permanently stored in a text file (appointments.txt) using simple file handling. This project demonstrates core programming concepts including loops, conditional statements, arrays, structures, functions, and file handling to automate hospital appointment management.

1. Introduction

Managing hospital appointments manually can be time-consuming and prone to errors. The Hospital Management System provides a simple solution for selecting areas, hospitals, and doctors, scheduling appointments, and storing relevant patient data. The system ensures that no more than 9 appointments are scheduled for a doctor on a single day and prevents double bookings. All functionality is implemented in C, using structures for appointments, functions for modularity, arrays for storing options, and text files for permanent storage.

2. Objectives

- Store and manage patient and doctor information efficiently.
- Schedule appointments while preventing overbooking of doctors.
- Generate unique patient IDs automatically.
- Implement file handling to save appointments permanently.
- Utilize structures, arrays, loops, and functions for organized programming.
- Create a simple, user-friendly, menu-driven console interface.

3. System Design

System Overview

Flow of the program:

Start → Display Main Menu → Select Area → Select Hospital → Select Doctor → Enter Patient Name, Gender, Reason → Enter Date → Select Time Slot → Check Availability → Store Appointment → Display Confirmation → Exit

Algorithm

1. Start the program.
2. Display the main menu with options to book appointment or exit.
3. Display area selection menu.
4. Display hospital options based on selected area.
5. Display doctor options based on selected hospital.
6. Take patient details: name, gender, reason for visit.
7. Take appointment date as input.
8. Check the number of appointments for the doctor on that date (max 9).
9. Show available time slots and allow selection.
10. Generate a unique patient ID.
11. Save appointment details in appointments.txt.
12. Display confirmation.
13. Repeat or exit the program.

Input & Output

Input:

- Area number
- Hospital number
- Doctor selection
- Patient name and gender
- Reason for visit
- Appointment date

Output:

- List of available doctors
- Available time slots
- Appointment confirmation with patient ID
- Error messages if date or time slot is unavailable
- Saved appointment record in file

4. Implementation

Language: C

Compiler/IDE: Code::Blocks / Dev C++

Key Features

- Automatic unique patient ID generation.
- Ensures no double bookings or more than 9 appointments per doctor per day.
- Saves all appointment data permanently in appointments.txt.
- Menu-driven interface for easy navigation.
- Includes patient name, gender, reason, doctor, date, time, hospital, and area in records

Code Snippet

```
#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#include <time.h>

#define MAX_LINE 512


typedef struct {

    char doctor[25];

    char reason[50];

    char area[25];

    char hospital[50];

    char date[12];

} Appointment;


void getDoctor(char doctor[], int area, int hospital);

int isSlotTaken(const char *filename, const char *date, const char *doctor, const char
*timing);

int countDoctorAppointmentsOnDate(const char *filename, const char *date, const char
*doctor);    //ensures only 9 bookings are made for each doctor

int SelectTimeForDoctor(const char *filename, const char *date, const char *doctor, char
chosenTiming[]);  //shows available time slots

int generateUniquePatientID(const char *filename);


int main() {

    int hospital, area, timeChoice;

    char doctor[25];
    char reason[50];

    char timing[15];

    char areaName[25], hospitalName[50], date[12];


    char patientName[50];

    int patientID;

    char gender;


    FILE *file;


    srand(time(NULL)); // randomly generates a new ID each time using current time


    while (1) {      //process repeated until user chooses to exit

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

        printf("    Hospital Appointment System");

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

        printf("------Menu------\n");

        printf("1. Book Appointment\n2. Exit\nEnter your choice: ");

        int menuChoice;

        if (scanf("%d", &menuChoice) != 1) {   //scanf() will return 0 if an integer isnt input.'1'
here means TRUE.

            while (getchar() != '\n');       //ensures input buffer is empty after user presses
ENTER before next input is taken

            printf("Invalid input. Try again.\n");

            continue;

        }
        if(menuChoice == 2){
            printf("Exiting the system. Goodbye!\n");
            break;
        } else if(menuChoice != 1){
            printf("Invalid choice. Try again.\n");
            continue;
        }


        // --- Select Area ---
        printf("\n------Select Area------\n");
        printf("1.Clifton\n2.DHA\n3.Gulshan-e-Iqbal\n4.Saddar\n5.North Nazimabad\n");
        printf("Enter your choice: ");
        if (scanf("%d", &area) != 1) {
            while (getchar() != '\n');
            printf("Invalid input. Returning to main menu.\n");
            continue;
        }


        printf("\n------Select Hospital------\n");
        switch (area) {
            case 1:
                printf("1.Agha Khan Health Centre\n2.Ziauddin Hospital\n3.South City
Hospital\nEnter your choice: ");
                scanf("%d", &hospital);
                break;
            case 2:
```

```
            printf("1.PAF Hospital\n2.MediCare Cardiac & General Hospital\n3.Australian
Concept Fertility Centre\nEnter your choice: ");
            scanf("%d", &hospital);
            break;
        case 3:
            printf("1.Liaquat National Hospital\n2.Mamji Hospital\n3.Tabba Heart
Institute\nEnter your choice: ");
            scanf("%d", &hospital);
            break;
        case 4:
            printf("1.JPMC\n2.Civil Hospital\n3.National Institute of Child Health\nEnter your
choice: ");
            scanf("%d", &hospital);
            break;
        case 5:
            printf("1.Ziauddin Hospital\n2.Saifee Hospital\n3.Abbasi Shaheed Hospital\nEnter
your choice: ");
            scanf("%d", &hospital);
            break;
        default:
            printf("Invalid area selection.\n");
            continue;
    }

    // --- Select Doctor ---
    getDoctor(doctor, area, hospital);

    getchar(); // clear newline
```

```
        printf("\nEnter Patient Name: ");

        fgets(patientName, sizeof(patientName), stdin);

        patientName[strcspn(patientName, "\n")] = '\0';  //clears newline after input


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

        scanf(" %c", &gender);

        getchar();


        printf("Please share the reason of your visit: ");

        fgets(reason, sizeof(reason), stdin);

        reason[strcspn(reason, "\n")] = '\0';


        printf("Enter appointment date (YYYY-MM-DD): ");

        fgets(date, sizeof(date), stdin);

        date[strcspn(date, "\n")] = '\0';


        // Check doctor's total appointments on this date

        if (countDoctorAppointmentsOnDate("appointments.txt", date, doctor) >= 9) {  //a
doctor has max 9 slots for 1 day if all 9 booked, choose another doctor

            printf("\nDoctor Fully Booked for This Date. Please choose another date or
doctor.\n");

            continue;

        }


        // --- Time slot selection ---

        timeChoice = SelectTimeForDoctor("appointments.txt", date, doctor, timing);

        if (timeChoice == -1) continue; // invalid time slot
```

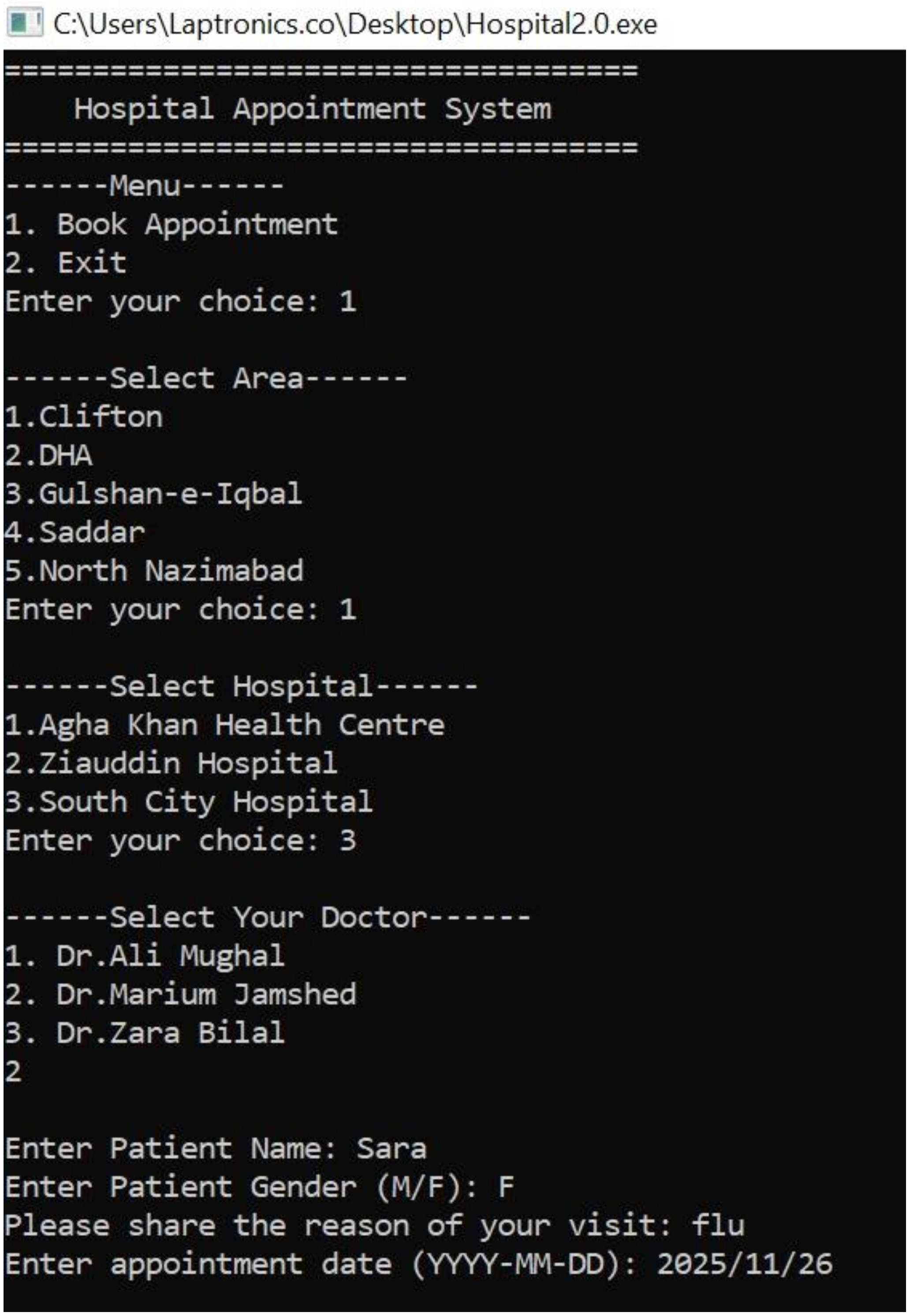
```
// --- Area Name ---
switch (area) {   //records selected area name
    case 1: strcpy(areaName, "Clifton"); break;
    case 2: strcpy(areaName, "DHA"); break;
    case 3: strcpy(areaName, "Gulshan-e-Iqbal"); break;
    case 4: strcpy(areaName, "Saddar"); break;
    case 5: strcpy(areaName, "North Nazimabad"); break;
    default: strcpy(areaName, "Unknown"); break;
}

// --- Hospital Name ---
if (area == 1) {  //records selected doctors name
    if (hospital == 1) strcpy(hospitalName, "Agha Khan Health Centre");
    else if (hospital == 2) strcpy(hospitalName, "Ziauddin Hospital");
    else strcpy(hospitalName, "South City Hospital");
} else if (area == 2) {
    if (hospital == 1) strcpy(hospitalName, "PAF Hospital");
    else if (hospital == 2) strcpy(hospitalName, "MediCare Cardiac & General Hospital");
    else strcpy(hospitalName, "Australian Concept Fertility Centre");
} else if (area == 3) {
    if (hospital == 1) strcpy(hospitalName, "Liaquat National Hospital");
    else if (hospital == 2) strcpy(hospitalName, "Mamji Hospital");
    else strcpy(hospitalName, "Tabba Heart Institute");
} else if (area == 4) {
    if (hospital == 1) strcpy(hospitalName, "JPMC");
    else if (hospital == 2) strcpy(hospitalName, "Civil Hospital");
    else strcpy(hospitalName, "National Institute of Child Health");
} else if (area == 5) {
    if (hospital == 1) strcpy(hospitalName, "Ziauddin Hospital");
    else if (hospital == 2) strcpy(hospitalName, "Saifee Hospital");
    else strcpy(hospitalName, "Abbasi Shaheed Hospital");
} else {
    strcpy(hospitalName, "Unknown");
}

// --- Generate random unique patient ID ---
patientID = generateUniquePatientID("appointments.txt");

// --- SAVE TO FILE ---
file = fopen("appointments.txt", "a");
if (!file) {
    printf("Error opening file!\n");
    return 1;
}

fprintf(file, "%s|%s|%s|%s|%s|%s|%d|%c|%s\n",
        date, areaName, hospitalName, doctor, timing, reason,
        patientID, gender, patientName);

fclose(file);
```

```
        // --- SUCCESS MESSAGE --- prints receipt
        printf("\n--------------------- Appointment Made Successfully ------------------------\n");
        printf("Patient Name: %s\nPatient ID: %d\nGender: %c\n", patientName, patientID,
gender);
        printf("Doctor: %s\nDate: %s\nTime: %s\nArea: %s\nHospital: %s\nReason: %s\n",
               doctor, date, timing, areaName, hospitalName, reason);
    }


    return 0;
}


// ---------------- FUNCTIONS ------------------------


int isSlotTaken(const char *filename, const char *date, const char *doctor, const char
*timing) {
    FILE *file = fopen(filename, "r");
    if (!file) return 0; //if file fails to open/error in opening the file/file doesnt exist --> we
will assume slot is available


    char line[MAX_LINE]; //buffer to read each line from the file.
    char *token;         // used to split line into parts
    char existingDate[32], existingDoctor[64], existingTiming[32];


    while (fgets(line, sizeof(line), file)) {  //loop to go through all lines of the file
        token = strtok(line, "|"); //splits lines where the chracater '|' occurs
        if (!token) continue;    //skips line as no data was found here.
        strncpy(existingDate, token, sizeof(existingDate)); existingDate[sizeof(existingDate)-1]
= '\0';//copy date from file and ensure null terminator is present at the end.
        token = strtok(NULL, "|"); if (!token) continue;
        token = strtok(NULL, "|"); if (!token) continue;   //skip the fields we dont currently need
and go straight to the doctor field
        token = strtok(NULL, "|"); if (!token) continue;
        strncpy(existingDoctor, token, sizeof(existingDoctor));
existingDoctor[sizeof(existingDoctor)-1] = '\0';
        token = strtok(NULL, "|"); if (!token) continue;  //time field
        strncpy(existingTiming, token, sizeof(existingTiming));
existingTiming[sizeof(existingTiming)-1] = '\0';
        existingTiming[strcspn(existingTiming, "\n")] = '\0';  //replaces newline with null
terminator

        if (strcmp(existingDate, date) == 0 && strcmp(existingDoctor, doctor) == 0 &&
strcmp(existingTiming, timing) == 0) { //if all 3 same then slot already booked
            fclose(file);
            return 1; //slot already booked
        }
    }
    fclose(file);
    return 0;   //slot available
}

int countDoctorAppointmentsOnDate(const char *filename, const char *date, const char
*doctor) {
    FILE *file = fopen(filename, "r");
    if (!file) return 0;
    char line[MAX_LINE], *token, existingDate[32], existingDoctor[64];
    int count = 0;


    while (fgets(line, sizeof(line), file)) {
        token = strtok(line, "|");
        if (!token) continue;  //empty line


        strncpy(existingDate, token, sizeof(existingDate)); existingDate[sizeof(existingDate)-1]
= '\0';
        token = strtok(NULL, "|"); if (!token) continue; //skop area
        token = strtok(NULL, "|"); if (!token) continue;  //skip hospital
        token = strtok(NULL, "|"); if (!token) continue;
        strncpy(existingDoctor, token, sizeof(existingDoctor));
existingDoctor[sizeof(existingDoctor)-1] = '\0';
        existingDoctor[strcspn(existingDoctor, "\n")] = '\0';


        if (strcmp(existingDate, date) == 0 && strcmp(existingDoctor, doctor) == 0) count++;
//counts number of appointments a doctor has on a specific date
    }
    fclose(file);
    return count;
}


int SelectTimeForDoctor(const char *filename, const char *date, const char *doctor, char
chosenTiming[]) {
    const char *slots[9] = {"8:00 AM", "9:00 AM", "10:00 AM", "11:00 AM","12:00 PM", "1:00
PM", "2:00 PM", "3:00 PM", "4:00 PM"};
    int availableIndices[9], availableCount = 0, i, userChoice;
    for (i = 0; i < 9; i++)

        if (!isSlotTaken(filename, date, doctor, slots[i])) //if slot free add to availableindices
array and increment available count

            availableIndices[availableCount++] = i; //stores first free slot at first index and
onwards,this will remove unavailable slots from options shown to user



    if (availableCount == 0) {  //no free slots

        printf("\nAll slots for Dr. %s on %s are booked.\n", doctor, date);

        return -1;

    }

    printf("\nAvailable time slots for Dr. %s on %s:\n", doctor, date);

    for (i = 0; i < availableCount; i++)

        printf("%d. %s\n", i+1, slots[availableIndices[i]]); // prints actual time : "8:00 AM"

    printf("Enter the number of preferred slot: ");

    if (scanf("%d", &userChoice) != 1 || userChoice < 1 || userChoice > availableCount) { //
input error control

        while(getchar()!='\n');

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

        return -1;

    }

    strcpy(chosenTiming, slots[availableIndices[userChoice-1]]);

    return availableIndices[userChoice-1]+1;

}


void getDoctor(char doctor[], int area, int hospital){

    int doc;

    printf("\n------Select Your Doctor------\n");
    if(area==1){ switch(hospital){ case 1: printf("1. Dr.Asma Majeed\n2. Dr.Talha
Rehman\n3. Dr.Wajiha Nazir\n"); break;

                                case 2: printf("1. Dr.Mujeeb Farhan\n2. Dr.Sara Taufeeq\n3. Dr.Emaan
Shams\n"); break;

                                case 3: printf("1. Dr.Ali Mughal\n2. Dr.Marium Jamshed\n3. Dr.Zara
Bilal\n"); break;} }

    else if(area==2){ switch(hospital){ case 1: printf("1. Dr.Safeer\n2. Dr.Saad Ali\n3.
Dr.Amna Rehan\n"); break;

                                    case 2: printf("1. Dr.Mahrukh\n2. Dr.Roha Nadeem\n3. Dr.Shaffan
Ahmed\n"); break;

                                    case 3: printf("1. Dr.Rabia Asghar\n2. Dr.Zaevia Haidar\n3. Dr.Ameera
Ahsan\n"); break;} }

    else if(area==3){ switch(hospital){ case 1: printf("1. Dr.Noor\n2. Dr.Naveen\n3. Dr.Shariq
Khan\n"); break;

                                    case 2: printf("1. Dr.Ayesha Kashif\n2. Dr.Maira\n3. Dr.Umar
Rehman\n"); break;

                                    case 3: printf("1. Dr.Hamza Rizwan\n2. Dr.Anoosha Adnan\n3.
Dr.Aleeya Zahra\n"); break;} }

    else if(area==4){ switch(hospital){ case 1: printf("1. Dr.Rehum Arif\n2. Dr.Mujtaba\n3.
Dr.Aayan Nasir\n"); break;

                                    case 2: printf("1. Dr.Khadija Bukhari\n2. Dr.Shanzay\n3. Dr.Ibrahim
Shahzad\n"); break;

                                    case 3: printf("1. Dr.Abiya\n2. Dr.Amna Kaleem\n3. Dr.Muhammad
Saif\n"); break;} }

    else if(area==5){ switch(hospital){ case 1: printf("1. Dr.Leena\n2. Dr.Taha Junaidi\n3.
Dr.Sohaib\n"); break;

                                    case 2: printf("1. Dr.Syeda Ramla\n2. Dr.Eshaal Fatima\n3.
Dr.Abrar\n"); break;

                                    case 3: printf("1. Dr.Ebad\n2. Dr.Zubair\n3. Dr.Adeel Javed\n");
break;} }


    if(scanf("%d",&doc)!=1){ while(getchar()!='\n'); strcpy(doctor,"Unknown"); return; }
```

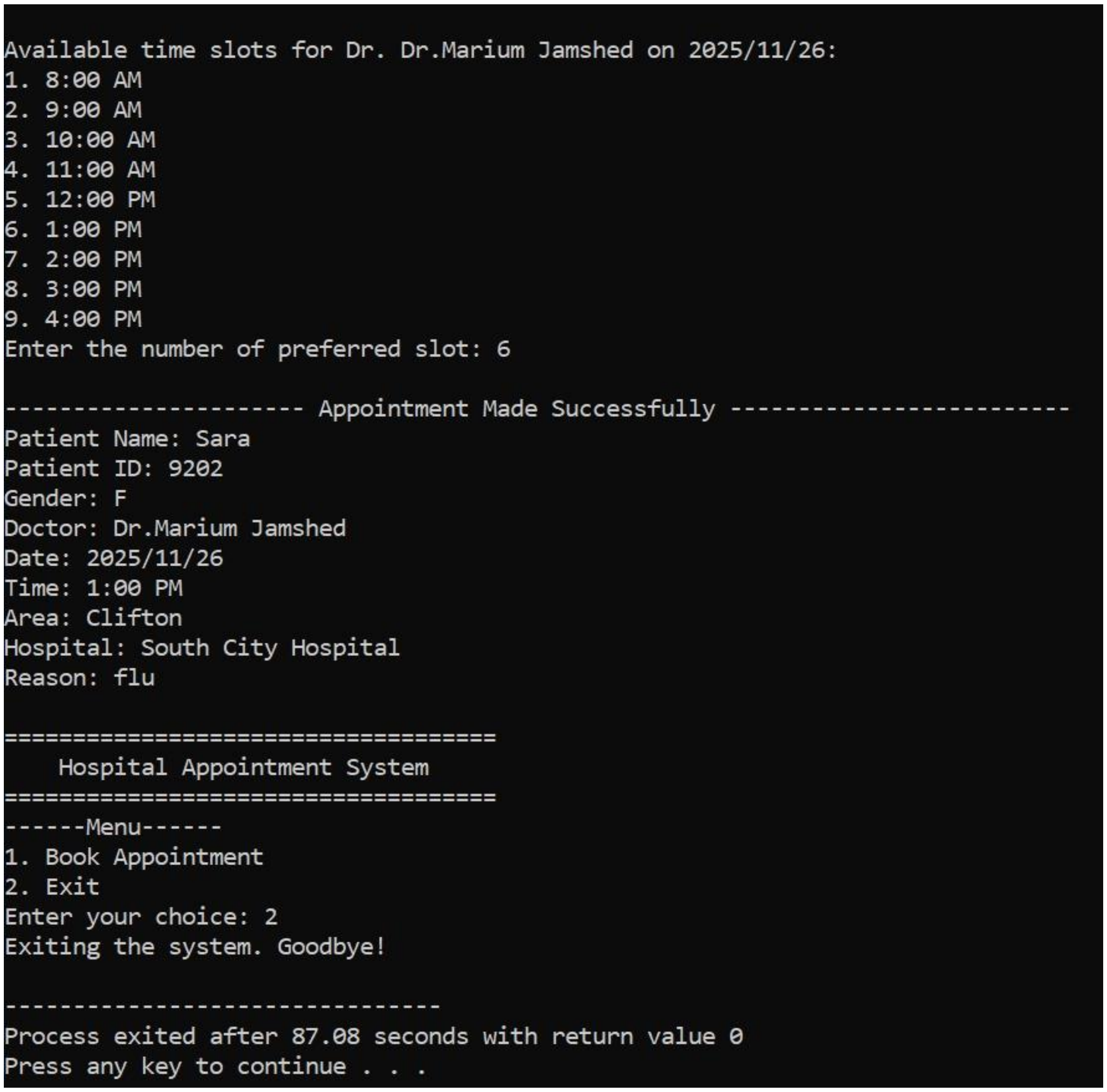
```
   // assign based on selection

   if(area==1){ if(hospital==1) strcpy(doctor, doc==1?"Dr.Asma Majeed":doc==2?"Dr.Talha
Rehman":"Dr.Wajiha Nazir");

            else if(hospital==2) strcpy(doctor, doc==1?"Dr.Mujeeb Farhan":doc==2?"Dr.Sara
Taufeeq":"Dr.Emaan Shams");

            else strcpy(doctor, doc==1?"Dr.Ali Mughal":doc==2?"Dr.Marium Jamshed":"Dr.Zara
Bilal"); }

   else if(area==2){ if(hospital==1) strcpy(doctor, doc==1?"Dr.Safeer":doc==2?"Dr.Saad
Ali":"Dr.Amna Rehan");

               else if(hospital==2) strcpy(doctor, doc==1?"Dr.Mahrukh":doc==2?"Dr.Roha
Nadeem":"Dr.Shaffan Ahmed");

               else strcpy(doctor, doc==1?"Dr.Rabia Asghar":doc==2?"Dr.Zaevia
Haidar":"Dr.Ameera Ahsan"); }

   else if(area==3){ if(hospital==1) strcpy(doctor,
doc==1?"Dr.Noor":doc==2?"Dr.Naveen":"Dr.Shariq Khan");

               else if(hospital==2) strcpy(doctor, doc==1?"Dr.Ayesha
Kashif":doc==2?"Dr.Maira":"Dr.Umar Rehman");

               else strcpy(doctor, doc==1?"Dr.Hamza Rizwan":doc==2?"Dr.Anoosha
Adnan":"Dr.Aleeya Zahra"); }

   else if(area==4){ if(hospital==1) strcpy(doctor, doc==1?"Dr.Rehum
Arif":doc==2?"Dr.Mujtaba":"Dr.Aayan Nasir");

               else if(hospital==2) strcpy(doctor, doc==1?"Dr.Khadija
Bukhari":doc==2?"Dr.Shanzay":"Dr.Ibrahim Shahzad");

               else strcpy(doctor, doc==1?"Dr.Abiya":doc==2?"Dr.Amna
Kaleem":"Dr.Muhammad Saif"); }

   else if(area==5){ if(hospital==1) strcpy(doctor, doc==1?"Dr.Leena":doc==2?"Dr.Taha
Junaidi":"Dr.Sohaib");

               else if(hospital==2) strcpy(doctor, doc==1?"Dr.Syeda Ramla":doc==2?"Dr.Eshaal
Fatima":"Dr.Abrar");

               else strcpy(doctor, doc==1?"Dr.Ebad":doc==2?"Dr.Zubair":"Dr.Adeel Javed"); }

}
```

```
// ---------------- GENERATE UNIQUE PATIENT ID ----------------
int generateUniquePatientID(const char *filename){
    int id, i;
    FILE *file;
    int unique;
    char line[MAX_LINE], *token;
    do{
        id = rand()%9000 + 1000; // 4-digit random number
        unique = 1;
        file = fopen(filename,"r");
        if(file){
            while(fgets(line,sizeof(line),file)){
                token = strtok(line,"|");
                for(i=0;i<6;i++) token = strtok(NULL,"|"); // skip to patientID
                if(token && atoi(token) == id){
                    unique = 0;
                    break;
                }
            }
            fclose(file);
        }
    }while(!unique);
    return id;
}
```

Sample Output


C:\Users\Laptronics.co\Desktop\Hospital2.0.exe

```
====================================
    Hospital Appointment System
====================================
------Menu------
1. Book Appointment
2. Exit
Enter your choice: 1

------Select Area------
1.Clifton
2.DHA
3.Gulshan-e-Iqbal
4.Saddar
5.North Nazimabad
Enter your choice: 1

------Select Hospital------
1.Agha Khan Health Centre
2.Ziauddin Hospital
3.South City Hospital
Enter your choice: 3

------Select Your Doctor------
1. Dr.Ali Mughal
2. Dr.Marium Jamshed
3. Dr.Zara Bilal
2

Enter Patient Name: Sara
Enter Patient Gender (M/F): F
Please share the reason of your visit: flu
Enter appointment date (YYYY-MM-DD): 2025/11/26
```



```
Available time slots for Dr. Dr.Marium Jamshed on 2025/11/26:
1. 8:00 AM
2. 9:00 AM
3. 10:00 AM
4. 11:00 AM
5. 12:00 PM
6. 1:00 PM
7. 2:00 PM
8. 3:00 PM
9. 4:00 PM
Enter the number of preferred slot: 6

--------------------- Appointment Made Successfully ------------------------
Patient Name: Sara
Patient ID: 9202
Gender: F
Doctor: Dr.Marium Jamshed
Date: 2025/11/26
Time: 1:00 PM
Area: Clifton
Hospital: South City Hospital
Reason: flu

====================================
    Hospital Appointment System
====================================
------Menu------
1. Book Appointment
2. Exit
Enter your choice: 2
Exiting the system. Goodbye!

--------------------------------
Process exited after 87.08 seconds with return value 0
Press any key to continue . . .
```

5. Testing & Results

Test No	Input	Expected Output	Actual Output	Status
1	Select area, hospital, doctor, date	Appointment stored in file	Appointment stored in file	✓
2	Attempt booking same doctor/date/time	Error message "Doctor Fully Booked"	Error message shown	✓
3	Schedule different date/time	Appointment stored	Appointment stored	✓
4	Enter patient name, gender, reason	Correct data saved in file	Correct data saved	✓

The system successfully prevented double bookings, saved all appointment data to file, and displayed confirmations accurately. Execution was efficient, and memory usage minimal.

6. Conclusion, Limitations & References

Conclusion

The Hospital Management System automates scheduling and management of patient appointments. It demonstrates the use of arrays, loops, structures, functions, conditional statements, and file handling in C. The program efficiently prevents overbooking and stores appointments permanently.

Limitations

- Only appointments are stored; doctor/patient details are pre-defined or minimal.
- No search, edit, or delete functionality for appointments.
- Limited input validation beyond date/time checks.
- Console-based interface only.

Future Enhancements

- Store doctor and patient information in separate files.
- Add search, edit, and delete functionality for appointments.
- Implement a graphical or web-based interface.
- Include authentication for secure access.
- Allow multiple patients per slot or simultaneous booking handling.

References

- Let Us C by Yashavant P. Kanetkar
- https://www.programiz.com/c-programming
- https://www.geeksforgeeks.org/c-programming-language/