4. Implementation

Tools used

Component	Technology
Language	C (ANSI C)
Compiler	GCC / MinGW / Turbo C / Code::Blocks
OS	Windows/Linux
Input	CLI (Command Line Interface)
Output	Terminal-based tabular and text output

Flowchart

Main()

└──> addDoctors()

└──> Display Menu

├── case 1 → addPatient() → suggestDoctor()

├── case 2 → listPatients()

├── case 3 → dischargePatient()

├── case 4 → listDoctors()

└── case 0 → exit()

CODE

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAX_PATIENTS 100
#define MAX_DOCTORS 10

struct Patient {
    int id;
    char name[50];
    int age;
    char gender[10];
    char dob[15];
    char address[100];
    char disease[50];
};

struct Doctor {
    int id;
    char name[50];
    char specialize[50];
};

struct Patient patients[MAX_PATIENTS];
```

```c
struct Doctor doctors[MAX_DOCTORS];
int numPatients = 0;
int numDoctors = 0;

void addDoctors() {
    strcpy(doctors[0].name, "Dr. Meena Sharma");
    strcpy(doctors[0].specialize, "Cardiologist");
    doctors[0].id = 101;

    strcpy(doctors[1].name, "Dr. Rajeev Menon");
    strcpy(doctors[1].specialize, "Neurologist");
    doctors[1].id = 102;

    strcpy(doctors[2].name, "Dr. Nisha Rao");
    strcpy(doctors[2].specialize, "General Physician");
    doctors[2].id = 103;

    strcpy(doctors[3].name, "Dr. Amit Khurana");
    strcpy(doctors[3].specialize, "Orthopedic");
    doctors[3].id = 104;

    strcpy(doctors[4].name, "Dr. Priya Singh");
    strcpy(doctors[4].specialize, "Dermatologist");
    doctors[4].id = 105;

    strcpy(doctors[5].name, "Dr. Nandini Iyer");
    strcpy(doctors[5].specialize, "Ophthalmologist");
    doctors[5].id = 106;

    numDoctors = 6;
}

void listDoctors() {
    printf("\n%-5s %-25s %-25s\n", "ID", "Name", "Specialization");
    printf("%-5s %-25s %-25s\n", "-----", "-------------------------", "-------------------------");
    for (int i = 0; i < numDoctors; i++) {
        printf("%-5d %-25s %-25s\n", doctors[i].id, doctors[i].name, doctors[i].specialize);
    }
}

void suggestDoctor(const char* disease) {
    printf("\nSuggested Doctor based on disease \"%s\":\n", disease);
    if (strstr(disease, "heart")) {
        printf("-> %s (Cardiologist)\n", doctors[0].name);
    } else if (strstr(disease, "brain") || strstr(disease, "neuro")) {
        printf("-> %s (Neurologist)\n", doctors[1].name);
    } else if (strstr(disease, "bone") || strstr(disease, "fracture")) {
        printf("-> %s (Orthopedic)\n", doctors[3].name);
    } else if (strstr(disease, "skin") || strstr(disease, "rash")) {
        printf("-> %s (Dermatologist)\n", doctors[4].name);
    } else if (strstr(disease, "eye") || strstr(disease, "vision")) {
        printf("-> %s (Ophthalmologist)\n", doctors[5].name);
    } else {
        printf("-> %s (General Physician)\n", doctors[2].name);
    }
}

void addPatient() {
    if (numPatients >= MAX_PATIENTS) {
        printf("Max patients reached!\n");
        return;
    }

    struct Patient p;
    p.id = numPatients + 1;

    printf("Enter Patient Name: ");
    scanf(" %[^\n]", p.name);

    printf("Enter Age: ");
    scanf("%d", &p.age);

    printf("Enter Gender: ");
```

```
    scanf(" %[^\n]", p.gender);

    printf("Enter Date of Birth (dd-mm-yyyy): ");
    scanf(" %[^\n]", p.dob);

    printf("Enter Address: ");
    scanf(" %[^\n]", p.address);

    printf("Enter Disease: ");
    scanf(" %[^\n]", p.disease);

    patients[numPatients++] = p;

    printf("\nPatient Registered Successfully!\n");
    suggestDoctor(p.disease);
}

void listPatients() {
    printf("\n%-5s %-20s %-5s %-10s %-15s %-25s %-20s\n",
        "ID", "Name", "Age", "Gender", "DOB", "Address", "Disease");
    printf("-----------------------------------------------------------------------------------------------------------\n");
    for (int i = 0; i < numPatients; i++) {
        struct Patient p = patients[i];
        printf("%-5d %-20s %-5d %-10s %-15s %-25s %-20s\n",
            p.id, p.name, p.age, p.gender, p.dob, p.address, p.disease);
    }
}

void dischargePatient() {
    int id, found = 0;
    printf("Enter Patient ID to discharge: ");
    scanf("%d", &id);

    for (int i = 0; i < numPatients; i++) {
        if (patients[i].id == id) {
            found = 1;
            for (int j = i; j < numPatients - 1; j++) {
                patients[j] = patients[j + 1];
            }
            numPatients--;
            printf("Patient ID %d discharged successfully.\n", id);
            break;
        }
    }

    if (!found) {
        printf("Patient ID not found.\n");
    }
}

int main() {
    int choice;
    addDoctors();

    while (1) {
        printf("\n==== Hospital Management System ====\n");
        printf("1. Add Patient\n");
        printf("2. List Patients\n");
        printf("3. Discharge Patient\n");
        printf("4. Show Doctors\n");
        printf("0. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1: addPatient(); break;
            case 2: listPatients(); break;
            case 3: dischargePatient(); break;
            case 4: listDoctors(); break;
            case 0: exit(0);
            default: printf("Invalid choice.\n");
        }
    }
```

```
    return 0;
}
```