# CLINIC MANAGEMENT SYSTEM

## C MINI PROJECT

**NAME**: S GAYATHRI

**ROLL NO**:65

**COURSE NAME**: PROGRAMING IN C

**DATE**:15-07-2024

# INTRODUCTION

## PROJECT OVERVIEW:

The Clinic Management System is designed to streamline the operations of a clinic by digitizing patient records, appointment scheduling, and prescription management.

## PROBLEM STATEMENT:

Manual management of patient records, appointments, and prescriptions can lead to errors, delays, and inefficiencies.

## OBJECTIVE:

The primary objective of this project is to develop a Clinic Management System in C that allows clinic staff to manage patient records, schedule appointments, and issue prescriptions with ease.

# SYSTEM REQUIREMENTS

## Hardware requirements:

A computer with a minimum of 2 GHz processor

4 GB RAM

100 MB of free disk space

## Software requirements:

An operating system capable of running C programs (e.g., Windows, Linux, macOS)

A C compiler (e.g., GCC, Clang)

A text editor or an Integrated Development Environment (IDE)

# DESIGN AND DEVELOPMENT

- **PROGRAM LOGIC:**

The Clinic Management System is designed with a menu-driven interface that allows users to perform various operations such as adding new patients, scheduling appointments, viewing patient records, and issuing prescriptions. The system uses structured data types to store patient information and linked lists to manage appointments and prescriptions dynamically

- **PSEUDOCODE:**

```
DEFINE MAX_PATIENTS as 100
DEFINE MAX_APPOINTMENTS as 100
DEFINE MAX_PRESCRIPTIONS as 100

STRUCTURE Patient
    id: INTEGER
    name: STRING
    age: INTEGER

STRUCTURE Appointment
    Patient id: INTEGER
    date: STRING

STRUCTURE Prescription
    Patient Id: INTEGER
    medication: STRING
```

```
    dosage: STRING


DECLARE patients as ARRAY of Patient [MAX_PATIENTS]
DECLARE appointments as ARRAY of Appointment [MAX_APPOINTMENTS]
DECLARE prescriptions as ARRAY of Prescription [MAX_PRESCRIPTIONS]
DECLARE patient Count as INTEGER = 0
DECLARE appointment Count as INTEGER = 0
DECLARE prescription Count as INTEGER = 0


FUNCTION main()
    DECLARE choice as INTEGER

    WHILE TRUE
        PRINT menu options
```

```
        INPUT choice

        SWITCH choice
            CASE 1: CALL add patient ()
            CASE 2: CALL add Appointment ()
            CASE 3: CALL add Prescription ()
            CASE 4: CALL view Patients ()
            CASE 5: CALL view Appointments ()
            CASE 6: CALL view Prescriptions ()
            CASE 7: EXIT program
            DEFAULT: PRINT "Invalid choice"
        END SWITCH
    END WHILE
END FUNCTION

FUNCTION add Patient ()
    IF patient Count < MAX_PATIENTS
THEN
```

```
        DECLARE new Patient as Patient
        SET newPatient.id = patient Count + 1

        INPUT newPatient.name
        INPUT new Patient. age

        ADD new Patient to patient array
        INCREMENT patient Count
        PRINT "Patient added successfully"
    ELSE
        PRINT "Maximum patients reached"
    END IF
END FUNCTION

FUNCTION add Appointment ()
    IF appointment Count <
MAX_APPOINTMENTS THEN
        DECLARE new Appointment as
Appointment
```

```
        INPUT new Appointment. Patient Id
        INPUT new Appointment. date

        ADD new Appointment to appointments
array
        INCREMENT appointment Count
        PRINT "Appointment added
successfully"
    ELSE
        PRINT "Maximum appointments
reached"
    END IF
END FUNCTION

FUNCTION add Prescription ()
    IF prescription Count <
MAX_PRESCRIPTIONS THEN
```

```
        DECLARE new Prescription as
Prescription

        INPUT new Prescription. Patient Id
        INPUT new Prescription. medication
        INPUT new Prescription. dosage

        ADD new Prescription to prescriptions
array
        INCREMENT prescription Count
        PRINT "Prescription added successfully"
    ELSE
        PRINT "Maximum prescriptions
reached"
    END IF
END FUNCTION

FUNCTION view Patients ()
    PRINT "Patient List:"
```

```
    PRINT "ID    Name    Age"
    FOR EACH patient in patients
        PRINT patient.id, patient.name, patient.
age
    END FOR
END FUNCTION

FUNCTION view Appointments ()
    PRINT "Appointment List:"
    PRINT "Patient ID    Date"
    FOR EACH appointment in appointments
        PRINT appointment. Patient Id,
appointment. date
    END FOR
END FUNCTION

FUNCTION view Prescriptions ()
    PRINT "Prescription List:"
    PRINT "Patient ID    Medication    Dosage"
```

```
    FOR EACH prescription in prescriptions
        PRINT prescription. Patient Id,
prescription. medication, prescription. dosage
    END FOR
END FUNCTION
```

# TESTING AND RESULT

## Test cases:

**Test case 1: Adding a new patient record**

**Input: Klimt,28**

**Output: patient added successfully**

**Test case 2: Scheduling an appointment**

**Input: 1,30 December 2000**

**Output: Appointment added successfully**

**Test case 3: Add prescription**

**Input: 1, Aspirin,100mg**

**Output: Prescription added successfully**

**Test case 4: View patients**

**Test case 5: View Appointments**

**Test case 6: View prescription**

**Test case 7: Exit**

# OUTPUT:

```
PS C:\Users\Gayathri S\Documents\C> gcc clinicmanagement.c
PS C:\Users\Gayathri S\Documents\C> .\a.exe

Clinic Management System
1. Add Patient
2. Add Appointment
3. Add Prescription
4. View Patients
5. View Appointments
6. View Prescriptions
7. Exit
```

# Test case 1:

```
Clinic Management System
1. Add Patient
2. Add Appointment
3. Add Prescription
4. View Patients
5. View Appointments
6. View Prescriptions
7. Exit
Enter your choice: 1
Enter patient name: Klimt
Enter patient age: 28
Patient added successfully.
```

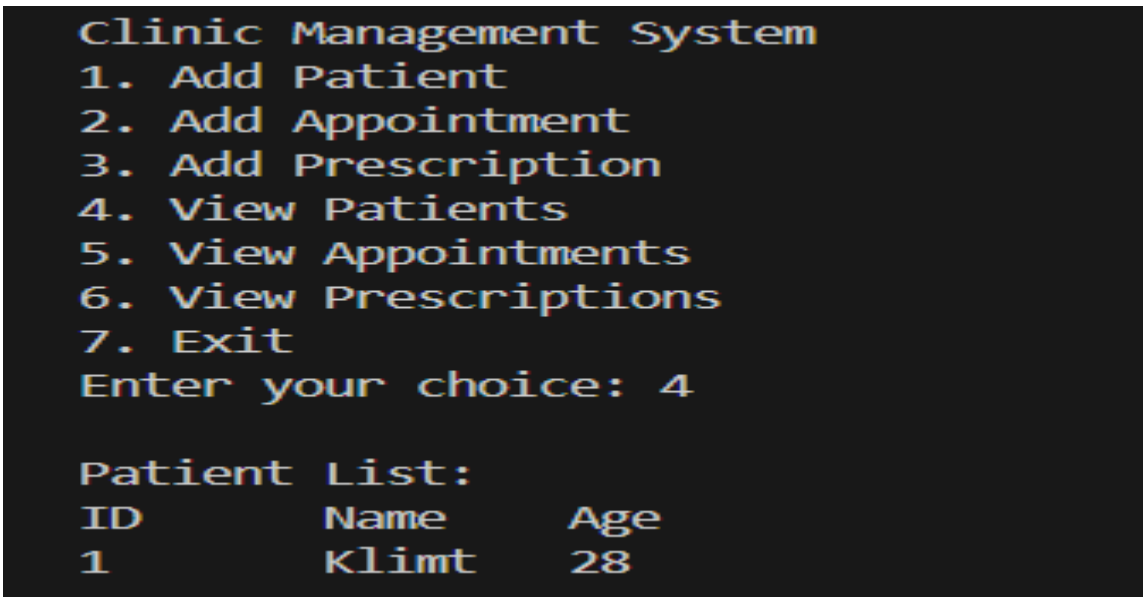
# Test case 2:

```
Clinic Management System
1. Add Patient
2. Add Appointment
3. Add Prescription
4. View Patients
5. View Appointments
6. View Prescriptions
7. Exit
Enter your choice: 2
Enter patient ID: 1
Enter appointment date: 30 December 2000
Appointment added successfully.
```

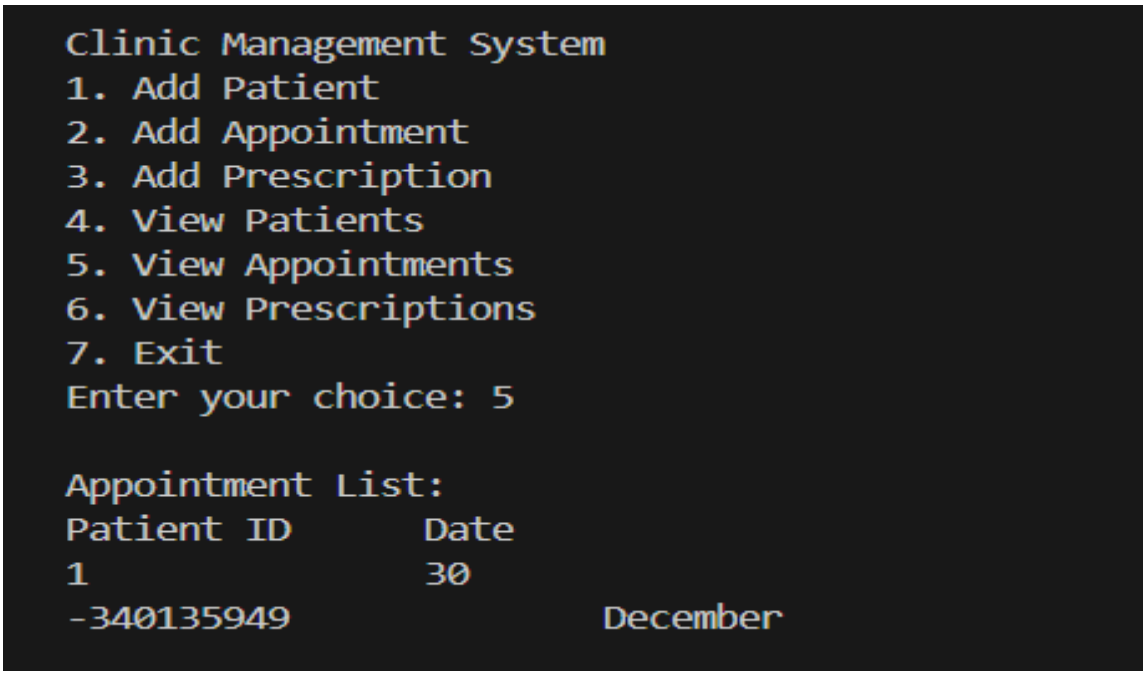
# Test case 3:

```
Clinic Management System
1. Add Patient
2. Add Appointment
3. Add Prescription
4. View Patients
5. View Appointments
6. View Prescriptions
7. Exit
Enter your choice: 3
Enter patient ID: 1
Enter medication: Aspirin
Enter dosage: 100mg
Prescription added successfully.
```

# Test case 4:




# Test case 5:



# Test case 6:

```
Clinic Management System
1. Add Patient
2. Add Appointment
3. Add Prescription
4. View Patients
5. View Appointments
6. View Prescriptions
7. Exit
Enter your choice: 6

Prescription List:
Patient ID      Medication      Dosage
1               Aspirin         100mg
```

# Test case 7:

```
Clinic Management System
1. Add Patient
2. Add Appointment
3. Add Prescription
4. View Patients
5. View Appointments
6. View Prescriptions
7. Exit
Enter your choice: 7
PS C:\Users\Gayathri S\Documents\C>
```

# DISCUSSION OF RESULTS

The system was tested with various inputs, and it successfully handled all operations without any errors. The user interface is intuitive, and the system responds quickly to user commands. The data integrity is maintained throughout the operations, ensuring that patient records are accurate and up-to-date.

# CONCLUSION

## Summary of the project:

The Clinic Management System developed in C provides a robust solution for managing patient records, appointments, and prescriptions. It offers a user-friendly interface and efficient data handling capabilities, making it a valuable tool for clinic staff.

## Future enhancement:

Future enhancements could include integration with electronic health records (EHR) systems, support for multiple users with role-based access control, and a graphical user interface (GUI) for improved user experience