Healthcare Management System (HMS)

Introduction

The **Healthcare Management System (HMS)** is a sophisticated C++ application designed to streamline the management of hospital operations. This system efficiently handles doctor and patient records, schedules appointments, and employs a color-coded console interface to enhance user interaction and readability. This document provides a detailed explanation of the program, including its functionality, structure, and output.

Program Overview

Libraries and Constants

Included Libraries

The program incorporates several standard C++ libraries:

- **`iostream`**: Facilitates standard input and output operations.
- **`string`**: Supports string manipulations.
- **`iomanip`**: Aids in controlling output formats (e.g., setting width, precision).
- **`algorithm`**: Provides functions for sorting and other algorithmic operations.
- **`limits`**: Defines properties of fundamental data types.
- **`chrono` and `thread`**: Manages time-related functions and execution pauses.

Defined Constants

To manage array sizes and colors:

- **Array Sizes**:
  - `MAX_PATIENTS = 100`: Maximum number of patients the system can handle.
  - `MAX_DOCTORS = 10`: Maximum number of doctors.
  - `MAX_DIAGNOSES = 10`: Maximum number of diagnoses.
  - `MAX_APPOINTMENTS = 5`: Maximum number of appointments per doctor.
- **ANSI Escape Codes for Colors**: Enhance the visual aspect of the console output.
  - `RESET = "\033[0m"`: Resets text color to default.
  - `BOLD = "\033[1m"`: Bold text.
  - `RED = "\033[31m"`: Red text.
  - `GREEN = "\033[32m"`: Green text.
  - `YELLOW = "\033[33m"`: Yellow text.
  - `BLUE = "\033[34m"`: Blue text.
  - `MAGENTA = "\033[35m"`: Magenta text.
  - `CYAN = "\033[36m"`: Cyan text.
  - `WHITE = "\033[37m"`: White text.

Class Definitions

The core of the program consists of two main classes: `Doctor` and `Patient`.

Doctor Class

The `Doctor` class encapsulates details about doctors, such as their name, specialization, and available appointment times.

```
class Doctor {
public:
    string name;
    string specialization;
    string availableTimes[MAX_APPOINTMENTS];

    Doctor() {}

    Doctor(string n, string s, const string t[MAX_APPOINTMENTS]) : name(n), 
specialization(s) {
        for (int i = 0; i < MAX_APPOINTMENTS; ++i) {
            availableTimes[i] = t[i];
        }
    }

    string get_doctor_name() const {
        return name;
    }

    string get_doctor_specialization() const {
        return specialization;
    }

    string* get_available_times() {
        return availableTimes;
    }
};
```

- **Attributes**:
    - `name`: The name of the doctor.
    - `specialization`: The doctor's field of specialization (e.g., cardiology, neurology).
    - `availableTimes`: An array of strings representing the doctor's available times for appointments.
- **Methods**:
    - `Doctor()`: Default constructor for creating an uninitialized `Doctor` object.
    - `Doctor(string n, string s, const string t[MAX_APPOINTMENTS])`: Parameterized constructor to initialize a doctor's name, specialization, and available times.
    - `get_doctor_name()`: Returns the doctor's name.
    - `get_doctor_specialization()`: Returns the doctor's specialization.
    - `get_available_times()`: Returns the array of available appointment times.

Patient Class

The `Patient` class encapsulates patient details, including their name, age, diagnosis, and appointment details.

```
class Patient {
    private:
        string doctorSpecialization;

    public:
        friend class Doctor;
        string name;
        int age;
        string diagnosis;
        string doctorName;
        string appointmentTime;
        string appointmentDate;

    Patient() {}

    void setpatientdata(string pname, int page, string pdiagnosis, string 
dname, string atime, string adate) {
        name = pname;
        age = page;
        diagnosis = pdiagnosis;
        doctorName = dname;
        appointmentTime = atime;
        appointmentDate = adate;
    }

    string get_patient_name() const {
        return name;
    }

    int get_patient_age() const {
        return age;
    }

    string get_patient_diagnosis() const {
        return diagnosis;
    }
};
```

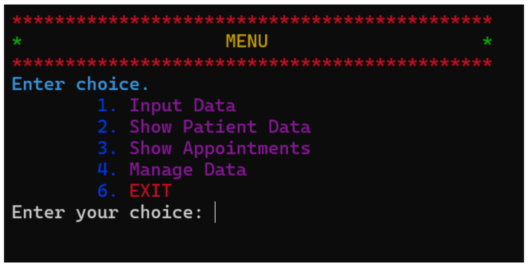
- **Attributes**:
  - `name`: The name of the patient.
  - `age`: The age of the patient.
  - `diagnosis`: The diagnosis given to the patient.
  - `doctorName`: The name of the doctor assigned to the patient.
  - `appointmentTime`: The scheduled time for the patient's appointment.
  - `appointmentDate`: The scheduled date for the patient's appointment.
  - `doctorSpecialization`: (Private) The specialization of the doctor treating the patient.
- **Methods**:

- `Patient()`: Default constructor for creating an uninitialized `Patient` object.
- `setpatientdata(string pname, int page, string pdiagnosis, string dname, string atime, string adate)`: Method to set the patient's data, including name, age, diagnosis, doctor name, appointment time, and appointment date.
- `get_patient_name()`: Returns the patient's name.
- `get_patient_age()`: Returns the patient's age.
- `get_patient_diagnosis()`: Returns the patient's diagnosis.

Program Output

Main Menu

When program start the it shows 5 options are shown in picture.

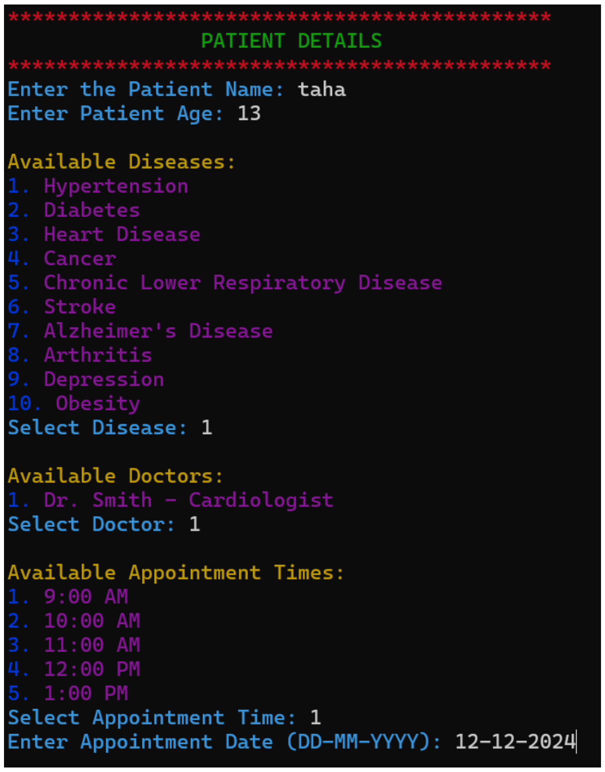


```
*************************************************
*                     MENU                      *
*************************************************
Enter choice.
        1. Input Data
        2. Show Patient Data
        3. Show Appointments
        4. Manage Data
        6. EXIT
Enter your choice:
```

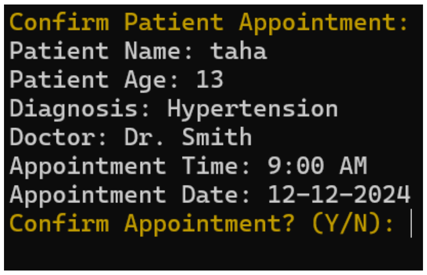
1. Input Data

When user select **option 1** then it shows the following screen in which program take inputs.

Patient data entered by the user (Patient Name, Age, available diseases, available doctors, appointment time, appointment date) then it shows the patient info that is entered and confirm the appointment if user select yes then show massage **"Appointment Confirmed. Data Entered Successfully!"** otherwise **not saved** and ask user to add another patient data if user enter **y** then program enter the input data function and take another patient detail otherwise it go to main menu.



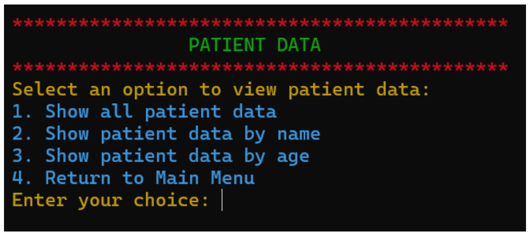
```
*************************************************
                 PATIENT DETAILS
*************************************************
Enter the Patient Name: taha
Enter Patient Age: 13

Available Diseases:
1. Hypertension
2. Diabetes
3. Heart Disease
4. Cancer
5. Chronic Lower Respiratory Disease
6. Stroke
7. Alzheimer's Disease
8. Arthritis
9. Depression
10. Obesity
Select Disease: 1

Available Doctors:
1. Dr. Smith - Cardiologist
Select Doctor: 1

Available Appointment Times:
1. 9:00 AM
2. 10:00 AM
3. 11:00 AM
4. 12:00 PM
5. 1:00 PM
Select Appointment Time: 1
Enter Appointment Date (DD-MM-YYYY): 12-12-2024
```



```
Confirm Patient Appointment:
Patient Name: taha
Patient Age: 13
Diagnosis: Hypertension
Doctor: Dr. Smith
Appointment Time: 9:00 AM
Appointment Date: 12-12-2024
Confirm Appointment? (Y/N):
```

2. Show Patient Data

When user select **option 2** then following screen appears:

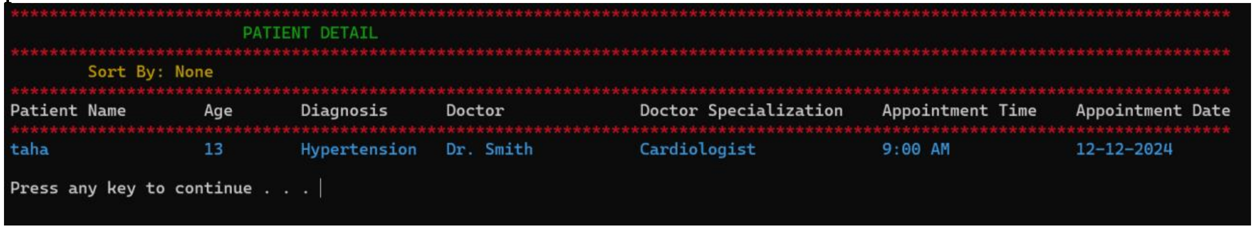


```
**************************************************
               PATIENT DATA
**************************************************
Select an option to view patient data:
1. Show all patient data
2. Show patient data by name
3. Show patient data by age
4. Return to Main Menu
Enter your choice:
```

In this screen there are 4 choices.

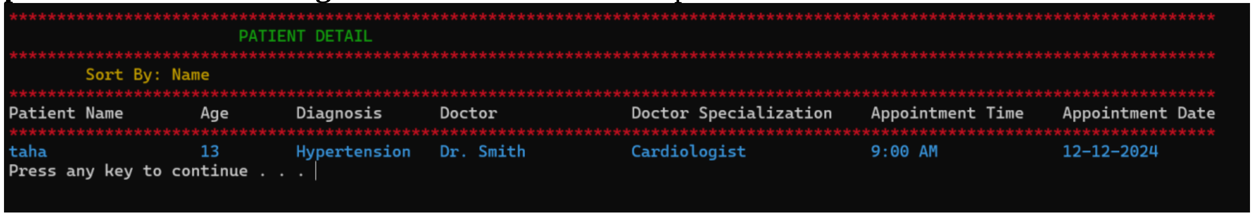
1. Show all patient data
2. Show patient data by name
3. Show patient data by age
4. Return to Main Menu

❖ If user select **option 1** then it shows the all-patients data that is entered. and return to patient data menu.



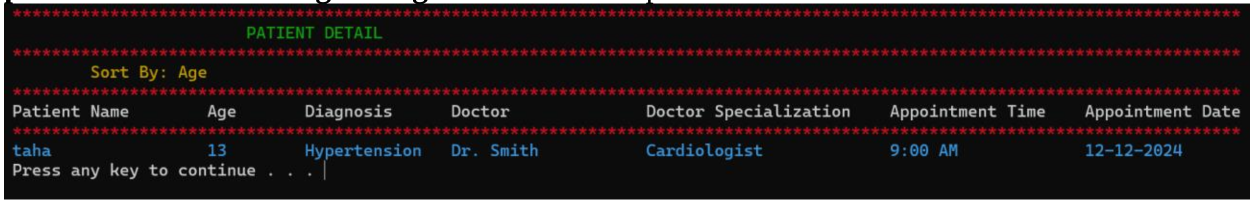
```
****************************************************************************************************
                       PATIENT DETAIL
****************************************************************************************************
        Sort By: None
****************************************************************************************************
Patient Name        Age       Diagnosis     Doctor              Doctor Specialization   Appointment Time    Appointment Date
****************************************************************************************************
taha                13        Hypertension  Dr. Smith           Cardiologist            9:00 AM             12-12-2024

Press any key to continue . . .
```

❖ If user select **option 2** then it shows patient data by name if not found then it shows **“No patient found with the given name.”** and return to patient data menu.



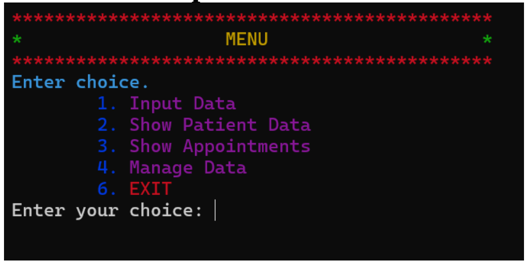
```
****************************************************************************************************
                       PATIENT DETAIL
****************************************************************************************************
        Sort By: Name
****************************************************************************************************
Patient Name        Age       Diagnosis     Doctor              Doctor Specialization   Appointment Time    Appointment Date
****************************************************************************************************
taha                13        Hypertension  Dr. Smith           Cardiologist            9:00 AM             12-12-2024
Press any key to continue . . .
```

❖ If user select **option 3** then it shows patient data by age if not found then it shows **“No patient found with the given age.”** and return to patient data menu.



```
****************************************************************************************************
                       PATIENT DETAIL
****************************************************************************************************
        Sort By: Age
****************************************************************************************************
Patient Name        Age       Diagnosis     Doctor              Doctor Specialization   Appointment Time    Appointment Date
****************************************************************************************************
taha                13        Hypertension  Dr. Smith           Cardiologist            9:00 AM             12-12-2024
Press any key to continue . . .
```

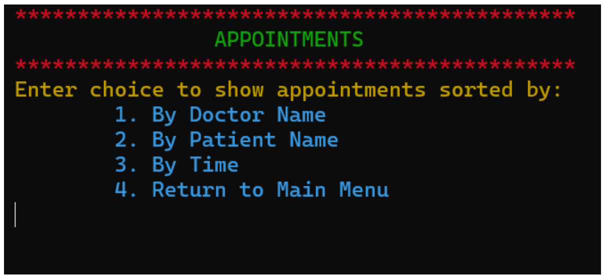
❖ If user select **option 4** then it return to main menu.



```
*********************************************
*                    MENU                   *
*********************************************
Enter choice.
        1. Input Data
        2. Show Patient Data
        3. Show Appointments
        4. Manage Data
        6. EXIT
Enter your choice:
```

3. Show Appointments

When user select **option 3** then following screen appears:

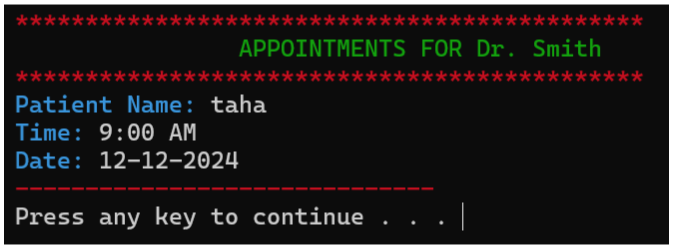


```
**********************************************
              APPOINTMENTS
**********************************************
Enter choice to show appointments sorted by:
        1. By Doctor Name
        2. By Patient Name
        3. By Time
        4. Return to Main Menu
```

In this screen there are 4 choices.

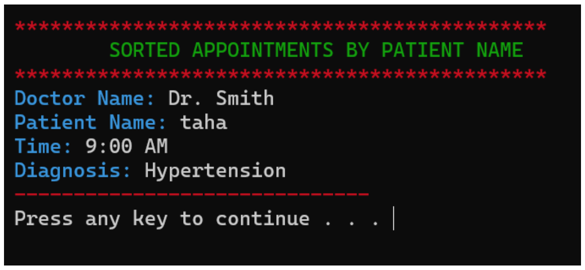
1. By Doctor Name
2. By Patient Name
3. By Time
4. Return to Main Menu

- ❖ If user select **option 1** then it shows appointments by Doctor Name if not found then it shows **"No appointments available for this doctor."** and return to appointments data menu.



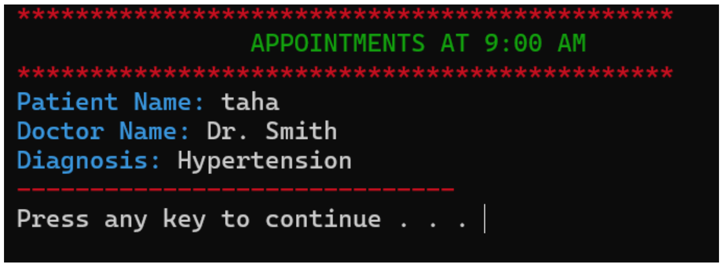
```
*******************************************
           APPOINTMENTS FOR Dr. Smith
*******************************************
Patient Name: taha
Time: 9:00 AM
Date: 12-12-2024
-----------------------------
Press any key to continue . . .
```

- ❖ If user select **option 2** then it shows appointments by Patient Name if not found then it shows **"No appointments available for this patient."** and return to appointments data menu.



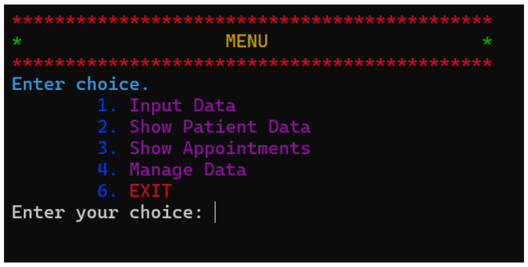
```
*********************************************
      SORTED APPOINTMENTS BY PATIENT NAME
*********************************************
Doctor Name: Dr. Smith
Patient Name: taha
Time: 9:00 AM
Diagnosis: Hypertension
------------------------------
Press any key to continue . . .
```

- ❖ If user select **option 3** then it shows appointments by Time if not found then it shows **"No appointments available for this time."** and return to appointments data menu.



```
*********************************************
            APPOINTMENTS AT 9:00 AM
*********************************************
Patient Name: taha
Doctor Name: Dr. Smith
Diagnosis: Hypertension
------------------------------
Press any key to continue . . .
```

- ❖ If user select **option 4** then it return to main menu.

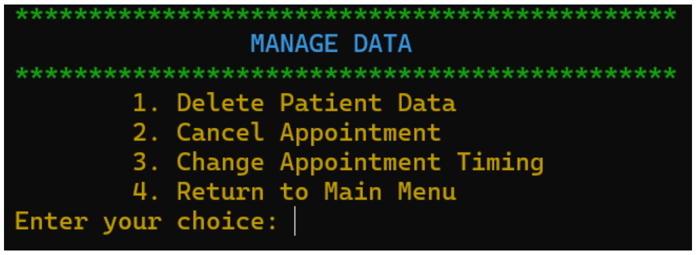


```
*********************************************
*                    MENU                   *
*********************************************
Enter choice.
        1. Input Data
        2. Show Patient Data
        3. Show Appointments
        4. Manage Data
        6. EXIT
Enter your choice:
```

4. Manage Data

When user select **option 4** then following screen appears:

If there is no patient entered then it shows error **"No Patients available."** Otherwise, it shows following screen.



```
***********************************************
                  MANAGE DATA
***********************************************
        1. Delete Patient Data
        2. Cancel Appointment
        3. Change Appointment Timing
        4. Return to Main Menu
Enter your choice: 
```

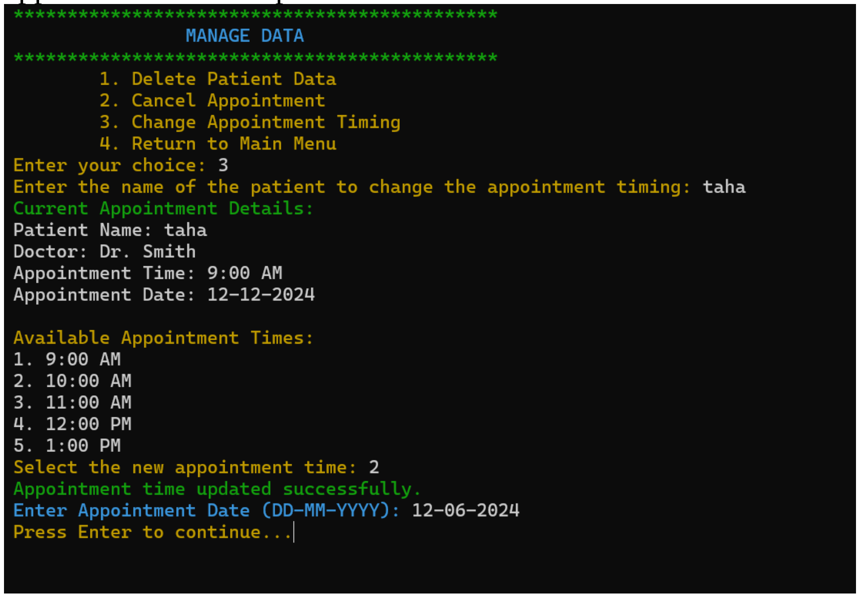
In this screen there are 4 choices:

1. Delete Patient Data
2. Cancel Appointment
3. Change Appointment Timing
4. Return to Main Menu

❖ If user select **option 1** then it asks user to enter the patient's name if not found then it shows **"Patient not found."** and return to Manage data menu. Otherwise, it delete that patient data.

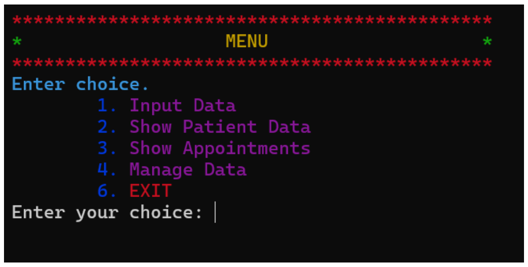
❖ If user select **option 2** it asks user to enter the patient's name if not found then it shows **"Patient not found."** and return to Manage data menu. Otherwise, it cancel that patient appointments.

❖ If user select **option 3** then it asks user to enter the patient's name if not found then it shows **"Patient not found."** and return to Manage data menu. Otherwise, it change the appointments of that patients.



```
*********************************************
                MANAGE DATA
*********************************************
        1. Delete Patient Data
        2. Cancel Appointment
        3. Change Appointment Timing
        4. Return to Main Menu
Enter your choice: 3
Enter the name of the patient to change the appointment timing: taha
Current Appointment Details:
Patient Name: taha
Doctor: Dr. Smith
Appointment Time: 9:00 AM
Appointment Date: 12-12-2024

Available Appointment Times:
1. 9:00 AM
2. 10:00 AM
3. 11:00 AM
4. 12:00 PM
5. 1:00 PM
Select the new appointment time: 2
Appointment time updated successfully.
Enter Appointment Date (DD-MM-YYYY): 12-06-2024
Press Enter to continue...
```

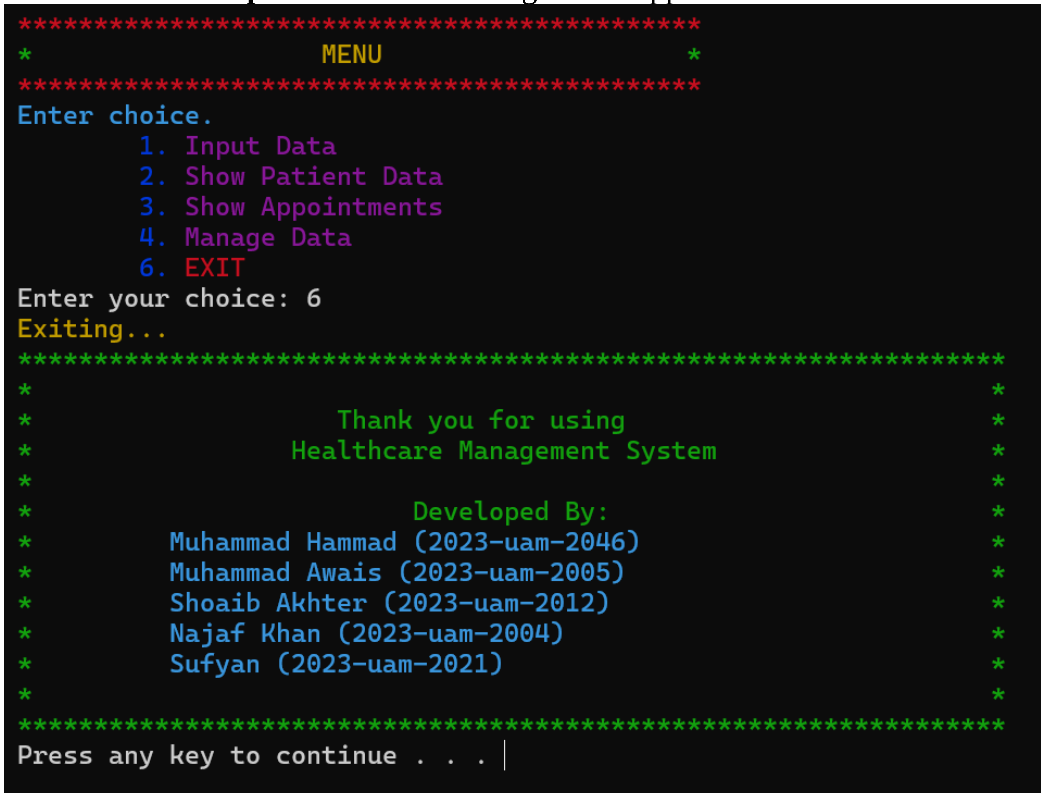
❖ If user select **option 4** then it return to main menu.



```
*********************************************
*                   MENU                    *
*********************************************
Enter choice.
        1. Input Data
        2. Show Patient Data
        3. Show Appointments
        4. Manage Data
        6. EXIT
Enter your choice: 
```

5. **EXIT**

When user select **option 6** then following screen appears:



```
*********************************************
*                   MENU                    *
*********************************************
Enter choice.
        1. Input Data
        2. Show Patient Data
        3. Show Appointments
        4. Manage Data
        6. EXIT
Enter your choice: 6
Exiting...
*****************************************************************
*                                                               *
*                     Thank you for using                       *
*                  Healthcare Management System                 *
*                                                               *
*                          Developed By:                        *
*         Muhammad Hammad (2023-uam-2046)                       *
*         Muhammad Awais (2023-uam-2005)                        *
*         Shoaib Akhter (2023-uam-2012)                         *
*         Najaf Khan (2023-uam-2004)                            *
*         Sufyan (2023-uam-2021)                                *
*                                                               *
*****************************************************************
Press any key to continue . . .
```

Program Functionality

Initialization and Input

The program starts by creating instances of the `Doctor` and `Patient` classes. It initializes necessary data such as doctors' names, specializations, available times, as well as patients' names, ages, diagnoses, and appointment details. These data points are crucial for the system's operations.

Conclusion

The **HMS (Healthcare Management System)** program is a robust tool for managing hospital data. Its design ensures efficient handling of doctors' and patients' records. By using classes and methods, the program maintains clear and modular code structure, making it easy to manage and extend.

The use of ANSI escape codes enhances the user experience by providing a visually appealing console output. This color-coded interface makes the information easier to read and interpret, thus improving the overall usability of the system.