# PSP [20ES104] COURSE PROJECT REPORT

**On**

# "Medical Appointment Management"

## Developed By:

| H.T.NO | STUDENT NAME |
|---|---|
| 2203A52005 | Sreeja .B |

## Under the Guidance of

Mr. Srinivas Aluvala, M.Tech. (Ph.D)

Assistant Professor

Submitted to

Department of Computer Science and Artificial Intelligence

SR University

Ananthasagar(V), Hasanparthy(M), Hanamkonda(Dist.) – 506371

www.sru.edu.in

**June 2023**

## Department of Computer Science and Artificial Intelligence

## CERTIFICATE

This is to certify that the PSP course project report entitled **“MEDICAL APPOINTMENT MANAGEMENT”** is a record of bonafide work carried out by the student(s) **SREEJA.B** bearing roll number(s) **2203a52005** of Computer Science and Artificial Intelligence department during the academic year 2022-23.

**Supervisor**

(Srinivas Aluvala)

# INDEX

## ***PROBLEM STATEMENT:***

Developing a C Application for medical appointment. The system should allow users to add appointments, cancel appointments, and view the appointment history.

→The system should support the following functionalities:

### 1.Declaration

A structure named appointment is declared for storing appointment data of the user.

### 2. Function prototypes

The functions named add appointment, cancel appointment ,and view appointment history are declared.

### 3.Main function

In main function, an array of Appointment structures called appointments is created, and the variable 'num appointments' is initialized to 0. The option variable is used to store the user's menu choice.

### 4.Looping Statement

Inside the while loop, the user is presented with a menu of options: add an appointment, cancel an appointment, or exit the program. The user's choice is stored in the option variable using scanf.

### 5.Switch case

The switch statement is used to perform the corresponding action based on the user's choice. If the user chooses to add an appointment, the add appointment function is called, passing the appointments array and num appointments as arguments. If the user chooses to cancel an appointment, the cancel appointment function is called. If the user chooses to exit, the program terminates using exit(0).

## *MODULES:*

- ❖ In this program the header files used are:
- <stdio.h>→It consists of the input and output functions like printf and scanf.
- <stdib.h>→Standard Library of C programming language declares various functions for type conversions, memory allocation, algorithms, and other similar cases.
- <string.h>→It is used for string handling functions and also for memory handling.
- ❖ A structure named appointment is declared which consists of the following structure members:

->Patient name(character data type)

->Doctor name(character data type)

->Date(character data type)

->Time(character data type)

- ❖ The memory allocation will be done in this program dynamically
- ❖ The application asks the person who runs theprogram to enter the details required for the medical appointment (like Name, date, time….).

❖ The following modules are used in this application:

1. **Add Appointment**: Users should be able to add a new appointment by providing the patient name, doctor name, date (in the format dd/mm/yyyy), and time (in the format hh:mm).

2. **Cancel Appointment:**

   - ➢ Users should be able to cancel an existing appointment by entering the patient name associated with the appointment.
   - ➢ If the appointment is found, it should be removed from the system, and a confirmation message should be displayed.
   - ➢ If no appointment is found for the given patient name, an appropriate message should be displayed.

3.**View Appointment History:** Users should be able to view the history of all appointments stored in the system.

- ➢ The appointment history should include the patient name, doctor name, date, and time for each appointment.

## *KNOWLEDGE REQUIRED TO DEVELOP THIS APPLICATION*

- **Control Statements** (if, switch)

  Control statements in C help the computer execute a certain logical statement and decide whether to enable the control of the flow through a certain set of statements or not.

- **Loop Statements** (while, for)

  We use the loop control statements in C language for performing various loop operations until we find the condition given in a program to be true. The control comes out of a loop statement when the condition given to us turns out to be false.

- **Arrays** (1D/2D-arrays)

  Arrays are the collection of data items of same data type. They help us to store multiple values.

- **Functions** (user defined functions)

A function is a block of code which only runs when it is called.
when it is called. You can pass data, known as parameters, into a function. Functions are used to perform certain actions, and they are important for reusing Code.

- **Structure** (structures)

  Structure is the collection of data items of different data types. It helps in storing multiple values of various data types.

- **Pointers** ( pointers to structures)

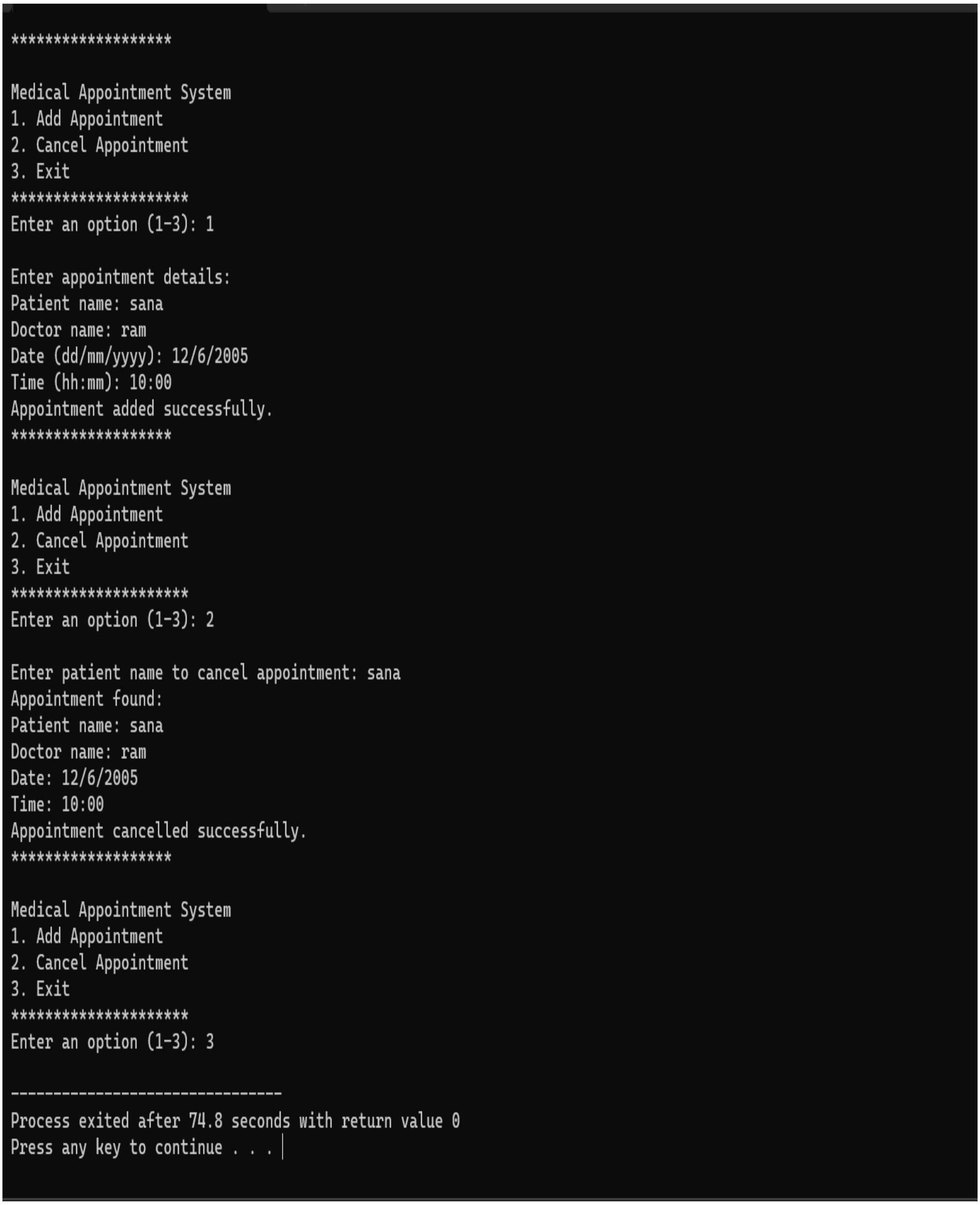
  Pointer is a user defined data type that creates special types of variables which can hold the address of primitive data type like char, int, float, double or user defined data type like function, pointer etc. or derived data type like array, structure, union, enum.

## *SOURCE CODE [.C FILE]:*

```
//Program: To demonstrate medical appointment
//Author: Sreeja. B
//HT.NO.: 2203A52005
//Date of project: 20-05-2023

 #include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAXAPPOINTMENTS 100

// Structure for storing appointment data

struct Appointment {
   char patientname[50];
   char doctorname[50];
   char date[20];
   char time[20];
};

// Function prototypes

void addappointment(struct Appointment *appointments, int *numappointments);
void cancelappointment(struct Appointment *appointments, int *numappointments);
void viewappointmenthistory(struct Appointment *appointments, int numappointments);
int main() {
   struct Appointment appointments[MAXAPPOINTMENTS];
   int numappointments = 0;
   int option;
   while (1) {
      printf(“******************/n”);
      printf("\nMedical Appointment System\n");
      printf("1. Add Appointment\n");
      printf("2. Cancel Appointment\n");
      printf("3. Exit\n");
     printf(“********************/n);
     printf("Enter an option (1-3");
     scanf("%d", &option);

      switch (option) {
         case 1:
            addappointment(appointments, &numappointments);
            break;
         case 2:
            cancelappointment(appointments, &numappointments);
            break;
         case 3:
             exit(0);
            default:
                printf("Invalid option. Please try again.\n");
        }
    }
    return 0;
}


// Function to add a new appointment

void addappointment(struct Appointment *appointments, int *numappointments) {
    if (*numappointments == MAXAPPOINTMENTS) {
        printf("Maximum number of appointments reached.\n");
        return;
    }

    printf("\nEnter appointment details:\n");
    printf("Patient name: ");
    scanf("%s", appointments[*num appointments].patientname);
    printf("Doctor name: ");
    scanf("%s", appointments[*num appointments].doctorname);
    printf("Date (dd/mm/yyyy): ");
    scanf("%s", appointments[*num appointments].date);
    printf("Time (hh:mm): ");
    scanf("%s", appointments[*num appointments].time);

    printf("Appointment added successfully.\n");

    (*num appointments)++;
}

// Function to cancel an appointment

  void cancel appointment(struct Appointment *appointments, int     *numappointments) {
    char patientname[50];
    int i, j, appointmentfound = 0;
    printf("\nEnter patient name to cancel appointment: ");
    scanf("%s", patientname);
    for (i = 0; i < *numappointments; i++) (
        if (strcmp(appointments[i].patientname, patientname) == 0) {
            printf("Appointment found:\n");
            printf("Patient name: %s\n", appointments[i].patientname);
            printf("Doctor name: %s\n", appointments[i].doctorname);
            printf("Date: %s\n", appointments[i].date);
            printf("Time: %s\n", appointments[i].time);

            // Shift all appointments after the cancelled appointment up   one index

            for ( j = i; j < *numappointments - 1; j++) {
                appointments[j] = appointments[j+1];
            }
            (*numappointments)--;
            Appointment found = 1;
            printf("Appointment cancelled successfully.\n");
            break;
}

    }




 if (!appointment found) {
        printf("No appointment found for patient '%s'.\n", patientname);
    }
}

// Function to view appointment
```

********************

## *RESULTS[OUTPUT]:*



```
*******************

Medical Appointment System
1. Add Appointment
2. Cancel Appointment
3. Exit
*********************
Enter an option (1-3): 1

Enter appointment details:
Patient name: sana
Doctor name: ram
Date (dd/mm/yyyy): 12/6/2005
Time (hh:mm): 10:00
Appointment added successfully.
*******************

Medical Appointment System
1. Add Appointment
2. Cancel Appointment
3. Exit
*********************
Enter an option (1-3): 2

Enter patient name to cancel appointment: sana
Appointment found:
Patient name: sana
Doctor name: ram
Date: 12/6/2005
Time: 10:00
Appointment cancelled successfully.
*******************

Medical Appointment System
1. Add Appointment
2. Cancel Appointment
3. Exit
*********************
Enter an option (1-3): 3

--------------------------------
Process exited after 74.8 seconds with return value 0
Press any key to continue . . .
```