

# EEL ACTIVITY NO.5

SARVESH MANDE B25CE1205 CE3 C

## Research:

In C language, a database can be created using arrays, structures, or files. Structures are used to group related data together, so all details of one record can be stored in a single unit. Arrays of structures can store many records, similar to rows in a database table.

For saving data permanently, C programs can use files. Functions like fopen, fread, fwrite, and fclose allow the program to store and read records even after the program is closed. This method is simple and works well for small programs like cinema booking systems, student records, or inventory management.

Using structures and files, a C program can add new records, display all records, search for specific records, update records, and sort them. While this is not as advanced as modern database systems, it provides the basic features of a database in a simple way.

Sources:

1. <https://www.geeksforgeeks.org/file-handling-c-classes/>
2. <https://www.programiz.com/c-programming/c-structures>
3. [https://www.tutorialspoint.com/cprogramming/c\\_file\\_io.htm](https://www.tutorialspoint.com/cprogramming/c_file_io.htm)
4. <https://en.wikipedia.org/wiki/Database>

## Analyze:

The Cinema Booking System is a simple program that helps manage movie ticket bookings. It stores customer details, movie names, show times, number of seats, ticket prices, and show dates using structures. Each booking is treated as a single record, and all bookings are stored in an array, which works like a basic database.

The program allows users to add new bookings, display all existing bookings, sort bookings by ticket price, and exit the program. Sorting is done using a simple

method to arrange bookings from lowest to highest price. Searching can be added to find a specific booking easily.

This program demonstrates basic programming concepts like structures, arrays, loops, and conditional statements. It shows how these concepts can be applied to solve real-world problems, like managing cinema bookings in an organized and efficient way.

## Ideation

The program is designed to manage cinema ticket bookings. Users can add new bookings, view all bookings, and sort them by ticket price. It uses structures to store booking details and arrays to hold multiple records. The system provides a simple menu interface for easy interaction.

## Algorithm:

1. Start
2. Declare a structure to store booking details including customer name, movie name, show time, number of seats, ticket price, and show date
3. Declare an array of structures to store multiple bookings
4. Initialize a variable to keep track of the number of bookings
5. Repeat the following steps until the user chooses to exit:
  - a. Display the menu with options to add booking, display bookings, sort bookings by price, and exit
  - b. Read the user's choice
  - c. If the choice is add booking:
    - i. Ask the user how many bookings they want to add
    - ii. For each booking, input customer name, movie name, show time, number of seats, ticket price, and show date

- iii. Store the booking details in the array
  - d. If the choice is display bookings:
    - i. Check if there are any bookings
    - ii. If yes, display the details of all bookings one by one
    - iii. If no, show a message that no bookings are available
  - e. If the choice is sort bookings by price:
    - i. Check if there are any bookings
    - ii. If yes, use a simple sorting method to arrange bookings from lowest to highest price
    - iii. Show a message that bookings are sorted
    - iv. If no, show a message that no bookings are available
  - f. If the choice is exit:
    - i. Stop

## Build:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
struct Cinema ;
```

```
char customer_name[20];
```

```
char movie_name[20];
```

```
char show_time[15];
```

```
int seats;
```

```
int ticket_price;
```

```
int date; // DDMMYYYY
```

```
} booking[100], temp;
```

```
int n = 0;
```

```
int main() {
```

```
    int choice, i, j;
```

```
int running = 1; // Control the loop

while (running) {

    printf("\nMENU\n");

    printf("1. Add Booking\n");

    printf("2. Display Bookings\n");

    printf("3. Sort Bookings by Price\n");

    printf("4. Exit\n");

    printf("Enter your choice: ");

    scanf("%d", &choice);
```

```
switch (choice) {  
  
    case 1:  
  
        printf("How many bookings do you want to add? ");  
  
        scanf("%d", &n);  
  
        for (i = 0; i < n; i++) {  
  
            printf("\n--- Booking %d ---\n", i + 1);  
  
            printf("Customer Name: ");  
  
            scanf("%s", booking[i].customer_name);  
  
            printf("Movie Name: ");  
  
            scanf("%s", booking[i].movie_name);  
  
            printf("Show Time: ");
```

```
scanf("%s", booking[i].show_time);

printf("Number of Seats: ");

scanf("%d", &booking[i].seats);

printf("Total Ticket Price: ");

scanf("%d", &booking[i].ticket_price);

printf("Show Date (DDMMYYYY): ");

scanf("%d", &booking[i].date);

}

break;
```

case 2:

```
if (n == 0) {  
  
    printf("No bookings available.\n");  
  
    break;  
  
}  
  
for (i = 0; i < n; i++) {  
  
    printf("\nBooking %d\n", i + 1);  
  
    printf("Customer Name: %s\n", booking[i].customer_name);  
  
    printf("Movie Name: %s\n", booking[i].movie_name);  
  
    printf("Show Time: %s\n", booking[i].show_time);  
  
    printf("Seats: %d\n", booking[i].seats);    printf("Ticket Price: %d\n",  
booking[i].ticket_price);  
  
    printf("Date: %d\n", booking[i].date);
```

```
}
```

```
break;
```

```
case 3:
```

```
if (n == 0) {
```

```
printf("No bookings to sort.\n");
```

```
break;
```

```
}
```

```
for (i = 0; i < n - 1; i++) {
```

```
    for (j = 0; j < n - i - 1; j++) {
```

```
        if (booking[j].ticket_price > booking[j + 1].ticket_price) {
```

```
temp = booking[j];  
  
booking[j] = booking[j + 1];  
  
booking[j + 1] = temp;  
  
}  
  
}  
  
printf("Bookings sorted by ticket price.\n");  
  
break;  
  
case 4:  
  
printf("Exiting... Thank you!\n");
```

```
running = 0; // Stop the loop
```

```
break;
```

```
default:
```

```
    printf("Invalid choice. Try again.\n");
```

```
}
```

```
}
```

```
return 0;
```

```
}
```

## Output

```
Output

MENU
1. Add Booking
2. Display Bookings
3. Sort Bookings by Price
4. Exit
Enter your choice: 1
How many bookings do you want to add? 1

--- Booking 1 ---
Customer Name: sarvesh
Movie Name: kantara
Show Time: 12.00
Number of Seats: 5
Total Ticket Price: 500
Show Date (DDMMYYYY): 12122025

MENU
1. Add Booking
2. Display Bookings
3. Sort Bookings by Price
4. Exit
Enter your choice: 2
|
Booking 1
Customer Name: sarvesh
Movie Name: kantara
Show Time: 12.00
Seats: 5
Ticket Price: 500
Date: 12122025

MENU
1. Add Booking
2. Display Bookings
3. Sort Bookings by Price
4. Exit
Enter your choice: 4
Exiting... Thank you!

== Code Execution Successful ==
```

## Implementation

[https://github.com/sarvesh00719/cinema-database  
manager-with-the-sorting-and-options.git](https://github.com/sarvesh00719/cinema-database-manager-with-the-sorting-and-options.git)