A MINI PROJECT REPORT

for

MINI PROJECT IN C (19CSE39)

MOVIE TICKET BOOKING SYSTEM

submitted by

S. YERRINATHA REDDY

1NH19ME096

III-D

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

NEW HORIZON COLLEGE OF ENGINEERING

Certificate

This is to certify that the mini project work titled

MOVIE TICKET BOOKING SYSTEM

Submitted in partial fulfillment of the degree of Bachelor of Engineering in Computer Science and Engineering

Submitted by

S. YERRINATHA REDDY

1NH19ME096

DURING

ODD SEMESTER 2020-2021

For

19CSE3

Signature of Reviewer Signature of HOD

SEMESTER END EXAMINATION

Name of the Examiner	*Signature with date*
1.	
2.	

MOVIE TICKET BOOKING SYSTEM

ORIGINALITY REPORT

10%	%	10%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Mustafa Aksu, Ali Karcı. "Pyramid Search: Skip Ring Data Structure-Based New Searching Algorithm", Journal of Circuits, Systems and Computers, 2018 Publication	2%
2	Mustafa Aksu, Ali Karci. "Skip List Data Structure Based New Searching Algorithm and Its Applications: Priority Search", International Journal of Advanced Computer Science and Publication	2%
3	W. Arthur Chapman. "Mastering C Programming", Springer Science and Business Media LLC, 1991 Publication	1%
4	Saumyendra Sengupta, Carl Phillip Korobkin. "C++", Springer Science and Business Media LLC, 1994 Publication	1%
5	Kancharla Bullibabu, Md AbidAli, K.	1%

Veeranjaneyulu. "Characterization and production of thermal Insulating Fired Clay Bricks with admixture of Bagasse and Palmyra fruit fiber", Materials Today: Proceedings, 2018
Publication

6	V. Ramachandran. "Lower bounds for parallel computation on linked structures", Proceedings of the second annual ACM symposium on Parallel algorithms and architectures - SPAA 90 SPAA 90, 1990 Publication	1 %
7	Xingni Zhou, Zhiyuan Ren, Yanzhuo Ma, Kai Fan, Xiang Ji. "Volume 1: Data structures based on linear relations", Walter de Gruyter GmbH, 2020 Publication	1 %
8	B. A. E. Meekings, T. P. Kudrycki, M. D. Soren. "Chapter 3 Functions", Springer Science and Business Media LLC, 1993 Publication	<1 %
9	Friesen, Jeff. "Exploring the Basic APIs Part 1", Learn Java for Android Development, 2013. Publication	<1 %
10	Tony Royce. "Chapter 3 Complex Data Structures", Springer Science and Business Media LLC, 1996 Publication	<1 %

11	W. Arthur Chapman. "Mastering C++ Programming", Springer Science and Business Media LLC, 1998 Publication	<1%
12	Sanchita Ghosh, Amit Konar. "Call Admission Control in Mobile Cellular Networks", Springer Science and Business Media LLC, 2013 Publication	<1%
13	Avelino J. Gonzalez. "Computer Programming in C for Beginners", Springer Science and Business Media LLC, 2020 Publication	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		

ABSTRACT

The purpose of developing this software project is to fully make easy system for an organization. All this work requires a lot of paper work, is extremely time-consuming job, and accordingly costly as well, as they have to hire more man power. Since, there is always a risk of human errors present in a manual system so the chances of errors are very high and to figure out such errors is also a very lengthy procedure.

Therefore, I decided to switch from a manual system to an automated computerized management system. This software is capable of calculating Ticket charges with reduction of discounts according to that organization. This application is developed in C using linked list. This program is used to display information of costumers and it avoids errors in calculating the prices.

With these things in mind, we overhauled the existing ticket booking system and necessary improvements to narrow down the process. This help in easier user interface and more organized manner of managing data, retrieving data and manipulating data. this helps in saving time and reduce huge paper work.

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would Be impossible without the mention of the people who made it possible, whose constant guidance and encouragement crowned our efforts with success.

I have great pleasure in expressing gratitude to Dr. Mohan Manghnani, Chairman of New Horizon Educational Institutions for providing necessary infrastructure and creating good environment.

I take this opportunity to express my profound gratitude to Dr. Manjunatha, Principal NHCE, for his constant support and encouragement.

I would also like to thank Dr. B. Rajalakshmi , senior assistant Professor Department of Computer Science and Engineering, for her constant support.

I express my gratitude to Mr. Praveen Pawaskar, Designation, my project guide, for constantly monitoring the development of the project and setting up precise deadlines. His valuable suggestions were the motivating factors in completing the work.

S. Yerrinatha reddy
(1NH19ME096)

CONTENTS **Pg. No**

<u>**LIST OF FIGURES**</u>

CHAPTER 1

INTRODUCTION

1.1 PROBLEM DEFINITION

Payment of tickets for the movies is that the main thing of any System of a corporation. The answer is to require care of the calculation of ticket prices as per rules of the corporate, tax calculation and a few of the varied deductions to be done from the customer’s including deductions like tax and fund deductions. It’s to get pay-slip, cheque summary and a few other reports. So, we'd like to calculate ticket accordingly.

1.2 OBJECTIVES

Main goal of developing movie ticket price calculator is to supply a simple way not only to automate all functionalities involved managing funds and salary for the workers of Company, But also to supplyfull functional to management of Company with the small print about usage of leave facility. Also calculate the salary of the worker accordingly to their position within the company whether or not they belong to daily wagers or monthly wagers.

1.3 METHODOLOGY TO BE FOLLOWED

The full-time employees included are those that work for an employer on an hourly basis. Total money compensation is defined as all cash payments earned by an employee during a year of employment. Salary calculator has detailed data for every individual worker who provided information on their compensation and compensable factors . Individual employee can use the salary calculator to know how the market rises for his or her services as employee are changing on the average.

1.4 EXPECTED OUTCOMES

First we need to input the name, id and the number of tickets sold , after giving the required details the system will ask for the type of payment . after the user gives the type of payment the user want the system will calculate the price according to the number of hours the movie tickets. After that the reduction of money due to taxes comes into action.

1.5 HARDWARE AND SOFTWARE REQUIREMENTS

1.5.1 HARDWARE REQUIREMENTS

* RAM: 1 GB
* Processor: Intel core

1.5.2 SOFTWARE REQUIREMENTS

* Operating System (Windows 10 or any other OS)
* Required Software: Turbo C++ or Dev C++

CHAPTER 2

DATA STRUCTURE

2.1 ARRAYS

An array is a collection of items stored at contiguous memory locations. The idea is to store multiple items of the same type together. This makes it easier to calculate the position of each element by simply adding an offset to a base value, i.e the memory location of the first element of the array generally denoted by the name of the array .For simplicity, we can think of an array a fleet of stairs where on each step is placed a Value (lets say one of your friends). Here , you can identify the location of any of your friends by simply knowing the count of the step they are on.

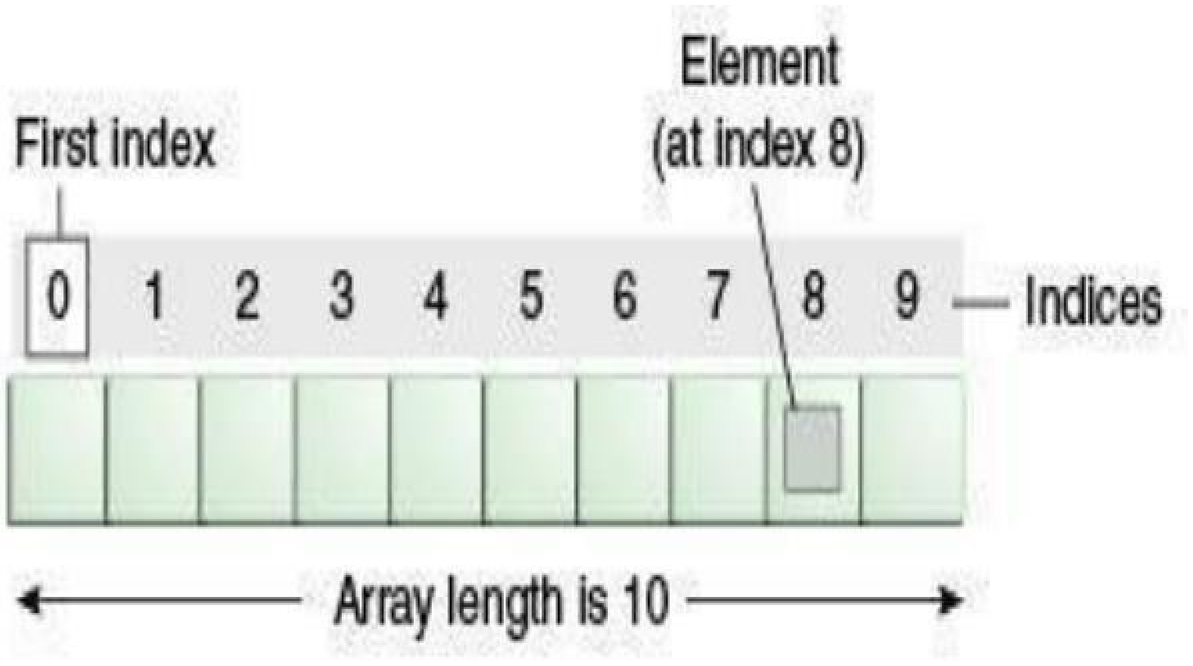


Fig 2.1.1 – Representation of arrays

Advantages of using arrays:

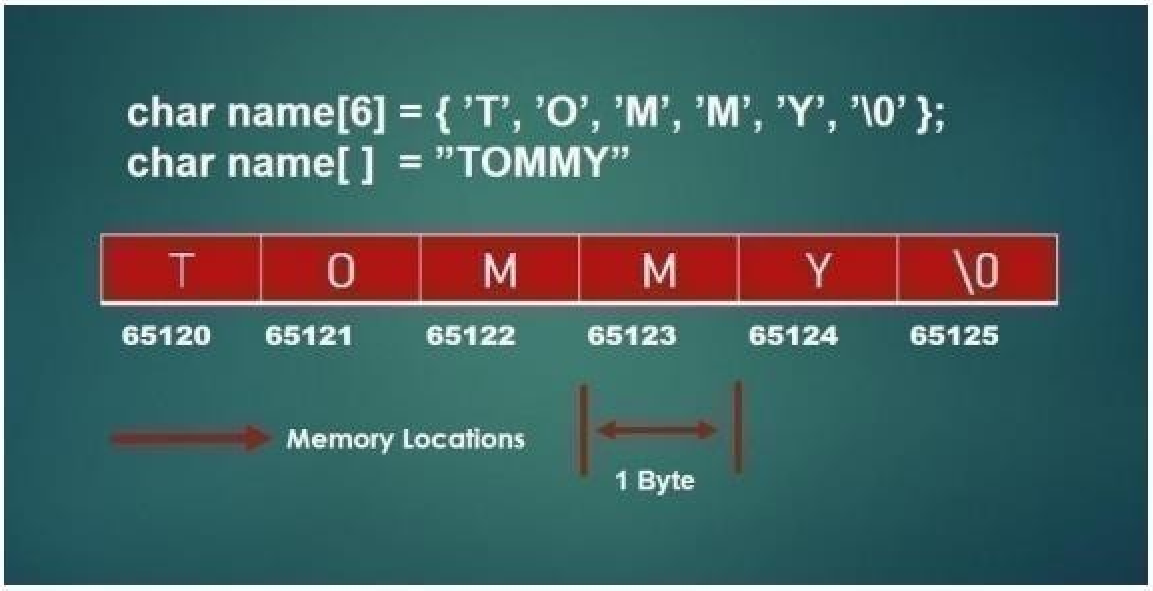
- Arrays allow random access of elements. This makes accessing elements by position faster.
- Arrays have better cache locality that can make a pretty big difference in performance.

Applications: -

* Implementation mathematical vectors and matrices, as well as other kinds of rectangular tables.
*Implementation other data structures, like heaps, hash tables, queues and stacks.

2.2 CHARACTER ARRAY

String is a sequence of characters that is treated as a single data item and terminated by null character '\0'. Remember that C language does not support strings as a data type. A string is actually one-dimensional array of characters in C language. These are often used to create meaningful and readable programs.



```c
char name[6] = { 'T', 'O', 'M', 'M', 'Y', '\0' };
char name[ ]  = "TOMMY"
```

Fig-2.2.2 Representation of character array

2.3.LINKED LIST

* A linked list is a linear data structure where each element is a separate object .

* Each element (we will call it a node) of a list is comprising of two items the data and a reference to the next node.The last node has a reference to null.The entry point into linked list is called the head of the list. It should be noted that head is not a separate node, but the reference to the first node . If the list is empty then the head is a null reference . linked list is a dynamic data structure. The number of nodes in a list is not fixed and can grow and shrink on demand. Any application which has to deal with an unknown number of objects will need to use a linked list. The principal benefit of a linked list over a conventional array is that the list elements can be easily inserted or removed without reallocation or reorganization of the entire structure because the data items need not be Stored contiguously in memory or on disk,while restructuring an array at run-time is much more expensive operation . Linked lists allow insertion and removal of nodes at any point in the list, and allow doing so with constant number of operations by keeping the link previous to the link being added or removed in memory during list traversal. A circular linked list is a variation of the linked list.It is a linked list whose nodes are connected in such a way that it forms a circle. In the circular linked list, the next pointer of the last node is not set to null but it contains the address of the first node thus forming a circle.The arrangement shown below is for a singly linked list.

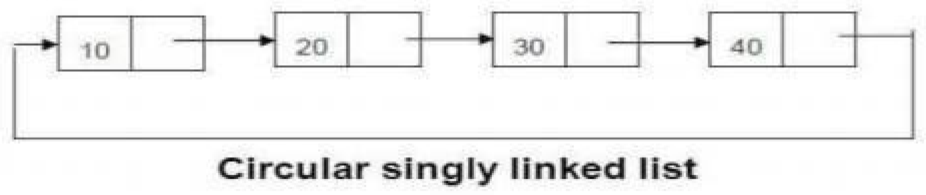


Fig-2.3.2 Circular singly linked list

pointer of first node is connected to the last node while the next pointer of the last node is connected to the first node.

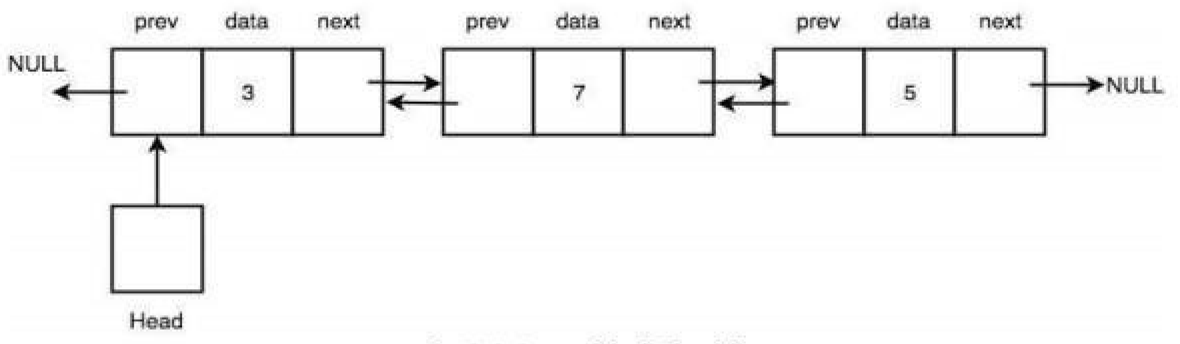


Fig-2.3.3 Doubly linked list

Doubly linked list is a type of linked list in which each node apart from storing its data has two links.The first link points to the previous node in the list and the second link points to the next node in the list . The first node of the list has its previous link pointing to NULL similarly
the last node of the list has its next node pointing to NULL. Doubly linked list is a type of linked list in which each node apart from storing its data has two links. The first link points to the previous node in the list and the second link points to the next node in the list. The first node of the list has its previous link pointing to NULL similarly the last node of the list has its next node pointing to NULL. Doubly linked list is a type of linked list in which each node apart from storing its data has two links. The first link points to the previous node in the list and the second link points to the next node in the list. The first node o f the list has its previous link pointing to NULL similarly the last node of the list has its next node pointing to NULL.

Advantages: -

* It is a Dynamic data Structure.

*It can increase and decrease size during runtime.

* Faster Access time

*Implementation of stacks and queues.

Applications: -

* Implementation Stacks, Queues.
* Implementation Graphs.
* Implementation of Hash Tables: - Each Bucket of the hash table is a linked list
* Undo functionality in Photoshop or Word. Linked list of states.
* A polynomial can be represented in a linked list.
* Polynomial operation, like addition or multiplication of polynomials, can be done using linked list.

Operations which can be formed are:

* Creation of a list

Creation operation is used to create a linked list Once a linked list is created with one node Insertion operation can be used to add more elements in a node. if(head==NULL)

```
{
new1=(struct list *) malloc (sizeof(struct list));
printf("\      nENTER      NAME:      ");
scanf("%s",≠w1->n)                          ;
new1→ptr=NULL;  head=new1;
}
```

• Insert

Insertion operation is used to insert a new node at any specified location of the linked list A new node may be inserted:

1) At the beginning of the linked list
2) At any specified position between in a linked list
3) At the end of linked list curr=head; while(curr->ptr!=NULL) curr=curr->ptr; curr→ptr=new1;

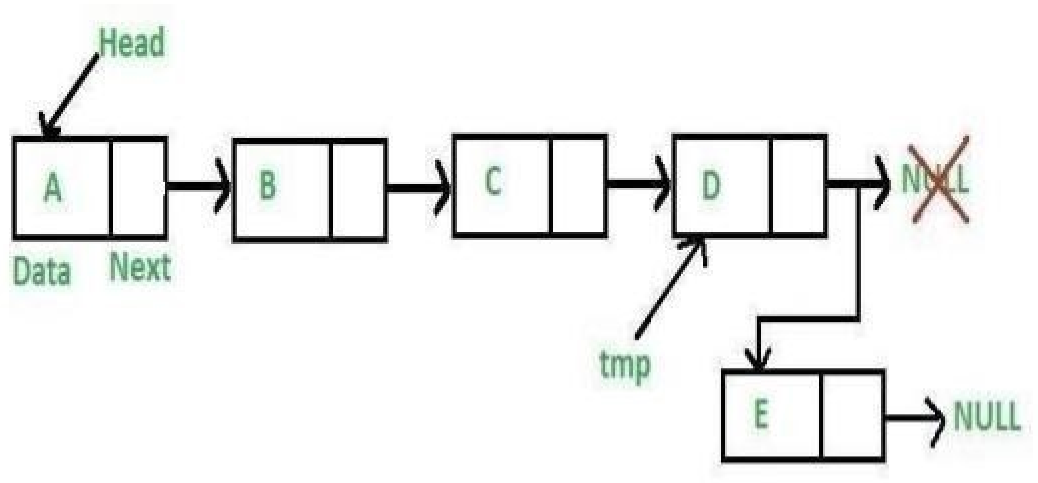


FIG 2.3.4

• Deletion

Deletion operations can be used to delete an item (or node) from linked list. A node may be deleted from:
1) Beginning of linked list
2) Specified location of the lined list
3) End of a linked list

• Search

Write a function that searches a given key ‘x’ in a given singly linked list. The function should return true If x is present in linked list and false otherwise. for
For example, if the key to be searched is 15 and linked list is 14→21→11→30→10, then function should Return false. If key to be searched is 14, then the function should return true . Initialize a node pointer,.
Current = head

Do following while current is not NULL
a) current->key is equal to the key being searched return true.
b) current = current->next

Return false

• Traversing and Displaying the elements in the list:

Traversing is the process of going through all the nodes from one end to the other end in linked list. In single linked list only,forward traversal is possible in double linked list both forward and reverse traversal is possible.

CHAPTER 3

DESIGN

3.1 DESIGN GOALS

3.1.1 The system should be capable of performing following functions:

- Store some of the basic information regarding transactions.
- Store data information of customer such as buying, tickets before tax, tax Percentage, total amount of tax paid, ticket price after task after tax, transactions on monthly basis.
- After calculating the transactions, the details and the details of the customer should be stored in the files.
- If we enter the details of the new customer the details of the customer and the transactions should be appended in the file.

3.2 ALGORITHM

- Start
- Get the input the customer details like name, id etc.
- Get the input of the number of tickets brought by the customer.
- Calculate the method of payment for the customer is required for calculating the total price.
- Get the type of payment.
- Calculate the total amount accordingly the method of payment to the customer.
- Calculate the total amount on the basis of numbers of tickets brought by the customer.
- Compute the overtime and paid extra amount for customer.
- Evaluate the Taxation of the ticket price according to the type of the payment.
- Evaluate the total amount after the customer's deduction amount is subtracted from the total amount calculated with respect to total amount and the type of payment.
- Display the ticket price to the user along with the movie details which has be inputted by the user and organization.
- Store the details of customer in files along with the final amount after all the amount is calculated • End

CHAPTER 4

Module 1

```
  void main()
 {
  booking();
         int choose;   do{
   printf("\n\n++++++++++++++++++++++++++++++++++++++++++++++++++++++
  +++++++++");
        printf("\ n\     n");
        printf("\t\t\n\n        FILM TICKET BOOKING SYSTEM ");
        printf("\n\n");
                         printf("\n\n
    ++++++++++++++++++++++++++++++++++++++++++ +");
        printf("\n\n Enter (1) TO ADD A FILM");
        printf("\n\n Enter (2) TO View All FILMS");
        printf("\n\n Enter (3) TO Find/search A FILM ");
        printf("\n\n Enter (4) TO Booking FILM Ticket ");
        printf("\n\n Enter (5) TO VIEW All TICKET Transactions");
       printf("\n\n Enter (0) TO quit ");

       printf("\nplease,Enter your Choice ::");
        scanf("%d",&choose);       system("cls");

       switch (choose)

{
case 1 :
                add_a_film();
        break;
case   2:
                view_all_films();
        break;
case 3:
```

```
                    find_film();
            break;
case4book_film_ticket()
;

                   break;



                   case 5:

            ticket_transactions();
            break;          case 0:
      exit(0);
break;

                    default:
                     printf("ENTERED CHOOSE DOESNOT EXIST!!!");
break;
           }
  }while(choose!=0);
  }
```

Module 2

```
void add_a_film()
 {
          FILE *fp;
          struct book y;
         printf("\nEnter digital movie code :- ");
 scanf("%s",y.code);
printf("\   nEnter   film   name   :-   ");
scanf("%s",y.name);
         printf("\nEnter film Release Date:- ");
scanf("%s",y.date);
         printf("\nEnter film Ticket cost:- ");

scanf("%d",&y.price);
fp=fopen("data.txt","x");
          if(fp == NULL)
         {



                   printf("file doesnot exist");
         }
         else
         {

 fprintf(fp,"%s %s %s %d \n",y.code,y.name,y.date,y.price);
 printf("saved Successfully in film transactions");
        };
  printf("\n\n");
      fclose(fp);  system("cls");}
```

Module 3

```
void find_film()
{
         struct book y;
FILE *fp;
 char choose[20];
  printf("Enter film code :");

 scanf("%s",choose);
 fp = fopen("data.txt","r");
                              if(fp == NULL)
          {
                     printf("file doesnot exist !");
           exit(1);
          }
          else
          {
                   while(getc(fp) != EOF)
                   {
                              fscanf(fp,"%s %s %s %d",y.code,y.name,y.date,&y.price);
                    if(strcmp(y.code,choose) == 0)
                            {
```

```
                                        printf("\n\n\n record found successfully \n\n");
            printf("\n\t\t\t film code ::%s",y.code);
    printf("\n\t\t\t film name ::%s",y.name);
    printf("\n\t\t\t film date ::%s",y.date);
                                        printf("\n\t\t\t cost of film ticket ::%d",y.price);
                            };
                    };
            };


    fclose(fp);
    system("cls"); };
```

Module 4

```
void view_all_films()
{
 char choose;  FILE *fp;
                fp = fopen("data.txt","r");
             if(fp == NULL)
             {
                          printf("sorry,file does not exist !");
 exit(1);

             }
             else
             {
                          system("cls");
                             while( ( choose = fgetc(fp) ) != EOF )
       printf("%c",choose);
             };
              fclose(fp);
};
```

Module 5

```
void book_film_ticket()
{
   struct book y;
           FILE *fp;
FILE *ufp;
               int total_seat,mobile,total_amount;
           char name[20];
char  choose;
char film_code[20];
 fp = fopen("data.txt","r");
if(fp == NULL)
           {
                         printf("file does not found !");

exit(1);
           }
            else
            {
                       system("cl");


  while( ( choose = fgetc(fp) ) != EOF )

  printf("%c",choose);
             };
              fclose(fp);
                    printf("\n For Book ticket, Choice your favorite film\n\n");
   printf("\n Enter film code :");
    scanf("%s",film_code);
  fp = fopen("data.txt","r");
              if(fp == NULL)
             {
                        printf("file doesnot exist!\n");
  exit(1);
             }
             else
             {
                    while(getc(fp) != EOF)
                  {
                                fscanf(fp,"%s %s %s %d",y.code,y.name,y.date,&y.price);
                              if(strcmp(y.code,film_code) == 0)
                          {
                                    printf("\n record found successfully \n\n");
 printf("\n\t\t\t film code ::%s",y.code);
printf("\n\t\t\t film name ::%s",y.name);
printf("\n\t\t\t film release date::%s",y.date);
                                       printf("\n\t\t\t cost of film ticket::%d",y.price);
                          }
                  }
          }
          printf("\n\n/////  Fill your personal details /////");
printf("\n\n\n enter your full name :");
scanf("%s",name);
             printf("\n\n enter your phone Number :");
           scanf("%d",&mobile);
         printf("\n enter total number Of film Tickets you want to book\n :");
scanf("%d",&total_seat);
total_amount = y.cost * total_seat;


printf("\n\n\n");
printf("\n\n  *****  ENJOY  YOUR  MOVIE  ***** \n");
          printf("\n\t\t\n  name : %s",name);
          printf("\n\t\t\n your mobile number : %d",mobile);
         printf("\n\t\t\n film name : %s",y.name);
                 printf("\n\t\t\t\n total no. of
seats you booked: %d",total_seat);
 printf("\n\t\t\t\n price per ticket : %d",y.price);




printf("\n\t\t\t\n total amount of tickets : %d",total_amount);
ufp=fopen("ticket_transactions.txt","x");
          if(ufp == NULL)
         {
                   printf("file does not exist\n");
         }
```

```
else      {
            fprintf(ufp,"%s %d %d %d %s %d \n",name,mobile,total_seat,total_amount,y.name,y.price);

                    printf("\n++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++");
                   printf("\n   Record inserted Successfully to the transactions file");
     printf("\n++++++++++++++++++++++++++++++++++++++++++++++++++++++++\n");
           }
   printf("\n\n");
            fclose(ufp);
            fclose(fp);
}
```

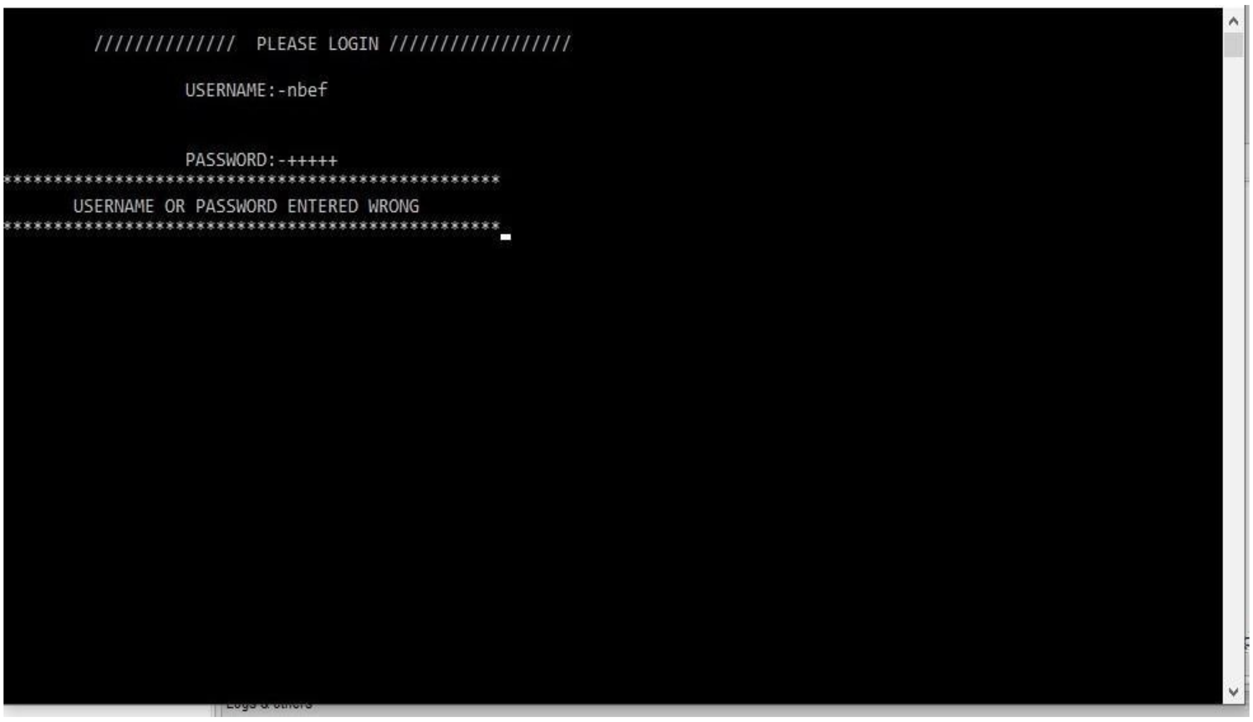
Module 6

```
void ticket_transactions()
{
 char choose;  FILE *fp;
                fp = fopen("ticket_Transactions.txt","r");
             if(fp == NULL)
            {
                        printf("file doesnot exist!!!\n\n");
                        exit(1);
            }
            else
            {           system("cls");

while( ( choose = fgetc(fp) ) != EOF )


printf("%c",choose);


            };
            fclose(fp);
};
```

CHAPTER 5

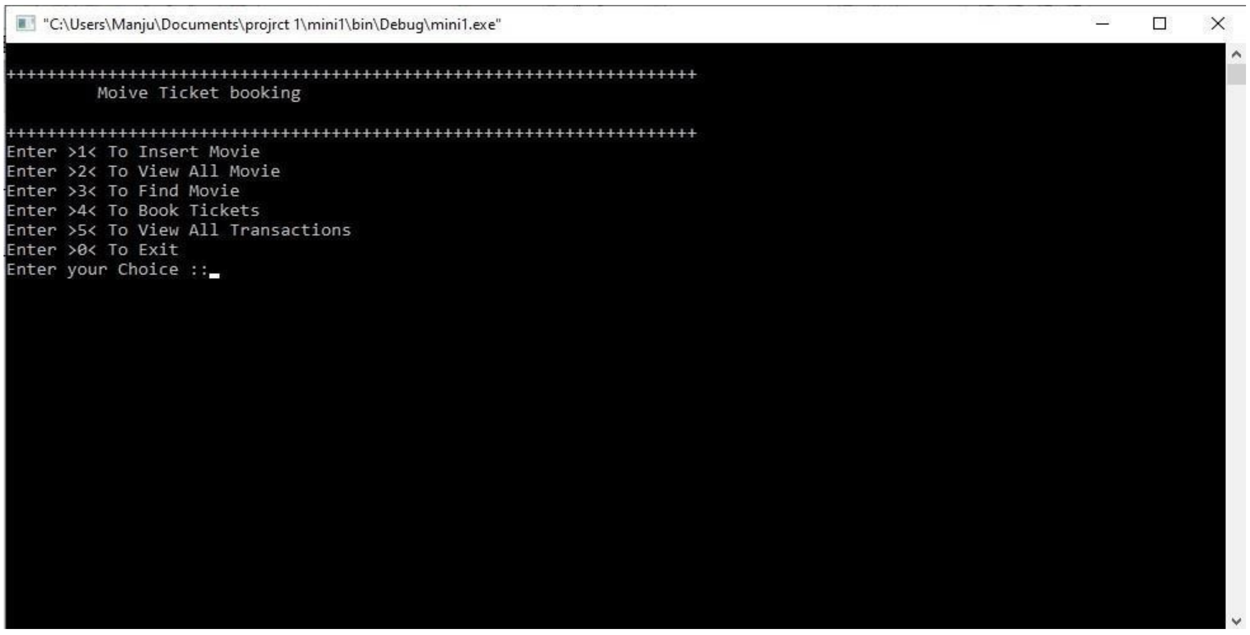
RESULTS

5.1 SAMPLE OUTPUT 1



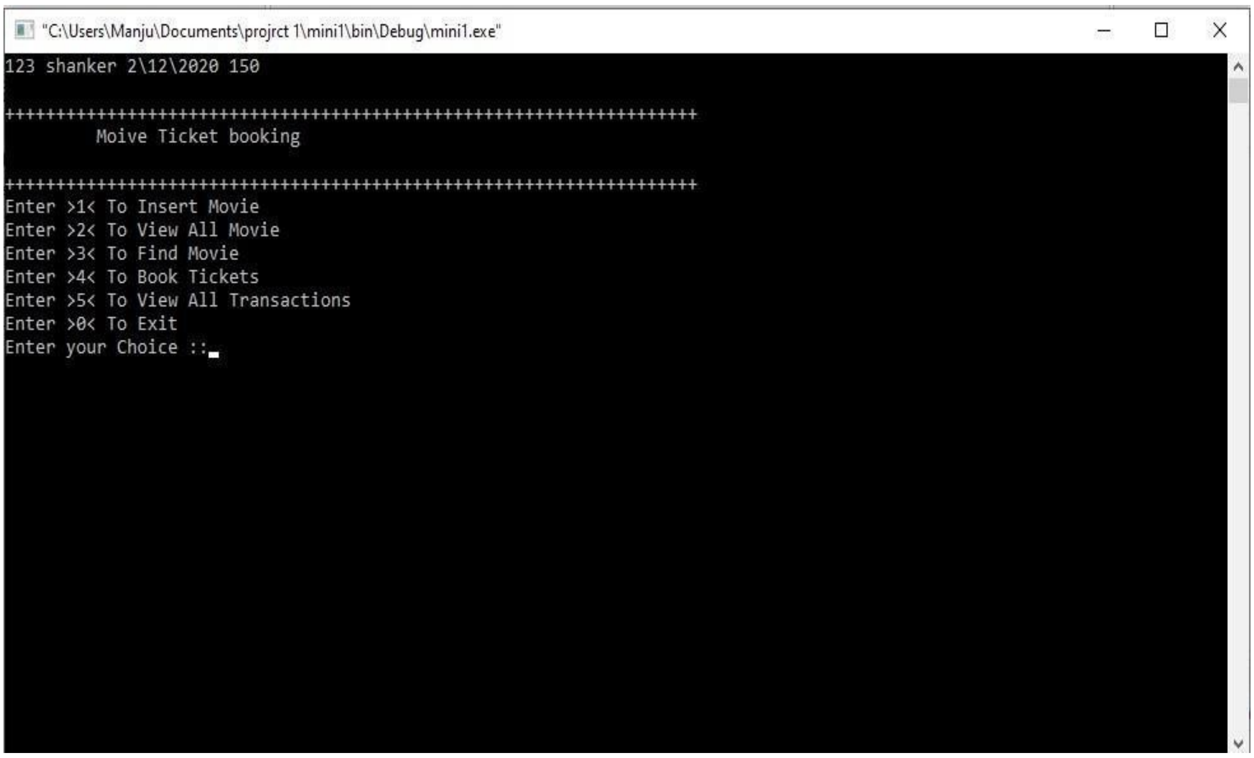
```
         //////////////  PLEASE LOGIN //////////////////

                  USERNAME:-nbef


                  PASSWORD:-+++++
*************************************************
       USERNAME OR PASSWORD ENTERED WRONG
*************************************************
```

5.2 SAMPLE OUTPUT 2



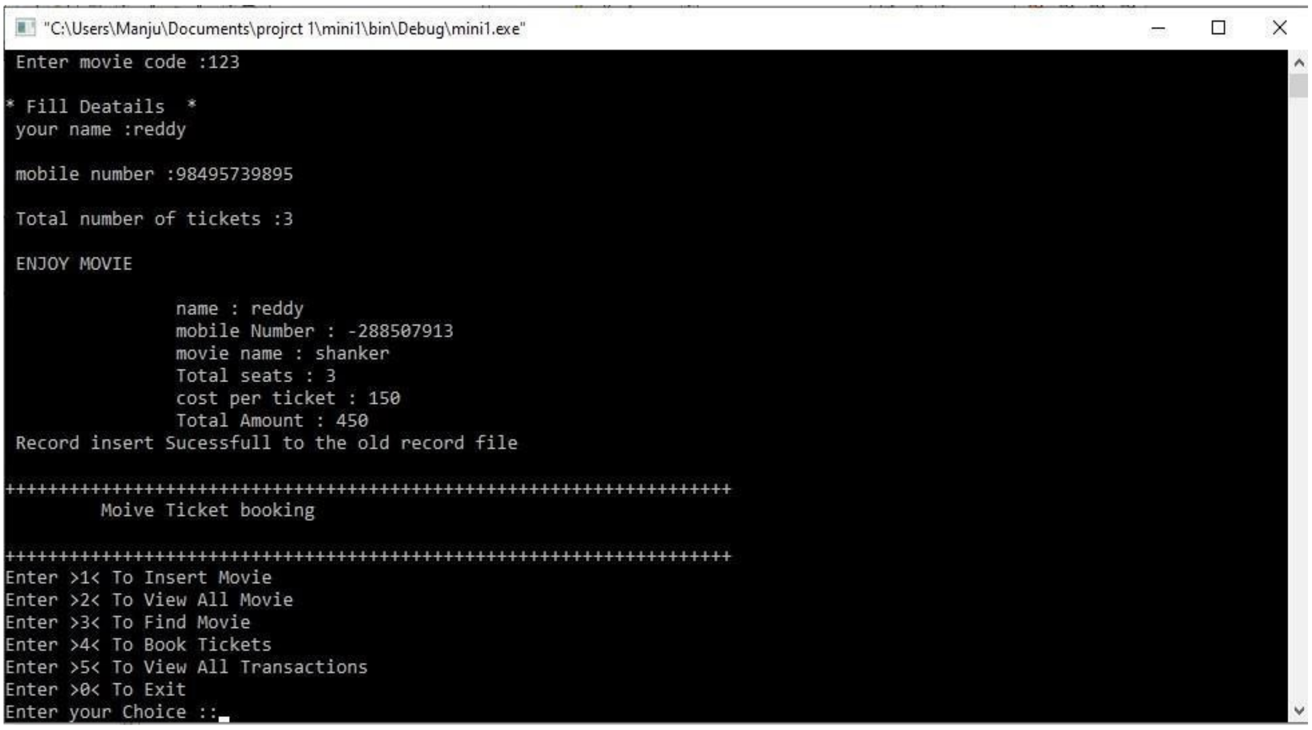
```
"C:\Users\Manju\Documents\projrct 1\mini1\bin\Debug\mini1.exe"

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
         Moive Ticket booking

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Enter >1< To Insert Movie
Enter >2< To View All Movie
Enter >3< To Find Movie
Enter >4< To Book Tickets
Enter >5< To View All Transactions
Enter >0< To Exit
Enter your Choice ::
```

5.3 SAMPLE OUTPUT 3



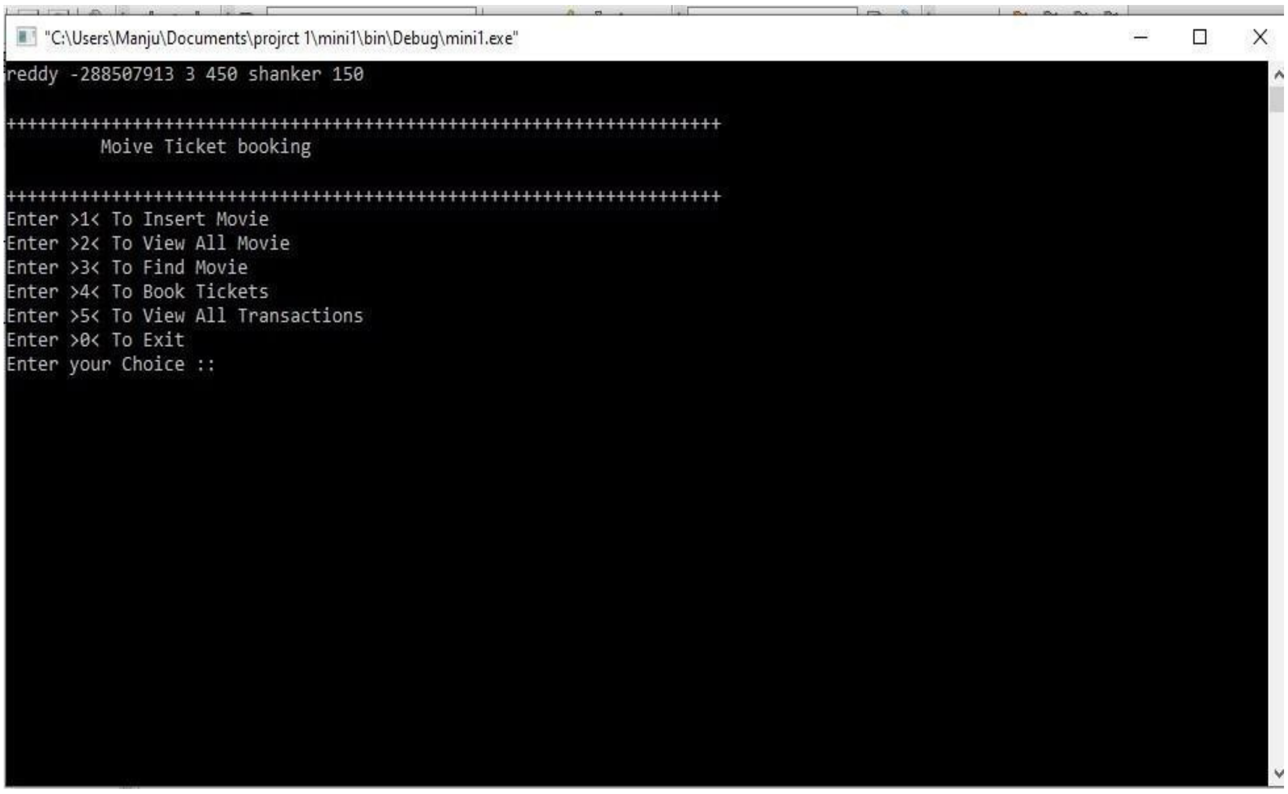
```
"C:\Users\Manju\Documents\projrct 1\mini1\bin\Debug\mini1.exe"
123 shanker 2\12\2020 150

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
         Moive Ticket booking

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Enter >1< To Insert Movie
Enter >2< To View All Movie
Enter >3< To Find Movie
Enter >4< To Book Tickets
Enter >5< To View All Transactions
Enter >0< To Exit
Enter your Choice ::
```

5.4 SAMPLE OUTPUT 4



```
"C:\Users\Manju\Documents\projrct 1\mini1\bin\Debug\mini1.exe"
 Enter movie code :123

* Fill Deatails  *
 your name :reddy

 mobile number :98495739895

 Total number of tickets :3

 ENJOY MOVIE

               name : reddy
               mobile Number : -288507913
               movie name : shanker
               Total seats : 3
               cost per ticket : 150
               Total Amount : 450
 Record insert Sucessfull to the old record file

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
         Moive Ticket booking

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Enter >1< To Insert Movie
Enter >2< To View All Movie
Enter >3< To Find Movie
Enter >4< To Book Tickets
Enter >5< To View All Transactions
Enter >0< To Exit
Enter your Choice ::
```

5.5 SAMPLE OUTPUT 5



```
"C:\Users\Manju\Documents\projrct 1\mini1\bin\Debug\mini1.exe"
reddy -288507913 3 450 shanker 150

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
         Moive Ticket booking

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Enter >1< To Insert Movie
Enter >2< To View All Movie
Enter >3< To Find Movie
Enter >4< To Book Tickets
Enter >5< To View All Transactions
Enter >0< To Exit
Enter your Choice ::
```

CONCLUSION

In the earlier days the people have to calculate the ticket price and seats manually but now this ticket calculator application will help in calculating the price on daily, weekly, and monthly basis. It is a Paperless mode of calculation which is also environmentally friendly . After calculating the price, the details of the customer and their final payout will be store in different files based on their type of payment and will be preserved permanently. If we want to store new customer details, we could also append the file using this application.

REFERENCES

1. Design a movie ticket booking system like Bookmyshow - GeeksforGeeks

2. Heap overflow and Stack overflow in C (tutorialspoint.com)

3. https://123projectlab.com/21-online-movie-ticket-booking-system/

4. Free Download Movie Ticket Booking Project in C/C++ with Source Code And Database c With Document - kashipara

5. Movie Ticket Booking in C/C++ With Source Code - ProjectNotes