**A MINI PROJECT ON**

**MOVIE TICKET BOOKING SYSTEM**

**Batch Members details:**

**Batch- 8 , 9 and 10**

|  |  |
| --- | --- |
| 17121A1222 | G Hemanth |
| 17121A1223 | G Vennila |
| 17121A1224 | G Rajasekhar |
| 17121A1225 | G Raveena |
| 17121A1226 | G Vanitha |
| 17121A1227 | G Jaswanth Tej |
| 17121A1231 | G T Gnana Prasanna |
| 17121A1232 | G Karthik |
| 17121A1233 | G Bharath |

**Department of Information Technology**

**SREE VIDYANIKETHAN ENGINEERING COLLEGE**

**(AUTONOMOUS)**

**(AFFILIATED TO J.N.T.U.A., ANANTAPUR)**

**SREE SAINATH NAGAR, TIRUPATHI – 517 102**



**CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Topic** | **Page Numbers** |
|  | Abstract |  |
| 1 | Introduction | 1 |
| 2 | Modules | 2-3 |
| 3 | Source code | 4-16 |
| 4 | Results | 17-19 |
| 5 | Design | 20 |
| 6 | Conclusion | 21 |
|  | References |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**ABSTRACT**

This project is made for providing the customer an anytime and anywhere service for booking the tickets in the cinema hall. And the user can easily be able to know about the movies released and then make their choice.

This movie ticket booking system is the process by which customer can purchase their movie tickets directly .Movie ticket booking system is measurable, cost effective and has very good user interface . Movie ticket booking system is useful for both promoters and customers equally. Customers can make online ticket booking at any instance of 24 hours a day. Customer can buy tickets without wasting their time . This project books the movie tickets for the user and displays the total cost.This project stores the records of customers who booked the tickets in a file which can be viewed by admin .

**User -**Any user who has registered with the system with valid user id and password can use the facility of booking the tickets online.

**Admin-**Admin provides the information about the presently running movies and has a provision to view the details of user.

**1. INTRODUCTION**

Now a days most of us use this facility of booking the tickets online. Book my show , Grab, Cinepolis are some of the apps used for booking movie tickets. Booking the tickets online is easier than the conventional booking at the theaters. The only thing you require is the good network connection and a basic knowledge of booking the tickets.

Coming to this project, User and Admin play their individual roles. The role of the Admin is to insert the data about new films. In order to book thetickets, first the user should login into the page with his details like username and password. Users are then able to see the list of films available**.** And they can choose a film based on their choice .On choosing the film ,users can book the number of tickets required on their choice.

**Advantages of the project**:

 Saves time of user.

 Fast and accurate.

 Provide better service to the customer.

**Applications of the project:**

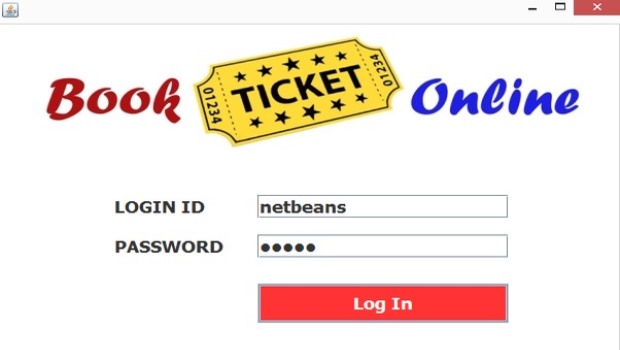
Some important goals of movie ticket booking system are as follows

 Movie Ticket Booking System project is aimed to provides facility to book cinema tickets anytime and from anywhere.

 One objective of this project is to minimize the number of staff at the ticket-box window.

 Provide a 24×7 service to the customer

**2.MODULES**

**User Modules**

1. Viewall movies

2. Find a movie

[[1]](#endnote-2)3. Book tickets

**Admin modules**

1. Insert a movie

2. old records

Fig 4.1 User login screen

**User Modules**

In order to perform user modules, first the user should login with his valid username and password. This is done using the function userlogin ().If the user fails to login for more than three times then the program terminates. After the successful login of user , he/she can be able to see

the following options.

**View all movies**

It displays the details of all the films which are stored in a file ie.., which are being inserted by admin. Void view() is the function used to perform this task.

**Find**

In order to find a particular movie the user is asked to enter the movie code and based on the code entered the movie is searched in a file in which the data of all the movies is stored.And this displays the details of the movie required by the user.

**Book Tickets**

On selecting a particular film,it displays the details of the film and asks the user to enter his details like name , mobile number etc and number of tickets required .Finally it displays the total cost of the tickets being booked and stores these details in a file of old records.

**Admin Modules**

In order to perform Admin modules , the Administrator should login into the system on entering his username and password correctly. This is done using the function

void admin\_login ( ).

**Insert movie**

A structure is declared with structure members code,movie name,date and cost.The data regarding a film is read using a variable of the structure and stored in a file.This process is done using the function void insert().

**Old records**

The transactions being made by the user can be viewed by the admin as they are stored in a file ‘old records’ after the ticket is being booked by the user.And this is implemented in the function void old\_record( ).

**Hardware and software requirements**

**Hardware**

RAM of minimum 1 GB

Hard disk of minimum 10 GB

**Software**

Operating system : Linux

Language used: c

**3. Source code**

#include<unistd.h>

#include<string.h>

#include<stdio.h>

#include<stdlib.h>

void adminmenu();

void menuuser();

void mainmenu();

void userlogin()

{

int a=0,i=0;

char uname[10],c=' ';

char code[10];

char \*password;

char user[10]="john";

char pass[10]="pass";

do

{

printf("\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* LOGIN FIRST \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ");

printf(" \n\n ENTER USERNAME:-");

scanf("%s", uname);

password=getpass("\n ENTER PASSWORD:- ");

if(strcmp(password,"pass")==0)

{

printf(" \n\n\n LOGIN IS SUCCESSFUL!!,You can proceede");

printf("\n\n\n\t\t\t\tPress any key to continue...");

menuuser();

break;

}

else

{

printf("\n SORRY !!!! LOGIN IS UNSUCESSFUL");

a++;

}

}while(a<=2);

if (a>2)

{

printf("\nSorry you have entered the wrong username and password for three times!!!");

}

system("clear");

exit(0);

}

void adminlogin()

{

int a=0,i=0;

char uname[10],c=' ';

char code[10];

char \*password;

char user[10]="admin";

char pass[10]="booking";

do

{

printf("\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* LOGIN FIRST \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ");

printf(" \n\n ENTER USERNAME:-");

scanf("%s", uname);

password=getpass("\n ENTER PASSWORD:- ");

if(strcmp(uname,"admin")==0 && strcmp(password,"booking")==0)

{

printf(" \n\n\n LOGIN IS SUCCESSFUL!!,You can proceede");

printf("\n\n\n\t\t\t\tPress any key to continue...");

adminmenu();

break;

}

else

{

printf("\n SORRY !!!! LOGIN IS UNSUCESSFUL");

a++;

}

}while(a<=2);

if (a>2)

{

printf("\nSorry you have entered the wrong username and password for three times!!!");

}

system("clear");

exit(0);

}

void mainmenu()

{

int op;

printf("\n\n \*\*\*\*\*\*\*\*\*\*\*\*WELCOME TO MOVIE TICKET BOOKING SYSTEM\*\*\*\*\*\*\*\*\*\*\*\*\*\* \n\n");

printf("Enter your Choice\n1.User\t\t2.Admin\t\t3.Exit\n");

scanf("%d",&op);

if(op==1)

userlogin();

else if(op==2)

adminlogin();

else if(op==3)

exit(0);

else

{

printf("Invalid option");

mainmenu();

}

}

struct book

{

char code[20];

char name[20];

char date[20];

int cost;

}b;

int seat = 60 ;

void insert();

void viewall();

void find();

void book\_ticket();

void old\_records();

void main()

{

mainmenu();

}

void menuuser()

{

int ch;

do{

printf("\n++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++");

printf("\n");

printf("\t Moive Ticket booking ");

printf("\n");

printf("\n++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++");

printf("\nEnter <1> To View All Movies");

printf("\nEnter <2> To Find Movie ");

printf("\nEnter <3> To Book Tickets");

printf("\nEnter <0> To Exit ");

printf("\nEnter your Choice ::");

scanf("%d",&ch);

system("clear");

switch (ch)

{

case 1:

viewall();

break;

case 2:

find();

break;

case 3:

book\_ticket();

break;

case 0:

mainmenu();

break;default:

printf("Wrong choice !");

break;

}

}while(ch!=0);

mainmenu();

}

void adminmenu()

{

int ch;

do{

printf("\n++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++");

printf("\n");

printf("\t Moive Ticket booking ");

printf("\n");

printf("\n++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++");

printf("\nEnter >1< To Insert Movie");

printf("\nEnter >2< To View All Transactions");

printf("\nEnter >0< To Exit ");

printf("\nEnter your Choice ::");

scanf("%d",&ch);

system("clear");

switch (ch)

{

case 1 :

insert();

break;

case 2:

old\_records();

break;

case 0:

mainmenu();

break;

default:

printf("Wrong choice !");

break;

}

}while(ch!=0);

mainmenu();

}

void insert()

{

FILE \*fp;

struct book b;

printf("Enter movie code :- ");

scanf("%s",b.code);

printf("Enter name :- ");

scanf("%s",b.name);

printf("Enter Release Date:- ");

scanf("%s",b.date);

printf("Enetr Ticket Price:- ");

scanf("%d",&b.cost);

fp=fopen("data.txt","a");

fprintf(fp,"%s %s %s %d \n",b.code,b.name,b.date,b.cost);

printf("Recorded Successfully");

printf("\n");

fclose(fp);

system("clear");

}

void find()

{

struct book b;

FILE \*fp;

char ch[20];

printf("Enter movie code :");

scanf("%s",ch);

fp = fopen("data.txt","r");

while(fscanf(fp,"%s %s %s %d",b.code,b.name,b.date,&b.cost)!= EOF)

{

if(strcmp(b.code,ch) == 0)

{

printf("\n Record Found\n");

printf("\n\t\tCode ::%s",b.code);

printf("\n\t\tmovie name ::%s",b.name);

printf("\n\t\tmovie date ::%s",b.date);

printf("\n\t\tprice of ticket ::%d",b.cost);

}

}

fclose(fp);

}

void viewall()

{

char ch;

FILE \*fp;

fp = fopen("data.txt","r");

if(fp == NULL)

{

printf("file does not found !");

menuuser();

}

else

{

system("clear");

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

printf("%c",ch);

}

fclose(fp);

}

void book\_ticket()

{

struct book b;

FILE \*fp;

FILE \*ufp;

int total\_seat,mobile,total\_amount;

char name[20];

char ch;

char movie\_code[20];

fp = fopen("data.txt","r");

if(fp == NULL)

{

printf("file does not found !");

userlogin();

}

else

{

system("clear");

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

printf("%c",ch);

}

fclose(fp);

int c2=0;

printf("\n For Book ticket Choice Movie(Enter Movie Code First Latter Of Movie)\n");

printf("\n Enter movie code :");

scanf("%s",movie\_code);

fp = fopen("data.txt","r");

fseek(fp,0,SEEK\_SET);

if(fp == NULL)

{

printf("file does not found !");

exit(1);

}

else

{

while( fscanf(fp,"%s %s %s %d",b.code,b.name,b.date,&b.cost) != EOF)

{

if(strcmp(b.code,movie\_code) == 0)

{

printf("\n Record Found\n");

printf("\n\t\tCode ::%s",b.code);

printf("\n\t\tMovie name ::%s",b.name);

printf("\n\t\tdate name ::%s",b.date);

printf("\n\t\tPrice of ticket::%d",b.cost);

c2=1;

}

if(c2==1)

break;

}

}

printf("\n\* Fill Deatails \*");

printf("\n your name :");

scanf("%s",name);

printf("\n mobile number :");

scanf("%d",&mobile);

printf("\n Total number of tickets :");

scanf("%d",&total\_seat);

total\_amount = b.cost \* total\_seat;

printf("\n ENJOY MOVIE \n");

printf("\n\t\tname : %s",name);

printf("\n\t\tmobile Number : %d",mobile);

printf("\n\t\tmovie name : %s",b.name);

printf("\n\t\tTotal seats : %d",total\_seat);

printf("\n\t\tcost per ticket : %d",b.cost);

printf("\n\t\tTotal Amount : %d",total\_amount);

ufp=fopen("oldrecords.txt","a");

if(ufp == NULL)

{

printf("FIle not Found");

}

else

{

fprintf(ufp,"%s%d%d%d%s%dn",name,mobile,total\_seat,total\_amount,b.name,b.cost);

printf("\n Tickets booked successfully....\n");

}

printf("\n");

fclose(ufp);

fclose(fp);

}

void old\_records()

{

char ch;

FILE \*fp;

system("clear");

fp = fopen("oldrecords.txt","r");

if(fp == NULL)

{

printf("file does not found !");

adminmenu();

}

else

{

system("clear");

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

printf("%c",ch);

}

fclose(fp);

}

4. RESULTS

4.1 User modules

In order to perform user modules , first the user should login with his valid username and password. On his successful login it displays the following options as shown in the Fig 4.1.1.



Fig 4.1.1 User login screen

**4.1.1 View all-** This gives the list of all the films inserted by the admin as shown in the fig 4.2



Fig 4.1.2 view all movies

**4.1.2 Find**

This asks the user to enter the movie code and gives the details of the movie .



Fig 4.1.3 Finding a movie

**4.1.3 Book tickets**

On selecting the desired movie, it asks the user to enter his/her details, no. of tickets required and displays the total cost of the tickets .

**4.2 Admin modules**

In order to perform admin modules, the admin should login with his username and password. If he/she enters wrong password or username for more than three times then the program terminates.



Fig 4.2.1 Admin login scrren

**4.2.1 Insert Movie**

This asks the Admin to enter the details regarding a movie and stores them in a file.



Fig 4.2.2 Insert Movie

**4.2.2 Old records**

This makes the admin to view the details of the users who has booked the tickets.



Fig 4.2.3 Old records

**5. Design of the project**

****

**6. CONCLUSION**

Finally to say, this project provides an easy environment to book the movie tickets. It minimize the number of staff at ticket counters .It provides a 24\*7 service to the customer. An user friendly interface for the admin to add the movie details. All the manual work is computerized .This helps the user to know the immediate information about the running movies. Some features can be added like to allow the users to comment on the movies, to give out the list of upcoming movies.

**References**

Neil Matthew and Richard stones “Beginning Linux Programming”

Byron Gottfried and Jithender Kumar c PROGRAMMING WITH C, Third Edition

**Title** : Movie Ticket Booking System

**Link**: https://code-projects.org

**Drafted on**: 08.09.18

1. [↑](#endnote-ref-2)