

**Movie Ticket Booking System.**

Team Member: -

17BCE0032 Diksha Singhania.

**Report Submitted for the**

**Final Project Review.**

**Course Code: CSE2003-Data Structures and Algorithm**

SLOT: G1

Professor: Sendhil Kumar K.S.

**Aim:**

 The main aim of this Movie World is online booking tickets for various movies in

Registered theatres by the users.

**Problem statement:**

Here details about Now playing movies and no of tickets available in aparticular theatre for particular movie, Movie Details, ticket cost for registered users

**Abstract:**

In the Movie Ticket booking system, I have tried to develop a platform to make the booking system more efficient, easier and fast. I will be developing the project using C language. It is the most commonly used language in the Procedure oriented programming languages. I will be defining various functions and structures for better implementation.

Movie ticket booking systems like book my show enables us to do the booking which is usually done at the ticket counter, that is the intended users can directly select from the various choices provided. The final output which comes to the user is movie ticket with the amount to be paid. Finally, it’s the user who has to decide whether to book the ticket or not.

This system is used for booking the ticket, searching for the movies, checking the availability of the seats and for the cancellation of the ticket.

This ticket booking system is beneficial to all the user who have to book the ticket online instead of standing in the long queues at the ticket counters. This system can also be used by the authorities for doing the bookings.

**Modules:**

The above system will be able to perform the following functions: -

First there will be a login for admin or user:

* **In admin:**

1. Book ticket

2. Display Booked Ticket

3. Movie Database

a. Add movies

b. sort movies

c. delete movies

d. search movies

e. modify movie timings

g. display movie information

4. Cancel ticket

5. Profile Setting

6. Exit

* **In user:**

1. Book tickets

2. Display Booked Tickets

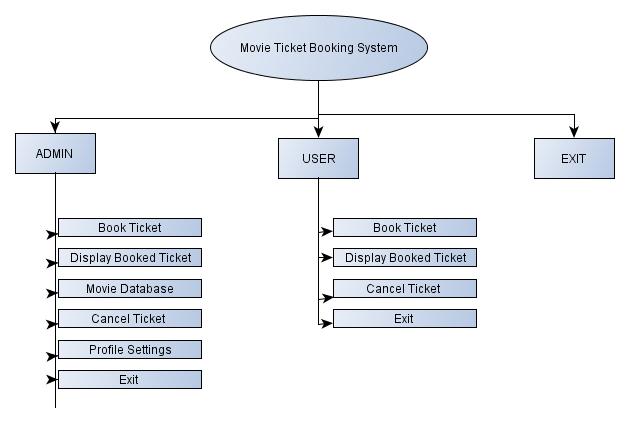
3. Cancel Ticket

4.Payment

5.Rating

. Exit

I will try to implement the above using **linked lists** and will try to implement **sorting**, and **searching** the database. I will use **text files** to save the data.

**Diagram:**

Header files used:

#include<malloc.h>

#include<string.h>

#include<stdlib.h>

#include<stdio.h>

**Functions Used:**

void Create\_new\_account();

void Cash\_Deposit();

void Cash\_withdrawl();

void Account\_information();

void Log\_out();

void display\_options();

void insert\_details();//for inserting movie details

void viewAll(); // for view all data

void find(); // for finding data

void book\_ticket(); //for booking tickets

void old\_record();

void admin();

void readadmin();

void writeadmin();

void adminprofile();

void loggedadmin();

int modifyadmin();

int deleteadmin();

void displayadmin();

void add\_rating();

**Data files used:**

Admin.txt

User.txt

Oldtransaction.txt

Data.txt

**Hardware and Software requirements:**

**Hardware:**

1.Any computer system or laptop

**Software requirement:**

1.Code blocks

2.gcc compiler

3.windows operating system

**Source code:**

#include<malloc.h>

#include<string.h>

#include<stdlib.h>

#include<stdio.h>

struct book

{

char code[20];

char name[20];

char date[20];

int cost;

char rate[10];

}b;

struct user\_rating

{

char name1[20];

char rate1[10];

}ra;

struct acc\_type

{

char bank\_name[20];

char bank\_branch[20];

char acc\_holder\_name[30];

int acc\_number;

char acc\_holder\_address[100];

float available\_balance;

};

struct acc\_type account[20];

/\*

printf("The above structure can be declared using

typedef like below");

typedef struct acc\_type

{

char bank\_name[20];

char bank\_branch[20];

char acc\_holder\_name[30];

int acc\_number;

char acc\_holder\_address[100];

float available\_balance;

}Acc\_detail;

Acc\_detail account[20];

\*/

int num\_acc;

void Create\_new\_account();

void Cash\_Deposit();

void Cash\_withdrawl();

void Account\_information();

void Log\_out();

void display\_options();

int seat = 200;

void insert\_details();//for inserting movie details

void viewAll(); // for view all data

void find(); // for finding data

void book\_ticket(); //for booking tickets

void old\_record(); //for view old recorrds of users,booked tickets

struct node

{

char adname[50],adpass[50];

struct node \*next;

};

struct node \*head=NULL,\*headd=NULL,\*p,\*r,\*last=NULL,\*lastt=NULL,\*q=NULL,\*s=NULL;

void admin();

void readadmin();

void writeadmin();

void adminprofile();

void loggedadmin();

int modifyadmin();

int deleteadmin();

void displayadmin();

void add\_rating();

int found=0;

int ch,j,x,c=' ',choice1,choice2;

char str1[20],str2[20],str3[20],str4[20];

char str5[]="admin",str6[]="admin",str7[20],str8[20];

char choice3[]="y",choice4[10],choice5[10];

char ticket[20][3];

char adname[50],adpass[50];

char input[255];

int status1;

int ch;

int n,m,i,src;

char umbrella;

char ticket1[20][3];

int main()

{

system("color f1");

FILE \*infile;

do{

printf("\n---------------------------------------------------------------------------\n");

printf(" Welcome to Movie Ticket Booking System" );

printf("\n----------------------------------------------------------------------------\n");

printf("\tLog in as:\n");

printf(" \n\t\t 1. ADMIN \n\n\t\t 2. USER \n\n\t\t 0. EXIT");

printf("\n----------------------------------------------------------------------------\n");

printf(" ");

printf("\n----------------------------------------------------------------------------\n");

printf("\n\tChoice:\t\t");

scanf("\t%d",&ch);

switch(ch)

{

case 1:

printf("\n----------------------------------------------------------------------------\n");

printf(" Welcome ADMIN " );

printf("\n----------------------------------------------------------------------------\n");

readadmin();

r=headd;

i=0;

printf("\n Enter Name:\t");

scanf("%s",str1);

printf("\n Enter Password:\t");

while (i<=9)

{

str2[i]=getch();

if(str2[i]==13)

break;

else printf("\*");

i++;

}

str2[i]='\0';

for(i=1;r!=NULL;i++)

{

if((strcmp(r->adname,str1)==0)&&(strcmp(r->adpass,str2)==0))

{

strcpy(adname,r->adname);

strcpy(adpass,r->adpass);

found=1;

}

r=r->next;

}

if((found)==1 || ((strcmp(str5,str1)==0) &&( strcmp(str6,str2)==0) ))

{

do

{ system("CLS");

printf("\n---------------------------------------------\n");

printf(" Menu ");

printf("\n---------------------------------------------\n");

printf("\n\t1. Book ticket\n\t");

printf("2. View All Transactions\n\t3. Add movies");

printf("\n\t4. View movies\n\t5.Cancel ticket\n\t6. Profile Settings\n\t0. EXIT \n\n\t");

printf("Choice:\t");

scanf("%d",&choice1);

switch(choice1)

{

case 0: exit(0);

break;

case 1:

system("CLS");

printf("\n----------------------------------------------------------------------------\n");

printf(" Ticket Booking" );

printf("\n----------------------------------------------------------------------------\n");

book\_ticket();

break;

case 2:

system("CLS");

printf("\n----------------------------------------------------------------------------\n");

printf(" View All transactions" );

printf("\n----------------------------------------------------------------------------\n");

old\_record();

break;

case 3:

system("CLS");

printf("\n----------------------------------------------------------------------------\n");

printf(" Insert Movies" );

printf("\n----------------------------------------------------------------------------\n");

insert\_details();

break;

case 4:

system("CLS");

printf("\n----------------------------------------------------------------------------\n");

printf(" View Movies" );

printf("\n----------------------------------------------------------------------------\n");

viewAll();

break;

case 5:

system("CLS");

printf("Your ticket has been canceled");

break;

case 6: system("CLS");

adminprofile();

}

printf("\n----------------------------------------------------------------------------\n");

printf("\n\tWant to Continue[Main Menu] (y/n):\t");

scanf("%s",&choice4);

strlwr(choice4);

}while((strcmp(choice3,choice4))==0);

}else

{

system("CLS");

printf("\n\n\n\t!! INVALID CREDENTIALS ...... Enter Correct Username And Password !!\n\n");

}

break;

case 2:

{ system("CLS");

printf("\n---------------------------------------------\n");

printf(" USER Menu ");

printf("\n---------------------------------------------\n");

int t;

do{

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

printf("\n");

printf("\t Moive Ticket booking ");

printf("\n");

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

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

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

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

printf("\nEnter >4< To Insert Movie");

printf("\nEnter >5< To View All Transactions");

printf("\nEnter >6< To Make the Payment");

printf("\nEnter >7< Add user rating");

printf("\nEnter >8< View user rating");

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

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

scanf("%d",&t);

system("cls");

switch (t)

{

case 4 :

system("CLS");

printf("Your ticket has been canceled");

break;

case 1:

viewAll();

break;

case 2:

find();

break;

case 3:

book\_ticket();

break;

case 5:

old\_record();

break;

case 6:

{

char option;

char f2f[50] = "19800023";

num\_acc=0;

while(1)

{

printf("\n\*\*\*\*\* Welcome to Bank Application \*\*\*\*\*\n");

printf("\nThe account no where the money is to be transferred is %s",f2f);

display\_options();

printf("Please enter any options (1/2/3/4/5) ");

printf("to continue : ");

option = getch();

printf("%c \n", option);

switch(option)

{

case '1':

Create\_new\_account();

break;

case '2':

Cash\_withdrawl();

break;

case '3':

Account\_information();

break;

case '4':

return 0;

case '5':

system("cls");

break;

default : system("cls");

printf("Please enter one of the options");

printf("(1/2/3/4/5) to continue \n ");

break;

}

}

}

break;

case 7:

add\_rating();

break;

case 8:

view\_rating();

break;

case 0:

exit(0);

break;

default:

printf("Wrong choice !");

break;

}

}while(t!=0);

}

}

}while(ch!=0);

}

void adminprofile()

{ do{system("CLS");

printf("\n---------------------------------------------\n");

printf(" PROFILE SETTINGS ");

printf("\n---------------------------------------------\n");

printf("\n\t1. Add Admins\n\t");

printf("2. Display Admins and Passwords\n\t3. Change Username and Password");

printf("\n\t4. Delete Admin\n\t0. EXIT \n\n\t");

printf("Choice:\t");

scanf("%d",&choice1);

switch(choice1)

{

case 1: system("CLS");

loggedadmin();

admin();

break;

case 2: system("CLS");

loggedadmin();

displayadmin();

break;

case 3:system("CLS");

loggedadmin();

modifyadmin();

break;

case 4:system("CLS");

loggedadmin();

deleteadmin();

break;

case 0:system("CLS");

exit(0);

break;

}

printf("\n----------------------------------------------------------------------------\n");

printf("\n\t Want to Continue[Profile Setting] (y/n):\t");

scanf("%s",&choice4);

strlwr(choice4);

}while((strcmp(choice3,choice4))==0);

}

void admin()

{ int i=0;

readadmin();

char str1[20],str2[20];

printf("\n Enter New Admin Name:\t");

scanf("%s",str1);

printf("\n Enter New Admin Password:\t");

while (i<=9)

{

str2[i]=getch();

if(str2[i]==13)

break;

else printf("\*");

i++;

}

str2[i]='\0';

printf("\n Confirm Password:\t");

i=0;

while (i<=9)

{

str3[i]=getch();

if(str3[i]==13)

break;

else printf("\*");

i++;

}

str3[i]='\0';

printf("\n Enter LOGGED IN ADMIN PASSWORD:\t");

i=0;

while (i<=9)

{

str4[i]=getch();

if(str4[i]==13)

break;

else printf("\*");

i++;

}

str4[i]='\0';

if( (strcmp(str2,str3)==0) && ((strcmp(str4,adpass)==0) || (strcmp(str4,str6)==0)))

{

struct node \*r;

r=lastt;

if(lastt==NULL){

i=1;

lastt=(struct node\*)malloc(sizeof(struct node));

lastt->next=NULL;

strcpy(lastt->adname,str1);

strcpy(lastt->adpass,str2);

r=lastt;

headd=lastt;

}else

{

r=lastt;

r->next=(struct node\*)malloc(sizeof(struct node));

r=r->next;

lastt=r;

strcpy(r->adname,str1);

strcpy(r->adpass,str2);

r->next=NULL;

}

writeadmin();

printf("\n\t!! Admin Profile Created Successfully !!");

}else

{

printf("\n\t!! Oopps !! .... Password Not Matched .....");

}

}

void readadmin()

{

int i,filempty=0;

FILE \*py=fopen("Admin.txt","r");

r=headd;

fseek(py,0,SEEK\_END);

if(r==NULL)

{

r=(struct node\*)malloc(sizeof(struct node));

headd=r;

}

int len=(int)ftell(py);

if(len<=0)

{

filempty=1;

r=NULL;

headd=r;

}

if(filempty==0)

{

rewind(py);

while(fscanf(py,"%s\t%s\n",r->adname,r->adpass))

{

if(feof(py))

{

break;

}

r->next=(struct node\*)malloc(sizeof(struct node));

r=r->next;

lastt=r;

r->next=NULL;

}

}

//fclose(py);

}

void writeadmin()

{

FILE \*ppy=fopen("Admin.txt","w");

int i;

struct node \*tempp;

tempp=headd;

if(tempp==NULL)

{

printf("list is empty");

}

else

{

for(i=0;tempp!=NULL;i++)

{

fprintf(ppy,"%s\t%s\n",tempp->adname,tempp->adpass);

tempp=tempp->next;

}//for

}//else

fclose(ppy);

}

void loggedadmin()

{

printf("\n\n\n---------------------------------------------\n");

printf(" Logged In Admin:\t ");

printf("%s",adname);

printf("\n---------------------------------------------\n");

}

int modifyadmin()

{

struct node \*r,\*s;

readadmin();

int loc,i;

displayadmin();

printf("\nEnter the location to Modify:\t");

scanf("%d",&loc);

if(r!=NULL)

{

if(loc==1)

{

r=headd;

headd=headd->next;

writeadmin();

free(r);

return(headd);

}

else{s=headd;}

for(i=1;i<loc-1;i++)

{

s=s->next;

}

r=s->next;

s->next=r->next;

writeadmin();

free(r);

n=n-1;

}

else

{

printf("\nSorry,The list is empty");

}

readadmin();

r=headd;

s=headd;

char str1[20],str2[20];

printf("\n Enter New Admin Name:\t");

scanf("%s",str1);

printf("\n Enter New Admin Password:\t");

i=0;

while (i<=9)

{

str2[i]=getch();

if(str2[i]==13)

break;

else printf("\*");

i++;

}

str2[i]='\0';

printf("\n Confirm Password:\t");

i=0;

while (i<=9)

{

str3[i]=getch();

if(str3[i]==13)

break;

else printf("\*");

i++;

}

str3[i]='\0';

printf("\n Enter LOGGED IN ADMIN PASSWORD:\t");

i=0;

while (i<=9)

{

str4[i]=getch();

if(str4[i]==13)

break;

else printf("\*");

i++;

}

str4[i]='\0';

r=(struct node\*)malloc(sizeof(struct node));

strcpy(r->adname,str1);

strcpy(r->adpass,str2);

if(loc==1)

{

r->next=headd;

headd=r;

writeadmin();

return(r);

}

s=headd;

for(i=1;i<loc-1;i++)

{

if(s!=NULL)

{

s=s->next;

}

}

r->next=s->next;

s->next=r;

n=n+1;

writeadmin();

printf("\n\t!! Admin Profile Created Successfully !!");

return(headd);

}

void displayadmin()

{

readadmin();

r=headd;

if(r==NULL)

{

printf("list is empty");

}

else

{

for(i=0;r!=NULL;i++)

{

printf("\n\tAdmin Name:\t\t"); //source of travel

printf("%s",r->adname);

printf("\t\tPassword:\t\t"); //source of travel

printf("%s",r->adpass);

r=r->next;

}

}

}

int deleteadmin()

{

struct node \*r,\*s;

readadmin();

int loc,i;

displayadmin();

printf("\nEnter the location to Delete:\t");

scanf("%d",&loc);

if(r!=NULL)

{

if(loc==1)

{

r=headd;

headd=headd->next;

writeadmin();

free(r);

return(headd);

}

else{s=headd;}

for(i=1;i<loc-1;i++)

{

s=s->next;

}

r=s->next;

s->next=r->next;

writeadmin();

free(r);

n=n-1;

}

else

{

printf("\nSorry,The list is empty");

}

displayadmin();

}

void add\_rating()

{

FILE \*fp;

struct user\_rating r;

printf("enter movie name:-");

scanf("%s",r.name1);

printf("\nenter the rating:-");

scanf("%s",r.rate1);

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

if(fp == NULL)

{

printf("FIle not Found");

}

else

{

fprintf(fp,"%s %s \n",r.name1,r.rate1);

printf("Recorded Successfully");

}

printf("\n");

fclose(fp);

system("cls");

}

int view\_rating()

{

char ch;

FILE \*fp;

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

if(fp == NULL)

{

printf("file does not found !");

exit(1);

}

else

{

system("cls");

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

printf("%c",ch);

}

fclose(fp);

}

void insert\_details()

{

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);

printf("Enter rating:-");

scanf("%s",&b.rate);

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

if(fp == NULL)

{

printf("FIle not Found");

}

else

{

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

printf("Recorded Successfully");

}

printf("\n");

fclose(fp);

system("cls");

}

void find() //find details

{

struct book b;

FILE \*fp;

char ch[20];

printf("Enter movie code :");

scanf("%s",ch);

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

if(fp == NULL)

{

printf("file does not found !");

exit(1);

}

else

{

while(getc(fp)!= EOF)

{

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

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

{

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

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);

printf("\n\t\trating of movie ::%s",b.rate);

}

}

}

fclose(fp);

system("cls");

}

void viewAll()

{

char ch;

FILE \*fp;

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

if(fp == NULL)

{

printf("file does not found !");

exit(1);

}

else

{

system("cls");

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

printf("%c",ch);

}

fclose(fp);

}

//for ticket booking

void book\_ticket()

{

struct book b;

FILE \*fp;

FILE \*ufp;

int total\_seat,mobile,total\_amount;

char name[20];

char ch; //used in display all movies

char movie\_code[20]; //for searching

// disply all moives by default for movie code

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

if(fp == NULL)

{

printf("file does not found !");

exit(1);

}

else

{

system("cls");

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

printf("%c",ch);

}

fclose(fp);

//display ends

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

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

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

//system("clear");

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

if(fp == NULL)

{

printf("file does not found !");

exit(1);

}

else

{

while(getc(fp)!= EOF)

{

fscanf(fp,"%s %s %s %d %s",b.code,b.name,b.date,&b.cost,b.rate);

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

{

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

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);

printf("\n\t\trating of movie ::%s",b.rate);

}

}

}

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);

displayseats();

printf("select the seats\n");

for(i=1;i<=total\_seat+1;i++)

{

gets(ticket[i]);

}

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

int response,f;

printf("do you want to preorder snacks for the movie\n");

scanf("%d",&response);

if (response==1)

{

printf("choose from the combos available\n");

printf("1.popcorn only -rs 90\n 2.coke+popcorn -rs 150\n 3.samossa+coke- rs 90\n 4.exit");

scanf("%d",&f);

switch(f)

{

case 1:

printf("you will get the order during interval on your seat\n you need to pay:");

printf("%d",total\_amount+90);

total\_amount=total\_amount+90;

break;

case 2:

printf("you will get the order during interval on your seat\n you need to pay:");

printf("%d",total\_amount+150);

total\_amount=total\_amount+90;

break;

case 3:

printf("you will get the order during interval on your seat\n you need to pay:");

printf("%d",total\_amount+90);

total\_amount=total\_amount+90;

break;

}

}

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

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

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

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

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

printf("\n \t\tseats selected:");

for(i=1;i<=total\_seat+1;i++)

{

puts(ticket[i]);

}

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

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

if(ufp == NULL)

{

printf("FIle not Found");

}

else

{

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

printf("\n Record insert Sucessfull to the old record file");

}

printf("\n");

fclose(ufp);

fclose(fp);

}

//for view all user transections

void old\_record()

{

char ch;

FILE \*fp;

//system("clear");

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

if(fp == NULL)

{

printf("file does not found !");

exit(1);

}

else

{

system("cls");

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

printf("%c",ch);

}

fclose(fp);

}

int displayseats()

{

printf("\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Platinum\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("A(1-20)--> [] [] [] [] [] [] [] [] [] \t[] [] [] [] [] [] [] [] [] [] []\n");

printf("B(1-20)--> [] [] [] [] [] [] [] [] [] \t[] [] [] [] [] [] [] [] [] [] []\n");

printf("\n");

printf("\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Gold\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("C(1-20)--> [] [] [] [] [] [] [] [] [] \t[] [] [] [] [] [] [] [] [] [] []\n");

printf("D(1-20)--> [] [] [] [] [] [] [] [] [] \t[] [] [] [] [] [] [] [] [] [] []\n");

printf("E(1-20)--> [] [] [] [] [] [] [] [] [] \t[] [] [] [] [] [] [] [] [] [] []\n");

printf("F(1-20)--> [] [] [] [] [] [] [] [] [] \t[] [] [] [] [] [] [] [] [] [] []\n");

printf("G(1-20)--> [] [] [] [] [] [] [] [] [] \t[] [] [] [] [] [] [] [] [] [] []\n");

printf("H(1-20)--> [] [] [] [] [] [] [] [] [] \t[] [] [] [] [] [] [] [] [] [] []\n");

printf("I(1-20)--> [] [] [] [] [] [] [] [] [] \t[] [] [] [] [] [] [] [] [] [] []\n");

printf("J(1-20)--> [] [] [] [] [] [] [] [] [] \t[] [] [] [] [] [] [] [] [] [] []\n");

printf("\n");

printf("\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Silver\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("K(1-20)--> [] [] [] [] [] [] [] [] [] \t[] [] [] [] [] [] [] [] [] [] []\n");

printf("L(1-20)--> [] [] [] [] [] [] [] [] [] \t[] [] [] [] [] [] [] [] [] [] []\n");

printf("M(1-20)--> [] [] [] [] [] [] [] [] [] \t[] [] [] [] [] [] [] [] [] [] []\n");

printf("N(1-20)--> [] [] [] [] [] [] [] [] [] \t[] [] [] [] [] [] [] [] [] [] []\n");

printf("O(1-20)--> [] [] [] [] [] [] [] [] [] \t[] [] [] [] [] [] [] [] [] [] []\n");

printf("P(1-20)--> [] [] [] [] [] [] [] [] [] \t[] [] [] [] [] [] [] [] [] [] []\n");

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Screen this way\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

}

void display\_options()

{

printf("\n1. Create new account \n");

printf("2. Cash withdrawl \n");

printf("3. Account information \n");

printf("4. Log out \n");

printf("5. Clear the screen and display available ");

printf("options \n\n");

}

/\* Function to create new account \*/

void Create\_new\_account()

{

char bank\_name[20];

char bank\_branch[20];

char acc\_holder\_name[30];

int acc\_number;

char acc\_holder\_address[100];

float available\_balance = 0;

fflush(stdin);

printf("\nEnter the bank name : ");

scanf("%s", &bank\_name);

printf("\nEnter the bank branch : ");

scanf("%s", &bank\_branch);

printf("\nEnter the account holder name : ");

scanf("%s", &acc\_holder\_name);

printf("\nEnter the account number(1 to 10): ");

scanf("%d", &acc\_number);

printf("\nEnter the account holder address : ");

scanf("%s", &acc\_holder\_address);

strcpy(account[acc\_number-1].bank\_name,bank\_name);

strcpy(account[acc\_number-1].bank\_branch,bank\_branch);

strcpy(account[acc\_number-1].acc\_holder\_name,

acc\_holder\_name);

account[acc\_number-1].acc\_number=acc\_number;

strcpy(account[acc\_number-1].acc\_holder\_address,

acc\_holder\_address);

account[acc\_number-1].available\_balance=available\_balance;

printf("\nAccount has been created successfully \n\n");

printf("Bank name : %s \n" ,

account[acc\_number-1].bank\_name);

printf("Bank branch : %s \n" ,

account[acc\_number-1].bank\_branch);

printf("Account holder name : %s \n" ,

account[acc\_number-1].acc\_holder\_name);

printf("Account number : %d \n" ,

account[acc\_number-1].acc\_number);

printf("Account holder address : %s \n" ,

account[acc\_number-1].acc\_holder\_address);

//num\_acc++;

}

// Displaying account informations

void Account\_information()

{

register int num\_acc = 0;

//if (!strcmp(customer,account[count].name))

while(strlen(account[num\_acc].bank\_name)>0)

{

printf("\nBank name : %s \n" ,

account[num\_acc].bank\_name);

printf("Bank branch : %s \n" ,

account[num\_acc].bank\_branch);

printf("Account holder name : %s \n" ,

account[num\_acc].acc\_holder\_name);

printf("Account number : %d \n" ,

account[num\_acc].acc\_number);

printf("Account holder address : %s \n" ,

account[num\_acc].acc\_holder\_address);

printf("Available balance : 2000 \n\n");

num\_acc++;

}

}

// Function to deposit amount in an account

void Cash\_withdrawl()

{

int acc\_no;

float withdraw\_money;

printf("Enter account number you want to withdraw money:");

scanf("%d",&acc\_no);

printf("\nEnter money you want to withdraw from account ");

scanf("%f",&withdraw\_money);

while (acc\_no=account[acc\_no-1].acc\_number)

{

printf("\*\*\*\*\*\*\*\*\*\*\*\*payment successful\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

break;

}acc\_no++;

}

**Literature survey:**

**Old Theatre Booking System**

The theatre provides infrastructure and facilities for a performance to take place, the same way in which a cinema provides screens and projectors to show movies. An audience patronize this for a fee. A booking system is used to ensure customers can purchase tickets for a given performance well in advance and avoid being disappointed at the last minute. Theatre owners also prefer tickets to be sold as early as possible, both for financial reasons [1], and to avoid a long queue at the entrance just before the performance is due to start. The theatre box office is the section where the bookings are made, being the first point of contact between the theatre and the public [1].

Before electronic printers were introduced in box office, ticket books were used. The theatre staff would also have a plan for each performance, with a corresponding ticket book. The customers would have access to a diagram of the theatre to indicate their preferred seats. The staff was also expected to know about the performance in question, so that customer questions could be dealt with in satisfactory manner [2]. The customer would finally pay for the agreed seats, and receive tickets printed with the respective seat numbers. The staff would have to carefully mark off the seats on the seating plan for every ticket sold, to avoid double booking.

Seats were also sold by telephone or mail order to those customers who could not readily pay a visit to the box office. In the mail order system, the customer would post a cheque with an order for the desired seats, date and time, with a stamped, self-addressed envelope [3]. If the order arrives early enough, the box office would book the seat requested, and post the appropriate tickets back to the customers, who would receive them at least two days after the order was first sent. However, if the order had been sent after desired seats had been sold, the box office would have to send the customers a refund, with the list of the remaining seats [2]. This process would then be repeated until the customers either accept an alternative booking, or demand a refund [2] if all desirable seats had been sold out. Therefore, mail order could also result in disappointment to the customer and a revenue loss to the cinema. To make the mail ordering process more effective, theatres also sold seats for a whole season of performance [2]

On some occasions, not every tickets would have been sold out, and some ticket holders would not attend. In order to maximize profits, the box office would sell any unreserved seats at subsidized rates a few hours before the show starts [3], [4]; and the subsidized tickets would be clearly indicated on the tickets stub (which is retained in the box office) for easy accountabilities at the end of the day [5]. Sales were sometimes deduced by counting the number of unsold tickets after the performance had begun [7], but due consideration would have to be given regarding any discounting ticket sold.

The theatre manager was concerned with various tickets sales report, such as master statement showing the number of seats at each price and the total potential income on ticket sales, daily reports, weekly reports and seasonal reports [5]; [8], An example of a ‘**Weekly Returns’** (gives the seats occupation as a percentage and total money collected for each day of the week) and **‘Final Returns’ (**which gives the total money collected in advance booking, at the doors and percentage of seats occupied). [6]

Therefore, theatre managers in the past made use of tickets sales reports and the reporting style varied from one theatre to the other .

**Modern Theatre Booking Systems** Although the fundamental principles of a theatre booking system have remained unchanged overtime, developments in communication and other technologies have had major impact on their implementation. In 1969, *Sweeting* [2] speculated that perhaps theatres could benefit from computerized ticketing system, similar to what airlines at the time had started to adopt. In the late 1960s, the personal computer was not common and were bulky, expensive industrial machines which would set up to execute tasks specific to the organization in question, making them too expensive for small and medium enterprises to acquire.

Langley in 1980 suggested that computer-based ticket system was only feasible for large theatre operations [5]. However, with today’s comparatively low hardware cost, the computer has become affordable and is common in most offices. Even the smallest of theatres coulf afford computerized system. This made it possible for theatre bookings to be done faster than mail order described in the previous section.

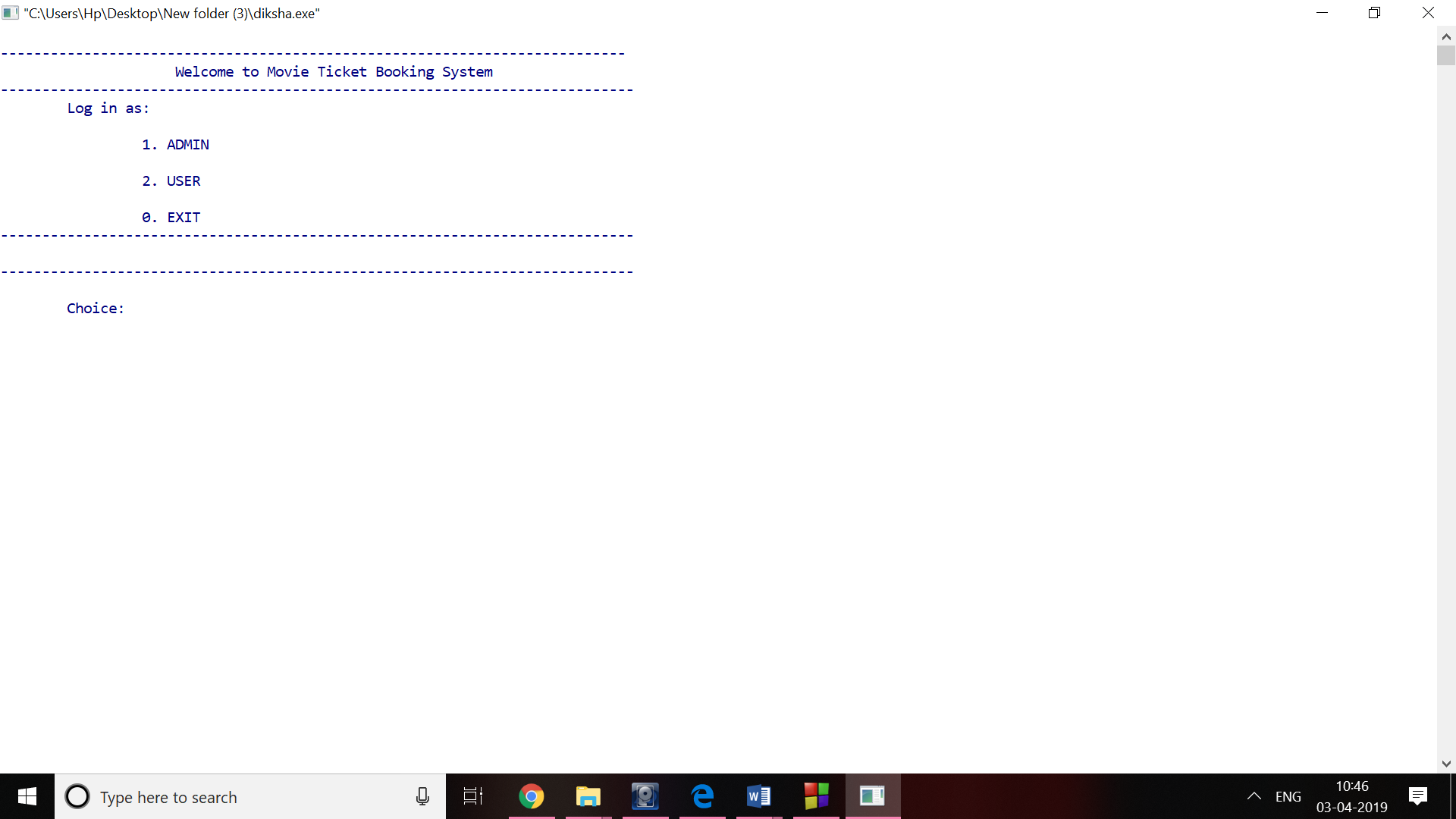
Reid in 1983 observed that computers were replacing the traditional paper and pencil systems, allow several customer care staff to simultaneous access the same seating plan, thus avoiding queues and improving on customer services, compared to the manual system [6]. Hillenbrand in 2001 proposed that theatres should also make use of the opportunity provided by other service providers such as toll free numbers, credit card processing, internet and mail order, all of which today’s consumers make use of. [1]

In addition to knowing about the performance, the modern day box office staff is expected to be able to give customers advice on facilities like restaurants and parking, as well as work outside normal working hours, in order to provide convenient opening hours to the public [4].

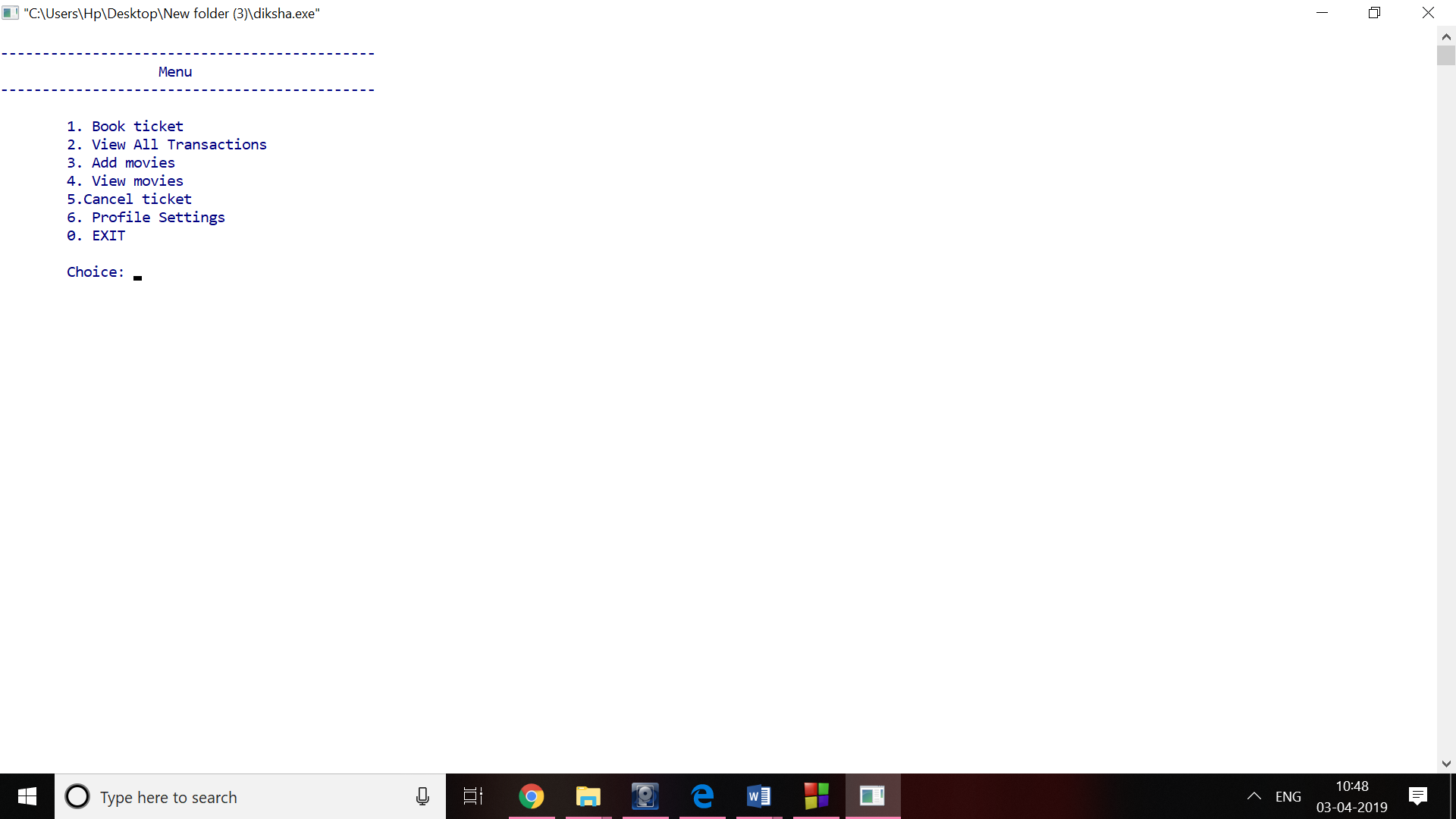
Since the theatre of today has to compete with other entertainment facilities available to customers, it has to market its services widely such a making discount tickets available to the press, producers, actors etc, as a means of promoting the theatre.

Modern day theatre managers still make use of the same types of report that were used decades ago, but have come to expect them instantly due to computerization. A computerized reservation system has the potential to provide the precise type of information required with just a mouse click. Comparing manual and computerized booking systems, Collins quotes a manual theatre manager describing it as ‘labor intensive, bulky and hard to make changes’ as opposed to ‘fast, easy and sleek’ respectively ([7] p. 125)

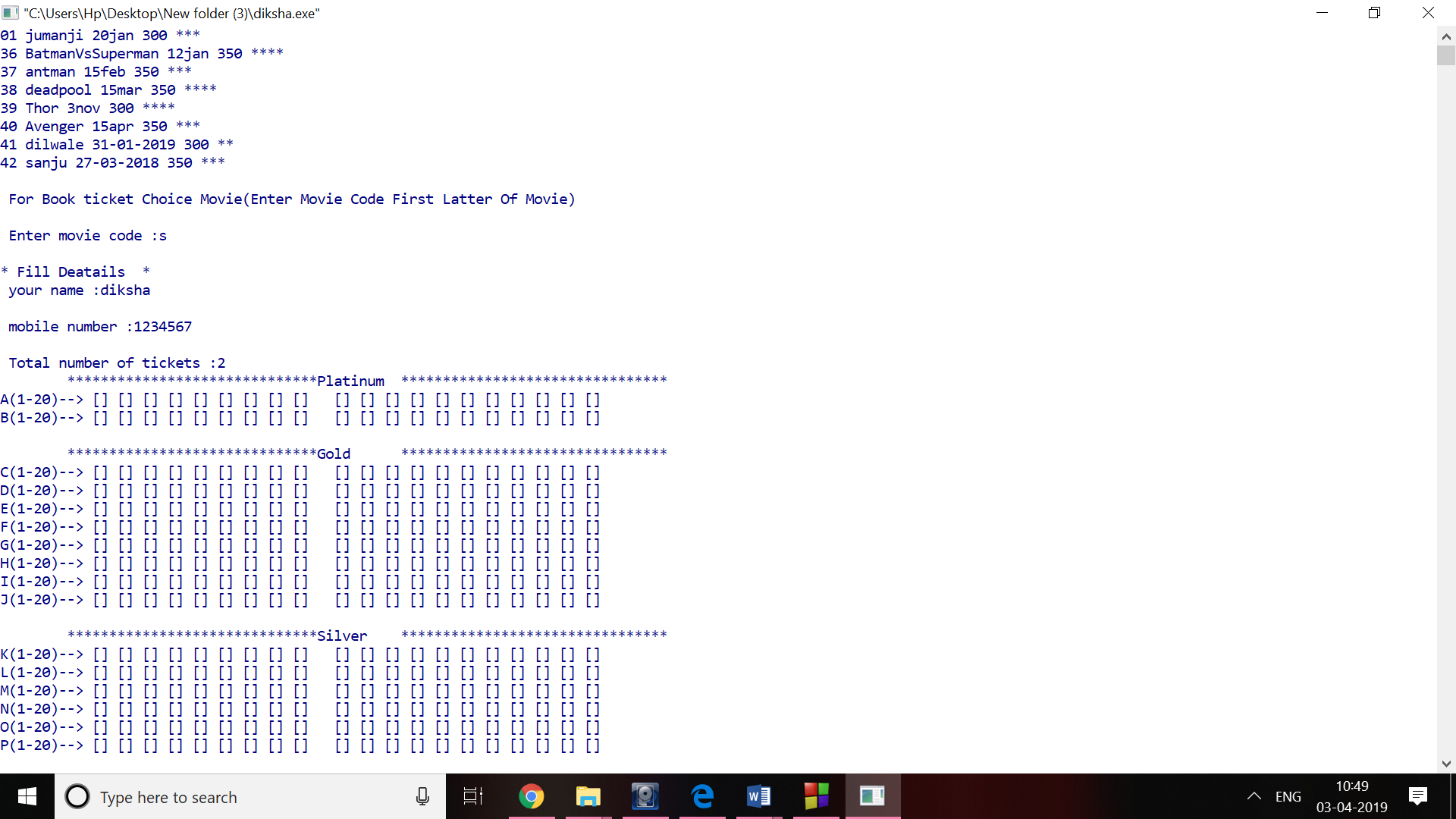
**Output:**

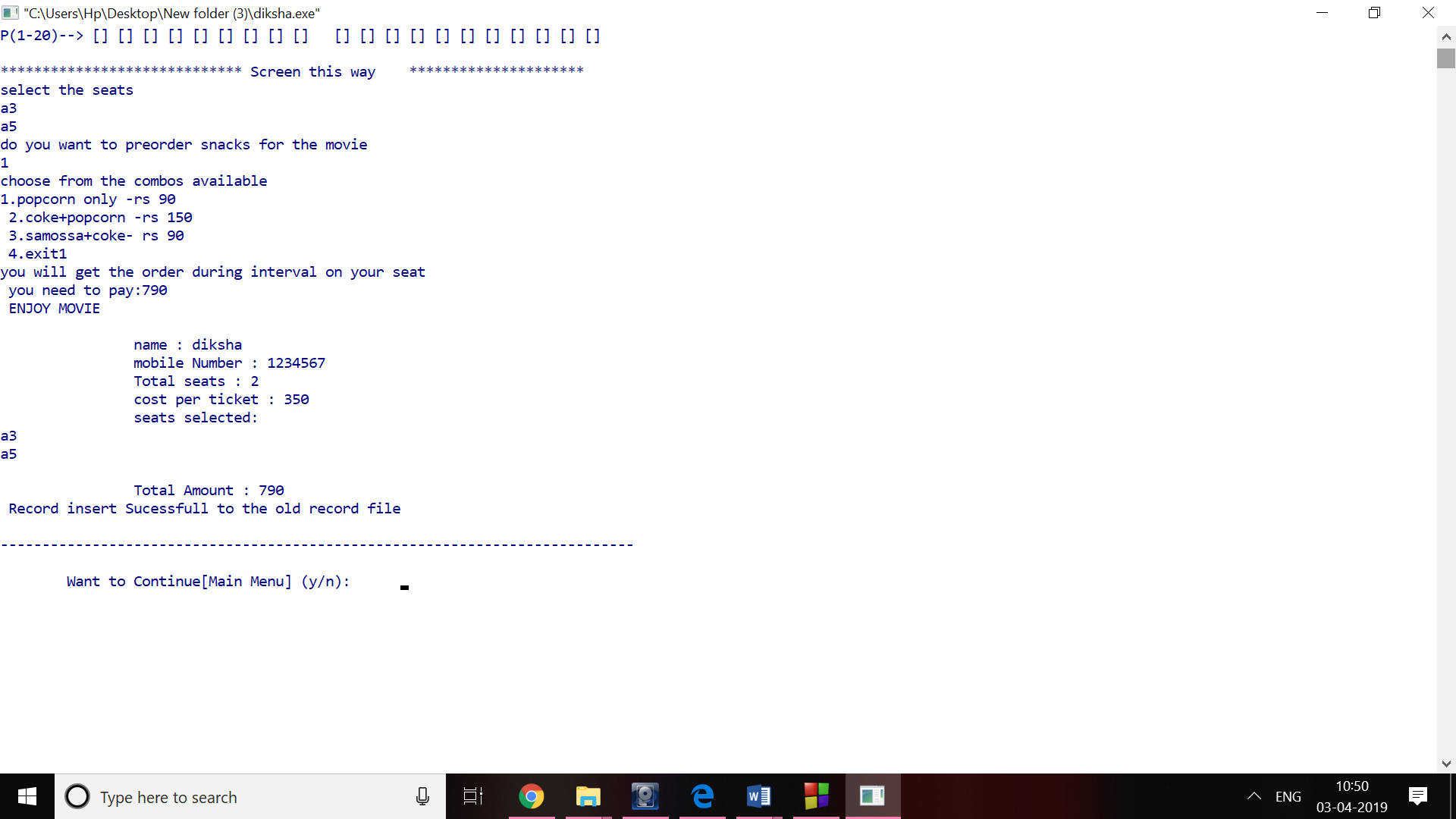


**In admin profile:**

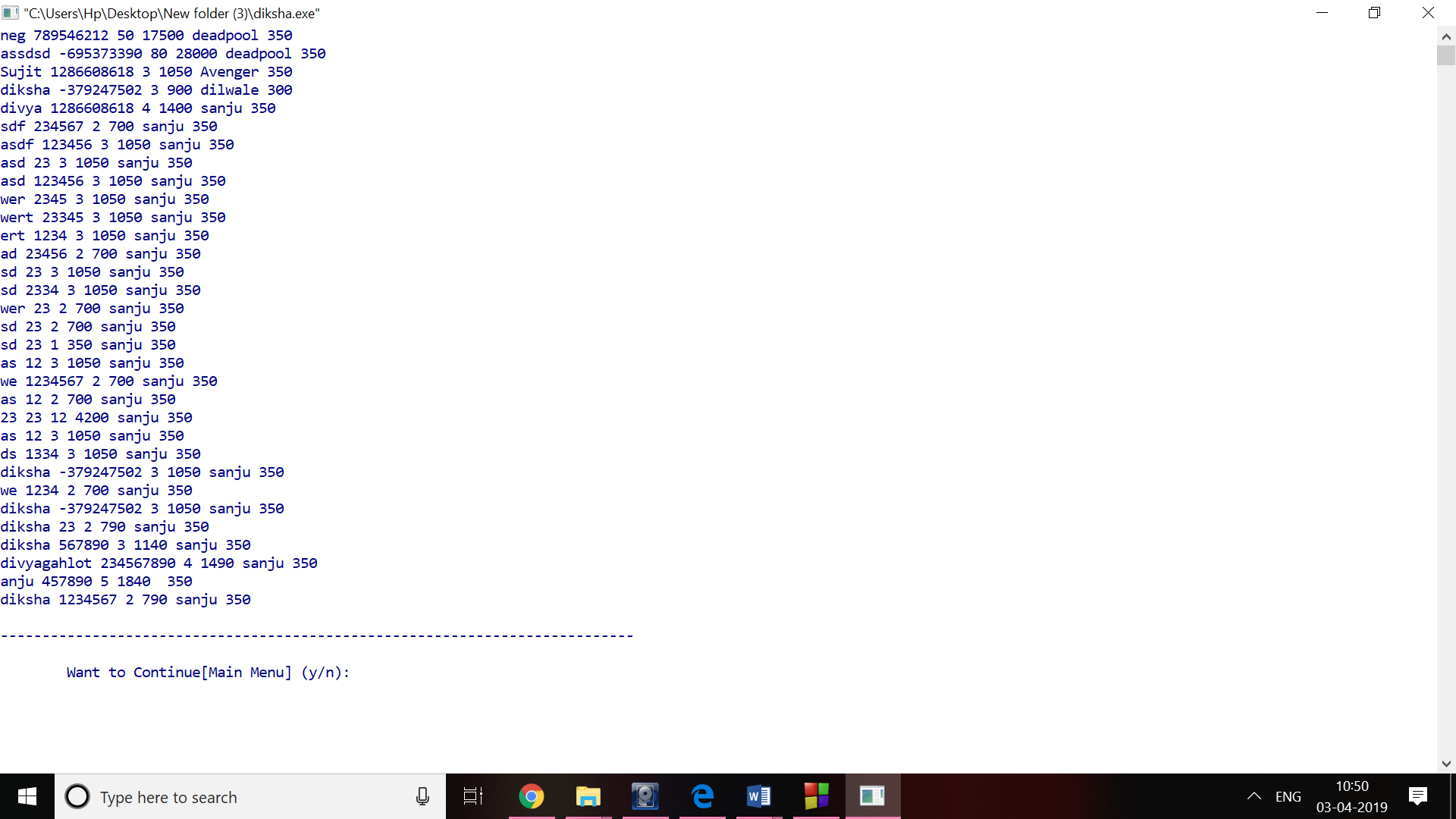


**Booking ticket:**

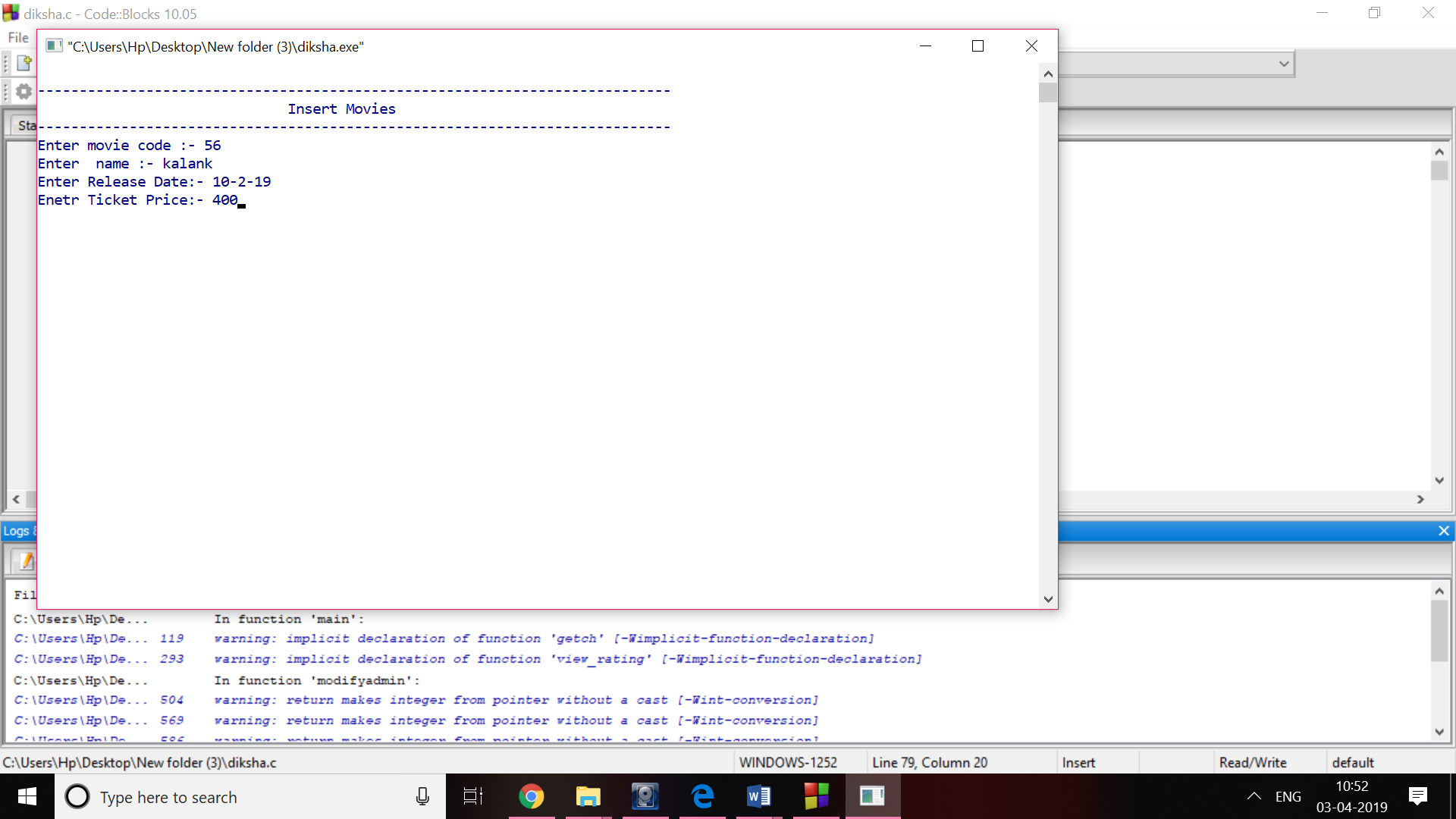




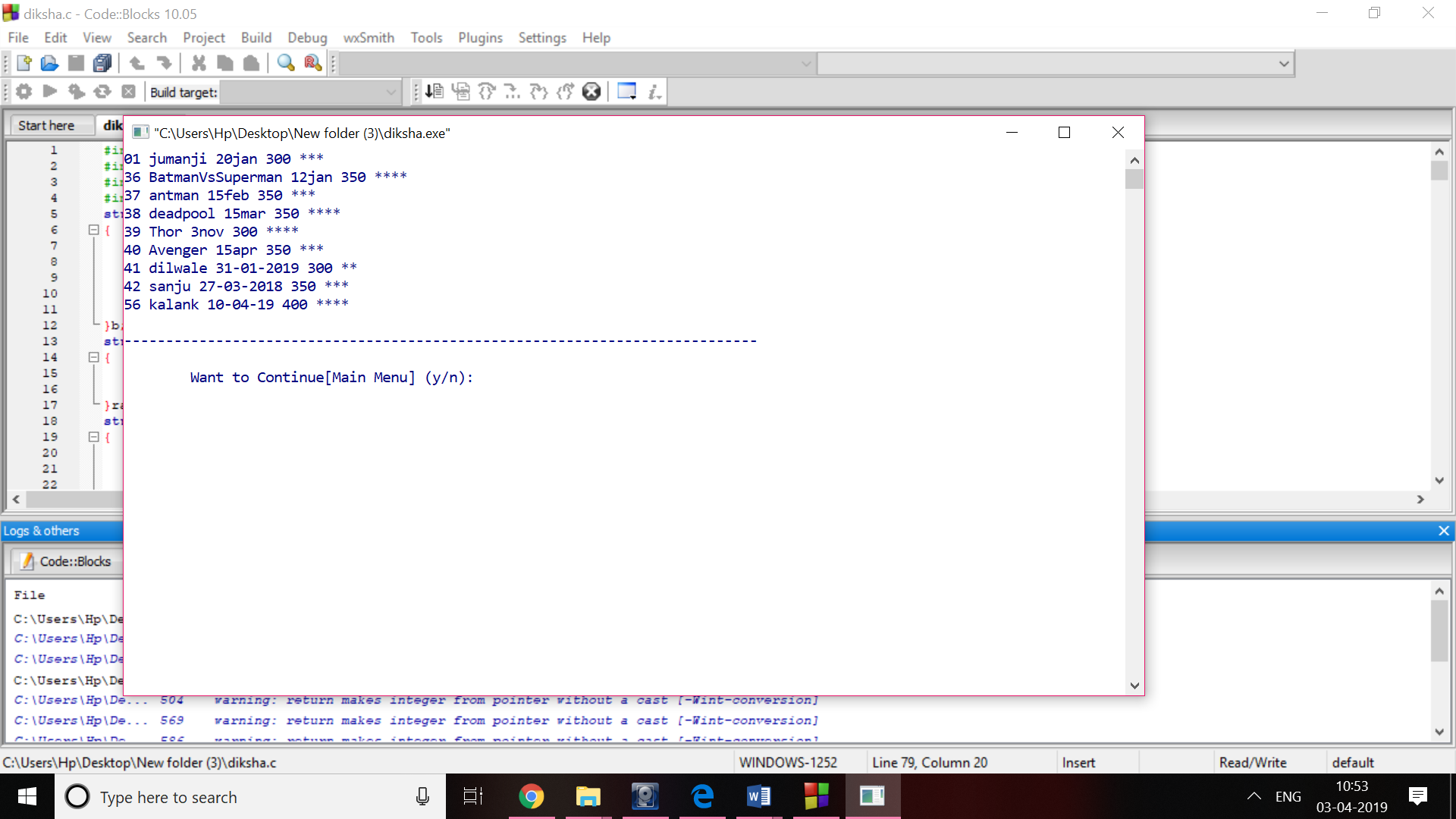
**View all transaction:**



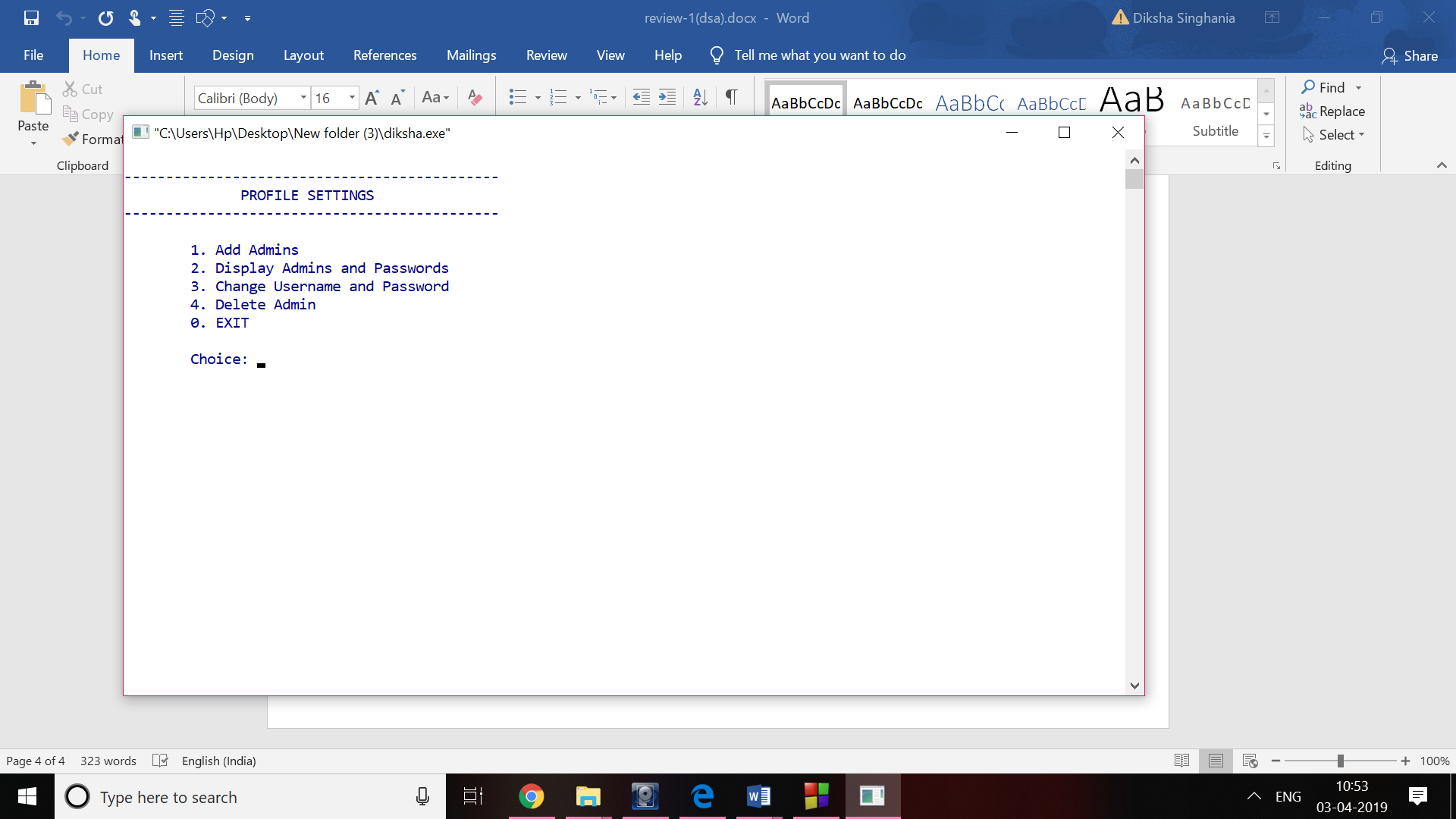
**Add movie**



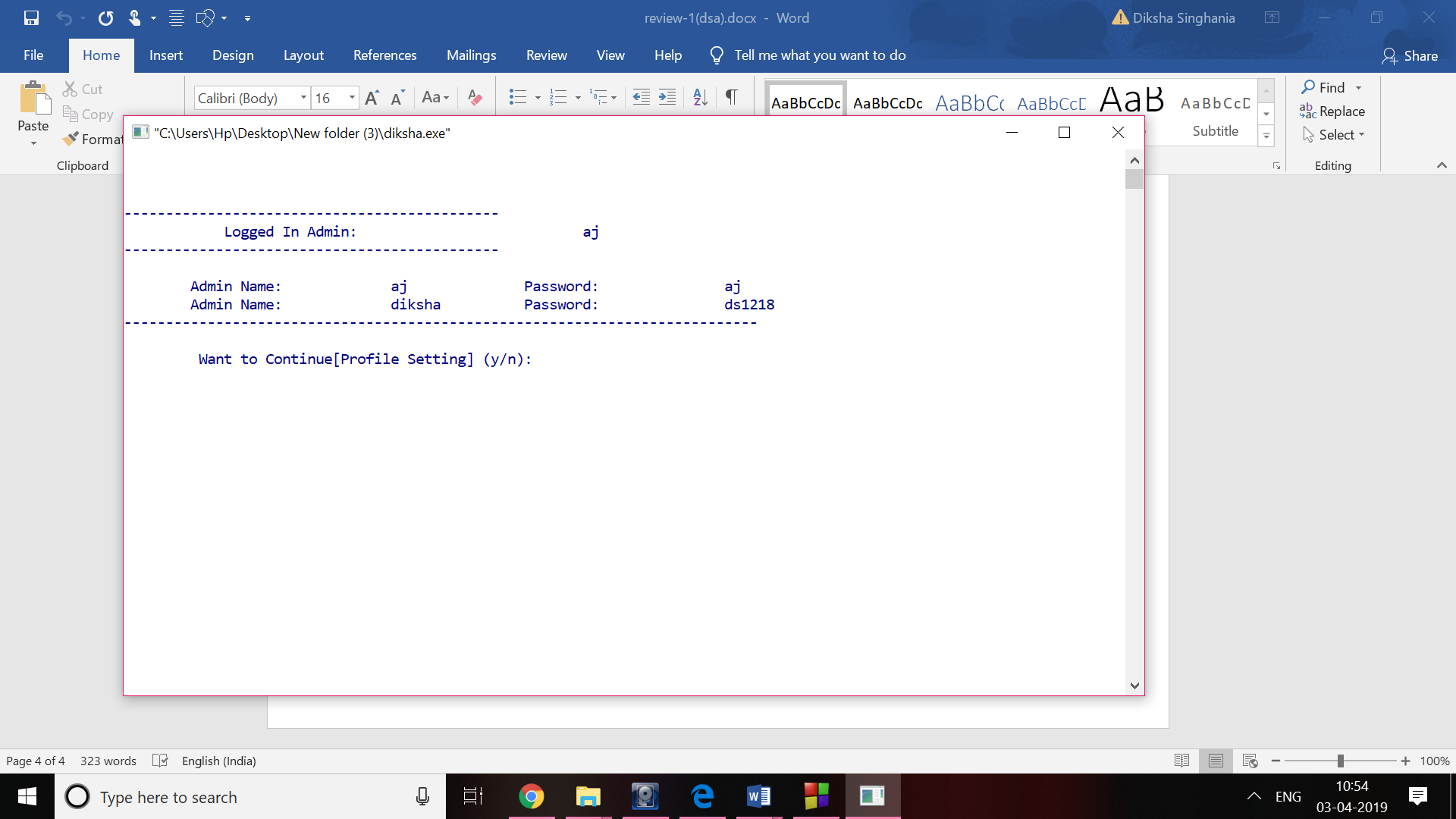
**View movie**



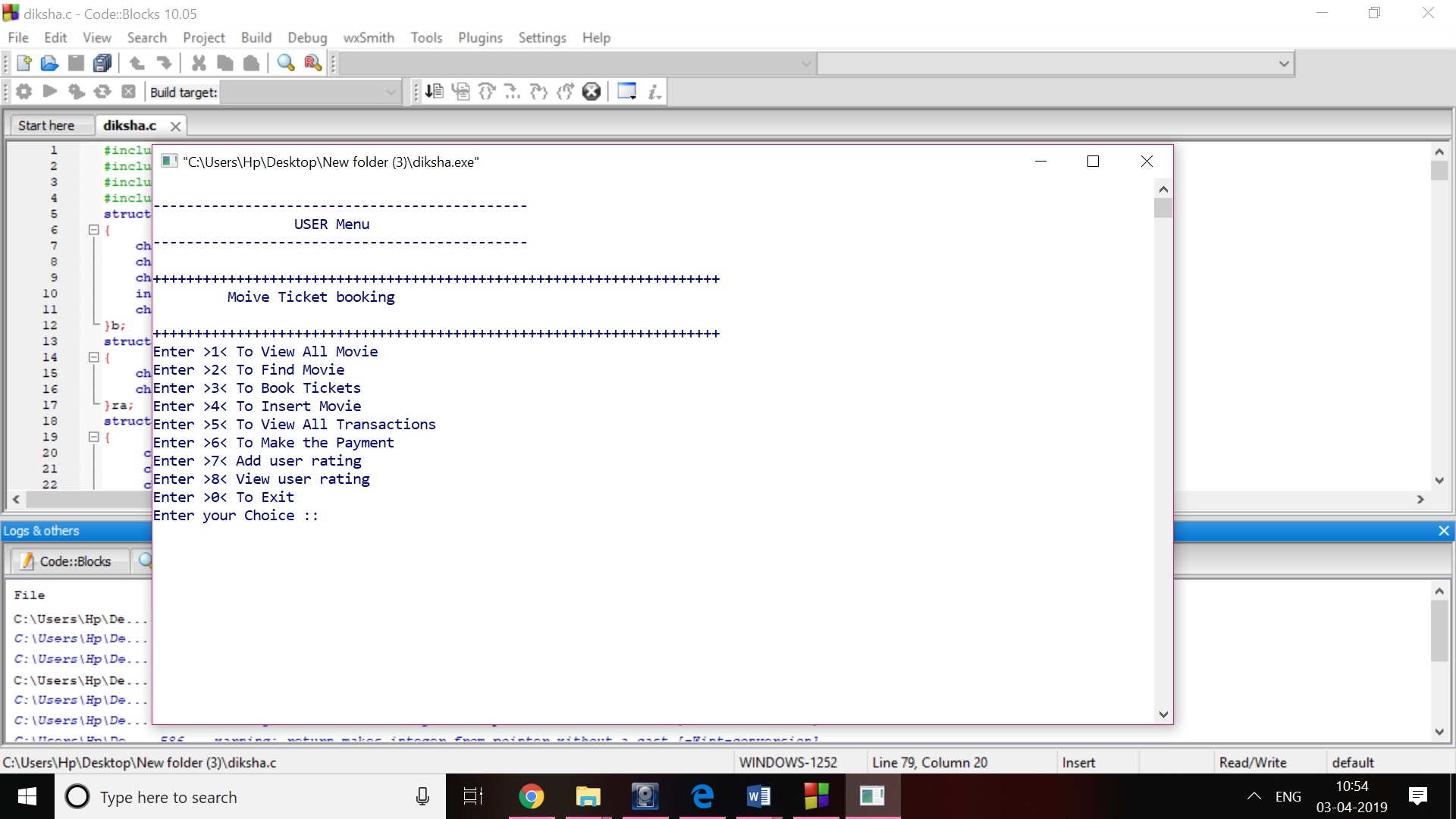
**Profile settings:**



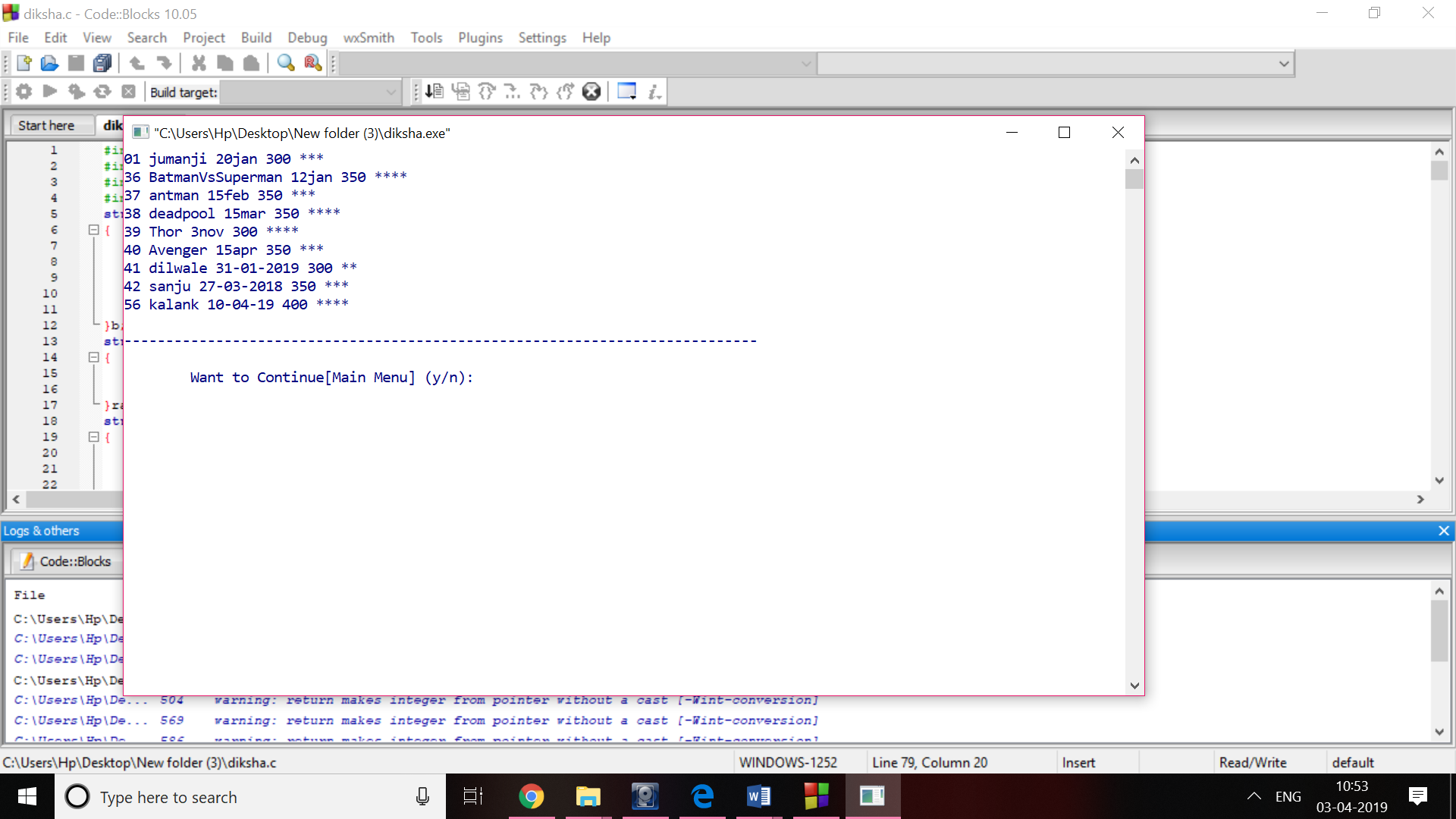
**Display admins and passwords**



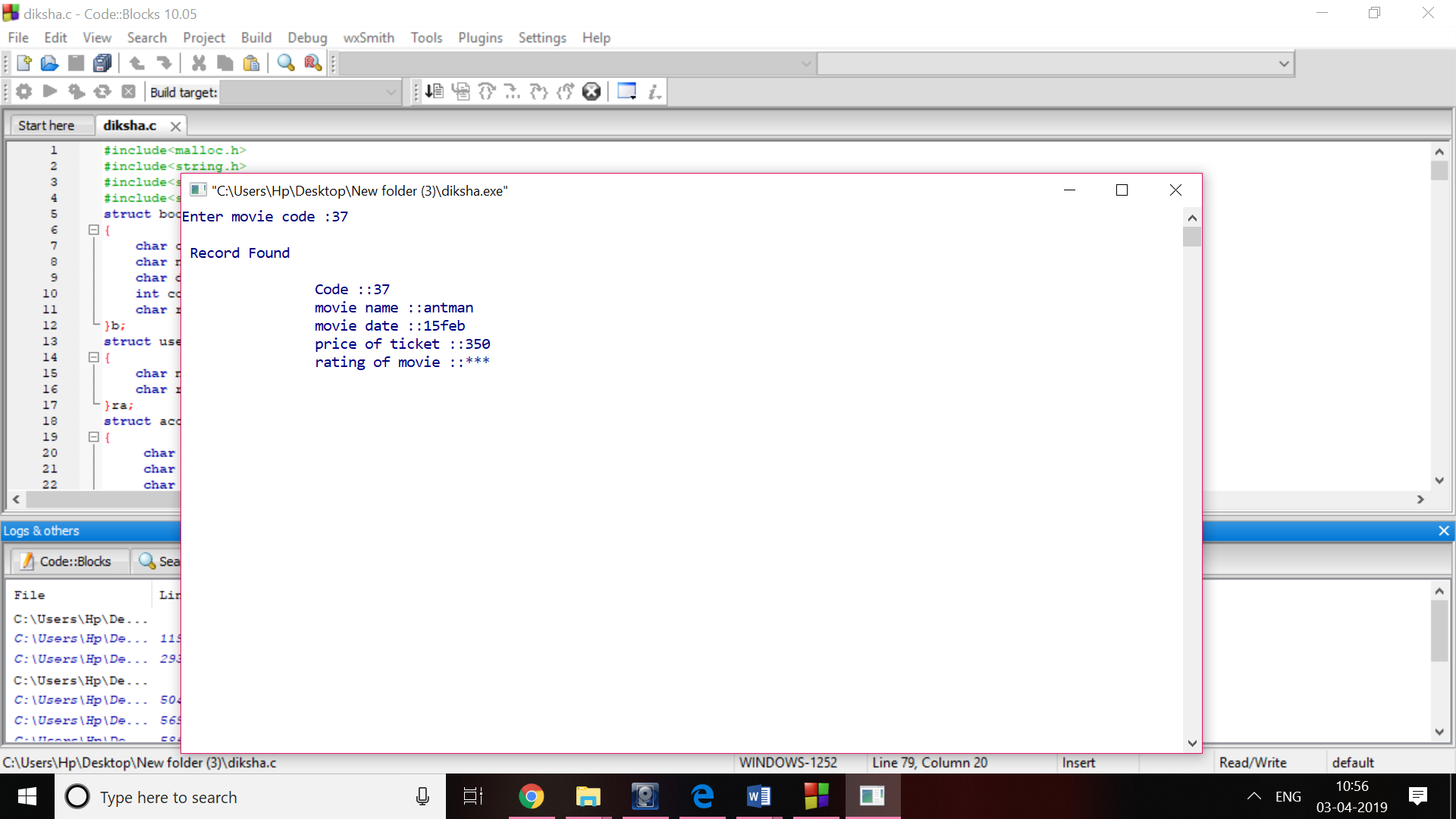
**User:**



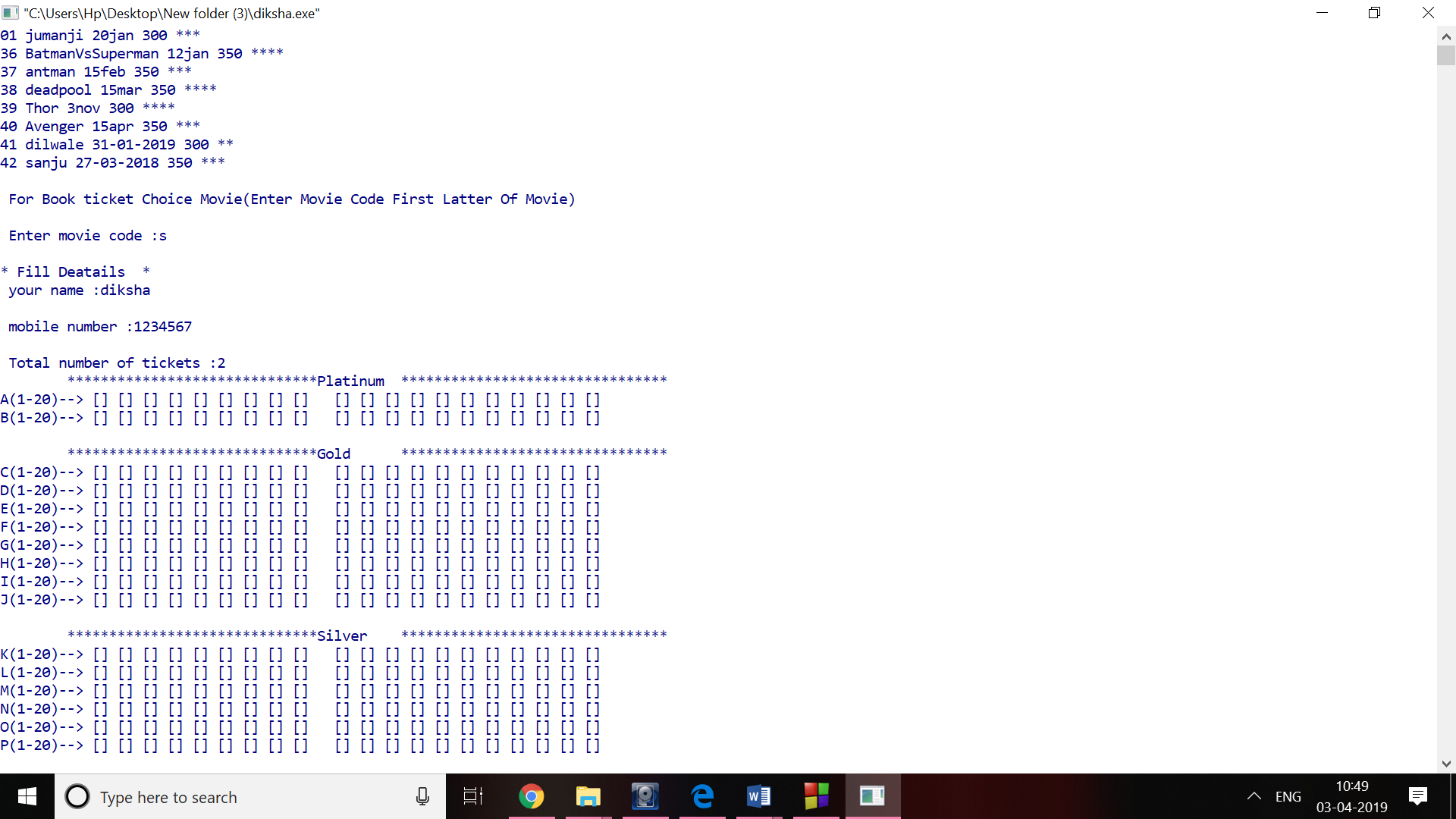
**View all movie**

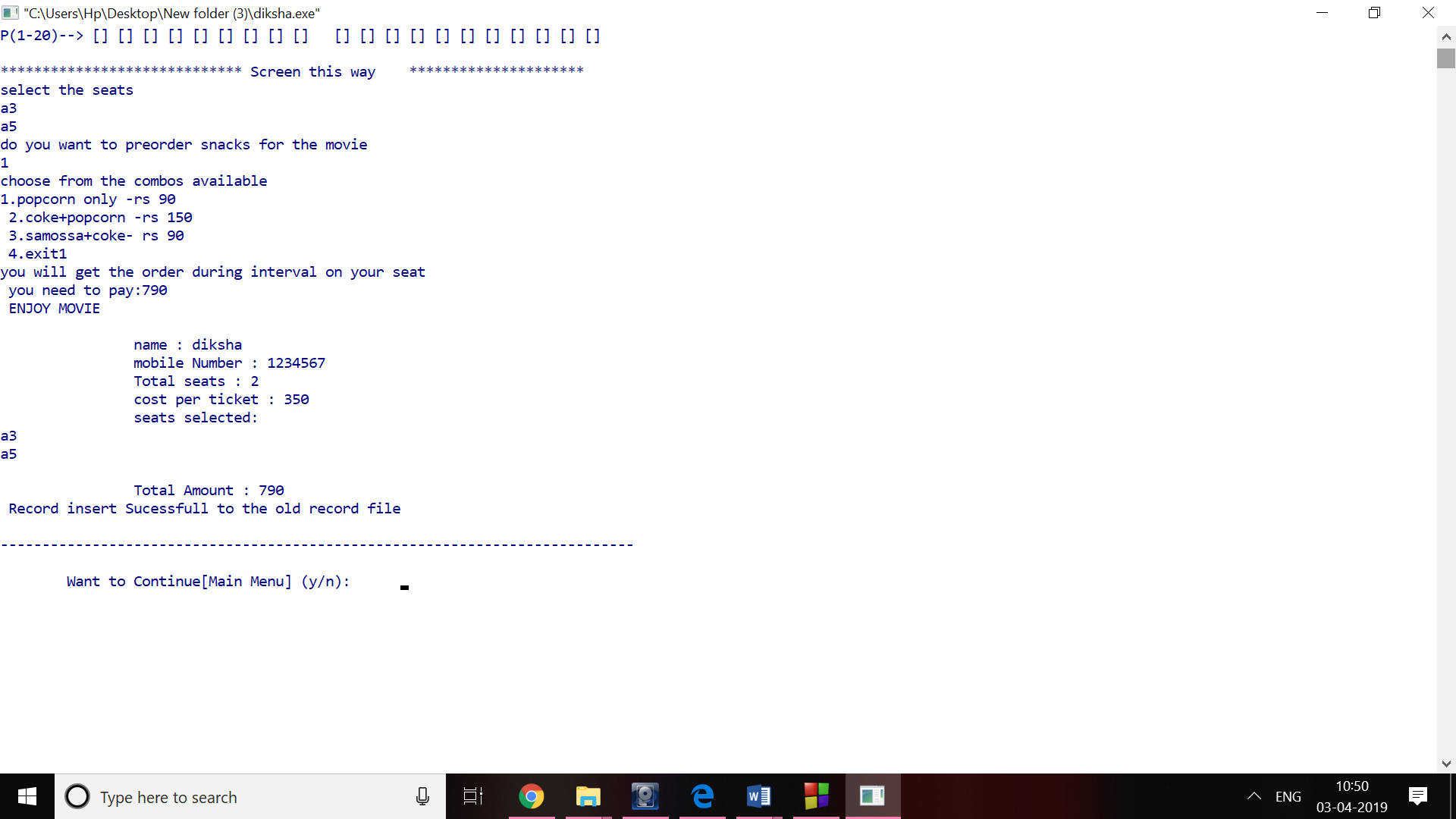


**Find a movie:**

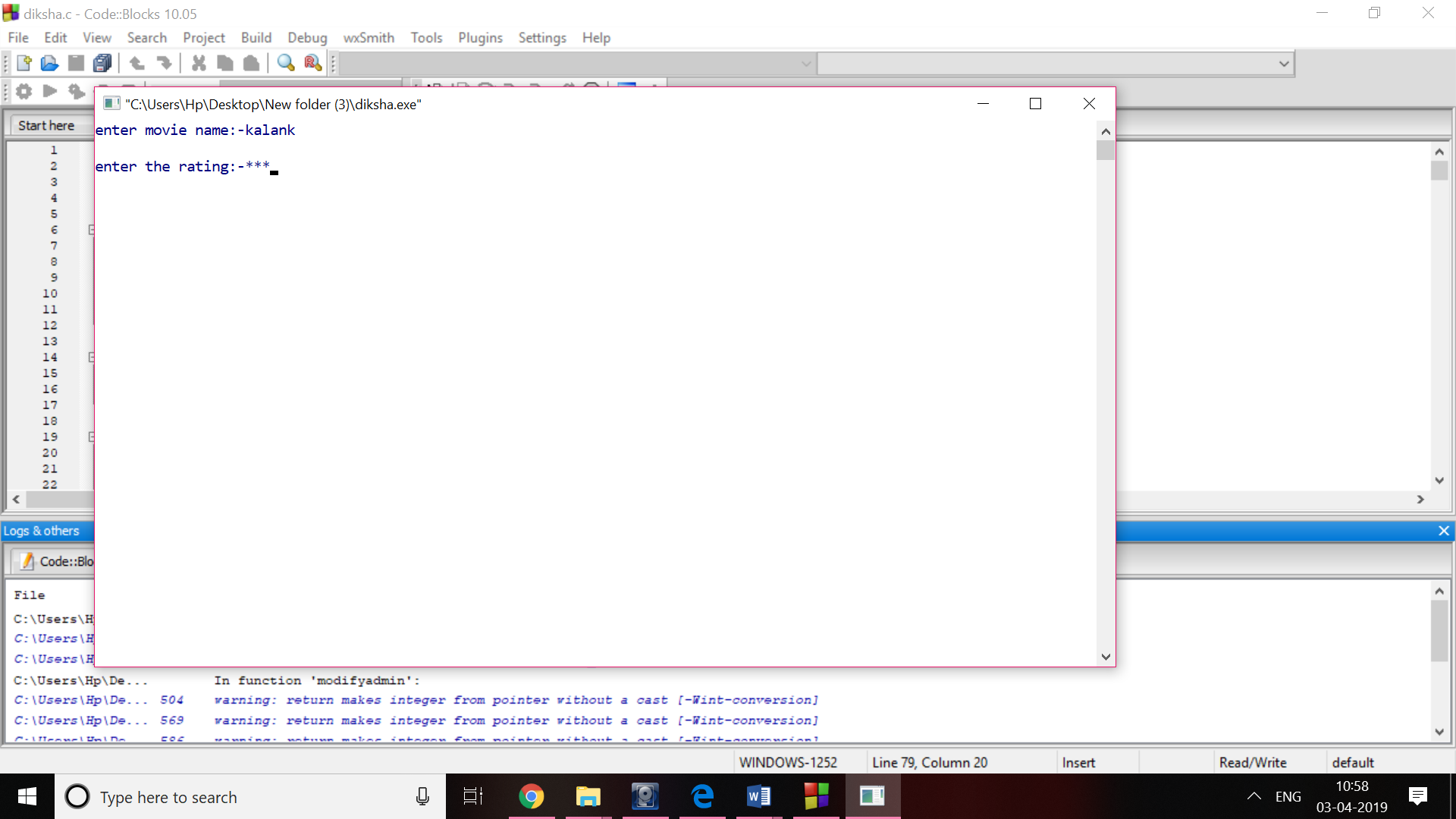


**Book ticket:**

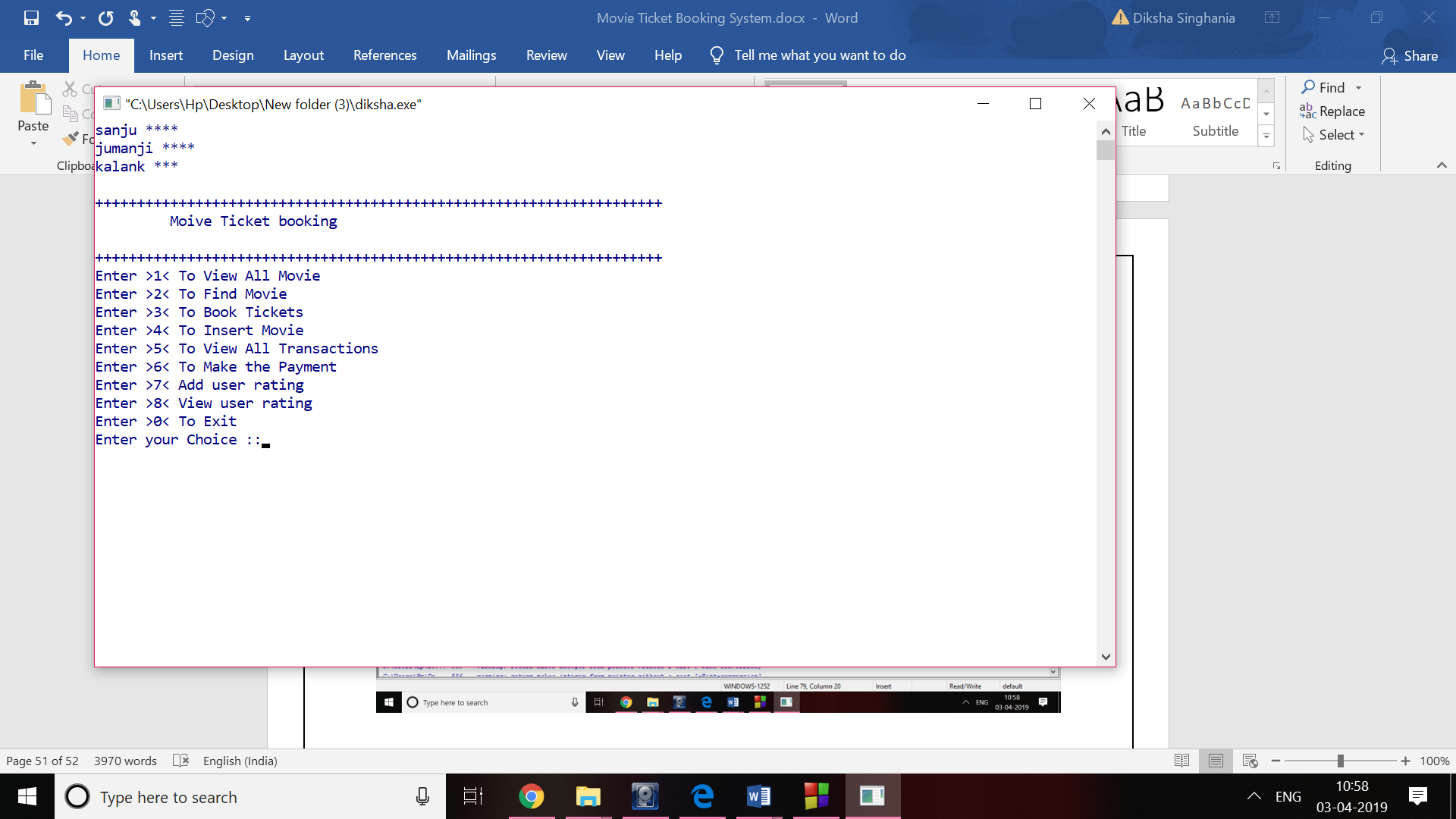




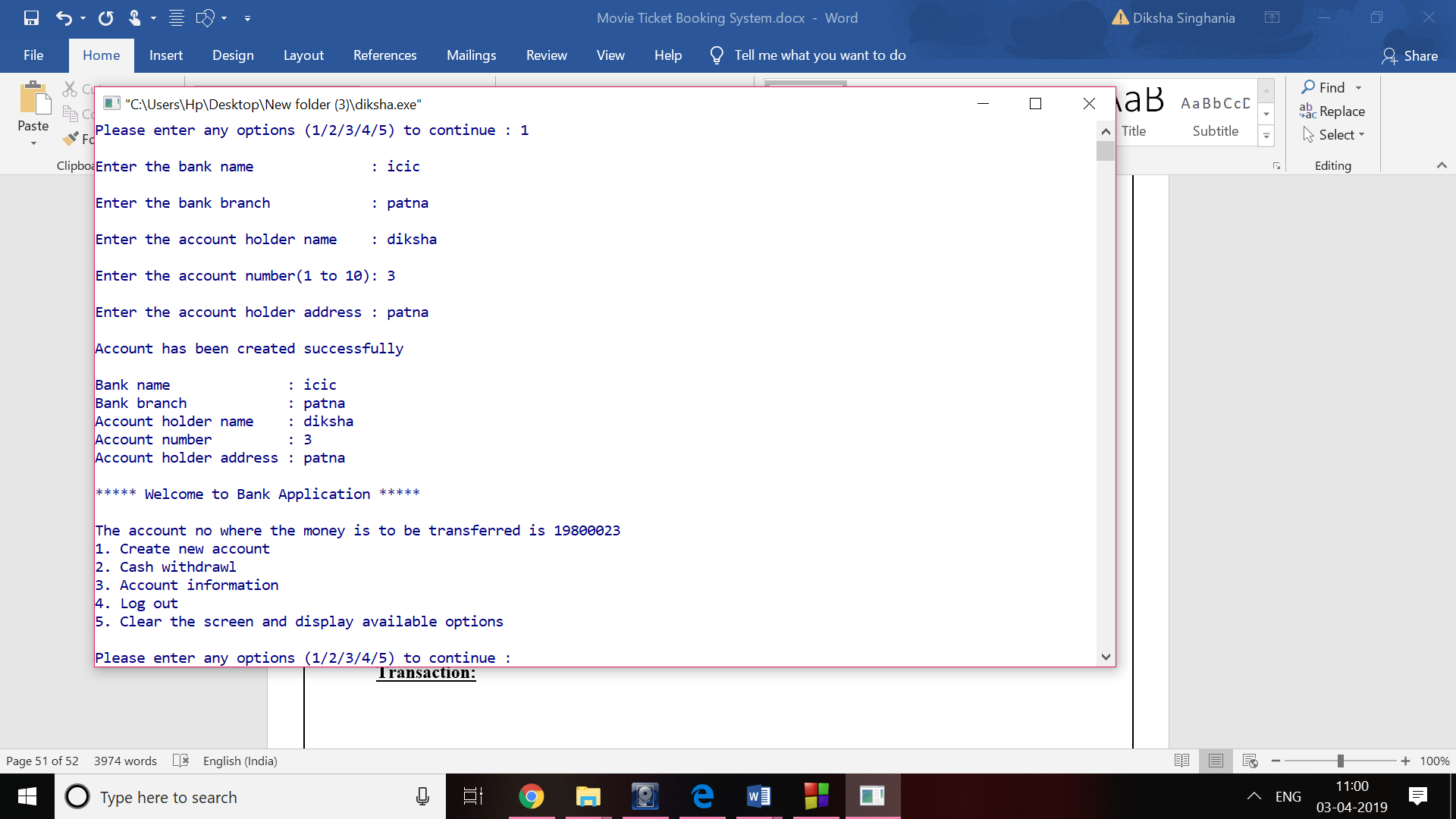
**Add user rating:**



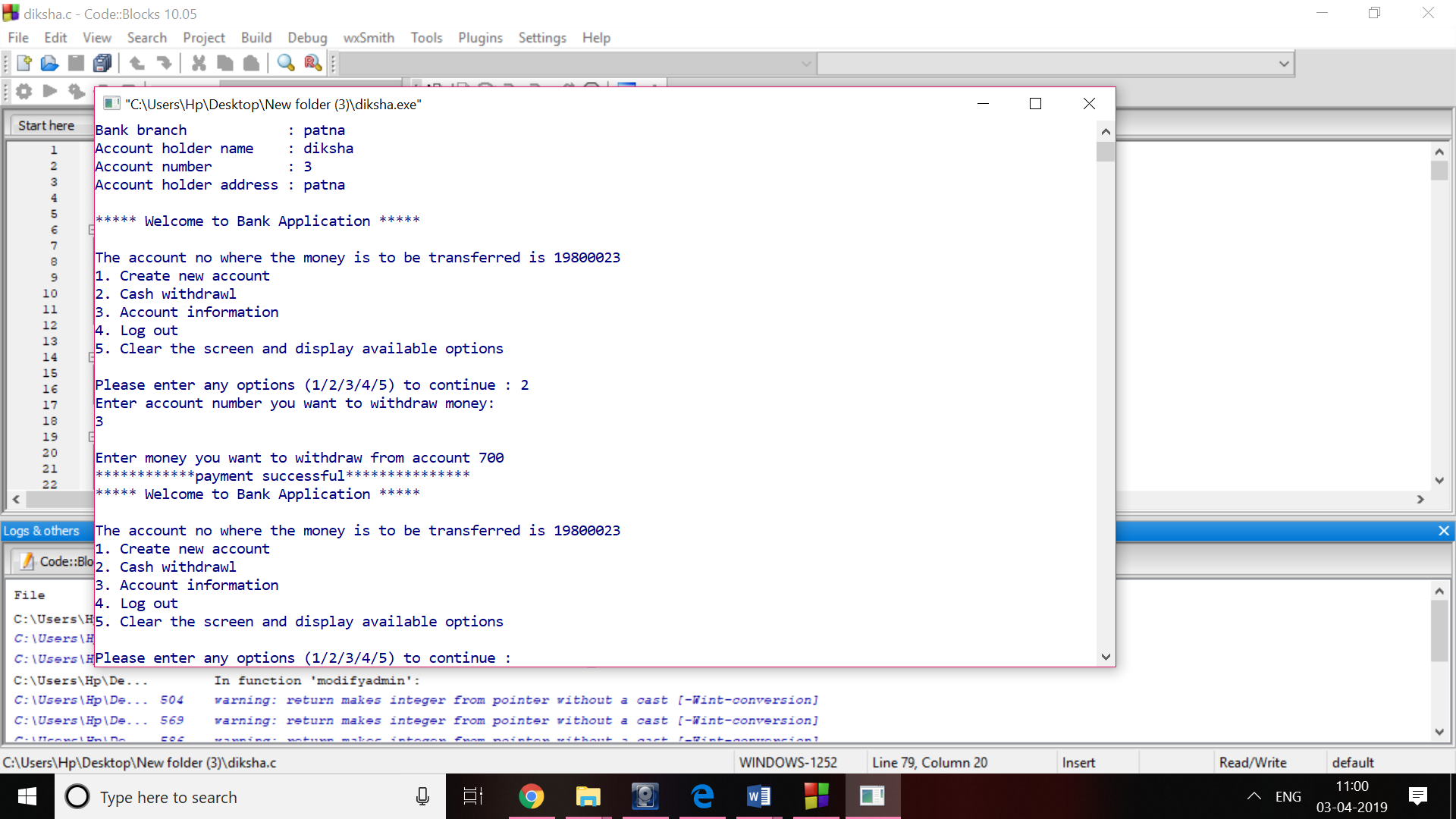
**View user rating:**



**Transaction:**



**Payment:**



**References:**

[1] *Hillenbrand, M 2001, Produce Your Play Without a Producer, Smith ans Kraus, Inc, Hanover, NH, USA.*

[2] *Schneider, Re & Ford, MJ 1993, The Well-Run Theatre, Drama Book Publishers, New York*

[3] *Pick, J 1985, The Theatre Industry, Comedian Publishing Group, UK.*

[4] *Carmack Celentano, S & Marshall, K 1998, Theatre Management, Players Press, Inc, USA*

[5] *Langley, S 1980, Theatre Management in America Principles and Practice, Drama Book Publishers, New York.*

[6] *Sweeting, E 1969, Theatre Administration, Sir Issac Pitman and Sons Ltd, London.*

[7] *Collins, T 2003, How Theatre Managers Manage, The Scarecrow Press, Inc.*