



CinePlus

AN ONLINE MOVIE BOOKING DASHBOARD

PROJECT REPORT

COMPUTER SCIENCE

Name- Shiva Shaklya

Board Roll no.- 21663386

School- DPS Indirapuram

Session- 2021-22

Certificate

This is to certify that SHIVA SHAKLYA of Class XII-F has prepared a project on “CINEPLUS: MOVIE BOOKING PROGRAM”. The project is a result of her efforts and endeavors. This project is found worthy of acceptance as the final project report for the subject Computer Science of Class XII.

She has prepared this project under my guidance.

Ms. Rinkoo Gupta
(Computer Science)
(DPS Indirapuram)

Acknowledgement

I would like to express a deep sense of gratitude towards my computer science teacher Ms. Rinkoo Gupta Ma'am for guiding through the course of my project. She always evinced keen interest in my work and her constructive advice and constant motivation have been responsible for the successful completion of the project.

I would like to extend my sincere thanks to everyone who have helped directly or indirectly in the completion of the project.

Index

Table of Contents

INTRODUCTION	5
MySQL and CSV files	7
CODE.....	10
OUTPUT.....	62
CONCLUSION.....	80
FUTURE IMPROVEMENTS	81
BIBLIOGRAPHY	82

INTRODUCTION

The aim of the project was to build an online movie booking dashboard called “CINEPLUS”.

Working Description

The build-up of the user-menu section is as follows:

1. Viewer

1.1. Register

1.2. Sign In

Δ Choose a Genre

- Click on a Movie
- Book Now
- Name, Date of Birth and Seats Required
- Confirm Booking + Exit

1.3. Back

2. Manager

2.1. Sign In

Δ Booking Details

Δ Update Poster

- Choose a Genre
- Click on a Movie
- Enter New Movie + Click Enter + Ok

Δ Database Change

∞ View Movie

∞ Change Details

- Seats Available

- Movie Name + No. Of Seats + Click Enter
- Change Show Timings
 - New Show Timings + Click Enter
- Cost Per Head
 - Enter Cost + Click Enter
- Add Movies
 - Enter Movie Details Required + Click Enter
- Delete Movies
 - Movie Name + Click Enter

2.2. Change Password

Δ Current Password + New Password

2.3. Back

MySQL and CSV files

Within MySQL 5 tables have been used:

```
mysql> show tables;
+-----+
| Tables_in_project1 |
+-----+
| all_movies
| booking_details
| displayed_movies
| managers
| users
+-----+
5 rows in set (0.00 sec)
```

1. All_movies

mname	descp	show_timing	duration_min	seat_avail	cost_per_head	genre	location
3 Idiots	Directed by Rajkumar Hirani	11:00am	180	50	250	Comedy	3Idiots.jpg
Jab We Meet	Directed by Imitiaz Ali	12:45pm	195	45	170	Comedy	J.jpg
Gravity	Directed by Alfonso Cuarón	10:00pm	95	50	230	Thriller	Gravity.jpg
World War Z	Directed by Marc Foster	12:00pm	140	30	300	Thriller	WorldWarZ.jpg
Inception	Directed by Christopher Nolan	2:00pm	120	44	450	Thriller	Inception.jpg
Avatar	Directed by James Cameron	6:00pm	115	35	235	Adventure	avatar.jpg
Constantine	Directed by Francis Lawrence	5:30pm	150	48	370	Action	Constantine.jpg
Veer Zaara	Directed by Yash Chopra	2:00pm	200	76	200	Romance	VeerZaara.jpg
Dil Chahta	Directed by Farhaan Akhtar	9:00am	165	45	245	Romance	Dilchahtahai.jpg
Little Women	Directed by Greta Gerwig	11:00am	100	20	350	Romance	LittleWomen.jpg
I Frankenstein	Directed by Stuart Veattie	12:00pm	75	34	150	Action	frank.jpg
The Dark Tower	Directed by Nikolaj Arcel	3:00pm	200	30	200	Action	dark.jpg
Annabelle	Directed by John Robert	9:00pm	155	30	350	Horror	annabelle.jpg
Cars	Directed by John Lasseter	2:00pm	90	52	250	Kids	cars.jpg
Toys Story	Directed by John Lasseter	12:00pm	130	90	100	Kids	toys.jpg
Frozen	Directed by Jennifer Michelle Lee	5:00pm	120	20	175	Kids	frozen.jpg
IT	Directed by Andres Muschietti	4:00pm	135	55	200	Horror	it.jpg
Insidious The Last Key	Directed by Adam Robitel	11:30pm	103	40	290	Horror	insidious.jpg
The Conjuring 2	Directed by James Wan	11:00pm	134	34	210	Horror	c2.jpg
Pineas and Ferb	Directed by Robert Hughes	4:30pm	90	45	120	Kids	pf.jpg
Tom and Jerry	Directed by Tim Story	3:00pm	101	60	100	Kids	TomJerry.jpg
Scooby Doo	Directed by Raja Gosnell	3:45pm	90	50	128	Kids	Scoob.jpg
Twilight	Directed by Catherine Hardwicke	5:00pm	126	50	200	Romance	twilight.jpg
The Parent Trap	Directed by Nancy Meyers	3:30pm	128	60	260	Comedy	parenttrap.jpg
The Fault In Our Stars	Directed by Josh Boone	2:00pm	133	90	280	Romance	stars.jpg
Extraction	Directed by Sam Hargrave	5:00pm	116	30	320	Thriller	extraction.jpg
Justice League	Directed by Zack Snyder	5:30pm	120	10	200	Action	JL.jpg
Interstellar	Directed by Christopher Nolan	4:00pm	175	15	150	Adventure	in.jpg
Jumanji: Welcome To The Jungle	Directed by Jake Kasdan	4:30pm	119	25	300	Adventure	jumangi.jpg
Jurassic World	Directed by Colin Trevorrow	3:45pm	124	24	250	Adventure	jurassicworld.jpg
Fast and Furious 8	Directed by F. Gary Gray	2:30pm	136	13	300	Action	ff8.jpg
Adventures In Babysitting	Directed by John Schultz	1:45pm	99	40	240	Adventure	babysitting.jpg
Home Alone	Directed by ABC	2:00pm	120	25	120	Kids	H.jpg
Ra.one	Directed by ABC	2:00pm	120	30	40	Action	r1.jpg

34 rows in set (0.10 sec)

2. Displayed_movies

```
mysql> select * from displayed_movies;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| mname | descp | show_timing | duration_min | seat_avail | cost_per_head | genre | location | position |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Interstellar | Directed by Christopher Nolan | 4:00pm | 175 | 15 | 150 | Adventure | in.jpg | 1 |
| Avatar | Directed by James Cameron | 6:00pm | 115 | 23 | 235 | Adventure | avatar.jpg | 2 |
| Jumanji-Welcome To The Jungle | Directed by Jake Kasdan | 4:30pm | 119 | 25 | 300 | Adventure | jumangi.jpg | 3 |
| Constantine | Directed by Francis Lawrence | 5:30pm | 150 | 40 | 370 | Action | constantine.jpg | 1 |
| Fast and Furious 8 | Directed by F. Gary Gray | 2:30pm | 136 | 13 | 300 | Action | ff8.jpg | 2 |
| The Dark Tower | Directed by Nikolaj Arcel | 3:00pm | 200 | 30 | 200 | Action | dark.jpg | 3 |
| Gravity | Directed by Alfonso Cuarón | 10:00pm | 95 | 40 | 230 | Thriller | gravity.jpg | 1 |
| World War Z | Directed by Marc Foster | 12:00pm | 140 | 30 | 300 | Thriller | worldwarz.jpg | 2 |
| Extraction | Directed by Sam Hargrave | 5:00pm | 116 | 30 | 320 | Thriller | extraction.jpg | 3 |
| 3 Idiots | Directed by Rajkumar Hirani | 11:00am | 180 | 50 | 250 | Comedy | 3idiots.jpg | 1 |
| The Parent Trap | Directed by Nancy Meyers | 3:30pm | 128 | 60 | 260 | Comedy | parenttrap.jpg | 2 |
| Home Alone | Directed by | 1:00pm | 55 | 88 | 400 | Comedy | h.jpg | 3 |
| Twilight | Directed by Catherine Hardwicke | 5:00pm | 126 | 50 | 200 | Romance | twilight.jpg | 1 |
| Little Women | Directed by Greta Gerwig | 11:00am | 100 | 20 | 350 | Romance | littlewomen.jpg | 2 |
| The Fault In Our Stars | Directed by Josh Boone | 2:00pm | 133 | 90 | 280 | Romance | stars.jpg | 3 |
| Phineas and Ferb | Directed by Robert Hughes | 4:30pm | 90 | 40 | 120 | Kids | pf.jpg | 1 |
| Frozen | Directed by Jennifer Michelle Lee | 5:00pm | 128 | 20 | 175 | Kids | frozen.jpg | 2 |
| Scooby Doo | Directed by Raja Gosnell | 3:40pm | 90 | 50 | 128 | Kids | scoob.jpg | 3 |
| Annabelle | Directed by John Robert | 9:00pm | 155 | 30 | 350 | Horror | annabelle.jpg | 1 |
| IT | Directed by Andres Muschietti | 4:00pm | 135 | 55 | 200 | Horror | it.jpg | 2 |
| Insidious The Last Key | Directed by Adam Robitel | 11:30pm | 103 | 40 | 290 | Horror | insidious.jpg | 3 |
+-----+-----+-----+-----+-----+-----+-----+-----+
21 rows in set (0.05 sec)
```

3. Booking_details

```
mysql> select * from booking_details;
+-----+-----+-----+-----+
| name | yob | mname | seats_booked |
+-----+-----+-----+-----+
| S | 2021 | Avatar | 3 |
| Payal | 2004 | Interstellar | 4 |
| Shiva | 2004 | Interstellar | 2 |
| Maitreyi | 1996 | Jab We Meet | 2 |
| Shiva | 2004 | Fast and Furious 8 | 4 |
| Anshi | 2003 | Interstellar | 3 |
| Shiva | 2004 | Interstellar | 3 |
| Anshi | 2003 | Interstellar | 4 |
| Payal | 2004 | Interstellar | 2 |
| Shiva | 2004 | Interstellar | 4 |
| S | 2004 | Avatar | 4 |
| S | 2004 | Gravity | 5 |
| S | 2004 | Avatar | 5 |
| Shiva | 2004 | Phineas and Ferb | 5 |
| Shiva | 2004 | Gravity | 5 |
| ABC | 2004 | Interstellar | 10 |
| Shiva | 2004 | Interstellar | 5 |
| Steve | 2021 | Fast and Furious 8 | 5 |
+-----+-----+-----+-----+
18 rows in set (0.04 sec)
```

4. Managers

```
mysql> select * from managers;
+-----+-----+
| username | password |
+-----+-----+
| CinePlus_Manager | ABC |
| Manager_CinePlus | SPA |
+-----+-----+
2 rows in set (0.03 sec)
```

5. Users

```
mysql> select * from users;
+-----+-----+
| username | password |
+-----+-----+
| bkshaklya | Bks%19690107 |
| H         | 1234      |
| M         | 112211    |
| M         | 1234      |
| mali      | 123       |
| mali      | sss       |
| P         | 1212      |
| S         | !@34      |
| S         | 1          |
| S         | 1234      |
| S         | 27884     |
| S         | 4732721   |
| S         | 891011    |
+-----+-----+
13 rows in set (0.00 sec)
```

CSV Files

Similar data has been stored in the csv files to allow for the setup of the program in different computers.

CODE

To ensure clarity and proper organization, 3 separate modules have been created for the program:

1. GUI_Interface
2. ViewerModule
3. ManagerModule

1. GUI_Interface

Code:

```
from tkinter import *
from tkinter import ttk
from tkinter import messagebox
import ViewerModule
import mysql.connector as con
mycon=con.connect(host="localhost",username="root",passwd="Shiva09@04",database="project1")
cur=mycon.cursor()
from PIL import Image,ImageTk
from tkinter.font import Font
import ManagerModule
#Sign In window
```

```
window1=Tk()
window1.title("Window 1")
window1.geometry("1300x800")
load=Image.open(r"C:\Users\MALTI
SHAKLYA\Desktop\Shiva\Python\Project\Bg.jpg")
render=ImageTk.PhotoImage(load)
img=Label(window1,image=render)
img.place(x=0,y=0)

'''width=window1.winfo_screenwidth()
height=window1.winfo_screenheight()

l=Label(window1,text=f"width:{width} height:{height}")
l.pack()'''

def click_B1():
    global pop1
    global tname
    global tpassword
    pop1=Toplevel(window1)
    pop1.title("Viewer Sign In")
    pop1["bg"]="White"

    lname=Label(pop1,text="Enter Username",font=("Copperplate
gothic",10),fg="black",bg="white")
    lname.place(x=50,y=25)
    tname=Entry(pop1,width=25)
    tname.place(x=200,y=25)

    lpassword=Label(pop1, text="Enter Password",font=("Copperplate
gothic",10),fg="black",bg="white")
    lpassword.place(x=50,y=55)
    tpassword=Entry(pop1,width=25,show="*")
    tpassword.place(x=200,y=55)
```

```
register=Button(pop1,text="Register",font=("Copperplate gothic",10),fg="black",bg="Orange",cursor="hand2",command=click_register)
register.place(x=100,y=90)

sign=Button(pop1,text="Sign In",font=("Copperplate gothic",10),fg="black",bg="light green",cursor="hand2",command=lambda:click_sign(B1))
sign.place(x=175,y=90)

back=Button(pop1,text="Back",font=("Copperplate gothic",10),fg="black",bg="red",cursor="hand2",command=click_back)
back.place(x=245,y=90)

pop1.geometry("550x200")
pop1.mainloop()

def click_B2():
    global pop1
    global tname
    global tpassword
    pop1=Toplevel(window1)
    pop1.title("Manager Sign In")
    pop1["bg"]="White"

    lname=Label(pop1,text="Enter Username",font=("Copperplate gothic",10),fg="black",bg="white")
    lname.place(x=50,y=25)
    tname=Entry(pop1,width=25)
    tname.place(x=200,y=25)

    lpassword=Label(pop1, text="Enter Password",font=("Copperplate gothic",10),fg="black",bg="white")
    lpassword.place(x=50,y=55)
    tpassword=Entry(pop1,width=25,show="*")
    tpassword.place(x=200,y=55)
```

```
    sign=Button(pop1,text="Sign In",font=("Copperplate gothic",10),fg="black",bg="light green",cursor="hand2",command=lambda:click_sign(B2))

    sign.place(x=100,y=90)

    change=Button(pop1,text="Change Password",font=("Copperplate gothic",10),fg="black",bg="Orange",cursor="hand2",command=lambda:change_password(tname.get()))

    change.place(x=168,y=90)

    back=Button(pop1,text="Back",font=("Copperplate gothic",10),fg="black",bg="red",cursor="hand2",command=click_back)

    back.place(x=300,y=90)

pop1.geometry("550x200")
pop1.mainloop()

def click_register():
    global l1alert
    username=tname.get()

    password=tpassword.get()

    if len(username)>0 and len(password)>0:
        cur.execute("use project1;")
        query2=("select username from users;")
        cur.execute(query2)
        data=cur.fetchall()
        try:
            if username not in data:
                query2=("insert into users (username,password) values (%s,%s);")
                data=(username,password)
```

```

        cur.execute(query2,data)
        mycon.commit()

        pop1.destroy()
        ViewerModule.open_window2()

    except:
        l1alert.destroy()
        l1alert=Label(pop1,text="username already exists, kindly
try another username",font=("Copperplate
gothic",10),bg="#3399FF",fg="white")
        l1alert.place(x=80,y=145)

    else:
        l1alert=Label(pop1,text="Check if data has been inserted
correctly and try again",font=("Copperplate gothic",10),bg="dark
red",fg="white")
        l1alert.place(x=80,y=145)

def click_sign(b):
    username=tname.get()
    password=tpassword.get()
    global l1alert
    if b==B1:
        if len(username)>0 and len(password)>0:
            cur.execute("use project1;")
            query2=("select username,password from users;")
            cur.execute(query2)
            data=cur.fetchall()

            if (username,password) in data:
                pop1.destroy()
                ViewerModule.open_window2()
            else:
                l1alert.destroy()

```

```
    l1alert=Label(pop1,text="Incorrect username or
password",font=("Copperplate gothic",10),bg="orange",fg="white")
    l1alert.place(x=80,y=145)

else:
    l1alert=Label(pop1,text="Check if data has been inserted
correctly and try again",font=("Copperplate gothic",10),bg="dark
red",fg="white")
    l1alert.place(x=80,y=145)

if b==B2:
    if len(username)>0 and len(password)>0:
        query2=("select username,password from managers;")
        cur.execute(query2)
        data=cur.fetchall()
        if (username,password) in data:
            pop1.destroy()
            ManagerModule.open_window3()

    else:
        l2alert=Label(pop1,text="Incorrect username or
password",font=("Copperplate gothic",10),bg="orange",fg="white")
        l2alert.place(x=80,y=145)

else:
    lalert=Label(pop1,text="Check if data has been inserted
correctly and try again",font=("Copperplate gothic",10),bg="dark
red",fg="white")
    lalert.place(x=80,y=145)

def click_back():
    pop1.destroy()

def new_password(user,recent,new):
    print(user)
```

```
query="select password from managers where
username='{}'".format(user)
cur.execute(query)
data=cur.fetchall()

for i in data:
    if list(i)==[recent]:
        query="update managers set password='{}' where
username='{}'".format(new,user)
        cur.execute(query)
        mycon.commit()

        messagebox.showinfo(title="Change
Password",message="Password changed Successfully")
        change.destroy()
        print("Password updated")

    else:
        change.destroy()
        messagebox.showerror(title="Change Password",
message="Password Incorrect")
        print("Incorrect Password")

def change_password(username):
    global change

    pop1.destroy()

    change=Toplevel(window1)
    change.title("Change Password")
    change["bg"]="White"
    change.geometry("550x200")
```

```
userL=Label(change,text="Enter current username: ",font=("Copperplate gothic",10),fg="black",bg="white")
userL.place(x=50,y=25)

userT=ttk.Entry(change,width=25)
userT.place(x=200,y=25)

recentL=Label(change,text="Enter current password: ",font=("Copperplate gothic",10),fg="black",bg="white")
recentL.place(x=50,y=50)

recentT=ttk.Entry(change,width=25)
recentT.place(x=200,y=50)

newL=Label(change,text="Enter new password: ",font=("Copperplate gothic",10),fg="black",bg="white")
newL.place(x=60,y=75)

newT=ttk.Entry(change,width=25)
newT.place(x=210,y=75)

C=Button(change,text="Enter",font=("Copperplate gothic",10),fg="black",
bg="blue",command=lambda:new_password(userT.get(),recentT.get(),newT.get()))
C.place(x=150,y=100)

def csv_write():
    import csv

    #Movie
    f=open("Movie_Details.csv","w",newline="")
```

```
w=csv.writer(f)
q=("select * from movie")
cur.execute(q)
d=cur.fetchall()
w.writerows(d)

#users
f=open("User_Details.csv", "w", newline="")
w=csv.writer(f)
q=("select * from users")
cur.execute(q)
d=cur.fetchall()
w.writerows(d)

#Managers
f=open("Manager_Details.csv", "w", newline="")
w=csv.writer(f)
q=("select * from managers")
cur.execute(q)
d=cur.fetchall()
w.writerows(d)
print("data inserted")

#Displayed movies
import csv
f=open("Displayed_movies.csv", "w", newline="")
w=csv.writer(f)
q=("select * from movies")
cur.execute(q)
d=cur.fetchall()
w.writerows(d)
print("data inserted")

#Sign In Text
```

```
f=Font(family="Craftsman PERSONAL USE",size="62")  
img=PhotoImage(file=r"C:\Users\MALTI  
SHAKLYA\Desktop\Shiva\Python\Project\T.png")  
L=Label(window1,image=img,fg="black")  
L.place(x=400,y=60)  
  
'''L2=Label(window1,text="OR",font=("Copperplate  
gothic",20,"bold"),fg="white",bg="black")  
L2.place(x=550,y=230)'''  
  
#Viewer Sign In  
img1=PhotoImage(file=r"C:\Users\MALTI  
SHAKLYA\Desktop\Shiva\Python\Project\viewer.png")  
B1=Button(window1,image=img1,  
bd=0,fg="black",activebackground="#000000",cursor="hand2",command=click_B1  
)  
B1.place(x=400,y=237)  
  
#Manager Sign In  
img2=PhotoImage(file=r"C:\Users\MALTI  
SHAKLYA\Desktop\Shiva\Python\Project\manager.png")  
B2=Button(window1,image=img2,bd=0,fg="#000000",activebackground="#000000",  
cursor="hand2",command=click_B2)  
B2.place(x=660,y=237)  
  
#csvfile_write()  
  
window1.mainloop()
```

2. ViewerModule

Code:

```
from tkinter import *  
  
from PIL import Image,ImageTk
```

```
from tkinter.font import Font

from tkinter import ttk

from tkinter import messagebox

import mysql.connector as con

mycon=con.connect(host="localhost",username="root",passwd="Shiva09@04",database="project1")

cur=mycon.cursor()

def click_book(N):

    global book

    pop2.destroy()

    book=Toplevel(window2)

    book.title("Enter Details")

    book['bg']="#00CCCC"

    book.geometry("500x400")

    nameL=Label(book,text="Enter Name",font=("Copperplate gothic",10),bg="#00CCCC",fg="black")

    nameL.place(x=100,y=20)

    name=ttk.Entry(book,width=20)

    name.place(x=180,y=20)

    dobL=Label(book,text="Select Year of Birth",font=("Copperplate gothic",10),bg="#00CCCC",fg="black")

    dobL.place(x=100,y=80)

    #creating birth year list

    year=[]

    for i in range(2021,1920,-1):

        year+=[i]

    #

    y=IntVar()

    dob=ttk.Combobox(book,textvariable=y)
```

```

dob['values']=year
dob.place(x=240,y=80)
dob.current(0)

seatsL=Label(book,text="No. of Seats to be Booked",font=("Copperplate
gothic",10),bg="#00CCCC",fg="black")
seatsL.place(x=100,y=140)

seats=ttk.Entry(book,width=5)
seats.place(x=260,y=140)

query="select seat_avail from displayed_movies where
mname='{}';".format(N)
cur.execute(query)
avail=cur.fetchall()
for i in avail:
    a=int(i[0])
    seat_avail=Label(book,text="Seats available:
{}".format(a),font=("Copperplate gothic",10),bg="#00CCCC",fg="black")
    seat_avail.place(x=290,y=140)

confirm=Button(book,text="Confirm Booking",font=("Copperplate
gothic",10),bg="dark
blue",fg="white",command=lambda:BookingDetails_toSQL(name,seats,y,N))
confirm.place(x=170,y=200)

book.mainloop()

def BookingDetails_toSQL(name,seats,y,N):
    #ADD DATE TIME STAMP
    try:
        query="create table booking_details (name varchar(25),yob
int,mname varchar(40),seats_booked int);"
        cur.execute(query)

```

```

        query="insert into booking_details
VALUES ('{}', {}, '{}', {})".format(name.get(), y.get(), N, seats.get())
        cur.execute(query)

    except:
        query="insert into booking_details
VALUES ('{}', {}, '{}', {})".format(name.get(), y.get(), N, seats.get())
        cur.execute(query)

    #

    query="select seat_avail from displayed_movies where
mname='{}';".format(N)
    cur.execute(query)
    avail=cur.fetchall()
    for i in avail:
        a=int(i[0])
    new_seats=a-int(seats.get())
    query1="update displayed_movies set seat_avail={} where
mname='{}';".format(new_seats,N)
    cur.execute(query1)
    query2="update all_movies set seat_avail={} where
mname='{}';".format(new_seats,N)
    cur.execute(query2)
    mycon.commit()

#CREATE INVOICE
book.destroy()
messagebox.showinfo(message="BOOKING CONFIRMED", parent=window2)

def details(N):
    global s
    g="\n"
    query="select
descp,cost_per_head,seat_avail,show_timing,duration_min,genre from
displayed_movies where mname='{}';".format(N)

```

```

        cur.execute(query)
        data=cur.fetchall()
        print(data)
        for i in data:
            i=list(i)
            s="Description: {}".format(i[0])+g+"Cost per seat:
{}".format(i[1])+g+"Seats Available: {}".format(i[2])+g+"Show Timings:
{}".format(i[3])+g+"Duration: {}".format(i[4])+"min"+g+"genre:
{}".format(i[5])
            print(s)
            descsp=Label(pop2,text=s,font=("Copperplate
gothic",15),bg="black",fg="white")
            descsp.place(x=100,y=50)

def open_pop2(N):
    global pop2
    pop2=Toplevel(window2)
    pop2.title("Book Movie")
    pop2['bg']="black"
    pop2.geometry("550x400")

    #Movie Title
    descsp=Label(pop2,text=N,font=("Algerian
Regular",20),bg="black",fg="white")
    descsp.place(x=200,y=0)

    details(N)

    #Book Button
    img=ImageTk.PhotoImage(Image.open("book.png"))

    book=Button(pop2,image=img,bg="black",activebackground="black",border=0,cu
rsor="hand2",command=lambda:click_book(N))
    book.place(x=200,y=250)

```

```
pop2.mainloop()

def destroy_window2():
    window2.destroy()
    open_window2()

def open_window2():
    global GB
    global GL
    global window2
    global genre

    window2=Toplevel()
    window2.title("Movie Booking Dashboard")
    window2.geometry("1300x800")
    load=Image.open(r"C:\Users\MALTI
SHAKLYA\Desktop\Shiva\Python\Project\H12.jpg")
    r=ImageTk.PhotoImage(load)
    img=Label(window2,image=r)
    img.place(x=0,y=0)

    #genre Combobox
    n=StringVar()
    genre=ttk.Combobox(window2,textvariable=n)

    genre['values']=['Action','Adventure','Thriller','Comedy','Romance','Horro
r','Kids special']
    genre.place(x=300,y=200)
    genre.current(0)

    GL=Label(window2,text="Choose a Genre")
    GL.place(x=200,y=200)
    GB=Button(window2,text="GO",cursor="hand2",command=lambda:choice(n))
    GB.place(x=500,y=200)
```

```
window2.mainloop()

def choice(n):
    global img1,img2,img3

    if n.get() == "Adventure":
        GB.destroy()
        GL.destroy()
        genre.destroy()

        load1=Image.open("ADVENTURE.jpg")
        r1=ImageTk.PhotoImage(load1)
        img12=Label(window2,image=r1)
        img12.place(x=0,y=0)

        imageB=PhotoImage(file="button.png")
        B=Button(window2,image=imageB, bd=0
,cursor="hand2",command=destroy_window2)
        B.place(x=580,y=540)

    #poster1
    query1="select * from displayed_movies where genre='Adventure'
and position=1;"
    cur.execute(query1)
    data=list(cur.fetchall())
    n1=data[0][0]
    location=data[0][-2]
    img1=ImageTk.PhotoImage(Image.open(location))

    p1=Button(window2,image=img1,cursor="hand2",command=lambda:open_pop2(n1))
    p1.place(x=290,y=180)
    mycon.commit()

    #poster2
```

```
query1="select * from displayed_movies where genre='Adventure'  
and position=2;"  
cur.execute(query1)  
data=list(cur.fetchall())  
n2=data[0][0]  
location=data[0][-2]  
img2=ImageTk.PhotoImage(Image.open(location))  
  
p2=Button(window2,image=img2,cursor="hand2",command=lambda:open_pop2(n2))  
p2.place(x=540,y=180)  
mycon.commit()  
  
#poster3  
query1="select * from displayed_movies where genre='Adventure'  
and position=3;"  
cur.execute(query1)  
data=list(cur.fetchall())  
n3=data[0][0]  
location=data[0][-2]  
img3=ImageTk.PhotoImage(Image.open(location))  
  
p3=Button(window2,image=img3,cursor="hand2",command=lambda:open_pop2(n3))  
p3.place(x=790,y=180)  
mycon.commit()  
  
elif n.get() == "Action":  
  
    loadAC=Image.open("ACTION.jpg")  
    rAC=ImageTk.PhotoImage(loadAC)  
    img12AC=Label(window2,image=rAC)  
    img12AC.place(x=0,y=0)  
  
    imageB=PhotoImage(file="button.png")  
    B=Button(window2,image=imageB, bd=0  
,cursor="hand2",command=destroy_window2)  
    B.place(x=580,y=540)
```

```
#poster1
query1="select * from displayed_movies where genre='Action' and
position=1;"
cur.execute(query1)
data=list(cur.fetchall())
n1=data[0][0]
location=data[0][-2]
img1=ImageTk.PhotoImage(Image.open(location))

p1=Button(window2,image=img1,cursor="hand2",command=lambda:open_pop2(n1))
p1.place(x=290,y=180)
mycon.commit()

#poster2
query1="select * from displayed_movies where genre='Action' and
position=2;"
cur.execute(query1)
data=list(cur.fetchall())
n2=data[0][0]
location=data[0][-2]
img2=ImageTk.PhotoImage(Image.open(location))

p2=Button(window2,image=img2,cursor="hand2",command=lambda:open_pop2(n2))
p2.place(x=540,y=180)
mycon.commit()

#poster3
query1="select * from displayed_movies where genre='Action' and
position=3;"
cur.execute(query1)
data=list(cur.fetchall())
n3=data[0][0]
location=data[0][-2]
img3=ImageTk.PhotoImage(Image.open(location))
```

```
p3=Button(window2,image=img3,cursor="hand2",command=lambda:open_pop2(n3))
p3.place(x=790,y=180)
mycon.commit()

elif n.get()=="Thriller":

    loadT=Image.open("THRILLER.jpg")
    rT=ImageTk.PhotoImage(loadT)
    imgT=Label(window2,image=rT)
    imgT.place(x=0,y=0)

    imageB=PhotoImage(file="button.png")
    B=Button(window2,image=imageB, bd=0
, cursor="hand2",command=destroy_window2)
    B.place(x=580,y=540)

    #poster1
    query1="select * from displayed_movies where genre='Thriller'
and position=1;"
    cur.execute(query1)
    data=list(cur.fetchall())
    n1=data[0][0]
    location=data[0][-2]
    img1=ImageTk.PhotoImage(Image.open(location))

p1=Button(window2,image=img1,cursor="hand2",command=lambda:open_pop2(n1))
p1.place(x=290,y=180)
mycon.commit()

#poster2
query1="select * from displayed_movies where genre='Thriller'
and position=2;"
cur.execute(query1)
data=list(cur.fetchall())
```

```
n2=data[0][0]
location=data[0][-2]
img2=ImageTk.PhotoImage(Image.open(location))

p2=Button(window2,image=img2,cursor="hand2",command=lambda:open_pop2(n2))
p2.place(x=540,y=180)
mycon.commit()

#poster3
query1="select * from displayed_movies where genre='Thriller' and position=3;"
cur.execute(query1)
data=list(cur.fetchall())
n3=data[0][0]
location=data[0][-2]
img3=ImageTk.PhotoImage(Image.open(location))

p3=Button(window2,image=img3,cursor="hand2",command=lambda:open_pop2(n3))
p3.place(x=790,y=180)
mycon.commit()

elif n.get() == "Comedy":

    loada=Image.open("COMEDY.jpg")
    ra=ImageTk.PhotoImage(loada)
    img125=Label(window2,image=ra)
    img125.place(x=0,y=0)

    imageB=PhotoImage(file="button.png")
    B=Button(window2,image=imageB, bd=0
    ,cursor="hand2",command=destroy_window2)
    B.place(x=580,y=540)

    #poster1
    query1="select * from displayed_movies where genre='Comedy' and position=1;"
```

```
        cur.execute(query1)
        data=list(cur.fetchall())
        n1=data[0][0]
        location=data[0][-2]
        img1=ImageTk.PhotoImage(Image.open(location))

p1=Button(window2,image=img1,cursor="hand2",command=lambda:open_pop2(n1))
p1.place(x=290,y=180)
mycon.commit()

#poster2
query1="select * from displayed_movies where genre='Comedy' and
position=2;"
        cur.execute(query1)
        data=list(cur.fetchall())
        n2=data[0][0]
        location=data[0][-2]
        img2=ImageTk.PhotoImage(Image.open(location))

p2=Button(window2,image=img2,cursor="hand2",command=lambda:open_pop2(n2))
p2.place(x=540,y=180)
mycon.commit()

#poster3
query1="select * from displayed_movies where genre='Comedy' and
position=3;"
        cur.execute(query1)
        data=list(cur.fetchall())
        n3=data[0][0]
        location=data[0][-2]
        img3=ImageTk.PhotoImage(Image.open(location))

p3=Button(window2,image=img3,cursor="hand2",command=lambda:open_pop2(n3))
p3.place(x=790,y=180)
mycon.commit()
```

```
elif n.get() == "Romance":  
    loadR=Image.open("ROMANCE.jpg")  
    rR=ImageTk.PhotoImage(loadR)  
    img126=Label(window2,image=rR)  
    img126.place(x=0,y=0)  
  
    imageB=PhotoImage(file="button.png")  
    B=Button(window2,image=imageB, bd=0  
,cursor="hand2",command=destroy_window2)  
    B.place(x=580,y=540)  
  
    #poster1  
    query1="select * from displayed_movies where genre='Romance' and  
position=1;"  
    cur.execute(query1)  
    data=list(cur.fetchall())  
    n1=data[0][0]  
    location=data[0][-2]  
    img1=ImageTk.PhotoImage(Image.open(location))  
  
    p1=Button(window2,image=img1,cursor="hand2",command=lambda:open_pop2(n1))  
    p1.place(x=290,y=180)  
    mycon.commit()  
  
    #poster2  
    query1="select * from displayed_movies where genre='Romance' and  
position=2;"  
    cur.execute(query1)  
    data=list(cur.fetchall())  
    n2=data[0][0]  
    location=data[0][-2]  
    img2=ImageTk.PhotoImage(Image.open(location))  
  
    p2=Button(window2,image=img2,cursor="hand2",command=lambda:open_pop2(n2))  
    p2.place(x=540,y=180)  
    mycon.commit()
```

```

#poster3
query1="select * from displayed_movies where genre='Romance' and
position=3;"
cur.execute(query1)
data=list(cur.fetchall())
n3=data[0][0]
location=data[0][-2]
img3=ImageTk.PhotoImage(Image.open(location))

p3=Button(window2,image=img3,cursor="hand2",command=lambda:open_pop2(n3))
p3.place(x=790,y=180)
mycon.commit()

elif n.get() == "Kids special":
    loadK=Image.open("KIDS.jpg")
    rK=ImageTk.PhotoImage(loadK)
    img124=Label(window2,image=rK)
    img124.place(x=0,y=0)

    imageB=PhotoImage(file="button.png")
    B=Button(window2,image=imageB, bd=0
,cursor="hand2",command=destroy_window2)
    B.place(x=580,y=540)

#poster1
query1="select * from displayed_movies where genre='Kids' and
position=1;"
cur.execute(query1)
data=list(cur.fetchall())
n1=data[0][0]
location=data[0][-2]
img1=ImageTk.PhotoImage(Image.open(location))

p1=Button(window2,image=img1,cursor="hand2",command=lambda:open_pop2(n1))
p1.place(x=290,y=180)

```

```
mycon.commit()

#poster2
query1="select * from displayed_movies where genre='Kids' and
position=2;"
cur.execute(query1)
data=list(cur.fetchall())
n2=data[0][0]
location=data[0][-2]
img2=ImageTk.PhotoImage(Image.open(location))

p2=Button(window2,image=img2,cursor="hand2",command=lambda:open_pop2(n2))
p2.place(x=540,y=180)
mycon.commit()

#poster3
query1="select * from displayed_movies where genre='Kids' and
position=3;"
cur.execute(query1)
data=list(cur.fetchall())
n3=data[0][0]
location=data[0][-2]
img3=ImageTk.PhotoImage(Image.open(location))

p3=Button(window2,image=img3,cursor="hand2",command=lambda:open_pop2(n3))
p3.place(x=790,y=180)
mycon.commit()

elif n.get() == "Horror":
    loadh=Image.open("HORROR.jpg")
    rh=ImageTk.PhotoImage(loadh)
    img123=Label(window2,image=rh)
    img123.place(x=0,y=0)

    imageB=PhotoImage(file="button.png")
```

```
B=Button(window2,image=imageB, bd=0  
,cursor="hand2",command=destroy_window2)  
B.place(x=580,y=540)  
  
#poster1  
query1="select * from displayed_movies where genre='Horror' and  
position=1;"  
cur.execute(query1)  
data=list(cur.fetchall())  
n1=data[0][0]  
location=data[0][-2]  
img1=ImageTk.PhotoImage(Image.open(location))  
  
p1=Button(window2,image=img1,cursor="hand2",command=lambda:open_pop2(n1))  
p1.place(x=290,y=180)  
mycon.commit()  
  
#poster2  
query1="select * from displayed_movies where genre='Horror' and  
position=2;"  
cur.execute(query1)  
data=list(cur.fetchall())  
n2=data[0][0]  
location=data[0][-2]  
img2=ImageTk.PhotoImage(Image.open(location))  
  
p2=Button(window2,image=img2,cursor="hand2",command=lambda:open_pop2(n2))  
p2.place(x=540,y=180)  
mycon.commit()  
  
#poster3  
query1="select * from displayed_movies where genre='Horror' and  
position=3;"  
cur.execute(query1)  
data=list(cur.fetchall())  
n3=data[0][0]
```

```

        location=data[0][-2]
        img3=ImageTk.PhotoImage(Image.open(location))

p3=Button(window2,image=img3,cursor="hand2",command=lambda:open_pop2(n3))
    p3.place(x=790,y=180)
    mycon.commit()

window2.mainloop()

```

3. ManagerModule

Code:

```

from tkinter import *

from tkinter import ttk

from tkinter import messagebox

import mysql.connector as con
mycon=con.connect(host="localhost",username="root",passwd="Shiva09@04",database="project1")
cur=mycon.cursor()

from PIL import ImageTk,Image

def open_window3():
    global window3
    window3=Toplevel()
    window3.title("Movie Booking Dashboard-Manager")
    window3.geometry("1300x800")

    #Background
    Back=ImageTk.PhotoImage(Image.open(r"AD.jpg"))
    img_B=Label(window3,image=Back,border=0)
    img_B.place(x=0,y=0)

```

```
#Sign In Button
img1=PhotoImage(file=r"Booking.png")
B1=Button(window3,image=img1, bd=0 , bg="#000000" ,
activebackground="#000000",command=booking_details)
B1.place(x=535,y=175)

img2=PhotoImage(file=r"update.png")
B2=Button(window3,image=img2, bd=0 , bg="#000000" ,
activebackground="#000000",command=update_poster)
B2.place(x=550,y=275)

img3=PhotoImage(file=r"db.png")
B3=Button(window3,image=img3, bd=0, bg="#000000",
activebackground="#000000",command=database_change)
B3.place(x=535,y=375)

window3.mainloop()

def booking_details():
    booking=Toplevel()
    booking.title("Booking Details")
    booking.geometry("790x320")
    booking['bg']="black"

    query="select * from booking_details"
    cur.execute(query)
    data=cur.fetchall()
    mycon.commit()
    #records=''
    #for record in data:
    #    records+=str(list(record))+"\n"
    #print_records=Label(booking,text=records,bg="black",fg="white")
    #print_records.place(x=20,y=20)

    rec_display=ttk.Treeview(booking,height=15)
```

```
rec_display["columns"]=("name","yob","mname","seats")
rec_display.column("#0",width=120)
rec_display.column("name",width=190)
rec_display.column("yob",width=200)
rec_display.column("mname",width=160)
rec_display.column("seats",width=120)

rec_display.heading("#0",text="Sno.")
rec_display.heading("name",text="Customer Name")
rec_display.heading("yob",text="Year Of Birth")
rec_display.heading("mname",text="Movie Name")
rec_display.heading("seats",text="Seats Booked")

print(len(data))

scrollbar=ttk.Scrollbar(booking,orient=VERTICAL,command=rec_display.yview)
rec_display.configure(yscroll=scrollbar.set)
scrollbar.place(x=1260,y=0)

for i in range(0,len(data)):

rec_display.insert(parent='',index='end',iid=i,text=i+1,values=data[i])

rec_display.place(x=0,y=0)

A=Label(booking,text="SELECT AND
SCROLL...",bg="black",fg="white",font=(90))
A.place(x=500,y=510)

def update_poster():
    global poster,n
    global GB,GL,genre
    poster=Toplevel()
    poster.title(r"Add\Delete\Update\\Poster")
```

```
poster.geometry("1300x800")
poster['bg']="black"

load=Image.open(r"HOME.jpg")
r=ImageTk.PhotoImage(load)
img=Label(poster,image=r)
img.place(x=0,y=0)

n=StringVar()
genre=ttk.Combobox(poster,textvariable=n)

genre['values']=['Action','Adventure','Thriller','Comedy','Romance','Horror','Kids special']
genre.place(x=300,y=200)
genre.current(0)

GL=Label(poster,text="Choose a Genre")
GL.place(x=200,y=200)

GB=Button(poster,text="GO",cursor="hand2",command=lambda:show_poster(n))
GB.place(x=500,y=200)

poster.mainloop()

def show_poster(n):
    global img1,img2,img3

    if n.get()=='Adventure':
        GB.destroy()
        GL.destroy()
        genre.destroy()

        load1=Image.open("ADVENTURE.jpg")
        r1=ImageTk.PhotoImage(load1)
        img12=Label(poster,image=r1)
```

```
    img12.place(x=0,y=0)

    imageB=PhotoImage(file="button.png")
    B=Button(poster,image=imageB, bd=0
, cursor="hand2",command=destroy_poster)
    B.place(x=580,y=540)

    #poster1
    query1="select * from displayed_movies where genre='Adventure'
and position=1;"
    c=1
    cur.execute(query1)
    data=list(cur.fetchall())
    n1=data[0][0]
    location=data[0][-2]
    img1=ImageTk.PhotoImage(Image.open(location))

p1=Button(poster,image=img1,cursor="hand2",command=lambda:open_poster(n1,1
))
    p1.place(x=290,y=180)
    mycon.commit()

    #poster2
    query1="select * from displayed_movies where genre='Adventure'
and position=2;"
    c=2
    cur.execute(query1)
    data=list(cur.fetchall())
    n2=data[0][0]
    location=data[0][-2]
    img2=ImageTk.PhotoImage(Image.open(location))

p2=Button(poster,image=img2,cursor="hand2",command=lambda:open_poster(n2,2
))
    p2.place(x=540,y=180)
```

```

mycon.commit()

#poster3
query1="select * from displayed_movies where genre='Adventure' and position=3;"
c=3
cur.execute(query1)
data=list(cur.fetchall())
n3=data[0][0]
location=data[0][-2]
img3=ImageTk.PhotoImage(Image.open(location))

p3=Button(poster,image=img3,cursor="hand2",command=lambda:open_poster(n3,3))
p3.place(x=790,y=180)
mycon.commit()

elif n.get() == "Action":

    loadAC=Image.open("ACTION.jpg")
    rAC=ImageTk.PhotoImage(loadAC)
    img12AC=Label(poster,image=rAC)
    img12AC.place(x=0,y=0)

    imageB=PhotoImage(file="button.png")
    B=Button(poster,image=imageB, bd=0
    ,cursor="hand2",command=destroy_poster)
    B.place(x=580,y=540)

    #poster1
    query1="select * from displayed_movies where genre='Action' and position=1;"
    c=1
    cur.execute(query1)
    data=list(cur.fetchall())
    n1=data[0][0]

```

```
location=data[0][-2]
img1=ImageTk.PhotoImage(Image.open(location))

p1=Button(poster,image=img1,cursor="hand2",command=lambda:open_poster(n1,1))
)
p1.place(x=290,y=180)
mycon.commit()

#poster2
query1="select * from displayed_movies where genre='Action' and
position=2;"
c=2
cur.execute(query1)
data=list(cur.fetchall())
n2=data[0][0]
location=data[0][-2]
img2=ImageTk.PhotoImage(Image.open(location))

p2=Button(poster,image=img2,cursor="hand2",command=lambda:open_poster(n2,2))
)
p2.place(x=540,y=180)
mycon.commit()

#poster3
query1="select * from displayed_movies where genre='Action' and
position=3;"
c=3
cur.execute(query1)
data=list(cur.fetchall())
n3=data[0][0]
location=data[0][-2]
img3=ImageTk.PhotoImage(Image.open(location))

p3=Button(poster,image=img3,cursor="hand2",command=lambda:open_poster(n3,3))
)
p3.place(x=790,y=180)
```

```
mycon.commit()

elif n.get() == "Thriller":

    loadT=Image.open("THRILLER.jpg")
    rT=ImageTk.PhotoImage(loadT)
    imgT=Label(poster,image=rT)
    imgT.place(x=0,y=0)

    imageB=PhotoImage(file="button.png")
    B=Button(poster,image=imageB, bd=0
    ,cursor="hand2",command=destroy_poster)
    B.place(x=580,y=540)

    #poster1
    query1="select * from displayed_movies where genre='Thriller'
and position=1;"

    cur.execute(query1)
    data=list(cur.fetchall())
    n1=data[0][0]
    location=data[0][-2]
    img1=ImageTk.PhotoImage(Image.open(location))

p1=Button(poster,image=img1,cursor="hand2",command=lambda:open_poster(n1,1
))
p1.place(x=290,y=180)
mycon.commit()

#poster2
query1="select * from displayed_movies where genre='Thriller'
and position=2;"
c=2
cur.execute(query1)
data=list(cur.fetchall())
```

```
n2=data[0][0]
location=data[0][-2]
img2=ImageTk.PhotoImage(Image.open(location))

p2=Button(poster,image=img2,cursor="hand2",command=lambda:open_poster(n2,2))
p2.place(x=540,y=180)
mycon.commit()

#poster3
query1="select * from displayed_movies where genre='Thriller' and position=3;"
c=3
cur.execute(query1)
data=list(cur.fetchall())
n3=data[0][0]
location=data[0][-2]
img3=ImageTk.PhotoImage(Image.open(location))

p3=Button(poster,image=img3,cursor="hand2",command=lambda:open_poster(n3,3))
p3.place(x=790,y=180)
mycon.commit()

elif n.get() == "Comedy":

    loada=Image.open("COMEDY.jpg")
    ra=ImageTk.PhotoImage(loada)
    img125=Label(poster,image=ra)
    img125.place(x=0,y=0)

    imageB=PhotoImage(file="button.png")
    B=Button(poster,image=imageB, bd=0
    ,cursor="hand2",command=destroy_poster)
    B.place(x=580,y=540)
```

```
#poster1
query1="select * from displayed_movies where genre='Comedy' and
position=1;"
c=1
cur.execute(query1)
data=list(cur.fetchall())
n1=data[0][0]
location=data[0][-2]
img1=ImageTk.PhotoImage(Image.open(location))

p1=Button(poster,image=img1,cursor="hand2",command=lambda:open_poster(n1,1
))
p1.place(x=290,y=180)
mycon.commit()

#poster2
query1="select * from displayed_movies where genre='Comedy' and
position=2;"
c=2
cur.execute(query1)
data=list(cur.fetchall())
n2=data[0][0]
location=data[0][-2]
img2=ImageTk.PhotoImage(Image.open(location))

p2=Button(poster,image=img2,cursor="hand2",command=lambda:open_poster(n2,2
))
p2.place(x=540,y=180)
mycon.commit()

#poster3
query1="select * from displayed_movies where genre='Comedy' and
position=3;"
c=3
cur.execute(query1)
data=list(cur.fetchall())
```

```
n3=data[0][0]
location=data[0][-2]
img3=ImageTk.PhotoImage(Image.open(location))

p3=Button(poster,image=img3,cursor="hand2",command=lambda:open_poster(n3,3))
p3.place(x=790,y=180)
mycon.commit()

elif n.get()=="Romance":
    loadR=Image.open("ROMANCE.jpg")
    rR=ImageTk.PhotoImage(loadR)
    img126=Label(poster,image=rR)
    img126.place(x=0,y=0)

    imageB=PhotoImage(file="button.png")
    B=Button(poster,image=imageB, bd=0
    ,cursor="hand2",command=destroy_poster)
    B.place(x=580,y=540)

#poster1
query1="select * from displayed_movies where genre='Romance' and
position=1;"
c=1
cur.execute(query1)
data=list(cur.fetchall())
n1=data[0][0]
location=data[0][-2]
img1=ImageTk.PhotoImage(Image.open(location))

p1=Button(poster,image=img1,cursor="hand2",command=lambda:open_poster(n1,1))
p1.place(x=290,y=180)
mycon.commit()

#poster2
```

```

        query1="select * from displayed_movies where genre='Romance' and
position=2;"

c=2

cur.execute(query1)

data=list(cur.fetchall())

n2=data[0][0]

location=data[0][-2]

img2=ImageTk.PhotoImage(Image.open(location))

p2=Button(poster,image=img2,cursor="hand2",command=lambda:open_poster(n2,2
))

p2.place(x=540,y=180)

mycon.commit()

#poster3

query1="select * from displayed_movies where genre='Romance' and
position=3;"

c=3

cur.execute(query1)

data=list(cur.fetchall())

n3=data[0][0]

location=data[0][-2]

img3=ImageTk.PhotoImage(Image.open(location))

p3=Button(poster,image=img3,cursor="hand2",command=lambda:open_poster(n3,3
))

p3.place(x=790,y=180)

mycon.commit()

elif n.get()=='Kids special':

    loadK=Image.open("KIDS.jpg")

    rK=ImageTk.PhotoImage(loadK)

    img124=Label(poster,image=rK)

    img124.place(x=0,y=0)

imageB=PhotoImage(file="button.png")

```

```
B=Button(poster,image=imageB, bd=0
,cursor="hand2",command=destroy_poster)
B.place(x=580,y=540)

#poster1
query1="select * from displayed_movies where genre='Kids' and
position=1;"
c=1
cur.execute(query1)
data=list(cur.fetchall())
n1=data[0][0]
location=data[0][-2]
img1=ImageTk.PhotoImage(Image.open(location))

p1=Button(poster,image=img1,cursor="hand2",command=lambda:open_poster(n1,1
))
p1.place(x=290,y=180)
mycon.commit()

#poster2
query1="select * from displayed_movies where genre='Kids' and
position=2;"
c=2
cur.execute(query1)
data=list(cur.fetchall())
n2=data[0][0]
location=data[0][-2]
img2=ImageTk.PhotoImage(Image.open(location))

p2=Button(poster,image=img2,cursor="hand2",command=lambda:open_poster(n2,2
))
p2.place(x=540,y=180)
mycon.commit()

#poster3
```

```
query1="select * from displayed_movies where genre='Kids' and
position=3;"

c=3

cur.execute(query1)
data=list(cur.fetchall())
n3=data[0][0]
location=data[0][-2]
img3=ImageTk.PhotoImage(Image.open(location))

p3=Button(poster,image=img3,cursor="hand2",command=lambda:open_poster(n3,3
))
p3.place(x=790,y=180)

mycon.commit()

elif n.get() == "Horror":
    loadh=Image.open("HORROR.jpg")
    rh=ImageTk.PhotoImage(loadh)
    img123=Label(poster,image=rh)
    img123.place(x=0,y=0)

    imageB=PhotoImage(file="button.png")
    B=Button(poster,image=imageB, bd=0
,cursor="hand2",command=destroy_poster)
    B.place(x=580,y=540)

#poster1
query1="select * from displayed_movies where genre='Horror' and
position=1;"

c=1

cur.execute(query1)
data=list(cur.fetchall())
n1=data[0][0]
location=data[0][-2]
img1=ImageTk.PhotoImage(Image.open(location))
```

```
p1=Button(poster,image=img1,cursor="hand2",command=lambda:open_poster(n1,1))
    p1.place(x=290,y=180)
    mycon.commit()

#poster2
query1="select * from displayed_movies where genre='Horror' and
position=2;"
c=2
cur.execute(query1)
data=list(cur.fetchall())
n2=data[0][0]
location=data[0][-2]
img2=ImageTk.PhotoImage(Image.open(location))

p2=Button(poster,image=img2,cursor="hand2",command=lambda:open_poster(n2,2))
    p2.place(x=540,y=180)
    mycon.commit()

#poster3
query1="select * from displayed_movies where genre='Horror' and
position=3;"
c=3
cur.execute(query1)
data=list(cur.fetchall())
n3=data[0][0]
location=data[0][-2]
img3=ImageTk.PhotoImage(Image.open(location))

p3=Button(poster,image=img3,cursor="hand2",command=lambda:open_poster(n3,3))
    p3.place(x=790,y=180)
    mycon.commit()
```

```

    poster.mainloop()

def open_poster(N,c):
    global update
    update=Toplevel()
    update.title("Booking Details")
    update['bg']="#003399"
    update.geometry("500x400")

    mnameL=Label(update,text="Enter New Movie Name",font=("Copperplate
gothic",10),bg="#003399",fg="white")
    mnameL.place(x=100,y=20)
    mnameT=ttk.Entry(update,width=20)
    mnameT.place(x=200,y=20)

    Enter=Button(update,text="Enter",font=("Copperplate
gothic",10),bg="#003399",fg="white",command=lambda:click_enter(mnameT,c))
    Enter.place(x=170,y=40)

    update.mainloop()

def click_enter(mnameT,c):
    try:
        query1="select * from all_movies where
mname='{}';".format(mnameT.get())
        cur.execute(query1)
        i=cur.fetchone()
        print(i)
        mname=i[0]
        desc=i[1]
        show_timing=i[2]
        duration_min=i[3]
        seat_avail=i[4]
        cost_per_head=i[5]
        genre=i[6]
    
```

```

location=i[7]
position=c

query2="""update displayed_movies set mname='{}',
descp='{}',
show_timing='{}',
duration_min={},
seat_avail={},
cost_per_head={},
location='{}'
where genre='{}' and
position={}""".format(mname,descp,show_timing,duration_min,seat_avail,cost
_per_head,location,genre,position)
cur.execute(query2)
mycon.commit()
update.destroy()
messagebox.showinfo(message="Poster Updated",parent=window3)
destroy_poster()

except:
    messagebox.showerror(message="Check whether the movie exists in
the data base and try again",parent=database)

def database_change():
    global img1,img2,img3,img4
    global database

    database=Toplevel(window3)
    database.title("Database Changes")

    database.geometry("1300x800")

    load=Image.open("DC.jpg")
    r=ImageTk.PhotoImage(load)
    A=Label(database,image=r,border=0)

```

```
A.place(x=0,y=0)

img1=PhotoImage(file="view.png")
B1=Button(database,image=img1, bd=0 , bg="#000000" ,
activebackground="#000000",command=show_movies)
B1.place(x=535,y=175)

img2=PhotoImage(file="change.png")
B2=Button(database,image=img2, bd=0 , bg="#000000" ,
activebackground="#000000",command=change_details)
B2.place(x=535,y=275)

img3=PhotoImage(file="add.png")
B3=Button(database,image=img3, bd=0 , bg="#000000" ,
activebackground="#000000",command=add_new)
B3.place(x=535,y=375)

img4=PhotoImage(file="delete.png")
B4=Button(database,image=img4, bd=0 , bg="#000000" ,
activebackground="#000000",command=delete_movie)
B4.place(x=535,y=475)

database.mainloop()

def show_movies():

    all_movies=Toplevel()
    all_movies.title("Booking Details")
    all_movies.geometry("500x400")
    all_movies['bg']="black"

    query="select * from all_movies;"
    cur.execute(query)
    data=cur.fetchall()
```

```

mycon.commit()

#records=''

#for record in data:
    #records+=str(list(record))+"\n"

#print_records=Label(all_movies,text=records,bg="black",fg="white")
#print_records.place(x=20,y=20)

rec_display=ttk.Treeview(all_movies,height=20)

rec_display["columns"]=("mname","description","show_timing","duration_min"
,"seat_avail","cost_per_head","genre","location")

rec_display.column("#0",width=120)
rec_display.column("mname",width=190)
rec_display.column("description",width=200)
rec_display.column("show_timing",width=160)
rec_display.column("duration_min",width=120)
rec_display.column("seat_avail",width=120)
rec_display.column("cost_per_head",width=120)
rec_display.column("genre",width=120)
rec_display.column("location",width=110)

rec_display.heading("#0",text="Sno.")
rec_display.heading("mname",text="Movie Name")
rec_display.heading("description",text="Description")
rec_display.heading("show_timing",text="Show Timing")
rec_display.heading("duration_min",text="Duration(in min)")
rec_display.heading("seat_avail",text="Seats Available")
rec_display.heading("cost_per_head",text="Cost Per Seat")
rec_display.heading("genre",text="Genre")
rec_display.heading("location",text="Poster Location")
print(len(data))

scrollbar=ttk.Scrollbar(all_movies,orient=VERTICAL,command=rec_display.yview)

rec_display.configure(yscroll=scrollbar.set)

```

```
scrollbar.place(x=1260,y=0)

for i in range(0,len(data)):

    rec_display.insert(parent='',index='end',iid=i,text=i+1,values=data[i])

    rec_display.place(x=0,y=0)

    A=Label(all_movies,text="SELECT AND
SCROLL...",bg="black",fg="white",font=(90))
    A.place(x=500,y=510)

    all_movies.mainloop()

def change_details():
    global cd

    cd=Toplevel()
    cd.title("Booking Details")
    cd['bg']="black"
    cd.geometry("500x400")

    img11=PhotoImage(file="seat.png")
    B1=Button(cd,image=img11, bd=0 , bg="#000000" ,
activebackground="#000000",command=change_seats)
    B1.place(x=90,y=75)

    img21=PhotoImage(file="st.png")
    B2=Button(cd,image=img21, bd=0 , bg="#000000" ,
activebackground="#000000",command=change_showtiming)
    B2.place(x=90,y=175)

    img31=PhotoImage(file="cost.png")
    B3=Button(cd,image=img31, bd=0, bg="#000000",
activebackground="#000000",command=change_cost)
```

```

B3.place(x=90,y=275)

cd.mainloop()

def change_seats():
    cd.destroy()

    n1=Toplevel()
    n1.title("Booking Details")
    n1['bg']="black"
    n1.geometry("500x400")

    mnameL=Label(n1,text="Enter New Movie Name",font=("Copperplate gothic",10),bg="#00CCCC",fg="black")
    mnameL.place(x=100,y=20)
    mnameT=ttk.Entry(n1,width=20)
    mnameT.place(x=300,y=20)

    seatsL=Label(n1,text="Enter Seats Available",font=("Copperplate gothic",10),bg="#00CCCC",fg="black")
    seatsL.place(x=100,y=40)
    seatsT=ttk.Entry(n1,width=20)
    seatsT.place(x=300,y=40)

    img=PhotoImage(file="ENTER.png")
    e=Button(n1,image=img,bd=0,
    bg="#000000",activebackground="#000000",command=lambda:up1(seatsT,mnameT))
    e.place(x=190,y=80)

    n1.mainloop()

def up1(seatsT,mnameT):

    query1="UPDATE displayed_movies set seat_avail={} where
mname='{}';".format(seatsT.get(),mnameT.get())
    cur.execute(query1)

```

```
query2="UPDATE all_movies set seat_avail={} where
mname='{}';".format(seatsT.get(),mnameT.get())
cur.execute(query2)
mycon.commit()

messagebox.showinfo(title=None,message="DATABASE SUCCESSFULLY
UPDATED",parent=database)
n1.destroy()

def change_showtiming():
    global n2
    cd.destroy()

    n2=Toplevel()
    n2.title("Booking Details")
    n2['bg']="black"
    n2.geometry("500x400")

    mnameL=Label(n2,text="Enter New Movie Name",font=("Copperplate
gothic",10),bg="#00CCCC",fg="black")
    mnameL.place(x=100,y=20)
    mnameT=ttk.Entry(n2,width=20)
    mnameT.place(x=300,y=20)

    show_timingL=Label(n2,text="Enter new show
timings",font=("Copperplate gothic",10),bg="#00CCCC",fg="black")
    show_timingL.place(x=100,y=40)
    show_timingT=ttk.Entry(n2,width=20)
    show_timingT.place(x=300,y=40)

    img=PhotoImage(file="ENTER.png")
    e=Button(n2,image=img,bd=0,
    bg="#000000",activebackground="#000000",command=lambda:up2(show_timingT,mn
ameT))
    e.place(x=190,y=80)
```

```

n2.mainloop()

def up2(show_timingT,mnameT):
    query1="UPDATE displayed_movies set show_timing='{}' where
mname='{}';".format(show_timingT.get(),mnameT.get())
    cur.execute(query1)
    query2="UPDATE all_movies set show_timing='{}' where
mname='{}';".format(show_timingT.get(),mnameT.get())
    cur.execute(query2)
    mycon.commit()
    messagebox.showinfo(title=None,message="DATABASE SUCCESSFULLY
UPDATED",parent=database)
    n2.destroy()

def change_cost():
    cd.destroy()
    global n3

    n3=Toplevel()
    n3.title("Booking Details")
    n3['bg']="black"
    n3.geometry("500x400")

    mnameL=Label(n3,text="Enter New Movie Name",font=("Copperplate
gothic",10),bg="#00CCCC",fg="black")
    mnameL.place(x=100,y=20)
    mnameT=ttk.Entry(n3,width=20)
    mnameT.place(x=300,y=20)

    costL=Label(n3,text="Enter cost per seat",font=("Copperplate
gothic",10),bg="#00CCCC",fg="black")
    costL.place(x=100,y=40)
    costT=ttk.Entry(n3,width=20)

```

```

costT.place(x=300,y=40)

img=PhotoImage(file="ENTER.png")
e=Button(n3,image=img,bd=0,
bg="#000000",activebackground="#000000",command=lambda:up3(costT,mnameT))
e.place(x=190,y=80)

n3.mainloop()

def up3(costT,mnameT):

    query1="UPDATE displayed_movies set cost_per_head={} where
mname='{}';".format(costT.get(),mnameT.get())
    cur.execute(query1)
    query2="UPDATE all_movies set cost_per_head={} where
mname='{}';".format(costT.get(),mnameT.get())
    cur.execute(query2)
    mycon.commit()
    messagebox.showinfo(title=None,message="DATABASE SUCCESSFULLY
UPDATED",parent=database)
    n3.destroy()

def add_new():

    global new
    new=Toplevel()
    new.title("Booking Details")
    new['bg']="#003399"
    new.geometry("500x400")

    mnameL=Label(new,text="Enter New Movie Name",font=("Copperplate
gothic",10),bg="#003399",fg="white")
    mnameL.place(x=100,y=20)
    mnameT=ttk.Entry(new,width=20)
    mnameT.place(x=300,y=20)

```

```
descpL=Label(new,text="Enter Description",font=("Copperplate gothic",10),bg="#003399",fg="white")
descpL.place(x=100,y=40)
descpT=ttk.Entry(new,width=20)
descpT.place(x=300,y=40)

show_timingL=Label(new,text="Enter Show Time",font=("Copperplate gothic",10),bg="#003399",fg="white")
show_timingL.place(x=100,y=60)
show_timingT=ttk.Entry(new,width=20)
show_timingT.place(x=300,y=60)

durationL=Label(new,text="Enter Duration in Minutes",font=("Copperplate gothic",10),bg="#003399",fg="white")
durationL.place(x=100,y=80)
durationT=ttk.Entry(new,width=10)
durationT.place(x=300,y=80)

seat_availL=Label(new,text="Enter Seats Available",font=("Copperplate gothic",10),bg="#003399",fg="white")
seat_availL.place(x=100,y=100)
seat_availT=ttk.Entry(new,width=20)
seat_availT.place(x=300,y=100)

cost_per_headL=Label(new,text="Enter Cost Per Seat",font=("Copperplate gothic",10),bg="#003399",fg="white")
cost_per_headL.place(x=100,y=120)
cost_per_headT=ttk.Entry(new,width=20)
cost_per_headT.place(x=300,y=120)

genreL=Label(new,text="Enter Genre",font=("Copperplate gothic",10),bg="#003399",fg="white")
genreL.place(x=100,y=140)
genreT=ttk.Entry(new,width=20)
genreT.place(x=300,y=140)
```

```

        locationL=Label(new,text="Enter Location of movie
poster",font=("Copperplate gothic",10),bg="#003399",fg="white")
        locationL.place(x=100,y=160)
        locationT=ttk.Entry(new,width=20)
        locationT.place(x=300,y=160)

        img=PhotoImage(file="ENTER.png")
        Enter=Button(new,image=img, bd=0, bg="#003399",
activebackground="#003399",command=lambda:enter_movie(mnameT,descpT,show_t
imingT,durationT,seat_availT,cost_per_headT,genreT,locationT))
        Enter.place(x=170,y=220)

        new.mainloop()

def
enter_movie(mnameT,descpT,show_timingT,durationT,seat_availT,cost_per_head
T,genreT,locationT):
    query="insert into all_movies values ('{}', '{}', '{}', {}, {},
{},'{}', '{}')".format(mnameT.get(), descpT.get(), show_timingT.get(),
durationT.get(), seat_availT.get(),
cost_per_headT.get(),genreT.get(),locationT.get())
    cur.execute(query)
    mycon.commit()

    messagebox.showinfo(title=None,message="MOVIE SUCCESSFULLY
ADDED",parent=database)
    new.destroy()

def delete_movie():
    global update
    update=Toplevel()
    update.title("Delete Movie Record")
    update['bg']="#003399"
    update.geometry("500x400")

```

```
mnameL=Label(update,text="Enter Movie To Be  
Deleted",font=("Copperplate gothic",10),bg="#003399",fg="white")  
mnameL.place(x=100,y=20)  
mnameT=ttk.Entry(update,width=20)  
mnameT.place(x=270,y=20)  
  
img=PhotoImage(file="ENTER.png")  
Enter=Button(update,image=img,bd=0,  
bg="#003399",activebackground="#003399",command=lambda:enter_delete(mnameT  
))  
Enter.place(x=190,y=60)  
  
update.mainloop()  
  
def enter_delete(mnameT):  
    query="DELETE from all_movies where mname='{}'".format(mnameT.get())  
    cur.execute(query)  
    mycon.commit()  
  
    messagebox.showinfo(title=None,message="MOVIE SUCCESSFULLY  
DELETED",parent=database)  
    update.destroy()  
  
def destroy_poster():  
    poster.destroy()  
    update_poster()
```

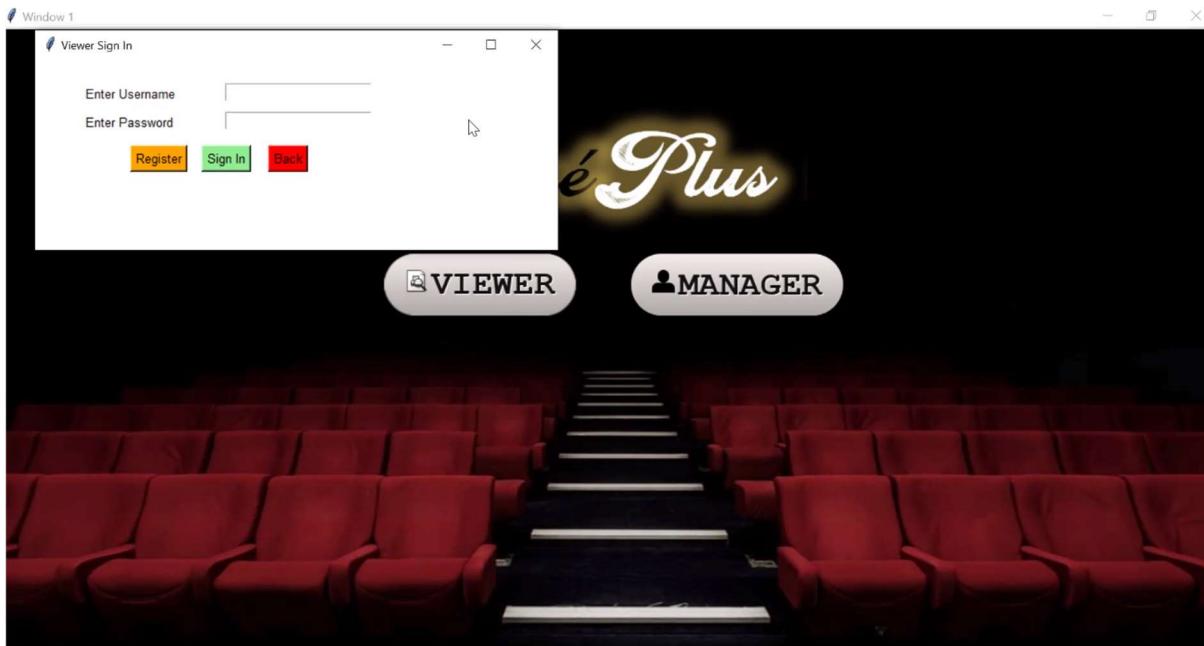
OUTPUT

As the program starts the following screen is displayed. It asks the user to either sign in as a viewer or as the manager.

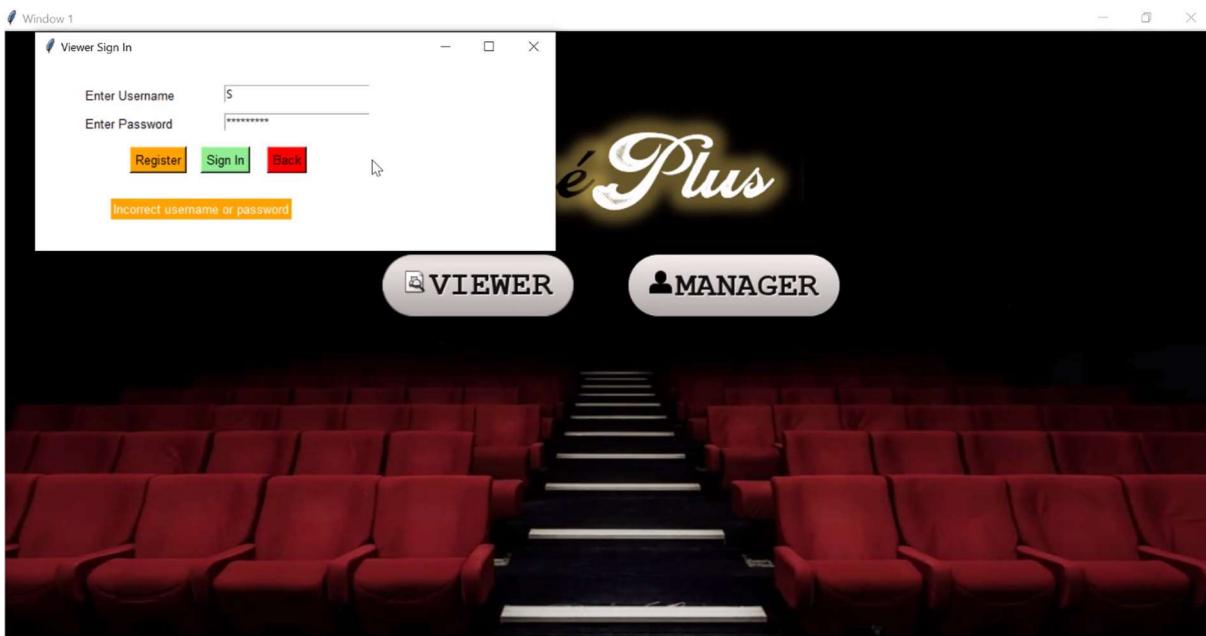
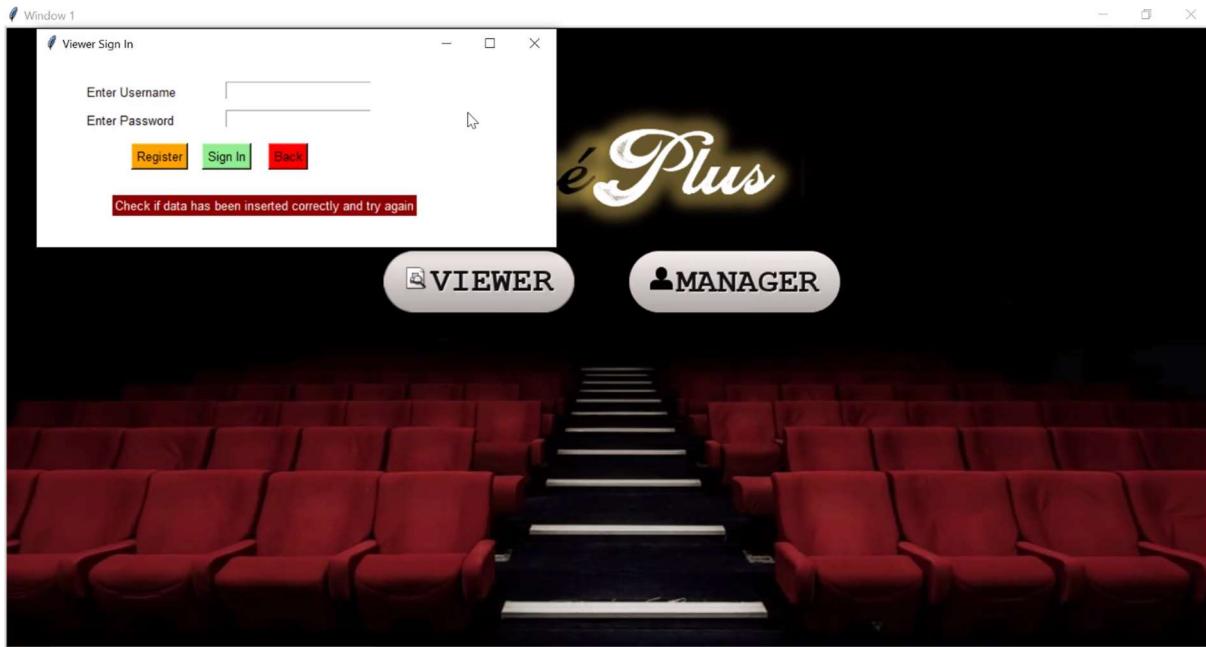


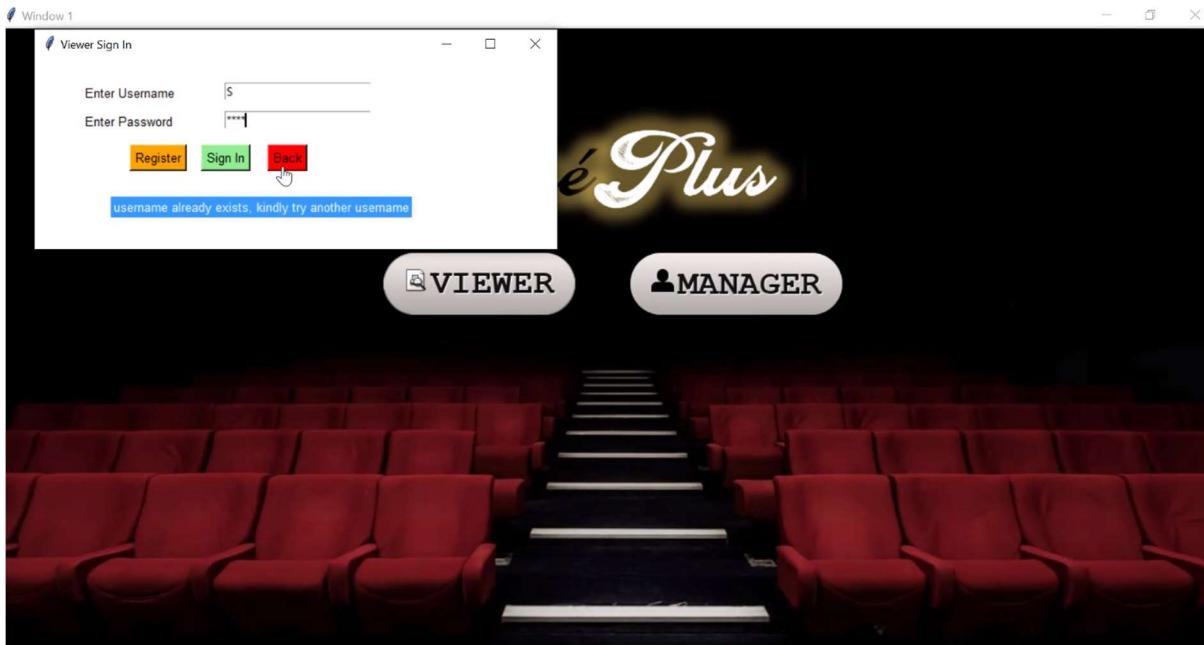
Viewer

If the user clicks on “Viewer”, the program prompts the user to either register (if they are a new user) or sign in (if their account already exists) by entering their username and password.

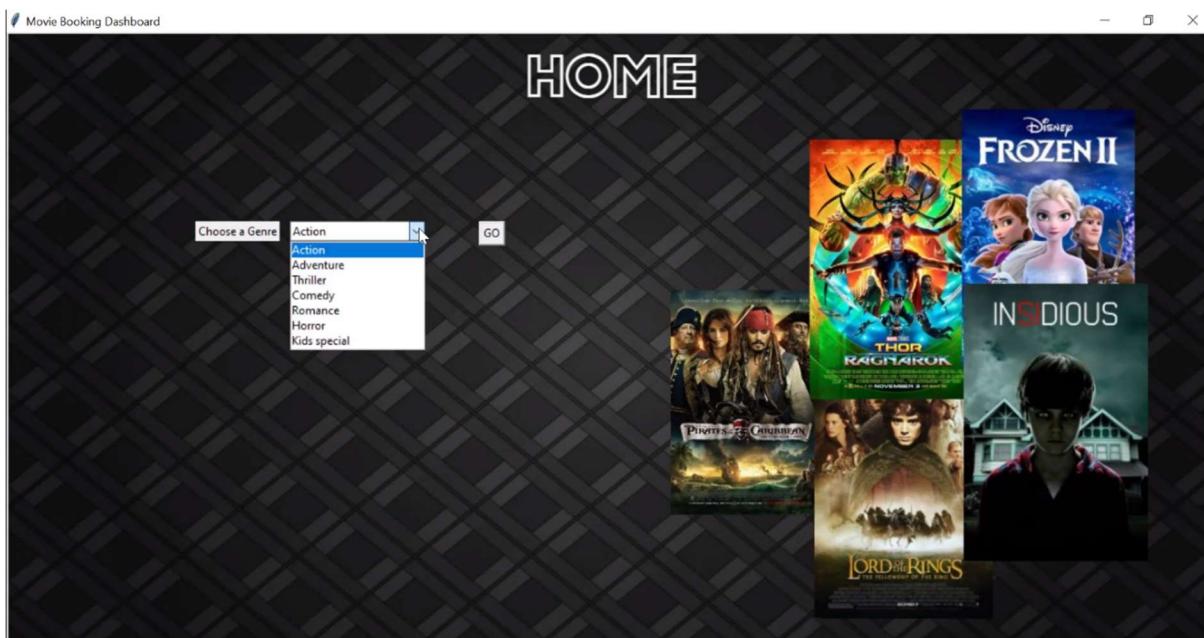


If the user does not enter any information or enters incorrect information of tries to register with an already existing (username,password) pair and clicks sign in, the following errors are generated respectively.

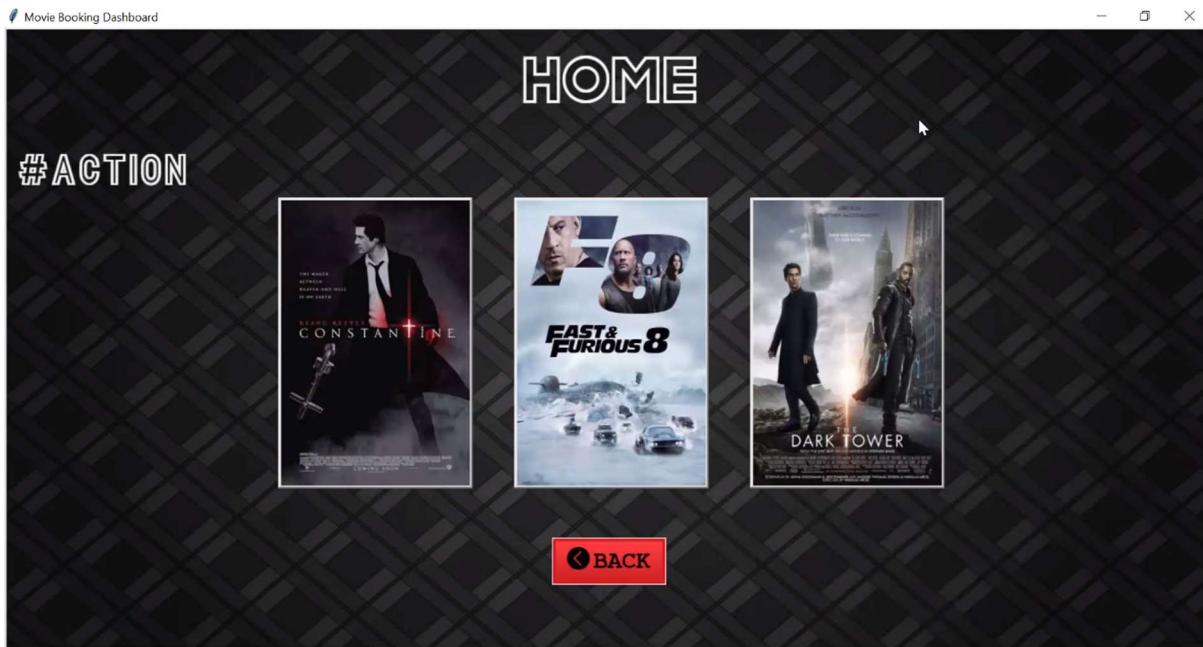




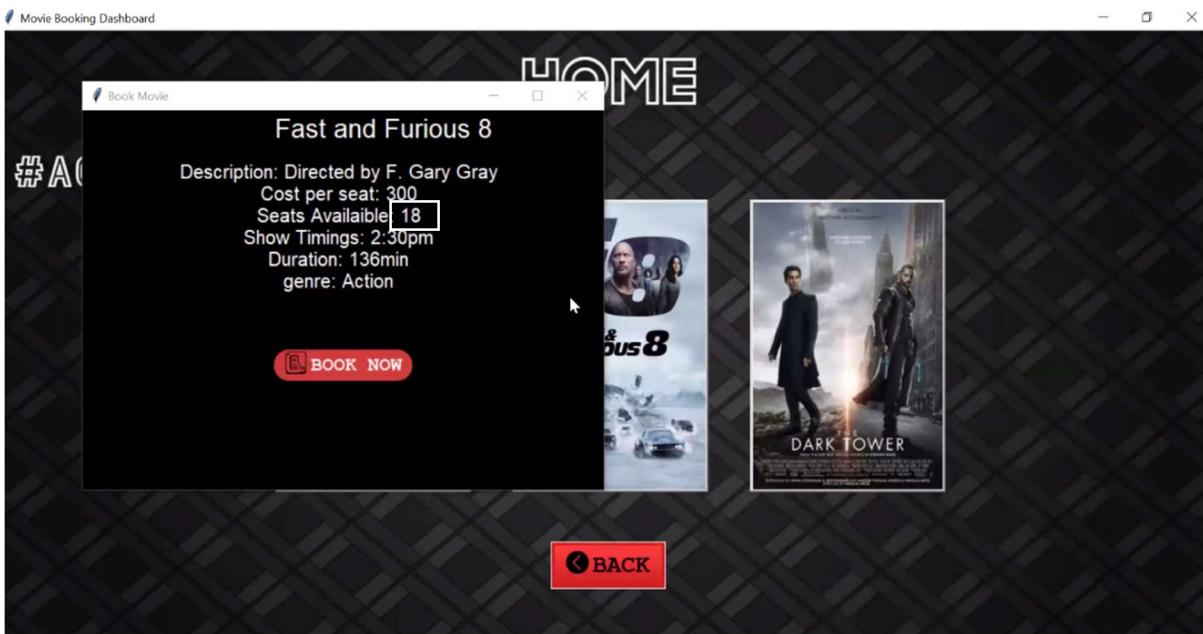
After signing in, the following screen is displayed where the user can choose their desired movie genre and click go.



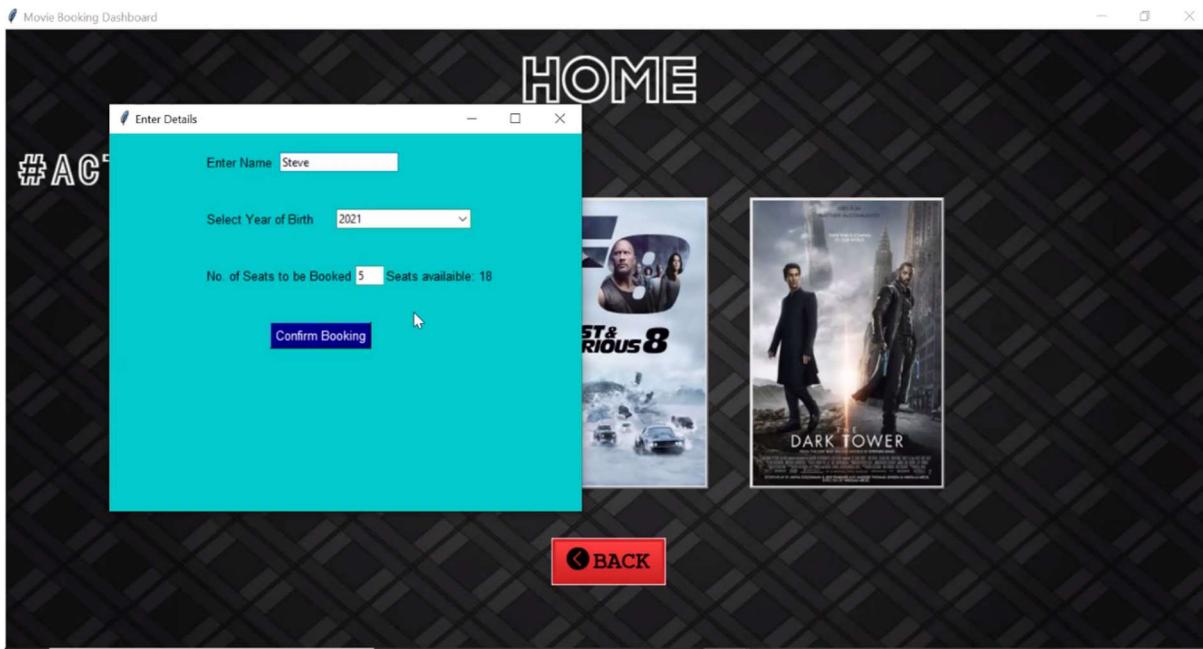
For demonstration we have taken “Action” as the chosen genre. The following screen is thus displayed.



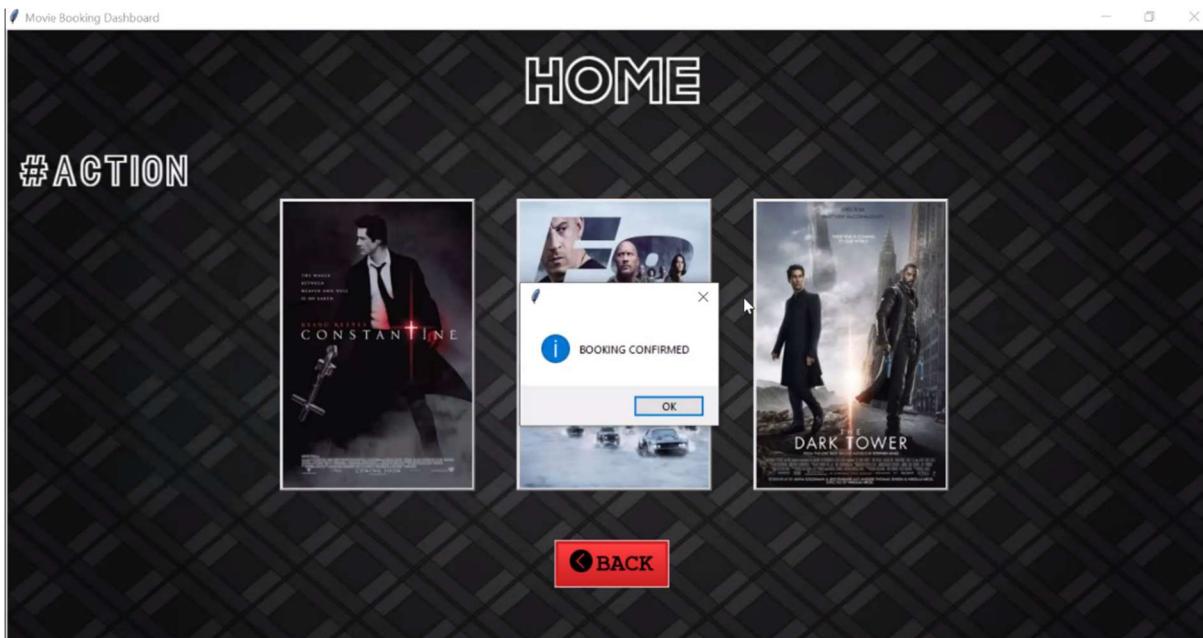
The user now can click on any of the displayed posters to obtain the movie details.



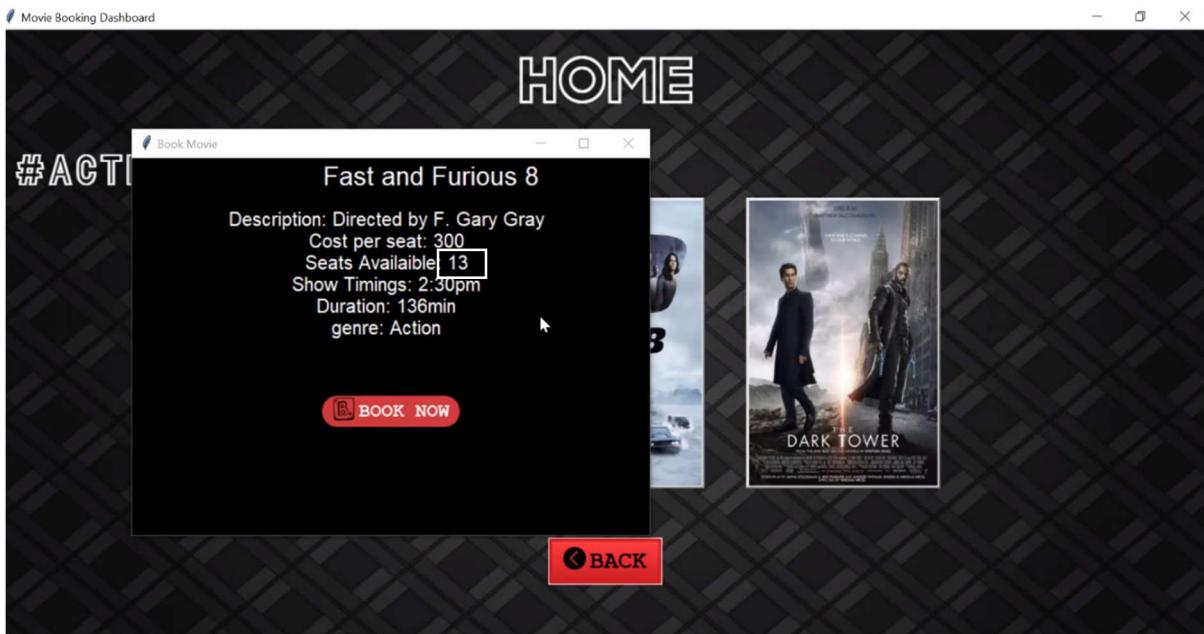
If the user chooses to book the movie by clicking on “BOOK NOW”, the following details are asked from the user.



As the user confirms the booking the following message is displayed.



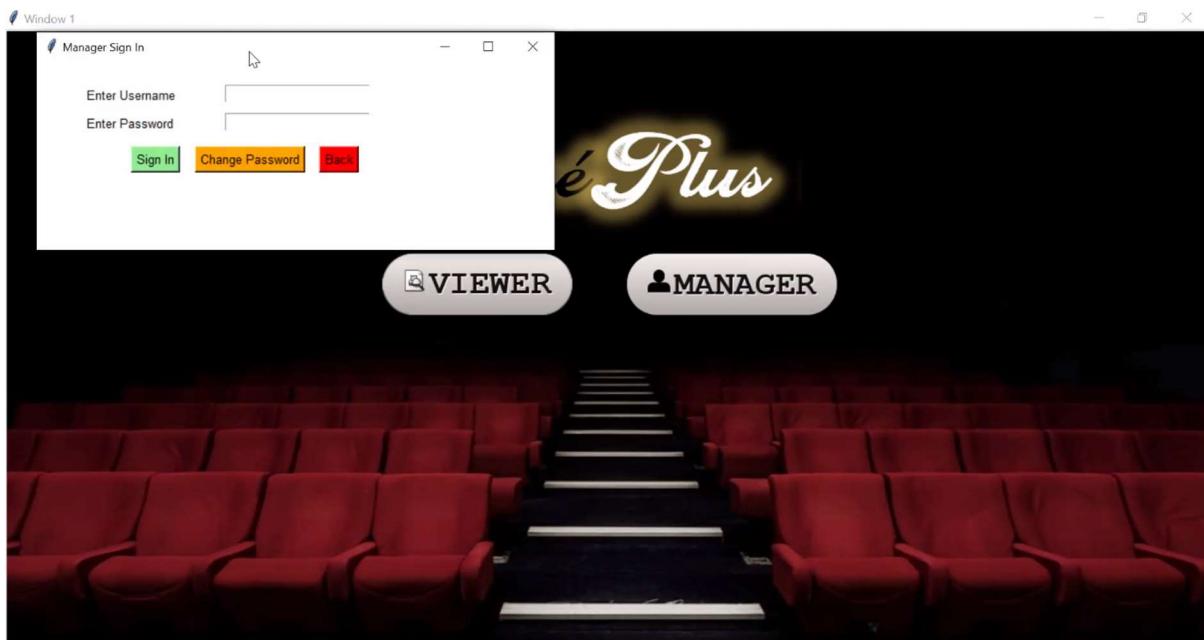
NOTE: The no. of seats available has been updated.



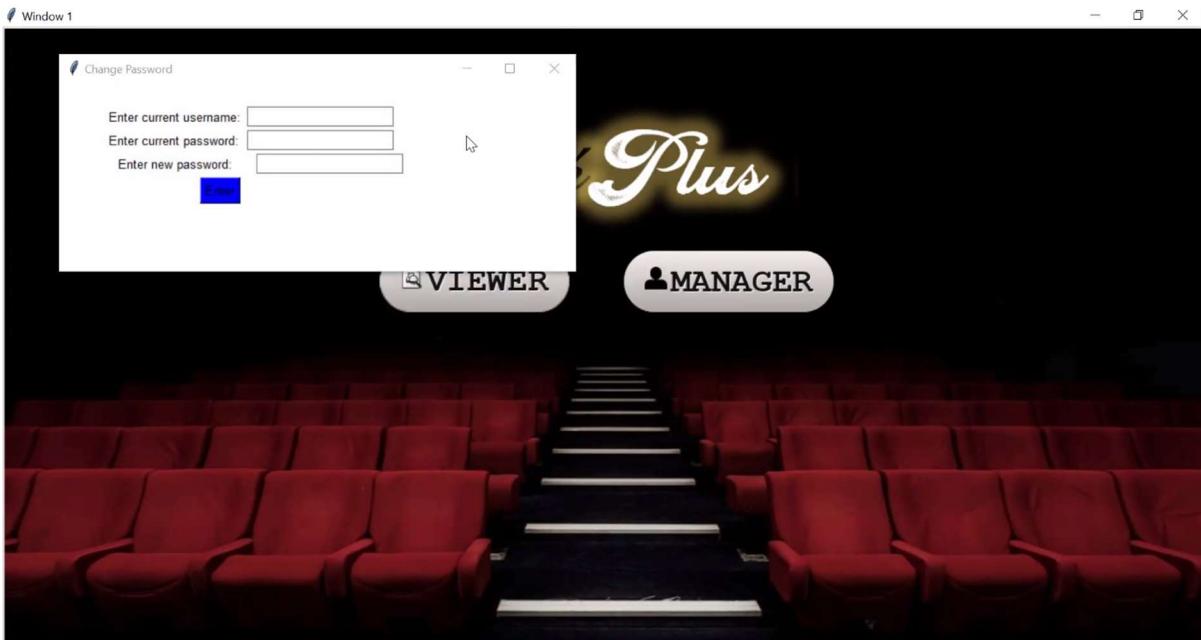
Now, the user can go back and book more movies as per their wish.

Manager

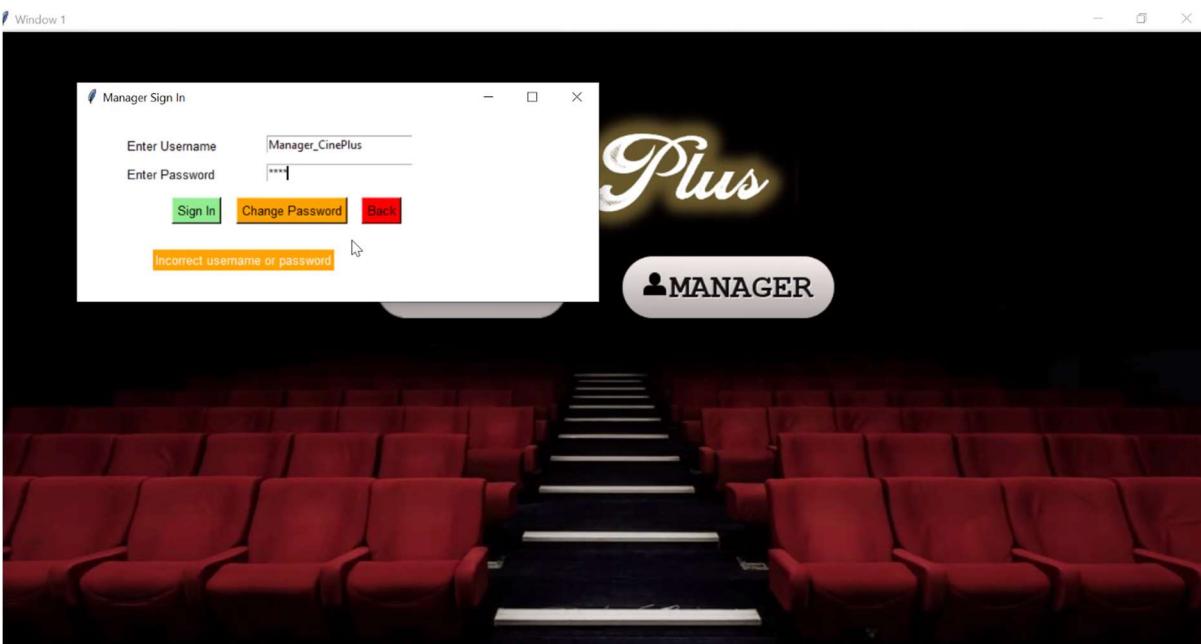
If the user clicks on “Manager”, the program prompts the user to enter their username and password.



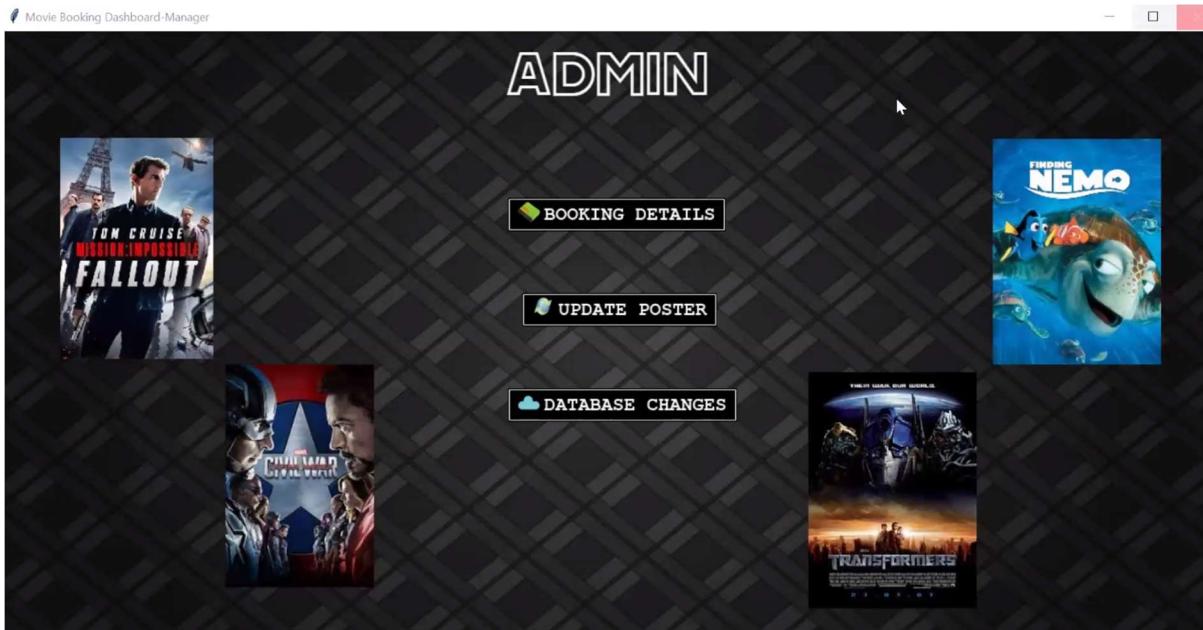
Here, the user is also given an option to change their password.



While signing in, if the user enters wrong credentials then, the following error is generated.



As the user signs in, the following screen is displayed.



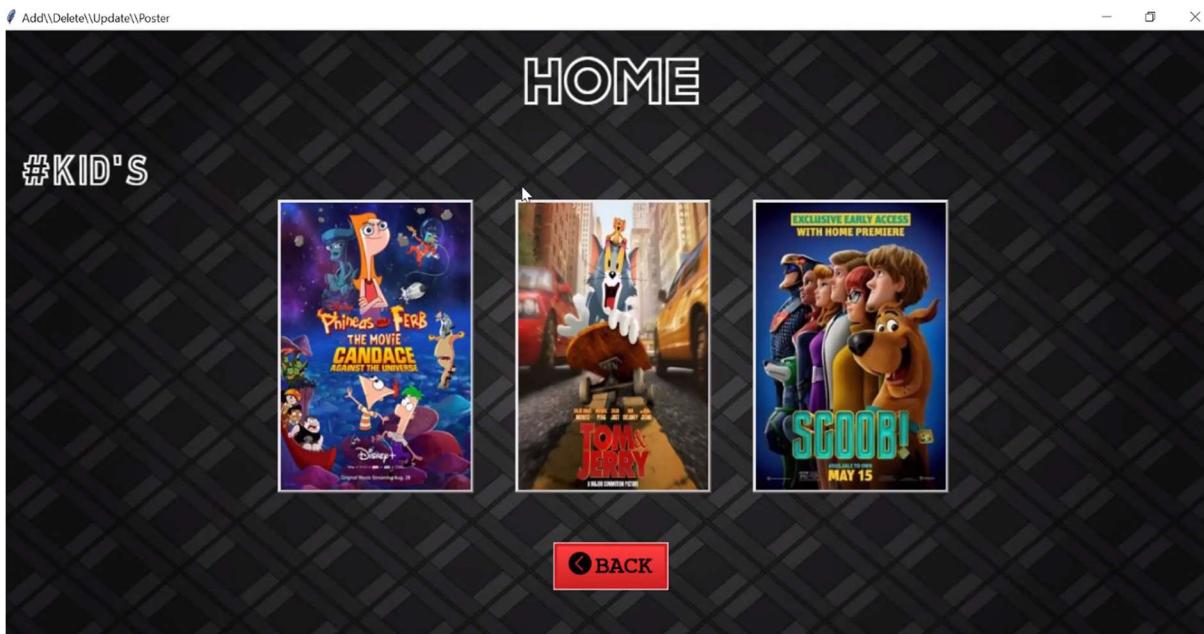
If the user click on “BOOKING DETAILS”, a popup is displayed which shows all of the bookings done through the viewer portal.

Booking Details				
Sno.	Customer Name	Year Of Birth	Movie Name	Seats Booked
4	Maitreyi	1996	Jab We Meet	2
5	Shiva	2004	Fast and Furious 8	4
6	Anshi	2003	Interstellar	3
7	Shiva	2004	Interstellar	3
8	Anshi	2003	Interstellar	4
9	Payal	2004	Interstellar	2
10	Shiva	2004	Interstellar	4
11	S	2004	Avatar	4
12	S	2004	Gravity	5
13	S	2004	Avatar	5
14	Shiva	2004	Phineas and Ferb	5
15	Shiva	2004	Gravity	5
16	ABC	2004	Interstellar	10
17	Shiva	2004	Interstellar	5
18	Steve	2021	Fast and Furious 8	5

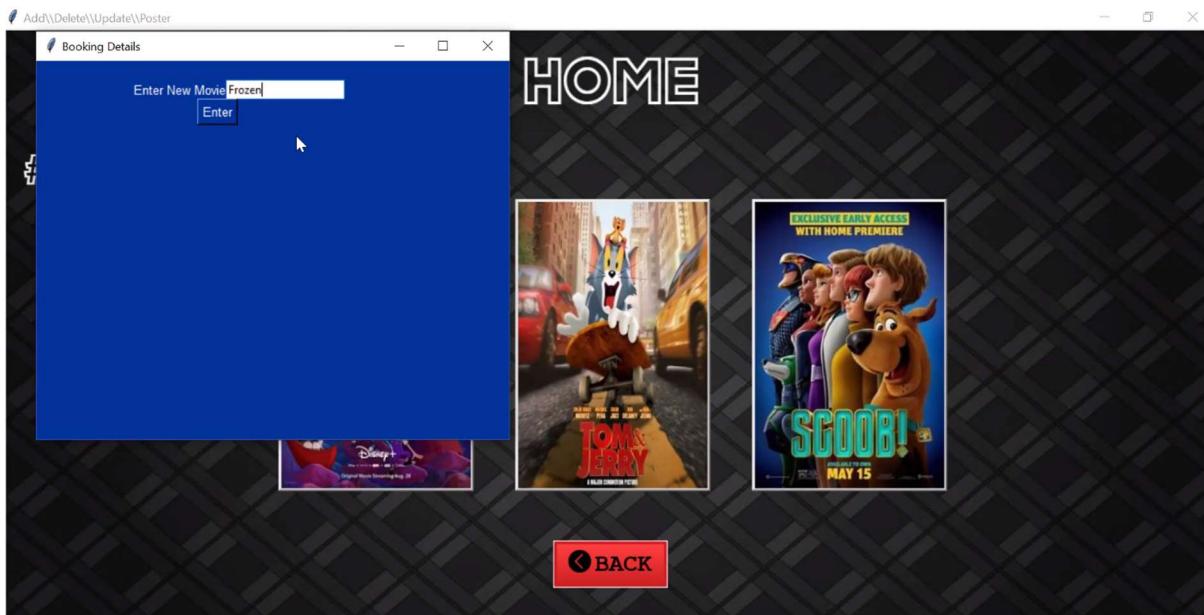
If the user clicks on “UPDATE POSTER”, a screen similar to the one in the viewer portal is displayed. The manager can choose any genre in which they want to update the poster. For demonstration we have chosen the genre “Kids Special”.

The screenshot shows the Admin dashboard of the Cineplus Movie Booking Dashboard. The title "ADMIN" is prominently displayed at the top center. Below it are four movie posters arranged in a 2x2 grid: "Mission Impossible: Fallout" (top-left), "FINDING NEMO" (top-right), "Captain America: Civil War" (bottom-left), and "Transformers" (bottom-right). Between the posters are three buttons: "BOOKING DETAILS" (top), "UPDATE POSTER" (middle), and "DATABASE CHANGES" (bottom). The background has a dark, textured pattern.

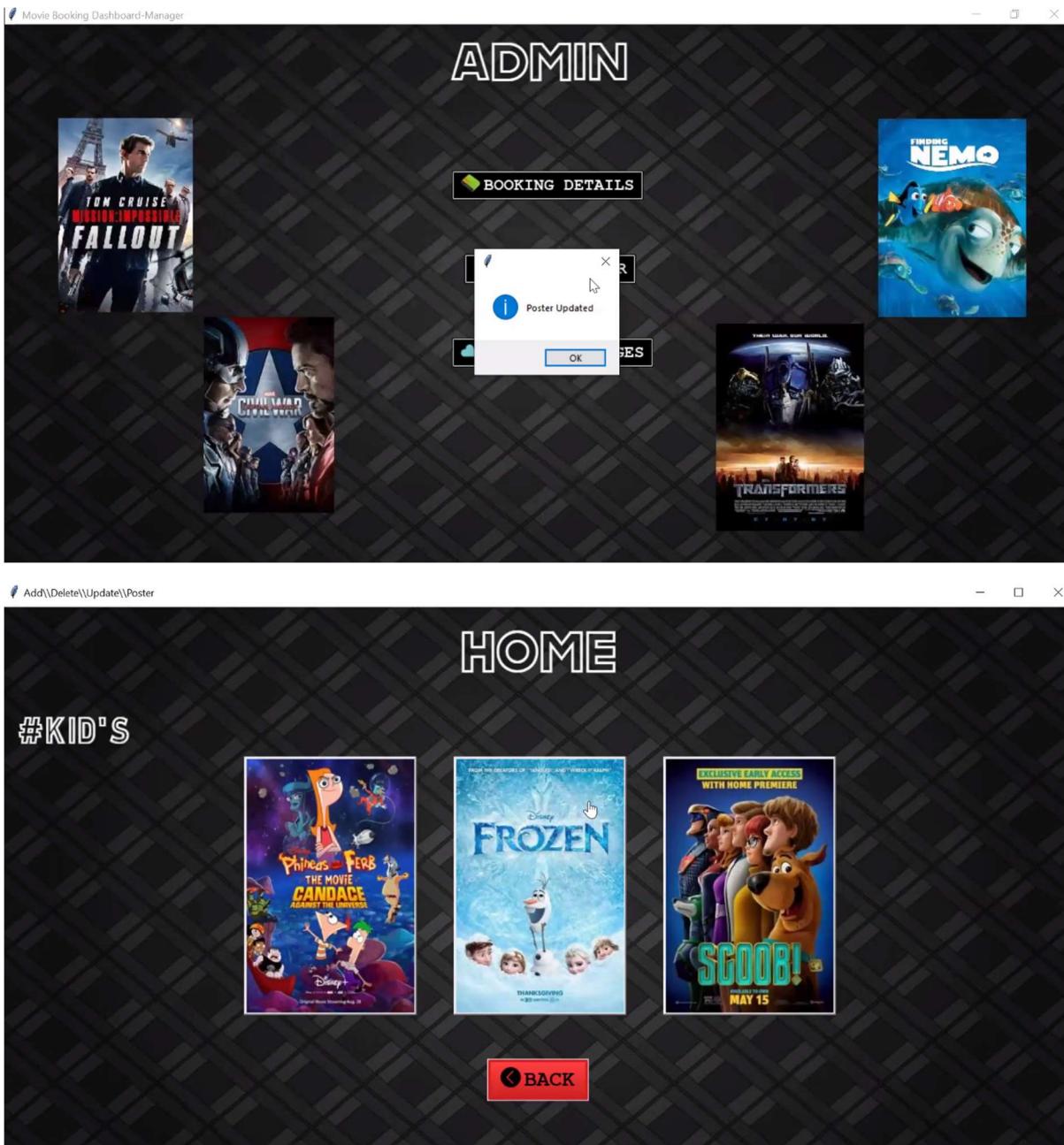
The screenshot shows the Home page of the Cineplus Movie Booking Dashboard. The title "HOME" is prominently displayed at the top center. Below it is a search bar with the placeholder "Choose a Genre" and a dropdown menu set to "Kids special". To the right of the search bar is a blue "GO" button with a magnifying glass icon. The background has a dark, textured pattern.



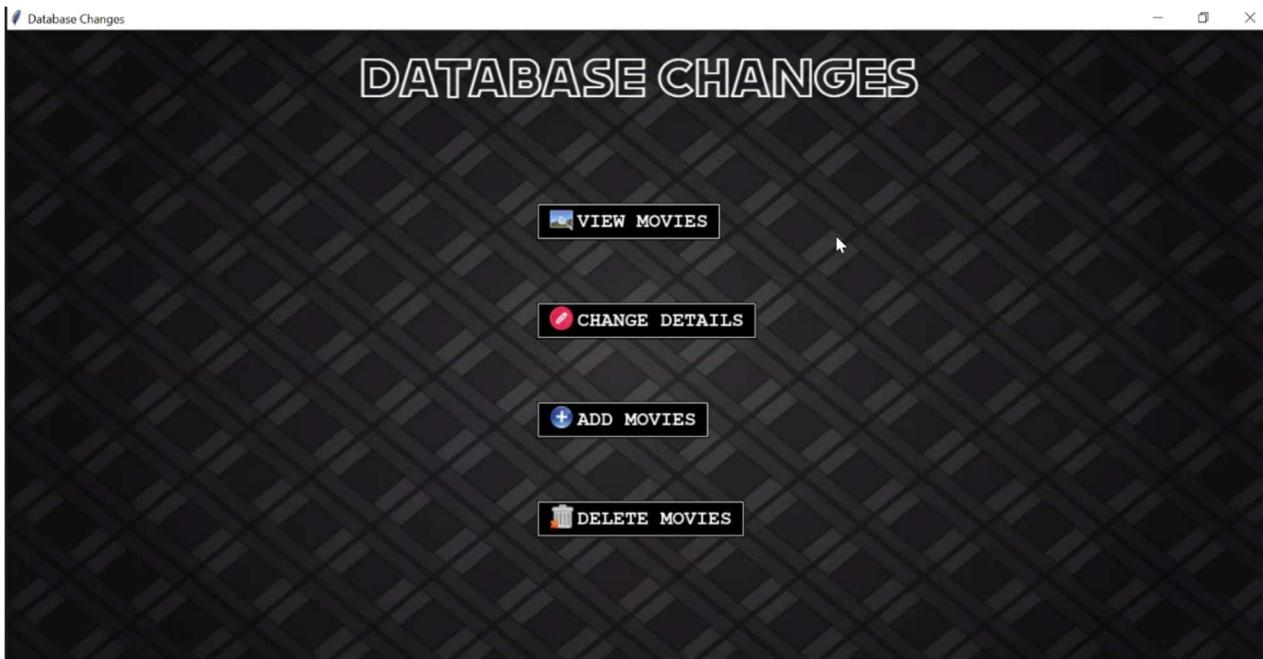
The manager can click on the poster they want to change. As they click on the desired poster, they are prompted to enter the new movie name whose poster has to be placed in the selected poster's place.



On clicking “Enter”, the desired poster is updated.



Back on the Admin Page. If the user clicks on “DATABASE CHANGES”, the following screen is displayed.



On clicking on “VIEW MOVIES”, the following screen with the complete details of all movies is displayed.

Booking Details									
Sno.	Movie Name	Description	Show Timing	Duration(in min)	Seats Available	Cost Per Seat	Genre	Poster Location	
1	3 Idiots	Directed by Rakeysh Omprakash Mehra	11:00am	180	70	250	Comedy	3idiots.jpg	
2	Jab We Met	Directed by Imtiaz Ali	12:45pm	195	45	170	Comedy	J.jpg	
3	Gravity	Directed by Alfonso Cuarón	10:00pm	95	50	230	Thriller	Gravity.jpg	
4	World War Z	Directed by Marc Forster	12:00pm	140	30	300	Thriller	WorldWarZ.jpg	
5	Inception	Directed by Christopher Nolan	2:00pm	120	44	450	Thriller	Inception.jpg	
6	Avatar	Directed by James Cameron	6:00pm	115	35	235	Adventure	avatar.jpg	
7	Constantine	Directed by Francis Lawrence	5:30pm	150	40	370	Action	Constantine.jpg	
8	Veer Zaara	Directed by Yash Chopra	2:00pm	200	76	200	Romance	VeerZaara.jpg	
9	Dil Chahta Hai	Directed by Farhan Akhtar	9:00am	165	45	245	Romance	DilChahtaHai.jpg	
10	Little Women	Directed by Greta Gerwig	11:00am	100	20	350	Romance	LittleWomen.jpg	
11	I Frankenstein	Directed by Stuart Veattie	12:00pm	75	34	150	Action	frank.jpg	
12	The Dark Tower	Directed by Nikolaj Arcel	3:00pm	200	30	200	Action	dark.jpg	
13	Annabelle	Directed by John R. Leonetti	9:00pm	155	30	350	Horror	annabelle.jpg	
14	Cars	Directed by John Lasseter	2:00pm	90	52	250	Kids	cars.jpg	
15	Toys Story	Directed by John Lasseter	12:00pm	130	90	100	Kids	toys.jpg	
16	Frozen	Directed by Jennifer Lee	5:00pm	120	20	175	Kids	frozen.jpg	
17	IT	Directed by Andrés Muschietti	4:00pm	135	55	200	Horror	it.jpg	
18	Insidious The Last Key	Directed by Adam Robitel	11:30pm	103	40	290	Horror	insidious.jpg	
19	The Conjuring 2	Directed by James Wan	11:00pm	134	34	210	Horror	c2.jpg	
20	Phineas and Ferb	Directed by Robert C. Cooper	4:30pm	90	45	120	Kids	pf.jpg	

SELECT AND SCROLL...

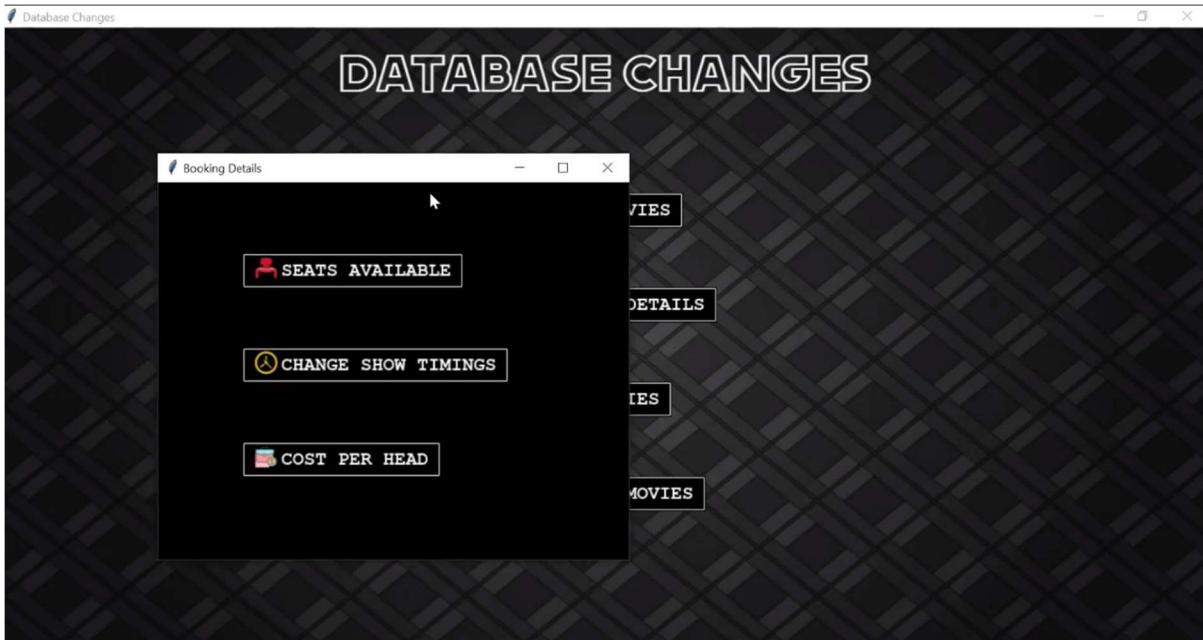
 Booking Details

Sno.	Movie Name	Description	Show Timing	Duration(in min)	Seats Available	Cost Per Seat	Genre	Poster Location
15	Toys Story	Directed by John Lasseter	12:00pm	130	90	100	Kids	toys.jpg
16	Frozen	Directed by Jennifer Michelle Lee	5:00pm	120	20	175	Kids	frozen.jpg
17	IT	Directed by Andres Muschietti	4:00pm	135	55	200	Horror	it.jpg
18	Insidious: The Last Key	Directed by Adam Robitel	11:30pm	103	40	290	Horror	insidious.jpg
19	The Conjuring 2	Directed by James Wan	11:00pm	134	34	210	Horror	c2.jpg
20	Phineas and Ferb	Directed by Robert Hughes	4:30pm	90	45	120	Kids	pf.jpg
21	Tom and Jerry	Directed by Tim Story	3:00pm	101	60	100	Kids	TomJerry.jpg
22	Scooby-Doo	Directed by Raji Gosnell	3:40pm	90	50	128	Kids	Scoob.jpg
23	Twilight	Directed by Catherine Hardwicke	5:00pm	126	50	200	Romance	twilight.jpg
24	The Parent Trap	Directed by Nancy Meyers	3:30pm	128	60	260	Comedy	parenttrap.jpg
25	The Fault In Our Stars	Directed by Josh Boone	2:00pm	133	90	280	Romance	stars.jpg
26	Extraction	Directed by Sam Hargrave	5:00pm	116	30	320	Thriller	extraction.jpg
27	Justice League	Directed by Zack Snyder	5:30pm	120	10	200	Action	JL.jpg
28	Interstellar	Directed by Christopher Nolan	4:00pm	175	15	150	Adventure	in.jpg
29	Jumanji: Welcome To The Jungle	Directed by Jake Kasdan	4:30pm	119	25	300	Adventure	jumangi.jpg
30	Jurassic World	Directed by Colin Trevorrow	3:45pm	124	24	250	Adventure	jurassicworld.jpg
31	Fast and Furious 8	Directed by F. Gary Gray	2:30pm	136	13	300	Action	ff8.jpg
32	Adventures In Babysitting	Directed by John Schultz	1:45pm	99	40	240	Adventure	babysitting.jpg
33	Home Alone	Directed by ABC	2:00pm	120	25	120	Kids	H.jpg
34	Ra.one	Directed by ABC	2:00pm	120	45	25	Action	r1.jpg

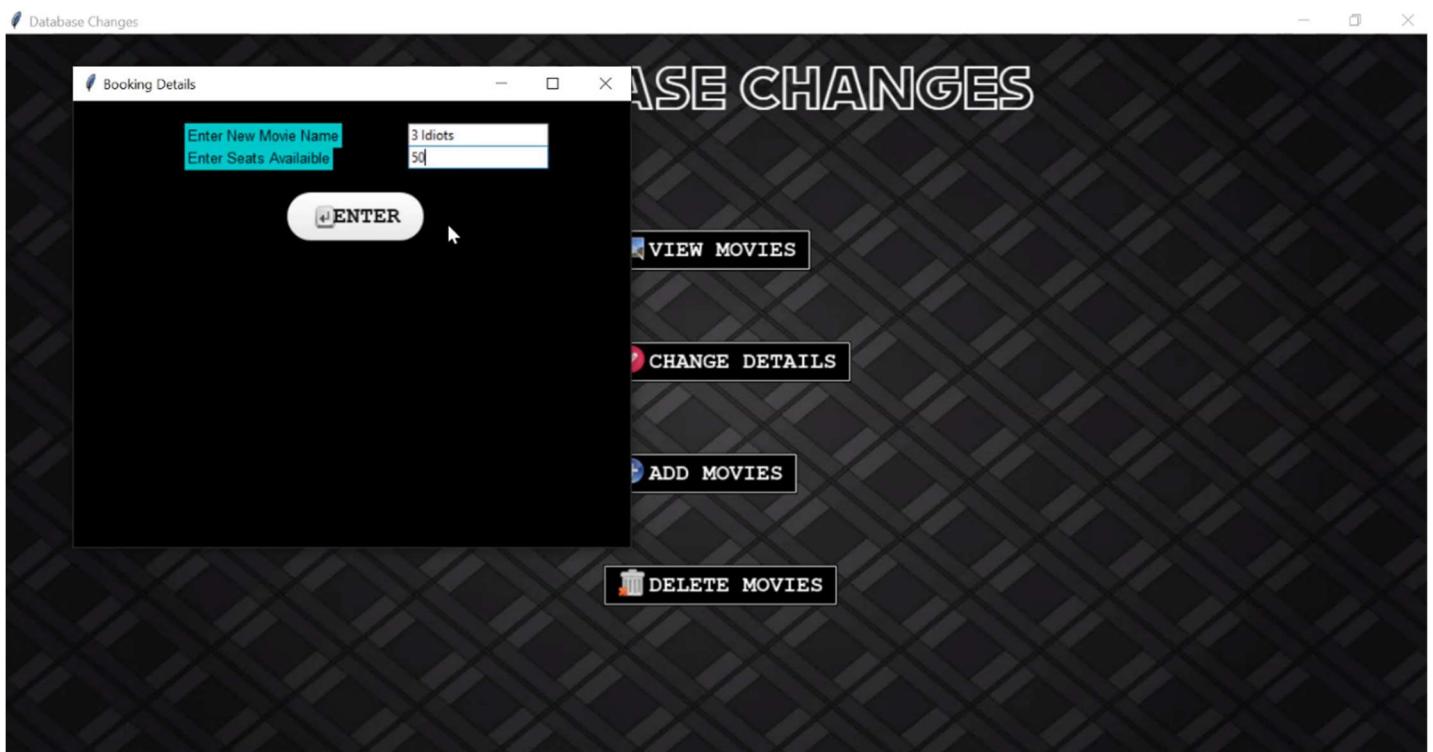
SELECT AND SCROLL...

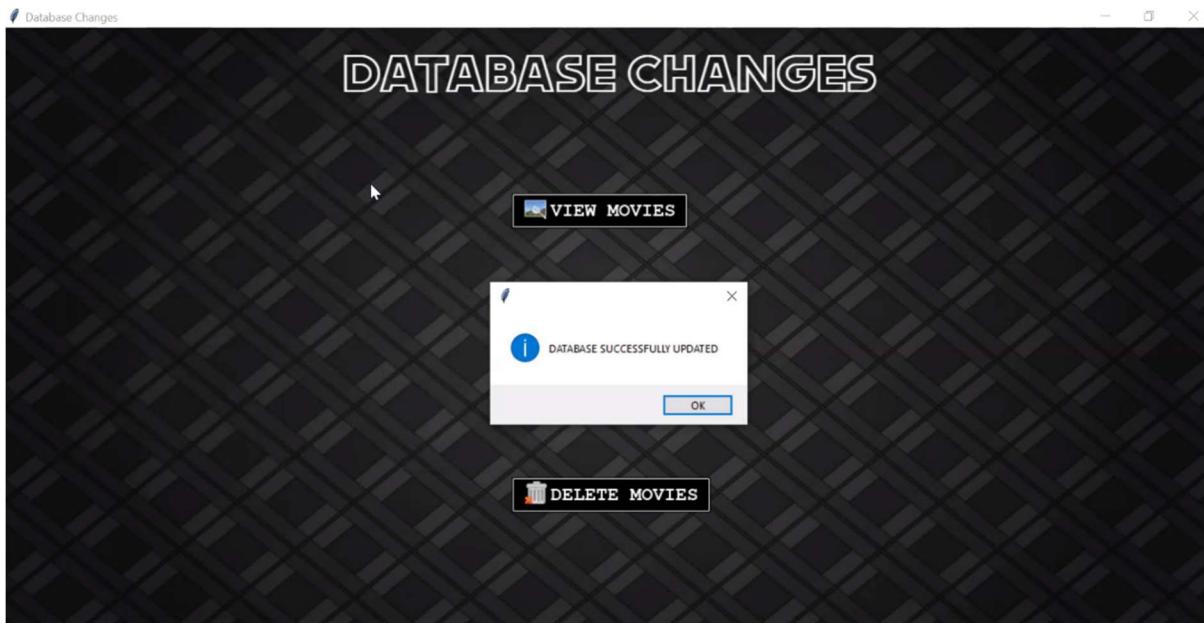
On clicking on change details, a popup is displayed which allows us to make a specific change in the movie details.





We can further make any necessary changes in the movie details by selecting the respective button and entering the movie name whose detail has to be changed and the new details.





(Before change)

Booking Details

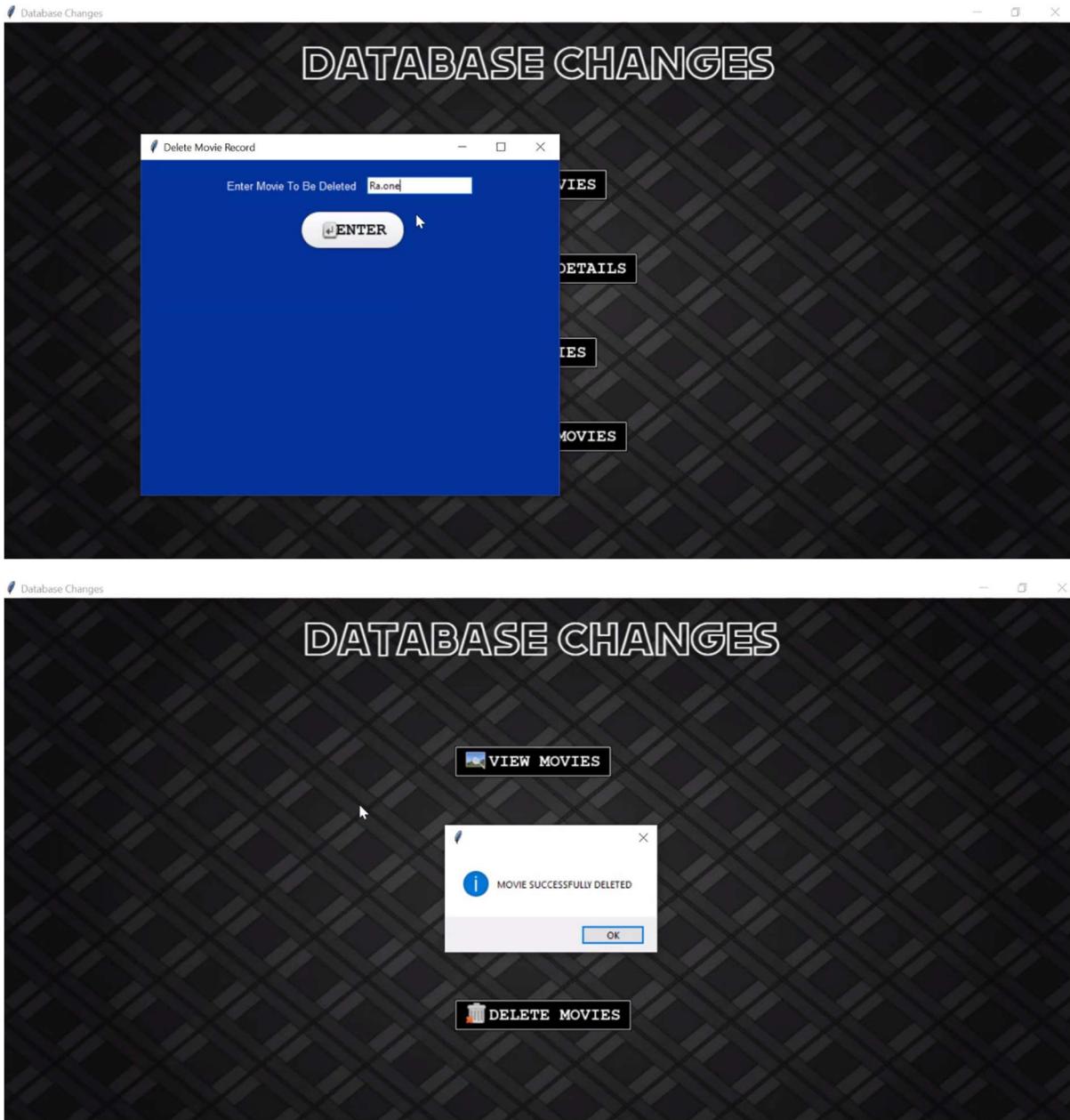
Sno.	Movie Name	Description	Show Timing	Duration(in min)	Seats Available
1	3 Idiots	Directed by Rajkumar Hirani	11:00am	180	70
2	Jab We Meet	Directed by Imitiaz Ali	12:45pm	195	45
3	Gravity	Directed by Alfonso Cuaron	10:00pm	95	50
4	World War Z	Directed by Marc Foster	12:00pm	140	30
5	Inception	Directed by Christopher Nolan	2:00pm	120	44
6	Avatar	Directed by James Cameron	6:00pm	115	35
7	Constantine	Directed by Francis Lawrence	5:30pm	150	40

(After change)

Booking Details

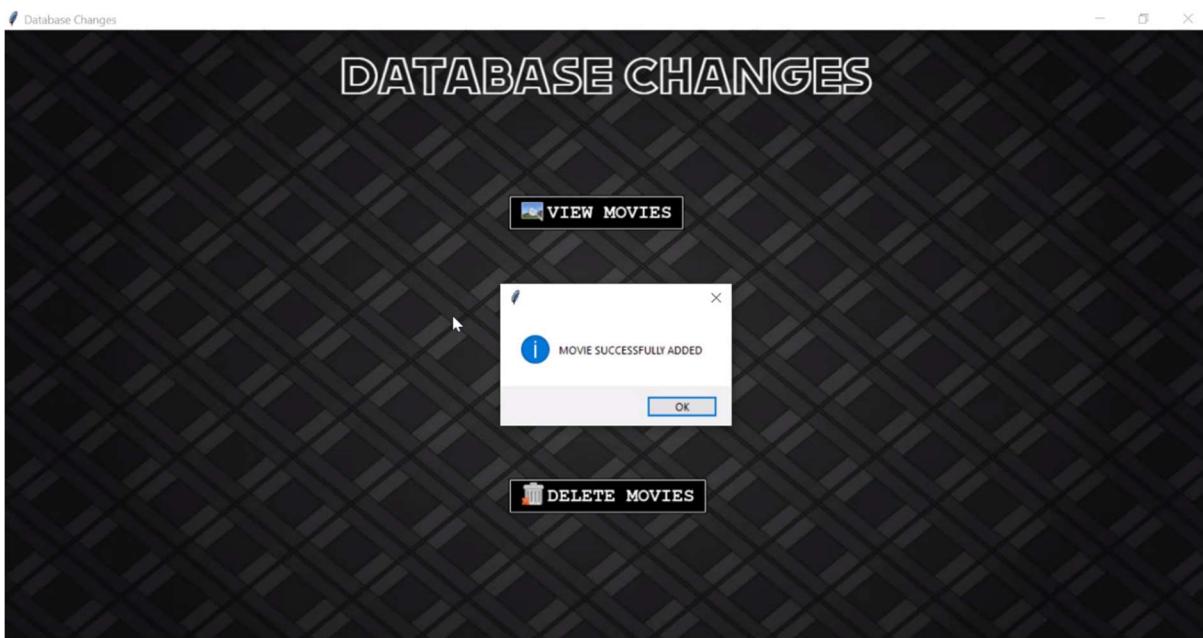
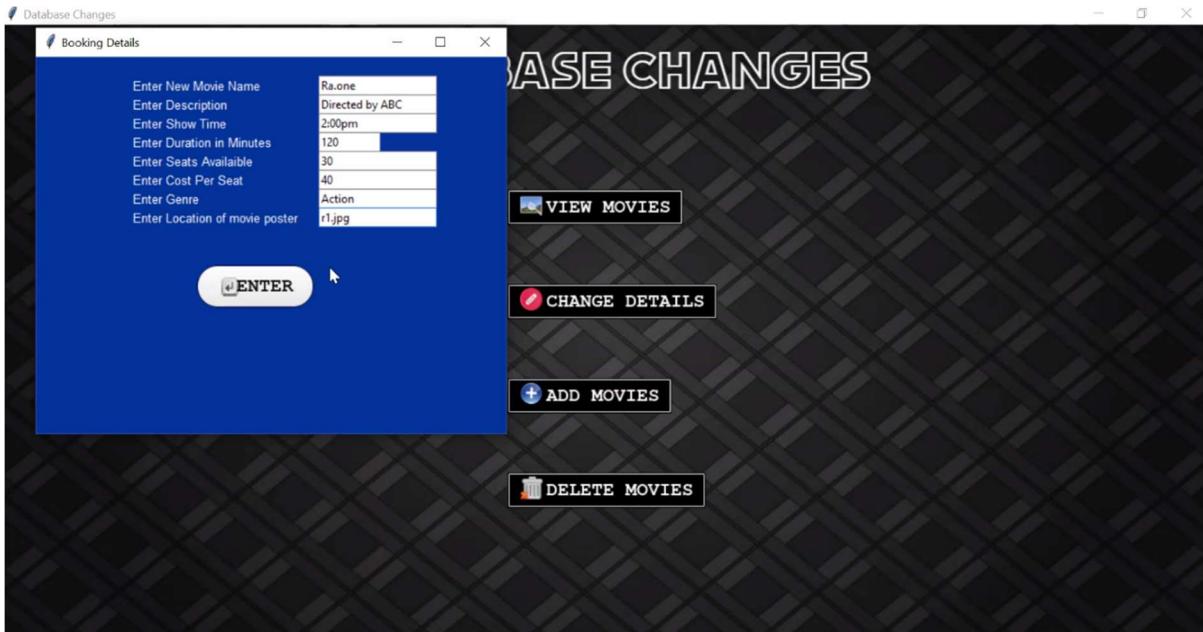
Sno.	Movie Name	Description	Show Timing	Duration(in min)	Seats Available
1	3 Idiots	Directed by Rajkumar Hirani	11:00am	180	50
2	Jab We Meet	Directed by Imitiaz Ali	12:45pm	195	45
3	Gravity	Directed by Alfonso Cuaron	10:00pm	95	50
4	World War Z	Directed by Marc Foster	12:00pm	140	30
5	Inception	Directed by Christopher Nolan	2:00pm	120	44
6	Avatar	Directed by James Cameron	6:00pm	115	35
7	Constantine	Directed by Francis Lawrence	5:30pm	150	40
8	Veer Zaara	Directed by Yash Chopra	2:00pm	200	76
9	Dil Chahta	Directed by Farhaan Akhtar	9:00am	165	45
10	Little Women	Directed by Greta Gerwig	11:00am	100	20

If the user clicks on “Delete movie”, they are prompted to enter the name of the movie to be deleted and the required message is displayed.



We can observe this change through the “View Movies” Button.

Similarly, we can add a movie to the database by entering all the required details.



CONCLUSION

The entire project with its thousands of lines of code and the tremendous amount of time and energy that has been spent by the group on this project has successfully borne fruit and the program works quite well.

This project shows the working of MySQL and Python and also provides a program that can be utilized to book movies both in the online and offline domain.

FUTURE IMPROVEMENTS

Due to lack of time we were unable to make further developments. However, the following improvements can be introduced in the future:

1. An OTP system
2. An invoice generator to provide a detailed record for the user.
3. Upgrade the graphical interface
4. Online snack booking
5. A subprogram for offers and discounts

BIBLIOGRAPHY

1. <https://www.geeksforgeeks.org/>
2. <https://stackoverflow.com/>
3. <https://www.youtube.com/>