**SOFTWARE SPECIFICATIONS**

* Operating System – Windows 10 pro
* Software – Python 3.11.2
* Language – English
* Programing Language: Python

INVESTIGATORY

PROJECT

2023-24

ACKNOWLEDGEMENT

A project is a golden opportunity for learning and self- development .I consider myself fortunate and privileged to have such wonderful mentors guide me the through the journey for the completion of this project.

My sincere thanks to ***Mrs. Pratheetha V.P,*** Principal, Sree Narayana Public School, Poothotta, who despite being extraordinarily busy, took time to advise and guide me. Her constant encouragement and insightful feedback has helped me achieve my objective.

I extend my heartfelt gratitude to our teacher ***Mrs. Neema*** for her patience and belief in us. Her

exemplary investment in the completion of the project, her critical reviewing of my project and assistance in solving each and every problem that occurred during implementation of this project has been great value to me.

Last but not the least. I would like to thank my family members whose support helped me complete the project within the deadline.

|  |  |  |
| --- | --- | --- |
| Sl. No | Topic | Page no |
| 1 | Software Specifications |  |
| 2 | Hardware Specifications |  |
| 3 | Acknowledgement |  |
| 4 | Introduction |  |
| 5 | Objective |  |
| 6 | Source Code |  |
| 7 | Output |  |
| 8 | Conclusion |  |
| 9 | Bibliography |  |

**INDEX**

MOVIE BOOKING SYSTEM

PROGRAM FOR ADMIN AND USER

import mysql.connector

# Connect to MySQL

db=mysql.connector.connect(host="localhost",user="root",password="root")

cursor = db.cursor()

#Create Database if they don't exist

db\_name = "movie\_booking"

cursor.execute(f"CREATE DATABASE IF NOT EXISTS {db\_name}")

cursor.execute(f"USE {db\_name}")

# Create tables if they don't exist

cursor.execute("CREATE TABLE IF NOT EXISTS users (id INT AUTO\_INCREMENT PRIMARY KEY,username VARCHAR(255) NOT NULL UNIQUE,password VARCHAR(255) NOT NULL,is\_admin BOOLEAN NOT NULL DEFAULT 0)")

cursor.execute("CREATE TABLE IF NOT EXISTS movies (id INT AUTO\_INCREMENT PRIMARY KEY,name VARCHAR(255) NOT NULL,ticket\_price DECIMAL(10, 2) NOT NULL,seats\_available INT NOT NULL)")

cursor.execute("CREATE TABLE IF NOT EXISTS bookings (id INT AUTO\_INCREMENT PRIMARY KEY,user\_id INT NOT NULL,movie\_id INT NOT NULL,FOREIGN KEY (user\_id) REFERENCES users(id),FOREIGN KEY (movie\_id) REFERENCES movies(id),seats\_booked INT NOT NULL DEFAULT 0)")

# Default admin credentials

default\_admin\_username = "admin"

default\_admin\_password = "admin"

# Function to check if admin exists

def admin\_exists():

cursor.execute("SELECT \* FROM users WHERE is\_admin = 1")

return cursor.fetchone() is not None

# Initialize admin account if it doesn't exist

if not admin\_exists():

cursor.execute("INSERT INTO users (username, password, is\_admin)VALUES (%s, %s, %s)", (default\_admin\_username, default\_admin\_password, 1))

db.commit()

# User registration function

def register\_user():

print()

username = input("Enter your username: ")

password = input("Enter your password: ")

cursor.execute("INSERT INTO users (username, password)VALUES (%s, %s)", (username, password))

db.commit()

print("Registration successful!")

print()

# Edit movie function (only for admin)

def edit\_movie():

print()

movie\_id = int(input("Enter movie ID to edit: "))

new\_name = input("Enter new movie name: ")

new\_ticket\_price = float(input("Enter new ticket price: "))

new\_seats\_available = int(input("Enter new seats available: "))

cursor.execute("UPDATE movies SET name = %s, ticket\_price = %s, seats\_available = %s WHERE id = %s", (new\_name, new\_ticket\_price, new\_seats\_available, movie\_id))

db.commit()

print()

print("Movie edited successfully!")

# Cancel movie function

def cancel\_booking(user\_id):

cursor.execute("SELECT \* FROM bookings WHERE user\_id = %s", (user\_id,))

bookings = cursor.fetchall()

if not bookings:

print("You have no bookings to cancel.")

return

print()

print()

print("Your Bookings:")

print()

for booking in bookings:

print(f"Booking ID: {booking[0]}, Movie ID: {booking[2]}, Seats Booked: {booking[3]}")

print()

booking\_id = int(input("Enter the Booking ID you want to cancel: "))

num\_seats\_to\_cancel = int(input("Enter the number of seats you want to cancel: "))

cursor.execute("SELECT seats\_booked, movie\_id FROM bookings WHERE id = %s AND user\_id = %s", (booking\_id, user\_id))

booking\_info = cursor.fetchone()

if booking\_info:

seats\_booked, movie\_id = booking\_info

cursor.execute("SELECT seats\_available FROM movies WHERE id = %s", (movie\_id,))

seats\_available = cursor.fetchone()[0]

if seats\_booked >= num\_seats\_to\_cancel:

cursor.execute("UPDATE bookings SET seats\_booked = seats\_booked - %s WHERE id = %s AND user\_id = %s", (num\_seats\_to\_cancel, booking\_id, user\_id))

cursor.execute("UPDATE movies SET seats\_available = seats\_available + %s WHERE id = %s", (num\_seats\_to\_cancel, movie\_id))

db.commit()

print()

print(f"{num\_seats\_to\_cancel} seat(s) canceled successfully!")

print()

else:

print(f"You can cancel up to {seats\_booked} seat(s) for this booking.")

else:

print("Invalid booking ID or this booking does not belong to you.")

# Remove movie functon (only for admin)

def remove\_movie():

movie\_id = int(input("Enter movie ID to remove: "))

cursor.execute("DELETE FROM movies WHERE id = %s", (movie\_id,))

db.commit()

print()

print("Movie removed successfully!")

# User login function

def user\_login():

print()

username = input("Enter your username: ")

password = input("Enter your password: ")

cursor.execute("SELECT \* FROM users WHERE username = %s AND password = %s", (username, password))

user = cursor.fetchone()

if user:

return user

else:

return None

# Admin login function

def admin\_login():

print()

username = input("Enter admin username: ")

password = input("Enter admin password: ")

cursor.execute("SELECT \* FROM users WHERE username = %s AND password = %s AND is\_admin = 1", (username, password))

admin = cursor.fetchone()

if admin:

return admin

else:

return None

# Add movie function (only for admin)

def add\_movie():

print()

movie\_name = input("Enter movie name: ")

ticket\_price = float(input("Enter ticket price: "))

seats\_available = int(input("Enter seats available: "))

cursor.execute("INSERT INTO movies (name, ticket\_price, seats\_available) VALUES (%s, %s, %s)", (movie\_name, ticket\_price, seats\_available))

db.commit()

print("Movie added successfully!")

print()

# Display movie details function

def display\_movie\_details():

cursor.execute("SELECT \* FROM movies")

movies = cursor.fetchall()

print()

print("Movie Details:")

print()

for movie in movies:

print(f"Movie ID: {movie[0]}, Name: {movie[1]}, Ticket Price: ₹{movie[2]}, Seats Available: {movie[3]}")

# Function to book a movie for a user

def book\_movie(user\_id):

display\_movie\_details()

print()

movie\_id = int(input("Enter the Movie ID you want to book: "))

num\_seats = int(input("Enter the number of seats you want to book: "))

cursor.execute("SELECT \* FROM movies WHERE id = %s AND seats\_available >= %s", (movie\_id, num\_seats))

movie = cursor.fetchone()

if movie:

cursor.execute("SELECT seats\_available FROM movies WHERE id = %s", (movie\_id,))

seats\_available = cursor.fetchone()[0]

if seats\_available >= num\_seats:

cursor.execute("INSERT INTO bookings (user\_id, movie\_id, seats\_booked) VALUES (%s, %s, %s)", (user\_id, movie\_id, num\_seats))

cursor.execute("UPDATE movies SET seats\_available = seats\_available - %s WHERE id = %s", (num\_seats, movie\_id))

db.commit()

print()

print(f"{num\_seats} seat(s) booked successfully!")

print()

else:

print(f"Not enough seats available. Maximum available seats: {seats\_available}")

else:

print("Invalid movie selection or not enough available seats.")

# Main program

if \_\_name\_\_ == "\_\_main\_\_":

while True:

print()

print("Welcome to the Movie Booking Page!")

print()

print("1. Login")

print("2. Register")

print("3. Exit") # Added option to exit

print()

choice = input("Enter your choice (1/2/3): ")

if choice == "1":

print()

print("1. User Login")

print("2. Admin Login")

print("3. Exit") # Added option to exit

print()

login\_choice = input("Enter your choice (1/2/3): ")

if login\_choice == "1":

user = user\_login()

if user:

print()

print(f"Logged in as user: {user[1]}.")

while True:

print()

print("1. Display movie details")

print("2. Book a movie")

print("3. Cancel a booked movie")

print("4. Logout")

print("5. Exit") # Added option to exit

print()

user\_choice = input("Enter your choice (1/2/3/4): ")

if user\_choice == "1":

display\_movie\_details()

elif user\_choice == "2":

book\_movie(user[0])# Pass user ID to book\_movie function

elif user\_choice == "3":

cancel\_booking(user[0])

elif user\_choice == "4":

break #25/09/23 added book & cancel break

elif user\_choice == "5": # Added option to exit

print("Thank you for using our Movie Booking App!")

exit()

else:

print("Invalid choice. Please enter 1, 2, 3, or 4.")

else:

print()

print("User login failed. Please try again.")

elif login\_choice == "2":

admin = admin\_login()

if admin:

print()

print(f"Logged in as admin: {admin[1]}.")

print()

while True:

print()

print("1. Add Movie")

print("2. Edit Movie")

print("3. Remove Movie")

print("4. Display Movie Details")

print("5. Logout")

print("6. Exit") # Added option to exit

print()

admin\_choice = input("Enter your choice (1/2/3/4): ")

if admin\_choice == "1":

add\_movie()

elif admin\_choice == "2":

edit\_movie()

elif admin\_choice == "3":

remove\_movie()

elif admin\_choice == "4":

display\_movie\_details()

elif admin\_choice == "5":

break

elif admin\_choice == "6": # Added option to exit

print("Thank you for using our Movie Booking App!")

exit()

else:

print("Invalid choice. Please enter 1, 2, 3, or 4.")

else:

print("Admin login failed. Please try again.")

elif login\_choice == "3": # Added option to exit

print("Thank you for using our Movie Booking App!")

exit()

else:

print("Invalid choice. Please enter 1, 2, or 3.")

elif choice == "2":

register\_user()

elif choice == "3": # Added option to exit

print("Thank you for using our Movie Booking App!")

exit()

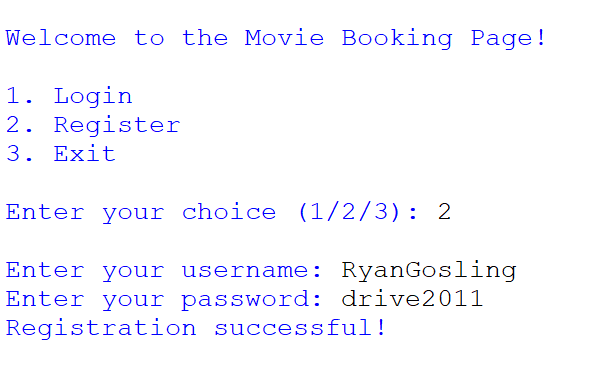
else:

print("Invalid choice. Please enter 1, 2, or 3.")

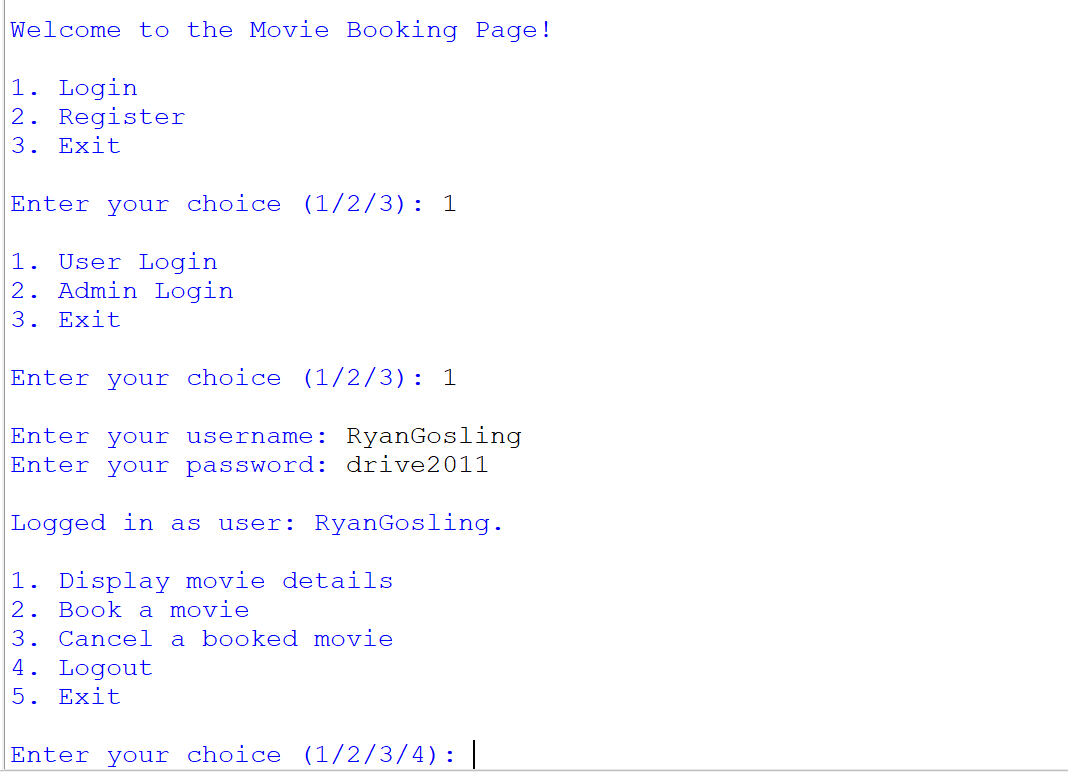
SOURCE CODE

OUTPUT

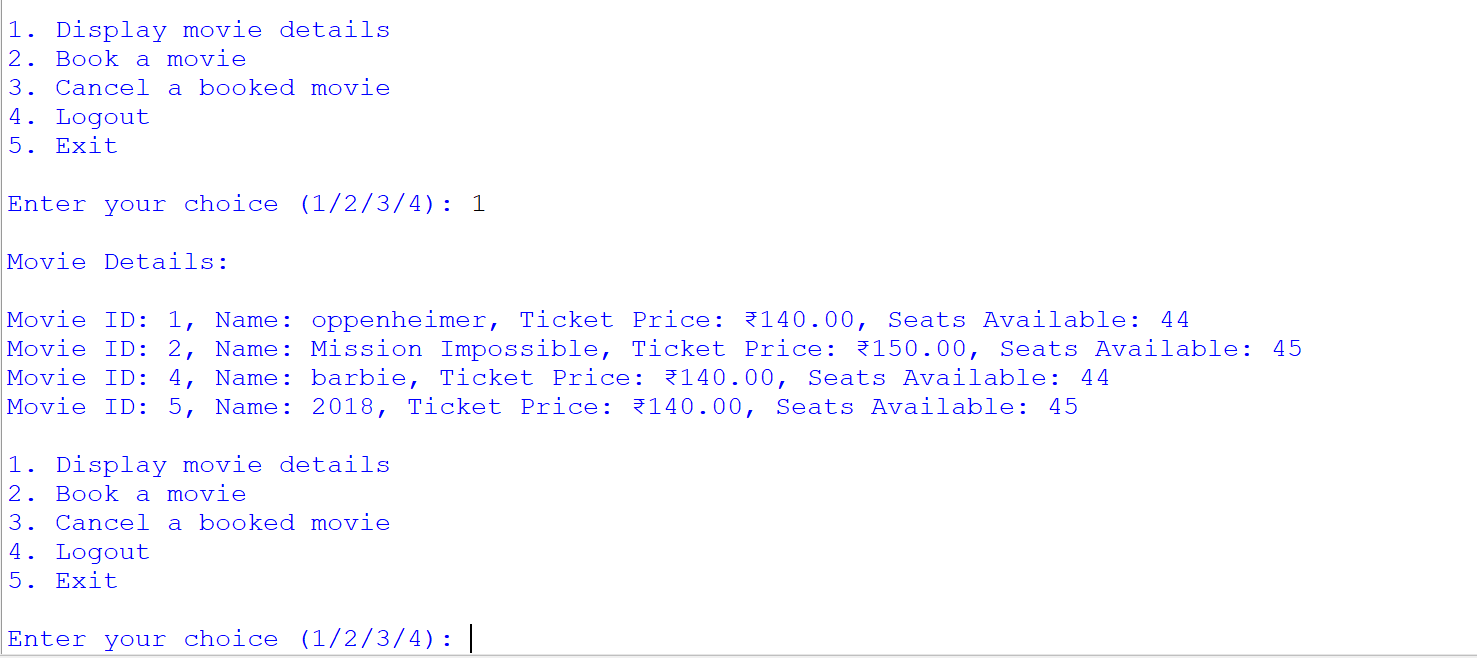
PROGRAM FOR USER TO REGISTER IF THE ARE NEW USER



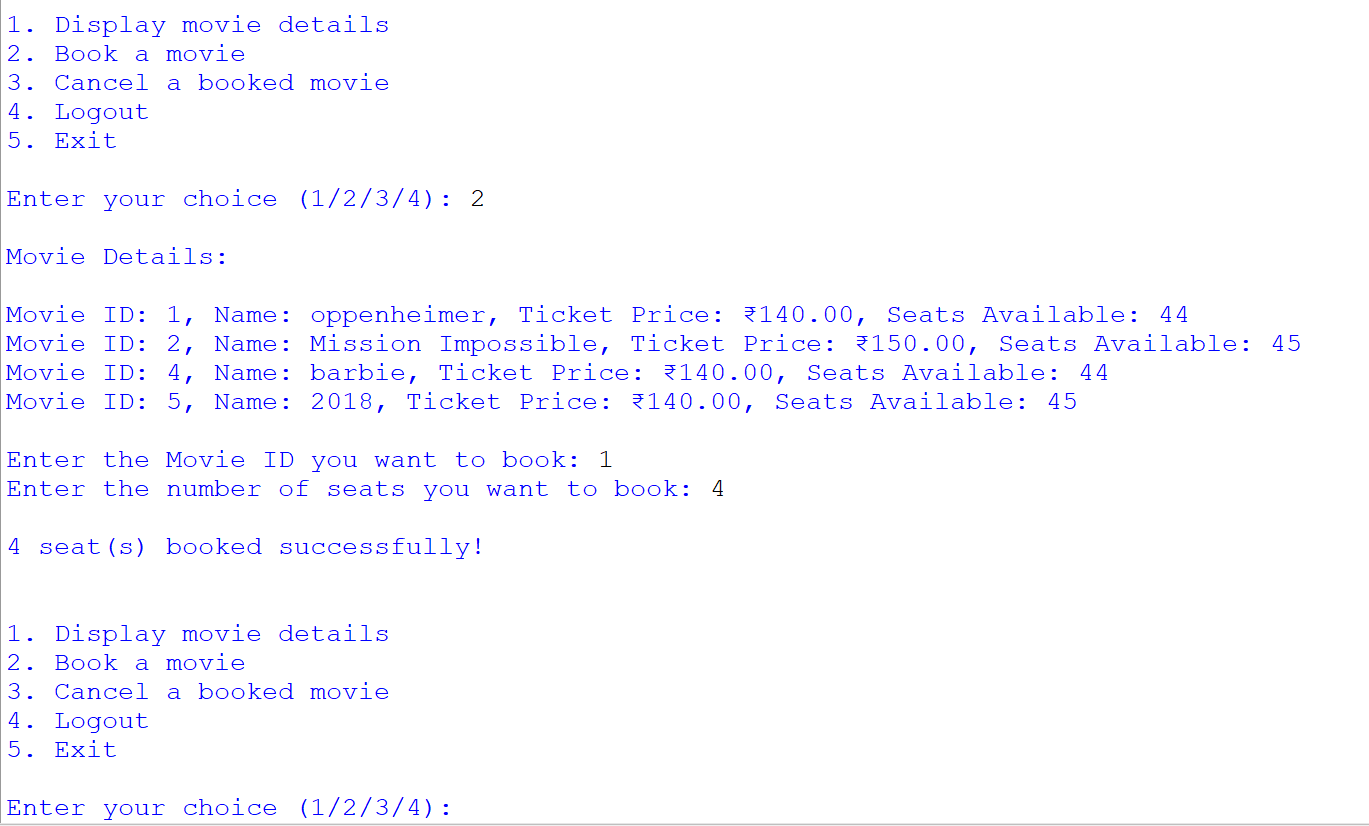
PROGRAM FOR USER TO LOG IN TO THE USER PAGE



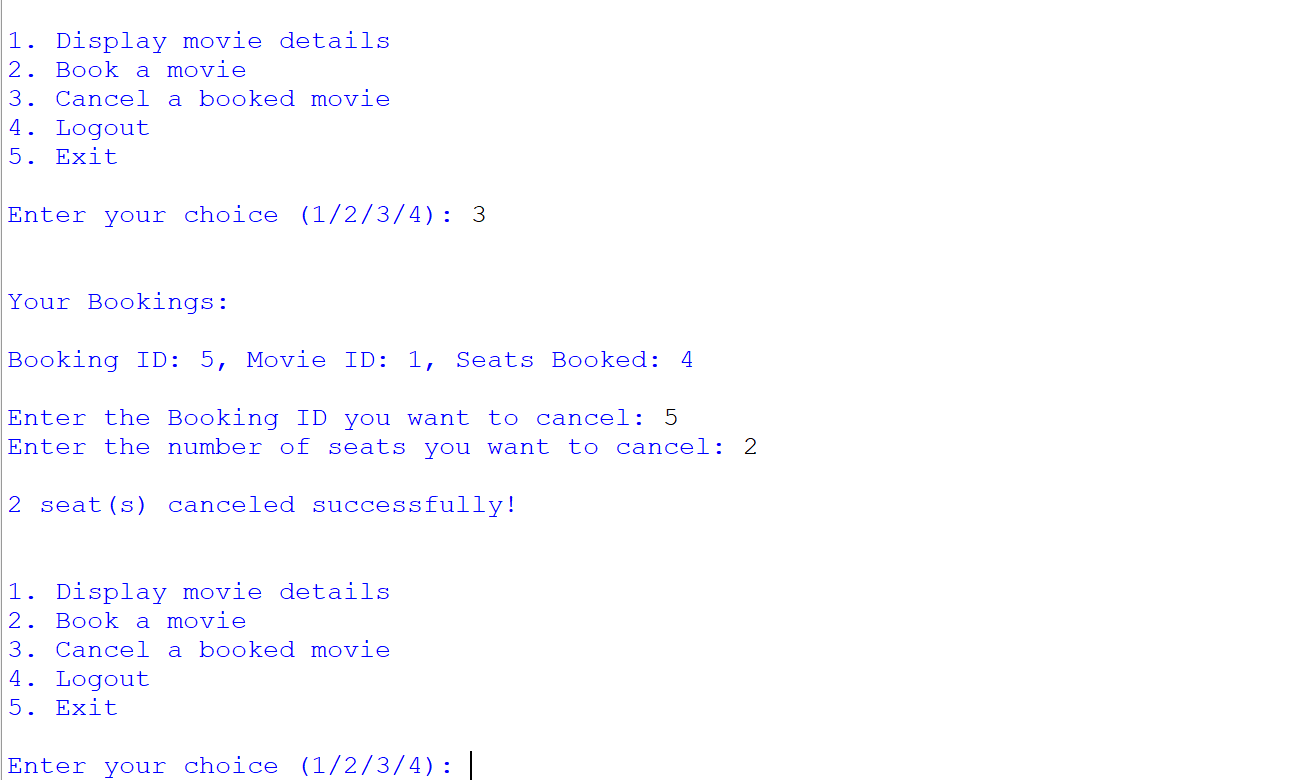
PROGRAM FOR THE USER TO DISPLAY MOVIE DETAILS



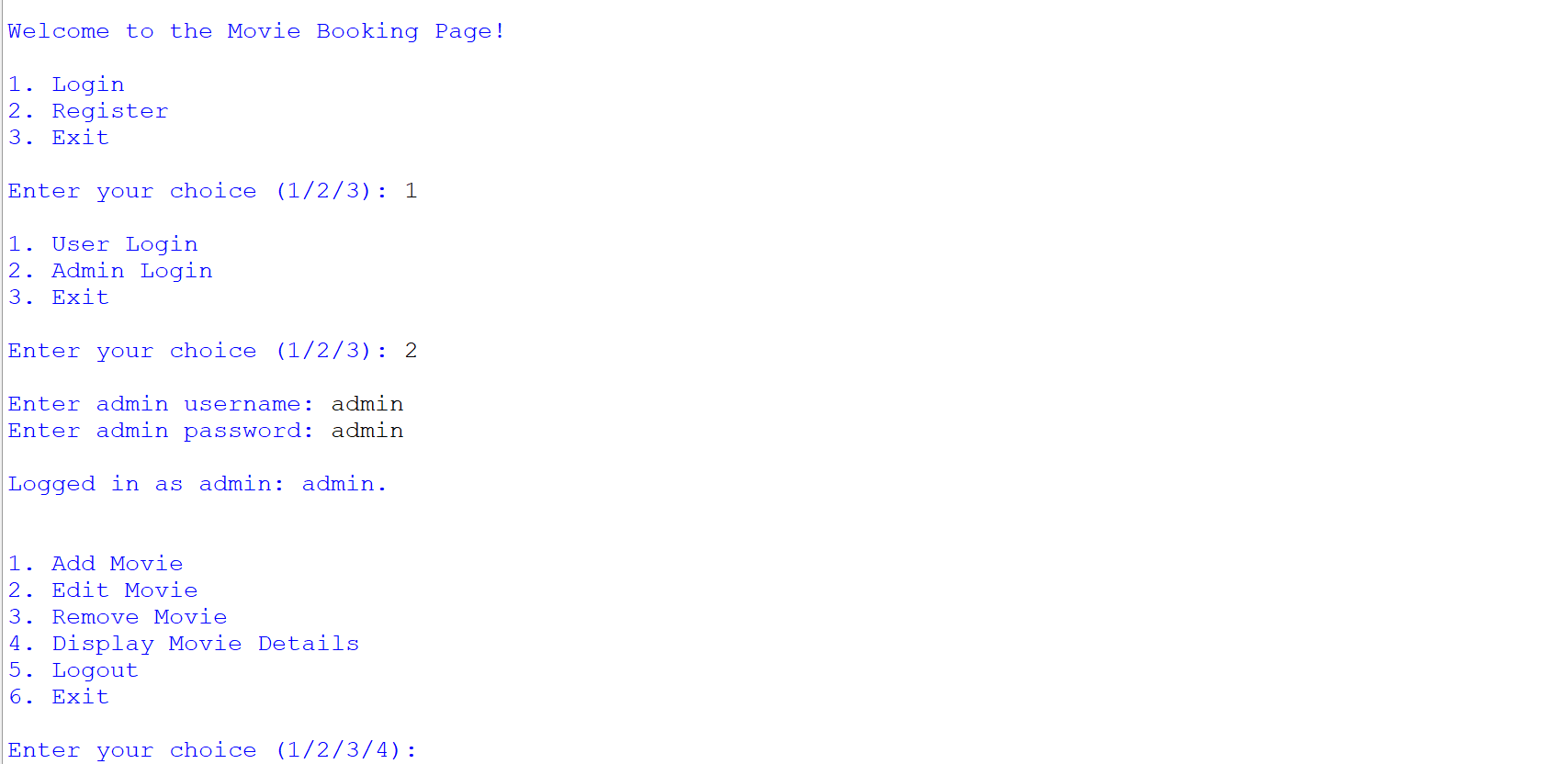
PROGRAM FOR THE USER TO BOOK A MOVIE



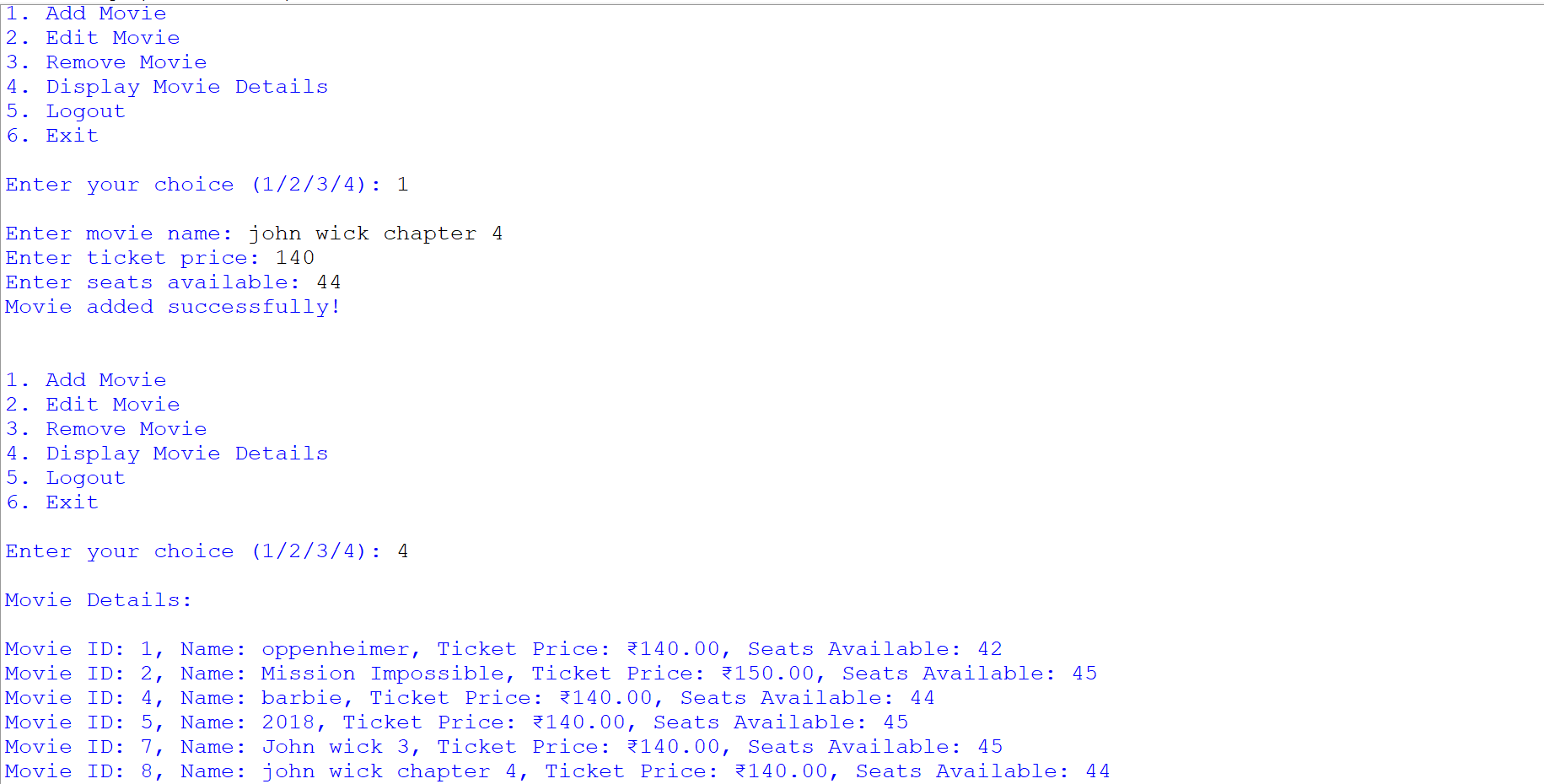
PROGRAM FOR THE USER TO CANCEL A BOOKED MOVIE



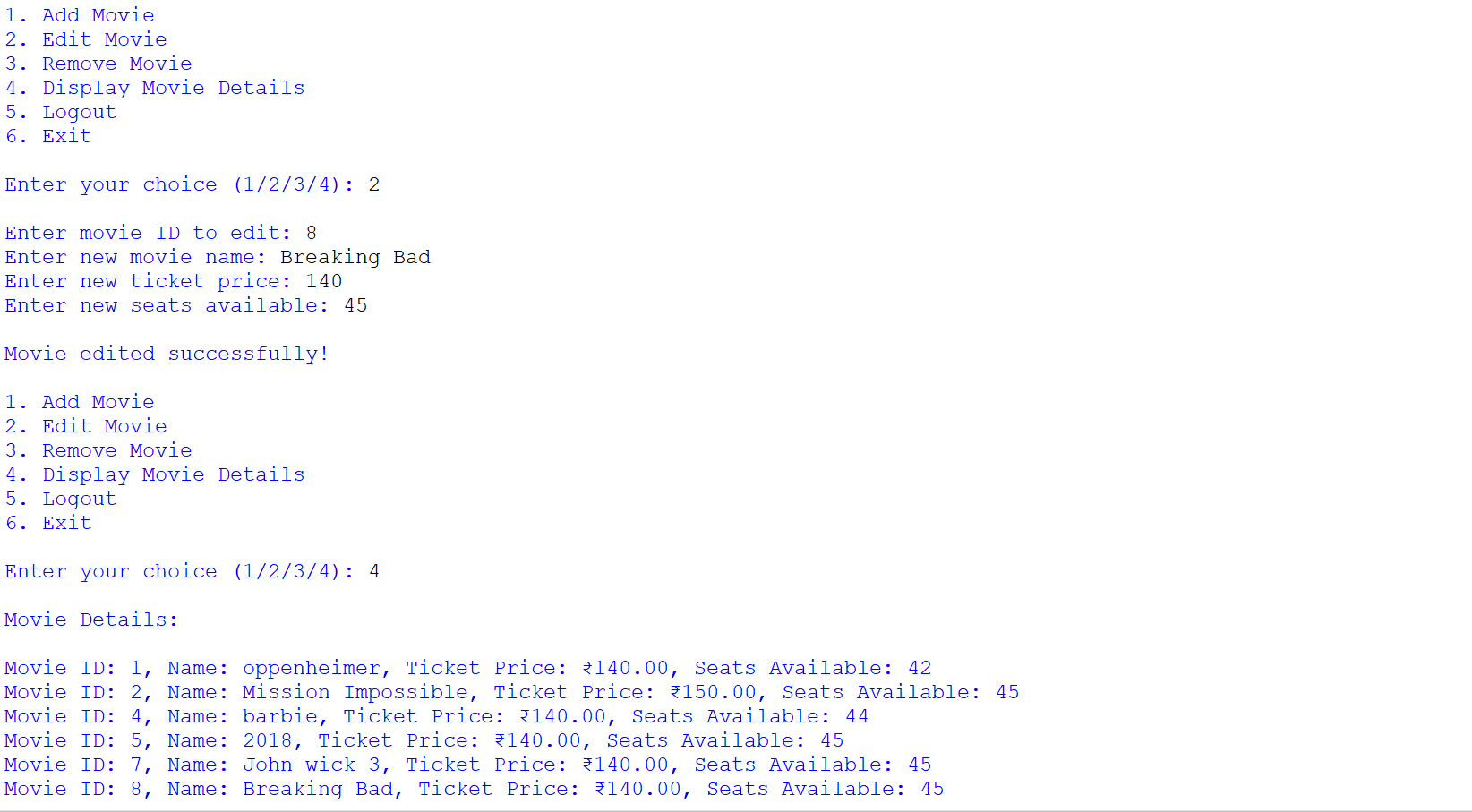
PROGRAM FOR THE ADMIN TO LOG IN ADMIN PAGE



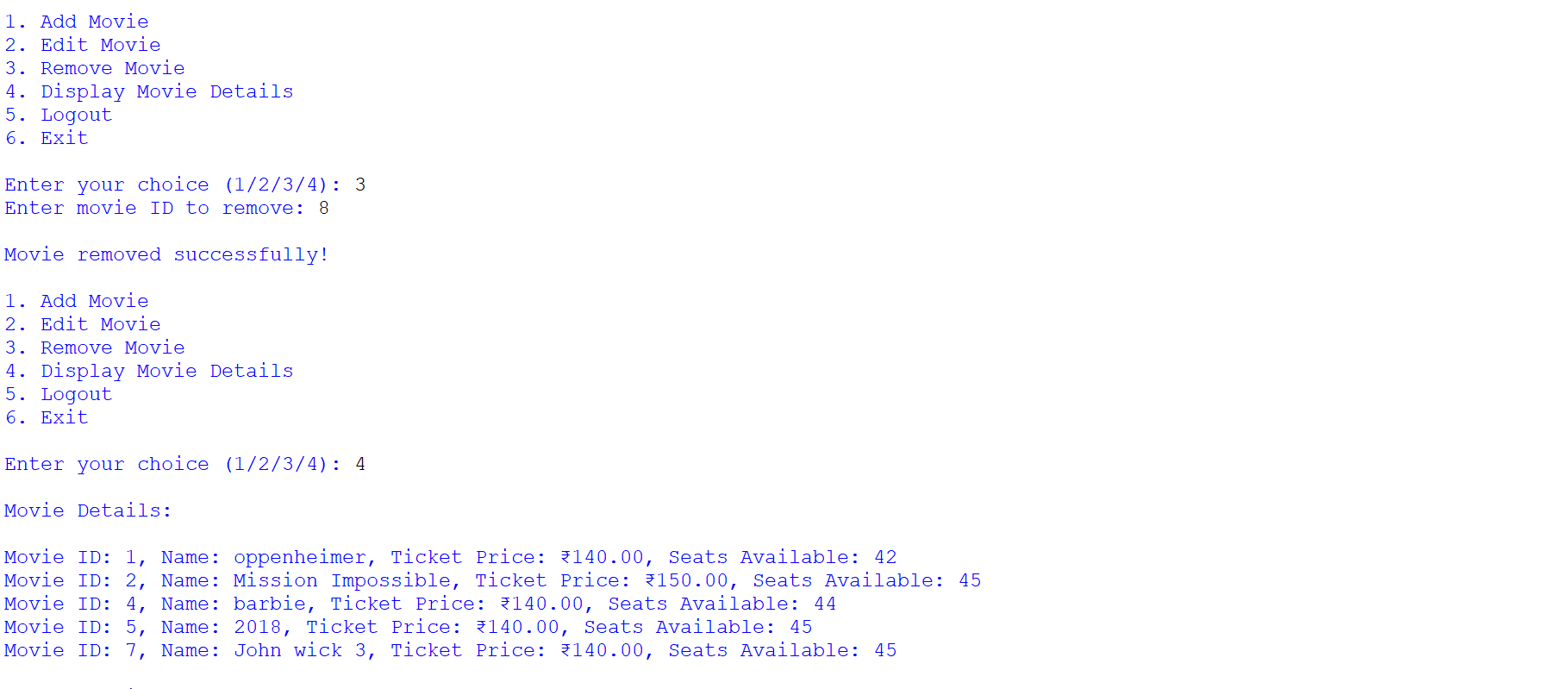
PROGRAM FOR THE ADMIN TO ADD A NEW MOVIE



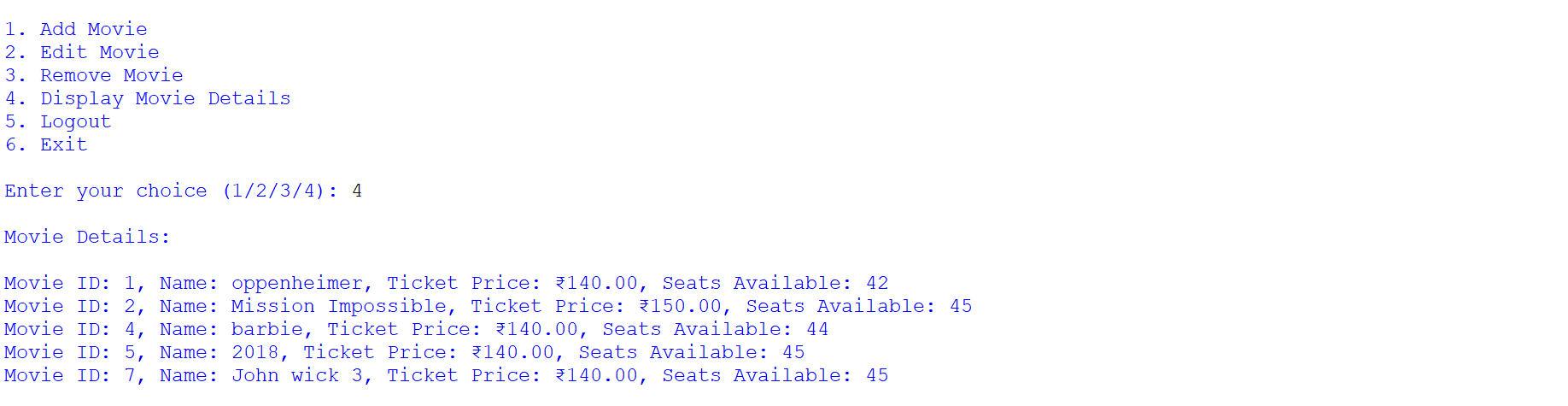
PROGRAM FOR THE ADMIN TO EDIT A MOVIE



PROGRAM FOR THE ADMIN TO REMOVE AN OLD MOVIE



PROGRAM FOR ADMIN TO CHECK ALL THE CHANGES MADE

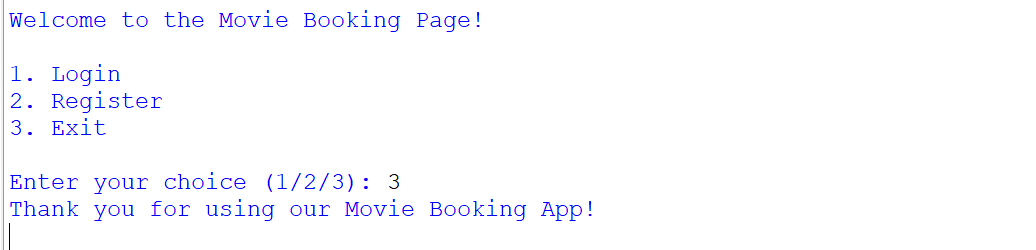


LOG OUT MESSAGE (BOTH USER AND ADMIN)





EXIT MESSAGE



BIBLIOGRAPHY

* Textbook for CBSE Class XII Computer Science with Python
* Textbook for CBSE Class XI Computer Science with Python
* [www.python.org](http://www.python.org)
* [www.geeksforgeeks.org](http://www.geeksforgeeks.org)
* [www.github.com](http://www.github.com)

INTRODUCTION

What does a normal family think do on a Sunday when everyone is at home bored? Of course going to the movie is a ideal answer for that question. A perfect way to spend free time, watching a new movie maybe one from your favorite actor/actress.

As we leap into the future we are able to book a movie from our comforts of our own home, which was not possible in the last decade.

This made us think and in response we decided to develop a ***Movie Booking System*** for need of the people who likes to watch movies as project for computer science.

OBJECTIVE

There are 2 main objective of the project one to for the admin and the other for user. The admin can add, remove, edit a movie while the user can book, cancel, display the movies.

The basic idea of the project is as follows.

First let’s take the user which *displays* the movies from which user can *book a movie* the specified number of seats or if they had already and they are not in mood to watch a movie they can *cancel* the number seats they had booked or ***remove*** extra seats if they wish.

Secondly the admin gets to *add* new movies, remove any old movie, and *edit* a movie if they made error while typing.

The program was developed with Python front end and MySQL as the rear end to store all needed information from user and admin.

CONCLUSION

This project has help us *in understanding databases and Python-MySQL connectivity* which are fundamental basics of programming.

Doing a project on *Movie Booking System*

Has helped and motivated us to create such programs to ease human life. The only limitations of this program is to add a movie manually.

We also understand the hard work, time and efforts put in by the programmers to write hundreds of lines of code to achieve something extraordinary.