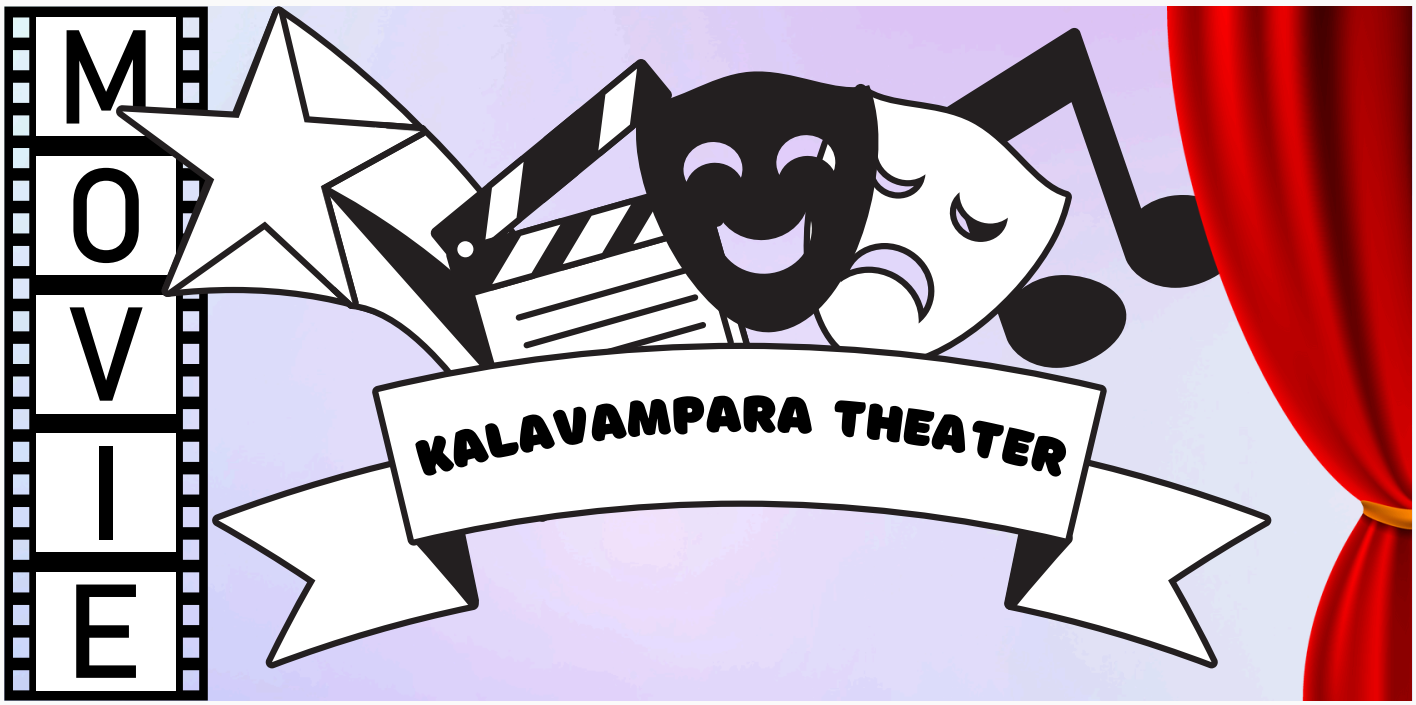
مـدرسـة أبوظبـي الـهنـديـة
Abu Dhabi Indian School
E-mail: adiscbse@emirates.net.ae | Web : www.adisuae.com
Tel : +9712 4488 025
Fax : +9712 4489 002
P.O. Box 46492
Abu Dhabi, U.A.E.



COMPUTER SCIENCE PROJECT

Name: Rijul Rajesh Menon

Class: 12A

GR. No.: 24944

CBSE Registration No.:

INDEX

ACKNOWLEDGEMENT

I would like to extend my heartfelt gratitude to Mrs. Lekshmi Sunil, my teacher, for granting me the invaluable opportunity to undertake this Computer Science project. Her support and guidance played an indispensable role in immersing me in extensive research and facilitating my acquisition of a multitude of new knowledge. Completing this project would have been a daunting task without her unwavering assistance.

Furthermore, I wish to express my appreciation to my friends, who generously dedicated their time to help me resolve any doubts, rectify errors, and ultimately assist me in successfully finishing the project. My intention in crafting this project was not solely to obtain good grades but also to apply the programming skills I have acquired in school to real life scenarios. Once again, I extend my heartfelt thanks to all those who contributed to the successful completion of this project.

KALAVAMPARA THEATER

Name: Rijul Rajesh Menon

Class: 12A

GR. No.: 24944

AIM OF THE PROJECT

The aim of this project is to develop an intuitive and user-friendly online theatre seat booking system using the Python inbuilt module, Tkinter. This system is designed to allow users to effortlessly browse and select their preferred movies, view available seating arrangements, and book seats according to their preferences in a seamless and efficient manner. By leveraging the capabilities of Tkinter, the project seeks to create a visually engaging interface that simplifies the booking process, enhances user experience, and optimizes the management of theatre seat reservations.

Ultimately, this project aspires to bridge the gap between movie enthusiasts and cinemas by providing an accessible platform that caters to the needs and conveniences of users, ensuring a smooth and enjoyable movie-going experience.

MODULES AND FUNCTIONS

MODULES

1. tkinter

This is the standard Python library used for creating graphical user interfaces (GUIs). It provides a wide range of tools to build desktop applications, including buttons, windows, labels, and other interactive elements.

2. tkinter.ttk

This is a submodule of tkinter that provides themed widgets. It includes advanced UI elements like Treeview, Notebook, Progressbar, etc., which look more modern than the default tkinter widgets.

3. random

This function from the random module is used to randomly select an item from a list or other sequence.

4. messagebox

This is used to display message boxes (alert, error, or informational messages) in the GUI. The message boxes can show messages to the user or ask for simple input.

5. webbrowser

This module allows the script to open web pages in a browser. It is typically used to automate the opening of URLs in the default web browser.

6. PIL (Pillow)

Pillow is an image-processing library. Image is used to open and manipulate images, while ImageTk is used to convert PIL images to a format that can be used in tkinter applications (e.g., to display images on the GUI).

7. mysql.connector

This module provides a Python interface to connect to a MySQL database. It allows the script to interact with MySQL databases, execute queries, and retrieve data.

8. turtle

The turtle module in Python is a popular tool for introducing basic graphics programming, especially for beginners. It provides a simple way to create drawings, animations, and shapes by controlling a "turtle" that moves around the screen. This turtle can be directed to move forward, backward, turn, draw lines, and change color, creating visual patterns or illustrations.

MAIN FUNCTIONS

1. WELCOMESCREEN(): The function uses Turtle graphics to draw a colorful rocket with propellers and a welcome message in a graphical window.
2. MENUSCREEN(): This function showcases the main window. Here, the user may login to book seats or use the admin login.

FOR ADMIN:

3. admin(): This function is only for the admin. It allows only one set of username and password.
4. details(): Here, each movie has its unique functions expressing the details of the users booked in that particular movie. And each movie again consists of functions like, search(), exit() and a table showcasing different users who have booked seats for that movie.

FOR CUSTOMERS:

5. loginscreen(): This function allows the users to login. It also allows new users to register. Then, the user is presented with three options - English, Hindi or Malayalam movies.
6. register(): This functions allows new users to register. The password, account id and customer name are saved to a customer table in movie database in mysql.
7. ENGLISH(): The user wants to watch an english film. He will be presented with two english films.

8. HINDI(): The user wants to watch an hindi film. He will be presented with two hindi films

9. MALAYALAM(): The user wants to watch an malayalam film. He will be presented with two malayalam films.

10. showtimes1() and showtimes2(): These are the two unique movies present in each of the languages i.e. there are a total of six movies. They are then directed to a page where all the details along with the trailer of the movie is given.

11. booking(): This function allows the user to book the number of seats and arrangement for his/her preferred movie.

12. snacks(): This functions showcases the different snacks the user may buy.

13. book(): The final bill and costing is given in this window along with the discounts included.

14. end(): This function is primary. This is the function which connects this python project to mysql database "movie" . The details of the customer along with the seats and snacks are saved to the unique table of that movie in mysql.

BRIEF DESCRIPTION OF THE PROJECT

The provided code represents a graphical user interface (GUI) application developed using the Tkinter library in Python. The project is an online booking space for movie enthusiasts and for anyone who wants to watch a movie. It creates a simpler and easier way for customers to explore various options.

Project Overview

1.The Welcome Screen:

The application starts with a welcoming screen that serves as the first point of interaction, featuring an attractive design that includes the application logo and a brief introduction, ensuring a user-friendly and an easy to use program.

2.The Main Screen:

The main screen is the hub of the application, where users can either login as a customer or admin. Customers can browse through a selection of currently available movies sorted in various languages. Each movie is represented with a poster thumbnail, title, and a brief synopsis to help users make informed choices.

3. Movie Details:

This section provides comprehensive information about the selected film, including a full synopsis, cast and crew details, duration, genre. Additionally, users can watch trailers, enhancing their decision-making experience.

4. Seat Booking:

To reserve a seat, simply choose the preferred seating option from the available layout. Whether you prefer an aisle, window, or center, our easy-to-use booking system allows you to select and confirm your choice in just a few clicks. Then, the customer may proceed to confirm their booking or even add some snacks too.

5. Snack Booking:

The Snack Booking System revolutionizes snack enjoyment for theater customers. Users can easily pre-order their favorite treats, ensuring they have exactly what they want during the show. With a wide selection of snacks and a straightforward booking process, ordering is hassle-free and convenient.

6. Admin Page:

From the Main Page, we an access the admin page by providing a set of username and its password. The admin page can view all the customers who booked a particular movie. We can also access the customer' s account id and password too.

7. Database Interaction:

The program connects to a MySQL database using the `mysql.connector` module to manage the customer details and movie details. It inserts and retrieves data related to mysql and complete orders.

8. Visual Elements:

The program uses various images, movie posters and trailers to enhance the visual appeal of the application.

9. Error Handling:

The application includes try-except blocks to handle potential errors during database connection and operations.

10. Account ID creation:

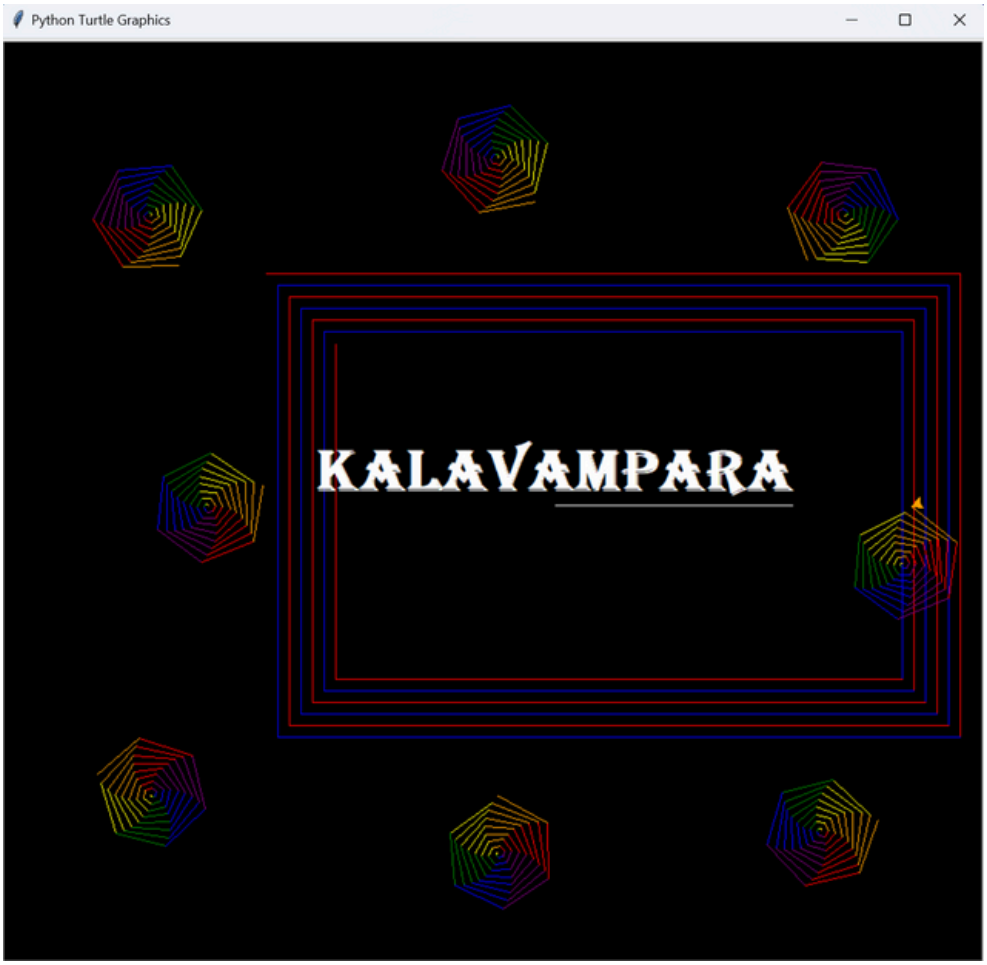
The account id creation for this site is very unique. It acts as the primary key for CUSTOMER table in movies database. Account id is created using the customer name and password. It concatenates the name and password, and saves it as their ASCII value.

Overall Purpose:

This platform aims to provide a seamless and user-friendly experience, allowing movie enthusiasts to browse through a wide selection of films, view showtimes, and select their preferred seating without any hassle. Our goal is to enhance the movie-going experience by offering convenient features such as secure payment options, and exclusive promotions. Whether you're planning a solo outing or a group event, our service ensures that booking your next cinematic adventure is just a few clicks away.

1.The Welcome Screen:

The application opens with a friendly welcome screen, which acts as the initial point of interaction. It boasts an appealing design that showcases the app logo along with a concise introduction, ensuring a user-friendly experience and an easy-to-navigate program. It consists of a colourful box and spirals of different colours with the Theater' s name.

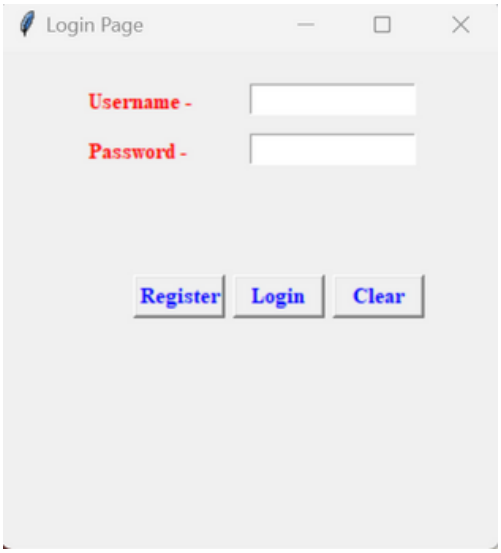


2.The Main Screen:

The main screen is the hub of the application, where users can either login as a customer or admin. The Admin login is in far top left while customer login in the top right corner. Customers can browse through a selection of currently available movies sorted in various languages.

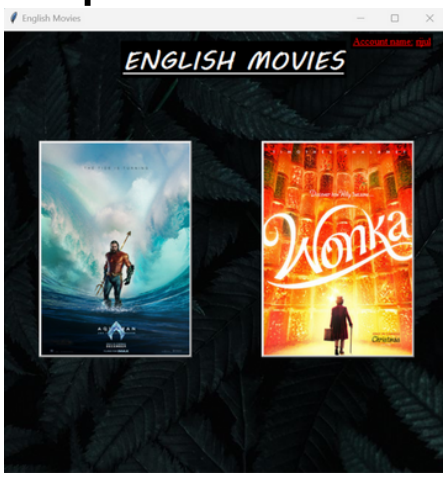
3.Login Page:

The login page is your secure and user-friendly entry point, featuring fields for your username and password. It is linked to mysql table CUSTOMER. New users may also register.



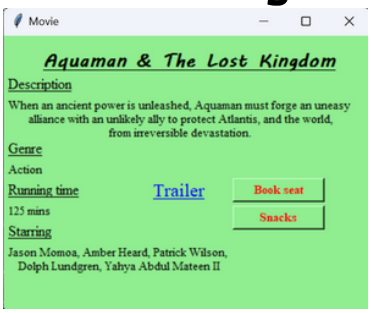
4.The Movie List:

Each of the languages shown in the main screen have two movies. The movies are shown with their official posters. The customer may choose a movie as per their wish.



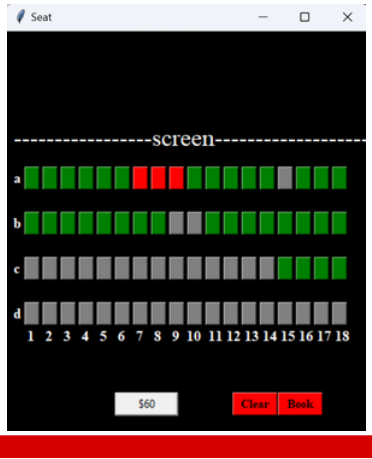
5.Movie Details:

Each movie is presented with its brief description, the genre of the movie, duration and cast of the film. A link is also attached, this link redirects the user to the official trailer of the movie. Then, user may book their seats according to availability and likings.



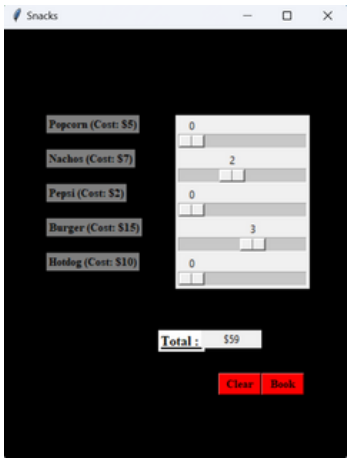
6.Seat Booking:

To ensure a hassle-free experience, plan ahead by checking schedules early to secure your preferred seats. Consider your seating preferences here. The Grey seats are the seats already booked. This data is gathered from mysql table of that movie. A clear button is also present which allows to clear all selected seats and book button to continue with the booking.



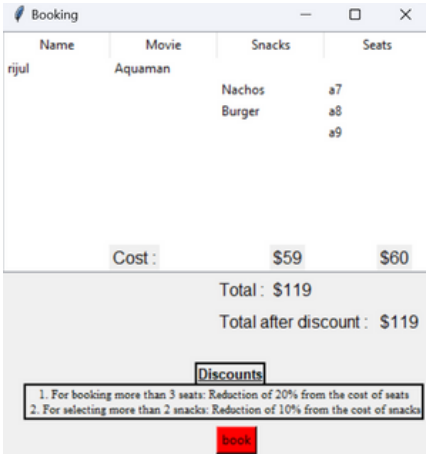
7.Snacks Booking:

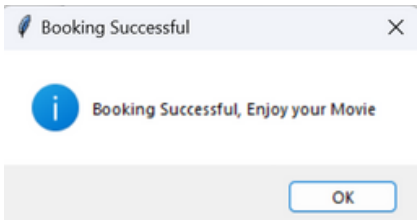
To ensure a comfortable experience, pre-order the snacks for the show. With vide range of food items, and lots of supply. A slider is in-cooperated for easier and simpler booking.



8.Final Receipt:

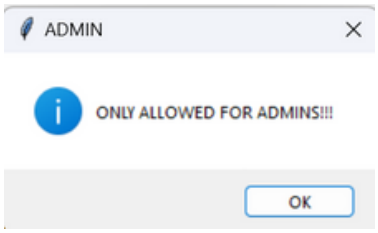
The receipt shows the total money along with the discounts available. It shows a descriptive detail of the order made of seats booking along with snacks ordered. Once the order is confirmed, the details are stored in mysql.

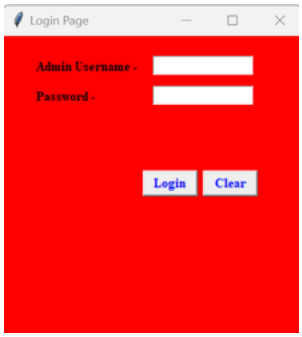




9.Admin Login:

From the Menu Screen, we can also login as Admin. First, it alerts the user that they will be trying to access admin. Only one certain set of Username and passcode can grant access.



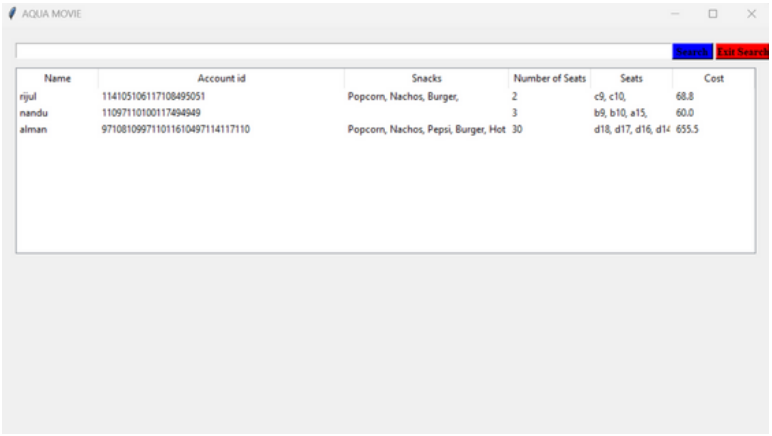


10.Admin Page:

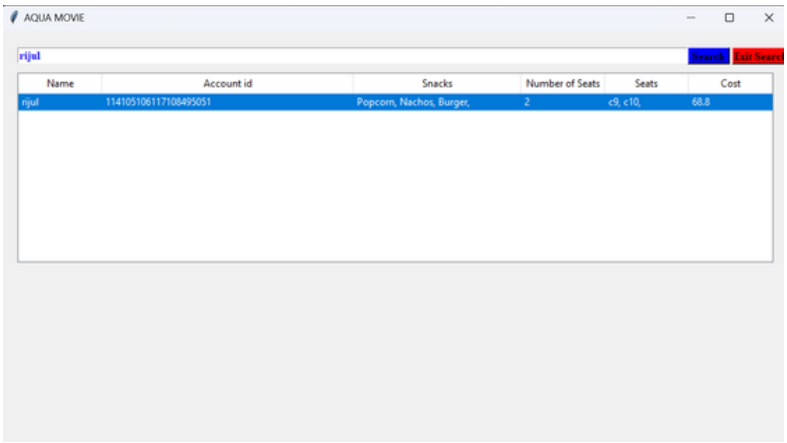
All the six available movies are shown. Each movie then showcases all the customers who have booked that particular movie.

11.Movie Admin:

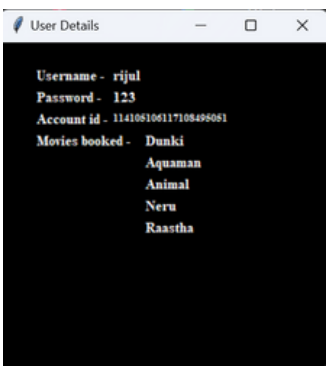
This window shows all the users which have booked in that particular movie. This data is retrieved from the mysql table of that movie table. There is a search bar which can used for easily finding a user. Additionally, if a user is clicked: It shows the entire details of the user like username, password, account id and all the movies booked by that particular user.



When user is searched:



When user is clicked:



USE OF TECHNOLOGY

1. MySQL

What is MySQL?

MySQL is an open-source relational database management system (RDBMS) that allows users to store, manage, and retrieve data efficiently. It is widely used for web applications, enterprise solutions, and other data-driven applications due to its reliability, flexibility, and compatibility with various programming languages, especially PHP, Java, and Python.

History of MySQL

MySQL was first released in 1995 by a Swedish company called MySQL AB, founded by David Axmark, Allan Larsson, and Michael Widenius. In 2008, MySQL was acquired by Sun Microsystems, and later, when Oracle Corporation acquired Sun in 2010, MySQL became part of Oracle. Despite concerns about corporate ownership, MySQL has continued to thrive as a popular open-source database.

Why is it Simple to Use?

MySQL is known for its simplicity and user-friendly interface, which allows both beginners and experienced developers to work with databases efficiently. It offers a straightforward SQL syntax, an intuitive command-line interface, and compatibility with graphical tools like MySQL Workbench, making database management easier and more accessible.

Reliability

MySQL is highly reliable and trusted for critical applications, from websites to enterprise systems. It offers data integrity, supports ACID (Atomicity, Consistency, Isolation, Durability) transactions, and provides features like replication and failover, which ensure high availability and prevent data loss.

Speed and Performance

MySQL is optimized for performance, handling large datasets and high transaction volumes efficiently. It is particularly known for its quick read operations, making it suitable for read-intensive applications. MySQL also supports indexing, partitioning, and other features that improve query speed and overall database performance.

Community Support

As an open-source software, MySQL has a large, active community of users and developers. This community provides extensive documentation, forums, and user-contributed resources, helping both new and advanced users troubleshoot issues and implement best practices. Regular updates and contributions from the community ensure that MySQL stays up-to-date and secure.

2. # Python

What is Python?

Python is a high-level, general-purpose programming language known for its readability and simplicity. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is widely used in web development, data science, artificial intelligence, and software engineering.

History of Python

Python was created in the late 1980s by Guido van Rossum and was first released in 1991. It was developed as a successor to the ABC programming language, with an emphasis on readability and ease of use. Over the years, Python has grown in popularity, especially with the rise of data science and machine learning, and it is now one of the most widely used programming languages in the world.

Why is it Simple to Use?

Python′ s syntax is clean, straightforward, and resembles natural language, making it easy for beginners to learn and use. Python eliminates the need for complex syntax seen in other languages, enabling faster development and reducing the likelihood of errors. Its extensive standard library simplifies tasks by providing pre-built functions for common operations.

Reliability

Python is a reliable language, widely used in various industries for both simple and complex applications. Its simplicity and extensive testing frameworks, like unittest and pytest, allow developers to build stable and maintainable applications. Python's community also rigorously reviews code, adding to its reliability.

Speed and Performance

While Python is not the fastest language in terms of raw execution speed, its performance is sufficient for most applications, and it supports high productivity, which often outweighs its slower runtime. For performance-critical tasks, Python can be combined with libraries written in faster languages, such as C and C++. Tools like PyPy (an alternative Python interpreter) can also improve performance.

Community Support

Python has a large and active community that continually develops new libraries, frameworks, and tools. The Python Software Foundation (PSF) maintains the language, ensuring it evolves while maintaining backward compatibility. The Python community provides extensive resources, including tutorials, documentation, forums, and conferences, which make learning and troubleshooting easy for developers.

SOURCE CODE

```
from tkinter import *
import tkinter as tk
from tkinter import ttk
from random import choice
from tkinter import messagebox
import webbrowser
from PIL import Image, ImageTk
import mysql.connector as mys
user1=""
i_d1=""
seluser=""
final=0
seatmoney=0
snckmoney=0
nseat=0
pop=0
nacho=0
pepsi=0
burger=0
hotdog=0
ver=0
end=0
n=[]
seataqua=[]
seatwonka=[]
seatdunki=[]
seatanimal=[]
seatneru=[]
seatraastha=[]


def MENUSCREEN():

    root = tk.Tk()
    root.geometry("1000x800")
    root.title(" O'BRIEN THEATER")
    img = ImageTk.PhotoImage(Image.open("theatre.jpg"))
    label=Label(root, image = img)
    label.pack()
    root.configure(bg='light green')
    lblmenu = tk.Label(root, text ="O'BRIEN THEATRE", fg= 'white', bg= 'black',font=('MV Boli',40,'bold'))
    lblmenu.place(x = 340, y = 15)
    lbllan = tk.Label(root, text ="Select your language preference:", bg= 'white',font=('MV Boli',20,
"underline"))
    lbllan.place(x = 320, y = 200)

    def admin():
        def funclear():
                txtUser.delete(0,END)
                txtpass.delete(0,END)
                txtUser.focus()
        def funlogin():
                if txtUser.get()=="adis" and txtpass.get()=="54321":
                    login.destroy()
                    root.destroy()
                    def movie():
                        mov = tk.Tk()
                        mov.geometry("500x600")
                        mov.title("ADMIN MOVIES")
                        mov.configure(bg='dark red')

                        def details():
                            userwin=tk.Tk()
                            userwin.geometry("300x300")
                            userwin.title("User Details")
                            userwin.configure(bg='black')
                            global seluser
                            lbluser = tk.Label(userwin, text ="Username -",fg="white",bg='black',font=("Times
New Roman Bold", 10) )
                            lbluser.place(x = 30, y = 20)
                            lblseluser = tk.Label(userwin, text =seluser,fg="white",bg='black',font=("Times
New Roman Bold", 10) )
                            lblseluser.place(x = 100, y = 20)
                            myconn=mys.connect(host="localhost",user="root",passwd="adis",database="movie")
                            mycur=myconn.cursor()
                            query="select * from customer where c_name='{}'".format(seluser)
                            mycur.execute(query)
                            rs=mycur.fetchone()
```

```
                            lblpass = tk.Label(userwin, text ="Password -",fg="white",bg='black',font=("Times
New Roman Bold", 10) )
                            lblpass.place(x = 30, y = 40)
                            lbluserpass = tk.Label(userwin, text =rs[2],fg="white",bg='black',font=("Times
New Roman Bold", 10) )
                            lbluserpass.place(x = 100, y = 40)
                            lblid = tk.Label(userwin, text ="Account id -",fg="white",bg='black',font=("Times
New Roman Bold", 10) )
                            lblid.place(x = 30, y = 60)
                            lblaccid = tk.Label(userwin, text =rs[1],fg="white",bg='black',font=("Times New
Roman Bold", 7) )
                            lblaccid.place(x = 100, y = 60)
                            lblmovie = tk.Label(userwin, text ="Movies booked
-",fg="white",bg='black',font=("Times New Roman Bold", 10) )
                            lblmovie.place(x = 30, y = 80)

                            z=0

                            myconn=mys.connect(host="localhost",user="root",passwd="adis",database="movie")
                            mycur=myconn.cursor()
                            query="select * from wonka where c_name='{}'".format(seluser)
                            mycur.execute(query)
                            rs1=mycur.fetchone()
                            if rs1!=None:
                                lblwonka = tk.Label(userwin, text ="Wonka",fg="white",bg='black',font=("Times
New Roman Bold", 10) )
                                lblwonka.place(x = 130, y = 80+z)
                                z+=20

                            myconn=mys.connect(host="localhost",user="root",passwd="adis",database="movie")
                            mycur=myconn.cursor()
                            query="select * from dunki where c_name='{}'".format(seluser)
                            mycur.execute(query)
                            rs2=mycur.fetchone()
                            if rs2!=None:
                                lbldunki = tk.Label(userwin, text ="Dunki",fg="white",bg='black',font=("Times
New Roman Bold", 10) )
                                lbldunki.place(x = 130, y = 80+z)
                                z+=20

                            myconn=mys.connect(host="localhost",user="root",passwd="adis",database="movie")
                            mycur=myconn.cursor()
                            query="select * from aquaman where c_name='{}'".format(seluser)
                            mycur.execute(query)
                            rs3=mycur.fetchone()
                            if rs3!=None:
                                lblaqua = tk.Label(userwin, text
="Aquaman",fg="white",bg='black',font=("Times New Roman Bold", 10) )
                                lblaqua.place(x = 130, y = 80+z)
                                z+=20

                            myconn=mys.connect(host="localhost",user="root",passwd="adis",database="movie")
                            mycur=myconn.cursor()
                            query="select * from animal where c_name='{}'".format(seluser)
                            mycur.execute(query)
                            rs4=mycur.fetchone()
                            if rs4!=None:
                                lblanimal = tk.Label(userwin, text
="Animal",fg="white",bg='black',font=("Times New Roman Bold", 10) )
                                lblanimal.place(x = 130, y = 80+z)
                                z+=20

                            myconn=mys.connect(host="localhost",user="root",passwd="adis",database="movie")
                            mycur=myconn.cursor()
                            query="select * from neru where c_name='{}'".format(seluser)
                            mycur.execute(query)
                            rs5=mycur.fetchone()
                            if rs5!=None:
                                lblneru = tk.Label(userwin, text ="Neru",fg="white",bg='black',font=("Times
New Roman Bold", 10) )
                                lblneru.place(x = 130, y = 80+z)
                                z+=20

                            myconn=mys.connect(host="localhost",user="root",passwd="adis",database="movie")
                            mycur=myconn.cursor()
                            query="select * from raastha where c_name='{}'".format(seluser)
                            mycur.execute(query)
                            rs6=mycur.fetchone()
                            if rs6!=None:
```

```
                                lblraastha = tk.Label(userwin, text
="Raastha",fg="white",bg='black',font=("Times New Roman Bold", 10) )
                                lblraastha.place(x = 130, y = 80+z)
                                z+=20


                            def wonka():
                                wonka = tk.Tk()
                                wonka.geometry("940x500")
                                wonka.title("WONKA MOVIE")

                                table = ttk.Treeview(wonka, columns = ('c_name', 'acc_id','snacks', 'number of
seats','seats','cost'), show = 'headings')
                                table.heading('c_name', text = 'Name')
                                table.heading('acc_id', text = 'Account id')
                                table.heading('snacks', text = 'Snacks')
                                table.heading('number of seats', text = 'Number of Seats')
                                table.heading('seats', text = 'Seats')
                                table.heading('cost', text = 'Cost')
                                table.place(x=20,y=50)

                                table.column('c_name', width=100)
                                table.column('acc_id', width=300)
                                table.column('snacks', width=200)
                                table.column('number of seats', width=100)
                                table.column('seats', width=100)
                                table.column('cost', width=100)

                                try:

myconn=mys.connect(host="localhost",user="root",passwd="adis",database="movie")
                                    mycur=myconn.cursor()
                                    query="select * from wonka"
                                    mycur.execute(query)
                                    rs=mycur.fetchall()

                                    def insert():
                                        n=len(rs)
                                        for i in range(0,n):
                                            table.insert(parent='', index=tk.END,
values=(rs[i][0],rs[i][1],rs[i][2],rs[i][3],rs[i][4],rs[i][5]))

                                    def search():
                                        x=srchbar.get()

myconn=mys.connect(host="localhost",user="root",passwd="adis",database="movie")
                                        mycur=myconn.cursor()
                                        query="select * from wonka where c_name='{}'".format(x)
                                        mycur.execute(query)
                                        rs=mycur.fetchall()
                                        if rs!=None:
                                            for item in table.get_children():
                                                table.delete(item)
                                            n=len(rs)
                                            for i in range(0,n):
                                                table.insert(parent='', index=tk.END,
values=(rs[i][0],rs[i][1],rs[i][2],rs[i][3],rs[i][4],rs[i][5]))

                                        else:
                                            messagebox.showinfo("Search Failed", "The entered user has not be
found")
                                    def exit():
                                        for item in table.get_children():
                                                table.delete(item)
                                        srchbar.delete(0,END)
                                        insert()
                                    insert()

                                    srchbar = tk.Entry(wonka, width = 800,fg="blue",font=("Times New Roman Bold",
10))
                                    srchbar.place(x = 20, y = 20, width = 800)
                                    btsrch= tk.Button(wonka, text ="Search",fg ='black', bg='blue', font=("Times
New Roman Bold", 10),command=search)
                                    btsrch.place(x = 820, y = 20, height = 20, width = 50)
                                    btexit= tk.Button(wonka, text ="Exit Search",fg ='black', bg='red',
font=("Times New Roman Bold", 10),command=exit)
                                    btexit.place(x = 873, y = 20, height = 20, width = 66)

                                    def on_select(event):
```

```
                                    global seluser
                                    seluser=""
                                    sel = table.selection()[0]
                                    selitems = table.item(sel)
                                    sellist = selitems['values']
                                    seluser+=sellist[0]
                                    details()

                                table.bind('<<TreeviewSelect>>', on_select)
                            except Exception as e:
                                print(e)

                        def aqua():
                            aqua = tk.Tk()
                            aqua.geometry("940x500")
                            aqua.title("AQUA MOVIE")

                            table = ttk.Treeview(aqua, columns = ('c_name', 'acc_id','snacks', 'number of
seats','seats','cost'), show = 'headings')
                            table.heading('c_name', text = 'Name')
                            table.heading('acc_id', text = 'Account id')
                            table.heading('snacks', text = 'Snacks')
                            table.heading('number of seats', text = 'Number of Seats')
                            table.heading('seats', text = 'Seats')
                            table.heading('cost', text = 'Cost')
                            table.place(x=20,y=50)

                            table.column('c_name', width=100)
                            table.column('acc_id', width=300)
                            table.column('snacks', width=200)
                            table.column('number of seats', width=100)
                            table.column('seats', width=100)
                            table.column('cost', width=100)

                            try:

myconn=mys.connect(host="localhost",user="root",passwd="adis",database="movie")
                                mycur=myconn.cursor()
                                query="select * from aquaman"
                                mycur.execute(query)
                                rs=mycur.fetchall()

                                def insert():
                                    n=len(rs)
                                    for i in range(0,n):
                                        table.insert(parent='', index=tk.END,
values=(rs[i][0],rs[i][1],rs[i][2],rs[i][3],rs[i][4],rs[i][5]))

                                def search():
                                    x=srchbar.get()

myconn=mys.connect(host="localhost",user="root",passwd="adis",database="movie")
                                    mycur=myconn.cursor()
                                    query="select * from aquaman where c_name='{}'".format(x)
                                    mycur.execute(query)
                                    rs=mycur.fetchall()
                                    if rs!=None:
                                        for item in table.get_children():
                                            table.delete(item)
                                        n=len(rs)
                                        for i in range(0,n):
                                            table.insert(parent='', index=tk.END,
values=(rs[i][0],rs[i][1],rs[i][2],rs[i][3],rs[i][4],rs[i][5]))

                                    else:
                                        messagebox.showinfo("Search Failed", "The entered user has not be
found")
                                def exit():
                                    for item in table.get_children():
                                            table.delete(item)
                                    srchbar.delete(0,END)
                                    insert()
                                insert()

                                srchbar = tk.Entry(aqua, width = 800,fg="blue",font=("Times New Roman Bold",
10))
                                srchbar.place(x = 20, y = 20, width = 800)
                                btsrch= tk.Button(aqua, text ="Search",fg ='black', bg='blue', font=("Times
New Roman Bold", 10),command=search)
```

```
                                btsrch.place(x = 820, y = 20, height = 20, width = 50)
                                btexit= tk.Button(aqua, text ="Exit Search",fg ='black', bg='red',
font=("Times New Roman Bold", 10),command=exit)
                                btexit.place(x = 873, y = 20, height = 20, width = 66)

                                def on_select(event):
                                    global seluser
                                    seluser=""
                                    sel = table.selection()[0]
                                    selitems = table.item(sel)
                                    sellist = selitems['values']
                                    seluser+=sellist[0]
                                    details()

                                table.bind('<<TreeviewSelect>>', on_select)
                            except Exception as e:
                                print(e)

                        def dunki():
                            dunki = tk.Tk()
                            dunki.geometry("940x500")
                            dunki.title("dunkiI MOVIE")

                            table = ttk.Treeview(dunki, columns = ('c_name', 'acc_id','snacks', 'number of
seats','seats','cost'), show = 'headings')
                            table.heading('c_name', text = 'Name')
                            table.heading('acc_id', text = 'Account id')
                            table.heading('snacks', text = 'Snacks')
                            table.heading('number of seats', text = 'Number of Seats')
                            table.heading('seats', text = 'Seats')
                            table.heading('cost', text = 'Cost')
                            #table.pack()
                            table.place(x=20,y=50)

                            table.column('c_name', width=100)
                            table.column('acc_id', width=300)
                            table.column('snacks', width=200)
                            table.column('number of seats', width=100)
                            table.column('seats', width=100)
                            table.column('cost', width=100)

                            try:

myconn=mys.connect(host="localhost",user="root",passwd="adis",database="movie")
                                mycur=myconn.cursor()
                                query="select * from dunki"
                                mycur.execute(query)
                                rs=mycur.fetchall()

                                def insert():
                                    n=len(rs)
                                    for i in range(0,n):
                                        table.insert(parent='', index=tk.END,
values=(rs[i][0],rs[i][1],rs[i][2],rs[i][3],rs[i][4],rs[i][5]))

                                def search():
                                    x=srchbar.get()

myconn=mys.connect(host="localhost",user="root",passwd="adis",database="movie")
                                    mycur=myconn.cursor()
                                    query="select * from dunki where c_name='{}'".format(x)
                                    mycur.execute(query)
                                    rs=mycur.fetchall()
                                    if rs!=None:
                                        for item in table.get_children():
                                            table.delete(item)
                                        n=len(rs)
                                        for i in range(0,n):
                                            table.insert(parent='', index=tk.END,
values=(rs[i][0],rs[i][1],rs[i][2],rs[i][3],rs[i][4],rs[i][5]))

                                    else:
                                        messagebox.showinfo("Search Failed", "The entered user has not be
found")
                                def exit():
                                    for item in table.get_children():
                                            table.delete(item)
                                    srchbar.delete(0,END)
                                    insert()
```

```
                                insert()

                                srchbar = tk.Entry(dunki, width = 800,fg="blue",font=("Times New Roman Bold",
10))
                                srchbar.place(x = 20, y = 20, width = 800)
                                btsrch= tk.Button(dunki, text ="Search",fg ='black', bg='blue', font=("Times
New Roman Bold", 10),command=search)
                                btsrch.place(x = 820, y = 20, height = 20, width = 50)
                                btexit= tk.Button(dunki, text ="Exit Search",fg ='black', bg='red',
font=("Times New Roman Bold", 10),command=exit)
                                btexit.place(x = 873, y = 20, height = 20, width = 66)

                                def on_select(event):
                                    global seluser
                                    seluser=""
                                    sel = table.selection()[0]
                                    selitems = table.item(sel)
                                    sellist = selitems['values']
                                    seluser+=sellist[0]
                                    details()

                                table.bind('<<TreeviewSelect>>', on_select)

                            except Exception as e:
                                print(e)

                        def animal():
                            animal = tk.Tk()
                            animal.geometry("940x500")
                            animal.title("animalAL MOVIE")

                            table = ttk.Treeview(animal, columns = ('c_name', 'acc_id','snacks', 'number of
seats','seats','cost'), show = 'headings')
                            table.heading('c_name', text = 'Name')
                            table.heading('acc_id', text = 'Account id')
                            table.heading('snacks', text = 'Snacks')
                            table.heading('number of seats', text = 'Number of Seats')
                            table.heading('seats', text = 'Seats')
                            table.heading('cost', text = 'Cost')
                            #table.pack()
                            table.place(x=20,y=50)

                            table.column('c_name', width=100)
                            table.column('acc_id', width=300)
                            table.column('snacks', width=200)
                            table.column('number of seats', width=100)
                            table.column('seats', width=100)
                            table.column('cost', width=100)

                            try:

myconn=mys.connect(host="localhost",user="root",passwd="adis",database="movie")
                                mycur=myconn.cursor()
                                query="select * from animal"
                                mycur.execute(query)
                                rs=mycur.fetchall()

                                def insert():
                                    n=len(rs)
                                    for i in range(0,n):
                                        table.insert(parent='', index=tk.END,
values=(rs[i][0],rs[i][1],rs[i][2],rs[i][3],rs[i][4],rs[i][5]))

                                def search():
                                    x=srchbar.get()

myconn=mys.connect(host="localhost",user="root",passwd="adis",database="movie")
                                    mycur=myconn.cursor()
                                    query="select * from animal where c_name='{}'".format(x)
                                    mycur.execute(query)
                                    rs=mycur.fetchall()
                                    if rs!=None:
                                        for item in table.get_children():
                                            table.delete(item)
                                        n=len(rs)
                                        for i in range(0,n):
                                            table.insert(parent='', index=tk.END,
values=(rs[i][0],rs[i][1],rs[i][2],rs[i][3],rs[i][4],rs[i][5]))
```

```
                                    else:
                                        messagebox.showinfo("Search Failed", "The entered user has not be
found")
                                def exit():
                                    for item in table.get_children():
                                            table.delete(item)
                                    srchbar.delete(0,END)
                                    insert()
                                insert()

                                srchbar = tk.Entry(animal, width = 800,fg="blue",font=("Times New Roman
Bold", 10))
                                srchbar.place(x = 20, y = 20, width = 800)
                                btsrch= tk.Button(animal, text ="Search",fg ='black', bg='blue', font=("Times
New Roman Bold", 10),command=search)
                                btsrch.place(x = 820, y = 20, height = 20, width = 50)
                                btexit= tk.Button(animal, text ="Exit Search",fg ='black', bg='red',
font=("Times New Roman Bold", 10),command=exit)
                                btexit.place(x = 873, y = 20, height = 20, width = 66)

                                def on_select(event):
                                    global seluser
                                    seluser=""
                                    sel = table.selection()[0]
                                    selitems = table.item(sel)
                                    sellist = selitems['values']
                                    seluser+=sellist[0]
                                    details()

                                table.bind('<<TreeviewSelect>>', on_select)

                            except Exception as e:
                                print(e)

                        def neru():
                            neru = tk.Tk()
                            neru.geometry("940x500")
                            neru.title("neru MOVIE")

                            table = ttk.Treeview(neru, columns = ('c_name', 'acc_id','snacks', 'number of
seats','seats','cost'), show = 'headings')
                            table.heading('c_name', text = 'Name')
                            table.heading('acc_id', text = 'Account id')
                            table.heading('snacks', text = 'Snacks')
                            table.heading('number of seats', text = 'Number of Seats')
                            table.heading('seats', text = 'Seats')
                            table.heading('cost', text = 'Cost')
                            #table.pack()
                            table.place(x=20,y=50)

                            table.column('c_name', width=100)
                            table.column('acc_id', width=300)
                            table.column('snacks', width=200)
                            table.column('number of seats', width=100)
                            table.column('seats', width=100)
                            table.column('cost', width=100)

                            try:

myconn=mys.connect(host="localhost",user="root",passwd="adis",database="movie")
                                mycur=myconn.cursor()
                                query="select * from neru"
                                mycur.execute(query)
                                rs=mycur.fetchall()

                                def insert():
                                    n=len(rs)
                                    for i in range(0,n):
                                        table.insert(parent='', index=tk.END,
values=(rs[i][0],rs[i][1],rs[i][2],rs[i][3],rs[i][4],rs[i][5]))

                                def search():
                                    x=srchbar.get()

myconn=mys.connect(host="localhost",user="root",passwd="adis",database="movie")
                                    mycur=myconn.cursor()
                                    query="select * from neru where c_name='{}'".format(x)
                                    mycur.execute(query)
                                    rs=mycur.fetchall()
```

```
                                    if rs!=None:
                                        for item in table.get_children():
                                            table.delete(item)
                                        n=len(rs)
                                        for i in range(0,n):
                                            table.insert(parent='', index=tk.END,
values=(rs[i][0],rs[i][1],rs[i][2],rs[i][3],rs[i][4],rs[i][5]))

                                    else:
                                        messagebox.showinfo("Search Failed", "The entered user has not be
found")
                                def exit():
                                    for item in table.get_children():
                                            table.delete(item)
                                    srchbar.delete(0,END)
                                    insert()
                                insert()

                                srchbar = tk.Entry(neru, width = 800,fg="blue",font=("Times New Roman Bold",
10))
                                srchbar.place(x = 20, y = 20, width = 800)
                                btsrch= tk.Button(neru, text ="Search",fg ='black', bg='blue', font=("Times
New Roman Bold", 10),command=search)
                                btsrch.place(x = 820, y = 20, height = 20, width = 50)
                                btexit= tk.Button(neru, text ="Exit Search",fg ='black', bg='red',
font=("Times New Roman Bold", 10),command=exit)
                                btexit.place(x = 873, y = 20, height = 20, width = 66)

                                def on_select(event):
                                    global seluser
                                    seluser=""
                                    sel = table.selection()[0]
                                    selitems = table.item(sel)
                                    sellist = selitems['values']
                                    seluser+=sellist[0]
                                    details()

                                table.bind('<<TreeviewSelect>>', on_select)

                            except Exception as e:
                                print(e)

                        def raastha():
                            raastha = tk.Tk()
                            raastha.geometry("940x500")
                            raastha.title("RAASTHA MOVIE")

                            table = ttk.Treeview(raastha, columns = ('c_name', 'acc_id','snacks', 'number of
seats','seats','cost'), show = 'headings')
                            table.heading('c_name', text = 'Name')
                            table.heading('acc_id', text = 'Account id')
                            table.heading('snacks', text = 'Snacks')
                            table.heading('number of seats', text = 'Number of Seats')
                            table.heading('seats', text = 'Seats')
                            table.heading('cost', text = 'Cost')
                            #table.pack()
                            table.place(x=20,y=50)

                            table.column('c_name', width=100)
                            table.column('acc_id', width=300)
                            table.column('snacks', width=200)
                            table.column('number of seats', width=100)
                            table.column('seats', width=100)
                            table.column('cost', width=100)

                            try:

myconn=mys.connect(host="localhost",user="root",passwd="adis",database="movie")
                                mycur=myconn.cursor()
                                query="select * from raastha"
                                mycur.execute(query)
                                rs=mycur.fetchall()

                                def insert():
                                    n=len(rs)
                                    for i in range(0,n):
                                        table.insert(parent='', index=tk.END,
values=(rs[i][0],rs[i][1],rs[i][2],rs[i][3],rs[i][4],rs[i][5]))
```

```
                                def search():
                                    x=srchbar.get()

myconn=mys.connect(host="localhost",user="root",passwd="adis",database="movie")
                                    mycur=myconn.cursor()
                                    query="select * from raastha where c_name='{}'".format(x)
                                    mycur.execute(query)
                                    rs=mycur.fetchall()
                                    if rs!=None:
                                        for item in table.get_children():
                                            table.delete(item)
                                        n=len(rs)
                                        for i in range(0,n):
                                            table.insert(parent='', index=tk.END,
values=(rs[i][0],rs[i][1],rs[i][2],rs[i][3],rs[i][4],rs[i][5]))

                                    else:
                                        messagebox.showinfo("Search Failed", "The entered user has not be
found")
                                def exit():
                                    for item in table.get_children():
                                            table.delete(item)
                                    srchbar.delete(0,END)
                                    insert()
                                insert()

                                srchbar = tk.Entry(raastha, width = 800,fg="blue",font=("Times New Roman
Bold", 10))
                                srchbar.place(x = 20, y = 20, width = 800)
                                btsrch= tk.Button(raastha, text ="Search",fg ='black', bg='blue',
font=("Times New Roman Bold", 10),command=search)
                                btsrch.place(x = 820, y = 20, height = 20, width = 50)
                                btexit= tk.Button(raastha, text ="Exit Search",fg ='black', bg='red',
font=("Times New Roman Bold", 10),command=exit)
                                btexit.place(x = 873, y = 20, height = 20, width = 66)

                                def on_select(event):
                                    global seluser
                                    seluser=""
                                    sel = table.selection()[0]
                                    selitems = table.item(sel)
                                    sellist = selitems['values']
                                    seluser+=sellist[0]
                                    details()

                                table.bind('<<TreeviewSelect>>', on_select)

                            except Exception as e:
                                print(e)

                        btwonka = tk.Button(mov, text ="Wonka",fg ='red',bg= 'orange',font=("Segoe Script
Bold", 20), command=wonka)
                        btwonka.place(x = 15, y = 50, height = 100, width = 180)
                        btaqua = tk.Button(mov, text ="Aquaman",fg ='red',bg= 'orange',font=("Segoe Script
Bold", 20), command=aqua)
                        btaqua.place(x = 300, y = 50, height = 100, width = 180)
                        btdunki = tk.Button(mov, text ="Dunki",fg ='red',bg= 'orange',font=("Segoe Script
Bold", 20), command=dunki)
                        btdunki.place(x = 15, y = 250, height = 100, width = 180)
                        btanimal = tk.Button(mov, text ="Animal",fg ='red',bg= 'orange',font=("Segoe Script
Bold", 20), command=animal)
                        btanimal.place(x = 300, y = 250, height = 100, width = 180)
                        btneru = tk.Button(mov, text ="Neru",fg ='red',bg= 'orange',font=("Segoe Script
Bold", 20), command=neru)
                        btneru.place(x = 15, y = 450, height = 100, width = 180)
                        btraastha = tk.Button(mov, text ="Raastha",fg ='red',bg= 'orange',font=("Segoe Script
Bold", 20), command=raastha)
                        btraastha.place(x = 300, y = 450, height = 100, width = 180)
                    movie()
                else:
                    messagebox.showinfo("Login Failed", "The entered info for admin is incorrect")


        messagebox.showinfo("ADMIN", "ONLY ALLOWED FOR ADMINS!!!")
        login = tk.Tk()
        login.geometry("300x300")
        login.title("Login Page")
        login.configure(bg='red')
        C = Canvas(login, bg ="Red", height = 250, width = 300)
```

```
        lbluser = tk.Label(login, text ="Admin Username -",fg="black",bg='red',font=("Times New Roman Bold",
10) )
        lbluser.place(x = 30, y = 20)
        txtUser = tk.Entry(login, width = 35,fg="blue",font=("Times New Roman Bold", 10))
        txtUser.place(x = 150, y = 20, width = 100)
        lblpass = tk.Label(login, text ="Password -",fg="black", bg='red',font=("Times New Roman Bold", 10))
        lblpass.place(x = 30, y = 50)
        txtpass = tk.Entry(login,show="*", width = 35,fg="blue",font=("Times New Roman Bold", 10))
        txtpass.place(x = 150, y = 50, width = 100)
        loginbtn = tk.Button(login, text ="Login", fg ='blue', font=("Times New Roman Bold", 10),command =
funlogin)
        loginbtn.place(x = 140, y = 135, width = 55)
        clearbtn = tk.Button(login, text ="Clear",fg ='blue', font=("Times New Roman Bold", 10),command =
funclear)
        clearbtn.place(x = 200, y = 135, width = 55)


    def English():
        root.destroy()
        frmeng=tk.Tk()
        frmeng.geometry("600x600")
        frmeng.title("English Movies")
        frmeng.configure(bg='black')
        def loginscreen():
            def funclear():
                txtUser.delete(0,END)
                txtpass.delete(0,END)
                txtUser.focus()

            def funlogin():
                user = txtUser.get()
                passw = txtpass.get()
                try:
                    myconn=mys.connect(host="localhost",user="root",passwd="adis",database="movie")
                    mycur=myconn.cursor()
                    query="select * from customer where c_name='{}' and pass='{}'".format(user,passw)
                    mycur.execute(query)
                    rs=mycur.fetchone()
                    if rs!=None and user==rs[0] and passw==rs[2]:
                        global ver
                        ver=1
                        global user1
                        user1=user
                        passw=txtpass.get()
                        l=user+passw
                        global i_d1
                        for ch in l:
                            i_d1+=str(ord(ch))

                        messagebox.showinfo("Login ", "Login Successful")
                        login.destroy()
                        btlogin.destroy()
                        lblacc = tk.Label(frmeng, text ="Account name:", fg ='red', bg= 'black',font=('Times
New Roman',10, "underline"))
                        lblacc.place(x = 470, y = 5)
                        lblus = tk.Label(frmeng, text =user1, fg ='red', bg= 'black',font=('Times New
Roman',10, "underline"))
                        lblus.place(x = 555, y = 5)
                    else:
                        messagebox.showinfo("Login", "Login Failed")
                except Exception as e:
                    print(e)

            def register():
                i_d=''
                user=txtUser.get()
                passw=txtpass.get()
                l=user+passw
                for ch in l:
                    i_d+=str(ord(ch))
                try:
                    myconn=mys.connect(host="localhost",user="root",passwd="adis",database="movie")
                    mycur=myconn.cursor()
                    query="insert into customer values('{}',{},'{}')".format(user,i_d,passw)
                    mycur.execute(query)
                    myconn.commit()
                    messagebox.showinfo("Registration", "Registration Successful, you may Login now")
                    funclear()
                except Exception as e:
```

```
                messagebox.showinfo("Registration", "Registration Successful, you may Login now")


        login = tk.Tk()
        login.geometry("300x300")
        login.title("Login Page")
        C = Canvas(login, bg ="Red", height = 250, width = 300)
        lbluser = tk.Label(login, text ="Username -",fg="red",font=("Times New Roman Bold", 10) )
        lbluser.place(x = 50, y = 20)
        txtUser = tk.Entry(login, width = 35,fg="blue",font=("Times New Roman Bold", 10))
        txtUser.place(x = 150, y = 20, width = 100)
        lblpass = tk.Label(login, text ="Password -",fg="red",font=("Times New Roman Bold", 10))
        lblpass.place(x = 50, y = 50)
        txtpass = tk.Entry(login,show="*", width = 35,fg="blue",font=("Times New Roman Bold", 10))
        txtpass.place(x = 150, y = 50, width = 100)
        loginbtn = tk.Button(login, text ="Login", fg ='blue', font=("Times New Roman Bold", 10),command
= funlogin)
        loginbtn.place(x = 140, y = 135, width = 55)
        regbtn = tk.Button(login, text ="Register", fg ='blue', font=("Times New Roman Bold", 10),command
= register)
        regbtn.place(x = 80, y = 135, width = 55)
        clearbtn = tk.Button(login, text ="Clear",fg ='blue', font=("Times New Roman Bold", 10),command =
funclear)
        clearbtn.place(x = 200, y = 135, width = 55)

    def showtimes1():
        frmtime=tk.Tk()
        frmtime.geometry("400x300")
        frmtime.title("Movie")
        frmtime.configure(bg='light green')
        def end():
            global nseat,final
            global user1, i_d1
            seat1=snacks1=''
            for ele in seataqua:
                seat1+= ele + ', '
            for ele in n:
                snacks1+= ele + ', '

            try:
                myconn=mys.connect(host="localhost",user="root",passwd="adis",database="movie")
                mycur=myconn.cursor()

                query = "INSERT INTO aquaman (c_name, acc_id, snacks, `number of seats`, seats, cost)
VALUES ('{}', '{}', '{}', {}, '{}',{})".format(user1, i_d1, snacks1, nseat, seat1,final)
                mycur.execute(query)
                myconn.commit()
                messagebox.showinfo("Booking Successful", "Booking Successful, Enjoy your Movie")

            except Exception as e:
                print(e)

        def book():
            global nseat
            if nseat>=1 and ver==1:
                book = tk.Tk()
                book.geometry("400x400")
                book.title("Booking")

                global user1
                global pop,nacho,pepsi,burger,hotdog

                btb=Button(book,text="book",fg='black',bg='red', command=end)
                btb.place(x=200,y=370)

                table = ttk.Treeview(book, columns = ('name', 'movie','snack', 'seat'), show =
'headings')
                table.heading('name', text = 'Name')
                table.heading('movie', text = 'Movie')
                table.heading('snack', text = 'Snacks')
                table.heading('seat', text = 'Seats')
                table.pack()

                table.column('name', width=100)
                table.column('movie', width=100)
                table.column('snack', width=100)
                table.column('seat', width=100)

                if pop==1:
```

```
                    n.append('Popcorn')
                if nacho==1:
                    n.append('Nachos')
                if pepsi==1:
                    n.append('Pepsi')
                if burger==1:
                    n.append('Burger')
                if hotdog==1:
                    n.append('Hotdog')

                name = user1
                movie = "Aquaman"
                table.insert(parent='', index=tk.END, values=(name, movie,'',''))

                max_items = max(len(n), len(seataqua))
                for i in range(max_items):
                    snack_value = n[i] if i < len(n) else ''
                    seat_value = seataqua[i] if i < len(seataqua) else ''
                    table.insert(parent='', index=tk.END, values=('','',snack_value, seat_value))

                global seatmoney,snckmoney,final
                total=seatmoney+snckmoney
                t=pop+nacho+pepsi+burger+hotdog
                d1=d2=0
                if nseat>3 and t>2:
                    d1=seatmoney-(0.2*seatmoney)
                    d2=snckmoney-(0.1*snckmoney)
                    final=d1+d2
                #Discount 1 : 10% off total if seats are greater than 3
                elif nseat>3:
                    d1=seatmoney-(0.2*seatmoney)
                    final=d1+snckmoney
                #Discount 2 : 10% off snckmoney if snacks are greater than 2
                elif t>2:
                    d2=snckmoney-(0.1*snckmoney)
                    final=d2+seatmoney
                else:
                    final=total

                lblcost = tk.Label(book, text ="Cost :",fg="black",font=("Arial", 12))
                lblcost.place(x = 100, y = 200)
                lbl1 = tk.Label(book, text ="$"+str(snckmoney),fg="black",font=("Arial", 12))
                lbl1.place(x = 250, y = 200)
                lbl2 = tk.Label(book, text ="$"+str(seatmoney),fg="black",font=("Arial", 12))
                lbl2.place(x = 350, y = 200)
                lbltot = tk.Label(book, text ="Total :",fg="black",font=("Arial", 12))
                lbltot.place(x = 200, y = 230)
                lbl3 = tk.Label(book, text ="$"+str(total),fg="black",font=("Arial", 12))
                lbl3.place(x = 250, y = 230)
                lblfinal = tk.Label(book, text ="Total after discount :",fg="black",font=("Arial", 12))
                lblfinal.place(x = 200, y = 260)
                lbl4 = tk.Label(book, text ="$"+str(final),fg="black",font=("Arial", 12))
                lbl4.place(x = 350, y = 260)
                lbldis = tk.Label(book, text ="Discounts",fg="black", borderwidth=2, relief='solid',
font=("Arial", 10,'underline','bold'))
                lbldis.place(x = 180, y = 310)
                lbldis1 = tk.Label(book, text ="1. For booking more than 3 seats: Reduction of 20% from
the cost of seats\n 2. For selecting more than 2 snacks: Reduction of 10% from the cost of
snacks",fg="black", borderwidth=2, relief='solid', font=("Times New Roman", 8))
                lbldis1.place(x = 20, y = 330)


            elif nseat<1:
                    messagebox.showinfo("Booking failed ", "Please Book Seats")
            elif ver==0:
                    messagebox.showinfo("Booking failed ", "Please Login First")

        def snacks():
            global snckmoney
            snack_quantities = {'Popcorn': 0, 'Nachos': 0, 'Pepsi': 0, 'Burger': 0, 'Hotdog': 0}
            def update_display():
                global snckmoney
                snckmoney = sum(snack_quantities[snack] * prices[snack] for snack in snacks_list)
                b1.config(text="$" + str(snckmoney))

            def clear():
                for snack in snacks_list:
                    scale_vars[snack].set(0)
                    snack_quantities[snack] = 0
```

```
                    update_display()

                def on_scale_change(event):
                    snack = event.widget.snack
                    snack_quantities[snack] = scale_vars[snack].get()
                    update_display()
                    global pop,hotdog,nacho,pepsi,burger
                    if snack == 'Popcorn' and snack_quantities[snack] > 0:
                        pop = 1
                    if snack == 'Hotdog' and snack_quantities[snack] > 0:
                        hotdog = 1
                    if snack == 'Burger' and snack_quantities[snack] > 0:
                        burger = 1
                    if snack == 'Pepsi' and snack_quantities[snack] > 0:
                        pepsi = 1
                    if snack == 'Nachos' and snack_quantities[snack] > 0:
                        nacho = 1

                frmsnck = tk.Tk()
                frmsnck.geometry("400x500")
                frmsnck.title("Snacks")
                frmsnck.configure(bg='black')

                btclr = tk.Button(frmsnck, text="Clear", fg='black', bg='Red', font=("Times New Roman Bold",
10), command=clear)
                btclr.place(x=250, y=400, width=50)

                btbook = tk.Button(frmsnck, text="Book", fg='black', bg='Red', font=("Times New Roman Bold",
10), command=book)
                btbook.place(x=300, y=400, width=50)

                lbltot = tk.Label(frmsnck, text="Total : ", fg='black', bg='white', font=('Times New Roman',
12, 'bold', 'underline'))
                lbltot.place(x=180, y=350)

                b1 = tk.Label(frmsnck, text="$" + str(snckmoney))
                b1.place(x=230, y=350, width=70)

                snacks_list = ['Popcorn', 'Nachos', 'Pepsi', 'Burger', 'Hotdog']
                prices = {'Popcorn': 5, 'Nachos': 7, 'Pepsi': 2, 'Burger': 15, 'Hotdog': 10}
                scale_vars = {}

                for idx, snack in enumerate(snacks_list):
                    lbl = tk.Label(frmsnck, text=f"{snack} (Cost: ${prices[snack]})", fg='black', bg='grey',
font=('Times New Roman', 10, 'bold'))
                    lbl.place(x=50, y=100 + idx * 40)

                    scale_vars[snack] = tk.Scale(frmsnck, from_=0, to=5, orient=tk.HORIZONTAL, length=150)
                    scale_vars[snack].place(x=200, y=100 + idx * 40)
                    scale_vars[snack].snack = snack
                    scale_vars[snack].bind("<ButtonRelease-1>", on_scale_change)

            def booking():
                def taken():
                    nonlocal buttons
                    try:
                        myconn=mys.connect(host="localhost",user="root",passwd="adis",database="movie")
                        mycur=myconn.cursor()

                        query = "select seats from aquaman"
                        mycur.execute(query)
                        result=mycur.fetchall()
                        booked_seats = []
                        for tup in result:
                            elements = [elem.strip() for elem in tup[0].split(',')]
                            booked_seats.extend([elem for elem in elements if elem])
                        myconn.commit()
                        for row in range(len(buttons)):
                            for col in range(len(buttons[row])):
                                seat_label = chr(ord('a') + row) + str(col + 1)
                                if seat_label in booked_seats:
                                    buttons[row][col].configure(bg="grey", state=tk.DISABLED)

                    except Exception as e:
                        print(e)

                def clear():
                    for button_row in buttons:
                        for button in button_row:
```

```
                        button.configure(bg="green")
                    global seatmoney, nseat
                    seatmoney = 0
                    nseat = 0
                    update_display()
                    seataqua.clear()

                def update_display():
                    b1.config(text="$" + str(seatmoney))

                def button_click(row, col):
                    global seatmoney, nseat
                    button = buttons[row][col]
                    seat_label = chr(ord('a') + row) + str(col + 1)
                    if button["bg"] == "green":
                        button.configure(bg="red")
                        seatmoney += 20
                        nseat += 1
                        seataqua.append(seat_label)

                    else:
                        button.configure(bg="green")
                        seatmoney -= 20
                        nseat -= 1
                        seataqua.remove(seat_label)

                    update_display()

                frmbook = tk.Tk()
                frmbook.geometry("400x450")
                frmbook.title("Seat")
                frmbook.configure(bg='black')
                lblscr = tk.Label(frmbook, text="-----------------screen-------------------", fg='white',
bg='black',
                                  font=('Times New Roman', 20))
                lblscr.place(x=5, y=100)

                btclr = tk.Button(frmbook, text="Clear", fg='black', bg='Red', font=("Times New Roman Bold",
10), command=clear)
                btclr.place(x=250, y=400, width=50)

                btbook = tk.Button(frmbook, text="Book", fg='black', bg='Red', font=("Times New Roman Bold",
10), command=book)
                btbook.place(x=300, y=400, width=50)

                b1 = tk.Button(frmbook, text="$" + str(seatmoney))
                b1.place(x=120, y=400, width=70)

                lbla = tk.Label(frmbook, text="a", fg='white', bg='black', font=("Times New Roman Bold", 12))
                lbla.place(x=5, y=150)
                lblb = tk.Label(frmbook, text="b", fg='white', bg='black', font=("Times New Roman Bold", 12))
                lblb.place(x=5, y=200)
                lblc = tk.Label(frmbook, text="c", fg='white', bg='black', font=("Times New Roman Bold", 12))
                lblc.place(x=5, y=250)
                lbld = tk.Label(frmbook, text="d", fg='white', bg='black', font=("Times New Roman Bold", 12))
                lbld.place(x=5, y=300)
                lblno = tk.Label(frmbook, text="1   2   3   4   5   6   7   8   9  10  11 12 13 14 15 16 17
18", fg='white', bg='black', font=("Times New Roman Bold", 12))
                lblno.place(x=20, y=325)

                buttons = []

                for row in range(4):
                    button_row = []
                    for col in range(18):
                        seat_label = chr(ord('a') + row) + str(col + 1)
                        button = tk.Button(frmbook, bg='green', height=1, width=1, command=lambda r=row,
c=col: button_click(r, c))
                        button.place(x=20 + col * 20, y=150 + row * 50)
                        button_row.append(button)
                    buttons.append(button_row)
                taken()

            btbook= tk.Button(frmtime, text ="Book seat",fg ='red',bg= 'light green',font=("Times New Roman
Bold", 10),command=booking)
            btbook.place(x = 250, y = 155, width = 100)
            btsnack= tk.Button(frmtime, text ="Snacks",fg ='red',bg= 'light green',font=("Times New Roman
Bold", 10),command=snacks)
            btsnack.place(x = 250, y = 185, width = 100)
```

```
            lblhd = tk.Label(frmtime, text ="Aquaman & The Lost Kingdom", bg= 'light green',font=('MV
Boli',15,'bold','underline'))
            lblhd.place(x = 40, y = 10)
            lbldescrpt = tk.Label(frmtime, text ="Description", bg= 'light green',font=('Times New Roman',12,
'underline'))
            lbldescrpt.place(x = 2, y = 40)
            lbldes = tk.Label(frmtime, text ="When an ancient power is unleashed, Aquaman must forge an
uneasy\n alliance with an unlikely ally to protect Atlantis, and the world,\n from irreversible
devastation.",
                              bg= 'light green',font=('Times New Roman',10))
            lbldes.place(x = 2, y = 65)
            lblgenre = tk.Label(frmtime, text ="Genre", bg= 'light green',font=('Times New Roman',12,
'underline'))
            lblgenre.place(x = 2, y = 110)
            lblgen = tk.Label(frmtime, text ="Action", bg= 'light green',font=('Times New Roman',10))
            lblgen.place(x = 2, y = 135)
            lblrt = tk.Label(frmtime, text ="Running time", bg= 'light green',font=('Times New Roman',12,
'underline'))
            lblrt.place(x = 2, y = 155)
            lblrun = tk.Label(frmtime, text ="125 mins", bg= 'light green',font=('Times New Roman',10))
            lblrun.place(x = 2, y = 180)
            lblstar = tk.Label(frmtime, text ="Starring", bg= 'light green',font=('Times New Roman',12,
'underline'))
            lblstar.place(x = 2, y = 200)
            lblact = tk.Label(frmtime, text ="Jason Momoa, Amber Heard, Patrick Wilson,\n Dolph Lundgren,
Yahya Abdul Mateen II", bg= 'light green',font=('Times New Roman',10))
            lblact.place(x = 2, y = 225)

            link = tk.Label(frmtime, text="Trailer",
            fg="blue", bg="light green", cursor="hand2",font=['Times',15,'underline'])
            link.place(x=160, y=155)
            link.bind("<Button-1>", lambda e:
webbrowser.open_new("https://youtu.be/UGc5Tzz19UY?si=sPxYM3b9lWpQ_j0H"))


        def showtimes2():
            frmtime=tk.Tk()
            frmtime.geometry("400x300")
            frmtime.title("Wonka")
            frmtime.configure(bg='light green')
            def end():
                global nseat,final
                global user1, i_d1
                seat1=snacks1=''
                for ele in seatwonka:
                    seat1+= ele + ', '
                for ele in n:
                    snacks1+= ele + ', '

                try:
                    myconn=mys.connect(host="localhost",user="root",passwd="adis",database="movie")
                    mycur=myconn.cursor()

                    query = "INSERT INTO wonka (c_name, acc_id, snacks, `number of seats`, seats, cost)
VALUES ('{}', '{}', '{}', {}, '{}',{})".format(user1, i_d1, snacks1, nseat, seat1,final)
                    mycur.execute(query)
                    myconn.commit()
                    messagebox.showinfo("Booking Successful", "Booking Successful, Enjoy your Movie")

                except Exception as e:
                    print(e)

            def book():
                global nseat
                if nseat>=1 and ver==1:
                    book = tk.Tk()
                    book.geometry("400x400")
                    book.title("Booking")

                    global user1
                    global pop,nacho,pepsi,burger,hotdog

                    btb=Button(book,text="book",fg='black',bg='red', command=end)
                    btb.place(x=200,y=370)

                    table = ttk.Treeview(book, columns = ('name', 'movie','snack', 'seat'), show =
'headings')
                    table.heading('name', text = 'Name')
                    table.heading('movie', text = 'Movie')
```

```
                    table.heading('snack', text = 'Snacks')
                    table.heading('seat', text = 'Seats')
                    table.pack()

                    table.column('name', width=100)
                    table.column('movie', width=100)
                    table.column('snack', width=100)
                    table.column('seat', width=100)

                    if pop==1:
                        n.append('Popcorn')
                    if nacho==1:
                        n.append('Nachos')
                    if pepsi==1:
                        n.append('Pepsi')
                    if burger==1:
                        n.append('Burger')
                    if hotdog==1:
                        n.append('Hotdog')

                    name = user1
                    movie = "Wonka"
                    table.insert(parent='', index=tk.END, values=(name, movie,'',''))

                    max_items = max(len(n), len(seatwonka))
                    for i in range(max_items):
                        snack_value = n[i] if i < len(n) else ''
                        seat_value = seatwonka[i] if i < len(seatwonka) else ''
                        table.insert(parent='', index=tk.END, values=('','',snack_value, seat_value))

                    global seatmoney,snckmoney,final
                    total=seatmoney+snckmoney
                    t=pop+nacho+pepsi+burger+hotdog
                    d1=d2=0
                    if nseat>3 and t>2:
                        d1=seatmoney-(0.2*seatmoney)
                        d2=snckmoney-(0.1*snckmoney)
                        final=d1+d2
                    #Discount 1 : 10% off total if seats are greater than 3
                    elif nseat>3:
                        d1=seatmoney-(0.2*seatmoney)
                        final=d1+snckmoney
                    #Discount 2 : 10% off snckmoney if snacks are greater than 2
                    elif t>2:
                        d2=snckmoney-(0.1*snckmoney)
                        final=d2+seatmoney
                    else:
                        final=total

                    lblcost = tk.Label(book, text ="Cost :",fg="black",font=("Arial", 12))
                    lblcost.place(x = 100, y = 200)
                    lbl1 = tk.Label(book, text ="$"+str(snckmoney),fg="black",font=("Arial", 12))
                    lbl1.place(x = 250, y = 200)
                    lbl2 = tk.Label(book, text ="$"+str(seatmoney),fg="black",font=("Arial", 12))
                    lbl2.place(x = 350, y = 200)
                    lbltot = tk.Label(book, text ="Total :",fg="black",font=("Arial", 12))
                    lbltot.place(x = 200, y = 230)
                    lbl3 = tk.Label(book, text ="$"+str(total),fg="black",font=("Arial", 12))
                    lbl3.place(x = 250, y = 230)
                    lblfinal = tk.Label(book, text ="Total after discount :",fg="black",font=("Arial", 12))
                    lblfinal.place(x = 200, y = 260)
                    lbl4 = tk.Label(book, text ="$"+str(final),fg="black",font=("Arial", 12))
                    lbl4.place(x = 350, y = 260)
                    lbldis = tk.Label(book, text ="Discounts",fg="black", borderwidth=2, relief='solid',
font=("Arial", 10,'underline','bold'))
                    lbldis.place(x = 180, y = 310)
                    lbldis1 = tk.Label(book, text ="1. For booking more than 3 seats: Reduction of 20% from
the cost of seats\n 2. For selecting more than 2 snacks: Reduction of 10% from the cost of
snacks",fg="black", borderwidth=2, relief='solid', font=("Times New Roman", 8))
                    lbldis1.place(x = 20, y = 330)


                elif nseat<1:
                        messagebox.showinfo("Booking failed ", "Please Book Seats")
                elif ver==0:
                        messagebox.showinfo("Booking failed ", "Please Login First")

            def snacks():
                global snckmoney
```

```
                snack_quantities = {'Popcorn': 0, 'Nachos': 0, 'Pepsi': 0, 'Burger': 0, 'Hotdog': 0}
                def update_display():
                    global snckmoney
                    snckmoney = sum(snack_quantities[snack] * prices[snack] for snack in snacks_list)
                    b1.config(text="$" + str(snckmoney))

                def clear():
                    for snack in snacks_list:
                        scale_vars[snack].set(0)
                        snack_quantities[snack] = 0
                    update_display()

                def on_scale_change(event):
                    snack = event.widget.snack
                    snack_quantities[snack] = scale_vars[snack].get()
                    update_display()
                    global pop,hotdog,nacho,pepsi,burger
                    if snack == 'Popcorn' and snack_quantities[snack] > 0:
                        pop = 1
                    if snack == 'Hotdog' and snack_quantities[snack] > 0:
                        hotdog = 1
                    if snack == 'Burger' and snack_quantities[snack] > 0:
                        burger = 1
                    if snack == 'Pepsi' and snack_quantities[snack] > 0:
                        pepsi = 1
                    if snack == 'Nachos' and snack_quantities[snack] > 0:
                        nacho = 1

                frmsnck = tk.Tk()
                frmsnck.geometry("400x500")
                frmsnck.title("Snacks")
                frmsnck.configure(bg='black')

                btclr = tk.Button(frmsnck, text="Clear", fg='black', bg='Red', font=("Times New Roman Bold",
10), command=clear)
                btclr.place(x=250, y=400, width=50)

                btbook = tk.Button(frmsnck, text="Book", fg='black', bg='Red', font=("Times New Roman Bold",
10), command=book)
                btbook.place(x=300, y=400, width=50)

                lbltot = tk.Label(frmsnck, text="Total : ", fg='black', bg='white', font=('Times New Roman',
12, 'bold', 'underline'))
                lbltot.place(x=180, y=350)

                b1 = tk.Label(frmsnck, text="$" + str(snckmoney))
                b1.place(x=230, y=350, width=70)

                snacks_list = ['Popcorn', 'Nachos', 'Pepsi', 'Burger', 'Hotdog']
                prices = {'Popcorn': 5, 'Nachos': 7, 'Pepsi': 2, 'Burger': 15, 'Hotdog': 10}
                scale_vars = {}

                for idx, snack in enumerate(snacks_list):
                    lbl = tk.Label(frmsnck, text=f"{snack} (Cost: ${prices[snack]})", fg='black', bg='grey',
font=('Times New Roman', 10, 'bold'))
                    lbl.place(x=50, y=100 + idx * 40)

                    scale_vars[snack] = tk.Scale(frmsnck, from_=0, to=5, orient=tk.HORIZONTAL, length=150)
                    scale_vars[snack].place(x=200, y=100 + idx * 40)
                    scale_vars[snack].snack = snack
                    scale_vars[snack].bind("<ButtonRelease-1>", on_scale_change)

            def booking():
                def taken():
                    nonlocal buttons
                    try:
                        myconn=mys.connect(host="localhost",user="root",passwd="adis",database="movie")
                        mycur=myconn.cursor()

                        query = "select seats from wonka"
                        mycur.execute(query)
                        result=mycur.fetchall()
                        booked_seats = []
                        for tup in result:
                            elements = [elem.strip() for elem in tup[0].split(',')]
                            booked_seats.extend([elem for elem in elements if elem])
                        myconn.commit()
                        for row in range(len(buttons)):
                            for col in range(len(buttons[row])):
```

```
                            seat_label = chr(ord('a') + row) + str(col + 1)
                            if seat_label in booked_seats:
                                buttons[row][col].configure(bg="grey", state=tk.DISABLED)

                    except Exception as e:
                        print(e)

                def clear():
                    for button_row in buttons:
                        for button in button_row:
                            button.configure(bg="green")
                    global seatmoney, nseat
                    seatmoney = 0
                    nseat = 0
                    update_display()
                    seatwonka.clear()

                def update_display():
                    b1.config(text="$" + str(seatmoney))

                def button_click(row, col):
                    global seatmoney, nseat
                    button = buttons[row][col]
                    seat_label = chr(ord('a') + row) + str(col + 1)
                    if button["bg"] == "green":
                        button.configure(bg="red")
                        seatmoney += 20
                        nseat += 1
                        seatwonka.append(seat_label)

                    else:
                        button.configure(bg="green")
                        seatmoney -= 20
                        nseat -= 1
                        seatwonka.remove(seat_label)

                    update_display()

                frmbook = tk.Tk()
                frmbook.geometry("400x450")
                frmbook.title("Seat")
                frmbook.configure(bg='black')
                lblscr = tk.Label(frmbook, text="------------------screen--------------------", fg='white',
bg='black',
                                  font=('Times New Roman', 20))
                lblscr.place(x=5, y=100)

                btclr = tk.Button(frmbook, text="Clear", fg='black', bg='Red', font=("Times New Roman Bold",
10), command=clear)
                btclr.place(x=250, y=400, width=50)

                btbook = tk.Button(frmbook, text="Book", fg='black', bg='Red', font=("Times New Roman Bold",
10), command=book)
                btbook.place(x=300, y=400, width=50)

                b1 = tk.Button(frmbook, text="$" + str(seatmoney))
                b1.place(x=120, y=400, width=70)

                lbla = tk.Label(frmbook, text="a", fg='white', bg='black', font=("Times New Roman Bold", 12))
                lbla.place(x=5, y=150)
                lblb = tk.Label(frmbook, text="b", fg='white', bg='black', font=("Times New Roman Bold", 12))
                lblb.place(x=5, y=200)
                lblc = tk.Label(frmbook, text="c", fg='white', bg='black', font=("Times New Roman Bold", 12))
                lblc.place(x=5, y=250)
                lbld = tk.Label(frmbook, text="d", fg='white', bg='black', font=("Times New Roman Bold", 12))
                lbld.place(x=5, y=300)
                lblno = tk.Label(frmbook, text="1   2   3   4   5   6   7   8   9  10  11 12 13 14 15 16 17
18", fg='white', bg='black', font=("Times New Roman Bold", 12))
                lblno.place(x=20, y=325)


                buttons = []


                for row in range(4):
                    button_row = []
                    for col in range(18):
                        seat_label = chr(ord('a') + row) + str(col + 1)
```

```
                        button = tk.Button(frmbook, bg='green', height=1, width=1, command=lambda r=row,
c=col: button_click(r, c))
                        button.place(x=20 + col * 20, y=150 + row * 50)
                        button_row.append(button)
                    buttons.append(button_row)
                taken()
            lblhd = tk.Label(frmtime, text ="Wonka", bg= 'light green',font=('MV
Boli',15,'bold','underline'))
            lblhd.place(x = 170, y = 10)
            btbook= tk.Button(frmtime, text ="Book seat",fg ='red',bg= 'light green',font=("Times New Roman
Bold", 10),command=booking)
            btbook.place(x = 250, y = 155, width = 100)
            btsnack= tk.Button(frmtime, text ="Snacks",fg ='red',bg= 'light green',font=("Times New Roman
Bold", 10),command=snacks)
            btsnack.place(x = 250, y = 185, width = 100)
            lbldescrpt = tk.Label(frmtime, text ="Description", bg= 'light green',font=('Times New Roman',12,
'underline'))
            lbldescrpt.place(x = 2, y = 40)
            lbldes = tk.Label(frmtime, text ="The Story will focus on a young Willy Wonka and his adventures
prior\n to opening the world’s most famous chocolate factory.",
                              bg= 'light green',font=('Times New Roman',10))
            lbldes.place(x = 2, y = 65)
            lblgenre = tk.Label(frmtime, text ="Genre", bg= 'light green',font=('Times New Roman',12,
'underline'))
            lblgenre.place(x = 2, y = 110)
            lblgen = tk.Label(frmtime, text ="Adventure", bg= 'light green',font=('Times New Roman',10))
            lblgen.place(x = 2, y = 135)
            lblrt = tk.Label(frmtime, text ="Running time", bg= 'light green',font=('Times New Roman',12,
'underline'))
            lblrt.place(x = 2, y = 155)
            lblrun = tk.Label(frmtime, text ="120 mins", bg= 'light green',font=('Times New Roman',10))
            lblrun.place(x = 2, y = 180)
            lblstar = tk.Label(frmtime, text ="Starring", bg= 'light green',font=('Times New Roman',12,
'underline'))
            lblstar.place(x = 2, y = 200)
            lblact = tk.Label(frmtime, text ="Rowan Atkinson, Timothee Chalamet,\n Calah Lane, Keegan Michael
Key", bg= 'light green',font=('Times New Roman',10))
            lblact.place(x = 2, y = 225)

            link = tk.Label(frmtime, text="Trailer",
            fg="blue", bg="light green", cursor="hand2",font=['Times',15,'underline'])
            link.place(x=160, y=155)
            link.bind("<Button-1>", lambda e:
webbrowser.open_new("https://youtu.be/otNh9bTjXWg?si=o9SxmqegS5VsVcEg"))

        bg = ImageTk.PhotoImage(Image.open("bg2.jpg"))
        button3= Button(frmeng, image = bg)
        button3.pack()
        lbllan = tk.Label(frmeng, text ="ENGLISH MOVIES", fg= 'white', bg= 'black',font=('MV
Boli',25,'bold','underline'))
        lbllan.place(x = 160, y = 15)
        img = ImageTk.PhotoImage(Image.open("aqua.jpg"))
        button1= Button(frmeng, image = img, command=showtimes1)
        button1.place(x = 50, y = 150)
        img2 = ImageTk.PhotoImage(Image.open("wonka.jpg"))
        button2= Button(frmeng, image = img2, command=showtimes2)
        button2.place(x = 350, y = 150)
        if ver==0:
            btlogin = tk.Button(frmeng, text ="login",fg ='white',bg= 'red',font=("Times New Roman Bold",
10),command=loginscreen)
            btlogin.place(x = 530, y = 5, width = 60)
        if ver==1:
            lblacc = tk.Label(frmeng, text ="Account name:", fg ='red', bg= 'black',font=('Times New
Roman',10, "underline"))
            lblacc.place(x = 470, y = 5)
            lblus = tk.Label(frmeng, text =user1, fg ='red', bg= 'black',font=('Times New Roman',10,
"underline"))
            lblus.place(x = 555, y = 5)
        root.mainloop()

    def Hindi():
        root.destroy()
        frmhin=tk.Tk()
        frmhin.geometry("600x600")
        frmhin.title("Hindi Movies")
        frmhin.configure(bg='light blue')
        bg = ImageTk.PhotoImage(Image.open("bg2.jpg"))
        button3= Button(frmhin, image = bg)
        button3.pack()
```

```
        lbllan = tk.Label(frmhin, text ="HINDI MOVIES", fg= 'white', bg= 'black',font=('MV
Boli',25,'bold','underline'))
        lbllan.place(x = 170, y = 15)

        def loginscreen():
            def funclear():
                txtUser.delete(0,END)
                txtpass.delete(0,END)
                txtUser.focus()

            def funlogin():
                user = txtUser.get()
                passw = txtpass.get()
                try:
                    myconn=mys.connect(host="localhost",user="root",passwd="adis",database="movie")
                    mycur=myconn.cursor()
                    query="select * from customer where c_name='{}' and pass='{}'".format(user,passw)
                    mycur.execute(query)
                    rs=mycur.fetchone()
                    if rs!=None and user==rs[0] and passw==rs[2]:
                        global ver
                        ver=1
                        global user1
                        user1=user
                        passw=txtpass.get()
                        l=user+passw
                        global i_d1
                        for ch in l:
                            i_d1+=str(ord(ch))
                        messagebox.showinfo("Login ", "Login Successful")
                        login.destroy()
                        btlogin.destroy()
                        lblacc = tk.Label(frmhin, text ="Account name:", fg ='red', bg= 'black',font=('Times
New Roman',10, "underline"))
                        lblacc.place(x = 470, y = 5)
                        lblus = tk.Label(frmhin, text =user1, fg ='red', bg= 'black',font=('Times New
Roman',10, "underline"))
                        lblus.place(x = 555, y = 5)
                    else:
                        messagebox.showinfo("Login", "Login Failed")
                except Exception as e:
                    print(e)

            def register():
                i_d=''
                user=txtUser.get()
                passw=txtpass.get()
                l=user+passw
                for ch in l:
                    i_d+=str(ord(ch))
                try:
                    myconn=mys.connect(host="localhost",user="root",passwd="adis",database="movie")
                    mycur=myconn.cursor()
                    query="insert into customer values('{}',{},'{}')".format(user,i_d,passw)
                    mycur.execute(query)
                    myconn.commit()
                    messagebox.showinfo("Registration", "Registration Successful, you may Login now")
                    funclear()
                except Exception as e:
                    messagebox.showinfo("Registration", "Registration Successful, you may Login now")


            login = tk.Tk()
            login.geometry("300x300")
            login.title("Login Page")
            C = Canvas(login, bg ="Red", height = 250, width = 300)
            lbluser = tk.Label(login, text ="Username -",fg="red",font=("Times New Roman Bold", 10) )
            lbluser.place(x = 50, y = 20)
            txtUser = tk.Entry(login, width = 35,fg="blue",font=("Times New Roman Bold", 10))
            txtUser.place(x = 150, y = 20, width = 100)
            lblpass = tk.Label(login, text ="Password -",fg="red",font=("Times New Roman Bold", 10))
            lblpass.place(x = 50, y = 50)
            txtpass = tk.Entry(login,show="*", width = 35,fg="blue",font=("Times New Roman Bold", 10))
            txtpass.place(x = 150, y = 50, width = 100)
            loginbtn = tk.Button(login, text ="Login", fg ='blue', font=("Times New Roman Bold", 10),command
= funlogin)
            loginbtn.place(x = 140, y = 135, width = 55)
            regbtn = tk.Button(login, text ="Register", fg ='blue', font=("Times New Roman Bold", 10),command
= register)
```

```
            regbtn.place(x = 80, y = 135, width = 55)
            clearbtn = tk.Button(login, text ="Clear",fg ='blue', font=("Times New Roman Bold", 10),command =
funclear)
            clearbtn.place(x = 200, y = 135, width = 55)

        if ver==0:
            btlogin = tk.Button(frmhin, text ="login",fg ='white',bg= 'red',font=("Times New Roman Bold",
10),command=loginscreen)
            btlogin.place(x = 530, y = 5, width = 60)
        if ver==1:
            lblacc = tk.Label(frmhin, text ="Account name:", fg ='red', bg= 'black',font=('Times New
Roman',10, "underline"))
            lblacc.place(x = 470, y = 5)
            lblus = tk.Label(frmhin, text =user1, fg ='red', bg= 'black',font=('Times New Roman',10,
"underline"))
            lblus.place(x = 555, y = 5)

        def showtimes1():
            frmtime=tk.Tk()
            frmtime.geometry("400x300")
            frmtime.title("Dunki")
            frmtime.configure(bg='light green')
            def end():
                global nseat,final
                global user1, i_d1
                seat1=snacks1=''
                for ele in seatdunki:
                    seat1+= ele + ', '
                for ele in n:
                    snacks1+= ele + ', '

                try:
                    myconn=mys.connect(host="localhost",user="root",passwd="adis",database="movie")
                    mycur=myconn.cursor()

                    query = "INSERT INTO dunki (c_name, acc_id, snacks, `number of seats`, seats, cost)
VALUES ('{}', '{}', '{}', {}, '{}',{})".format(user1, i_d1, snacks1, nseat, seat1,final)
                    mycur.execute(query)
                    myconn.commit()
                    messagebox.showinfo("Booking Successful", "Booking Successful, Enjoy your Movie")

                except Exception as e:
                    print(e)

            def book():
                global nseat
                if nseat>=1 and ver==1:
                    book = tk.Tk()
                    book.geometry("400x400")
                    book.title("Booking")

                    global user1
                    global pop,nacho,pepsi,burger,hotdog

                    btb=Button(book,text="book",fg='black',bg='red', command=end)
                    btb.place(x=200,y=370)

                    table = ttk.Treeview(book, columns = ('name', 'movie','snack', 'seat'), show =
'headings')
                    table.heading('name', text = 'Name')
                    table.heading('movie', text = 'Movie')
                    table.heading('snack', text = 'Snacks')
                    table.heading('seat', text = 'Seats')
                    table.pack()

                    table.column('name', width=100)
                    table.column('movie', width=100)
                    table.column('snack', width=100)
                    table.column('seat', width=100)

                    if pop==1:
                        n.append('Popcorn')
                    if nacho==1:
                        n.append('Nachos')
                    if pepsi==1:
                        n.append('Pepsi')
                    if burger==1:
                        n.append('Burger')
                    if hotdog==1:
```

```
                    n.append('Hotdog')

                name = user1
                movie = "dunki"
                table.insert(parent='', index=tk.END, values=(name, movie,'',''))

                max_items = max(len(n), len(seatdunki))
                for i in range(max_items):
                    snack_value = n[i] if i < len(n) else ''
                    seat_value = seatdunki[i] if i < len(seatdunki) else ''
                    table.insert(parent='', index=tk.END, values=('','',snack_value, seat_value))

                global seatmoney,snckmoney,final
                total=seatmoney+snckmoney
                t=pop+nacho+pepsi+burger+hotdog
                d1=d2=0
                if nseat>3 and t>2:
                    d1=seatmoney-(0.2*seatmoney)
                    d2=snckmoney-(0.1*snckmoney)
                    final=d1+d2
                #Discount 1 : 10% off total if seats are greater than 3
                elif nseat>3:
                    d1=seatmoney-(0.2*seatmoney)
                    final=d1+snckmoney
                #Discount 2 : 10% off snckmoney if snacks are greater than 2
                elif t>2:
                    d2=snckmoney-(0.1*snckmoney)
                    final=d2+seatmoney
                else:
                    final=total

                lblcost = tk.Label(book, text ="Cost :",fg="black",font=("Arial", 12))
                lblcost.place(x = 100, y = 200)
                lbl1 = tk.Label(book, text ="$"+str(snckmoney),fg="black",font=("Arial", 12))
                lbl1.place(x = 250, y = 200)
                lbl2 = tk.Label(book, text ="$"+str(seatmoney),fg="black",font=("Arial", 12))
                lbl2.place(x = 350, y = 200)
                lbltot = tk.Label(book, text ="Total :",fg="black",font=("Arial", 12))
                lbltot.place(x = 200, y = 230)
                lbl3 = tk.Label(book, text ="$"+str(total),fg="black",font=("Arial", 12))
                lbl3.place(x = 250, y = 230)
                lblfinal = tk.Label(book, text ="Total after discount :",fg="black",font=("Arial", 12))
                lblfinal.place(x = 200, y = 260)
                lbl4 = tk.Label(book, text ="$"+str(final),fg="black",font=("Arial", 12))
                lbl4.place(x = 350, y = 260)
                lbldis = tk.Label(book, text ="Discounts",fg="black", borderwidth=2, relief='solid',
font=("Arial", 10,'underline','bold'))
                lbldis.place(x = 180, y = 310)
                lbldis1 = tk.Label(book, text ="1. For booking more than 3 seats: Reduction of 20% from
the cost of seats\n 2. For selecting more than 2 snacks: Reduction of 10% from the cost of
snacks",fg="black", borderwidth=2, relief='solid', font=("Times New Roman", 8))
                lbldis1.place(x = 20, y = 330)


            elif nseat<1:
                    messagebox.showinfo("Booking failed ", "Please Book Seats")
            elif ver==0:
                    messagebox.showinfo("Booking failed ", "Please Login First")

        def snacks():
            global snckmoney
            snack_quantities = {'Popcorn': 0, 'Nachos': 0, 'Pepsi': 0, 'Burger': 0, 'Hotdog': 0}
            def update_display():
                global snckmoney
                snckmoney = sum(snack_quantities[snack] * prices[snack] for snack in snacks_list)
                b1.config(text="$" + str(snckmoney))

            def clear():
                for snack in snacks_list:
                    scale_vars[snack].set(0)
                    snack_quantities[snack] = 0
                update_display()

            def on_scale_change(event):
                snack = event.widget.snack
                snack_quantities[snack] = scale_vars[snack].get()
                update_display()
                global pop,hotdog,nacho,pepsi,burger
                if snack == 'Popcorn' and snack_quantities[snack] > 0:
```

```
                    pop = 1
                if snack == 'Hotdog' and snack_quantities[snack] > 0:
                    hotdog = 1
                if snack == 'Burger' and snack_quantities[snack] > 0:
                    burger = 1
                if snack == 'Pepsi' and snack_quantities[snack] > 0:
                    pepsi = 1
                if snack == 'Nachos' and snack_quantities[snack] > 0:
                    nacho = 1

            frmsnck = tk.Tk()
            frmsnck.geometry("400x500")
            frmsnck.title("Snacks")
            frmsnck.configure(bg='black')

            btclr = tk.Button(frmsnck, text="Clear", fg='black', bg='Red', font=("Times New Roman Bold",
10), command=clear)
            btclr.place(x=250, y=400, width=50)

            btbook = tk.Button(frmsnck, text="Book", fg='black', bg='Red', font=("Times New Roman Bold",
10), command=book)
            btbook.place(x=300, y=400, width=50)

            lbltot = tk.Label(frmsnck, text="Total : ", fg='black', bg='white', font=('Times New Roman',
12, 'bold', 'underline'))
            lbltot.place(x=180, y=350)

            b1 = tk.Label(frmsnck, text="$" + str(snckmoney))
            b1.place(x=230, y=350, width=70)

            snacks_list = ['Popcorn', 'Nachos', 'Pepsi', 'Burger', 'Hotdog']
            prices = {'Popcorn': 5, 'Nachos': 7, 'Pepsi': 2, 'Burger': 15, 'Hotdog': 10}
            scale_vars = {}

            for idx, snack in enumerate(snacks_list):
                lbl = tk.Label(frmsnck, text=f"{snack} (Cost: ${prices[snack]})", fg='black', bg='grey',
font=('Times New Roman', 10, 'bold'))
                lbl.place(x=50, y=100 + idx * 40)

                scale_vars[snack] = tk.Scale(frmsnck, from_=0, to=5, orient=tk.HORIZONTAL, length=150)
                scale_vars[snack].place(x=200, y=100 + idx * 40)
                scale_vars[snack].snack = snack
                scale_vars[snack].bind("<ButtonRelease-1>", on_scale_change)

        def booking():
            def taken():
                nonlocal buttons
                try:
                    myconn=mys.connect(host="localhost",user="root",passwd="adis",database="movie")
                    mycur=myconn.cursor()

                    query = "select seats from dunki"
                    mycur.execute(query)
                    result=mycur.fetchall()
                    booked_seats = []
                    for tup in result:
                        elements = [elem.strip() for elem in tup[0].split(',')]
                        booked_seats.extend([elem for elem in elements if elem])
                    myconn.commit()
                    for row in range(len(buttons)):
                        for col in range(len(buttons[row])):
                            seat_label = chr(ord('a') + row) + str(col + 1)
                            if seat_label in booked_seats:
                                buttons[row][col].configure(bg="grey", state=tk.DISABLED)

                except Exception as e:
                    print(e)

            def clear():
                for button_row in buttons:
                    for button in button_row:
                        button.configure(bg="green")
                global seatmoney, nseat
                seatmoney = 0
                nseat = 0
                update_display()
                seatdunki.clear()

            def update_display():
```

```
                    b1.config(text="$" + str(seatmoney))

                def button_click(row, col):
                    global seatmoney, nseat
                    button = buttons[row][col]
                    seat_label = chr(ord('a') + row) + str(col + 1)
                    if button["bg"] == "green":
                        button.configure(bg="red")
                        seatmoney += 20
                        nseat += 1
                        seatdunki.append(seat_label)

                    else:
                        button.configure(bg="green")
                        seatmoney -= 20
                        nseat -= 1
                        seatdunki.remove(seat_label)

                    update_display()

                frmbook = tk.Tk()
                frmbook.geometry("400x450")
                frmbook.title("Seat")
                frmbook.configure(bg='black')
                lblscr = tk.Label(frmbook, text="-----------------screen-------------------", fg='white',
bg='black',
                                  font=('Times New Roman', 20))
                lblscr.place(x=5, y=100)

                btclr = tk.Button(frmbook, text="Clear", fg='black', bg='Red', font=("Times New Roman Bold",
10), command=clear)
                btclr.place(x=250, y=400, width=50)

                btbook = tk.Button(frmbook, text="Book", fg='black', bg='Red', font=("Times New Roman Bold",
10), command=book)
                btbook.place(x=300, y=400, width=50)

                b1 = tk.Button(frmbook, text="$" + str(seatmoney))
                b1.place(x=120, y=400, width=70)

                lbla = tk.Label(frmbook, text="a", fg='white', bg='black', font=("Times New Roman Bold", 12))
                lbla.place(x=5, y=150)
                lblb = tk.Label(frmbook, text="b", fg='white', bg='black', font=("Times New Roman Bold", 12))
                lblb.place(x=5, y=200)
                lblc = tk.Label(frmbook, text="c", fg='white', bg='black', font=("Times New Roman Bold", 12))
                lblc.place(x=5, y=250)
                lbld = tk.Label(frmbook, text="d", fg='white', bg='black', font=("Times New Roman Bold", 12))
                lbld.place(x=5, y=300)
                lblno = tk.Label(frmbook, text="1   2   3   4   5   6   7   8   9  10  11 12 13 14 15 16 17
18", fg='white', bg='black', font=("Times New Roman Bold", 12))
                lblno.place(x=20, y=325)


                buttons = []


                for row in range(4):
                    button_row = []
                    for col in range(18):
                        seat_label = chr(ord('a') + row) + str(col + 1)
                        button = tk.Button(frmbook, bg='green', height=1, width=1, command=lambda r=row,
c=col: button_click(r, c))
                        button.place(x=20 + col * 20, y=150 + row * 50)
                        button_row.append(button)
                    buttons.append(button_row)
                taken()

            lblhd = tk.Label(frmtime, text ="Dunki", bg= 'light green',font=('MV
Boli',15,'bold','underline'))
            lblhd.place(x = 170, y = 10)
            btbook= tk.Button(frmtime, text ="Book seat",fg ='red',bg= 'light green',font=("Times New Roman
Bold", 10),command=booking)
            btbook.place(x = 250, y = 155, width = 100)
            btsnack= tk.Button(frmtime, text ="Snacks",fg ='red',bg= 'light green',font=("Times New Roman
Bold", 10),command=snacks)
            btsnack.place(x = 250, y = 185, width = 100)
            lbldescrpt = tk.Label(frmtime, text ="Description", bg= 'light green',font=('Times New Roman',12,
'underline'))
            lbldescrpt.place(x = 2, y = 40)
```

```
            lbldes = tk.Label(frmtime, text ="4 friends from a village in Punjab share a common dream: to go
England.\n Dunki is a hilarious and heartwarming saga of a perilous journey, borders,\n friendships,
nostalgia for home, and love that towers\n above it all.",
                              bg= 'light green',font=('Times New Roman',10))
            lbldes.place(x = 2, y = 65)
            lblgenre = tk.Label(frmtime, text ="Genre", bg= 'light green',font=('Times New Roman',12,
'underline'))
            lblgenre.place(x = 2, y = 120)
            lblgen = tk.Label(frmtime, text ="Adventure", bg= 'light green',font=('Times New Roman',10))
            lblgen.place(x = 2, y = 145)
            lblrt = tk.Label(frmtime, text ="Running time", bg= 'light green',font=('Times New Roman',12,
'underline'))
            lblrt.place(x = 2, y = 165)
            lblrun = tk.Label(frmtime, text ="120 mins", bg= 'light green',font=('Times New Roman',10))
            lblrun.place(x = 2, y = 190)
            lblstar = tk.Label(frmtime, text ="Starring", bg= 'light green',font=('Times New Roman',12,
'underline'))
            lblstar.place(x = 2, y = 210)
            lblact = tk.Label(frmtime, text ="Rowan Atkinson, Timothee Chalamet,\n Calah Lane, Keegan Michael
Key", bg= 'light green',font=('Times New Roman',10))
            lblact.place(x = 2, y = 235)

            link = tk.Label(frmtime, text="Trailer",
            fg="blue", bg="light green",cursor="hand2",font=['Times',15,'underline'])
            link.place(x=160, y=155)
            link.bind("<Button-1>", lambda e:
webbrowser.open_new("https://youtu.be/ACKQDAlAfFE?si=FarPE9zOrQwU6ijq"))

        def showtimes2():
            frmtime=tk.Tk()
            frmtime.geometry("400x300")
            frmtime.title("Animal")
            frmtime.configure(bg='light green')
            def end():
                global nseat,final
                global user1, i_d1
                seat1=snacks1=''
                for ele in seatanimal:
                    seat1+= ele + ', '
                for ele in n:
                    snacks1+= ele + ', '

                try:
                    myconn=mys.connect(host="localhost",user="root",passwd="adis",database="movie")
                    mycur=myconn.cursor()

                    query = "INSERT INTO animal (c_name, acc_id, snacks, `number of seats`, seats, cost)
VALUES ('{}', '{}', '{}', {}, '{}',{})".format(user1, i_d1, snacks1, nseat, seat1,final)
                    mycur.execute(query)
                    myconn.commit()
                    messagebox.showinfo("Booking Successful", "Booking Successful, Enjoy your Movie")

                except Exception as e:
                    print(e)

            def book():
                global nseat
                if nseat>=1 and ver==1:
                    book = tk.Tk()
                    book.geometry("400x400")
                    book.title("Booking")

                    global user1
                    global pop,nacho,pepsi,burger,hotdog

                    btb=Button(book,text="book",fg='black',bg='red', command=end)
                    btb.place(x=200,y=370)

                    table = ttk.Treeview(book, columns = ('name', 'movie','snack', 'seat'), show =
'headings')
                    table.heading('name', text = 'Name')
                    table.heading('movie', text = 'Movie')
                    table.heading('snack', text = 'Snacks')
                    table.heading('seat', text = 'Seats')
                    table.pack()

                    table.column('name', width=100)
                    table.column('movie', width=100)
                    table.column('snack', width=100)
```

```
                    table.column('seat', width=100)

                    if pop==1:
                        n.append('Popcorn')
                    if nacho==1:
                        n.append('Nachos')
                    if pepsi==1:
                        n.append('Pepsi')
                    if burger==1:
                        n.append('Burger')
                    if hotdog==1:
                        n.append('Hotdog')

                    name = user1
                    movie = "animal"
                    table.insert(parent='', index=tk.END, values=(name, movie,'',''))

                    max_items = max(len(n), len(seatanimal))
                    for i in range(max_items):
                        snack_value = n[i] if i < len(n) else ''
                        seat_value = seatanimal[i] if i < len(seatanimal) else ''
                        table.insert(parent='', index=tk.END, values=('','',snack_value, seat_value))

                    global seatmoney,snckmoney,final
                    total=seatmoney+snckmoney
                    t=pop+nacho+pepsi+burger+hotdog
                    d1=d2=0
                    if nseat>3 and t>2:
                        d1=seatmoney-(0.2*seatmoney)
                        d2=snckmoney-(0.1*snckmoney)
                        final=d1+d2
                    #Discount 1 : 10% off total if seats are greater than 3
                    elif nseat>3:
                        d1=seatmoney-(0.2*seatmoney)
                        final=d1+snckmoney
                    #Discount 2 : 10% off snckmoney if snacks are greater than 2
                    elif t>2:
                        d2=snckmoney-(0.1*snckmoney)
                        final=d2+seatmoney
                    else:
                        final=total

                    lblcost = tk.Label(book, text ="Cost :",fg="black",font=("Arial", 12))
                    lblcost.place(x = 100, y = 200)
                    lbl1 = tk.Label(book, text ="$"+str(snckmoney),fg="black",font=("Arial", 12))
                    lbl1.place(x = 250, y = 200)
                    lbl2 = tk.Label(book, text ="$"+str(seatmoney),fg="black",font=("Arial", 12))
                    lbl2.place(x = 350, y = 200)
                    lbltot = tk.Label(book, text ="Total :",fg="black",font=("Arial", 12))
                    lbltot.place(x = 200, y = 230)
                    lbl3 = tk.Label(book, text ="$"+str(total),fg="black",font=("Arial", 12))
                    lbl3.place(x = 250, y = 230)
                    lblfinal = tk.Label(book, text ="Total after discount :",fg="black",font=("Arial", 12))
                    lblfinal.place(x = 200, y = 260)
                    lbl4 = tk.Label(book, text ="$"+str(final),fg="black",font=("Arial", 12))
                    lbl4.place(x = 350, y = 260)
                    lbldis = tk.Label(book, text ="Discounts",fg="black", borderwidth=2, relief='solid',
font=("Arial", 10,'underline','bold'))
                    lbldis.place(x = 180, y = 310)
                    lbldis1 = tk.Label(book, text ="1. For booking more than 3 seats: Reduction of 20% from
the cost of seats\n 2. For selecting more than 2 snacks: Reduction of 10% from the cost of
snacks",fg="black", borderwidth=2, relief='solid', font=("Times New Roman", 8))
                    lbldis1.place(x = 20, y = 330)


                elif nseat<1:
                        messagebox.showinfo("Booking failed ", "Please Book Seats")
                elif ver==0:
                        messagebox.showinfo("Booking failed ", "Please Login First")

            def snacks():
                global snckmoney
                snack_quantities = {'Popcorn': 0, 'Nachos': 0, 'Pepsi': 0, 'Burger': 0, 'Hotdog': 0}
                def update_display():
                    global snckmoney
                    snckmoney = sum(snack_quantities[snack] * prices[snack] for snack in snacks_list)
                    b1.config(text="$" + str(snckmoney))

                def clear():
```

```
                    for snack in snacks_list:
                        scale_vars[snack].set(0)
                        snack_quantities[snack] = 0
                    update_display()

                def on_scale_change(event):
                    snack = event.widget.snack
                    snack_quantities[snack] = scale_vars[snack].get()
                    update_display()
                    global pop,hotdog,nacho,pepsi,burger
                    if snack == 'Popcorn' and snack_quantities[snack] > 0:
                        pop = 1
                    if snack == 'Hotdog' and snack_quantities[snack] > 0:
                        hotdog = 1
                    if snack == 'Burger' and snack_quantities[snack] > 0:
                        burger = 1
                    if snack == 'Pepsi' and snack_quantities[snack] > 0:
                        pepsi = 1
                    if snack == 'Nachos' and snack_quantities[snack] > 0:
                        nacho = 1

                frmsnck = tk.Tk()
                frmsnck.geometry("400x500")
                frmsnck.title("Snacks")
                frmsnck.configure(bg='black')

                btclr = tk.Button(frmsnck, text="Clear", fg='black', bg='Red', font=("Times New Roman Bold",
10), command=clear)
                btclr.place(x=250, y=400, width=50)

                btbook = tk.Button(frmsnck, text="Book", fg='black', bg='Red', font=("Times New Roman Bold",
10), command=book)
                btbook.place(x=300, y=400, width=50)

                lbltot = tk.Label(frmsnck, text="Total : ", fg='black', bg='white', font=('Times New Roman',
12, 'bold', 'underline'))
                lbltot.place(x=180, y=350)

                b1 = tk.Label(frmsnck, text="$" + str(snckmoney))
                b1.place(x=230, y=350, width=70)

                snacks_list = ['Popcorn', 'Nachos', 'Pepsi', 'Burger', 'Hotdog']
                prices = {'Popcorn': 5, 'Nachos': 7, 'Pepsi': 2, 'Burger': 15, 'Hotdog': 10}
                scale_vars = {}

                for idx, snack in enumerate(snacks_list):
                    lbl = tk.Label(frmsnck, text=f"{snack} (Cost: ${prices[snack]})", fg='black', bg='grey',
font=('Times New Roman', 10, 'bold'))
                    lbl.place(x=50, y=100 + idx * 40)

                    scale_vars[snack] = tk.Scale(frmsnck, from_=0, to=5, orient=tk.HORIZONTAL, length=150)
                    scale_vars[snack].place(x=200, y=100 + idx * 40)
                    scale_vars[snack].snack = snack
                    scale_vars[snack].bind("<ButtonRelease-1>", on_scale_change)

            def booking():
                def taken():
                    nonlocal buttons
                    try:
                        myconn=mys.connect(host="localhost",user="root",passwd="adis",database="movie")
                        mycur=myconn.cursor()

                        query = "select seats from animal"
                        mycur.execute(query)
                        result=mycur.fetchall()
                        booked_seats = []
                        for tup in result:
                            elements = [elem.strip() for elem in tup[0].split(',')]
                            booked_seats.extend([elem for elem in elements if elem])
                        myconn.commit()
                        for row in range(len(buttons)):
                            for col in range(len(buttons[row])):
                                seat_label = chr(ord('a') + row) + str(col + 1)
                                if seat_label in booked_seats:
                                    buttons[row][col].configure(bg="grey", state=tk.DISABLED)

                    except Exception as e:
                        print(e)
```

```
                def clear():
                    for button_row in buttons:
                        for button in button_row:
                            button.configure(bg="green")

                    global seatmoney, nseat
                    seatmoney = 0
                    nseat = 0
                    update_display()
                    seatanimal.clear()

                def update_display():
                    b1.config(text="$" + str(seatmoney))

                def button_click(row, col):
                    global seatmoney, nseat
                    button = buttons[row][col]
                    seat_label = chr(ord('a') + row) + str(col + 1)
                    if button["bg"] == "green":
                        button.configure(bg="red")
                        seatmoney += 20
                        nseat += 1
                        seatanimal.append(seat_label)

                    else:
                        button.configure(bg="green")
                        seatmoney -= 20
                        nseat -= 1
                        seatanimal.remove(seat_label)

                    update_display()

                frmbook = tk.Tk()
                frmbook.geometry("400x450")
                frmbook.title("Seat")
                frmbook.configure(bg='black')
                lblscr = tk.Label(frmbook, text="-----------------screen--------------------", fg='white',
bg='black',
                                  font=('Times New Roman', 20))
                lblscr.place(x=5, y=100)

                btclr = tk.Button(frmbook, text="Clear", fg='black', bg='Red', font=("Times New Roman Bold",
10), command=clear)
                btclr.place(x=250, y=400, width=50)

                btbook = tk.Button(frmbook, text="Book", fg='black', bg='Red', font=("Times New Roman Bold",
10), command=book)
                btbook.place(x=300, y=400, width=50)

                b1 = tk.Button(frmbook, text="$" + str(seatmoney))
                b1.place(x=120, y=400, width=70)

                lbla = tk.Label(frmbook, text="a", fg='white', bg='black', font=("Times New Roman Bold", 12))
                lbla.place(x=5, y=150)
                lblb = tk.Label(frmbook, text="b", fg='white', bg='black', font=("Times New Roman Bold", 12))
                lblb.place(x=5, y=200)
                lblc = tk.Label(frmbook, text="c", fg='white', bg='black', font=("Times New Roman Bold", 12))
                lblc.place(x=5, y=250)
                lbld = tk.Label(frmbook, text="d", fg='white', bg='black', font=("Times New Roman Bold", 12))
                lbld.place(x=5, y=300)
                lblno = tk.Label(frmbook, text="1   2   3   4   5   6   7   8   9  10  11 12 13 14 15 16 17
18", fg='white', bg='black', font=("Times New Roman Bold", 12))
                lblno.place(x=20, y=325)


                buttons = []


                for row in range(4):
                    button_row = []
                    for col in range(18):
                        seat_label = chr(ord('a') + row) + str(col + 1)
                        button = tk.Button(frmbook, bg='green', height=1, width=1, command=lambda r=row,
c=col: button_click(r, c))
                        button.place(x=20 + col * 20, y=150 + row * 50)
                        button_row.append(button)
                    buttons.append(button_row)
                taken()
```

```
            lblhd = tk.Label(frmtime, text ="Animal", bg= 'light green',font=('MV
Boli',15,'bold','underline'))
            lblhd.place(x = 170, y = 10)
            btbook= tk.Button(frmtime, text ="Book seat",fg ='red',bg= 'light green',font=("Times New Roman
Bold", 10),command=booking)
            btbook.place(x = 250, y = 155, width = 100)
            btsnack= tk.Button(frmtime, text ="Snacks",fg ='red',bg= 'light green',font=("Times New Roman
Bold", 10),command=snacks)
            btsnack.place(x = 250, y = 185, width = 100)
            lbldescrpt = tk.Label(frmtime, text ="Description", bg= 'light green',font=('Times New Roman',12,
'underline'))
            lbldescrpt.place(x = 2, y = 40)
            lbldes = tk.Label(frmtime, text ="Animal revolves around a troubled father-son relationship, set
in the\n backdrop of extreme bloodshed of the underworld which leads to the\n protagonist to turn into a
psychopath.",
                              bg= 'light green',font=('Times New Roman',10))
            lbldes.place(x = 2, y = 65)
            lblgenre = tk.Label(frmtime, text ="Genre", bg= 'light green',font=('Times New Roman',12,
'underline'))
            lblgenre.place(x = 2, y = 110)
            lblgen = tk.Label(frmtime, text ="Action", bg= 'light green',font=('Times New Roman',10))
            lblgen.place(x = 2, y = 135)
            lblrt = tk.Label(frmtime, text ="Running time", bg= 'light green',font=('Times New Roman',12,
'underline'))
            lblrt.place(x = 2, y = 155)
            lblrun = tk.Label(frmtime, text ="210 mins", bg= 'light green',font=('Times New Roman',10))
            lblrun.place(x = 2, y = 180)
            lblstar = tk.Label(frmtime, text ="Starring", bg= 'light green',font=('Times New Roman',12,
'underline'))
            lblstar.place(x = 2, y = 200)
            lblact = tk.Label(frmtime, text ="Ranbir Kapoor, Tripti Dimri,\n Anil Kapoor, Rashmika Mandana",
bg= 'light green',font=('Times New Roman',10))
            lblact.place(x = 2, y = 225)

            link = tk.Label(frmtime, text="Trailer",
            fg="blue", bg="light green",cursor="hand2",font=['Times',15,'underline'])
            link.place(x=160, y=155)
            link.bind("<Button-1>", lambda e:
webbrowser.open_new("https://youtu.be/8FkLRUJj-o0?si=csJ_3nPv6RIyhixT"))

        img1 = ImageTk.PhotoImage(Image.open("dunki.jpg"))
        button1= Button(frmhin, image = img1, command = showtimes1)
        button1.place(x = 50, y = 150)
        img2 = ImageTk.PhotoImage(Image.open("animal.jpg"))
        button2= Button(frmhin, image = img2, command = showtimes2)
        button2.place(x = 350, y = 150)
        root.mainloop()

    def Malayalam():
        root.destroy()
        frmmal=tk.Tk()
        frmmal.geometry("600x600")
        frmmal.title("Malayalam Movies")
        frmmal.configure(bg='light blue')
        bg = ImageTk.PhotoImage(Image.open("bg2.jpg"))
        button3= Button(frmmal, image = bg)
        button3.pack()
        lbllan = tk.Label(frmmal, text ="MALAYALAM MOVIES", fg= 'white', bg= 'black',font=('MV
Boli',25,'bold','underline'))
        lbllan.place(x = 120, y = 15)

        def loginscreen():
            def funclear():
                txtUser.delete(0,END)
                txtpass.delete(0,END)
                txtUser.focus()

            def funlogin():
                user = txtUser.get()
                passw = txtpass.get()
                try:
                    myconn=mys.connect(host="localhost",user="root",passwd="adis",database="movie")
                    mycur=myconn.cursor()
                    query="select * from customer where c_name='{}' and pass='{}'".format(user,passw)
                    mycur.execute(query)
                    rs=mycur.fetchone()
                    if rs!=None and user==rs[0] and passw==rs[2]:
                        global ver
                        ver=1
```

```
                        global user1
                        user1=user
                        passw=txtpass.get()
                        l=user+passw
                        global i_d1
                        for ch in l:
                            i_d1+=str(ord(ch))
                        messagebox.showinfo("Login ", "Login Successful")
                        login.destroy()
                        btlogin.destroy()
                        lblacc = tk.Label(frmmal, text ="Account name:", fg ='red', bg= 'black',font=('Times
New Roman',10, "underline"))
                        lblacc.place(x = 470, y = 5)
                        lblus = tk.Label(frmmal, text =user1, fg ='red', bg= 'black',font=('Times New
Roman',10, "underline"))
                        lblus.place(x = 555, y = 5)
                    else:
                        messagebox.showinfo("Login", "Login Failed")
                except Exception as e:
                    print(e)

            def register():
                i_d=''
                user=txtUser.get()
                passw=txtpass.get()
                l=user+passw
                for ch in l:
                    i_d+=str(ord(ch))
                try:
                    myconn=mys.connect(host="localhost",user="root",passwd="adis",database="movie")
                    mycur=myconn.cursor()
                    query="insert into customer values('{}',{},'{}')".format(user,i_d,passw)
                    mycur.execute(query)
                    myconn.commit()
                    messagebox.showinfo("Registration", "Registration Successful, you may Login now")
                    funclear()
                except Exception as e:
                    messagebox.showinfo("Login", "Login Failed")


            login = tk.Tk()
            login.geometry("300x300")
            login.title("Login Page")
            C = Canvas(login, bg ="Red", height = 250, width = 300)
            lbluser = tk.Label(login, text ="Username -",fg="red",font=("Times New Roman Bold", 10) )
            lbluser.place(x = 50, y = 20)
            txtUser = tk.Entry(login, width = 35,fg="blue",font=("Times New Roman Bold", 10))
            txtUser.place(x = 150, y = 20, width = 100)
            lblpass = tk.Label(login, text ="Password -",fg="red",font=("Times New Roman Bold", 10))
            lblpass.place(x = 50, y = 50)
            txtpass = tk.Entry(login,show="*", width = 35,fg="blue",font=("Times New Roman Bold", 10))
            txtpass.place(x = 150, y = 50, width = 100)
            loginbtn = tk.Button(login, text ="Login", fg ='blue', font=("Times New Roman Bold", 10),command
= funlogin)
            loginbtn.place(x = 140, y = 135, width = 55)
            regbtn = tk.Button(login, text ="Register", fg ='blue', font=("Times New Roman Bold", 10),command
= register)
            regbtn.place(x = 80, y = 135, width = 55)
            clearbtn = tk.Button(login, text ="Clear",fg ='blue', font=("Times New Roman Bold", 10),command =
funclear)
            clearbtn.place(x = 200, y = 135, width = 55)

        if ver==0:
            btlogin = tk.Button(frmmal, text ="login",fg ='white',bg= 'red',font=("Times New Roman Bold",
10),command=loginscreen)
            btlogin.place(x = 530, y = 5, width = 60)
        if ver==1:
            lblacc = tk.Label(frmmal, text ="Account name:", fg ='red', bg= 'black',font=('Times New
Roman',10, "underline"))
            lblacc.place(x = 470, y = 5)
            lblus = tk.Label(frmmal, text =user1, fg ='red', bg= 'black',font=('Times New Roman',10,
"underline"))
            lblus.place(x = 555, y = 5)

        def showtimes1():
            frmtime=tk.Tk()
            frmtime.geometry("400x300")
            frmtime.title("Neru")
            frmtime.configure(bg='light green')
```

```
            def end():
                global nseat,final
                global user1, i_d1
                seat1=snacks1=''
                for ele in seatneru:
                    seat1+= ele + ', '
                for ele in n:
                    snacks1+= ele + ', '

                try:
                    myconn=mys.connect(host="localhost",user="root",passwd="adis",database="movie")
                    mycur=myconn.cursor()

                    query = "INSERT INTO neru (c_name, acc_id, snacks, `number of seats`, seats, cost) VALUES
('{}', '{}', '{}', {}, '{}',{})".format(user1, i_d1, snacks1, nseat, seat1,final)
                    mycur.execute(query)
                    myconn.commit()
                    messagebox.showinfo("Booking Successful", "Booking Successful, Enjoy your Movie")

                except Exception as e:
                    print(e)

            def book():
                global nseat
                if nseat>=1 and ver==1:
                    book = tk.Tk()
                    book.geometry("400x400")
                    book.title("Booking")

                    global user1
                    global pop,nacho,pepsi,burger,hotdog

                    btb=Button(book,text="book",fg='black',bg='red', command=end)
                    btb.place(x=200,y=370)

                    table = ttk.Treeview(book, columns = ('name', 'movie','snack', 'seat'), show =
'headings')
                    table.heading('name', text = 'Name')
                    table.heading('movie', text = 'Movie')
                    table.heading('snack', text = 'Snacks')
                    table.heading('seat', text = 'Seats')
                    table.pack()

                    table.column('name', width=100)
                    table.column('movie', width=100)
                    table.column('snack', width=100)
                    table.column('seat', width=100)

                    if pop==1:
                        n.append('Popcorn')
                    if nacho==1:
                        n.append('Nachos')
                    if pepsi==1:
                        n.append('Pepsi')
                    if burger==1:
                        n.append('Burger')
                    if hotdog==1:
                        n.append('Hotdog')

                    name = user1
                    movie = "neru"
                    table.insert(parent='', index=tk.END, values=(name, movie,'',''))

                    max_items = max(len(n), len(seatneru))
                    for i in range(max_items):
                        snack_value = n[i] if i < len(n) else ''
                        seat_value = seatneru[i] if i < len(seatneru) else ''
                        table.insert(parent='', index=tk.END, values=('','',snack_value, seat_value))

                    global seatmoney,snckmoney,final
                    total=seatmoney+snckmoney
                    t=pop+nacho+pepsi+burger+hotdog
                    d1=d2=0
                    if nseat>3 and t>2:
                        d1=seatmoney-(0.2*seatmoney)
                        d2=snckmoney-(0.1*snckmoney)
                        final=d1+d2
                    #Discount 1 : 10% off total if seats are greater than 3
                    elif nseat>3:
```

```
                    d1=seatmoney-(0.2*seatmoney)
                    final=d1+snckmoney
                #Discount 2 : 10% off snckmoney if snacks are greater than 2
                elif t>2:
                    d2=snckmoney-(0.1*snckmoney)
                    final=d2+seatmoney
                else:
                    final=total

                lblcost = tk.Label(book, text ="Cost :",fg="black",font=("Arial", 12))
                lblcost.place(x = 100, y = 200)
                lbl1 = tk.Label(book, text ="$"+str(snckmoney),fg="black",font=("Arial", 12))
                lbl1.place(x = 250, y = 200)
                lbl2 = tk.Label(book, text ="$"+str(seatmoney),fg="black",font=("Arial", 12))
                lbl2.place(x = 350, y = 200)
                lbltot = tk.Label(book, text ="Total :",fg="black",font=("Arial", 12))
                lbltot.place(x = 200, y = 230)
                lbl3 = tk.Label(book, text ="$"+str(total),fg="black",font=("Arial", 12))
                lbl3.place(x = 250, y = 230)
                lblfinal = tk.Label(book, text ="Total after discount :",fg="black",font=("Arial", 12))
                lblfinal.place(x = 200, y = 260)
                lbl4 = tk.Label(book, text ="$"+str(final),fg="black",font=("Arial", 12))
                lbl4.place(x = 350, y = 260)
                lbldis = tk.Label(book, text ="Discounts",fg="black", borderwidth=2, relief='solid',
font=("Arial", 10,'underline','bold'))
                lbldis.place(x = 180, y = 310)
                lbldis1 = tk.Label(book, text ="1. For booking more than 3 seats: Reduction of 20% from
the cost of seats\n 2. For selecting more than 2 snacks: Reduction of 10% from the cost of
snacks",fg="black", borderwidth=2, relief='solid', font=("Times New Roman", 8))
                lbldis1.place(x = 20, y = 330)


            elif nseat<1:
                    messagebox.showinfo("Booking failed ", "Please Book Seats")
            elif ver==0:
                    messagebox.showinfo("Booking failed ", "Please Login First")

        def snacks():
            global snckmoney
            snack_quantities = {'Popcorn': 0, 'Nachos': 0, 'Pepsi': 0, 'Burger': 0, 'Hotdog': 0}
            def update_display():
                global snckmoney
                snckmoney = sum(snack_quantities[snack] * prices[snack] for snack in snacks_list)
                b1.config(text="$" + str(snckmoney))

            def clear():
                for snack in snacks_list:
                    scale_vars[snack].set(0)
                    snack_quantities[snack] = 0
                update_display()

            def on_scale_change(event):
                snack = event.widget.snack
                snack_quantities[snack] = scale_vars[snack].get()
                update_display()
                global pop,hotdog,nacho,pepsi,burger
                if snack == 'Popcorn' and snack_quantities[snack] > 0:
                    pop = 1
                if snack == 'Hotdog' and snack_quantities[snack] > 0:
                    hotdog = 1
                if snack == 'Burger' and snack_quantities[snack] > 0:
                    burger = 1
                if snack == 'Pepsi' and snack_quantities[snack] > 0:
                    pepsi = 1
                if snack == 'Nachos' and snack_quantities[snack] > 0:
                    nacho = 1

            frmsnck = tk.Tk()
            frmsnck.geometry("400x500")
            frmsnck.title("Snacks")
            frmsnck.configure(bg='black')

            btclr = tk.Button(frmsnck, text="Clear", fg='black', bg='Red', font=("Times New Roman Bold",
10), command=clear)
            btclr.place(x=250, y=400, width=50)

            btbook = tk.Button(frmsnck, text="Book", fg='black', bg='Red', font=("Times New Roman Bold",
10), command=book)
            btbook.place(x=300, y=400, width=50)
```

```
                lbltot = tk.Label(frmsnck, text="Total : ", fg='black', bg='white', font=('Times New Roman',
12, 'bold', 'underline'))
                lbltot.place(x=180, y=350)

                b1 = tk.Label(frmsnck, text="$" + str(snckmoney))
                b1.place(x=230, y=350, width=70)

                snacks_list = ['Popcorn', 'Nachos', 'Pepsi', 'Burger', 'Hotdog']
                prices = {'Popcorn': 5, 'Nachos': 7, 'Pepsi': 2, 'Burger': 15, 'Hotdog': 10}
                scale_vars = {}

                for idx, snack in enumerate(snacks_list):
                    lbl = tk.Label(frmsnck, text=f"{snack} (Cost: ${prices[snack]})", fg='black', bg='grey',
font=('Times New Roman', 10, 'bold'))
                    lbl.place(x=50, y=100 + idx * 40)

                    scale_vars[snack] = tk.Scale(frmsnck, from_=0, to=5, orient=tk.HORIZONTAL, length=150)
                    scale_vars[snack].place(x=200, y=100 + idx * 40)
                    scale_vars[snack].snack = snack
                    scale_vars[snack].bind("<ButtonRelease-1>", on_scale_change)

            def booking():
                def taken():
                    nonlocal buttons
                    try:
                        myconn=mys.connect(host="localhost",user="root",passwd="adis",database="movie")
                        mycur=myconn.cursor()

                        query = "select seats from neru"
                        mycur.execute(query)
                        result=mycur.fetchall()
                        booked_seats = []
                        for tup in result:
                            elements = [elem.strip() for elem in tup[0].split(',')]
                            booked_seats.extend([elem for elem in elements if elem])
                        myconn.commit()
                        for row in range(len(buttons)):
                            for col in range(len(buttons[row])):
                                seat_label = chr(ord('a') + row) + str(col + 1)
                                if seat_label in booked_seats:
                                    buttons[row][col].configure(bg="grey", state=tk.DISABLED)

                    except Exception as e:
                        print(e)

                def clear():
                    for button_row in buttons:
                        for button in button_row:
                            button.configure(bg="green")
                    global seatmoney, nseat
                    seatmoney = 0
                    nseat = 0
                    update_display()
                    seatneru.clear()

                def update_display():
                    b1.config(text="$" + str(seatmoney))

                def button_click(row, col):
                    global seatmoney, nseat
                    button = buttons[row][col]
                    seat_label = chr(ord('a') + row) + str(col + 1)
                    if button["bg"] == "green":
                        button.configure(bg="red")
                        seatmoney += 20
                        nseat += 1
                        seatneru.append(seat_label)

                    else:
                        button.configure(bg="green")
                        seatmoney -= 20
                        nseat -= 1
                        seatneru.remove(seat_label)

                    update_display()

                frmbook = tk.Tk()
                frmbook.geometry("400x450")
```

```
                frmbook.title("Seat")
                frmbook.configure(bg='black')
                lblscr = tk.Label(frmbook, text="-----------------screen-------------------", fg='white',
bg='black',
                                  font=('Times New Roman', 20))
                lblscr.place(x=5, y=100)

                btclr = tk.Button(frmbook, text="Clear", fg='black', bg='Red', font=("Times New Roman Bold",
10), command=clear)
                btclr.place(x=250, y=400, width=50)

                btbook = tk.Button(frmbook, text="Book", fg='black', bg='Red', font=("Times New Roman Bold",
10), command=book)
                btbook.place(x=300, y=400, width=50)

                b1 = tk.Button(frmbook, text="$" + str(seatmoney))
                b1.place(x=120, y=400, width=70)

                lbla = tk.Label(frmbook, text="a", fg='white', bg='black', font=("Times New Roman Bold", 12))
                lbla.place(x=5, y=150)
                lblb = tk.Label(frmbook, text="b", fg='white', bg='black', font=("Times New Roman Bold", 12))
                lblb.place(x=5, y=200)
                lblc = tk.Label(frmbook, text="c", fg='white', bg='black', font=("Times New Roman Bold", 12))
                lblc.place(x=5, y=250)
                lbld = tk.Label(frmbook, text="d", fg='white', bg='black', font=("Times New Roman Bold", 12))
                lbld.place(x=5, y=300)
                lblno = tk.Label(frmbook, text="1   2   3   4   5   6   7   8   9  10  11 12 13 14 15 16 17
18", fg='white', bg='black', font=("Times New Roman Bold", 12))
                lblno.place(x=20, y=325)


                buttons = []


                for row in range(4):
                    button_row = []
                    for col in range(18):
                        seat_label = chr(ord('a') + row) + str(col + 1)
                        button = tk.Button(frmbook, bg='green', height=1, width=1, command=lambda r=row,
c=col: button_click(r, c))
                        button.place(x=20 + col * 20, y=150 + row * 50)
                        button_row.append(button)
                    buttons.append(button_row)
                taken()

            btbook= tk.Button(frmtime, text ="Book seat",fg ='red',bg= 'light green',font=("Times New Roman
Bold", 10),command=booking)
            btbook.place(x = 250, y = 155, width = 100)
            btsnack= tk.Button(frmtime, text ="Snacks",fg ='red',bg= 'light green',font=("Times New Roman
Bold", 10),command=snacks)
            btsnack.place(x = 250, y = 185, width = 100)
            lblhd = tk.Label(frmtime, text ="Neru", bg= 'light green',font=('MV Boli',15,'bold','underline'))
            lblhd.place(x = 170, y = 10)
            lbldescrpt = tk.Label(frmtime, text ="Description", bg= 'light green',font=('Times New Roman',12,
'underline'))
            lbldescrpt.place(x = 2, y = 40)
            lbldes = tk.Label(frmtime, text ="Set against the backdrop of the Indian legal system. NERU is a
gripping\n legal and emotional drama that unfolds the tale of Sara, a blind sculpture\n artist who seeks
justice after a traumatic incident.",
                              bg= 'light green',font=('Times New Roman',10))
            lbldes.place(x = 2, y = 65)
            lblgenre = tk.Label(frmtime, text ="Genre", bg= 'light green',font=('Times New Roman',12,
'underline'))
            lblgenre.place(x = 2, y = 110)
            lblgen = tk.Label(frmtime, text ="Drama", bg= 'light green',font=('Times New Roman',10))
            lblgen.place(x = 2, y = 135)
            lblrt = tk.Label(frmtime, text ="Running time", bg= 'light green',font=('Times New Roman',12,
'underline'))
            lblrt.place(x = 2, y = 155)
            lblrun = tk.Label(frmtime, text ="165 mins", bg= 'light green',font=('Times New Roman',10))
            lblrun.place(x = 2, y = 180)
            lblstar = tk.Label(frmtime, text ="Starring", bg= 'light green',font=('Times New Roman',12,
'underline'))
            lblstar.place(x = 2, y = 200)
            lblact = tk.Label(frmtime, text ="Mohanlal, Siddique,\n Anaswara, Jagadish", bg= 'light
green',font=('Times New Roman',10))
            lblact.place(x = 2, y = 225)

            link = tk.Label(frmtime, text="Trailer",
```

```
            fg="blue", bg="light green", cursor="hand2",font=['Times',15,'underline'])
            link.place(x=160, y=155)
            link.bind("<Button-1>", lambda e:
webbrowser.open_new("https://youtu.be/abuLOH7xs8I?si=YgLa7WDfZXsf95lz"))


        def showtimes2():
            frmtime=tk.Tk()
            frmtime.geometry("400x300")
            frmtime.title("Raastha")
            frmtime.configure(bg='light green')
            def end():
                global nseat,final
                global user1, i_d1
                seat1=snacks1=''
                for ele in seatraastha:
                    seat1+= ele + ', '
                for ele in n:
                    snacks1+= ele + ', '

                try:
                    myconn=mys.connect(host="localhost",user="root",passwd="adis",database="movie")
                    mycur=myconn.cursor()

                    query = "INSERT INTO raastha (c_name, acc_id, snacks, `number of seats`, seats, cost)
VALUES ('{}', '{}', '{}', {}, '{}',{})".format(user1, i_d1, snacks1, nseat, seat1,final)
                    mycur.execute(query)
                    myconn.commit()
                    messagebox.showinfo("Booking Successful", "Booking Successful, Enjoy your Movie")

                except Exception as e:
                    print(e)

            def book():
                global nseat
                if nseat>=1 and ver==1:
                    book = tk.Tk()
                    book.geometry("400x400")
                    book.title("Booking")

                    global user1
                    global pop,nacho,pepsi,burger,hotdog

                    btb=Button(book,text="book",fg='black',bg='red', command=end)
                    btb.place(x=200,y=370)

                    table = ttk.Treeview(book, columns = ('name', 'movie','snack', 'seat'), show =
'headings')
                    table.heading('name', text = 'Name')
                    table.heading('movie', text = 'Movie')
                    table.heading('snack', text = 'Snacks')
                    table.heading('seat', text = 'Seats')
                    table.pack()

                    table.column('name', width=100)
                    table.column('movie', width=100)
                    table.column('snack', width=100)
                    table.column('seat', width=100)

                    if pop==1:
                        n.append('Popcorn')
                    if nacho==1:
                        n.append('Nachos')
                    if pepsi==1:
                        n.append('Pepsi')
                    if burger==1:
                        n.append('Burger')
                    if hotdog==1:
                        n.append('Hotdog')

                    name = user1
                    movie = "raastha"
                    table.insert(parent='', index=tk.END, values=(name, movie,'',''))

                    max_items = max(len(n), len(seatraastha))
                    for i in range(max_items):
                        snack_value = n[i] if i < len(n) else ''
                        seat_value = seatraastha[i] if i < len(seatraastha) else ''
                        table.insert(parent='', index=tk.END, values=('','',snack_value, seat_value))
```

```
                    global seatmoney,snckmoney,final
                    total=seatmoney+snckmoney
                    t=pop+nacho+pepsi+burger+hotdog
                    d1=d2=0
                    if nseat>3 and t>2:
                        d1=seatmoney-(0.2*seatmoney)
                        d2=snckmoney-(0.1*snckmoney)
                        final=d1+d2
                    #Discount 1 : 10% off total if seats are greater than 3
                    elif nseat>3:
                        d1=seatmoney-(0.2*seatmoney)
                        final=d1+snckmoney
                    #Discount 2 : 10% off snckmoney if snacks are greater than 2
                    elif t>2:
                        d2=snckmoney-(0.1*snckmoney)
                        final=d2+seatmoney
                    else:
                        final=total

                    lblcost = tk.Label(book, text ="Cost :",fg="black",font=("Arial", 12))
                    lblcost.place(x = 100, y = 200)
                    lbl1 = tk.Label(book, text ="$"+str(snckmoney),fg="black",font=("Arial", 12))
                    lbl1.place(x = 250, y = 200)
                    lbl2 = tk.Label(book, text ="$"+str(seatmoney),fg="black",font=("Arial", 12))
                    lbl2.place(x = 350, y = 200)
                    lbltot = tk.Label(book, text ="Total :",fg="black",font=("Arial", 12))
                    lbltot.place(x = 200, y = 230)
                    lbl3 = tk.Label(book, text ="$"+str(total),fg="black",font=("Arial", 12))
                    lbl3.place(x = 250, y = 230)
                    lblfinal = tk.Label(book, text ="Total after discount :",fg="black",font=("Arial", 12))
                    lblfinal.place(x = 200, y = 260)
                    lbl4 = tk.Label(book, text ="$"+str(final),fg="black",font=("Arial", 12))
                    lbl4.place(x = 350, y = 260)
                    lbldis = tk.Label(book, text ="Discounts",fg="black", borderwidth=2, relief='solid',
font=("Arial", 10,'underline','bold'))
                    lbldis.place(x = 180, y = 310)
                    lbldis1 = tk.Label(book, text ="1. For booking more than 3 seats: Reduction of 20% from
the cost of seats\n 2. For selecting more than 2 snacks: Reduction of 10% from the cost of
snacks",fg="black", borderwidth=2, relief='solid', font=("Times New Roman", 8))
                    lbldis1.place(x = 20, y = 330)


                elif nseat<1:
                        messagebox.showinfo("Booking failed ", "Please Book Seats")
                elif ver==0:
                        messagebox.showinfo("Booking failed ", "Please Login First")

            def snacks():
                global snckmoney
                snack_quantities = {'Popcorn': 0, 'Nachos': 0, 'Pepsi': 0, 'Burger': 0, 'Hotdog': 0}
                def update_display():
                    global snckmoney
                    snckmoney = sum(snack_quantities[snack] * prices[snack] for snack in snacks_list)
                    b1.config(text="$" + str(snckmoney))

                def clear():
                    for snack in snacks_list:
                        scale_vars[snack].set(0)
                        snack_quantities[snack] = 0
                    update_display()

                def on_scale_change(event):
                    snack = event.widget.snack
                    snack_quantities[snack] = scale_vars[snack].get()
                    update_display()
                    global pop,hotdog,nacho,pepsi,burger
                    if snack == 'Popcorn' and snack_quantities[snack] > 0:
                        pop = 1
                    if snack == 'Hotdog' and snack_quantities[snack] > 0:
                        hotdog = 1
                    if snack == 'Burger' and snack_quantities[snack] > 0:
                        burger = 1
                    if snack == 'Pepsi' and snack_quantities[snack] > 0:
                        pepsi = 1
                    if snack == 'Nachos' and snack_quantities[snack] > 0:
                        nacho = 1

                frmsnck = tk.Tk()
```

```
                frmsnck.geometry("400x500")
                frmsnck.title("Snacks")
                frmsnck.configure(bg='black')

                btclr = tk.Button(frmsnck, text="Clear", fg='black', bg='Red', font=("Times New Roman Bold",
10), command=clear)
                btclr.place(x=250, y=400, width=50)

                btbook = tk.Button(frmsnck, text="Book", fg='black', bg='Red', font=("Times New Roman Bold",
10), command=book)
                btbook.place(x=300, y=400, width=50)

                lbltot = tk.Label(frmsnck, text="Total : ", fg='black', bg='white', font=('Times New Roman',
12, 'bold', 'underline'))
                lbltot.place(x=180, y=350)

                b1 = tk.Label(frmsnck, text="$" + str(snckmoney))
                b1.place(x=230, y=350, width=70)

                snacks_list = ['Popcorn', 'Nachos', 'Pepsi', 'Burger', 'Hotdog']
                prices = {'Popcorn': 5, 'Nachos': 7, 'Pepsi': 2, 'Burger': 15, 'Hotdog': 10}
                scale_vars = {}

                for idx, snack in enumerate(snacks_list):
                    lbl = tk.Label(frmsnck, text=f"{snack} (Cost: ${prices[snack]})", fg='black', bg='grey',
font=('Times New Roman', 10, 'bold'))
                    lbl.place(x=50, y=100 + idx * 40)

                    scale_vars[snack] = tk.Scale(frmsnck, from_=0, to=5, orient=tk.HORIZONTAL, length=150)
                    scale_vars[snack].place(x=200, y=100 + idx * 40)
                    scale_vars[snack].snack = snack
                    scale_vars[snack].bind("<ButtonRelease-1>", on_scale_change)

            def booking():
                def taken():
                    nonlocal buttons
                    try:
                        myconn=mys.connect(host="localhost",user="root",passwd="adis",database="movie")
                        mycur=myconn.cursor()

                        query = "select seats from raastha"
                        mycur.execute(query)
                        result=mycur.fetchall()
                        booked_seats = []
                        for tup in result:
                            elements = [elem.strip() for elem in tup[0].split(',')]
                            booked_seats.extend([elem for elem in elements if elem])
                        myconn.commit()
                        for row in range(len(buttons)):
                            for col in range(len(buttons[row])):
                                seat_label = chr(ord('a') + row) + str(col + 1)
                                if seat_label in booked_seats:
                                    buttons[row][col].configure(bg="grey", state=tk.DISABLED)

                    except Exception as e:
                        print(e)

                def clear():
                    for button_row in buttons:
                        for button in button_row:
                            button.configure(bg="green")
                    global seatmoney, nseat
                    seatmoney = 0
                    nseat = 0
                    update_display()
                    seatraastha.clear()

                def update_display():
                    b1.config(text="$" + str(seatmoney))

                def button_click(row, col):
                    global seatmoney, nseat
                    button = buttons[row][col]
                    seat_label = chr(ord('a') + row) + str(col + 1)
                    if button["bg"] == "green":
                        button.configure(bg="red")
                        seatmoney += 20
                        nseat += 1
                        seatraastha.append(seat_label)
```

```
                    else:
                        button.configure(bg="green")
                        seatmoney -= 20
                        nseat -= 1
                        seatraastha.remove(seat_label)

                    update_display()

                frmbook = tk.Tk()
                frmbook.geometry("400x450")
                frmbook.title("Seat")
                frmbook.configure(bg='black')
                lblscr = tk.Label(frmbook, text="------------------screen--------------------", fg='white',
bg='black',
                                  font=('Times New Roman', 20))
                lblscr.place(x=5, y=100)

                btclr = tk.Button(frmbook, text="Clear", fg='black', bg='Red', font=("Times New Roman Bold",
10), command=clear)
                btclr.place(x=250, y=400, width=50)

                btbook = tk.Button(frmbook, text="Book", fg='black', bg='Red', font=("Times New Roman Bold",
10), command=book)
                btbook.place(x=300, y=400, width=50)

                b1 = tk.Button(frmbook, text="$" + str(seatmoney))
                b1.place(x=120, y=400, width=70)

                lbla = tk.Label(frmbook, text="a", fg='white', bg='black', font=("Times New Roman Bold", 12))
                lbla.place(x=5, y=150)
                lblb = tk.Label(frmbook, text="b", fg='white', bg='black', font=("Times New Roman Bold", 12))
                lblb.place(x=5, y=200)
                lblc = tk.Label(frmbook, text="c", fg='white', bg='black', font=("Times New Roman Bold", 12))
                lblc.place(x=5, y=250)
                lbld = tk.Label(frmbook, text="d", fg='white', bg='black', font=("Times New Roman Bold", 12))
                lbld.place(x=5, y=300)
                lblno = tk.Label(frmbook, text="1   2   3   4   5   6   7   8   9  10  11 12 13 14 15 16 17
18", fg='white', bg='black', font=("Times New Roman Bold", 12))
                lblno.place(x=20, y=325)


                buttons = []


                for row in range(4):
                    button_row = []
                    for col in range(18):
                        seat_label = chr(ord('a') + row) + str(col + 1)
                        button = tk.Button(frmbook, bg='green', height=1, width=1, command=lambda r=row,
c=col: button_click(r, c))
                        button.place(x=20 + col * 20, y=150 + row * 50)
                        button_row.append(button)
                    buttons.append(button_row)
                taken()

            lblhd = tk.Label(frmtime, text ="Raastha", bg= 'light green',font=('MV
Boli',15,'bold','underline'))
            lblhd.place(x = 170, y = 10)
            btbook= tk.Button(frmtime, text ="Book seat",fg ='red',bg= 'light green',font=("Times New Roman
Bold", 10),command=booking)
            btbook.place(x = 250, y = 155, width = 100)
            btsnack= tk.Button(frmtime, text ="Snacks",fg ='red',bg= 'light green',font=("Times New Roman
Bold", 10),command=snacks)
            btsnack.place(x = 250, y = 185, width = 100)
            lbldescrpt = tk.Label(frmtime, text ="Description", bg= 'light green',font=('Times New Roman',12,
'underline'))
            lbldescrpt.place(x = 2, y = 40)
            lbldes = tk.Label(frmtime, text ="Shahana and Faizal arrive in Oman to find Shahana's mother,
missing for\n  22 years. Facing difficulties, they journey deeper into the desert, lost.\n Police try to
locate them.",
                              bg= 'light green',font=('Times New Roman',10))
            lbldes.place(x = 2, y = 65)
            lblgenre = tk.Label(frmtime, text ="Genre", bg= 'light green',font=('Times New Roman',12,
'underline'))
            lblgenre.place(x = 2, y = 110)
            lblgen = tk.Label(frmtime, text ="Thriller", bg= 'light green',font=('Times New Roman',10))
            lblgen.place(x = 2, y = 135)
```

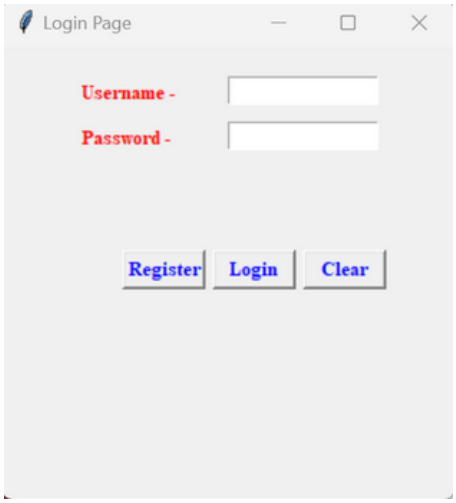
```
            lblrt = tk.Label(frmtime, text ="Running time", bg= 'light green',font=('Times New Roman',12,
'underline'))
            lblrt.place(x = 2, y = 155)
            lblrun = tk.Label(frmtime, text ="140 mins", bg= 'light green',font=('Times New Roman',10))
            lblrun.place(x = 2, y = 180)
            lblstar = tk.Label(frmtime, text ="Starring", bg= 'light green',font=('Times New Roman',12,
'underline'))
            lblstar.place(x = 2, y = 200)
            lblact = tk.Label(frmtime, text ="Sarjano Khalid, Anagha Narayanan, Aradhya Ann", bg= 'light
green',font=('Times New Roman',10))
            lblact.place(x = 2, y = 225)

            link = tk.Label(frmtime, text="Trailer",
            fg="blue", bg="light green",cursor="hand2",font=['Times',15,'underline'])
            link.place(x=160, y=155)
            link.bind("<Button-1>", lambda e:
webbrowser.open_new("https://youtu.be/3j99pxFEuys?si=2sYHG-WfowDRGSFb"))

        img1 = ImageTk.PhotoImage(Image.open("neru.jpg"))
        button1= Button(frmmal, image = img1, command=showtimes1)
        button1.place(x = 50, y = 150)
        img2 = ImageTk.PhotoImage(Image.open("raastha.jpg"))
        button2= Button(frmmal, image = img2, command=showtimes2)
        button2.place(x = 350, y = 150)
        root.mainloop()

    bteng = tk.Button(root, text ="English",fg ='red',bg= 'orange',font=("Segoe Script Bold",
20),command=English)
    bteng.place(x = 170, y = 300, height = 100, width = 180)
    bthin = tk.Button(root, text ="Hindi",fg ='red',bg= 'orange',font=("Segoe Script Bold",
20),command=Hindi)
    bthin.place(x = 420, y = 300, height = 100, width = 180)
    btmal = tk.Button(root, text ="Malayalam",fg ='red',bg= 'orange',font=("Segoe Script Bold",
20),command=Malayalam)
    btmal.place(x = 670, y = 300, height = 100, width = 180)
    imgadm = ImageTk.PhotoImage(Image.open("admin.png"))
    btadm= Button(root, image = imgadm, command=admin)
    btadm.place(x = 10, y = 5)



    def loginscreen():
        def funclear():
            txtUser.delete(0,END)
            txtpass.delete(0,END)
            txtUser.focus()

        def funlogin():
            user = txtUser.get()
            passw = txtpass.get()
            try:
                myconn=mys.connect(host="localhost",user="root",passwd="adis",database="movie")
                mycur=myconn.cursor()
                query="select * from customer where c_name='{}' and pass='{}'".format(user,passw)
                mycur.execute(query)
                rs=mycur.fetchone()
                if rs!=None and user==rs[0] and passw==rs[2]:
                    global ver
                    ver=1
                    global user1
                    user1=user
                    passw=txtpass.get()
                    l=user+passw
                    global i_d1
                    for ch in l:
                        i_d1+=str(ord(ch))
                    messagebox.showinfo("Login ", "Login Successful")
                    login.destroy()
                    btlogin.destroy()
                    lblacc = tk.Label(root, text ="Account name: ", fg ='red', bg= 'black',font=('Times New
Roman',10, "underline"))
                    lblacc.place(x = 850, y = 5)
                    lblus = tk.Label(root, text =user, fg ='red', bg= 'black',font=('Times New Roman',10,
"underline"))
                    lblus.place(x = 935, y = 5)

                else:
                    messagebox.showinfo("Login", "Login Failed")
            except Exception as e:
```

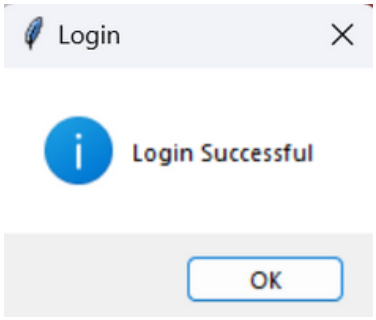
```
                print(e)

        def register():
            i_d=''
            user=txtUser.get()
            passw=txtpass.get()
            l=user+passw
            for ch in l:
                i_d+=str(ord(ch))
            try:
                myconn=mys.connect(host="localhost",user="root",passwd="adis",database="movie")
                mycur=myconn.cursor()
                query="insert into customer values('{}',{},'{}')".format(user,i_d,passw)
                mycur.execute(query)
                myconn.commit()
                messagebox.showinfo("Registration Successful", "Registration Successful, You may login now")
                funclear()
            except Exception as e:
                messagebox.showinfo("Registration Failed", "Registration Failed, Account already exists")


        login = tk.Tk()
        login.geometry("300x300")
        login.title("Login Page")
        C = Canvas(login, bg ="Red", height = 250, width = 300)
        lbluser = tk.Label(login, text ="Username -",fg="red",font=("Times New Roman Bold", 10) )
        lbluser.place(x = 50, y = 20)
        txtUser = tk.Entry(login, width = 35,fg="blue",font=("Times New Roman Bold", 10))
        txtUser.place(x = 150, y = 20, width = 100)
        lblpass = tk.Label(login, text ="Password -",fg="red",font=("Times New Roman Bold", 10))
        lblpass.place(x = 50, y = 50)
        txtpass = tk.Entry(login,show="*", width = 35,fg="blue",font=("Times New Roman Bold", 10))
        txtpass.place(x = 150, y = 50, width = 100)
        loginbtn = tk.Button(login, text ="Login", fg ='blue', font=("Times New Roman Bold", 10),command =
funlogin)
        loginbtn.place(x = 140, y = 135, width = 55)
        regbtn = tk.Button(login, text ="Register", fg ='blue', font=("Times New Roman Bold", 10),command =
register)
        regbtn.place(x = 80, y = 135, width = 55)
        clearbtn = tk.Button(login, text ="Clear",fg ='blue', font=("Times New Roman Bold", 10),command =
funclear)
        clearbtn.place(x = 200, y = 135, width = 55)

    btlogin = tk.Button(root, text ="login",fg ='white',bg= 'red',font=("Times New Roman Bold",
10),command=loginscreen)
    btlogin.place(x = 930, y = 5, width = 60)

    root.mainloop()


def WELCOMESCREEN():
    import turtle
    win = turtle.Screen()
    win.bgcolor('black')

    polygon = turtle.Turtle()
    num_sides = 4
    side_length = 500
    side_width = 400
    polygon.speed(10)
    polygon.penup()
    polygon.goto(-200,200)

    for i in range(2,10,2):#i=0,1,2,3,4,5,6
        color = ["red","blue","blue","red"]
        for i in range(num_sides):
            polygon.color(color[i%4])
            polygon.pendown()
            polygon.forward(side_length)
            polygon.right(90)
            polygon.forward(side_width)
            polygon.right(90)

            side_length-=10
            side_width-=10

    polygon.hideturtle()

    turtle.goto(50, 0)
```

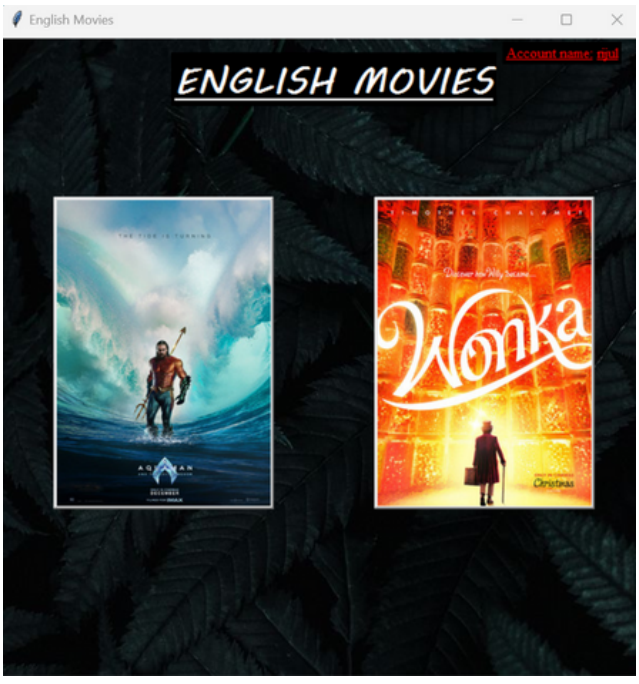
```
    turtle.hideturtle()

    spiral = turtle.Turtle()
    spiral.speed(0)
    spiral.penup()
    spiral.goto(0,300)

    colors = ["red", "orange", "yellow", "green", "blue", "purple"]


    for i in range(50):
        spiral.pendown()
        spiral.color(colors[i % 6])
        spiral.forward(i * 3 / 360 + i)
        spiral.left(59)
    spiral.penup()
    spiral.goto(-250,0)
    for i in range(50):
        spiral.pendown()
        spiral.color(colors[i % 6])
        spiral.forward(i * 3 / 360 + i)
        spiral.left(59)
    spiral.penup()
    spiral.goto(0,-300)
    for i in range(50):
        spiral.pendown()
        spiral.color(colors[i % 6])
        spiral.forward(i * 3 / 360 + i)
        spiral.left(59)
    spiral.penup()
    spiral.goto(-300,-250)
    for i in range(50):
        spiral.pendown()
        spiral.color(colors[i % 6])
        spiral.forward(i * 3 / 360 + i)
        spiral.left(59)
    spiral.penup()
    spiral.goto(300,250)
    for i in range(50):
        spiral.pendown()
        spiral.color(colors[i % 6])
        spiral.forward(i * 3 / 360 + i)
        spiral.left(59)
    spiral.penup()
    spiral.goto(-300,250)
    for i in range(50):
        spiral.pendown()
        spiral.color(colors[i % 6])
        spiral.forward(i * 3 / 360 + i)
        spiral.left(59)
    spiral.penup()
    spiral.goto(280,-280)
    for i in range(50):
        spiral.pendown()
        spiral.color(colors[i % 6])
        spiral.forward(i * 3 / 360 + i)
        spiral.left(59)
    spiral.penup()
    spiral.goto(350,-50)
    for i in range(50):
        spiral.pendown()
        spiral.color(colors[i % 6])
        spiral.forward(i * 3 / 360 + i)
        spiral.left(59)
    turtle.color("white")
    turtle.write("O'Brien Theater", move=True, font=("Algerian",40,"bold"), align="center")
    turtle.exitonclick()

WELCOMESCREEN()
MENUSCREEN()
```

SAMPLE OUTPUT

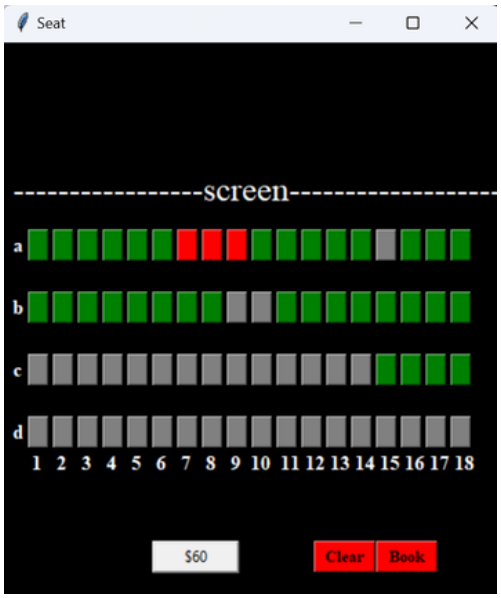

KALAVAMPARA THEATER
KALAVAMPARA THEATRE
login
Select your language preference:
English
Hindi
Malayalam

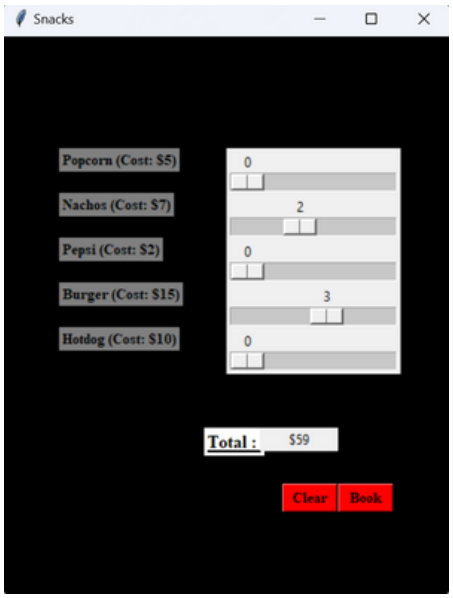

Login Page
Username -
Password -
Register
Login
Clear

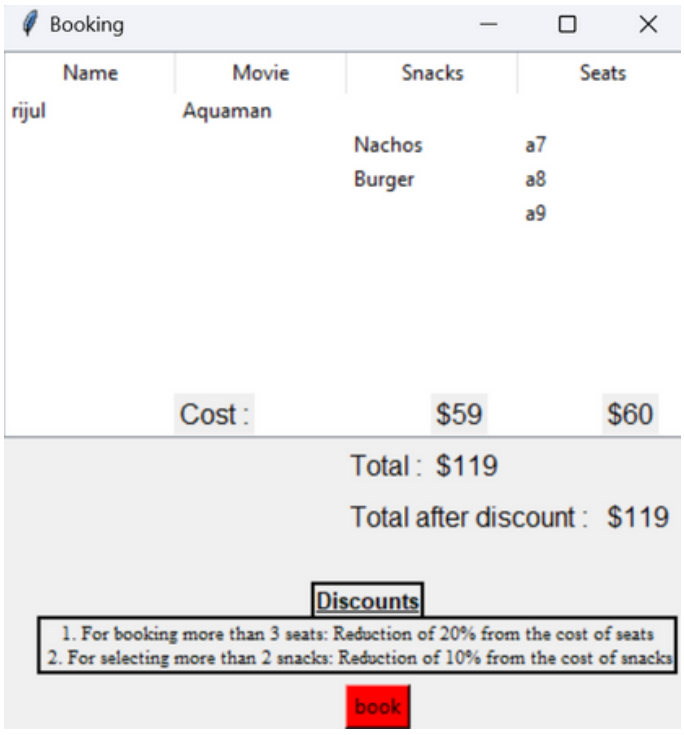

Login
Login Successful
OK

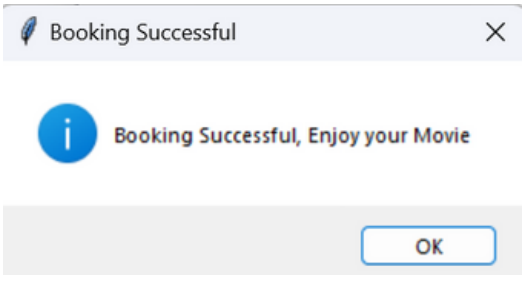

English Movies
Account name: rijul
ENGLISH MOVIES
Wonka


Movie
Aquaman & The Lost Kingdom
Description
When an ancient power is unleashed, Aquaman must forge an uneasy alliance with an unlikely ally to protect Atlantis, and the world, from irreversible devastation.
Genre
Action
Running time
125 mins
Trailer
Book seat
Snacks
Starring
Jason Momoa, Amber Heard, Patrick Wilson, Dolph Lundgren, Yahya Abdul Mateen II


Seat
screen
a
b
c
d
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
$60
Clear
Book


Snacks
Popcorn (Cost: $5)
Nachos (Cost: $7)
Pepsi (Cost: $2)
Burger (Cost: $15)
Hotdog (Cost: $10)
0
2
0
3
0
Total :
$59
Clear
Book


Booking
Name
Movie
Snacks
Seats
rijul
Aquaman
Nachos
a7
Burger
a8
a9
Cost :
$59
$60
Total : $119
Total after discount : $119
Discounts
1. For booking more than 3 seats: Reduction of 20% from the cost of seats
2. For selecting more than 2 snacks: Reduction of 10% from the cost of snacks
book


Booking Successful
Booking Successful, Enjoy your Movie
OK

MERITS AND DEMERITS

Merits (Advantages)

User-Friendly Interface

The project is designed with an intuitive user interface that allows users to easily select seats, making it accessible even to those with limited technical skills.

Efficient Seat Management

The use of buttons to represent seats and color coding (e.g., green for available, red for selected, grey for booked) makes it easy for users to understand seat availability visually.

Real-Time Database Integration

By using MySQL to store and retrieve booking data, the project ensures that seat availability is updated in real-time, preventing double bookings and enhancing reliability.

Clear and Responsive Feedback

Features like the "clear" button and total price display provide instant feedback, allowing users to adjust their selections easily and understand the cost implications immediately.

Receipt Generation Capability

Once implemented, the receipt generation functionality will provide users with confirmation of their booking, enhancing user satisfaction and providing a reliable reference.

Scalability

The modular structure of the project can potentially support future expansions, such as adding different theaters or movies, due to its organized layout and SQL database integration.

Demerits (Disadvantages)

Lack of Advanced Error Handling

If unexpected issues occur, such as database connection errors, the user experience may suffer due to insufficient error handling or feedback for the user on such issues.

Performance Limitations

If the application handles a large number of users or seats simultaneously, performance may slow down, as Python isn′ t as fast for high-concurrency tasks as some other languages.

Limited Aesthetic Customization

Since the interface is built in Tkinter, which has basic GUI capabilities, the design may appear outdated or simplistic compared to modern booking applications.

Potential Data Integrity Issues

Without thorough validation, there could be cases of data inconsistencies if a seat state is not updated properly in the database. For example, if a booking attempt fails, the system might need additional checks to reset seat availability.

No Online Accessibility

As a desktop-based application, users need to install it on their device, which limits accessibility compared to a web-based application that could be accessed from any device.

Dependency on Local Database Setup

The application requires a local MySQL server setup, which could be a barrier for users who aren′ t familiar with database configuration, potentially limiting the user base.

BIBLIOGRAPHY

To develop this project many references were used:

1. Computer Science Textbook Class 12: Sumita Arora
2. Chapter Notes
3. https://www.google.com
4. https://www.python.org
5. https://www.mysql.org
6. https://www.stackoverflow.com
7. https://www.geeksforgeeks.org