Multiplex Management System

Version 1.0 (2020-2021)

Computer Science (083) Project

Developed By

Kavya Manchanda

Priyanshi Ahuja

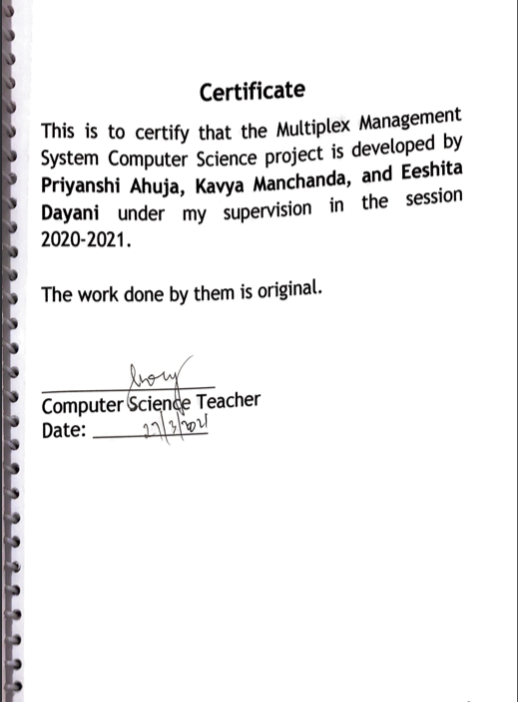
Eeshita Dayani

Delhi Public School, R.K.Puram, New Delhi

www.dpsrkp.net

Index

| Sno | Description | Pageno |
| --- | --- | --- |
| 1 | Certificate | 3 |
| 2 | Acknowledgement & References | 4 |
| 3 | Introduction | 5-7 |
| 4 | Source Code | 8-18 |
| 5 | Output Screen | 19-28 |
| 6 | Hardware & Software requirement | 29 |

****

Here is the scanned copy of the signature page of the checked project file. File was signed by Mohitendra Dey sir, my CS teacher. I got my printed project file checked before schools were shut down and boards cancelled. The printed and checked version is a printout of this document.

**Acknowledgement**

I would like to express my sincere gratitude to my computer teacher Mr. Mohitendra Dey for his vital support, guidance and encouragement without which this project would not come forth from my side, who helped me complete the project by giving ideas,thoughts and made this project easy and accurate.

I wish to thank my parents for their undivided support and interest who inspired me and encouraged me to go my own way, without which I would be unable to complete my project.

**Reference**

1.Classnotes.

**Introduction**

Keeping in view of the growing requirements of effective and efficient solutions for various problems, with our knowledge of Python and SQL, in this project, we have tried to provide a simple solution for the Multiplex Industry.

Our project has complete management solutions to take care of all the bookings in a Multiplex facility (modifying movie lists and descriptions, member management and booking movies)

We have divided the program into small user-defined functions to take care of all future upgrades and expansion. The following is the list of functions along with their description. We have also included an indicator of whom this function can be used by i.e. the user or the administrator.

| Sno | Function Name | Description | Type |
| --- | --- | --- | --- |
| 1 | **def ViewMenu()** | To display the table of the movies currently available with their code, name/type, rating, timings, hall, age rating, ticket prices, cast and seats available. | Admin & User function |
| 2 | **def AddMovie()** | To add a new movie to the existing movie table | Admin function |
| 3 | **def PriceUpdate()** | To update the price of an existing movie in the movie table | Admin function |
| 4 | **def ChangeTimings()** | To update the timings of an existing movie in the movie table | Admin function |
| 5 | **def DeleteMovie()** | Delete a movie from the movie table | Admin function |
| 6 | **def ViewAllMemberInfo()** | View the member table | Admin function |
| 7 | **def BookTicket()** | Use the fields in the table to book a ticket for a customer—charge them to ticket fees and reduce overall availability of the number of tickets | User function |
| 8 | **def Register()** | To register a new member into the Multiplex system | User function |
| 9 | **def ViewUserInfo()** | To ask for a registration number and display all relevant information for the respective member | User function |
| 10 | **def UpdateInfo()** | To update the information of an existing member | User function |
| 11 | **def DeleteInfo()** | To delete a member from the table | User function |

**Binary Files/CSV Files/Tables with structure used in the program**

**File/Table Names: MemberInfoTable.csv**

| **Sno** | **FieldName/ Column Name** | **Data Type** | **Description** |
| --- | --- | --- | --- |
| **1** | **Name** | **String** | **Contains the names of the MultiPlex customer members** |
| **2** | **Gender** | **String** | **Contains the gender of the members** |
| **3** | **Age** | **Integer** | **Age of the members** |
| **4** | **Phone Number** | **Integer** | **Phone number/contact details of the member** |
| **5** | **Email** | **String** | **Email of the member** |
| **6** | **Membership Number** | **String** | **Membership Number of the member** |
| **7** | **Membership Expiry Date** | **String** | **The expiry date of their membership** |
| **8** | **Membership Level** | **String** | **Level of their membership on the basis of which the movie discount will be decided (Normal, Gold, Platinum)** |

**File/Table Name: MovieTable2.csv**

| **Sno** | **FieldName/ Column Name** | **Data Type** | **Description** |
| --- | --- | --- | --- |
| **1** | **Movie Code** | **String** | **A unique code given to each movie playing; used for recognition purposes** |
| **2** | **Movie name and Type** | **String** | **Name of the movie and type i.e. Regular/3D** |
| **3** | **Rating** | **String** | **Critique ratings given to the movie; informs customers of the quality of each movie** |
| **4** | **Show timings and Hall** | **String** | **Timings and Placement of the movie for ease in customer service and selection** |
| **5** | **Age Rating** | **String** | **Age rating given to the movie ie. 10+/18+** |
| **6** | **Ticket Prices** | **Integer** | **Price per ticket/per seat** |
| **7** | **The cast of the Movie** | **String** | **Actors and actresses of the movie** |
| **8** | **Seats available** | **String** | **Maximum seating capacity of the movie; the number reduces as the admin sells the seats through our management system** |

**External Packages/Modules used in the Program/Software**

Google.colab   
CSV  
Random  
Datetime

**Salient Features of the Project**

* User-friendly menu-driven options
* Exceptions used to avoid inconvenience to users
* Data Validation to avoid ambiguous/wrong/invalid data entry
* Inclusive options
* Various options for the functionality
* Clear division and roles for the users of this management system

**Source Code**

**# Project Title : Multiplex Management System**

**# Version : 1.0 2020-2021**

**# Developed By : Eeshita Dayani, Kavya Manchanda, Priyanshi Ahuja**

**# Guide : Mr. Mohitendra Dey**

**# Last Updated On: 2020-12-12**

1) from google.colab import files

uploaded = files.upload()  
  
2) from google.colab import files

uploaded = files.upload()

print("Welcome to PVR Cinemas!")

import csv

import random

from datetime import date

def ViewMenu():

try:

with open("Movietable2.csv","r") as f:

cR=csv.reader(f)

print('-'\*240)

for r in cR:

print("%10s | %25s | %7s | %20s | %10s | %18s | %32s | %17s |"%(r[0].ljust(10), r[1].ljust(25), r[2].ljust(7), r[3].ljust(20), r[4].ljust(10), r[5].ljust(18), r[6].ljust(32), r[7].ljust(17)))

except FileNotFoundError:

print("Movietable2.csv File Not Found")

#ViewMenu()

def AddMovie():

q="Y"

while q=="Y":

ViewMenu()

print("These time slots are occupied")

with open("Movietable2.csv","a") as F:

cW=csv.writer(F)

a,b,c=random.randint(0,9),random.randint(0,9),random.randint(0,9)

mcode\_S=str(a)+str(b)+str(c)

mcode= int(mcode\_S)

movie=input("Enter movie name and type:")

rating=input('Enter movie Rating:')

Showtime=input("Enter show timings and Hall:")

age=input("Enter age rating:")

ticketprice=int(input("Enter ticket price:"))

Cast=input("Enter movie cast:")

cW.writerow([mcode,movie,rating,Showtime,age,ticketprice,Cast,100])

print('Movie has been added!')

q=input("Want to enter more?(Y/N)").upper()

if q=="N":

print("Thank you!")

break

else:

print("invalid input")

ViewMenu()

#AddMovie()

def PriceUpdate():

ViewMenu()

try:

with open('Movietable2.csv', 'r+') as cF:

cR = csv.reader(cF)

L=[]

M\_Code=input('Enter Movie code:')

for row in cR:

if row[0]==M\_Code:

print('Okay! Current price of the movie is:', "Rs.", row [5])

i=row

else:

L.append(row)

with open("Movietable2.csv","w") as x:

cW=csv.writer(x)

Upd\_price=input('Enter updated price:')

i[5]=Upd\_price

L.append(i)

cW.writerows(L)

print("Updated")

ViewMenu()

except FileNotFoundError:

print("Movietable2.csv File not found")

#PriceUpdate()

def ChangeTimings():

ViewMenu()

print("These time slots are occupied")

try:

with open('Movietable2.csv', 'r+') as cF:

cR = csv.reader(cF)

M\_Code=input('Enter Movie code:')

L=[]

for row in cR:

if row[0]==M\_Code:

print('Okay! Current timings of the movie are:', row [3])

i=row

else:

L.append(row)

with open("Movietable2.csv","w") as d:

cW=csv.writer(d)

Upd\_timings=input('Enter updated timings:')

i[3]=Upd\_timings

L.append(i)

cW.writerows(L)

print("Updated")

ViewMenu()

except FileNotFoundError:

print("Movietable2.csv File not found")

#ChangeTimings()

def DeleteMovie():

lines = []

ViewMenu()

movies= input("Please enter a movie's code to be deleted")

try:

with open('Movietable2.csv', 'r') as readFile:

reader = csv.reader(readFile)

for row in reader:

lines.append(row)

for field in row:

if field == movies:

lines.remove(row)

with open('Movietable2.csv', 'w') as writeFile:

writer = csv.writer(writeFile)

writer.writerows(lines)

except:

print("movie not found, please enter a valid input")

def BookTicket():

Today = date.today()

j=input("Enter your membership number:")

with open("MemberInfoTable.csv","r") as c:

cr=csv.reader(c)

for x in cr:

if x[5]==j:

a=x[2]

i=x

break

print("Welcome",i[0])

ViewMenu()

m=int(input("Which movie do you want to book. Enter Code?"))

with open("Movietable2.csv","r") as t:

u=csv.reader(t)

L=list(u)

S=L[1::]

for n in S:

if int(n[0])==int(m):

b=n[4]

if int(a)<int(b[0:2]):

print("You are not eligible to watch this movie")

return

with open("Movietable2.csv","r+") as t:

u=csv.reader(t)

cf=csv.writer(t)

L=list(u)

for R in L[1::]:

if int(R[0])==int(m):

print("Okay, the movie timings are",R[3])

J=input("Would you like to proceed?(Y/N)").upper()

if J=="Y":

print('We have ',R[7],"seats available")

s=int(input("How many seats would you like to book?"))

if s>int(R[7]):

print('We do not have',s,"seats available today. Sorry for the inconvenience caused.")

else:

R[7]=int(R[7])-int(s)

iprice=int(R[5])

if i[7]=="Normal":

fprice=(iprice\*(95/100))\*s

elif i[7]=="Gold":

fprice=(iprice\*(92/100))\*s

elif i[7]=="Platinum":

fprice=(iprice\*(90/100))\*s

print("Your ticket has been booked for",R[1],"at",R[3],"on",Today,"for",s,"seats. Please pay an amount of",fprice)

break

elif J=="N":

break

with open("Movietable2.csv","w") as t:

cf=csv.writer(t)

cf.writerows(L)

#BookTicket()

def ViewAllMemberInfo():

try:

with open('MemberInfoTable.csv', 'r') as cF:

cR = csv.reader(cF)

print('-'\*250)

for r in cR:

print(" %15s | %10s | %5s | %12s | %25s | %18s | %21s | %20s |"%( r[0].ljust(15), r[1].ljust(10), r[2].ljust(5), r[3].ljust(12),r[4].ljust(25), r[5].ljust(18), r[6].ljust(21), r[7].ljust(20)))

except FileNotFoundError:

print("MemberInfoTabletable.csv File Not Found")

#ViewAllMemberInfo()

def Register():

today = date.today()

print("Today's date:", today)

print("Welcome!")

with open('MemberInfoTable.csv',"a") as f:

cW=csv.writer(f)

while True:

name=input("Enter your name:")

gender=input("Enter gender:")

age=int(input("Enter your age:"))

phoneno=int(input("Enter phone number:"))

email=input("Enter email address (optional,press space if you don't have an email):")

a,b,c,d=random.randint(0,9),random.randint(0,9),random.randint(0,9),random.randint(0,9)

List=['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z']

f=random.choice(List)

mno=str(f)+str(a)+str(b)+str(c)+str(d)

res=input("What membership level do you want? (N: Normal ; G: Gold ; P: Platinum)").upper()

mex=int(input("For how many years do you want the membership?"))

if res=='N':

Mb\_price= int(mex) \* 100

mlevel='Normal'

elif res=='G':

Mb\_price= int(mex) \* 200

mlevel='Gold'

elif res=='P':

Mb\_price=int(mex) \* 300

mlevel='Platinum'

Date=str(today)

print("Congratulations! You have been registered as a",mlevel,"member till",int(Date[0:4])+int(mex),"-",Date[5:7],"-",Date[8:10])

print('Your membership number is', mno)

print("Please pay an amount of Rs.", Mb\_price)

Y=str(int(Date[0:4])+int(mex))+"-"+str(Date[5:7])+"-"+str(Date[8:10])

cW.writerow([name,gender,age,phoneno,email,mno,Y,mlevel])

q=input("Want to register another member?(Y/N)").upper()

if q=="N":

break

elif q=="Y":

continue

else:

print("invalid input")

#Register()

def ViewUserInfo():

try:

with open('MemberInfoTable.csv', 'r') as cF:

cR = csv.reader(cF)

Member\_no=input('Enter your Membership Number:')

for row in cR:

if row[5]==Member\_no:

print('Your information:', row)

except FileNotFoundError:

print("MemberInfoTable.csv File Not Found")

#ViewUserInfo()

def UpdateInfo():

try:

with open("MemberInfoTable.csv","r+") as cF:

cR=csv.reader(cF)

L=[]

M\_no=input('Enter membership number:')

l=[]

c=0

for i in cR:

l.append(i[5])

if M\_no in l:

c=1

else:

print("Sorry, membership number not found")

if c==1:

with open("MemberInfoTable.csv","r+") as cF:

cR=csv.reader(cF)

for row in cR:

if row[5]==M\_no:

print("Your information is:",row)

x=row

else:

L.append(row)

with open("MemberInfoTable.csv","w") as cD:

cW=csv.writer(cD)

choice=input('Do you want to update phone number or email? (Enter P/E)').upper()

if choice== 'P':

u=int(input("enter new phone number"))

x[3]=u

elif choice== 'E':

e=input("enter e-mail address")

x[4]=e

print("Updated!")

L.append(x)

cW.writerows(L)

except FileNotFoundError:

print("MemberInfoTable.csv not found")

#UpdateInfo()

def DeleteInfo():

try:

with open('MemberInfoTable.csv', 'r+') as f:

cF=csv.reader(f)

cW=csv.writer(f)

L=[]

M\_no=input('Enter your membership number:')

for row in cF:

if not(M\_no==row[5]):

L.append(row)

with open('MemberInfoTable.csv', 'w') as f:

cW=csv.writer(f)

cW.writerows(L)

print("Your account has been deleted. Thank you!")

except FileNotFoundError:

print("MemberInfoTable.csv File Not Found")

#DeleteInfo()

while True:

ASK=(input('Select choice: U:User ; A:Admin ; Q:Quit')).upper()

if ASK=='A':

while True:

x=input('Enter password:')

if x=='admin123':

break

else:

print('Incorrect password!')

while True:

OPT=(input('VW: View Movie Menu ; ADD: Add Movie ; PRC: Price Update ; TM: Change Timings ; VWMB: View Members Information ; DELM: Delete Movie Q: Quit ')).upper()

if OPT=='VW':

ViewMenu()

elif OPT=='ADD':

AddMovie()

elif OPT=='PRC':

PriceUpdate()

elif OPT=='TM':

ChangeTimings()

elif OPT=='VWMB':

ViewAllMemberInfo()

elif OPT=='DELM':

DeleteMovie()

elif OPT=="Q":

print("Thank you")

break

else:

print("Please enter valid option")

elif ASK=='U':

while True:

OPT=(input('RG: Register ; VWIN: View User Info ; VW: View Movie Menu ; CH: Book your Ticket ; UPD: Update Information ; DEL: Delete your account, Q: Quit ')).upper()

if OPT=='RG':

Register()

elif OPT=='VWIN':

ViewUserInfo()

elif OPT=='VW':

ViewMenu()

elif OPT=='CH':

BookTicket()

elif OPT=="UPD":

UpdateInfo()

elif OPT=="DEL":

DeleteInfo()

elif OPT=='Q':

print("Thank you")

break

else:

print("Please enter valid option")

elif ASK=='Q':

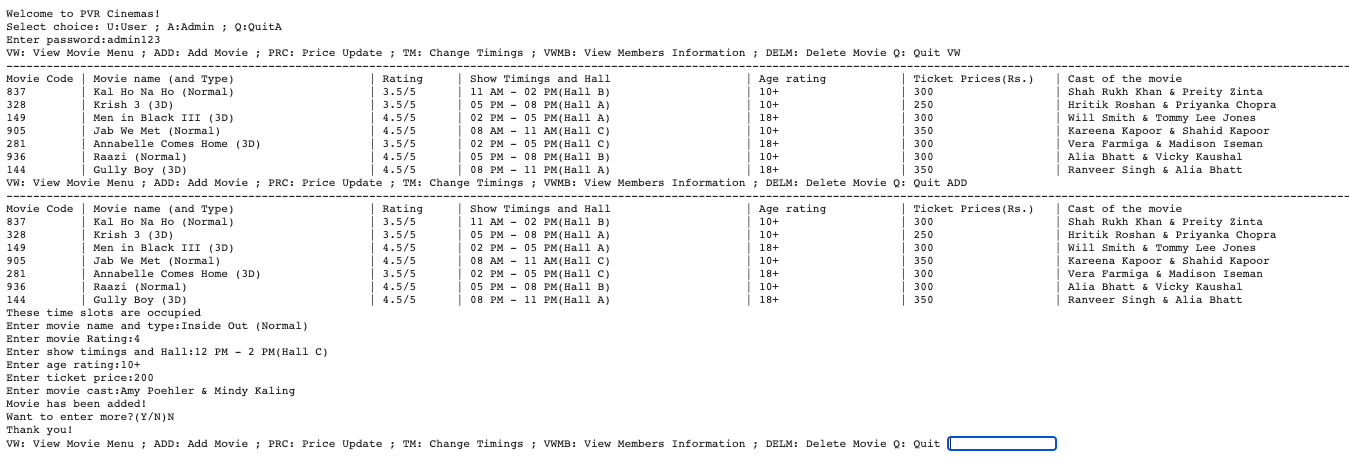
print('Thank you')

break

else:

print('Please enter valid option')

**Output Screen (Admin)**

Display and Add Movies

Price Update

|  |
| --- |

Timings Update

|  |
| --- |

View Member Info

|  |
| --- |

Delete a Movie

|  |
| --- |

**Output Screen (User)**Register Member

|  |
| --- |

View User Info

|  |
| --- |

View Menu and Book Ticket

|  |
| --- |

Update Info

|  |
| --- |

Delete Member and Quit

|  |
| --- |

Hardware & Software Requirement

Hardware Requirement

Laptop and MacBook

Intel core i5

With at least 2 GB RAM

10 MB free space on Hard Disk

LCD/LED

Operating System & Compiler

MS Windows and MacOS

colab.research.google.com