CONTENTS

|  |  |  |
| --- | --- | --- |
| ***s.no*** | ***topic*** | ***Page no*** |
|  | ***Project statement*** |  |
|  | ***Csv, mysql tables*** |  |
|  | ***PYTHON LIBRARIES*** |  |
|  | ***Mainprogram.py*** |  |
|  | ***Login.py*** |  |
|  | ***Project limitations*** |  |
|  | ***bibliography*** |  |
|  | ***conclusion*** |  |
|  |  |  |

MOVIE TICKETING MANAGEMENT

AIM:

The aim of the project is to make customers to book online tickets and snacks for the movies they would like to watch and enjoy in soapanasundari cinemas.

TYPES OF USERS:

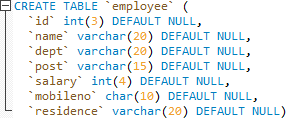
* ADMIN:The admin can view thier soapanasundari cinemas is running well as their wish.they can remove,add employee ,view the statistics of the average seats filled and beverges ordered and also can view the upcoming movie booking status
* MANAGER:The manager should work acoording to the admins wish.they can also add,remove ,alter employee details and they should upload the upcoming movies timing information .they can view the statistics of the average seats filled and beverges ordered and also can view the upcoming movie booking status.

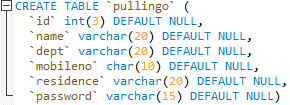
HOW DATAS ARE STORED?

Datas are stored in csv files and mysql tables

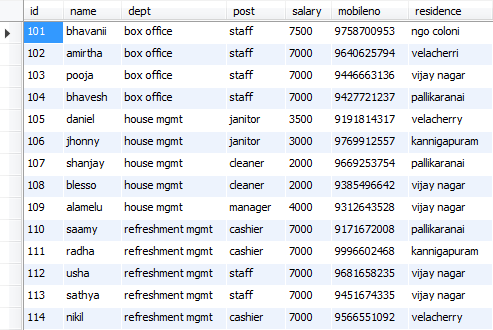
DATA ANALYSIS AND VISUALISATION:

The graph for average seats filled for each movie and the type of beverages which are been ordred in a particular day can be seen.



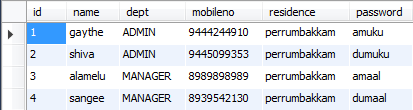


Employee.sql

******

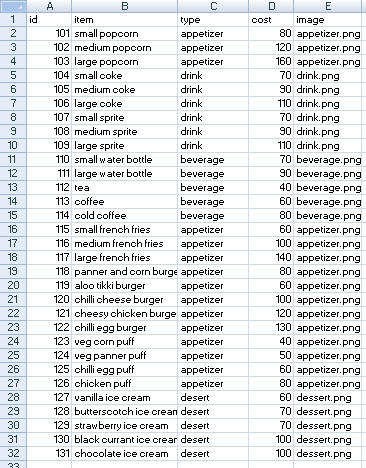
This table contains the employee details of cinemas

Pullingo.sql



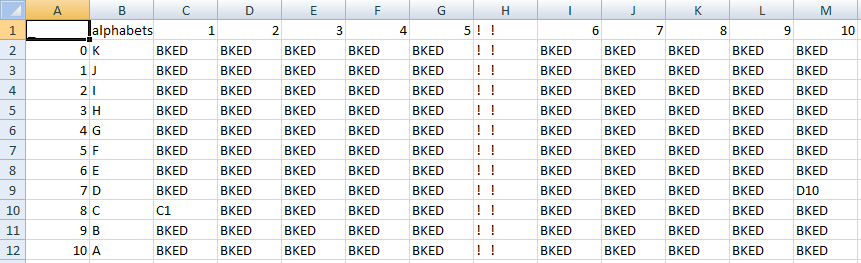
This table contains information of manager and admin

Refreshments.csv



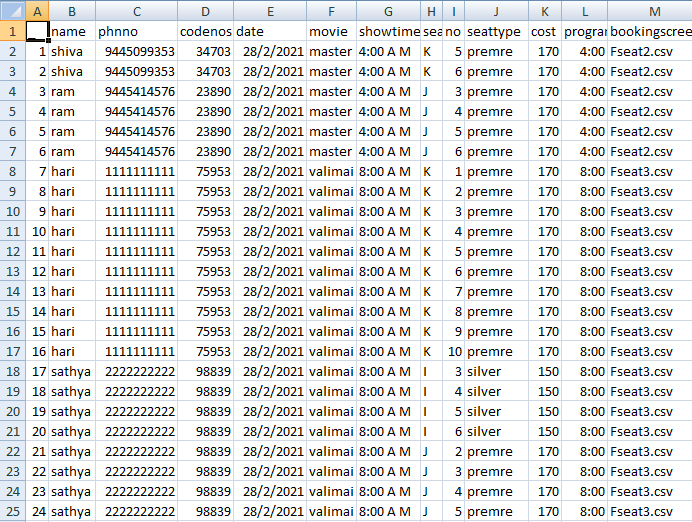
This csv file contains the types of snacks available

Fseat5.csv



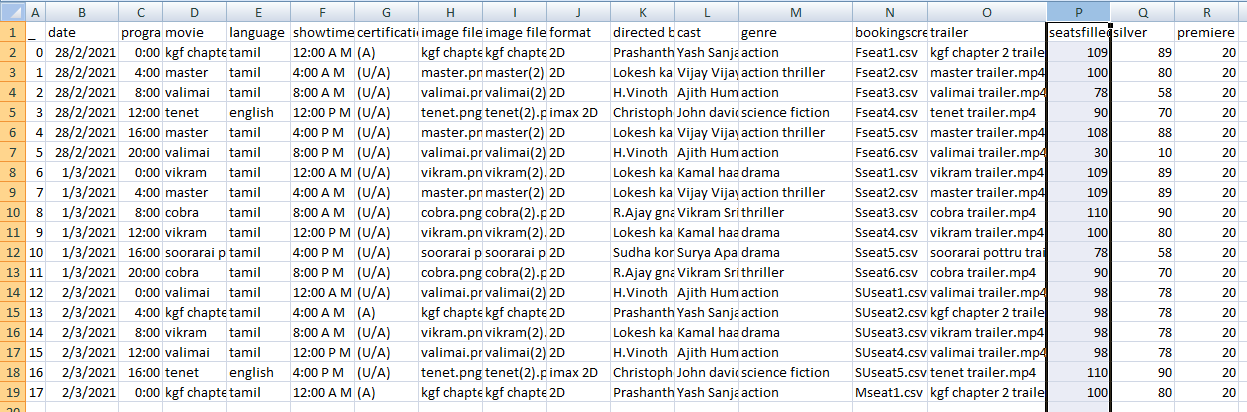
***This csv contains the information of wheather the seat is booked (BKED) or not***

Customer.csv



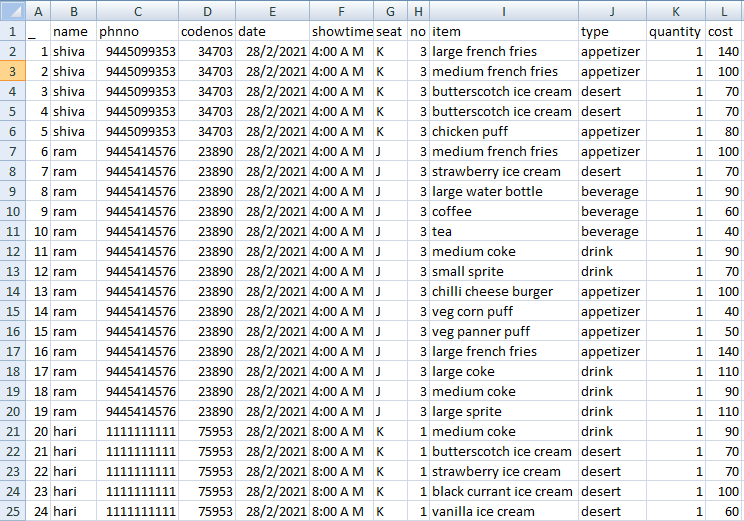
This csv file contains the information of the customer and the movie details of the booked tickets

Movie timing pro.csv



This contains all the movie shows and its details

Customer snacks.csv



***This csv contains details of the snacks, the customers have ordered***

***PYTHON LIBRARIES :***

|  |  |
| --- | --- |
| ***LIBRARY*** | ***DESCRIPTION*** |
| ***PySimpleGUI*** |  |
| ***Pandas*** |  |
| ***Datetime*** |  |
| ***random*** |  |
| ***numpy*** |  |
| ***os.startfile*** |  |
|  |  |

***Mainproject.py***

import PySimpleGUI as sg

import pandas as pd

import datetime

import random as rd

from pygame import mixer

from datetime import datetime

import datetime as dt

import numpy as np

from os import startfile

#background music

mixer.init()

mixer.music.load("Soppana Sundari.mp3")

mixer.music.set\_volume(0.008)

mixer.music.play(-1)

#to create a unique value of dataframe's column which is stored in a list

def unique\_value(dataframe,column):

lt=[]

for var in dataframe[column]:

if var not in lt:

lt=lt+[var]

elif var in lt:

lt=lt

return(lt)

#to create a unique value of list which is stored in a list

def uniquelist(lisa):

lt=[]

for var in lisa:

if var not in lt:

lt=lt+[var]

elif var in lt:

lt=lt

return(lt)

#csv file of customer details

customer=pd.read\_csv("customer.csv")

customer=customer.set\_index("\_")

#csv file of ordered snacks details of customer

customersnacks=pd.read\_csv("customer snacks.csv")

customersnacks=customersnacks.set\_index("\_")

vari=0

while vari<1:

layout0=[[sg.T("SOAPANASUNDARI CINEMAS",size=(27,1),justification='c',font=("gothic",40))],

[sg.Image('opening.png')],

[sg.Button('BOOK TICKETS',size=(28,1)),sg.Button('MY DETAILS',size=(28,1)),

sg.Button('CANCEL TICKETS',size=(29,1))],

[sg.B("THEATRE REVIEWS",size=(90,1))],

[sg.Button('EXIT',size=(90,1))]]

window0=sg.Window('SOAPANASUNDARI CINEMAS',layout0,finalize=True)

eventing,values=window0.read()

window0.close()

onetimeselect=0

multitime=0

variable0=0

if eventing=="THEATRE REVIEWS":

vari=vari-1

startfile("moviereview.mp4")

elif eventing=='BOOK TICKETS':

while variable0<1:

now=dt.datetime.now()

nowstr=now.strftime("%d/%m/%Y %H:%M:%S")

today,time=nowstr.split(' ')

todayday,todaymonth,todayyear=today.split('/')

tomorrow=dt.datetime.now()+dt.timedelta(1)

dayaftertomorrow=dt.datetime.now()+dt.timedelta(2)

tomorrow=tomorrow.strftime("%d/%m/%y")

tomorrowday,tomorrowmonth,tomorrowyear=tomorrow.split('/')

dayaftertomorrow=dayaftertomorrow.strftime("%d/%m/%y")

dayaftertomorrowday,dayaftertomorrowmonth,dayaftertomorrowyear=dayaftertomorrow.split('/')

tomorrow=tomorrowday+'/'+tomorrowmonth+'/'+todayyear[:2]+tomorrowyear

dayaftertomorrow=dayaftertomorrowday+'/'+dayaftertomorrowmonth+'/'+todayyear[:2]+dayaftertomorrowyear

df=pd.read\_csv('movie timing pro.csv')

df=df.set\_index('\_')

for ggwp in df.index:

timeobj=df.loc[ggwp]['date']+(" ")+df.loc[ggwp]['programtime']

timeformat=datetime.strptime(timeobj,"%d/%m/%Y %H:%M")

timeformatstr=timeformat.strftime("%d/%m/%Y %H:%M")

timeformatday,timeformattime=timeformatstr.split(' ')

if timeformat<now :

df=df.drop([ggwp])

elif timeformat > dt.datetime.now()+dt.timedelta(2):

df=df.drop([ggwp])

uniquedate=unique\_value(df,'date')

uniquemoviename=unique\_value(df,'movie')

uniquemovieimage=unique\_value(df,'image file 1')

uniquemovieimage2=unique\_value(df,'image file 2')

lst=[]

onetimeselect=1+onetimeselect

if len(uniquedate)==0:

sg.popup("sorry booking is not available")

vari=vari-1

break

elif onetimeselect==1: #to set default date as today in the layout

lst.append(sg.T(uniquedate[0],size=(40,1),justification='center'))

showdate=uniquedate[0]

showdatemovies=[]

uniqueshowdatemovies=[]

uniqueimages=[]

showdatetime=[]

for kg in (df[df.date==uniquedate[0]]).values:

showdatemovies.append(kg[2])

showdatetime.append(kg[4])

uniqueshowdatemovies=unique\_value(df[df.date==uniquedate[0]],'movie')

uniqueimages=unique\_value(df[df.date==uniquedate[0]],'image file 1')

uniqueimages2=unique\_value(df[df.date==uniquedate[0]],'image file 2')

uniquedate.remove(uniquedate[0])

for wwe in uniquedate:

lst.append(sg.B(wwe,size=(40,1)))

else: #to set user selected date in the layout

for wwe in uniquedate:

if event!=wwe :

lst.append(sg.B(wwe,size=(40,1)))

elif event==wwe or showdate==wwe:

lst.insert(uniquedate.index(wwe),sg.T(wwe,size=(40,1),justification='center'))

showdate=wwe

showdatemovies=[]

uniqueshowdatemovies=[]

uniqueimages=[]

showdatetime=[]

for kg in (df[df.date==wwe]).index:

showdatemovies.append(df[df.date==wwe].loc[kg]["movie"])

showdatetime.append(df[df.date==wwe].loc[kg]["showtime"])

uniqueshowdatemovies=unique\_value(df[df.date==wwe],'movie')

uniqueimages=unique\_value(df[df.date==wwe],'image file 1')

bookingimage=[]

bookingbutton=[]

bookingtext=[]

for dj in range(len(uniqueimages)):

bookingimage.append(sg.Image(filename=uniqueimages[dj]) )

bookingbutton.append(sg.B("BOOK NOW",key=uniqueshowdatemovies[dj],size=(29,1)))

bookingtext.append(sg.T(uniqueshowdatemovies[dj].upper(),

size=(22,1),font=('Times',15),justification='c'))

layout1=[

[sg.T("SOAPANASUNDARI CINEMAS",size=(32,1),justification='c',font=("arial",40))],

lst,

[sg.T(" ")],

[sg.T('--------------------------------------------------------------------------Showing On '+showdate+'-----

------',justification='center',size=(124,1))],

[sg.T(' ')],

bookingtext,

bookingimage,

bookingbutton,

[sg.Button('BACK',size=(124,1))],

[sg.T(' ')]]

window1=sg.Window('Showing on '+showdate,layout1)

event,values=window1.read()

window1.close()

jaks=3

if event=='BACK' or event==sg.WIN\_CLOSED:

jaks=0

vari=vari-1

event=showdate

for horn in uniquedate:

if event==horn and jaks!=0:

jaks=1

variable0=variable0-1

event=horn

for horn in uniqueshowdatemovies:

if event==horn and jaks!=0:

jaks=2

if jaks==2:

sandy=0

while sandy<1:

timebutton=[]

imagebutton=[]

directedby=[sg.T('Directed by:')]

cast=[sg.T('Cast:')]

genre=[sg.T('Genre:')]

for buti in uniqueshowdatemovies:

if event==buti:

imagebutton.append(sg.Image(df.loc[(df.loc[(df[df.movie==buti]).index]['image

file2']).index[0]]['image file 2']))

for kuti in range(len(showdatemovies)):

if showdatemovies[kuti]==event:

timebutton.append(sg.B(showdatetime[kuti],size=(7,1)))

toein=0

for bravo in df.index:

if df.loc[bravo]['movie']==event and toein==0:

toein=1

certification=df.loc[bravo]['certification']

language=df.loc[bravo]['language']

formt=df.loc[bravo]['format']

directedby.append(sg.T(df.loc[bravo]['directed by']))

cast.append(sg.T(df.loc[bravo]['cast']))

genre.append(sg.T(df.loc[bravo]['genre']))

trailer=df.loc[bravo]['trailer']

layout1=[[sg.T('-----------------------------On '+showdate+'------------------',justification='center')],

imagebutton,

[sg.B("WATCH TRAILER")],

[sg.T(event.upper(),size=(17,1),font=("arial",15)),sg.T('

'+certification+”'+language+formt)],

genre,

cast,

directedby,

timebutton,

[sg.B('BACK',size=(35,1))]]

windo1=sg.Window(event,layout1)

even,values=windo1.read()

windo1.close()

if even=='BACK' or even==sg.WIN\_CLOSED:

event=showdate

variable0=variable0-1

elif even=="WATCH TRAILER":

sandy=sandy-1

startfile(trailer)

else:

for loki in df.index:

if df.loc[loki]['movie']==event and df.loc[loki]['date']==showdate

anddf.loc[loki]['showtime']==even :

strfttime=df.loc[loki]['programtime']

if certification=="(A)":

sg.popup(" ADULT(18+) movie")

u=0

while u<1:

a=pd.read\_csv(df.loc[loki]['bookingscreen'])

a=a.set\_index("\_")

seatsfilled=0

seatsavailable=0

for mrf in a.index:

for kcg in range(1,11):

if a.loc[mrf][str(kcg)]=="BKED":

seatsfilled=seatsfilled+1

else:

seatsavailable=seatsavailable+1

noseatslist=[]

if seatsavailable==0:

sg.popup("House Full")

sandy=sandy-1

else:

for pen in range(1,11):

if seatsavailable >= pen:

noseatslist.append(sg.B(str(pen)))

else:

noseatslist.append(sg.T(str(pen),size=(3,1),justification='c'))

layout=[[sg.T("Number of Seats to Book",size=(30,1),font=("gothic",14))],

noseatslist,

[sg.B('back')]]

window=sg.Window("Number

ofSeats",layout,default\_button\_element\_size=(3,1), auto\_size\_buttons=False)

gym,vals = window.read()

window.close()

i=0

if gym==sg.WIN\_CLOSED or gym=='back' :

sandy=sandy-1

else:

for ps in range(1,11):

if gym==str(ps):

noseats=ps

p=0

seatrow=[]

seatno=[]

index=[]

silverseats=[]

premiereseats=[]

while i<=noseats: #for booking the seats in a particular csv file of booking

a=pd.read\_csv(df.loc[loki]['bookingscreen'])

a=a.set\_index("\_")

kkk=[]

jjj=[]

iii=[]

hhh=[]

ggg=[]

fff=[]

eee=[]

ddd=[]

ccc=[]

bbb=[]

aaa=[]

rows=[kkk,jjj,iii,hhh,ggg,fff,eee,ddd,ccc,bbb,aaa]

for var in a.index:

for dor in range(1,11):

if a.loc[var][str(dor)]=="BKED":

rows[var].append(sg.T("BKED"))

elif a.loc[var][str(dor)]==" THIS ":

rows[var].append(sg.T(" THIS "))

else:

rows[int(var)].append(sg.B(a.loc[int(var)][str(dor)]))

for sumo in rows:

sumo.insert(-2,sg.T("| |"))

sumo.insert(2,sg.T("| |"))

sumo.insert(0,sg.T("| |"))

sumo.append(sg.T("| |"))

alphabets=["K","J","I","H","G","F","E","D","C","B","A"]

if i==noseats and i==len(seatno) :

butt=sg.B("BOOK")

else:

butt=sg.T("")

back=sg.B("back")

if i<=noseats:

lay=[[sg.T("SOAPANASUNDARICINEMAS",size=(55,1),

justification='c',font=("arial",15))],

[sg.T("ShowDate:"+showdate,size=(25,1)),sg.T("Movie:"+

event,size=(25,1)),sg.T("ShowTime:"+even,size=(25,1))],

[sg.T("EXT"),sg.T(" "),sg.T("EXT")],

kkk,

jjj,

[sg.T("| | \_\_\_ \_\_\_ \_\_\_ \_ | |\_\_\_ \_\_\_ \_\_\_ \_\_ \_\_ \_\_ \_\_\_

\_\_\_ \_\_ \_\_\_ | |\_\_\_ \_\_\_ \_\_\_ \_\_ | |")],

iii,hhh,ggg,fff,eee,ddd,ccc,bbb,aaa,

[sg.T("| \_\_\_ \_\_\_ \_\_\_ \_\_\_ | |\_\_\_ \_\_\_ \_\_\_ \_\_ \_\_ \_\_ \_\_\_ \_\_\_

\_\_\_ \_\_\_ \_\_\_ | | \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_|")],

[sg.T("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")],

[sg.T(" A L L E Y E S T H I S W A Y P L

E A S E ")],

[sg.T("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_") ],

[butt,back,sg.T("BKED - already booked"),

sg.T(" THIS - seat selected by you")],

[sg.T(" "),sg.T("SILVER(A-I): Rs

150",font=("arial",10)),sg.T(" "),sg.T("PREMIERE(J,K): Rs

170",font=("arial",10)),sg.T(" seats left for selection

:"+str(noseats-i)+" ")]]

window=sg.Window("bookingscreen",lay,

default\_button\_element\_size=(4,1), auto\_size\_buttons=False)

bucket,vals = window.read()

window.close()

if i<noseats :

for dum in a.index:

for col in range(1,11):

if bucket== a.loc[dum][str(col)]:

for gg in range(len(a.loc[dum][str(col)])):

if gg==0:

seatrow.append((a.loc[dum][str(col)])[0])

seatno.append(str(col))

a.loc[dum][str(col)]=" THIS "

index.append(dum)

a.to\_csv(df.loc[loki]['bookingscreen'])

i=i+1

if bucket=="back" and len(seatno)!=0:

i=i-1

a.loc[index[-1]][seatno[-1]]=seatrow[-1]+seatno[-1]

del(seatrow[-1])

del(seatno[-1])

del(index[-1])

a.to\_csv(df.loc[loki]['bookingscreen'])

elif bucket==sg.WIN\_CLOSED :

if len(seatno)!=0:

i=0

for k in range(len(seatno)):

a.loc[index[-1]][seatno[-1]]=seatrow[-1]+seatno[-1]

del(seatrow[-1])

del(seatno[-1])

del(index[-1])

else:

i=1+noseats

u=u-1

a.to\_csv(df.loc[loki]['bookingscreen'])

silverseats=[]

premiereseats=[]

elif bucket=="back" and len(seatno)==0:

u=u-1

i=noseats+1

if bucket=="BOOK":

now1=dt.datetime.now()

now2=now1+dt.timedelta(minutes=4)

i=noseats+1

for hi in range(len(seatrow)):

if seatrow[hi]=="J" or seatrow[hi]=="K":

premiereseats.append((seatrow[hi],seatno[hi]))

else:

silverseats.append((seatrow[hi],seatno[hi]))

orderedsnacksname=[]

orderedsnacksprice=[]

sunny=0

bill=0

while sunny<1: # refreshments

refreshments=pd.read\_csv("movie refreshments.csv")

refreshments=refreshments.set\_index("id")

uniquesnackstype=unique\_value(refreshments,"type")

uniquesnacksimage=unique\_value(refreshments,"image")

textsnackstype=[]

imagesnacksimage=[]

snacksbutton=[]

for tip in range(len(uniquesnackstype)):

textsnackstype.append(sg.T(uniquesnackstype[tip].upper(),

size=(27,1),font=("arial",13),justification="c"))

imagesnacksimage.append(sg.Image(uniquesnacksimage[tip]))

snacksbutton.append(sg.B("ORDER

NOW",key=uniquesnackstype[tip]))

snackswindow=[[sg.T("R E F R E S H M E N T

S",size=(31,1),font=("Times",40),justification='c')],

textsnackstype,

imagesnacksimage,

snacksbutton,

[sg.B("PREVIOUS",size=(60,1)),sg.B("NEXT",size=(60,1))] ]

dora=sg.Window("",snackswindow,default\_button\_element\_size

=(29,1), auto\_size\_buttons=False)

item,vals=dora.read()

dora.close()

blesso=0

while blesso<1:

snacksname=[]

snacksprice=[]

chatbutton=[]

refreshmentsname=[]

refreshmentsprice=[]

refreshmentlisting=[]

for yoyo in uniquesnackstype:

if item==yoyo:

refreshments=refreshments[refreshments.type==item]

refreshmentlisting.append([sg.T(yoyo.upper()

,size=(30,1),font=("arial",20),justification='c')])

refreshmentlisting.append([sg.T("Item

Name",size=(8,1),font=("Times",17)),sg.T("cost",

size=(8,1),font=("Times",17),justification='c'),sg.T("

click to

add",size=(15,1),font=("Times",17),justification='c')])

for jog in refreshments.index:

snacksname.append(sg.T(refreshments.

loc[jog]["item"],size=(15,1)))

snacksprice.append(sg.T("Rs:"+str(refreshments.

loc[jog]["cost"]),size=(15,1)))

chatbutton.append(sg.B(refreshments.loc[jog]["item"]))

refreshmentsname.append(refreshments.loc[jog]["item"])

refreshmentsprice.append(refreshments.loc[jog]["cost"])

refreshmentlisting.append([sg.T(refreshments.

loc[jog]["item"],size=(20,1)),

sg.T("Rs:"+str(refreshments.loc[jog]["cost"]),

size=(20,1)),sg.B("ADD",key=refreshments.loc[jog]["item"])])

refreshmentlisting.append([sg.B("BACK")])

typ=refreshmentlisting

typewind=sg.Window("",typ,default\_

button\_element\_size=(14,1), auto\_size\_buttons=False)

mad,vals=typewind.read()

typewind.close()

if mad=="BACK" or mad==sg.WIN\_CLOSED:

sunny=sunny-1

for thor in range(len(refreshmentsname)):

if mad==refreshmentsname[thor]:

sg.popup(mad,"+1 Item Added")

orderedsnacksname.append(mad)

orderedsnacksprice.append(int(refreshmentsprice[thor]))

blesso=blesso-1

blesso=blesso+1

if item=="NEXT":

rice=0

bill=0

while rice<1:

orderedsnacksquantity=[]

uniqueorderedsnacksname=uniquelist(orderedsnacksname)

uniqueorderedsnacksprice=uniquelist(orderedsnacksprice)

for wiz in uniqueorderedsnacksname:

quan=0

for electro in orderedsnacksname:

if wiz==electro:

quan=quan+1

orderedsnacksquantity.append(quan)

finalrefreshmentlisting=[]

if len(orderedsnacksname)!=0:

finalrefreshmentlisting.append([sg.T("---------------------------------

----------")])

finalrefreshmentlisting.append([sg.T("SNACKS

BILL",size=(30,1),font=("times",20),justification='c')])

finalrefreshmentlisting.append([sg.T("---------------------------------

-----------")])

finalrefreshmentlisting.append([sg.T("Items cost quantity

click to delete(-

1)",size=(45,1),font=("arial",15),justification='c')])

for kar in range(len(uniqueorderedsnacksname)):

finalrefreshmentlisting.append(([sg.T(uniqueordered

snacksname[kar],size=(15,1)),

sg.T(orderedsnacksprice[kar],size=(15,1)),

sg.T(orderedsnacksquantity[kar],

size=(5,1)),sg.B("REMOVE",

key=uniqueorderedsnacksname[kar])]))

finalrefreshmentlisting.append([sg.T("---------------------------------

-----------")])

finalrefreshmentlisting.append([sg.T("Totalcost",size=(15,1))

,sg.T("Rs:"+str(sum(orderedsnacksprice)),size=(15,1)),

sg.T("Totalquantity="+str(len(

orderedsnacksprice)),size=(15,1),justification='l')])

finalrefreshmentlisting.append([sg.B("Back"),sg.B("Proceed")

,sg.B("Delete All")])

else:

finalrefreshmentlisting.append([sg.B("Proceed",

size=(20,1)),sg.B("Back",size=(20,1))])

lake=finalrefreshmentlisting

corn=sg.Window("RefreshmentBill",lake,

default\_button\_element\_size=(14,1), auto\_size\_buttons=False)

bad,vals=corn.read()

corn.close()

if (bad=="Back" or bad==sg.WIN\_CLOSED) :

rice=rice

sunny=sunny-1

elif bad!="Proceed" and bad!="Delete All":

bat=0

iq=-1

for god in range(len(orderedsnacksname)):

if orderedsnacksname[god]==bad :

bat=bat+1

for monk in range(len(uniqueorderedsnacksname)):

if bad==uniqueorderedsnacksname[monk]:

iq=orderedsnacksquantity[monk]

ram=monk

if bat==iq:

sai=god

finish=1

if finish==1:

del(orderedsnacksprice[sai])

del(orderedsnacksname[sai])

del(orderedsnacksquantity[ram])

rice=rice-1

if bad=="Proceed":

print("")

elif bad=="Delete All":

orderedsnacksname=[]

orderedsnacksprice=[]

orderedsnacksquantity=[]

sunny=sunny-1

rice=rice+1

if item=="PREVIOUS" or item==sg.WIN\_CLOSED:

now3=dt.datetime.now()

bill=1

if now3>now2:

orderedsnacksname=[]

orderedsnacksprice=[]

orderedsnacksquantity=[]

sandy=sandy-1

i=noseats+1

zing=index

for gaythe in range(len(zing)):

a.loc[zing[gaythe]][seatno[gaythe]]=

seatrow[gaythe]+seatno[gaythe]

a.to\_csv(df.loc[loki]['bookingscreen'])

else:

print(now3,now2,now1)

i=noseats

silverseats=[]

premiereseats=[]

sunny=sunny+1

#total bill

if bill==0 :

ticketcost=len(silverseats)\*150 + len(premiereseats)\*170

onlinecharges=(len(silverseats) + len(premiereseats))\*40

totalcost=ticketcost+onlinecharges

snackscost=sum(orderedsnacksprice)

finalcost=snackscost+totalcost

bookingconfirm=[[sg.T("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")],

[sg.T("BILL",size=(15,1),font=("Times",20),justification='c')],

[sg.T("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")],

[sg.T("Would you like to proceed further ? ")],

[sg.T("Ticketcost Rs:"+str(ticketcost))],

[sg.T("Onlinecharges Rs:"+str(onlinecharges))],

[sg.T("Totalcost Rs:"+str(totalcost))],

[sg.T("Snackscost Rs:"+str(snackscost))],

[sg.T("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")],

[sg.T("Finalcost Rs:"+str(finalcost))],

[sg.T("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")],

[sg.B("No"),sg.T(" "),sg.B("Yes")]]

work=sg.Window("Confirmation",bookingconfirm

,default\_button\_element\_size=(3,1), auto\_size\_buttons=False)

confirm,values=work.read()

work.close()

if confirm=="Yes":

for gaythe in a.index:

for jaga in range(1,11):

if a.loc[gaythe][str(jaga)]==" THIS ":

a.loc[gaythe][str(jaga)]="BKED"

sg.popup("Payment Successful")

g=0

#for creating a 5 digit unique code number for the user

while g<1:

codeno=rd.randint(10000,99999)

for xx in customer['codenos']:

if xx==codeno:

g=-1

g=g+1

codeno=str(codeno)

ww=0

while ww<1:

num=[[sg.Text("Enter your Name")],[sg.Input()],

[sg.T("Enter your Phone Number")],[sg.Input()],

[sg.Button("Ok")]]

tata=sg.Window("Enter Details",num)

ting,vals=tata.read()

tata.close()

number=(vals[1])

customername=vals[0]

number=str(number)

if len(number)!=10 or ting==sg.WIN\_CLOSED :

sg.popup("Enter Valid Number")

ww=ww-1

ww=ww+1

for you in range(len(silverseats)):

if len(customer)!=0:

customer.loc[max(customer.index)+1]=customername,

number,codeno,showdate,event,even,silverseats[you][0],

silverseats[you][1],"silver",str(150)

,strfttime,df.loc[loki]['bookingscreen']

else:

customer.loc[1]=customername,number,codeno,showdate,

event,even,silverseats[you][0], silverseats[you][1],"silver"

,str(150),strfttime,df.loc[loki]['bookingscreen']

for you in range(len(premiereseats)):

if len(customer)!=0:

customer.loc[max(customer.index)+1]=customername

,number,codeno,showdate,event,even,

premiereseats[you][0],premiereseats[you][1],"premre",str(170),strfttime,df.loc[loki]['bookingscreen']

else:

customer.loc[1]=customername,number,codeno,

showdate,event,even,premiereseats[you][0],

premiereseats[you][1],"premre",str(170),

strfttime,df.loc[loki]['bookingscreen']

for you in range(len(orderedsnacksname)):

kasx=refreshments[refreshments.item==

orderedsnacksname[you]]

if len(customersnacks)!=0:

customersnacks.loc[max(customersnacks.index)+1]=

customername ,number,codeno,showdate,even,seatrow[0]

,seatno[0],orderedsnacksname[you],kasx.loc[max(kasx.index)]

["type"],str(1),orderedsnacksprice[you]

else:

customersnacks.loc[1]=customername,number,codeno

,showdate,even,seatrow[0],

seatno[0],orderedsnacksname[you],

kasx.loc[max(kasx.index)]["type"]

,str(1),orderedsnacksprice[you]

df=pd.read\_csv("movie timing pro.csv")

df=df.set\_index("\_")

df.loc[loki,"seatsfilled"]=seatsfilled+len(seatrow)

df.loc[loki,"silver"]=df.loc[loki,"silver"]+len(silverseats)

df.loc[loki,"premiere"]=df.loc[loki,"premiere"]+len(premiereseats)

df.to\_csv("movie timing pro.csv")

customersnacks.to\_csv("customer snacks.csv")

customer.to\_csv("customer.csv")

a.to\_csv(df.loc[loki]['bookingscreen'])

sg.popup("your code number is:",codeno)

sg.popup("Thank you for booking.",

"view your details in MY Details")

vari=vari-1

else:

sandy=sandy-1

zing=index

for gaythe in range(len(zing)):

a.loc[zing[gaythe]][seatno[gaythe]]=seatrow[gaythe]

+seatno[gaythe]

a.to\_csv(df.loc[loki]['bookingscreen'])

u=u+1

sandy=sandy+1

variable0=variable0+1

elif eventing=='EXIT' or eventing==sg.WIN\_CLOSED:

sg.popup("Thank you for visiting","Have a Good Day!")

else:

ww=0

while ww<1:

num=[[sg.Text("enter your name")],[sg.Input()],

[sg.T("enter your phone number")],[sg.Input()],

[sg.T("enter your codeno")],[sg.Input()],

[sg.B("submit"),sg.B("back")]]

tata=sg.Window("number",num)

ting,vals=tata.read()

tata.close()

looty=0

if ting=="submit":

number=(vals[1])

customername=vals[0]

codeno=vals[2]

codeno=str(codeno)

number=str(number)

if len(number)!=10 :

sg.popup("enter valid number")

ww=ww-1

else:

pain=0

messi=0

while pain<1:

ktm=0

for guc in customer.index:

if messi==0:

if customer.loc[guc]["name"]==customername and

str(customer.loc[guc]["phnno"])==str(number) and

str(customer.loc[guc]["codenos"])==str(codeno) and ktm==0:

looty=1

ktm=1

customerdetails=customer[customer.codenos==str(codeno)]

if len(customerdetails)==0:

customerdetails=customer[customer.codenos==int(codeno)]

bookingscreenloc=unique\_value(customerdetails,"bookingscreen")

customerdetails=customerdetails.drop(["name","phnno","codenos"

,"programtime","bookingscreen"],axis=1)

customersnacksdetails=customersnacks[customersnacks

.codenos==str(codeno)]

if len(customersnacksdetails)==0:

customersnacksdetails=customersnacks[customersnacks

.codenos==int(codeno)]

customersnacksdetails=customersnacksdetails.drop(["name"

,"phnno","codenos","seat","no","date","showtime"],axis=1)

b=pd.read\_csv(bookingscreenloc[0])

b=b.set\_index("alphabets")

totaltickets=len(customerdetails)

if len(customersnacksdetails)==0:

sg.popup("name:"+customername+" "+"phnno:"+number+"

"+"codenos:"+str(codeno),customerdetails,

"totaltickets:"+str(totaltickets),title="details")

else:

sg.popup("name:"+customername+" "+"phnno:"+number+"

"+"codenos:"+str(codeno),customerdetails,"totaltickets:"+str(totaltickets),"SNACKS DETAILS ",customersnacksdetails,"quantity:"+str(len(customersnacksdetails)),title="details")

if eventing!='MY DETAILS':

customerdetails=customer[customer.codenos==str(codeno)]

if len(customerdetails)==0:

customerdetails=customer[customer.codenos==int(codeno)]

customerdetails=customerdetails.drop(["name","phnno"

,"codenos","bookingscreen"],axis=1)

now=dt.datetime.now()

for t in customerdetails.index:

print("")

timeobj=customerdetails.loc[t]['date']+("

")+customerdetails.loc[t]['programtime']

timeformat=datetime.strptime(timeobj,"%d/%m/%Y %H:%M")-

dt.timedelta(minutes=30)

if timeformat<now:

sg.popup("you cant cancel since time is up")

else:

customerdetailslist=[[sg.T("tick to cancel")],[sg.T(" ",size=(10,1))

,sg.T("seat alp",size=(10,1)),sg.T("seat no",size=(10,1))]]

cusindex=list(customerdetails.index)

cusseatalp=list(customerdetails.seat)

cusseatno=list(customerdetails.no)

cusseatalpno=[]

indexdel=[]

for pro in range(len(cusindex)):

customerdetailslist.append([sg.T(str((pro+1)),size=(10,1)),

sg.T(cusseatalp[pro],size=(10,1)),

sg.T(str(cusseatno[pro]),size=(10,1)),sg.Checkbox("remove",

key=cusseatalp[pro]+str(cusseatno[pro]),size=(10,1))])

cusseatalpno.append(cusseatalp[pro]+str(cusseatno[pro]))

customerdetailslist.append([sg.B("OK"),sg.B("Back")])

icewiz=customerdetailslist

punk=sg.Window("cancellation",icewiz)

dumal,vals=punk.read()

punk.close()

if dumal==sg.WIN\_CLOSED or dumal=="Back":

ww=ww-1

elif dumal=="OK":

cashreturn=0

df=pd.read\_csv('movie timing pro.csv')

df=df.set\_index('\_')

for maxi in range(len(cusseatalpno)):

if vals[cusseatalpno[maxi]]==True:

for kgd in df.index:

if df.loc[kgd]["movie"]==customerdetails.loc[max(

customerdetails.index)]["movie"] and

df.loc[kgd]["showtime"]==customerdetails.loc[max(customerdetails.index)]["showtime"] and df.loc[kgd]["date"]==customerdetails.loc[max(customerdetails.index)]["date"] :

df.loc[kgd,"seatsfilled"]=df.loc[kgd,"seatsfilled"] - 1

if cusseatalpno[maxi][0] in ("K","J"):

df.loc[kgd,"premiere"]=df.loc[kgd,"premiere"] - 1

else:

df.loc[kgd,"silver"]=df.loc[kgd,"silver"] - 1

df.to\_csv('movie timing pro.csv')

cashreturn=cashreturn+int(customerdetails

.loc[cusindex[maxi]]["cost"])

customerdetails=customerdetails.drop(cusindex[maxi])

customer=customer.drop(cusindex[maxi])

b.loc[cusseatalp[maxi],str(cusseatno[maxi])]=cusseatalpno[maxi]

b.to\_csv(bookingscreenloc[0])

customer.to\_csv("customer.csv")

customerdetails=customer[customer.codenos==int(codeno)]

if len(customerdetails)==0:

customerdetails=customer[customer.codenos==str(codeno)]

sg.popup("scucessfully cancelled and cash

Rs:"+str(cashreturn)+"retured")

messi=1

vari=vari-1

else:

vari=vari-1

pain=pain+1

if looty==0:

sg.popup("your mobileno or codeno or name is invalid")

ww=ww-1

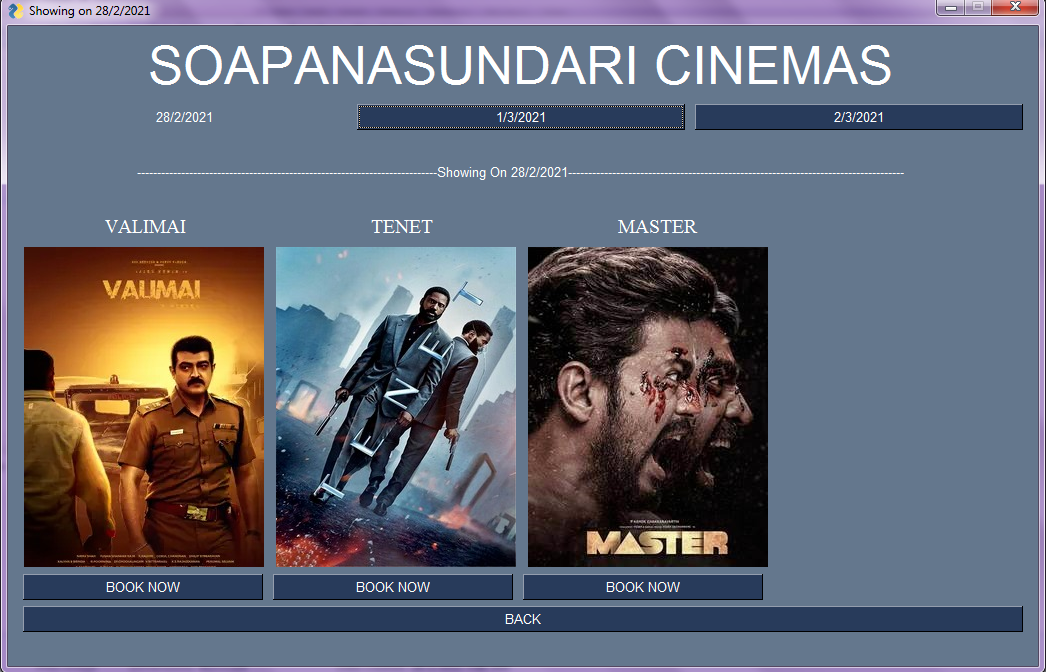
elif ting!="submit" :

vari=vari-1

ww=ww+1

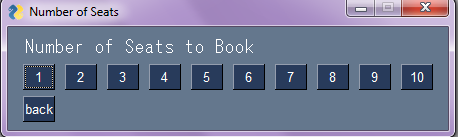
vari=vari+1



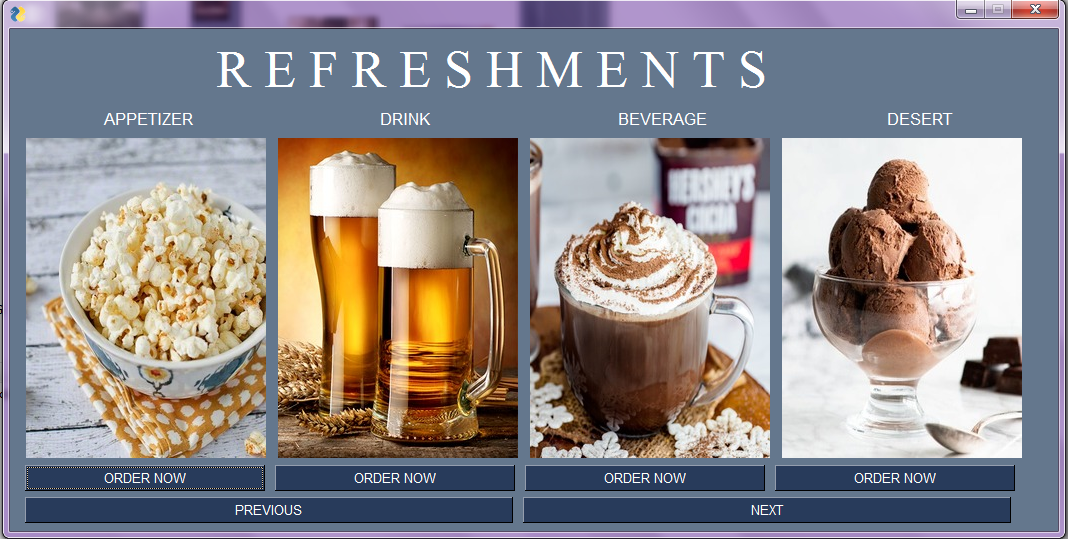




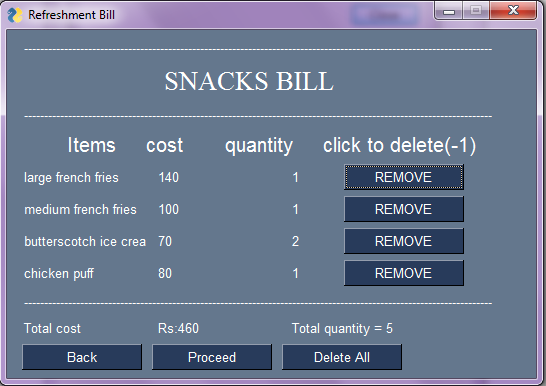


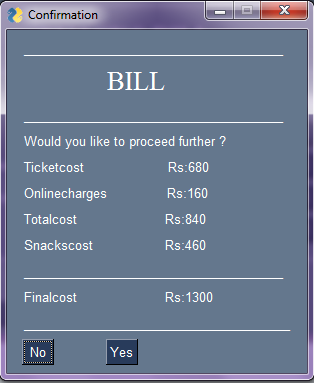


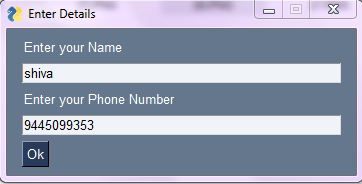


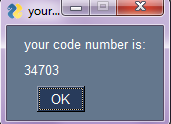




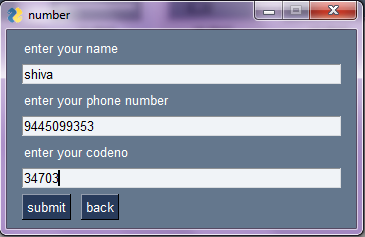


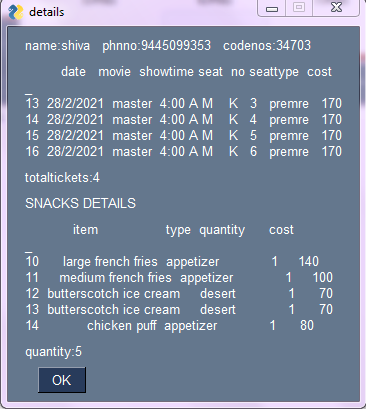




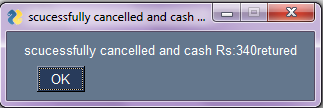












login.py

import PySimpleGUI as sg

import pandas as pd

import mysql.connector as mc

import datetime as dt

from datetime import datetime

import matplotlib.pyplot as plt

def unique\_value(dataframe,column):

lt=[]

for var in dataframe[column]:

if var not in lt:

lt=lt+[var]

elif var in lt:

lt=lt

return(lt)

def dataframe(instruction):

opodu=""

mydb=mc.connect(

host='localhost',

database='test',

user='root',

password='#Mysql123'

)

cursor=mydb.cursor()

cursor.execute(instruction)

mydb.commit()

return ""

def dataframefetchall(instruction):

mydb=mc.connect(

host='localhost',

database='test',

user='root',

password='#Mysql123'

)

cursor=mydb.cursor()

cursor.execute(instruction)

record=cursor.fetchall()

empdf=pd.DataFrame(record,columns=cursor.column\_names)

return empdf

now=dt.datetime.now()

login=dataframefetchall("select \* from pullingo")

login=login.set\_index("id")

movieinfo=pd.read\_csv("movie timing pro.csv")

movieinfo=movieinfo.set\_index("\_")

print(movieinfo)

df=movieinfo

future=movieinfo

dayaftertomorrow=dt.datetime.now()+dt.timedelta(2)

print(movieinfo)

for ggwp in df.index:

timeobj=df.loc[ggwp]['date']+(" ")+df.loc[ggwp]['programtime']

timeformat=datetime.strptime(timeobj,"%d/%m/%Y %H:%M")

timeformatobj=df.loc[ggwp]['date']

timeformatfut=datetime.strptime(timeformatobj,"%d/%m/%Y")

timeformatstr=timeformat.strftime("%d/%m/%Y %H:%M")

timeformatday,timeformattime=timeformatstr.split(' ')

if timeformat<now :

df=df.drop([ggwp])

if timeformatfut < dayaftertomorrow :

future=future.drop([ggwp])

print(",",movieinfo)

originalscreen=["Mseat1.csv","Mseat2.csv","Mseat3.csv","Mseat4.csv","Mseat5.csv","Mseat6.csv","Tseat1.csv","Tseat2.csv","Tseat3.csv","Tseat4.csv","Tseat5.csv","Tseat6.csv","Wseat1.csv","Wseat2.csv","Wseat3.csv","Wseat4.csv","Wseat5.csv","Wseat6.csv","THseat1.csv","THseat2.csv","THseat3.csv","THseat4.csv","THseat5.csv","THseat6.csv",'Fseat1.csv', 'Fseat2.csv', 'Fseat3.csv','Fseat4.csv', 'Fseat5.csv', 'Fseat6.csv', 'Sseat1.csv', 'Sseat2.csv', 'Sseat3.csv', 'Sseat4.csv', 'Sseat5.csv', 'Sseat6.csv', 'SUseat1.csv', 'SUseat2.csv', 'SUseat3.csv', 'SUseat4.csv', 'SUseat5.csv', 'SUseat6.csv']

customer=pd.read\_csv("customer.csv")

customer=customer.set\_index("\_")

customersnacks=pd.read\_csv("customer snacks.csv")

customersnacks=customersnacks.set\_index("\_")

c=0

while c<1:

layout=[[sg.Button("admin",size=(25,1))],

[sg.Button("manager",size=(25,1))],

[sg.Button("close",size=(25,1))]]

window=sg.Window("LOGIN AS",layout)

event,vals = window.read()

window.close()

d=0

while d<1:

if event=="admin" or event=="manager":

layout=[[sg.Text("name")],[sg.Input()],

[sg.Text("password")],[sg.Input()],

[sg.Button("submit")],[sg.Button("back")]]

window=sg.Window("details",layout)

eve,vals = window.read()

window.close()

name=vals[0]

password=vals[1]

if eve=="back" or eve==sg.WIN\_CLOSED :

c=c-1

elif eve=="submit":

noo=1

for i in login.index:

e=0

while e<1:

if str(login.loc[i]["name"])==str(name) and str(login.loc[i]["password"])

==str(password) and login.loc[i]["dept"].lower()==event:

noo=0

if event=="ADMIN(owner)":

layout=[[sg.Button("re-set MYdetails",size=(25,1))],

[sg.Button("view bookingscreen",size=(25,1))],

[sg.Button("movieshow INFO",size=(25,1))],

[sg.B("SHOW HISTORY",size=(25,1))],

[sg.B("GRAPHS",size=(25,1))],

[sg.Button("back"),sg.Button("log out")]]

window=sg.Window(event,layout)

eventing,vals = window.read()

window.close()

else:

layout=[[sg.Button("re-set employee details",size=(25,1))],

[sg.Button("re-set MYdetails",size=(25,1))],

[sg.Button("view bookingscreen",size=(25,1))],

[sg.Button("movieshow INFO",size=(25,1))],

[sg.B("SHOW HISTORY",size=(25,1))],

[sg.B("Update shows",size=(25,1))],

[sg.B("GRAPHS",size=(25,1))],

[sg.Button("back"),sg.Button("log out")]]

window=sg.Window(event,layout)

eventing,vals = window.read()

window.close()

if eventing=="log out":

c=c-1

elif eventing=="back" or eventing==sg.WIN\_CLOSED:

d=d-1

elif eventing=="re-set employee details":

employee=dataframefetchall("select \* from employee")

employee=employee.set\_index("id")

h=0

while h<1:

layout=[[sg.T("whats your aim")],[sg.B("back"),sg.B("Update"),sg.B("Add")]]

window=sg.Window("details",layout)

pat,cummins=window.read()

window.close()

if pat=="Update" :

rider=0

indexid=list(employee.index)

while rider <len(indexid) :

robi=indexid[rider]

layout=[[sg.T("current details")],

[sg.T("name: "),sg.InputText(employee.loc[robi]["name"])],

[sg.T("place: "),sg.InputText(employee.loc[robi]["residence"])],

[sg.T("number; "),sg.InputText(employee.loc[robi]["mobileno"])],

[sg.T("salary; "),sg.InputText(employee.loc[robi]["salary"])],

[sg.T("dept; "),sg.InputText(employee.loc[robi]["dept"])],

[sg.T("post; "),sg.InputText(employee.loc[robi]["post"])],

[sg.B("update"),sg.B("Delete"),sg.B("Previous"),

sg.B("Next"),sg.B("back")]]

window=sg.Window("details",layout)

ian,vali=window.read()

window.close()

if ian=="update" :

jack=0

if len(vali[0])>4:

jack=jack+1

else:

sg.popup("name char should be greater than 4")

if len(vali[1])>3:

jack=jack+1

else:

sg.popup("place char should be greater than 3")

if len(vali[2])==10:

numberall=list(employee.mobileno)

numberall.remove(employee.loc[robi]["mobileno"])

booyah=0

for kolu in numberall:

if str(kolu)==str(vali[3]):

booyah=booyah+1

if booyah==0:

jack=jack+1

else:

sg.popup("entered number already exist")

else:

sg.popup("number digit should be greater than 10")

if len(vali[3])>=4:

jack=jack+1

else:

sg.popup("salaray is too low")

if len(vali[4])>2:

jack=jack+1

else:

sg.popup("enter correct department")

if len(vali[5])>3:

jack=jack+1

else:

sg.popup("enter correct post")

if jack==6:

sg.popup("updated")

employee=dataframe('update employee set

name='+'"'+vali[0]+'"'+','+'dept='+'"'+vali[4]+'"'+','+'post='+'"'

+vali[5]+'"'+','+'salary='+vali[3]+','+'mobileno='+'"'+vali[2]+'"'

+','+'residence ='+'"'+vali[1]+'"'+' where id='+str(robi))

employee=dataframefetchall("select \* from employee")

employee=employee.set\_index("id")

rider=0

indexid=list(employee.index)

elif ian=="Delete":

layout=[[sg.T("Are u sure you want to delete")],

[sg.B("Yes"),sg.B("No")]]

window=sg.Window("details",layout)

kaka,po=window.read()

window.close()

if kaka=="Yes":

dele=dataframe("delete from employee where id="+str(robi))

employee=dataframefetchall("select \* from employee")

employee=employee.set\_index("id")

indexid=list(employee.index)

elif ian=="Next":

if rider==len(indexid)-1:

rider=0

else:

rider=rider+1

elif ian=="Previous":

if rider==0:

rider=len(indexid)-1

else:

rider=rider-1

else:

break

elif pat== "Add":

employee=dataframefetchall("select \* from employee")

employee=employee.set\_index("id")

tring=0

while tring<1:

layout=[[sg.T("current details")],

[sg.T("name: "),sg.Input()],

[sg.T("place: "),sg.Input()],

[sg.T("number; "),sg.Input()],

[sg.T("salary; "),sg.Input()],

[sg.T("dept; "),sg.Input()],

[sg.T("post; "),sg.Input()],

[sg.B("update"),sg.B("back")]]

window=sg.Window("details",layout)

ian,vali=window.read()

window.close()

if ian=="update" :

jack=0

if len(vali[0])>4:

jack=jack+1

else:

sg.popup("name char should be greater than 4")

if len(vali[1])>3:

jack=jack+1

else:

sg.popup("place char should be greater than 3")

if len(vali[2])==10:

jack=jack+1

else:

sg.popup("number digit should be greater than 10")

if len(vali[3])>=4:

jack=jack+1

else:

sg.popup("salaray is too low")

if len(vali[4])>2:

jack=jack+1

else:

sg.popup("enter correct department")

if len(vali[5])>3:

jack=jack+1

else:

sg.popup("enter correct post")

if jack==6:

employee=dataframe("insert into employee values"+str(tuple((

max(employee.index)+1,vali[0],vali[4],vali[5],int(vali[3]),vali[2],vali[4]))))

employee=dataframefetchall("select \* from employee")

employee=employee.set\_index("id")

sg.popup("Added")

else:

tring=tring-1

else:

h=h-1

tring=tring+1

h=h+1

e=e-1

elif eventing=="re-set MYdetails":

g=0

while g<1:

layout=[[sg.T("current details")],[sg.T("name:"),sg.InputText(login.loc[i]

["name"])],[sg.T("password:"),sg.InputText(login.loc[i]["password"])],

[sg.T("place: "),sg.InputText(login.loc[i]["residence"])],[sg.T("number;

"),sg.InputText(login.loc[i]["mobileno"])],

[sg.B("update"),sg.B("back")]]

window=sg.Window("details",layout)

sacrifice,vali= window.read()

window.close()

if sacrifice=="update":

jack=0

if len(vali[0])>4:

jack=jack+1

else:

sg.popup("name char should be greater than 4")

if len(vali[1])>5:

passwordall=list(login.password)

passwordall.remove(login.loc[i]["password"])

booyah=0

for kolu in passwordall:

if kolu==vali[1]:

booyah=booyah+1

if booyah==0:

jack=jack+1

else:

sg.popup("please change your password")

else:

sg.popup("password char should be greater than 4")

if len(vali[2])>3:

jack=jack+1

else:

sg.popup("place char should be greater than 3")

if len(vali[3])==10:

numberall=list(login.mobileno)

numberall.remove(login.loc[i]["mobileno"])

booyah=0

for kolu in numberall:

if str(kolu)==str(vali[3]):

booyah=booyah+1

if booyah==0:

jack=jack+1

else:

sg.popup("entered number already exist")

else:

sg.popup("number digit should be greater than 10")

if jack==4:

login=dataframe('update pullingo set

name='+'"'+vali[0]+'"'+','+'mobileno='+'"'+vali[3]

+'"'+','+'residence='+'"'+vali[2]+'"'+','

+'password='+'"'+vali[1]+'"'+' where id='+str(i))

login=dataframefetchall("select \* from pullingo")

login=login.set\_index("id")

e=e-1

sg.popup("updated")

name=vali[0]

password=vali[1]

else:

g=g-1

else:

g=g

e=e-1

g=g+1

elif eventing=="GRAPHS":

yendi=0

while yendi <1:

layout=[[sg.Button("MOVIES vs SEATS",size=(25,1))],

[sg.Button("refreshments graph",size=(25,1))],

[sg.Button("back")]]

window=sg.Window(event,layout)

eting,vals = window.read()

window.close()

if eting=="MOVIES vs SEATS":

oldmovieinfo=movieinfo

for kgh in df.index:

oldmovieinfo.drop([kgh])

movienames=unique\_value(movieinfo,"movie")

seatsfilledlist=[]

for asd in movienames:

oldies=oldmovieinfo[oldmovieinfo.movie==asd]

seatsfilledlist.append(sum(oldies.seatsfilled)//len(oldies))

plt.bar(movienames,seatsfilledlist)

plt.xlabel("movie names")

plt.ylabel("number of seats filled (total seats=110)")

plt.title("average seats filled")

plt.show(block=False)

elif eting=="refreshments graph":

customersnacksdates=unique\_value(customersnacks,"date")

datessbutton=[]

for iop in customersnacksdates:

datessbutton.append([sg.B(iop)])

datessbutton.append([sg.B("back")])

layout=datessbutton

window=sg.Window("dates",layout)

spea,ker = window.read()

window.close()

if spea in customersnacksdates:

partcustomersnacks=customersnacks[customersnacks.date==spea]

types=unique\_value(partcustomersnacks,"type")

typelist=[]

for asd in types:

oldies=partcustomersnacks[partcustomersnacks.type==asd]

typelist.append(len(oldies))

plt.bar(types,typelist)

plt.xlabel("snacks type ")

plt.ylabel("quantity of snacks")

plt.title("refreshment quantity vs snacks type on "+spea)

plt.show(block=False)

elif eting=="seats vs days":

uniquedates=unique\_value(movieinfo,"date")

silve=[]

premer=[]

for qst in uniquedates:

rummy=movieinfo[movieinfo.date==qst]

silve.append(sum(rummy.silver)/len(rummy))

premer.append(sum(rummy.premiere)/len(rummy))

plt.bar(uniquedates,silve)

plt.bar(uniquedates,premer)

plt.xlabel("dates")

plt.ylabel("no of seats(tot silver=90,tot premiere =20)")

plt.title("average no of seats booked")

plt.legend()

plt.grid()

plt.show(block=False)

elif eting=="seats vs time":

uniquetime=unique\_value(movieinfo,"showtime")

tota=[]

for qst in uniquetime:

rummy=movieinfo[movieinfo.showtime==qst]

tota.append(sum(rummy.seatsfilled)/len(rummy))

plt.plot(uniquetime,tota)

plt.xlabel("time")

plt.ylabel("no of seats(tot number of tickets =110)")

plt.title("average no of seats booked")

plt.legend()

plt.grid()

plt.show(block=False)

yendi=yendi+1

e=e-1

elif eventing=="movieshow INFO":

jakpot=0

dfindex=list(df.index)

while jakpot<len(df):

robi=dfindex[jakpot]

if robi in future.index:

djb=sg.B("delete")

else:

djb=sg.T("")

layout=[[sg.T(" Movie details " )],

[sg.T("date :"),sg.T(df.loc[robi]["date"])],

[sg.T("movie :"),sg.InputText(df.loc[robi]["movie"])],

[sg.T("language :"),sg.T(df.loc[robi]["language"])],

[sg.T("showtime :"),sg.T(df.loc[robi]["showtime"])],

[sg.T("certification :"),sg.InputText(df.loc[robi]["certification"])],

[sg.T("image file 1 :"),sg.InputText(df.loc[robi]["image file 1"])],

[sg.T("image file 2 :"),sg.InputText(df.loc[robi]["image file 2"])],

[sg.T("format :"),sg.InputText(df.loc[robi]["format"])],

[sg.T("directed by :"),sg.InputText(df.loc[robi]["directed by"])],

[sg.T("cast :"),sg.InputText(df.loc[robi]["cast"])],

[sg.T("genre :"),sg.InputText(df.loc[robi]["genre"])],

[sg.T("bookingscreen :"),sg.T(df.loc[robi]["bookingscreen"])],

[sg.T("trailer :"),sg.InputText(df.loc[robi]["trailer"])],

[sg.T("seatsfilled :"),sg.T(df.loc[robi]["seatsfilled"])],

[sg.T("silver type :"),sg.T(df.loc[robi]["silver"])],

[sg.T("premiere type :"),sg.T(df.loc[robi]["premiere"])],

[sg.B("Previous"),sg.B("Next"),sg.B("update"),sg.B("graph"),

djb,sg.B("back")]]

window=sg.Window("details",layout)

iron,man=window.read()

window.close()

if iron=="Next":

if jakpot==len(df)-1:

jakpot=0

else:

jakpot=jakpot+1

elif iron=="Previous":

if jakpot==0:

jakpot=len(df)-1

else:

jakpot=jakpot-1

elif iron=="update":

fz=list(df[df.movie==df.loc[robi]["movie"]].index)

for dfg in fz:

movieinfo.loc[dfg,"movie"]=man[0]

movieinfo.loc[dfg,"certification"]=man[1]

movieinfo.loc[dfg,"image file 1"]=man[2]

movieinfo.loc[dfg,"image file 2"]=man[3]

movieinfo.loc[dfg,"format"]=man[4]

movieinfo.loc[dfg,"directed by"]=man[5]

movieinfo.loc[dfg,"cast"]=man[6]

movieinfo.loc[dfg,"genre"]=man[7]

movieinfo.loc[dfg,"trailer"]=man[8]

movieinfo.to\_csv("movie timing pro.csv")

df=movieinfo

for ggwp in df.index:

timeobj=df.loc[ggwp]['date']+(" ")+df.loc[ggwp]['programtime']

timeformat=datetime.strptime(timeobj,"%d/%m/%Y %H:%M")

timeformatobj=df.loc[ggwp]['date']

timeformatfut=datetime.strptime(timeformatobj,"%d/%m/%Y")

timeformatstr=timeformat.strftime("%d/%m/%Y %H:%M")

timeformatday,timeformattime=timeformatstr.split(' ')

if timeformat<now :

df=df.drop([ggwp])

elif iron=="delete":

df=df.drop([robi])

future=future.drop([robi])

movieinfo=movieinfo.drop([robi])

movieinfo.to\_csv("movie timing pro.csv")

jakpot=0

dfindex=list(df.index)

sg.popup("deleted")

elif iron=="graph":

xxxx=[df.loc[robi]["silver"]//90,df.loc[robi]["premiere"]//20]

print(xxxx)

plt.bar(["silver","premiere"],xxxx)

plt.yticks([0,1],xxxx[0])

plt.show()

else :

e=e-1

break

elif eventing=="SHOW HISTORY":

jakpot=0

movieinfoindex=list(movieinfo.index)

while jakpot<len(movieinfo):

robi=movieinfoindex[jakpot]

layout=[[sg.T(" Movie details " )],

[sg.T("date :"),sg.T(movieinfo.loc[robi]["date"])],

[sg.T("movie :"),sg.T(movieinfo.loc[robi]["movie"])],

[sg.T("language :"),sg.T(movieinfo.loc[robi]["language"])],

[sg.T("showtime :"),sg.T(movieinfo.loc[robi]["showtime"])],

[sg.T("certification :"),sg.T(movieinfo.loc[robi]["certification"])],

[sg.T("image file 1 :"),sg.T(movieinfo.loc[robi]["image file 1"])],

[sg.T("image file 2 :"),sg.T(movieinfo.loc[robi]["image file 2"])],

[sg.T("format :"),sg.T(movieinfo.loc[robi]["format"])],

[sg.T("directed by :"),sg.T(movieinfo.loc[robi]["directed by"])],

[sg.T("cast :"),sg.T(movieinfo.loc[robi]["cast"])],

[sg.T("genre :"),sg.T(movieinfo.loc[robi]["genre"])],

[sg.T("bookingscreen :"),sg.T(movieinfo.loc[robi]["bookingscreen"])],

[sg.T("trailer :"),sg.T(movieinfo.loc[robi]["trailer"])],

[sg.B("Previous"),sg.B("Next"),sg.B("back")]]

window=sg.Window("details",layout)

iron,man=window.read()

window.close()

if iron=="Next":

if jakpot==len(movieinfo)-1:

jakpot=0

else:

jakpot=jakpot+1

elif iron=="Previous":

if jakpot==0:

jakpot=len(movieinfo)-1

else:

jakpot=jakpot-1

else:

e=e-1

break

elif eventing=="Update shows":

layout=[[sg.T("whats your aim")],[sg.B("back"),sg.B("Add New

movie"),sg.B("Upload existing movie"),]]

window=sg.Window("details",layout)

pat,cummins=window.read()

window.close()

now=dt.datetime.now()

nowstr=now.strftime("%d/%m/%Y %H:%M:%S")

today,time=nowstr.split(' ')

todayday,todaymonth,todayyear=today.split('/')

tomorrow=dt.datetime.now()+dt.timedelta(1)

dayaftertomorrow=dt.datetime.now()+dt.timedelta(2)

dayaftertomorrow1=dt.datetime.now()+dt.timedelta(2)

day4=dt.datetime.now()+dt.timedelta(3)

day5=dt.datetime.now()+dt.timedelta(4)

day6=dt.datetime.now()+dt.timedelta(5)

day7=dt.datetime.now()+dt.timedelta(6)

tomorrow=tomorrow.strftime("%d/%m/%y")

tomorrowday,tomorrowmonth,tomorrowyear=tomorrow.split('/')

dayaftertomorrow=dayaftertomorrow.strftime("%d/%m/%y")

dayaftertomorrowday,dayaftertomorrowmonth,dayafter

tomorrowyear=dayaftertomorrow.split('/')

day4=day4.strftime("%d/%m/%y")

day4day,day4month,day4year=day4.split('/')

day5=day5.strftime("%d/%m/%y")

day5day,day5month,day5year=day5.split('/')

day6=day6.strftime("%d/%m/%y")

day6day,day6month,day6year=day6.split('/')

day7=day7.strftime("%d/%m/%y")

day7day,day7month,day7year=day7.split('/')

if todayday[0]=="0":

todayday=todayday.replace("0","",1)

if todaymonth[0]=="0":

todaymonth=todaymonth.replace("0","",1)

if tomorrowday[0]=="0":

tomorrowday=tomorrowday.replace("0","",1)

if tomorrowmonth[0]=="0":

tomorrowmonth=tomorrowmonth.replace("0","",1)

if dayaftertomorrowday[0]=="0":

dayaftertomorrowday=dayaftertomorrowday.replace("0","",1)

if dayaftertomorrowmonth[0]=="0":

dayaftertomorrowmonth=dayaftertomorrowmonth.replace("0","",1)

if day4day[0]=="0":

day4day=day4day.replace("0","",1)

if day4month[0]=="0":

day4month=day4month.replace("0","",1)

if day5day[0]=="0":

day5day=day5day.replace("0","",1)

if day5month[0]=="0":

day5month=day5month.replace("0","",1)

if day6day[0]=="0":

day6day=day6day.replace("0","",1)

if day6month[0]=="0":

day6month=day6month.replace("0","",1)

if day7day[0]=="0":

day7day=day7day.replace("0","",1)

if day7month[0]=="0":

day7month=day7month.replace("0","",1)

today=todayday+'/'+todaymonth+'/'+todayyear

tomorrow=tomorrowday+'/'+tomorrowmonth+'/'+todayyear[:2]+tomorrowyear

dayaftertomorrow=dayaftertomorrowday+'/'+dayaftertom

orrowmonth+'/'+todayyear[:2]+dayaftertomorrowyear

day4=day4day+'/'+day4month+'/'+todayyear[:2]+day4year

day5=day5day+'/'+day5month+'/'+todayyear[:2]+day5year

day6=day6day+'/'+day6month+'/'+todayyear[:2]+day6year

day7=day7day+'/'+day7month+'/'+todayyear[:2]+day7year

datestot=[today,tomorrow,dayaftertomorrow,day4,day5,day6,day7]

if pat=="Add New movie":

programtimes=unique\_value(movieinfo,"programtime")

certification=unique\_value(movieinfo,"certification")

shows=list(movieinfo.showtime)

screen=list(movieinfo.bookingscreen)

layout=[[sg.T("date: ",size=(12,1)),sg.Listbox(datestot,key="0")],

[sg.T("program time: ",size=(12,1)),sg.Listbox(programtimes,key="1")],

[sg.T("movie name: ",size=(12,1)),sg.Input()],

[sg.T("language: ",size=(12,1)),sg.Listbox(["tamil","english","telugu",

"hindi","malayalam"],key="3")],

[sg.T("certification: ",size=(12,1)),sg.Listbox(certification,key="4")],

[sg.T("imagefile 1: ",size=(12,1)),sg.Input()],

[sg.T("imagefile 2: ",size=(12,1)),sg.Input()],

[sg.T("format: ",size=(12,1)),sg.Listbox(["2D","3D"],key="7")],

[sg.T("directed by: ",size=(12,1)),sg.Input()],

[sg.T("cast: ",size=(12,1)),sg.Input()],

[sg.T("genre: ",size=(12,1)),sg.Input()],

[sg.T("bookingscreen: ",size=(12,1)),sg.Listbox(screen,key="11")],

[sg.T("trailer ",size=(12,1)),sg.Input()],

[sg.B("update"),sg.B("back")]]

window=sg.Window("details",layout)

keer,thana=window.read()

window.close()

if keer=="update":

if thana["0"] and thana["1"] and thana["4"]

and thana["3"] and thana["4"] and thana["7"] and thana["11"]:

jaddu=0

for ggwp in movieinfo.index:

timeobj=movieinfo.loc[ggwp]['date']+("

")+movieinfo.loc[ggwp]['programtime']

timeformat=datetime.strptime(timeobj,"%d/%m/%Y %H:%M")

timeformatstr=timeformat.strftime("%d/%m/%Y %H:%M")

timeformatday,timeformattime=timeformatstr.split(' ')

if timeformatday==thana["0"][0] and timeformattime==thana["1"][0]:

jaddu=1

for kol in range(len(programtimes)):

if programtimes[kol]==thana["1"][0]:

showssel=shows[kol]

if jaddu==0:

movieinfo.loc[max(movieinfo.index)+1]=thana["0"][0]

,thana["1"][0],thana[0],thana["3"][0],showssel,

thana["4"][0],thana[1],thana[2],thana["7"][0],thana[3]

,thana[4],thana[5],thana["11"][0],thana[6],0,0,0

df.loc[max(movieinfo.index)+1]=thana["0"][0],thana["1"][0]

,thana[0],thana["3"][0],

showssel,thana["4"][0],thana[1],thana[2],thana["7"][0]

,thana[3],thana[4],thana[5],thana["11"][0],thana[6],0,0,0

movieinfo.to\_csv("movie timing pro.csv")

if dayaftertomorrow1 < datetime.strptime(thana["0"][0],"%d/%m/%Y"):

if len(future)==0:

future.loc[0]=thana["0"][0],thana["1"][0]

,thana["2"][0],thana["3"][0],showssel,

certification,imagefile1,imagefile2,

format,directedby,cast,genre,thana["5"][0],trailer,0,0,0

else:

future.loc[max(future.index)+1]=thana["0"][0],

thana["1"][0],thana["2"][0],thana["3"][0],

showssel,certification,imagefile1,imagefile2,format,directedby

,cast,genre,thana["5"][0],trailer,0,0,0

else:

sg.popup("sorry chosse other date /time")

else:

sg.popup("select proplerly")

if pat=="Upload existing movie":

hulk=0

while hulk<1:

movienames=unique\_value(movieinfo,"movie")

programtimes=unique\_value(movieinfo,"programtime")

shows=unique\_value(movieinfo,"showtime")

fakescreen=list(movieinfo.bookingscreen)

screen=[]

for hhh in originalscreen:

kiclu=0

for jjjj in fakescreen:

if hhh==jjjj:

kiclu=1

if kiclu==0:

screen.append(hhh)

seatis=pd.read\_csv(hhh)

seatis=seatis.set\_index("\_")

for uyt in seatis.index:

for pqr in range(1,11):

seatis.loc[uyt,str(pqr)]=seatis.loc[uyt]["alphabets"]+str(pqr)

seatis.to\_csv(hhh)

#if len(screen)==0:

# sg.popup("sorry not available")

# e=e-1

# break

layout=[[sg.T("date: ",size=(12,1)),sg.Listbox(datestot,key="0")],

[sg.T("program time: ",size=(12,1)),sg.Listbox(programtimes,key="1")],

[sg.T("movie name: ",size=(12,1)),sg.Listbox(movienames,key="2")],

[sg.T("language: ",size=(12,1)),sg.Listbox(["tamil","english","telugu",

"hindi","malayalam"],key="3")],

[sg.T("bookingscreen: ",size=(12,1)),sg.Listbox(screen,key="5")],

[sg.B("update"),sg.B("back")]]

window=sg.Window("details",layout)

keer,thana=window.read()

window.close()

if keer=="update":

if thana["0"] and thana["1"] and thana["2"] and thana["3"] and thana["5"]:

jaddu=0

for ggwp in movieinfo.index:

timeobj=movieinfo.loc[ggwp]['date']+("

")+movieinfo.loc[ggwp]['programtime']

timeformat=datetime.strptime(timeobj,"%d/%m/%Y %H:%M")

timeformatstr=timeformat.strftime("%d/%m/%Y %H:%M")

timeformatday,timeformattime=timeformatstr.split(' ')

if movieinfo.loc[ggwp]['date']==thana["0"][0] and

movieinfo.loc[ggwp]['programtime']==thana["1"][0]:

jaddu=1

if jaddu==0:

for kol in range(len(programtimes)):

if programtimes[kol]==thana["1"][0]:

showssel=shows[kol]

for kjh in movieinfo.index:

if movieinfo.loc[kjh]["movie"]==thana["2"][0]:

certification=movieinfo.loc[kjh]["certification"]

imagefile1= movieinfo.loc[kjh]["image file 1"]

imagefile2= movieinfo.loc[kjh]["image file 2"]

format = movieinfo.loc[kjh]["format"]

directedby= movieinfo.loc[kjh]["directed by"]

cast = movieinfo.loc[kjh]["cast"]

genre = movieinfo.loc[kjh]["genre"]

trailer = movieinfo.loc[kjh]["trailer"]

movieinfo.loc[max(movieinfo.index)+1]=thana["0"][0],thana["1"][0]

,thana["2"][0],thana["3"][0],showssel,certification,imagefile1

,imagefile2,format,directedby,cast,genre,thana["5"][0],

trailer,0,0,0

if len(df)==0:

df.loc[0]=thana["0"][0],thana["1"][0],thana["2"][0],thana["3"][0]

,showssel,certification,imagefile1,imagefile2,format,directedby

,cast,genre,thana["5"][0],trailer,0,0,0

else:

df.loc[max(df.index)+1]=thana["0"][0],thana["1"][0],

thana["2"][0],thana["3"][0],showssel,certification,

imagefile1,imagefile2,format,directedby,cast,

genre,thana["5"][0],trailer,0,0,0

movieinfo.to\_csv("movie timing pro.csv")

if dayaftertomorrow1 < datetime.strptime(thana["0"][0],"%d/%m/%Y"):

if len(future)==0:

future.loc[0]=thana["0"][0],thana["1"][0],thana["2"][0],

thana["3"][0],showssel,certification,imagefile1,imagefile2,

format,directedby,cast,genre,thana["5"][0],trailer,0,0,0

else:

future.loc[max(future.index)+1]=thana["0"][0],thana["1"][0]

,thana["2"][0],thana["3"][0],showssel,certification,imagefile1

,imagefile2,format,directedby,cast,genre,thana["5"][0],

trailer,0,0,0

sg.popup("updated")

else:

sg.popup("sorry chosse other date /time")

hulk=hulk-1

else:

sg.popup("select proplerly")

hulk=hulk-1

hulk=hulk+1

e=e-1

elif eventing=="view bookingscreen":

f=0

while f<1:

movies=list(df.movie)

shows=list(df.showtime)

dates=list(df.date)

screen=list(df.bookingscreen)

listscreen=[]

for gan in range(len(shows)):

listscreen.append([sg.B(movies[gan].upper()+" "+ dates[gan]+" -

"+shows[gan],size=(35,1),key=screen[gan])])

listscreen.append([sg.B("back")])

layout=listscreen

window=sg.Window("details",layout)

tik,vals = window.read()

window.close()

if tik=="back":

e=e

for dum in range(len(screen)):

if tik==screen[dum]:

junga=pd.read\_csv(screen[dum])

kkk=[]

jjj=[]

iii=[]

hhh=[]

ggg=[]

fff=[]

eee=[]

ddd=[]

ccc=[]

bbb=[]

aaa=[]

rows=[kkk,jjj,iii,hhh,ggg,fff,eee,ddd,ccc,bbb,aaa]

for var in junga.index:

for dor in range(1,11):

rows[var].append(sg.T(junga.loc[var][str(dor)],size=(5,1)))

for sumo in rows:

sumo.insert(-2,sg.T("| |"))

sumo.insert(2,sg.T("| |"))

sumo.insert(0,sg.T("| |"))

sumo.append(sg.T("| |"))

lay=[[sg.T("SOAPANASUNDARI

CINEMAS",size=(55,1),justification='c',font=("arial",15))],

[sg.T("ShowDate:"+dates[dum],size=(25,1)),sg.T("Movie:"

+movies[dum],size=(25,1)),

sg.T("ShowTime:"+shows[dum],size=(25,1))],

[sg.T("EXT"),sg.T(" "),sg.T("EXT")],

kkk,

jjj,

[sg.T("| | \_\_ \_\_\_ \_ | |\_\_\_ \_\_\_ \_\_ \_\_\_ | |\_\_\_ \_\_ | |")],

iii,

hhh,

ggg,

fff,

eee,

ddd,

ccc,

bbb,

aaa,

[sg.T("| \_\_\_ \_\_\_ | |\_\_\_ \_\_\_ \_\_\_ | | \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_|")],

[sg.T("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")],

[sg.T(" A L L E Y E S T H I S W A Y P L E A S E ")],

[sg.T("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_") ],

[sg.T("BKED - booked")],

[sg.B("back"),sg.T("SILVER(A-I): Rs 150",font=("arial",10)),sg.T("

"),sg.T("PREMIERE(J,K):

Rs 170",font=("arial",10))]]

window=sg.Window("bookingscreen",lay,default\_

button\_element\_size=(4,1), auto\_size\_buttons=False)

bucket,vals = window.read()

window.close()

if bucket=="back" or bucket==sg.WIN\_CLOSED:

f=f-1

f=f+1

e=e-1

e=e+1

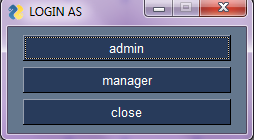
if noo == 1:

d=d-1

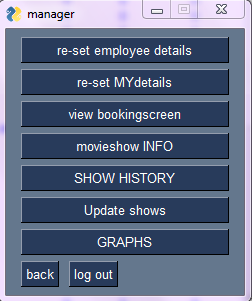
sg.popup("your password or name is wrong")

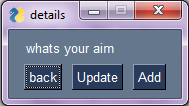
d=d+1

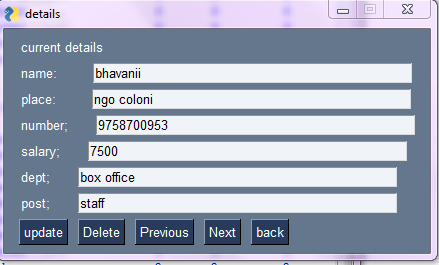
c=c+1

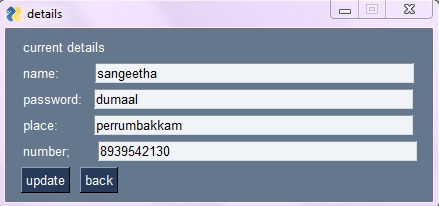


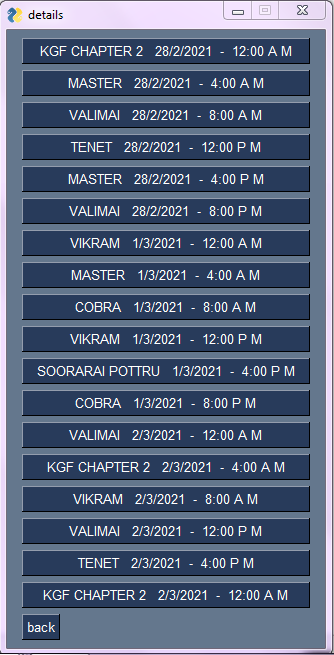


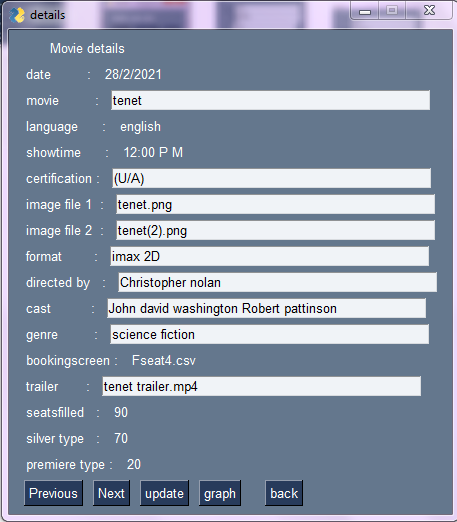


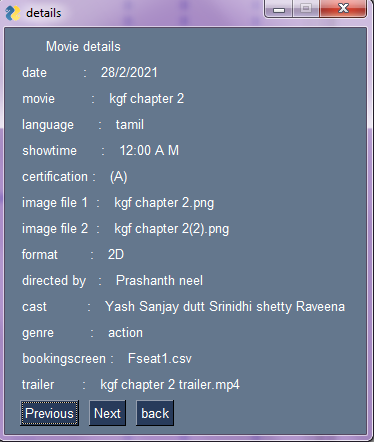


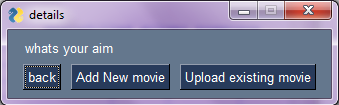


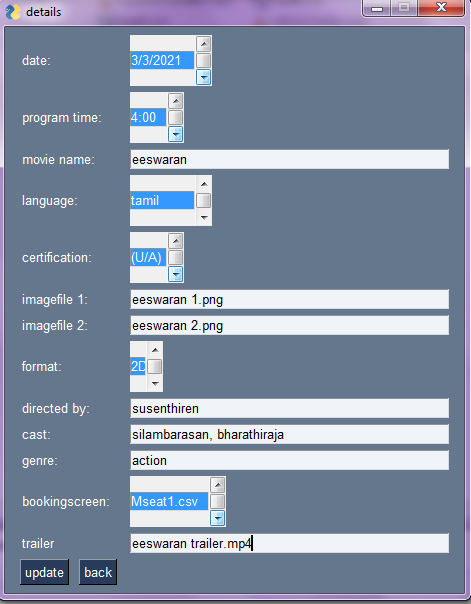


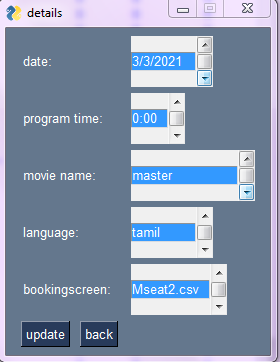


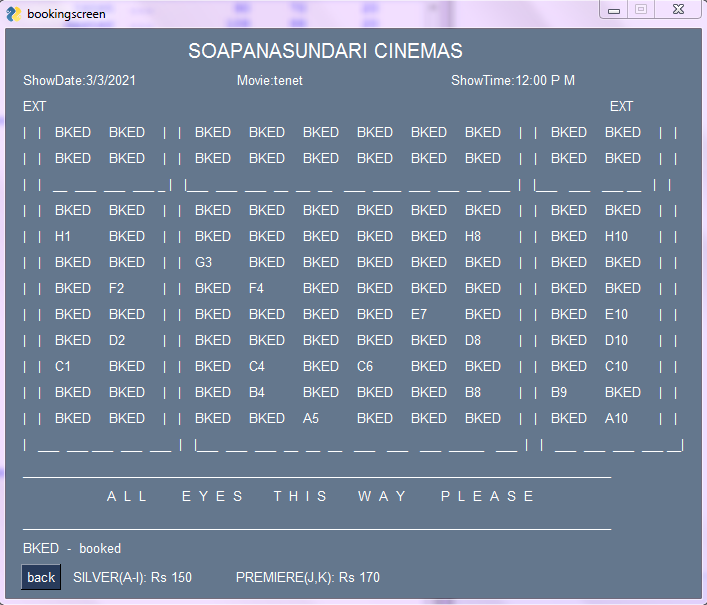


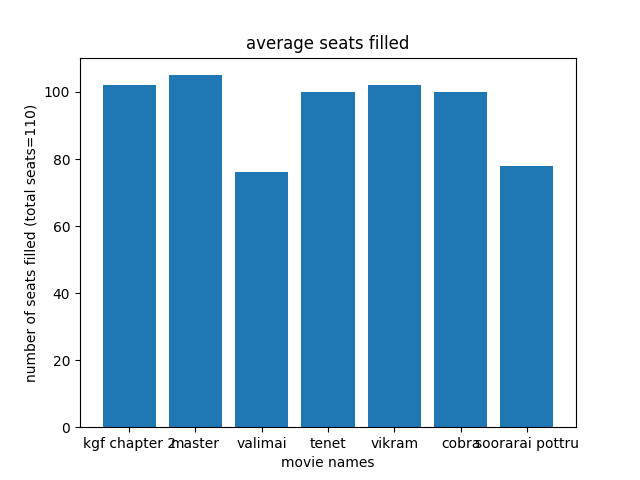


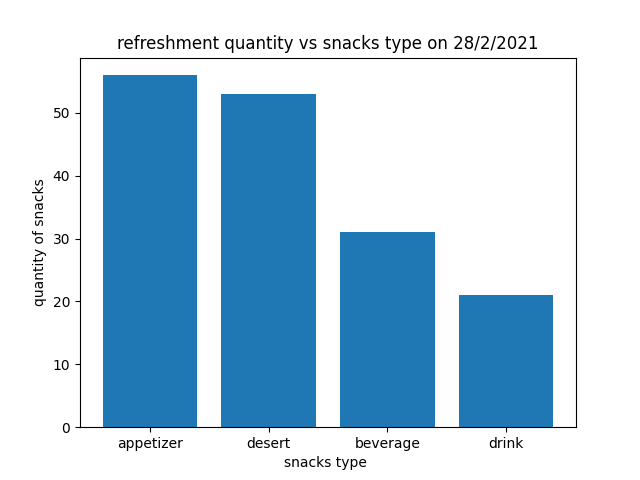


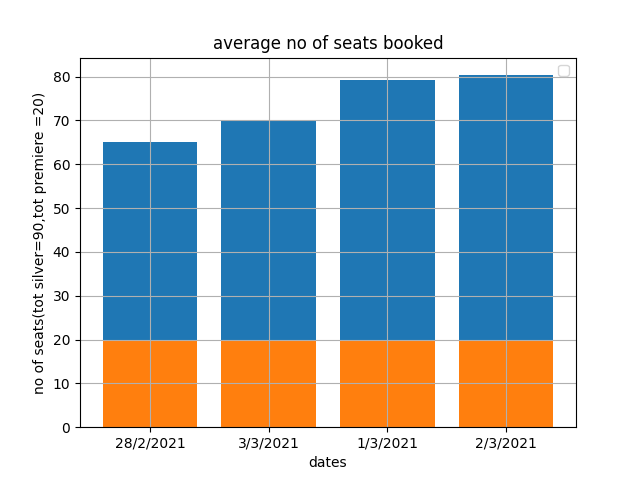


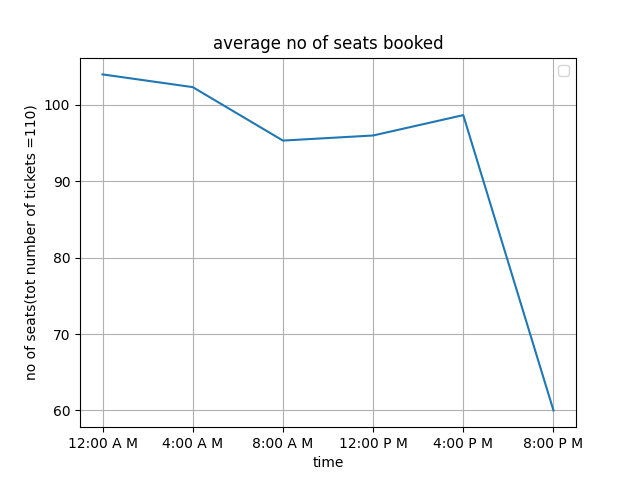












PROJECT LIMITATIONS:

1. The cinema contain only 1 screen
2. The showtime is only 12 A. M ,4 A. M, 8 A. M, 12 P. M, 4 P.M, 8 P.M it cannot be changed
3. A new refreshments type csnnot be added
4. the booking is only for current day and 2 days ahead .

REFERENCES:

1.https://pysimplegui.readthedocs.io/en/latest/cookbook/

Conclusion:

The project titled \_\_ \_\_ \_\_\_ \_\_\_\_ \_\_\_ \_\_\_\_ \_\_\_\_\_ \_\_\_\_\_\_ \_\_\_done by \_\_ \_ \_ \_ \_\_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_for the academic year 2020-2021

has been completed,,compiled,tested, and executed successfully, ensuring no plagiarism and violaion of copyright issues .