A Project Report

ON

MOVIE TICKET MANAGEMENT

For

AISSCE 2021 Examination

[As a part of the Informatics Practices Course (New)]

BY

Soumya Sangam Jha

Class 12 I

Birla Public School

Birla Public School

DOHA-QATAR

Affiliated to Central Board of Secondary Education, Delhi, India

Affiliation No: 6330006

BONAFIDE CERTIFICATE

INFORMATICS PRACTICES

This is certified to be the Bonafide work done by

Miss/Master……………………………………………

…… Roll No ……….……………. Of Class……………….....in

Birla Public School, Doha during the year …………

Dated………………………. Teacher-in-Charge

School Seal Signature of the Principal

Submitted for the ALL INDIA SENIOR SCHOOL CERTIFICATE EXAMINATION ON______ /_____ / 2021 held at Birla Public School, Doha

Date Internal Examiner External Examiner

ACKNOWLEDGEMENT

Primarily I would like to thank the entire Management team for helping me complete this project with success. Then, I express my sincere thanks to our beloved principal sir, Mr. AP Sharma, who has always motivated and encouraged us to try our best for the successful execution of this project.

I would also like to acknowledge with much appreciation our Vice Principal Ma'am Ms. Radhika Rele for providing me with all the facilities required for the project.

I pay my deep sense of gratitude to my Informatics Practices teacher, Ms. Vinu Susan George, whose valuable guidance helped me complete this project. Ma'am's suggestions and instructions have served as an important contributor towards the completion of this project.

INDEX

INTRODUCTION

This software project is developed to automate the functionalities of a movie ticket counter. The purpose of this software is to enable users to book a movie ticket and edit its details or cancel their booking whenever and wherever they want. The popularities of various things are also given in form of different graphics to enhance the decision making.

Our software mainly consists of a computerized database which is connected to an application program for easy access. Using this application program or front – end. We can retrieve, store, update or delete information in the database proficiently.

This software is a simple yet effective tool to reserve tickets for your favorite movies.

For the coding of the software project, Python Jupyter notebook is used as a front-end for getting an integrated platform which answers user friendly and comprehensible coding.

This software, being simple in design and working, does not require much of training to users, and can be used as a powerful tool for automating a Movie Ticket Center.

An open source Relational Database Management System, MySQL is used as a back-end as per the requirement of the CBSE curriculum of Informatics Practices courses (New).

System Specifications

Hardware Specifications

The used hardware are:

Laptop with Intel (R) Core (TM) i3-4030U CPU @1.90 GHz processor having 4.00 GB RAM

Software Specifications

Microsoft Windows 10 Pro as Operating System

Jupyter Notebook Python 3 as Front End

MySQL 8.0 as Back End

Microsoft Word for Project Documentation

OBJECTIVE & SCOPE OF THE PROJECT

The objective of the project is to develop a computerized ticketing system for movies and to save our time. This software project is also dined to enhance the decision making of the user by providing graphical analysis of various factors. It also allows them to view the up – to date information.

The proposed software system is expected to do the following functionality:

- ★ To provide a user friendly experience while booking tickets.
- ★ The proposed system should maintain all the information and records and generate a report whenever required.
- ★ To reduce data redundancy as very powerful RDBMS is used as the back end.
- ★ To allow the users to update or delete their records whenever required.
- ★ To provide user friendly interface to interact with a centralized database based on client – server architecture.

In its current scope, the software enables user to retrieve, update and delete information from a centralized database in MySQL. This software does not requires much training time of the users due to its limited functionality and simplicity.

Certain elements of this project has been done considering the COVID-19 situation. For example, seat selection is not an available option because based on the number booked tickets, the management will decide the seats to accommodate maximum social distancing. Also, the remaining seats depend on the number of tickets booked for a particular movie for the entire day instead of the particular showing to ensure that no seat is reused until all the theatres undergo deep cleaning at the end of the day.

Despite the best efforts of the developers the following limitations and functional boundaries are evident, which limits the scope of the application software

1. There is no provision to delete the old records automatically.
2. Some applications are like updating of the movies are not implemented in the project. It requires the managers to manually update the data from the back end.

So as far as future scope of the project is concerned, firstly it is open to any modular expansion, i.e. Other modules or functions can be designed and embedded according to the need of the user. The project can be modified without much effort.

Theoretical Background

What is a Database?

Introduction and Concepts

A database may be defined as a collection of interrelated data stored together to serve multiple applications. It contains information about one particular enterprise. It maintains any information that may be necessary to the decision-making process involved in the management of the organization. The data is stored is such a way that it is independent of the programs that use the data. A common and controlled approach is used in adding new data and in modifying and retrieving existing data within the database.

A database management system (DBMS) refers to a software that is responsible for storing, maintaining and utilizing databases. A database along with a DBMS is referred to as a database system.

A data model is a set of concepts that describe the type of data and associated constraints.

Some common data models are Relational, object-Oriented, Network and Hierarchical data model.

A relational data model comprises of data in forms of relations or tables. A table is a combination of rows (also known as records or tuples) and columns (also known as fields or attributes) which stores logically related data. Some common Relational Database Management Systems are Oracle database, MySQL, PostgreSQL etc.

A primary key is a set of one or more attributes that can uniquely identify tuples within a relation. A non-key attribute, whose values are derived from the primary key of some other table is known as a foreign key.

Queries help to add, retrieve, update or delete the data and perform in-built or custom calculations.

Role of RDBMS

A database system should be a repository of the data needed for an organizations data processing. That data should be accurate, private and protected from damage. It should also be organized so that diverse applications with different data requirements can employ the data.

The ways in which end user want to utilize existing data constantly changes. The extent to which these demands can be satisfied determines the overall value of the database system.

A database system is expected to do the following:

- ➔ Reduce data redundancy
- ➔ Maintain data consistency
- ➔ Facilitate sharing of data
- ➔ Enforce data standards
- ➔ Provide data security

Most of the database management systems have the following capabilities:

- ★ Retrieval, insertion, deletion and modification of the data stored.
- ★ Creating of a table or deleting it
- ★ Altering the table structure or renaming its elements
- ★ Grant or revoke editing privileges
- ★ Perform various transactions
- ★ Perform mathematical operations

Introduction to MySQL

MySQL is a freely available open source Relational Database Management System that uses Structured Query Language (SQL). In a MySQL database, information is stored in tables. A single MySQL database can contain many tables at once and store thousands of individual records. MySQL provides you with a rich set of features that support a secure environment for storing, maintaining and accessing data.

MySQL was created and is supported by MySQL AB, a company based in Sweden. Micheal Widenius was the chief inventor of MySQL. MySQL has been named after his daughter My. The logo of MySQL, the dolphin, is named as Sakila.

MySQL database system refers to the combination of a MySQL server instance and a MySQL database. It operates using client/server architecture in which the server runs on the machine containing the databases and client connect to the server over a network. It is also a multi user database system.

The MySQL server listens for client requests coming in over the network and accesses database contents according to those requests and provides that to the clients. Clients are programs that connect to the database server and issue queries in a pre-specified format.

Some key features of MySQL are:

★ It is fast, reliable and easy to use. If the server hardware is optimal, MySQL runs very fast. It is also a high-performance, relatively simple database system and available free of cost.

★ MySQL supports Structured Query Language.

★ It is a secure RDBMS. It offers a privilege and password system that is very flexible and secure, and that allows host-based verification. All the password traffic is encrypted when you connect to a server.

★ It is an Open source database. It is a part of LAMP (Linux, Apache, MySQL, PHP / Perl / Python) environment which is a fast growing open source enterprise.

★ MySQL provides portability as it has been tested with a broad range of different compilers and can work on many different platforms.

★ MySQL provides many datatypes to support different types of data. It also supports mixed-length and variable-length records.

★ MySQL can handle large databases.

★ Clients can connect to MySQL server using several protocols.

★ The server can provide error messages to clients in many languages.

Introduction to Jupyter Notebook

Python is an interpreted language and you need python interpreter to execute Python programs. As python is becoming preferred choice for multiple applications such as data sciences, business applications, numeric and scientific computing etc., you need to additional libraries to use in Python for the intended use. In Python, the packages are not the part of the python standard library. Python distributions repackage default CPython distribution with some additional packages and libraries for specialized usage. Anaconda distribution is one such distribution which is used for data management, analysis and visualization of large data sets. It contains many packages and modules including Jupyter.

The Jupyter Notebook application allows you to create and edit documents that display the input and output of a Python script. Once saved, you can share these files with others.

The notebook extends the console-based approach to interactive computing by providing a web based application which can capture the whole computation process. The computation process includes developing, documenting, and executing code, as well as communicating the results.

It combines two components:

1. A web application

It is a browser-based tool for interactive experience of writing documents. The documents combine explanatory text, mathematics, computations and their rich media output.

2. Notebook documents

   It is a representation of all content visible in the web application, including inputs and outputs of the computations, explanatory text, mathematics, images, and rich media representations of objects.

The notebook contains various cells. The code is typed in these cells. The cell uses the kernel that you chose when you started the notebook. Running a cell will execute its contents. Variables and imports can be shared across various cells. This makes it easy to logically separate your code without needing to reimport libraries or recreate variables or functions in every cell. The square braces next to the word *in* to the left of the cell auto fill with a number that indicates the order that you ran the cells.

Jupyter notebook allows you to create a text file, a terminal or a folder in your browser as well. A terminal allows you to execute arbitrary commands inside of your notebook server. This can be useful for downloading data, copying files, inspecting processes, or editing files.

Some remarkable features of Jupyter Notebook is mentioned below:

★ It supports various cell types like Code, Markdown, Raw NBConvert. Headings can be written in Markdown cells using # characters. The Raw NBConvert cell type allows you to control the formatting in a very specific way when converting from a Notebook to another format.

★ We can create lists (bullet points) by using dashes, plus signs, or asterisks.

★ We can convert the notebook into HTML, PDF, ReStructured Text, Executable script and many more.

★ It facilitates exporting and hence sharing of notebooks.

The Jupyter Notebook is quite useful for learning and teaching a programming language such as Python as well as for sharing your data.

Problem Definition and Analysis

The hardest part of building a software system is deciding what to build. No other part of the conceptual work is as difficult as establishing the detailed technical requirement. Defining and applying good, complete requirements are hard to work, and success in this endeavor has eluded many of us. Yet, we continue to make progress.

Problem definition helps present the problem in a way that can be examined from a number of perspectives. Problem definition is one of the most essential steps in this creation process. Without defining a problem, developers do not know what to build, customers do not know what to expect, and there is no way to validate that the built system satisfies the requirement.

Problem Definition and Analysis is the activity that includes learning about the problem at hand, understanding the requirements of the users, trying to find out the target audience, and understanding all the constraints on the solution.

Organizations' need the ability to quickly adapt to the various changes in their business environment. One of the important disciplines that help them withstand the competition is problem analysis. It improves performance and seizes all the opportunities at hand.

It includes the following:

- ➔ Identifying and recording the needs of the users.
- ➔ Creating a document that describes the external behavior and the associated constraints that may satisfy those requirements.
- ➔ Analyzing and validating the documents to ensure consistency, integrity, and feasibility
- ➔ Evolution of needs.

After the analysis of the functioning of the ticketing system, the proposed System is expected to do the following:

- ➔ To provide a user-friendly experience while booking tickets.
- ➔ To maintain all the records properly and retrieve it whenever required.
- ➔ To reduce data redundancy as a very powerful RDBMS is used as the back-end.
- ➔ To allow the users to update or delete their records whenever required.
- ➔ To provide a user-friendly interface to interact with a centralized database based on client-server architecture.

Coding

```
import pandas as pd
import matplotlib.pyplot as plt
from IPython.display import display
print('Welcome to Elite Tickets.')

import mysql.connector as sqltr
con=sqltr.connect(host='localhost',user='root',passwd='bpsdoha',database='project')
if con.is_connected==False:
    print('Error in connection')
cursor=con.cursor()

while True:
    print('Date: 20 September, 2020')
    print('Day: Sunday')
    print('''Movies available:''')
    mdf=pd.DataFrame({'Movie ID':['E1493','A3032','E4523','E5099','A6041'],'Movie Name':['Movie1','Movie2','Movie3','Movie4','Movie5'],'Theatre No':['T1','T3','T4','T5','T6'],'Age Rating':['R','PG','PG','PG-13','PG'],'Language':['English','Arabic','English','English','Arabic'],'Morning':['8:00','8:15','8:45','9:15','9:45'],'Afternoon':['12:15','12:45','13:00','14:15','14:45'],'Evening':['16:30','16:45','17:00','17:30','17:45'],'Night':['20:00','20:30','20:45','21:15','21:45']},index=['i)','ii)','iii)','iv)','v)'])
    display(mdf)
    print('''1) Book tickets
2) View Booking Details
3) Edit Booking Details
4) Cancel Booking
5)Popularity''')
    print()
    choice=input('Enter your choice(1,2,3,4,5): ')
    print()

    if choice=='1':
        print('Booking tickets')
        while True:
            info=pd.read_sql_query("Select * from ticket;",con)
            movieg=info.groupby('MovieID')
            QID=int(input('Enter your Qatar ID number: '))
            MID=input('Enter your movie ID: ')
            while True:
                try:
                    moviedf=movieg.get_group(MID)
                except KeyError:
                    if MID=='E1493' or MID=='A3032' or MID=='E4523' or MID=='E5099' or
MID=='A6041':
                        BookedTickets=0
                        print('We have 200 seats remaining')
                        break
                    else:
                        print('Invalid movie ID')
                        MID=input('Enter a valid movie ID: ')
                        continue
                else:
                    BookedTickets=moviedf['NoOfTickets'].sum()
                    if BookedTickets==200:
                        print('No tickets available for',MID)
                        print('We have 0 seats remaining')
                        DiffBookQ=input('Do you want to make a booking for another
movie?(y/n): ')
                        if DiffBookQ=='y' or DiffBookQ=='yes' or DiffBookQ=='Yes' or
DiffBookQ=='Y' or DiffBookQ=='YES':
                            MID=input('Enter MovieID: ')
                            continue
                        else:
                            break
                    else:
                        print('We have',200-BookedTickets,'seats remaining')
                        break
            if BookedTickets==200:
                if DiffBookQ=='y' or DiffBookQ=='yes' or DiffBookQ=='Yes' or
DiffBookQ=='Y' or DiffBookQ=='YES':
                    pass
                else:
                    break
            Name=input('Enter your name: ')
            PhoneNo=int(input('Enter your Phone number: '))
            NoOfTickets=int(input('Enter the number of tickets: '))
            while True:
                if BookedTickets+NoOfTickets>200:
                    print('Exceeded total number of theatre seats')
                    print('We only have',200-BookedTickets,'seats remaining')
                    NoOfTickets=int(input('Enter a valid number of tickets: '))
                elif NoOfTickets<1:
                    print('Invalid number of tickets')
                    NoOfTickets=int(input('Enter a valid number of tickets: '))
                else:
                    break
            Shift=input('Enter the shift(Morning,Afternoon,Evening,Night): ')
            Snacks=input('Enter preferred snacks(Popcorn,Nachos,Fries,Chips): ')
            cursor.execute("Insert into ticket values({},'{}','{}',{},{},'{}','{}')".format(QID,MID,Name,PhoneNo,NoOfTickets,Shift,Snacks))
            con.commit()
            if NoOfTickets>1:
                print('Tickets booked')
            else:
                print('Ticket booked')
            print()
            BookAQ=input('Do you want to make another booking?(y/n): ')
            if BookAQ=='y' or BookAQ=='yes' or BookAQ=='Yes' or BookAQ=='Y' or
BookAQ=='YES':
                print()
                continue
            else:
                break

    elif choice=='2':
        print('Viewing Booking Details')
        info=pd.read_sql_query("Select * from ticket;",con)
        group=info.groupby(['QatarID','MovieID'])
        while True:
            QID=int(input('Enter your Qatar ID number: '))
            MID=input('Enter your movie ID: ')
            Specific=group.get_group((QID,MID))
            Specific.set_index('QatarID',inplace=True)
            print('Name         : ',Specific.Name[QID])
            print('Phone Number : ',Specific.PhoneNo[QID])
            print('No. of tickets: ',Specific.NoOfTickets[QID])
            print('Shift        : ',Specific.Shift[QID])
            print('Snacks       : ',Specific.Snacks[QID])
            print()
            ViewAQ=input('Do you want to view details of a different booking?(y/n): ')
            if ViewAQ=='y' or ViewAQ=='yes' or ViewAQ=='Yes' or ViewAQ=='Y' or ViewAQ=='YES':
                print()
                continue
            else:
                break

    elif choice=='3':
        print('Editing Booking Details')
        QID=int(input('Enter your Qatar ID number: '))
        MID=input('Enter your movie ID: ')
        info=pd.read_sql_query("Select * from ticket;",con)
        movieg=info.groupby('MovieID')
        group=info.groupby(['QatarID','MovieID'])
        Specific=group.get_group((QID,MID))
        Specific.set_index('QatarID',inplace=True)
        print('1)Name         : ',Specific.Name[QID])
        print('2)Phone Number : ',Specific.PhoneNo[QID])
        print('3)No. of tickets: ',Specific.NoOfTickets[QID])
        print('4)Shift        : ',Specific.Shift[QID])
        print('5)Snacks       : ',Specific.Snacks[QID])
        while True:
            Detail=input('Which detail do you want to edit?(1,2,3,4,5): ')
            if Detail=='1':
                col='Name'
                DN='Name'
            elif Detail=='2':
                col='PhoneNo'
                DN='Phone number'
            elif Detail=='3':
                col='NoOfTickets'
                DN='Number of tickets'
            elif Detail=='4':
                col='Shift'
                DN='Shift(Morning,Afternoon,Evening,Night)'
            elif Detail=='5':
                col='Snacks'
                DN='Snacks(Popcorn,Nachos,Fries,Chips)'
            else:
                print('Invalid Option')
                continue
            if Detail=='1' or Detail=='4' or Detail=='5':
                Change=input('Enter new value for '+DN+': ')
                cursor.execute("Update ticket set "+col+"='{}' where QatarID={} and
    MovieID='{}'".format(Change,QID,MID))
            elif Detail=='2':
                Change=int(input('Enter new value for '+DN+': '))
                cursor.execute("Update ticket set "+col+"={} where QatarID={} and
    MovieID='{}'".format(Change,QID,MID))
            elif Detail=='3':
                moviedf=movieg.get_group(MID)
                BookedTickets=moviedf['NoOfTickets'].sum()-Specific.NoOfTickets[QID]
                print('We have',200-BookedTickets,'seats remaining')
                Change=int(input('Enter new value for '+DN+': '))
                while True:
                    if BookedTickets+Change>200:
                        print('Exceeded total number of theatre seats')
                        print('We only have',200-BookedTickets,'seats remaining')
                        Change=int(input('Enter a valid number of tickets: '))
                    elif Change<1:
                        print('Invalid number of tickets')
                        Change=int(input('Enter a valid number of tickets: '))
                    else:
                        break
                cursor.execute("Update ticket set "+col+"={} where QatarID={} and
MovieID='{}'".format(Change,QID,MID))
            else:
                pass
            con.commit()
            print('Updated details:')
            info=pd.read_sql_query("Select * from ticket;",con)
            group=info.groupby(['QatarID','MovieID'])
            Specific=group.get_group((QID,MID))
            Specific.set_index('QatarID',inplace=True)
            print('1)Name        : ',Specific.Name[QID])
            print('2)Phone Number : ',Specific.PhoneNo[QID])
            print('3)Total tickets: ',Specific.NoOfTickets[QID])
            print('4)Shift        : ',Specific.Shift[QID])
            print('5)Snacks       : ',Specific.Snacks[QID])
            print()
            EditAQ=input('Do you want to edit another detail?(y/n): ')
            if EditAQ=='y' or EditAQ=='yes' or EditAQ=='Yes' or EditAQ=='Y' or
EditAQ=='YES':
                print()
                continue
            else:
                break

    elif choice=='4':
        print('Cancelling Booking')
        while True:
            QID=int(input('Enter your Qatar ID number: '))
            MID=input('Enter your movie ID: ')
            info=pd.read_sql_query("Select * from ticket;",con)
            group=info.groupby(['QatarID','MovieID'])
            Specific=group.get_group((QID,MID))
            Specific.set_index('QatarID',inplace=True)
            print('Name         : ',Specific.Name[QID])
            print('Phone Number : ',Specific.PhoneNo[QID])
            print('Total tickets: ',Specific.NoOfTickets[QID])
            print('Shift        : ',Specific.Shift[QID])
            print('Snacks       : ',Specific.Snacks[QID])
            CancelQ=input('Are you sure you want to cancel the above booking?(y/n):
')
            if CancelQ=='y'or CancelQ=='Y' or CancelQ=='Yes' or CancelQ=='yes' or
CancelQ=='YES':
                cursor.execute("Delete from ticket where QatarID={} and
MovieID='{}'".format(QID,MID))
                con.commit()
                print('Booking Cancelled')
            else:
                pass
            print()
            CAQ=input('Do you want to cancel another booking?(y/n): ')
            if CAQ=='y'or CAQ=='Y' or CAQ=='Yes' or CAQ=='yes' or CAQ=='YES':
                print()
                continue
            else:
                break

    elif choice=='5':
        print('Popularity')
        info=pd.read_sql_query("Select * from ticket;",con)
        while True:
            print('''1)Popularity of Snacks
```

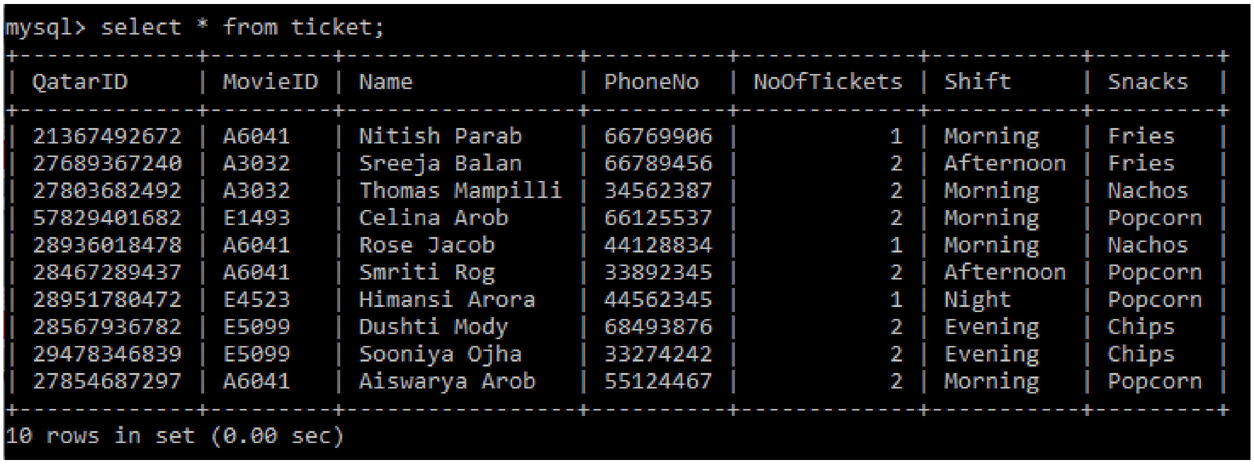
```
2)Popularity of Movies
3)Rush Hours''')
        Pop=input('Which option do you want to see?(1,2,3): ')
        if Pop=='1':
            g=info.groupby('Snacks')
            Size=g.size()
            spie=plt.pie(g['NoOfTickets'].sum(),labels=Size.index,autopct='%0.2f%%')
            plt.title('Popularity of Snacks')
            plt.show()
        elif Pop=='2':
            g=info.groupby('MovieID')
            Size=g.size()
            mpie=plt.pie(g['NoOfTickets'].sum(),labels=Size.index,autopct='%0.2f%%'
)
            plt.title('Popularity of Movies')
            plt.show()
        elif Pop=='3':
            g=info.groupby('Shift')
            Size=g.size()
            xaxis=Size.index
            yaxis=g['NoOfTickets'].sum()
            rbar=plt.bar(xaxis,yaxis)
            plt.xlabel('Shift')
            plt.ylabel('No of People')
            plt.title('Rush Hours')
```

```
                plt.show()
            else:
                print('Invalid option')
            GAgainQ=input('Do you want to see another popularity option?(y/n): ')
            if GAgainQ=='y'or GAgainQ=='Y' or GAgainQ=='Yes' or GAgainQ=='yes' or
GAgainQ=='YES':
                continue
            else:
                break

    else:
        print('Invalid Option')
    print()
    OverallAQ=input('Do you want to select another option?(y/n): ')
    if OverallAQ=='y'or OverallAQ=='Y' or OverallAQ=='Yes' or OverallAQ=='yes' or
OverallAQ=='YES':
        print()
        continue
    else:
        break

con.close()
```

Output

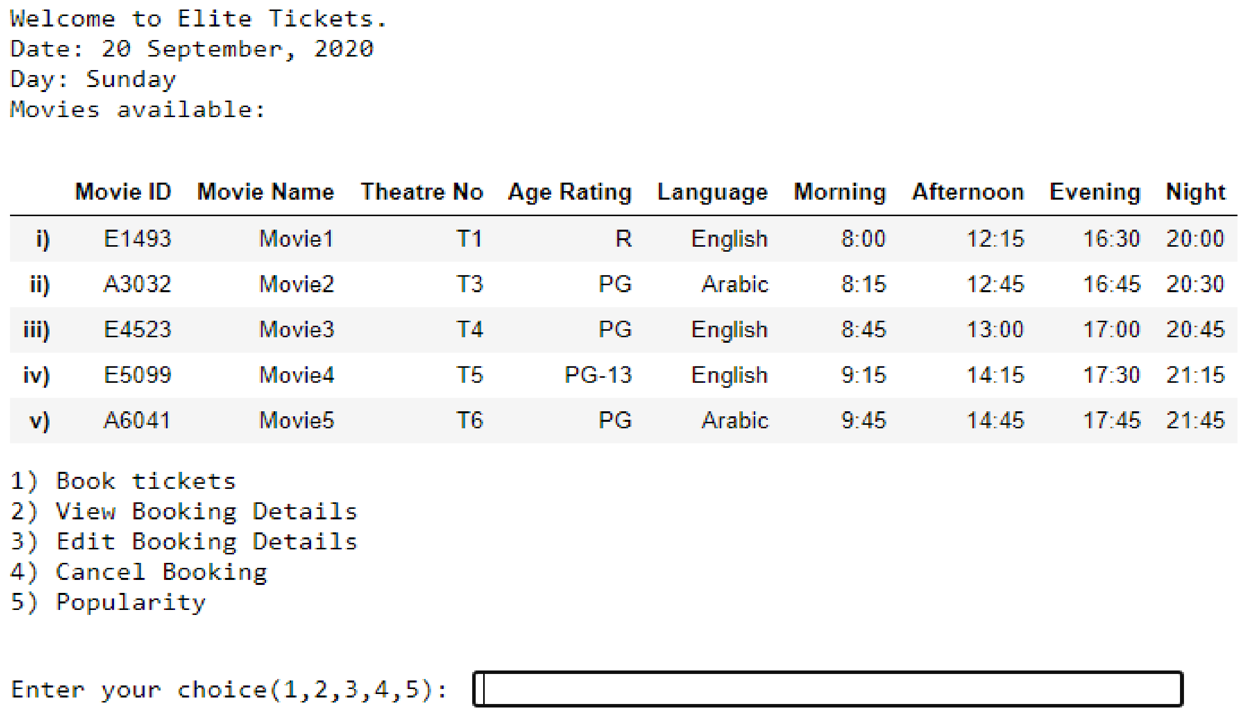
Database – Project

Table- Ticket



```
mysql> select * from ticket;
+-------------+---------+-----------------+----------+-------------+-----------+---------+
| QatarID     | MovieID | Name            | PhoneNo  | NoOfTickets | Shift     | Snacks  |
+-------------+---------+-----------------+----------+-------------+-----------+---------+
| 21367492672 | A6041   | Nitish Parab    | 66769906 |           1 | Morning   | Fries   |
| 27689367240 | A3032   | Sreeja Balan    | 66789456 |           2 | Afternoon | Fries   |
| 27803682492 | A3032   | Thomas Mampilli | 34562387 |           2 | Morning   | Nachos  |
| 57829401682 | E1493   | Celina Arob     | 66125537 |           2 | Morning   | Popcorn |
| 28936018478 | A6041   | Rose Jacob      | 44128834 |           1 | Morning   | Nachos  |
| 28467289437 | A6041   | Smriti Rog      | 33892345 |           2 | Afternoon | Popcorn |
| 28951780472 | E4523   | Himansi Arora   | 44562345 |           1 | Night     | Popcorn |
| 28567936782 | E5099   | Dushti Mody     | 68493876 |           2 | Evening   | Chips   |
| 29478346839 | E5099   | Sooniya Ojha    | 33274242 |           2 | Evening   | Chips   |
| 27854687297 | A6041   | Aiswarya Arob   | 55124467 |           2 | Morning   | Popcorn |
+-------------+---------+-----------------+----------+-------------+-----------+---------+
10 rows in set (0.00 sec)
```

Front End

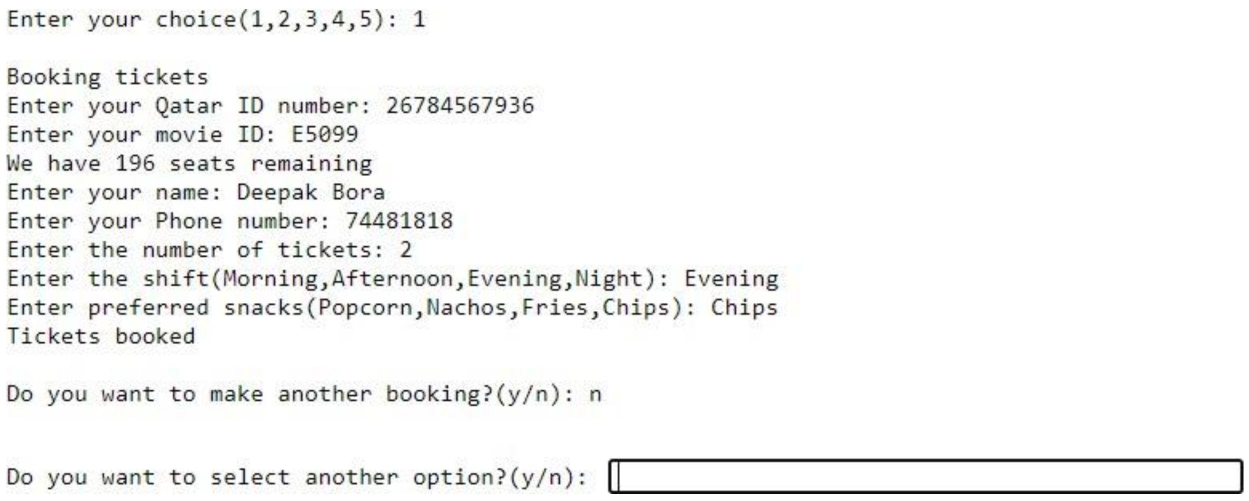


```
Welcome to Elite Tickets.
Date: 20 September, 2020
Day: Sunday
Movies available:
```

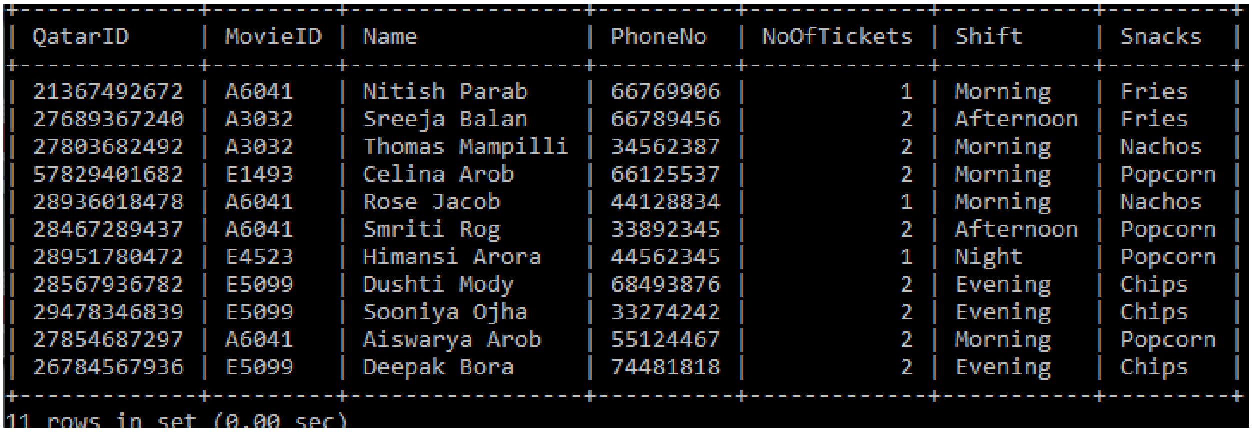
	Movie ID	Movie Name	Theatre No	Age Rating	Language	Morning	Afternoon	Evening	Night
i)	E1493	Movie1	T1	R	English	8:00	12:15	16:30	20:00
ii)	A3032	Movie2	T3	PG	Arabic	8:15	12:45	16:45	20:30
iii)	E4523	Movie3	T4	PG	English	8:45	13:00	17:00	20:45
iv)	E5099	Movie4	T5	PG-13	English	9:15	14:15	17:30	21:15
v)	A6041	Movie5	T6	PG	Arabic	9:45	14:45	17:45	21:45

```
1) Book tickets
2) View Booking Details
3) Edit Booking Details
4) Cancel Booking
5) Popularity

Enter your choice(1,2,3,4,5):
```

Choice 1 – Booking a ticket

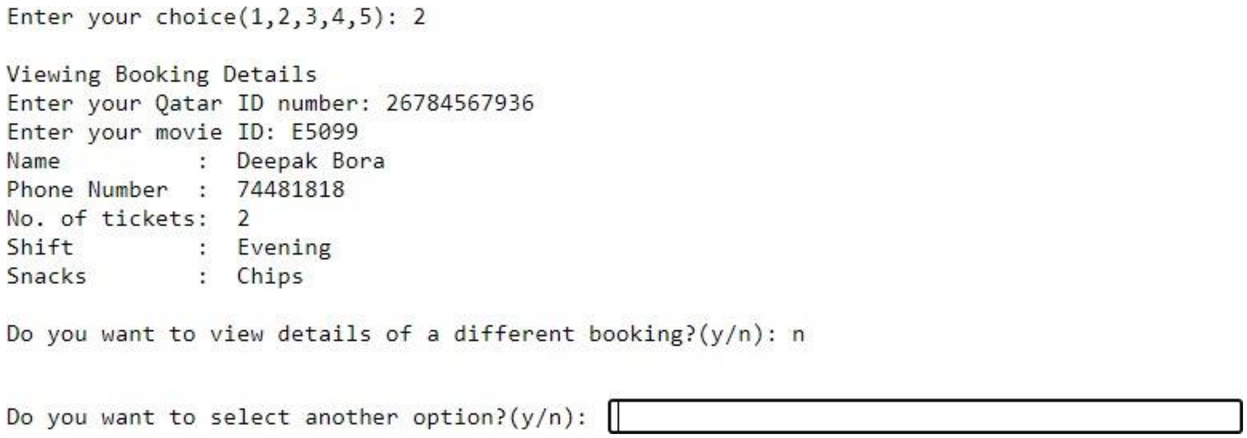


```
Enter your choice(1,2,3,4,5): 1

Booking tickets
Enter your Qatar ID number: 26784567936
Enter your movie ID: E5099
We have 196 seats remaining
Enter your name: Deepak Bora
Enter your Phone number: 74481818
Enter the number of tickets: 2
Enter the shift(Morning,Afternoon,Evening,Night): Evening
Enter preferred snacks(Popcorn,Nachos,Fries,Chips): Chips
Tickets booked

Do you want to make another booking?(y/n): n


Do you want to select another option?(y/n):
```



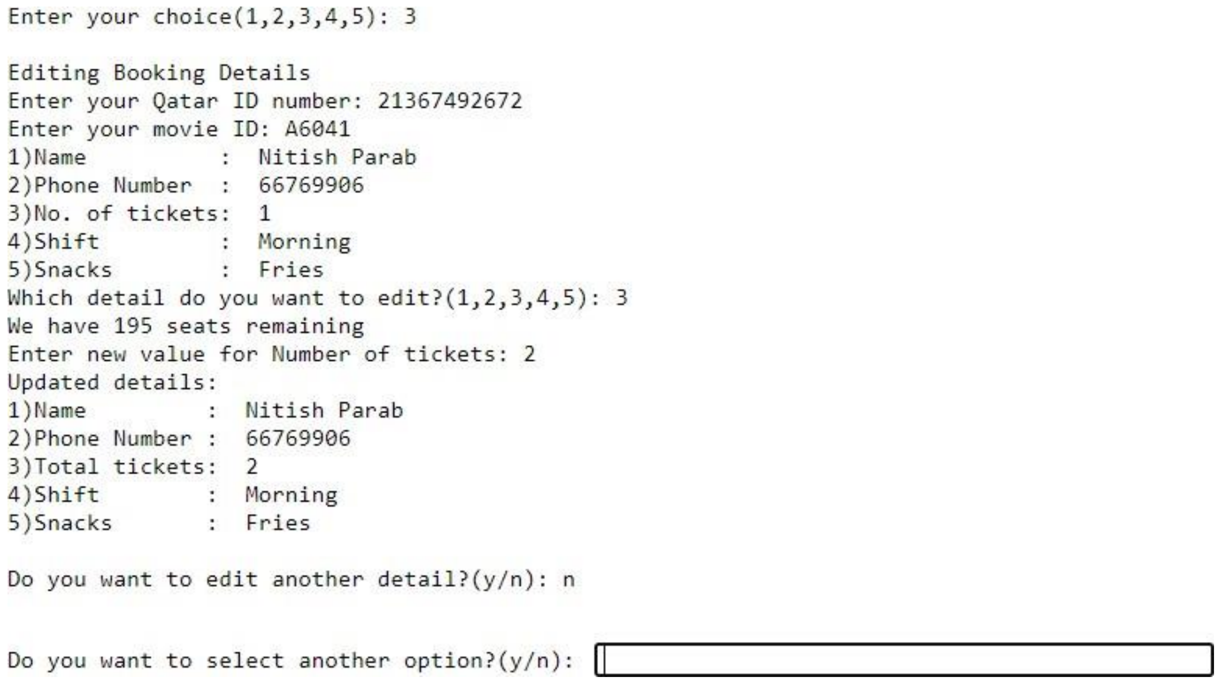
```
+-------------+---------+-----------------+----------+-------------+-----------+---------+
| QatarID     | MovieID | Name            | PhoneNo  | NoOfTickets | Shift     | Snacks  |
+-------------+---------+-----------------+----------+-------------+-----------+---------+
| 21367492672 | A6041   | Nitish Parab    | 66769906 |           1 | Morning   | Fries   |
| 27689367240 | A3032   | Sreeja Balan    | 66789456 |           2 | Afternoon | Fries   |
| 27803682492 | A3032   | Thomas Mampilli | 34562387 |           2 | Morning   | Nachos  |
| 57829401682 | E1493   | Celina Arob     | 66125537 |           2 | Morning   | Popcorn |
| 28936018478 | A6041   | Rose Jacob      | 44128834 |           1 | Morning   | Nachos  |
| 28467289437 | A6041   | Smriti Rog      | 33892345 |           2 | Afternoon | Popcorn |
| 28951780472 | E4523   | Himansi Arora   | 44562345 |           1 | Night     | Popcorn |
| 28567936782 | E5099   | Dushti Mody     | 68493876 |           2 | Evening   | Chips   |
| 29478346839 | E5099   | Sooniya Ojha    | 33274242 |           2 | Evening   | Chips   |
| 27854687297 | A6041   | Aiswarya Arob   | 55124467 |           2 | Morning   | Popcorn |
| 26784567936 | E5099   | Deepak Bora     | 74481818 |           2 | Evening   | Chips   |
+-------------+---------+-----------------+----------+-------------+-----------+---------+
11 rows in set (0.00 sec)
```

Choice 2 – View Booking Details

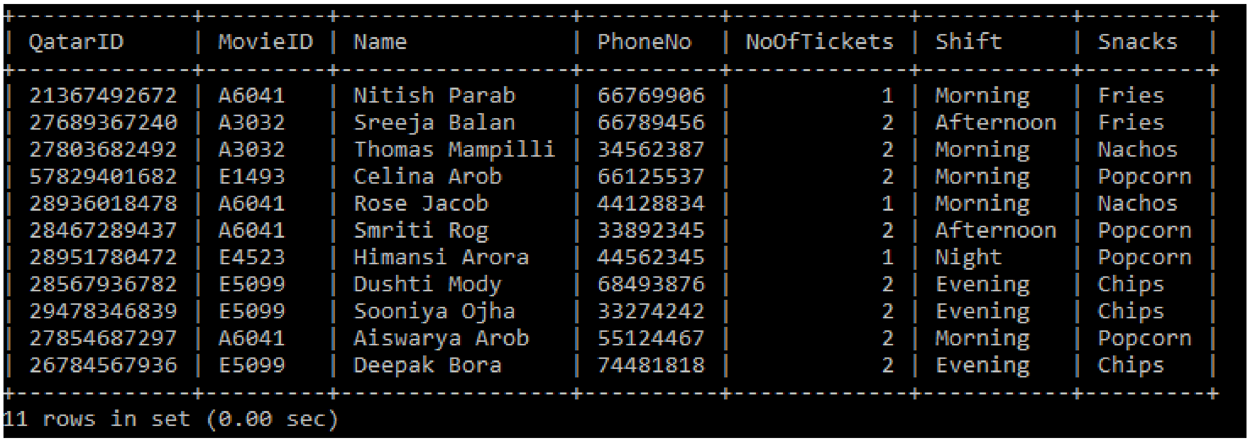


```
Enter your choice(1,2,3,4,5): 2

Viewing Booking Details
Enter your Qatar ID number: 26784567936
Enter your movie ID: E5099
Name          :  Deepak Bora
Phone Number  :  74481818
No. of tickets:  2
Shift         :  Evening
Snacks        :  Chips

Do you want to view details of a different booking?(y/n): n


Do you want to select another option?(y/n):
```

Choice 3 – Editing Booking Details

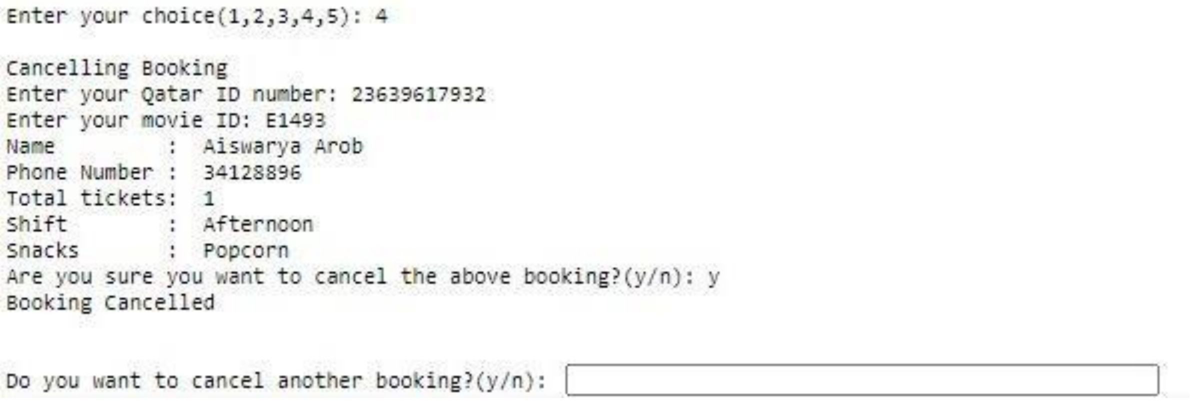


```
Enter your choice(1,2,3,4,5): 3

Editing Booking Details
Enter your Qatar ID number: 21367492672
Enter your movie ID: A6041
1)Name          :  Nitish Parab
2)Phone Number  :  66769906
3)No. of tickets:  1
4)Shift         :  Morning
5)Snacks        :  Fries
Which detail do you want to edit?(1,2,3,4,5): 3
We have 195 seats remaining
Enter new value for Number of tickets: 2
Updated details:
1)Name         :  Nitish Parab
2)Phone Number :  66769906
3)Total tickets:  2
4)Shift        :  Morning
5)Snacks       :  Fries

Do you want to edit another detail?(y/n): n


Do you want to select another option?(y/n):
```

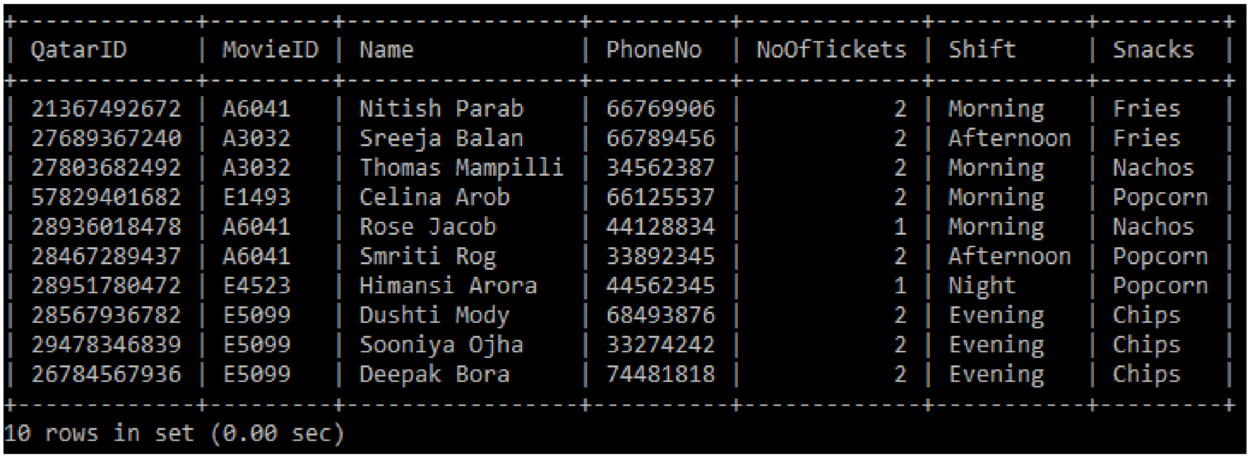


```
+-------------+---------+-----------------+----------+-------------+-----------+---------+
| QatarID     | MovieID | Name            | PhoneNo  | NoOfTickets | Shift     | Snacks  |
+-------------+---------+-----------------+----------+-------------+-----------+---------+
| 21367492672 | A6041   | Nitish Parab    | 66769906 |           1 | Morning   | Fries   |
| 27689367240 | A3032   | Sreeja Balan    | 66789456 |           2 | Afternoon | Fries   |
| 27803682492 | A3032   | Thomas Mampilli | 34562387 |           2 | Morning   | Nachos  |
| 57829401682 | E1493   | Celina Arob     | 66125537 |           2 | Morning   | Popcorn |
| 28936018478 | A6041   | Rose Jacob      | 44128834 |           1 | Morning   | Nachos  |
| 28467289437 | A6041   | Smriti Rog      | 33892345 |           2 | Afternoon | Popcorn |
| 28951780472 | E4523   | Himansi Arora   | 44562345 |           1 | Night     | Popcorn |
| 28567936782 | E5099   | Dushti Mody     | 68493876 |           2 | Evening   | Chips   |
| 29478346839 | E5099   | Sooniya Ojha    | 33274242 |           2 | Evening   | Chips   |
| 27854687297 | A6041   | Aiswarya Arob   | 55124467 |           2 | Morning   | Popcorn |
| 26784567936 | E5099   | Deepak Bora     | 74481818 |           2 | Evening   | Chips   |
+-------------+---------+-----------------+----------+-------------+-----------+---------+
11 rows in set (0.00 sec)
```

Choice 4 – Cancelling Booking



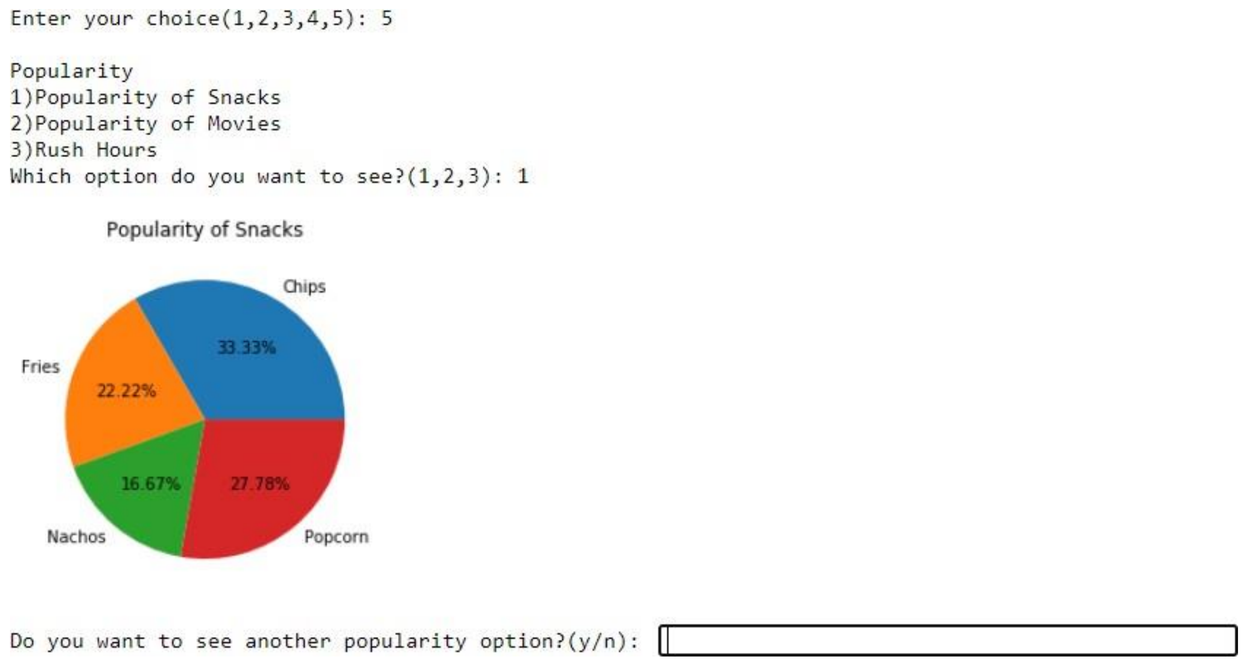
```
Enter your choice(1,2,3,4,5): 4

Cancelling Booking
Enter your Qatar ID number: 23639617932
Enter your movie ID: E1493
Name          :  Aiswarya Arob
Phone Number  :  34128896
Total tickets:  1
Shift         :  Afternoon
Snacks        :  Popcorn
Are you sure you want to cancel the above booking?(y/n): y
Booking Cancelled

Do you want to cancel another booking?(y/n):
```



```
+-------------+---------+-----------------+----------+-------------+-----------+---------+
| QatarID     | MovieID | Name            | PhoneNo  | NoOfTickets | Shift     | Snacks  |
+-------------+---------+-----------------+----------+-------------+-----------+---------+
| 21367492672 | A6041   | Nitish Parab    | 66769906 |           2 | Morning   | Fries   |
| 27689367240 | A3032   | Sreeja Balan    | 66789456 |           2 | Afternoon | Fries   |
| 27803682492 | A3032   | Thomas Mampilli | 34562387 |           2 | Morning   | Nachos  |
| 57829401682 | E1493   | Celina Arob     | 66125537 |           2 | Morning   | Popcorn |
| 28936018478 | A6041   | Rose Jacob      | 44128834 |           1 | Morning   | Nachos  |
| 28467289437 | A6041   | Smriti Rog      | 33892345 |           2 | Afternoon | Popcorn |
| 28951780472 | E4523   | Himansi Arora   | 44562345 |           1 | Night     | Popcorn |
| 28567936782 | E5099   | Dushti Mody     | 68493876 |           2 | Evening   | Chips   |
| 29478346839 | E5099   | Sooniya Ojha    | 33274242 |           2 | Evening   | Chips   |
| 26784567936 | E5099   | Deepak Bora     | 74481818 |           2 | Evening   | Chips   |
+-------------+---------+-----------------+----------+-------------+-----------+---------+
10 rows in set (0.00 sec)
```

Choice 5 – Popularity

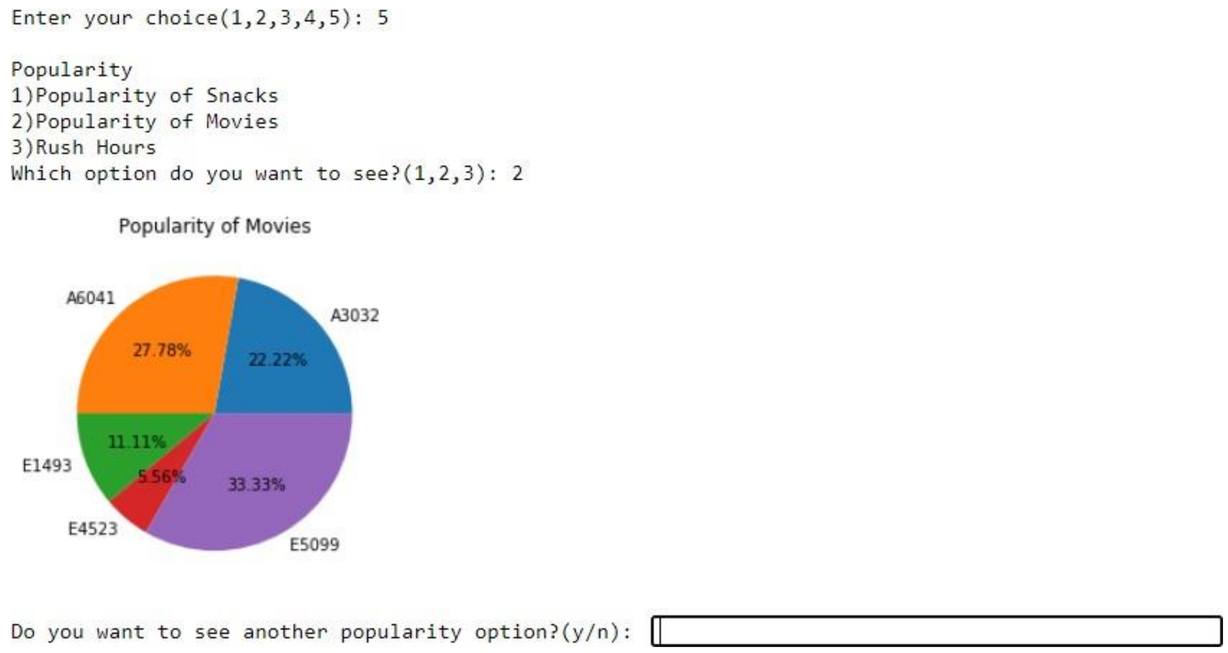
Option 1 – Popularity of Snacks



```
Enter your choice(1,2,3,4,5): 5

Popularity
1)Popularity of Snacks
2)Popularity of Movies
3)Rush Hours
Which option do you want to see?(1,2,3): 1
```

```
Do you want to see another popularity option?(y/n):
```

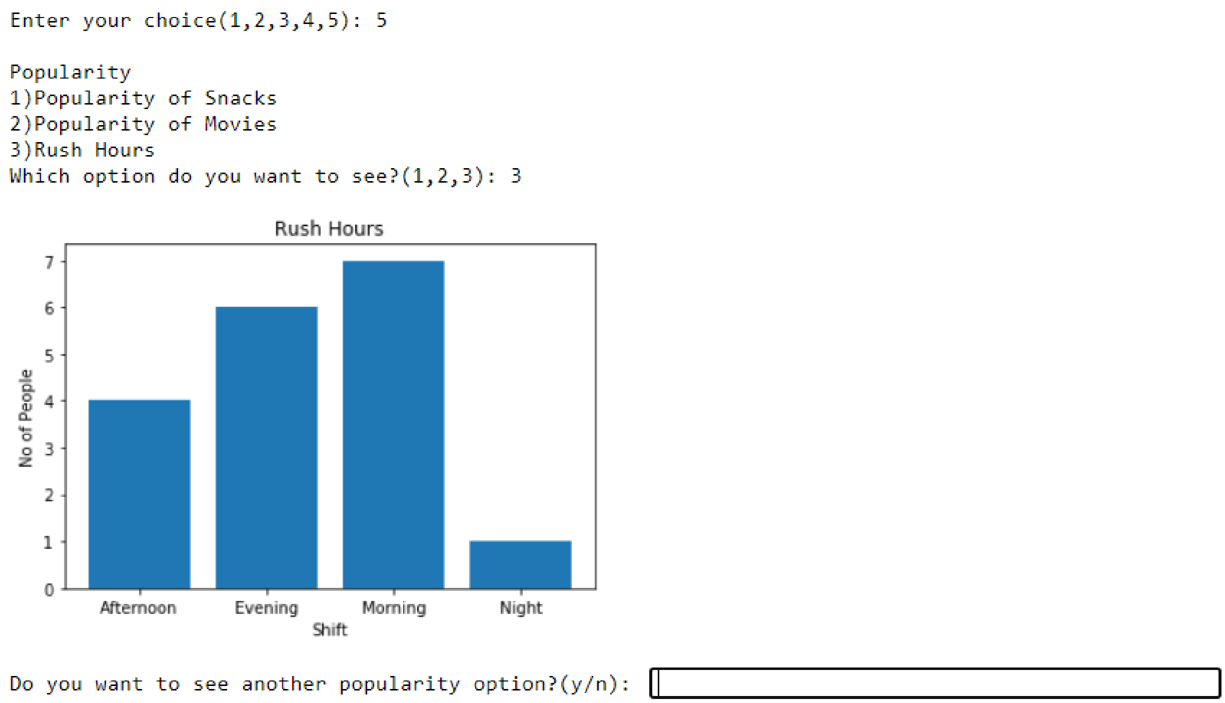
Option 2 – Popularity of Movies



```
Enter your choice(1,2,3,4,5): 5

Popularity
1)Popularity of Snacks
2)Popularity of Movies
3)Rush Hours
Which option do you want to see?(1,2,3): 2
```

```
Do you want to see another popularity option?(y/n):
```

Option 3 – Rush Hours



```
Enter your choice(1,2,3,4,5): 5

Popularity
1)Popularity of Snacks
2)Popularity of Movies
3)Rush Hours
Which option do you want to see?(1,2,3): 3
```

```
Do you want to see another popularity option?(y/n):
```

Bibliography

- Informatics Practices a textbook for class XII -Sumita Arora
- Informatics Practices a textbook for class XI -Sumita Arora
- https://www.irmconnects.com/importance-problem-analysis/
- https://jupyter.org/documentation.html
- https://docs.anaconda.com/ae-notebooks/user-guide/basic-tasks/apps/jupyter/
- https://realpython.com/jupyter-notebook-introduction/