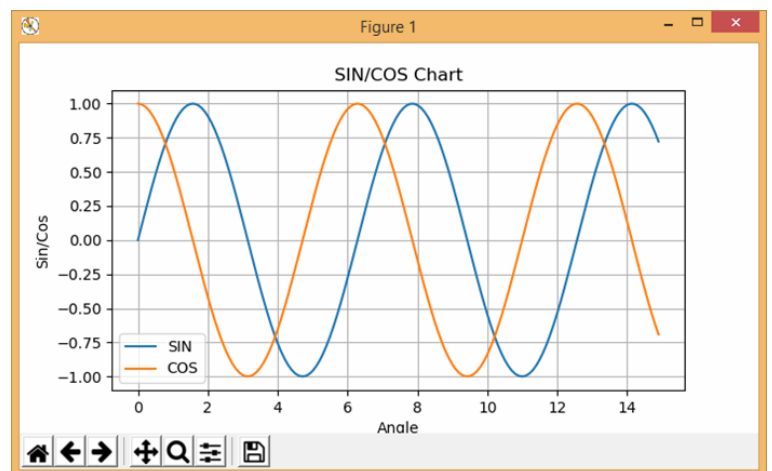




DELHI PRIVATE SCHOOL DUBAI

INFORMATICS PRACTICES PROJECT

```
159 <td class="menuitem" style="background: url(images/wiekze/1.jpg); background-size: 100px 100px; background-position: center; background-repeat: no-repeat; text-align: center; vertical-align: middle; padding: 5px;">
160 </td>
161 </tr>
162 </table>
163 </div>
164 </body>
165 </html>
166 <script type="text/javascript">
167 var currentImage = "bigImage1";
168 var pages = Math.ceil(photos.length / 9);
169 updatePages();
170 updateAllImages();
171 // document.getElementById("bigImage0").src = "images/wiekze/" + photos[(page * 9)];
172 // document.getElementById("bigImage0").style.display = "block";
173 changePhotoDescription('1');
174
175 function updatePages() {
176     var j = 0;
177
178     var html = '<table style="width: 330px;" cellpadding="0" cellspacing="0" border="0"><tr>';
179     if (page != 0) {
180         html = html + '<a href="#" onclick="page=0; updatePages(); updateAllImages();"></a>';
181     }
182     html += '</td><td style="text-align: center;">';
```



Names: Aathif, Mithish,
Tanaya

Class: XII C

Topic: In-flight entertainment system

INDEX

S.No	Topics	Page
1	Acknowledgement	
2	Project Aim	
3	Project Description	
4	Program Source Code	
5	Sample Output	
6	Possible Improvements	
7	Bibliography	

ACKNOWLEDGEMENT

We are overwhelmed in all humbleness and gratefulness to acknowledge our depth to all those who have helped us to put these ideas together. We would sincerely like to thank our school principal Mrs. Rashmi Nandkeolyar and teachers for giving us the support to research on the given topic. It would not have been possible without the kind help and constant support of many individuals such as our teacher Ms. Almas Begum. We respect and thank her, for providing us with an opportunity to do the project work and giving us all the support and guidance, which helped us to complete the project duly. We are extremely thankful to her for providing such nice support and guidance.

We have enjoyed working on this project. It has also helped us increase our general knowledge and research skills. The completion of this project could not have been accomplished without the support of our classmates and teachers.

We would also like to express our gratitude towards our parents for their kind cooperation and encouragement which helped us to the fulfilment of this project. We would like to express our special gratitude to them for giving us such kind attention and time. Our appreciation also goes to our friends in the development of this project.

Project Aim

To incorporate the management of data using python modules of pandas, matplotlib while importing data from MySQL and CSV files to create a menu-driven program that is informative, real-life applicable and user-friendly.

Project Description

Our project is an in-flight entertainment system in which the user can select from a wide array of options to get the song or movie which they prefer, based on the airline and the route they are currently on. There are around 50 options from which the user can select from.

To make the airline database, we had to create the SQL database which contains airline and route information. This was done through our prior knowledge of SQL which we had learnt in classes 11 and 12. We had used several data types in the SQL database which includes date and math functions. Similarly, we had created the databases containing the information for movies and songs in the same way. After the database creation was complete, we had exported the data into a csv file which would be used when the user runs the program.

The user would first select their airline information based on their preference. On selecting an option from a menu, the program will input the CSV file and through several operations using python pandas and matplotlib, they would be able to view the output.

On selecting their preferred airline and route, the interface would prompt the user to select a movie which they prefer based on the airline they have selected. The program would again input the CSV file and the same process would take place, as seen in the initial case.

Finally, after the movie is selected, the user would be able to select a song title based on their preferences on artists and genre etc, through the aforementioned process.

To make the interface clean and simple, we have incorporated the use of several functions such as the while loop function. The loop would be done until the user is satisfied with their option. We have also added an if function with breaks as well.

Source code:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

print("-----In-Flight entertainment-----")

print("-----Welcome to Dubai airport-----")

pd.set_option('display.max_rows',15)
pd.set_option('display.max_columns',20)
pd.set_option('display.precision',2)
df = pd.read_csv('Mithish_CSV.csv')
df.set_index("Sno",drop=True,inplace=True)
ans="Yes"
while ans=="Y" or ans=="y" or ans=="Yes" or ans=="yes":
    print("")
    print("Airline Database")
    print("Please choose from the available options: ")
    print("1.To print the entire the database")
    print("2.To get information of the dataframe and their datatypes")
    print("3. The specific columns by its heading")
    print("4. The information of a route by the index number")
    print("5.The total number of rows and columns")
    print("6. The details of the first 'n' data")
    print("7. The details of the last 'n' data")
    print("8. The details of the route whose fare is more than a specified amount")
    print("9. The details of the route having highest and lowest rating")
    print("10. Display the headings of the dataframe")
    print("11.Append a row")
    print("12. Terminate a row")
    print("13.To plot a graph between destination and rating")
    print("14.Exit")
    opt=int(input('Enter the option you would like to choose: '))
    if opt==1:
        print("The entire database: ")
        print(df)
    elif opt==2:
        print("The database information is")
        print(df.info)
    elif opt==3:
        print("The available headings are:",df.columns)
        n=int(input("Enter number of columns whoose details must be displayed: "))
        for i in range(n):
            heading=input("Enter the column heading: ")
```

```

        print(df.loc[:, [heading]])
elif opt==4:
    y=int(input("Enter the index"))
    x=y-1
    yx=df.iloc[[x]]
    print("The ",y," route is")
    print(yx)

elif opt==5:
    print("The total number of rows and columns are: ")
    (a,b)=(df.shape)
    print("The total number of rows is",a)
    print("The total number of columns is",b)
elif opt==6:
    h=int(input("Enter the number of routes "))
    print(df.head(h))
elif opt==7:
    t=int(input("Enter the number of routes "))
    print(df.tail(t))
elif opt==8:
    r_df=df[["Airline","CostInAED"]]
    a=int(input("Enter the amount"))
    res=r_df[(r_df["CostInAED"]>=a)]
    print("The routes whose fare is more than ",a)
    print(res)
elif opt==9:
    print("Highest rating")
    maxi=df["Rating"].max()
    print(df.loc[df.Rating==maxi,:])
    print("Lowest Rating")
    mini=df["Rating"].min()
    print(df.loc[df.Rating==mini,:])
elif opt==10:
    print("The headings are")
    print(df.columns)
elif opt==11:
    print("The options for the new row are: ")
    x={'Sno':21,'Airline':'Delta','FlightNo':410,'Origin':'Dubai DXB','Destination':'Atlanta',
      'DepartureTime':130,'ArrivalTime':2130,'Aircraft':777,'Duration':1400,'MaxPax':300,
      'CostInAED':5500,'Rating':2.6,'IFE':'Y'}
    y={'Sno':21,'Airline':'Air New Zealand','FlightNo':30,'Origin':'Dubai
DXB','Destination':'Auckland',
      'DepartureTime':130,'ArrivalTime':1530,'Aircraft':777,'Duration':1700,'MaxPax':300,
      'CostInAED':5500,'Rating':4.5,'IFE':'Y'}
    print("The options are:")
    print(x)
    print(y)
    op=input("Enter x or y: ")

```

```

if op=="x":
    "sa=input("Enter which row you want to add, a or b: ")
    df=df.append(a,ignore_index=True)
    print("The dataframe is: ",df)
elif op=="y":
    df=df.append(b,ignore_index=True)
    print("The dataframe is: ",df)
else:
    print("Enter valid option")
elif opt==12:
    r=int(input("Enter the index of the row you want to delete"))
    df.drop(df.index[r])
    print(df.loc[0:r,:])
elif opt==13:
    dest=df.Destination
    rate=df.Rating
    plt.plot(dest,rate)
    plt.xlabel('Destination',fontsize=11)
    plt.ylabel('Rating',fontsize=11)
    plt.xticks(rotation=45)
    plt.yticks(rotation=90)
    plt.show()

elif opt==14:
    break
print("-----Thank you for checking the Flight info-----")
answ="yes"
while answ=="yes":
    op=input("Would like to check the In-Flight Movie(1)/Music(2) options: ")
    if op=="1" or op=="Movie":
        pd.set_option('display.precision',2)
        df = pd.read_csv('Aathif_CSV.csv')
        pd.set_option('display.max_rows',15)
        pd.set_option('display.max_columns',20)
        pd.set_option('display.precision',2)
        df.set_index("Sno",drop=True, inplace=True)
        ans="Yes"
        while ans=="Yes" or ans=="yes" or ans=="y":
            print("")
            print("----- Welcome To AMOVIES -----")
            print("-----")
            print("These are the available options:")
            print("1)To check the names of all the information by giving heading of the movies")
            print("2)Dataframe and Datatypes")
            print("3)The info of a movie by its index number")
            print("4)Shape of dataframe")
            print("5)The details of specific columns")
            print("6)Movie info based on Title")

```

```

print("7)The details of the first n movies")
print("8)The details of the last n movies")
print("9)The details of the movies with highest and lowest vote average")
print("10)Sorting the dataframe based on Revenue")
print("11)To create a smaller dataframe based on a specific column")
print("12)To sort the movies")
print("13)Retrive the details of movies whoose runtime is in a specific range")
print("14)Retrive a list of all the movies")
print("15)To Append a new row")
print("16)To terminate a row")
print("17)Retrive numeric values of the coloumns")
print("18)Visualisation of Data")
print("19)To print the entire the database")
ot=int(input('Enter the chosen task by giving its index number '))
if ot==1:
    print("The names of the headings")
    print(df.columns)
elif ot==2:
    print("The information about the database")
    print(df.info)
elif ot==3:
    x=int(input("Enter the index "))
    a=x-1
    x_movie=df.iloc[[a]]
    print("The ",x," movie is")
    print(x_movie)
elif ot==4:
    print("The total number of rows and columns are: ")
    (a,b)=(df.shape)
    print("The total number of rows is",a)
    print("The total number of columns is",b)
elif ot==5:
    print("The available headings are:",df.columns)
    n=int(input("Enter number of headings whoose,details must be displayed "))
    for i in range(n):
        hd=input("Enter the heading ")
        print(df.loc[:,[hd]])
elif ot==6:
    print("The available movies are")
    print(df["Title"])
    df = df.set_index('Title')
    t=input("Enter the Title of the movie ")
    result = df.loc[t]
    print("Details of the movie ")
    print(result)
elif ot==7:
    f=int(input("Enter the number of movies "))
    print(df.head(f))

```



```

elif ot==8:
    l=int(input("Enter the number of movies "))
    print(df.tail(l))
elif ot==9:
    print("Highest vote average")
    mx=df["Vote_average"].max()
    print(df.loc[df.Vote_average==mx,:])
    print("Lowest vote avg")
    mn=df["Vote_average"].min()
    print(df.loc[df.Vote_average==mn,:])
elif ot==10:
    s_df=df[["Title","Budget"]]
    x=int(input("Enter the ammount more "))
    result=s_df[(s_df["Budget"]>=x)]
    print("The movies whoose budget is more than ",x)
    print(result)
elif ot==11:
    print("The available coloumns are: ")
    print(df.columns)
    s=int(input("Enter the number of columns "))
    l=[]
    for i in range(s):
        col=input("Enter the name of the coloumns ")
        l.append(col)
    x_df=df[l]
    print("The dataframe is")
    print(x_df)
elif ot==12:
    small_df=df[["Title","Language","Budget","Runtime"]]
    h=input("Do you want to sort the data based on revenue or runtime ")
    o=input("Do you want to arrange it in Increasing(1) or Decreasing(2) enter the respective
number ")
    if h=="revenue":
        if o=="1":
            rez1=small_df.sort_values(by="Budget",ascending=True)
            print(rez1)
        elif o=="2":
            rez2=small_df.sort_values(by="Budget",ascending=False)
            print(rez2)
    elif h=="runtime":
        if o=="1":
            rez3=small_df.sort_values(by="Runtime",ascending=True)
            print(rez3)
        elif o=="2":
            rez4=small_df.sort_values(by="Runtime",ascending=False)
            print(rez4)
    else:
        print("Enter a valid value")

```

```

elif ot==13:
    small=df[["Title","Language","Revenue","Runtime","Popularity"]]
    print("Please ensure the range is between 50 and 192")
    b=int(input("Enter the minimum range for the runtime "))
    a=int(input("Enter the maximum range for the runtime "))
    rex=small[(small["Runtime"]>=b) & (small["Runtime"]<=a)]
    print(rex)
elif ot==14:
    print("The movies are: ")
    print(df["Title"])
elif ot==15:
    print("the options of new row are: ")
    a={"Title": "Love
Simon","Budget":15000000,"Genres":"Comedy,Romance","ID":435,"Popularity":18.9478,"Status":"
Released","Language":"en","Runtime":132,"Vote_average":8.8}
    b={"Title":"Alex Strangelove"
,"Budget":1000000,"Genres":"Comedy,Romance","ID":2345,"Popularity":16.989,"Status":"Released"
,"Language":"en","Runtime":132,"Vote_average":8.7}
    c={"Title":"Blue Is the Warmest
Colour","Budget":9000000,"Genres":"Comedy,Romance","ID":8745,"Popularity":18.867,"Status":"R
eleased","Language":"en","Runtime":132,"Vote_average":7.5}
    print("Available options are: ")
    print(a)
    print("")
    print(b)
    print("")
    print(c)
    print("")
    sd=input("Enter the option to be added: ")
    if sd=="a":
        df=df.append(a,ignore_index=True)
    elif sd=="b":
        df=df.append(b,ignore_index=True)
    elif sd=="c":
        df=df.append(c,ignore_index=True)
    else:
        print("Enter the valid value")
    print(df)
elif ot==16:
    gh=int(input("Index of row you want to drop"))
    df.drop(df.index[gh])
    print(df.loc[0:gh,:])
elif ot==17:
    small_df=df[["Title","Budget","Runtime","Vote_average"]]
    print(small_df.describe())
elif ot==18:
    cntr=(df["Production_Country"])
    count_us=0

```

```

count_uk=0
count_eu=0
count_au=0
count_ca=0
for i in range(0,25):
    c=cntr.iloc[i]
    if c=="United States":
        count_us=count_us+1
    elif c=="United Kingdom":
        count_uk=count_uk+1
    elif c=="Australia":
        count_au=count_au+1
    elif c=="Europe":
        count_eu=count_eu+1
    elif c=="Canada":
        count_ca=count_ca+1
x=["United States","United Kingdom","Australia","Europe","Canada"]
y=[count_us,count_uk,count_au,count_eu,count_ca]
plt.bar(x,y,color=["red","blue","green","cyan","orange"],width=0.25)
plt.title("Movie Production Map")
plt.xlabel("Countries")
plt.ylabel("Movies Produced")
plt.show()
elif ot==19:
    print("The entire database:")
    print(df)
ans=input("Do you want to continue ")
if ans=="yes":
    ans="Yes"
elif ans=="Yes":
    ans="Yes"
elif ans=="y":
    ans="Yes"
elif ans=="No":
    break
elif ans=="no":
    break
else:
    print("Enter a valid value")
    ans=input("Do you want to continue")
print("-----Thank you, Have a nice
day-----")
elif op=="2" or op=="Music":
    pd.set_option('display.max_rows',15)
    pd.set_option('display.max_columns',20)
    pd.set_option('display.precision',2)
    df = pd.read_csv("Tanaya_CSV.csv")
    df.set_index("Sno",drop=True, inplace=True)

```

```

ans="Yes"
while ans=="Y" or ans=="y" or ans=="Yes" or ans=="yes":
    print("")
    print("Welcome To SONGS PLAYLIST ")
    print("Please choose from the available options: ")
    print("1)Print the entire the database")
    print("2)Get information of the dataframe and their datatypes")
    print("3)Display the specific columns by its heading")
    print("4)Display the details of a song by its name")
    print("5)Display the information of a particular song by the index number")
    print("6)Find the total number of rows and columns")
    print("7)Display the details of the first 'n' songs")
    print("8)Display the details of the last 'n' songs")
    print("9)Display the details of the song whose revenue generated is more than a specified
amount")
    print("10)Display the details of the songs having highest and lowest vote average")
    print("11)Display all the headings of the dataframe")
    print("12)Sort the songs in ascending order")
    print("13)Sort the songs in descending order")
    print("14)Display a list of all the songs")
    print("15)Add a new row")
    print("16)Delete a row")
    print("17)Use the describe function on the columns")
    print("18)Draw a bar graph based on the production country")
    opt=int(input('Enter the option you would like to choose: '))
    if opt==1:
        print("The entire database: ")
        print(df)
    elif opt==2:
        print("The database information is")
        print(df.info)
    elif opt==3:
        print("The available headings are:",df.columns)
        n=int(input("Enter number of columns whose details must be displayed: "))
        for i in range(n):
            heading=input("Enter the column heading: ")
            print(df.loc[:,[heading]])
    elif opt==4:
        print("The available songs are")
        print(df["Title"])
        df=df.set_index('Title')
        t=input("Enter the Title of the song: ")
        res=df.loc[t]
        print("Details of the song: ")
        print(res)
    elif opt==5:
        y=int(input("Enter the index"))
        x=y-1

```

```

y_song=df.iloc[[x]]
print("The ",y," song is")
print(y_song)

elif opt==6:
    print("The total number of rows and columns are: ")
    (a,b)=(df.shape)
    print("The total number of rows is",a)
    print("The total number of columns is",b)
elif opt==7:
    h=int(input("Enter the number of movies "))
    print(df.head(h))
elif opt==8:
    t=int(input("Enter the number of movies "))
    print(df.tail(t))
elif opt==9:
    r_df=df[["Title","Revenue"]]
    a=int(input("Enter the amount"))
    res=r_df[r_df["Revenue"]>=a]
    print("The movies whose revenue is more than ",a)
    print(res)
elif opt==10:
    print("Highest vote average")
    maxi=df["Vote_average"].max()
    print(df.loc[df.Vote_average==maxi,:])
    print("Lowest vote avg")
    mini=df["Vote_average"].min()
    print(df.loc[df.Vote_average==mini,:])
elif opt==11:
    print("The headings are")
    print(df.columns)
elif opt==12:
    a=input("The column to be sorted: ")
    x_df=df.sort_values(by=a,ascending=True)
    print(x_df)
elif opt==13:
    b=input("The column to be sorted: ")
    y_df=df.sort_values(by=b,ascending=False)
    print(y_df)
elif opt==14:
    print("All the movies are: ")
    print(df["Title"])
elif opt==15:
    print("The options for the new row are: ")
    a={"Index":16,"Title":"Watermelon Sugar","Genres":["{'id': 39, 'name':
'Pop'}]]","ID":4985,"Language":"en",

```

"Overview": "Watermelon Sugar is a song by English singer Harry Styles from his second studio album Fine Line (2019), included as the album's second track. Styles wrote the song with Mitch Rowland and its producers Tyler Johnson and Kid Harpoon.",

"Producer": "[{'name': 'Tyler Johnson and Kid Harpoon', 'id': 915}]", "Production_countries": "United states of america",

"Release_date": "05/15/2020", "Revenue": 483700000, "Runtime": 189, "Spoken_languages": "[{'iso_639_1': 'en', 'name': 'English'}]",

"Status": "Released", "Vote_average": 8.8}

b = {"Index": 16, "Title": "Imagine", "Genres": "[{'id': 10, 'name': 'Rock'}]", "ID": 47447, "Language": "en",

"Overview": "Imagine is a song by English rock musician John Lennon from his 1971 album of the same name.",

"Producer": "[{'name': 'John Lennon and Yoko Ono', 'id': 909}]", "Production_countries": "United states of america",

"Release_date": "10/11/1971", "Revenue": 45687000, "Runtime": 183, "Spoken_languages": "[{'iso_639_1': 'en', 'name': 'English'}]",

"Status": "Released", "Vote_average": 8.7}

```
print("The options are: ")
```

```
print(a)
```

```
print(b)
```

```
xp=input("Enter which row you want to add, a or b: ")
```

```
if xp=="a":
```

```
    df=df.append(a,ignore_index=True)
```

```
    print("The dataframe is: ",df)
```

```
elif xp=="b":
```

```
    df=df.append(b,ignore_index=True)
```

```
    print("The dataframe is: ",df)
```

```
else:
```

```
    print("Enter valid option")
```

```
elif opt==16:
```

```
    r=int(input("Enter the index of the row you want to delete"))
```

```
    df.drop(df.index[r])
```

```
    print(df.loc[0: r,:])
```

```
elif opt==17:
```

```
    print(df.describe())
```

```
elif opt==18:
```

```
    Production_countries=["Jamaica","Europe","UK","France","USA"]
```

```
    Y=[1,1,1,2,15]
```

```
    plt.bar(Production_countries,Y,color=["red","black","green","cyan","blue"],width=0.25)
```

```
    plt.title("Bar Graph")
```

```
    plt.xlabel("Country")
```

```
    plt.ylabel("Number of Songs")
```

```
    plt.show()
```

```
ans=input("Do you want to continue ")
```

```
if ans=="yes":
```

```
    ans="Yes"
```

```

elif ans=="Yes":
    ans="Yes"
elif ans=="y":
    ans=="Yes"
elif ans=="Y":
    ans=="Yes"
elif ans=="No":
    break
elif ans=="no":
    break
elif ans=="N":
    break
elif ans=="n":
    break
else:
    print("Enter a valid value")
    ans=input("Do you want to continue")
print("-----Thank you, Have a nice
day-----")
else:
    print("Enter a valid input")
    op=input("Would like to check the In-Flight Movie(1)/Music(2) options")
    qe=input("Would you like to continue: ")
    if qe=="yes" or qe=="y" or qe=="Yes":
        answ="yes"
    else:
        break
print("-----Thank you,Hope to see you
again-----")

```

Program Output:

-----In-Flight entertainment-----
-----Welcome to MAT airways-----

Airline Database

Please choose from the available options:

- 1.To print the entire the database
- 2.To get information of the dataframe and their datatypes
3. The specific columns by its heading
4. The information of a route by the index number
- 5.The total number of rows and columns
6. The details of the first 'n' data
7. The details of the last 'n' data
8. The details of the route whose fare is more than a specified amount
9. The details of the route having highest and lowest rating
10. Display the headings of the dataframe
- 11.Append a row
12. Terminate a row
- 13.To plot a graph between destination and rating
- 14.Exit

Enter the option you would like to choose: 6

Enter the number of routes 2

	Airline	FlightNo	Origin	Destination	DepartureTime \	
Sno						
1	Emirates	1	Dubai DXB	London Heathrow LHR		745
2	Emirates	201	Dubai DXB	New York JFK		845

	ArrivalTime	Aircraft	Duration	MaxPax	CostInAED	Rating	IFE
Sno							
1	1225	A380	700	489	3000	4.3	Y
2	1530	A380	1450	489	5500	3.8	Y

9. The details of the route having highest and lowest rating
10. Display the headings of the dataframe
11. Append a row
12. Terminate a row
13. To plot a graph between destination and rating
14. Exit

Enter the option you would like to choose: 10

The headings are

```
Index(['Airline', 'FlightNo', 'Origin', 'Destination', 'DepartureTime',  
      'ArrivalTime', 'Aircraft', 'Duration', 'MaxPax', 'CostInAED', 'Rating',  
      'IFE'],  
      dtype='object')
```

Airline Database

Please choose from the available options:

1. To print the entire the database
2. To get information of the dataframe and their datatypes
3. The specific columns by its heading
4. The information of a route by the index number
5. The total number of rows and columns
6. The details of the first 'n' data
7. The details of the last 'n' data
8. The details of the route whose fare is more than a specified amount
9. The details of the route having highest and lowest rating
10. Display the headings of the dataframe
11. Append a row
12. Terminate a row
13. To plot a graph between destination and rating
14. Exit

Enter the option you would like to choose: 14

-----Thank you for checking the Flight info-----

-----Thank you for checking the Flight info-----
Would like to check the In-Flight Movie(1)/Music(2) options: 2

Welcome To SONGS PLAYLIST

Please choose from the available options:

- 1)Print the entire the database
 - 2)Get information of the dataframe and their datatypes
 - 3)Display the specific columns by its heading
 - 4)Display the details of a song by its name
 - 5)Display the information of a particular song by the index number
 - 6)Find the total number of rows and columns
 - 7)Display the details of the first 'n' songs
 - 8)Display the details of the last 'n' songs
 - 9)Display the details of the song whose revenue generated is more than a specified amount
 - 10)Display the details of the songs having highest and lowest vote average
 - 11)Display all the headings of the dataframe
 - 12)Sort the songs in ascending order
 - 13)Sort the songs in descending order
 - 14)Display a list of all the songs
 - 15)Add a new row
 - 16)Delete a row
 - 17)Use the describe function on the columns
 - 18)Draw a bar graph based on the production country
- Enter the option you would like to choose: 14

All the movies are:

Sno

- | | |
|---|------------------|
| 1 | Abilene |
| 2 | Ace of Spades |
| 3 | Above and Beyond |
| 4 | Alive |
| 5 | All Blues |

...

- | | |
|----|-----------|
| 16 | New Rules |
| 17 | One Dance |
| 18 | Perfect |
| 19 | Rockabye |
| 20 | Stiches |

Name: Title, Length: 20, dtype: object

Do you want to continue yes

Welcome To SONGS PLAYLIST

Please choose from the available options:

- 1)Print the entire the database
 - 2)Get information of the dataframe and their datatypes
 - 3)Display the specific columns by its heading
 - 4)Display the details of a song by its name
 - 5)Display the information of a particular song by the index number
 - 6)Find the total number of rows and columns
 - 7)Display the details of the first 'n' songs
 - 8)Display the details of the last 'n' songs
 - 9)Display the details of the song whoose revenue generated is more than a specified amount
 - 10)Display the details of the songs having highest and lowest vote average
 - 11)Display all the headings of the dataframe
 - 12)Sort the songs in ascending order
 - 13)Sort the songs in descending order
 - 14)Display a list of all the songs
 - 15)Add a new row
 - 16)Delete a row
 - 17)Use the describe function on the coloumns
 - 18)Draw a bar graph based on the production country
- Enter the option you would like to choose: 18

Would like to check the In-Flight Movie(1)/Music(2) options: 1

----- Welcome To AMOVIES -----

These are the available options:

- 1)To check the names of all the information by giving heading of the movies
 - 2)Dataframe and Datatypes
 - 3)The info of a movie by its index number
 - 4)Shape of dataframe
 - 5)The details of specific columns
 - 6)Movie info based on Title
 - 7)The details of the first n movies
 - 8)The details of the last n movies
 - 9)The details of the movies with highest and lowest vote average
 - 10)Sorting the dataframe based on Revenue
 - 11)To create a smaller dataframe based on a specific column
 - 12)To sort the movies
 - 13)Retrieve the details of movies whoose runtime is in a specific range
 - 14)Retrieve a list of all the movies
 - 15)To Append a new row
 - 16)To terminate a row
 - 17)Retrieve numeric values of the coloumns
 - 18)Visualisation of Data
 - 19)To print the entire the database
- Enter the chosen task by giving its index number 4

The total number of rows and columns are:

The total number of rows is 25

The total number of columns is 11

Do you want to continue yes

----- Welcome To AMOVIES -----

These are the available options:

- 1)To check the names of all the information by giving heading of the movies
- 2)Dataframe and Datatypes
- 3)The info of a movie by its index number
- 4)Shape of dataframe
- 5)The details of specific columns
- 6)Movie info based on Title
- 7)The details of the first n movies
- 8)The details of the last n movies
- 9)The details of the movies with highest and lowest vote average
- 10)Sorting the dataframe based on Revenue
- 11)To create a smaller dataframe based on a specific column
- 12)To sort the movies
- 13)Retrieve the details of movies whose runtime is in a specific range
- 14)Retrieve a list of all the movies
- 15)To Append a new row
- 16)To terminate a row
- 17)Retrieve numeric values of the columns
- 18)Visualisation of Data
- 19)To print the entire the database

Enter the chosen task by giving its index number 9

Highest vote average

	Title	Budget	Genres	ID	Popularity	Status	Language \
Sno							
16	Casino	5.20e+07	Action,Thriller	4584.0	7.28	Released	en

	Vote_average	Runtime	Date_of_Production	Production_Country
Sno				

16	7.8	178	1994-10-15	Canada
----	-----	-----	------------	--------

Lowest vote avg

	Title	Budget	Genres	ID	Popularity	\
Sno						
8	Tom and Huck	7.80e+05	Action,Drama,Family	45325.0	2.56	
20	Money Train	6.00e+07	Action,Adventure	8012.0	7.34	

	Status	Language	Vote_average	Runtime	Date_of_Production	\
Sno						
8	Released	en	5.4	97	2006-03-12	
20	Released	en	5.4	103	1997-09-17	

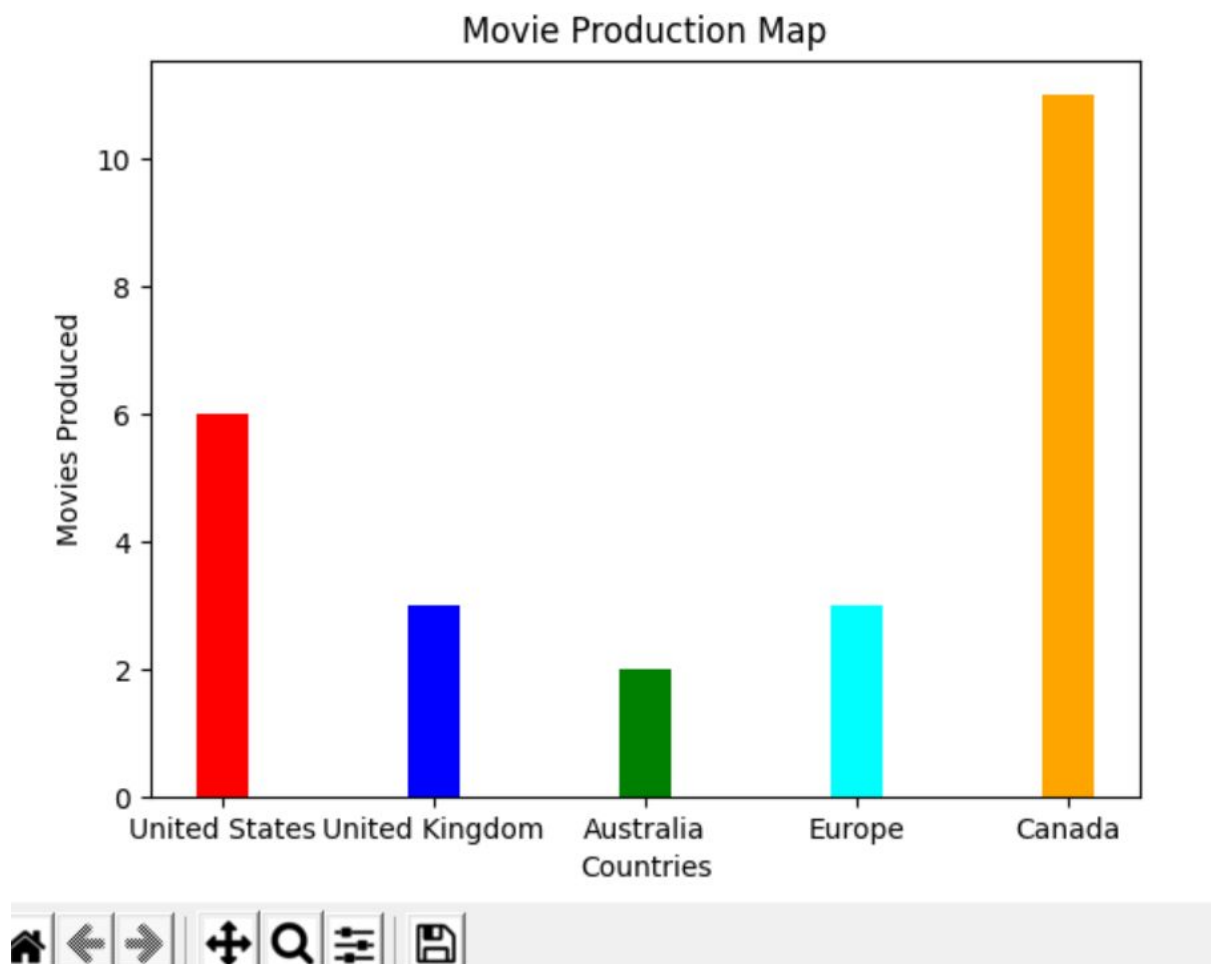
	Production_Country
Sno	
8	United States
20	United Kingdom

Would like to check the In-Flight Movie(1)/Music(2) options: 1

----- Welcome To AMOVIES -----

These are the available options:

- 1)To check the names of all the information by giving heading of the movies
 - 2)Dataframe and Datatypes
 - 3)The info of a movie by its index number
 - 4)Shape of dataframe
 - 5)The details of specific columns
 - 6)Movie info based on Title
 - 7)The details of the first n movies
 - 8)The details of the last n movies
 - 9)The details of the movies with highest and lowest vote average
 - 10)Sorting the dataframe based on Revenue
 - 11)To create a smaller dataframe based on a specific column
 - 12)To sort the movies
 - 13)Retrieve the details of movies whose runtime is in a specific range
 - 14)Retrieve a list of all the movies
 - 15)To Append a new row
 - 16)To terminate a row
 - 17)Retrieve numeric values of the columns
 - 18)Visualisation of Data
 - 19)To print the entire the database
- Enter the chosen task by giving its index number 18



Possible Improvements:

There could be the following improvements in our project:

1. The python code could be written in a more concise manner as it would be easier to comprehend.
2. Comment lines could be incorporated to make the program more understandable by the examiner and user. By doing this, the user would understand what we want to convey through the program and its main idea.
3. The project quality could be enhanced by better connectivity within the program. This would enable the user to operate the program with much ease.
4. The use of a comment box where the user can leave their remarks would be a great help to us. This will indirectly help us as we can constructively use the remarks to polish up our program.
5. Our program could be made visually appealing and attractive with the use of graphics and graphical user interface.

Bibliography:

https://www.w3schools.com/python/python_ml_scatterplot.asp

<https://stackoverflow.com/questions/>

<https://github.com/topics/music-database>

<https://matplotlib.org/>