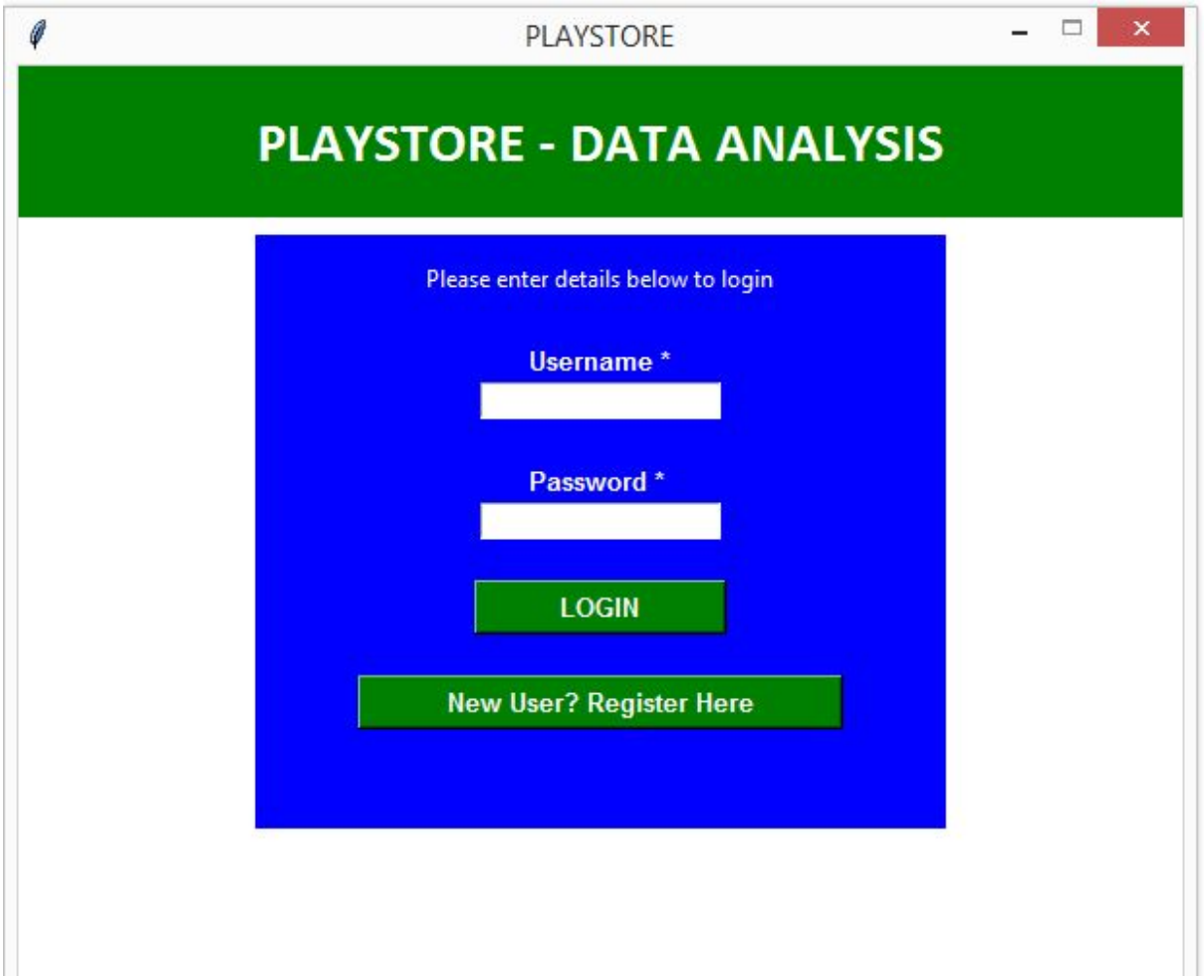
Application Design for an app launch on Google PlayStore



Submitted by : Jayesh Rane
Date of Submission: 10th July 2019.

Submitted to : M - Intellect Global
Under the Guidance of : Junaid Khateeb
(Director, Khateeb Institute of Technical Education)

Certificate Of completion

This is to certify that , Mr Jayesh Rane has successfully implemented an application designed to study the data and generate insights for an app launch on Google PlayStore.

The Application has been accepted as a completed project as it meets all the requirements specified.

12th July 2019

_____
(Khateeb Institute of Technical Education)

ACKNOWLEDGEMENT

We express our deepest thanks to our principal Dr. S. P. Kallurkar ,Atharva College Of Engineering, Mumbai for allowing us to carry out the industrial training and supporting us throughout.
We express our deepest thanks to Ms. Charmi Chaniara and Ms. Samira Nigrel,our python faculties, for taking part in useful decision , guidance and necessary equipment for the project .We choose this moment to acknowledge their contribution gratefully.
We also express our deepest thanks to our HOD Neelima Pathak for allowing us to carry our this project and helped us in all the way so that we could gain a practical experience of the industry.
We also take this opportunity to thank Ms. Snigdha Bangal for guiding us in the right path and being of immense help to us. Finally we thank all other unnamed who helped us in various ways to gain knowledge and have a good training.

Contents

Section 1:System Requirement Specifications.

1.1 Introduction

This project focuses on developing an application which provides the analysis of the play store to the user. The app provides useful data in the graphical format which is easy to understand even for a layman user. The system can be used for calculating number of downloads for each app or a specific category, or which apps have to best reviews and should an app be free or paid in orde to increase its downloads.

1.2 Significance

This software can be helpful to those individuals who want to launch their own apps in the play store. This software can give an insight about which apps are most popular amongst the public so that the user can focus on developing such kinds of apps. Also the user can know which category of apps has the highest ratings in order to know the favourite category. These stats can help an individual
a lot for his/her future app launches or app developement.

1.3 Requirements

Hardware:1. PC/Laptop
2. 2GB RAM
3. 500GB Memory
4. Intel i3 processor

Software: 1.Python 3.6
2.Anaconda Navigator
3.Spyder 3.3.3
4. Wamp Server 64bit 3.17

1.4 Expected Functionalities

Following are some of the functionalities expected by the client:
1) What is the percentage download in each category on the playstore.
2) How many apps have managed to get the following number of downloads
   a) Between 10,000 and 50,000
   b) Between 50,000 and 150000
   c) Between 150000 and 500000
   d) Between 500000 and 5000000
   e) More than 5000000
3) Which category of apps have managed to get the most,least and an average of 2,50,000 downloads atleast.

4) Which category of apps have managed to get the highest maximum average ratings from the users.
5) What is the download trend category wise over the period for which the data is being made available.
6) For the years 2016,2017,2018 what are the category of apps that have got the most and the least downloads. What is the percentage increase or decrease that the apps have got over the period of three years.
7) All those apps , whose android version is not an issue and can work with varying devices ,what is the percentage increase or decrease in the downloads.
8) Amongst sports, entertainment,social media,news,events,travel and games,which is the category of app that is most likely to be downloaded in the coming years, kindly make a prediction and back it with suitable findings.
9) All those apps who have managed to get over 1,00,000 downloads, have they managed to get an average rating of 4.1 and above? can we conclude something in co-relation to the number of downloads and the ratings received.
10) Across all the years ,which month has seen the maximum downloads for each of the category. What is the ratio of downloads for the app that qualifies as teen versus mature 17+

Section 2: Technology Used.

2.1 Python

The syntax rules of Python allow you to express concepts without writing additional code. At the same time, Python, unlike other programming languages, emphasizes on code readability, and allows you to use English keywords instead of punctuations. Hence, you can use Python to build custom applications without writing additional code.In this application 3.6 version of python is used. The readable and clean code base will help you to maintain and update the software without putting extra time and effort.At present, Python is supports many operating systems. You can even use Python interpreters to run the code on specific platforms and tools.Also, its language features support various concepts in functional and aspect-oriented programming. At the same time, Python also features a dynamic type system and automatic memory management. The programming paradigms and language features help you to use Python for developing large and complex software applications.

2.2 Spyder IDE

Spyder is a powerful scientific environment written in Python, for Python, and designed by and for scientists, engineers and data analysts. It offers a unique combination of the advanced editing, analysis, debugging, and profiling functionality of a comprehensive development tool with the data exploration,

interactive execution, deep inspection, and beautiful visualization capabilities of a scientific package.The easy way to get up and running with Spyder on any of our supported platforms is to download it as part of the Anaconda distribution, and use the conda package and environment manager to keep it and your other packages installed and up to date. We recommend the latest 64-bit Python 3 version, unless you have specific requirements that dictate otherwise.

2.3 Wamp Server

WampServer is a utility designed to allows you to create Web applications and manage your server and databases. It allows you to create web applications with Apache2, PHP and a MySQL database. It also comes with PHPMyAdmin and SQLiteManager to easily manage your databases. WampServer installs automatically (installer), and its usage is very intuitive. WampServer is the only packaged solution that will allow you to reproduce your production server. Once WampServer is installed, you have the possibility to add as many Apache, MySQL, and PHP releases as you want. WampServer also has a tray icon to manage your server and its settings.

2.4 Libraries

1. Pandas:
   Pandas is a high-level data manipulation tool developed by Wes McKinney. It is built on the Numpy package and its key data structure is called the DataFrame. DataFrames allow you to store and manipulate tabular data in rows of observations and columns of variables.

2. Matplotlib
   Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack.Matplotlib consists of several plots like line, bar, scatter, histogram etc.

3. Seaborn
   Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

4. Numpy
   NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

5. Tkinter

   Python offers multiple options for developing GUI . Out of all the GUI methods, tkinter is most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter outputs the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.

Section 3: Datasets.

3.1 Observations

From the given datasets we can observe that there are 10,000 plus rows and 13 columns in the first dataset and 60,000 plus rows and 5 columns in the second dataset. Dataset one has the columns named like App,Category,Ratings,Review,Size,Installs,Price,,Type,Content Rating,Genre,Last Update,Current Version and Android Ver. While the second database have the columns like App,Translated Review,Sentiment,Sentiment Polarity and Sentiment Subjectivity.

3.2 Data Wrangling

From the databases we can observe that all the data is not in the proper format to be analyzed. Thus we have to convert the data into proper format for further calculations.The missing data as well as the rows which does not contain values in the proper datatype has been removed.'+' symbols from the 'Installs' column has been removed to make the column values into integers. Also the values in the 'Last Updated 'columns has been converted into date-time format as required. '$' symbol from the 'Price' column has been removed. Duplicate values from the datasets has been removed.Due to this pre processing of data it becomes easier to do the further analysis and prevent any datatype errors.

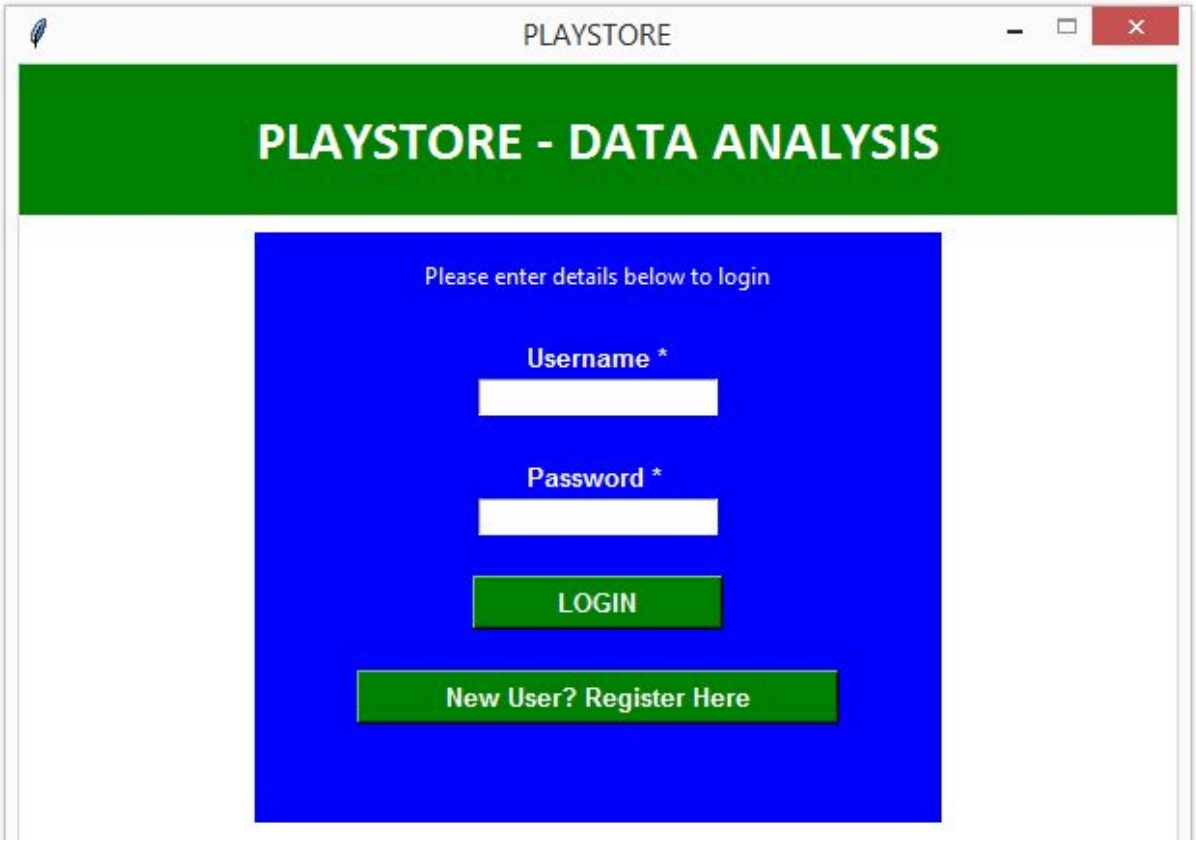
3.3 Database Software

The software used for the creation and maintenance of the database is Wamp server. It is a very easy tool to create and manage databases for the applications.We can easily create tables for our database entries and add columns to it.It also comes with PHPMyAdmin and SQLiteManager to easily manage your databases. WampServer installs automatically (installer), and its usage is very intuitive. WampServer is the only packaged solution that will allow you to reproduce your production server. Once WampServer is installed, you have

the possibility to add as many Apache, MySQL, and PHP releases as you want. WampServer also has a tray icon to manage your server and its settings.

Section 4: Screenshots

1. Main Screen

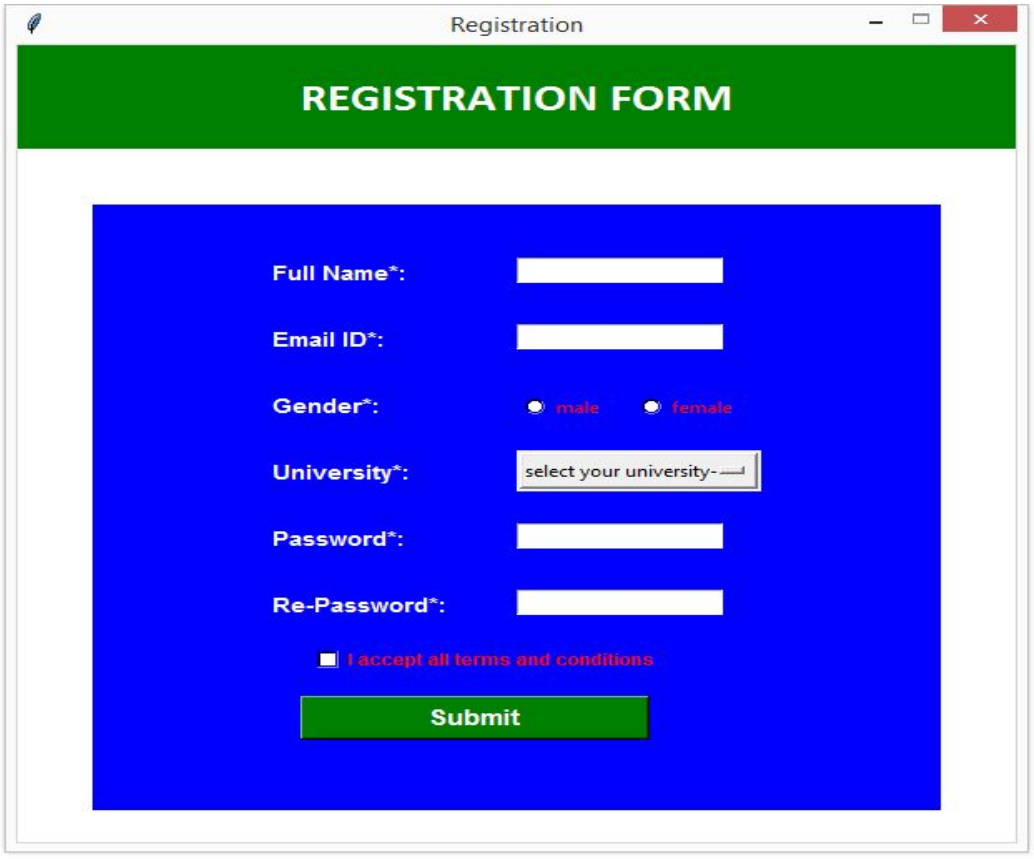


Code:

```
def main_screen():
    global screen, username_verify,password_verify
    screen=Tk()
    username_verify = StringVar()
    password_verify = StringVar()
    screen.title("PLAYSTORE")
    adjustWindow(screen)
                    Label(screen,text="PLAYSTORE - DATA ANALYSIS", width="500",
height="2",font=("Calibri",22,'bold'),fg='white',bg='green').pack()
    Label(text="",bg='white').pack()
    Label(screen, text="", bg='blue',width='50', height='20').place(relx=0.5,rely=0.4,anchor=CENTER)
    Label(screen, text="Please enter details below to login", bg='blue', fg='white').pack()
    Label(screen,text="",bg="blue").pack()
    Label(screen,text="Username *",font=("Open Sans",10,'bold'),bg="blue",fg='white').pack()
    Entry(screen, textvar=username_verify).pack()
    Label(screen, text="", bg='blue').pack()
    Label(screen, text="Password * ", font=("Open Sans", 10, 'bold'), bg='blue', fg='white').pack()
    Entry(screen, textvar=password_verify, show="*").pack()
    Label(screen, text="", bg='blue').pack()
      Button(screen, text="LOGIN", bg="green", width=15, height=1, font=("Open Sans", 10, 'bold'),
fg='white', command=login_verify).pack()
    Label(screen, text="", bg='blue').pack()
```

```
    Button(screen, text="New User? Register Here", height="1", width="30", bg='green', font=("Open Sans", 10, 'bold'), fg='white', command=register ).pack()
    screen.mainloop()
        # Button(screen,text="Exit", command=do_exit,bg="yellow",fg="black",font=("Open Sans", 16,"bold")).place(x=0,y=100)
main_screen()
```

2. Registration Form

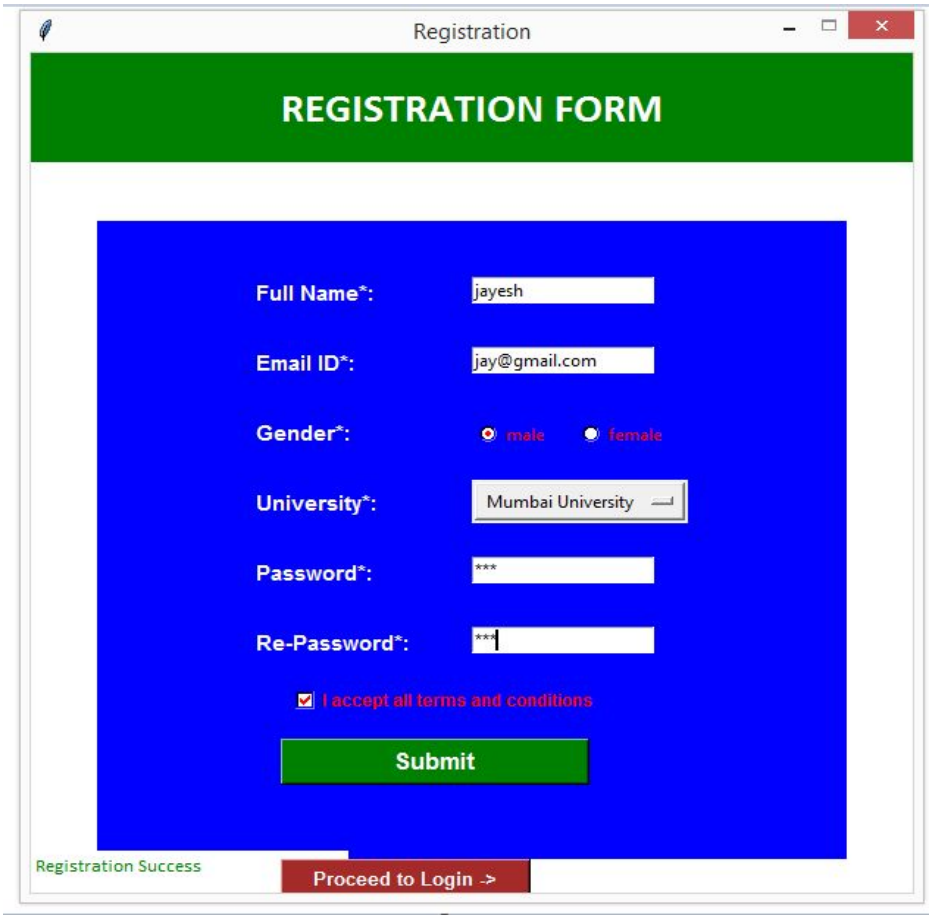


Code:

```
def register():
    global screen1, fullname, email, password, repassword,university,gender,tnc
    fullname = StringVar()
    email = StringVar()
    password = StringVar()
    repassword = StringVar()
    university = StringVar()
    gender = IntVar()
    tnc = IntVar()
    screen1 = Toplevel(screen)
    screen1.title("Registration")
    adjustWindow(screen1)
    Label(screen1, text = "REGISTRATION FORM", width = '50', height='2', font=("Calibri",22,'bold'),fg='white',bg='green').pack()
    Label(screen1,text ="",bg="blue", width='72',height='30').place(x=45, y=120)
    Label(screen1, text="Full Name*:",font=("Open Sans",11,'bold'),fg='white',bg='blue',anchor=W).place(x=150,y=160)
```

```
    Entry(screen1,textvar=fullname).place(x=300,y=160)
    Label(screen1, text="Email ID*:",font=("Open
Sans",11,'bold'),fg='white',bg='blue',anchor=W).place(x=150,y=210)
    Entry(screen1,textvar=email).place(x=300,y=210)
    Label(screen1, text="Gender*:",font=("Open
Sans",11,'bold'),fg='white',bg='blue',anchor=W).place(x=150,y=260)
    Radiobutton(screen1,text="male",variable=gender,value=1,bg='blue',fg='red').place(x=300,y=260)

Radiobutton(screen1,text="female",variable=gender,value=2,bg='blue',fg='red').place(x=370,y=260)
    Label(screen1, text="University*:", font=("Open Sans", 11, 'bold'), fg='white', bg='blue',
anchor=W).place(x=150, y=310)
    list1 = ['Mumbai University', 'Savitribai Phule Pune Univeristy','Gujarat Technological University',
'JNTU Kakinada', 'University of Delhi', 'Anna University']
    droplist = OptionMenu(screen1, university, *list1)
    droplist.config(width=17)
    university.set('--select your university--')
    droplist.place(x=300, y=305)
    Label(screen1, text="Password*:",font=("Open
Sans",11,'bold'),fg='white',bg='blue',anchor=W).place(x=150,y=360)
    Entry(screen1, textvar=password, show="*").place(x=300, y=360)
    Label(screen1, text="Re-Password*:", font=("Open Sans", 11, 'bold'), fg='white', bg='blue',
anchor=W).place(x=150, y=410)
    entry_4 = Entry(screen1, textvar=repassword, show="*")
    entry_4.place(x=300, y=410)
    Checkbutton(screen1, text="I accept all terms and conditions", variable=tnc, bg='blue',
font=("Open Sans", 9, 'bold'), fg='red').place(x=175, y=450)
    Button(screen1, text='Submit', width=20, font=("Open Sans", 13, 'bold'), bg='green',
fg='white',command=register_user).place(x=170, y=490)
```

3. Registration Success

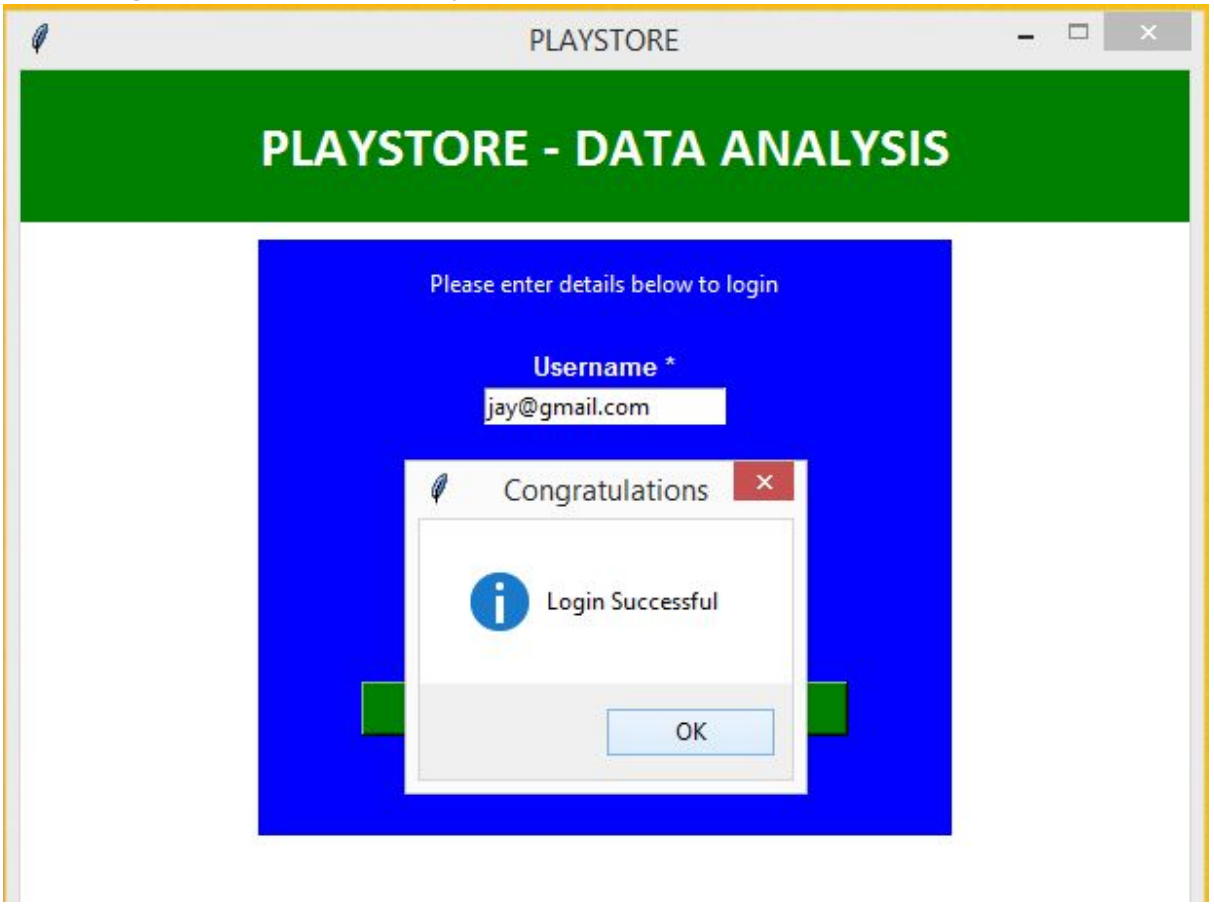


Code:

```
def register_user():
    if (fullname.get() and email.get() and password.get() and repassword.get() and gender.get()):
        if (university.get() == "--select your university--"):
            Label(screen1,text="Please select your
university",fg="red",font=('Calibri',11),width='30',anchor=W,bg='white').place(x=0,y=570)
            return
        else:
            if (tnc.get()):
                if (re.match("^.+@(\[?)[a-zA-Z0-9-.]+.([a-zA-Z]{2,3}|[0-9]{1,3})(]?)$", email.get())):
                    if (password.get() == repassword.get()):
                        gender_value = 'male'
                        if (gender.get()==2):
                            gender_value='female'
                        connection =
pymysql.connect(host='localhost',user='root',passwd="",database='edumate')
                        cursor = connection.cursor()
                        insert_query = "INSERT INTO
student_details(fullname,email,password,gender,university)VALUES('"+ fullname.get() + "', '"+
email.get() + "', '"+ password.get() + "', '"+ gender_value + "', '"+ university.get() + "' );"
                        cursor.execute(insert_query)
                        connection.commit()
                        connection.close()
                        Label(screen1,text="Registration
Success",fg='green',font=('calibri',10),width='30',anchor=W,bg='white').place(x=0,y=570)
```

```
                Button(screen1,text='Proceed to Login ->', width=20,font=('open
sans',10,'bold'),bg='brown',fg='white',command=screen1.destroy).place(x=170,y=576)
            else:
                Label(screen1, text="Password does not match", fg="red", font=("calibri", 11),
width='30', anchor=W, bg='white').place(x=0, y=570)
                return
        else:
            Label(screen1, text="Please enter valid email id", fg="red", font=("calibri", 11),
width='30', anchor=W, bg='white').place(x=0, y=570)
            return
    else:
        Label(screen1, text="Please accept the agreement", fg="red",font=("calibri", 11),
width='30', anchor=W, bg='white').place(x=0, y=570)
        return
else:
    Label(screen1, text="Please fill all the details", fg="red",font=("calibri", 11), width='30',
anchor=W, bg='white').place(x=0, y=570)
    return
```

4. Login Successfully

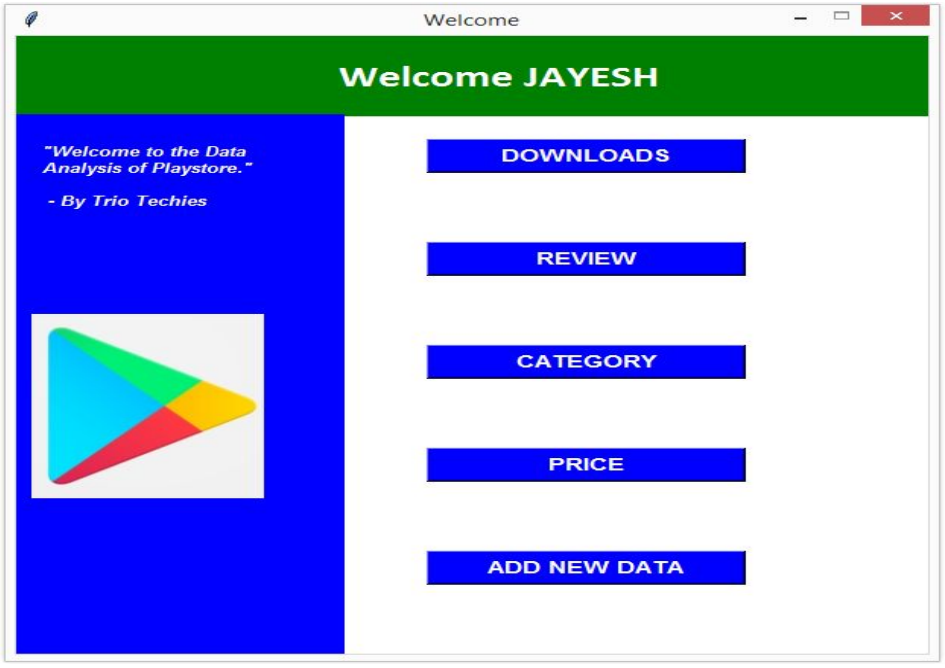


Code:

```
def login_verify():
    global studentID
    connection = pymysql.connect(host='localhost',user='root',passwd="",database='edumate')
    cursor = connection.cursor()
```

```
    select_query = "SELECT * FROM student_details where email = '" + username_verify.get() + "'
AND password = '" + password_verify.get() + "';"
    cursor.execute(select_query)
    student_info = cursor.fetchall()
    connection.commit()
    connection.close()
    if student_info:
        messagebox.showinfo("Congratulations","Login Successful")
        studentID = student_info[0][0]
        welcome_page(student_info)
    else:
        messagebox.showerror("Error","Invalid username or password")
```

5. Welcome Screen



Code:

```
def welcome_page(student_info):
    global screen2
    screen2 = Toplevel(screen)
    screen2.title("Welcome")
    adjustWindow(screen2)
    Label(screen2, text="Welcome " +student_info[0][1], width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen2, text="",bg="blue",width='30',height='50').place(x=0,y=76)
    Message(screen2, text='"Welcome to the Data Analysis of Playstore."\n\n - By Trio
Techies',width='180', font=("Helvetica",10, 'bold', 'italic'), fg='white', bg='blue', anchor =
CENTER).place(x=10, y=100)
    photo = PhotoImage(file="play.png") # opening left side image - Note: If image is in same
folderthen no need to mention the full path
    label = Label(screen2, image=photo ,text="") # attaching image to the label
    label.place(x=10, y=270)
    label.image = photo
```

```
  # it is necessary in Tkinter to keep a instance of image to displayimage in labe
    #label1 = Label(screen2, text="") # attaching image to the label
    #label1.place(x=200, y=78)
    Button(screen2, text='DOWNLOADS', width=20, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=analysis1).place(x=270, y=100)
    Button(screen2, text='REVIEW', width=20, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=analysis2).place(x=270, y=200)
    Button(screen2, text='CATEGORY', width=20, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=analysis3).place(x=270, y=300)
    Button(screen2, text='PRICE', width=20, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=analysis4).place(x=270, y=400)
    Button(screen2, text='ADD NEW DATA', width=20, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=analysis7).place(x=270, y=500)
    #Button(screen2,text="Exit", command=do_exit,bg="yellow",fg="black",font=("Open Sans",
16,"bold")).place(x=0,y=100)
```

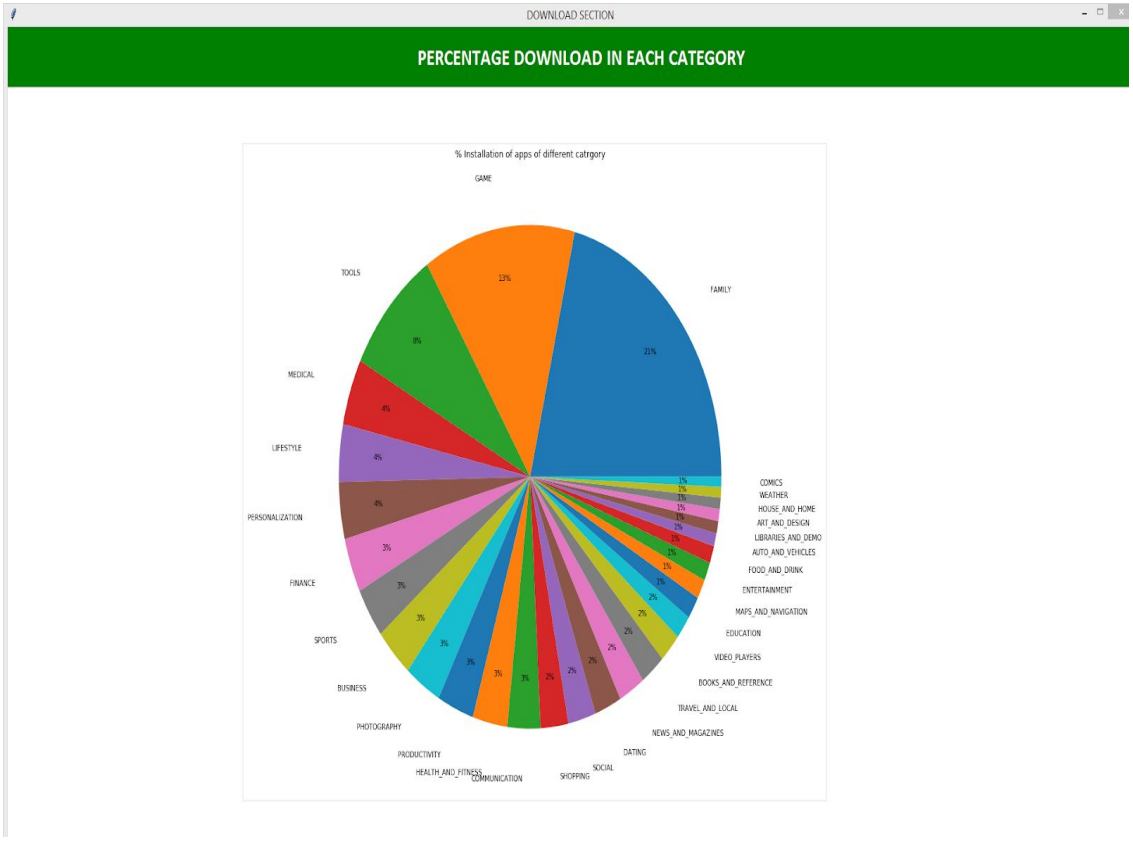
6. Downloads

Code:

```
def analysis1():
    global screen3
    screen3=Toplevel(screen)
    screen3.title("DOWNLOAD SECTION")
    adjustWindow(screen3)
    Label(screen3, text="EVERYTHING ABOUT DOWNLOADS", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Button(screen3, text='PERCENTAGE DOWNLOAD IN EACH CATEGORY', width=50,
font=("Open Sans", 13, 'bold'),bg='blue', fg='white',command=percentage_download).place(x=50,
y=100)
```

```
    Button(screen3, text='NO OF DOWNLOADS', width=50, font=("Open Sans", 13, 'bold'),bg='blue', fg='white',command=downloads).place(x=50, y=150)
    Button(screen3, text='DOWNLOAD TREND CATEGORY WISE', width=50, font=("Open Sans", 13, 'bold'),bg='blue', fg='white',command=download_category).place(x=50, y=200)
    Button(screen3, text='DOWNLOAD VS RATING', width=50, font=("Open Sans", 13, 'bold'),bg='blue', fg='white',command=analysis5).place(x=50, y=250)
    Button(screen3, text='DOWNLOAD VS CONTENT RATING', width=50, font=("Open Sans", 13, 'bold'),bg='blue', fg='white',command=download_content_rating).place(x=50, y=300)
    Button(screen3, text='DOWNLOAD VS APP SIZE', width=50, font=("Open Sans", 13, 'bold'),bg='blue', fg='white',command=download_size).place(x=50, y=350)
    Button(screen3, text='DOWNLOAD VS MONTH', width=50, font=("Open Sans", 13, 'bold'),bg='blue', fg='white',command=installs_month).place(x=50, y=400)
    Button(screen3, text='DOWNLOAD RATIO', width=50, font=("Open Sans", 13, 'bold'),bg='blue', fg='white',command=download_ratio).place(x=50, y=450)
    Button(screen3, text='DOWNLOAD VS CATEGORY', width=50, font=("Open Sans", 13, 'bold'),bg='blue', fg='white',command=download_category1).place(x=50, y=500)
    Button(screen3, text='DOWNLOAD VS ANDROID VERSION', width=50, font=("Open Sans", 13, 'bold'),bg='blue', fg='white',command=download_and).place(x=50, y=550)
    #Button(screen3, text='PRICE', width=50, font=("Open Sans", 13, 'bold'),bg='brown', fg='white',command=price).place(x=50, y=450)

        #Button(screen3,text="Exit", command=do_exit,bg="yellow",fg="black",font=("Open Sans", 16,"bold")).place(x=0,y=100)
```

7. Percentage Download in Each Category



8. No of Downloads

Code for Information:

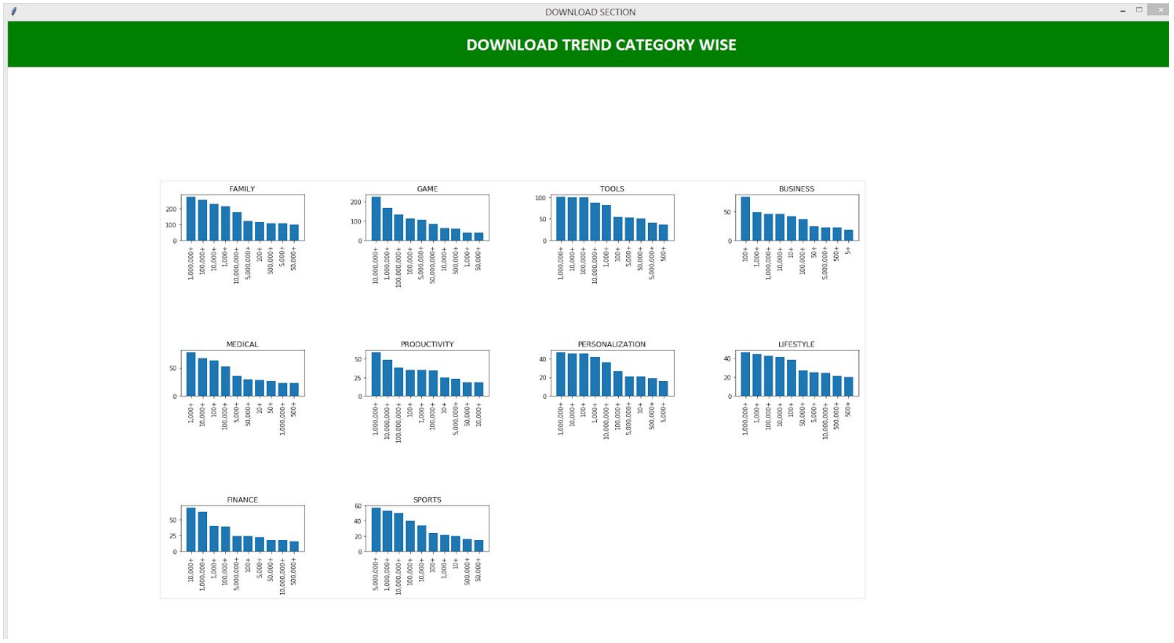
```
a=df.Installs.value_counts()[10000]
#print(a)
b=df.Installs.value_counts()[50000]
#print(b)
print("No of apps having downloads between 10 thousand and 50 thousand are ",a)
b1=df.Installs.value_counts()[100000]
b2=b+b1
print("No of apps having downloads between 50k and 1.5 lakh are ",b2)
print("No of apps having download between 1.5 lakh and 5 lalkh are ",b1)
c=df.Installs.value_counts()[500000]
#print(c)
d=df.Installs.value_counts()[1000000]
#print(d)
e=c+d
print("No apps having downloads between 5 lakh and 50 lakh are",e)
f=df.Installs.value_counts()[5000000]
g=df.Installs.value_counts()[10000000]
h=df.Installs.value_counts()[50000000]
i=df.Installs.value_counts()[100000000]
j=df.Installs.value_counts()[500000000]
k=df.Installs.value_counts()[1000000000]
l=f+g+h+i+j+k
```

print("No of apps having downloads more than 50 lakhs are",l)

GUI Code:

```
def downloads():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("DOWNLOAD SECTION")
    adjustWindow(screen5)
    Label(screen5, text="NO OF DOWNLOADS", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='No of apps having downloads between 10 thousand and 50 thousand are  986
\n\n No of apps having downloads between 50k and 1.5 lakh are  1550 \n\n No of apps having
download between 1.5 lakh and 5 lalkh are  1094 \n\n No apps having downloads between 5 lakh and
50 lakh are 1917 \n\n No of apps having downloads more than 50 lakhs are
1978',width='80',height='30' ,font=("Helvetica",10, 'bold', 'italic'), fg='white', bg='blue').place(x=0,
y=100)
    #Button(screen5,text="Exit", command=do_exit,bg="yellow",fg="black",font=("Open Sans",
16,"bold")).place(x=0,y=100)
```

9. Download Trend Category Wise



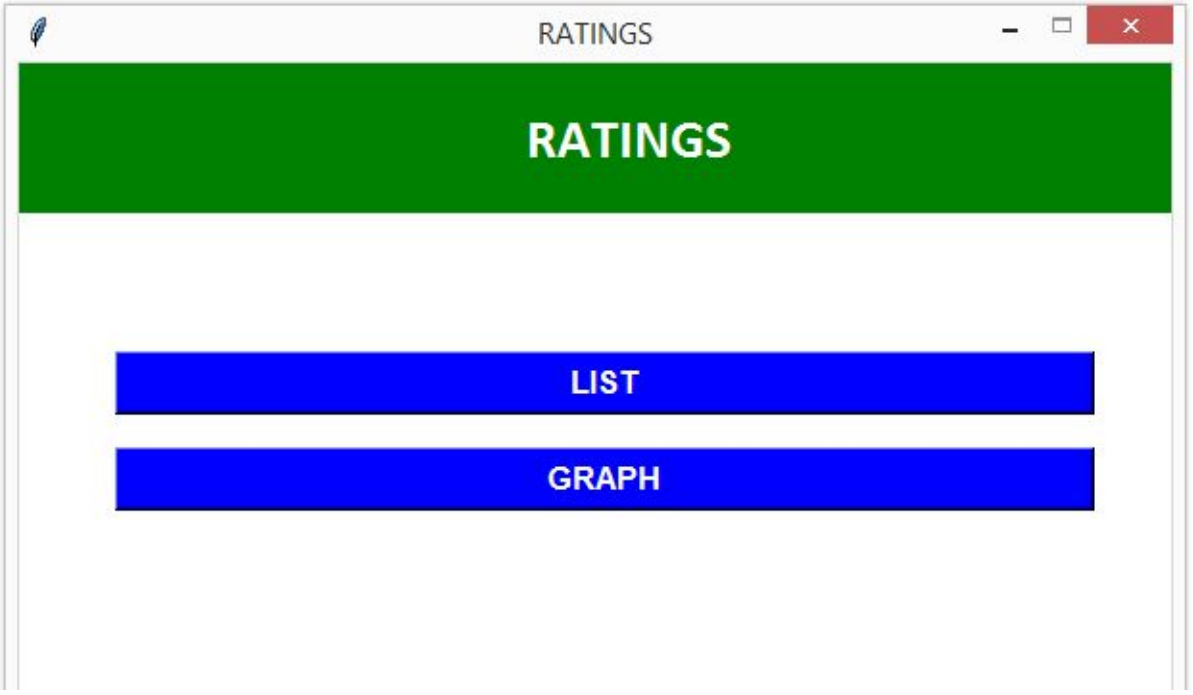
Code for Graphs:

```
gle_ply_store = pd.read_csv('C:\\Users\\Siddhesh\\Desktop\\jayesh\\Database 1')
def downloads(google_play_store):
    google_play_store.loc[google_play_store['Installs'] == "1,000,000+",'Installs'] = 1000000
    google_play_store.loc[google_play_store['Installs'] == "100,000+",'Installs'] = 100000
    google_play_store.loc[google_play_store['Installs'] == "10,000+",'Installs'] = 10000
    google_play_store.loc[google_play_store['Installs'] == "10,000,000+",'Installs'] = 10000000
    google_play_store.loc[google_play_store['Installs'] == "1,000+",'Installs'] = 1000
    google_play_store.loc[google_play_store['Installs'] == "100+",'Installs'] = 100
    google_play_store.loc[google_play_store['Installs'] == "5,000,000+",'Installs'] = 5000000
```

```
    google_play_store.loc[google_play_store['Installs'] == "500,000+",'Installs'] = 500000
    google_play_store.loc[google_play_store['Installs'] == "50,000+",'Installs'] = 50000
    google_play_store.loc[google_play_store['Installs'] == "5,000+",'Installs'] = 5000
    google_play_store.loc[google_play_store['Installs'] == "10+",'Installs'] = 10
    google_play_store.loc[google_play_store['Installs'] == "500+",'Installs'] = 500
    google_play_store.loc[google_play_store['Installs'] == "50,000,000+",'Installs'] = 50000000
    google_play_store.loc[google_play_store['Installs'] == "100,000,000+",'Installs'] = 100000000
    google_play_store.loc[google_play_store['Installs'] == "50+",'Installs'] = 50
    google_play_store.loc[google_play_store['Installs'] == "5+",'Installs'] = 5
    google_play_store.loc[google_play_store['Installs'] == "1+",'Installs'] = 1
    google_play_store.loc[google_play_store['Installs'] == "500,000,000+",'Installs'] = 500000000
    google_play_store.loc[google_play_store['Installs'] == "1,000,000,000+",'Installs'] = 1000000000
    google_play_store.loc[google_play_store['Installs'] == "0+",'Installs'] = 0
    return google_play_store
list_DFs = {}
list_DFs['FAMILY'] =
pd.DataFrame(gle_ply_store[gle_ply_store['Category']=='FAMILY']).sort_values(by=['Installs'],ascen
ding=False)
list_DFs['GAME'] =
pd.DataFrame(gle_ply_store[gle_ply_store['Category']=='GAME']).sort_values(by=['Installs'],ascendi
ng=False)
list_DFs['TOOLS'] =
pd.DataFrame(gle_ply_store[gle_ply_store['Category']=='TOOLS']).sort_values(by=['Installs'],ascend
ing=False)
list_DFs['BUSINESS'] =
pd.DataFrame(gle_ply_store[gle_ply_store['Category']=='BUSINESS']).sort_values(by=['Installs'],asc
ending=False)
list_DFs['MEDICAL'] =
pd.DataFrame(gle_ply_store[gle_ply_store['Category']=='MEDICAL']).sort_values(by=['Installs'],asc
ending=False)
list_DFs['PRODUCTIVITY'] =
pd.DataFrame(gle_ply_store[gle_ply_store['Category']=='PRODUCTIVITY']).sort_values(by=['Instal
ls'],ascending=False)
list_DFs['PERSONALIZATION'] =
pd.DataFrame(gle_ply_store[gle_ply_store['Category']=='PERSONALIZATION']).sort_values(by=['I
nstalls'],ascending=False)
list_DFs['LIFESTYLE'] =
pd.DataFrame(gle_ply_store[gle_ply_store['Category']=='LIFESTYLE']).sort_values(by=['Installs'],as
cending=False)
list_DFs['FINANCE'] =
pd.DataFrame(gle_ply_store[gle_ply_store['Category']=='FINANCE']).sort_values(by=['Installs'],asce
nding=False)
list_DFs['SPORTS'] =
pd.DataFrame(gle_ply_store[gle_ply_store['Category']=='SPORTS']).sort_values(by=['Installs'],ascen
ding=False)
list_DFs['SPORTS'] =
pd.DataFrame(gle_ply_store[gle_ply_store['Category']=='SPORTS']).sort_values(by=['Installs'],ascen
ding=False)
list_DFs['ART_AND_DESIGN'] =
pd.DataFrame(gle_ply_store[gle_ply_store['Category']=='ART_AND_DESIGN']).sort_values(by=['In
stalls'],ascending=False)
```

```
list_DFs['COMMUNICATION'] =
pd.DataFrame(gle_ply_store[gle_ply_store['Category']=='COMMUNICATION']).sort_values(by=['In
stalls'],ascending=False)
list_DFs['ENTERTAINMENT'] =
pd.DataFrame(gle_ply_store[gle_ply_store['Category']=='ENTERTAINMENT']).sort_values(by=['Ins
talls'],ascending=False)
list_DFs['SOCIAL'] =
pd.DataFrame(gle_ply_store[gle_ply_store['Category']=='SOCIAL']).sort_values(by=['Installs'],ascen
ding=False)
list_DFs['VIDEO_PLAYERS'] =
pd.DataFrame(gle_ply_store[gle_ply_store['Category']=='VIDEO_PLAYERS']).sort_values(by=['Inst
alls'],ascending=False)

fig = plt.figure(figsize=(25,20))
for i,(k,v) in enumerate(list_DFs.items()):
    fig.add_subplot(4,4,i+1)
    plt.bar(v['Installs'].value_counts()[:25].index,v['Installs'].value_counts()[:25].values)
    print(k)
    print(v['Installs'].value_counts())
    plt.xticks(rotation=90)
    plt.title(k)
    #plt.annotate("App Size Count",)
    fig.subplots_adjust(wspace = 0.5,hspace =2.4)
plt.show()
```

10. Rating



Code:

```
def analysis5():
    global screen7
    screen7=Toplevel(screen)
```

```
    screen7.title("RATINGS")
    adjustWindow(screen7)
    Label(screen7, text="RATINGS",width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Button(screen7, text='LIST', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=listl).place(x=50, y=150)
    Button(screen7, text='GRAPH', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=download_rating).place(x=50, y=200)
```

11. List of Genres

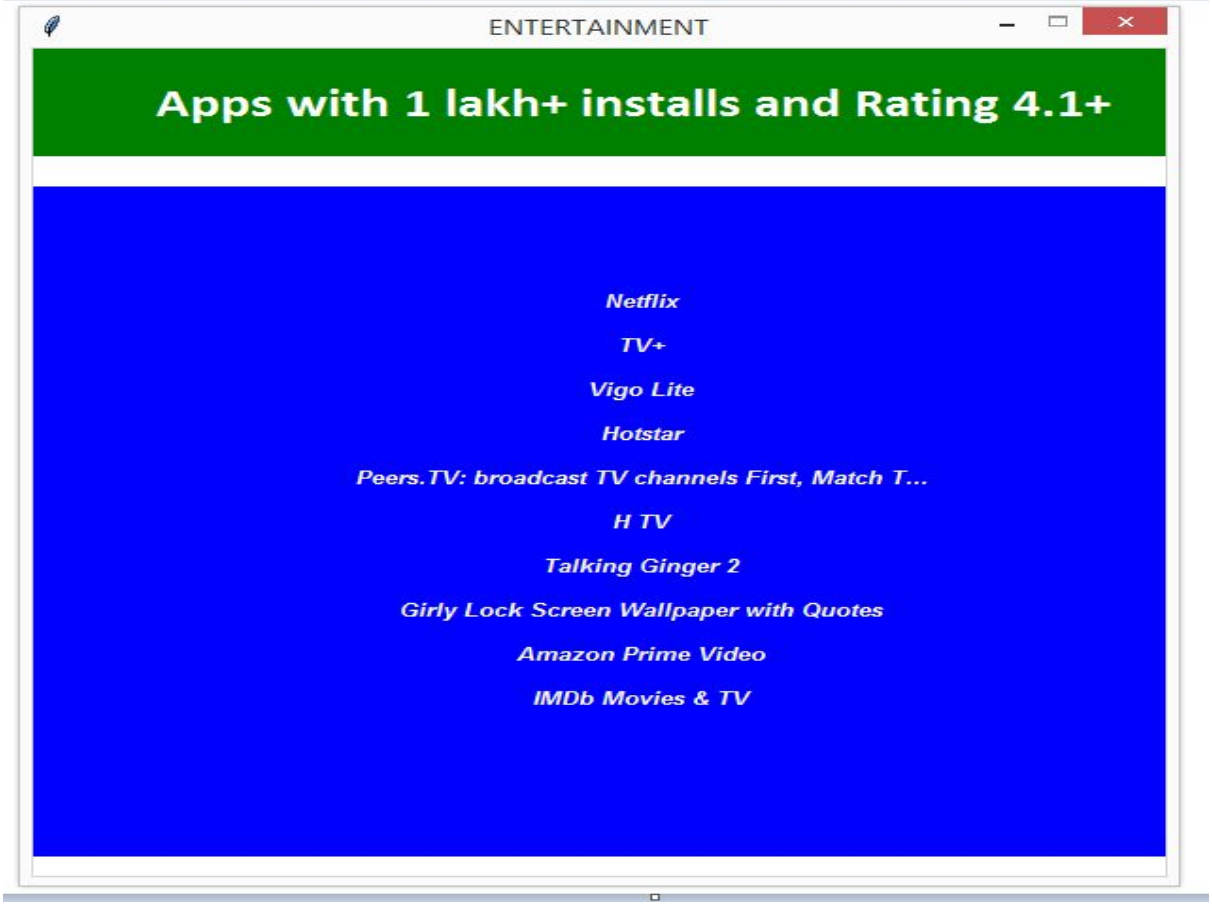
Code for Information:

```
genres=  list(df2['index'][0:10])
d = pd.DatetimeIndex(playstore_data['Last Updated'])
playstore_data['year'] = d.year
playstore_data['month'] = d.month
d=d.month
print(d)
for i in genres:
    play = playstore_data[(playstore_data['Installs'] >= '100,000+') & (playstore_data['Genres'] == i) &
(playstore_data['Rating'] >= 4.1)]['App']
    print('')
    print('Printing 10 Apps with 100,000+ installs and Rating >= 4.1 {}'.format(i))
    print('----------------------------------------------------------------------')
    print(play[:10])
```

Code for GUI:

```
def listl():
    global screen7
    screen7=Toplevel(screen)
    screen7.title("RATINGS")
    adjustWindow(screen7)
    Label(screen7, text="GENRE",width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Button(screen7, text='Entertainment', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=enter).place(x=50, y=100)
    Button(screen7, text='Education', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=edu).place(x=50, y=150)
    Button(screen7, text='Business', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=bus).place(x=50, y=200)
    Button(screen7, text='Medical', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=med).place(x=50, y=250)
    Button(screen7, text='Productivity', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=pro).place(x=50, y=300)
    Button(screen7, text='Personalization', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=per).place(x=50, y=350)
    Button(screen7, text='Lifestyle', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=life).place(x=50, y=400)
    Button(screen7, text='Finance', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=fin).place(x=50, y=450)
    Button(screen7, text='Sports', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=sport).place(x=50, y=500)
    Button(screen7, text='Tools', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=tool).place(x=50, y=550)
```

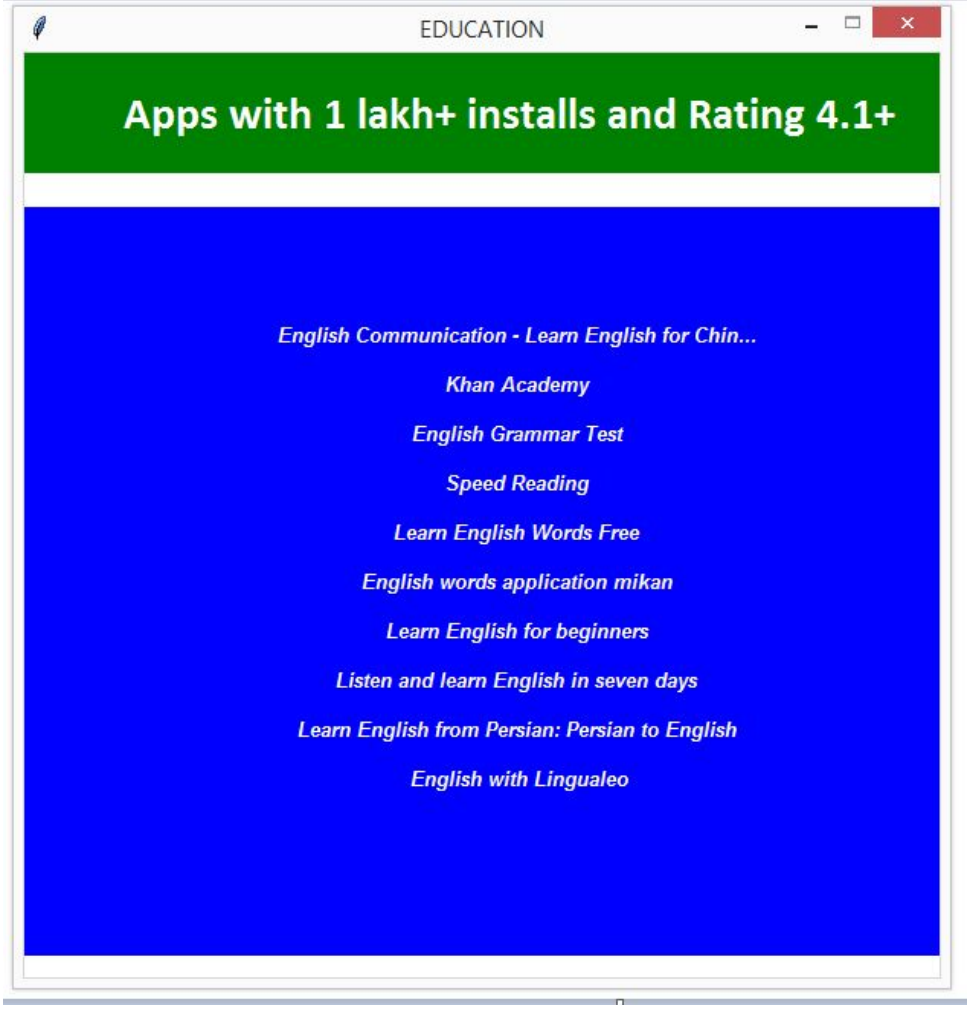
12. Entertainment



Code:

```
def enter():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("ENTERTAINMENT")
    adjustWindow(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="Apps with 1 lakh+ installs and Rating 4.1+", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='Netflix\n\nTV+\n\nVigo Lite\n\nHotstar\n\nPeers.TV: broadcast TV channels
First, Match T...\n\nH TV\n\nTalking Ginger 2\n\nGirly Lock Screen Wallpaper with
Quotes\n\nAmazon Prime Video\n\nIMDb Movies & TV\n\n',width='80',height='30'
,font=("Helvetica",10, 'bold', 'italic'), fg='white', bg='blue').place(x=0, y=100)
```

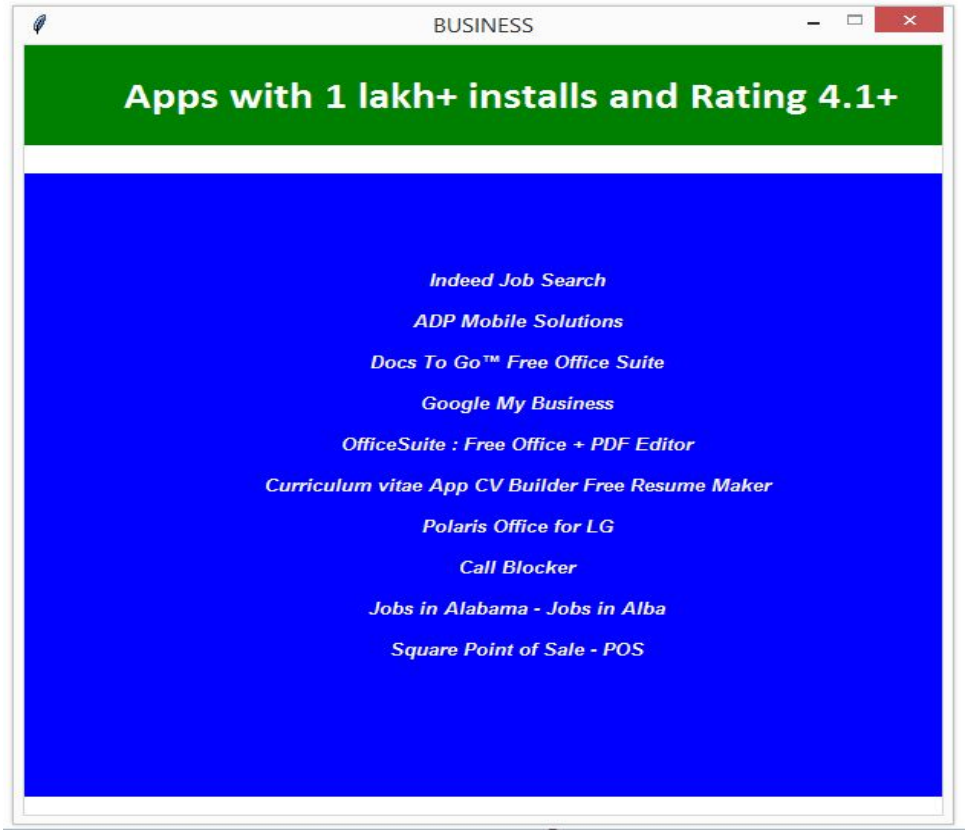
13. Education



Code:

```
def edu():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("EDUCATION")
    adjustWindow(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="Apps with 1 lakh+ installs and Rating 4.1+", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='English Communication - Learn English for Chin...\n\nKhan
Academy\n\nEnglish Grammar Test\n\nSpeed Reading\n\nLearn English Words Free\n\nEnglish
words application mikan\n\nLearn English for beginners\n\nListen and learn English in seven
days\n\nLearn English from Persian: Persian to English\n\n English with
Lingualeo\n\n',width='80',height='30' ,font=("Helvetica",10, 'bold', 'italic'), fg='white',
bg='blue').place(x=0, y=100)
```

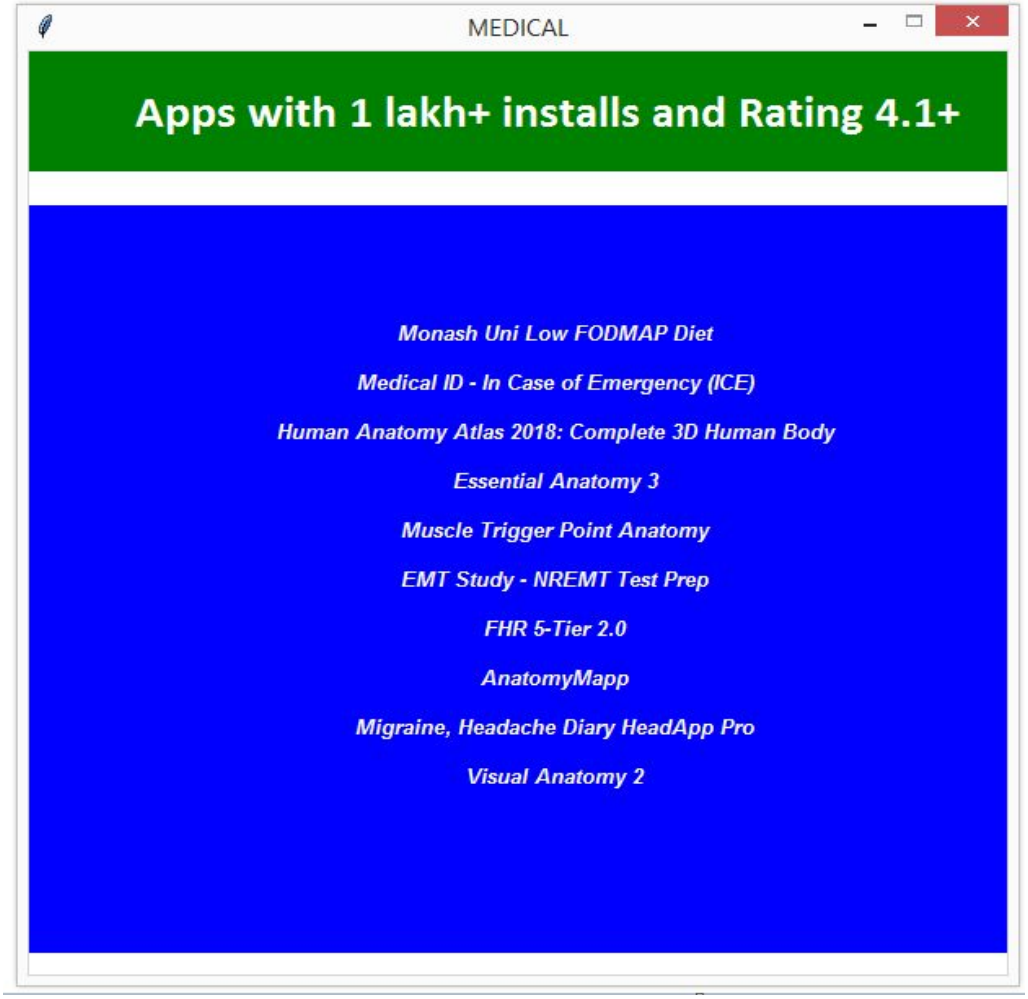
14. Business



Code:

```
def bus():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("BUSINESS")
    adjustWindow(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="Apps with 1 lakh+ installs and Rating 4.1+", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='Indeed Job Search\n\nADP Mobile Solutions\n\nDocs To Go™ Free Office
Suite\n\nGoogle My Business\n\nOfficeSuite : Free Office + PDF Editor\n\nCurriculum vitae App
CV Builder Free Resume Maker\n\nPolaris Office for LG\n\nCall Blocker\n\nJobs in Alabama - Jobs
in Alba\n\nSquare Point of Sale - POS\n\n',width='80',height='30' ,font=("Helvetica",10, 'bold',
'italic'), fg='white', bg='blue').place(x=0, y=100)
```

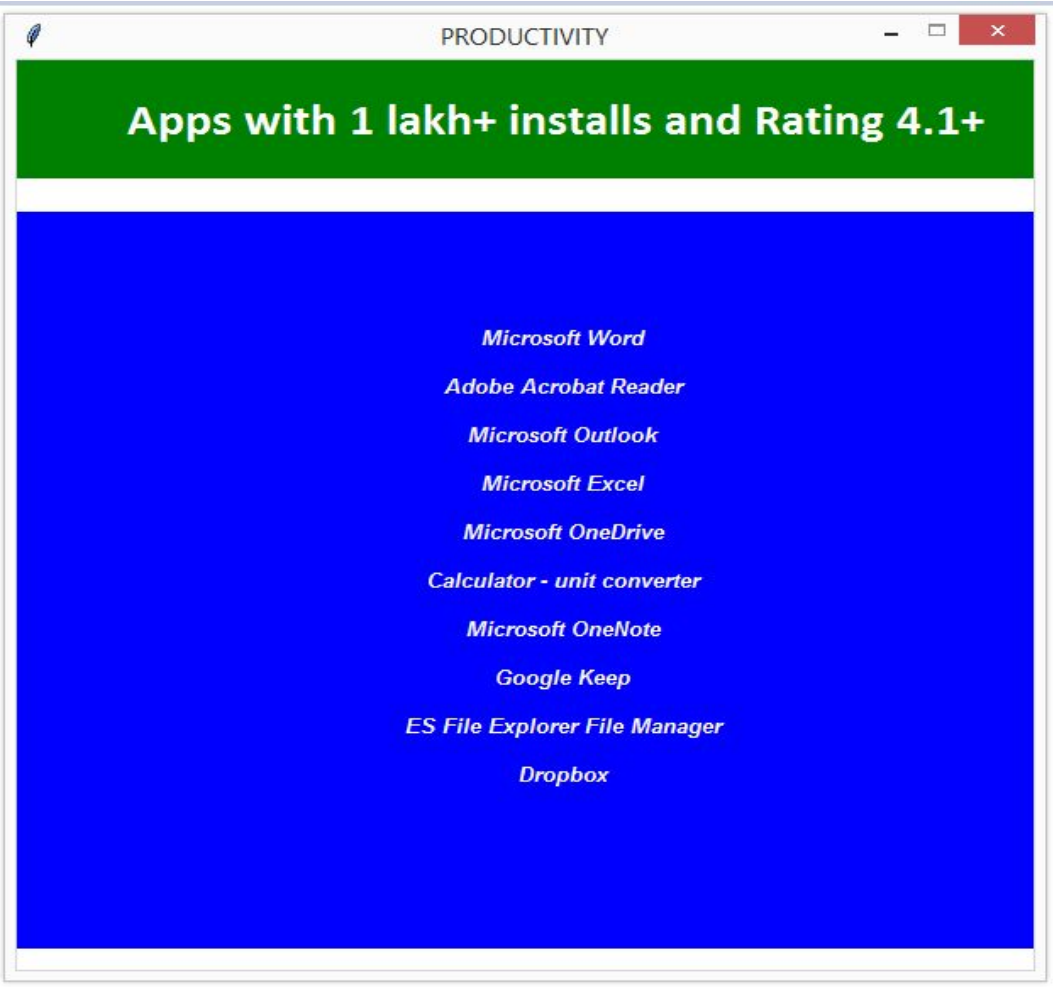
15. Medical



Code:

```
def med():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("MEDICAL")
    adjustWindow(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="Apps with 1 lakh+ installs and Rating 4.1+", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='Monash Uni Low FODMAP Diet\n\nMedical ID - In Case of Emergency
(ICE)\n\nHuman Anatomy Atlas 2018: Complete 3D Human Body\n\nEssential Anatomy
3\n\nMuscle Trigger Point Anatomy\n\nEMT Study - NREMT Test Prep\n\nFHR 5-Tier
2.0\n\nAnatomyMapp\n\nMigraine, Headache Diary HeadApp Pro\n\nVisual Anatomy
2\n\n',width='80',height='30' ,font=("Helvetica",10, 'bold', 'italic'), fg='white', bg='blue').place(x=0,
y=100)
```

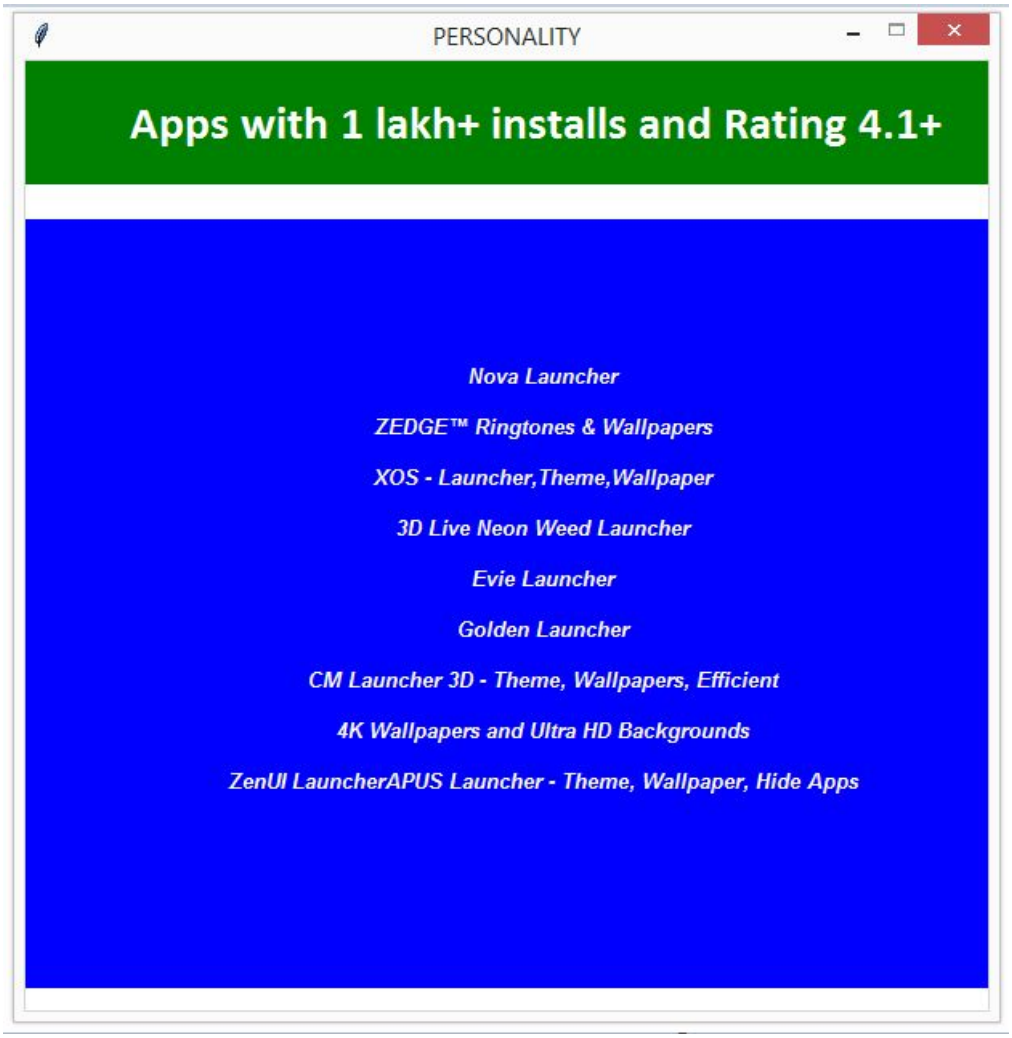
16. Productivity



Code:

```
def pro():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("PRODUCTIVITY")
    adjustWindow(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="Apps with 1 lakh+ installs and Rating 4.1+", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='Microsoft Word\n\nAdobe Acrobat Reader\n\nMicrosoft
Outlook\n\nMicrosoft Excel\n\nMicrosoft OneDrive\n\nCalculator - unit converter\n\nMicrosoft
OneNote\n\nGoogle Keep\n\nES File Explorer File Manager\n\nDropbox\n\n',width='80',height='30'
,font=("Helvetica",10, 'bold', 'italic'), fg='white', bg='blue').place(x=0, y=100)
```

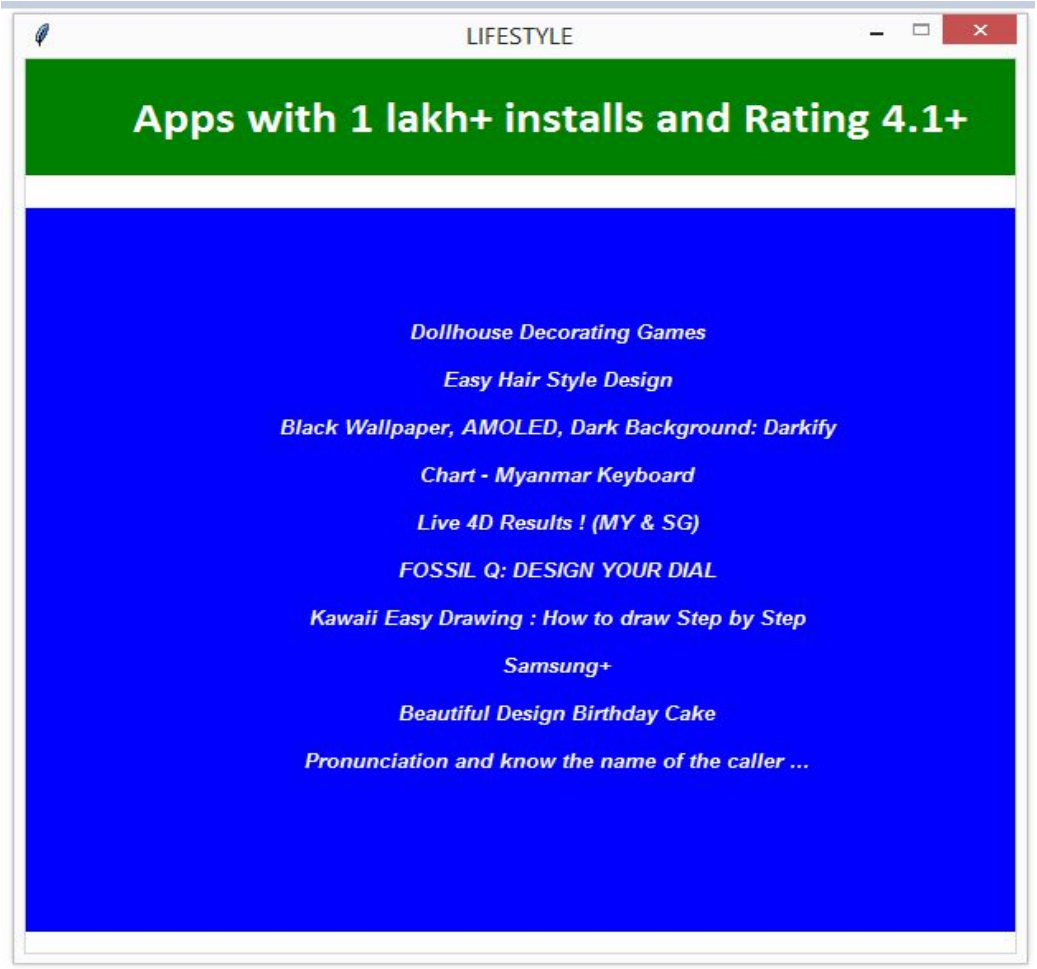
17. Personality



Code:

```
def per():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("PERSONALITY")
    adjustWindow(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="Apps with 1 lakh+ installs and Rating 4.1+", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='Nova Launcher\n\nZEDGE™ Ringtones & Wallpapers\n\nXOS -
Launcher,Theme,Wallpaper\n\n3D Live Neon Weed Launcher\n\nEvie Launcher\n\nGolden
Launcher\n\nCM Launcher 3D - Theme, Wallpapers, Efficient\n\n4K Wallpapers and Ultra HD
Backgrounds\n\nZenUI LauncherAPUS Launcher - Theme, Wallpaper, Hide
Apps\n\n',width='80',height='30' ,font=("Helvetica",10, 'bold', 'italic'), fg='white',
bg='blue').place(x=0, y=100)
```

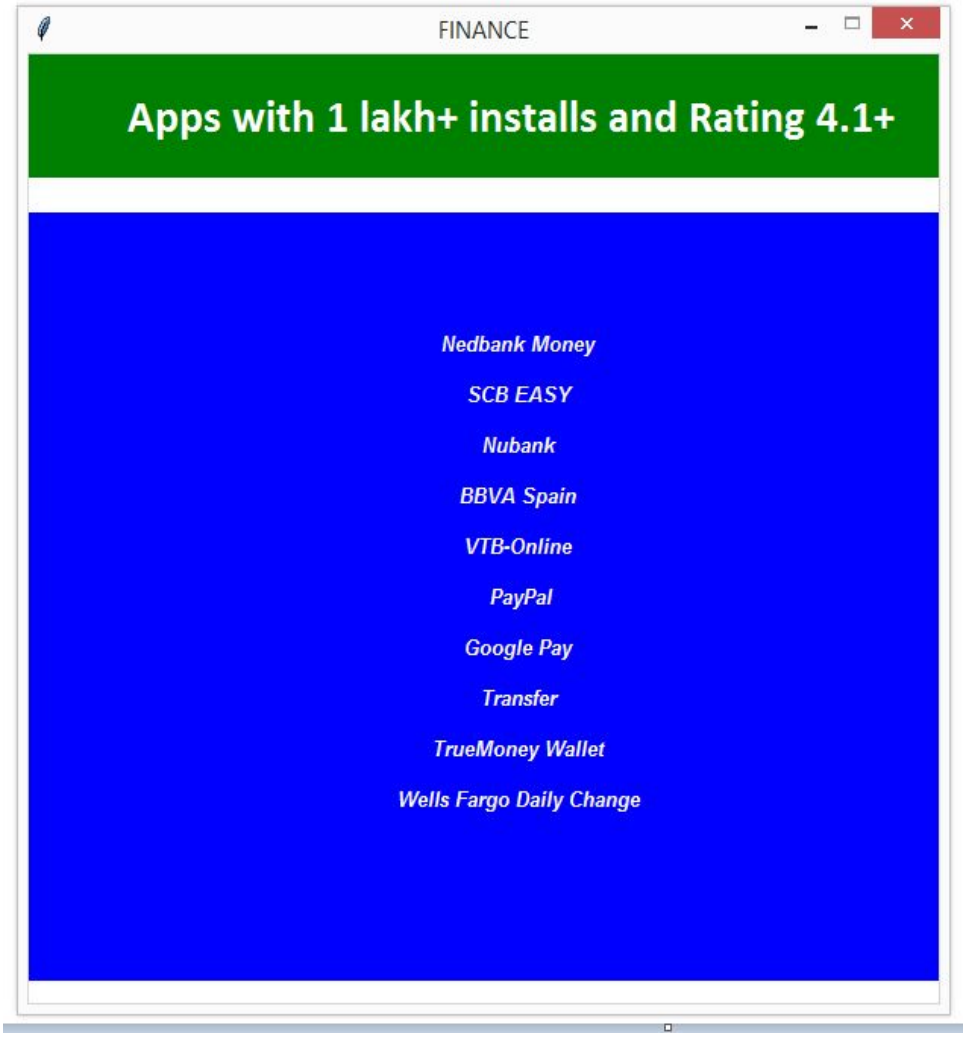
18. Lifestyle



Code:

```
def life():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("LIFESTYLE")
    adjustWindow(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="Apps with 1 lakh+ installs and Rating 4.1+", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='Dollhouse Decorating Games\n\nEasy Hair Style Design\n\nBlack Wallpaper,
AMOLED, Dark Background: Darkify\n\nChart - Myanmar Keyboard\n\nLive 4D Results ! (MY &
SG)\n\nFOSSIL Q: DESIGN YOUR DIAL\n\nKawaii Easy Drawing : How to draw Step by
Step\n\nSamsung+\n\nBeautiful Design Birthday Cake\n\nPronunciation and know the name of the
caller ...\n\n',width='80',height='30' ,font=("Helvetica",10, 'bold', 'italic'), fg='white',
bg='blue').place(x=0, y=100)
```

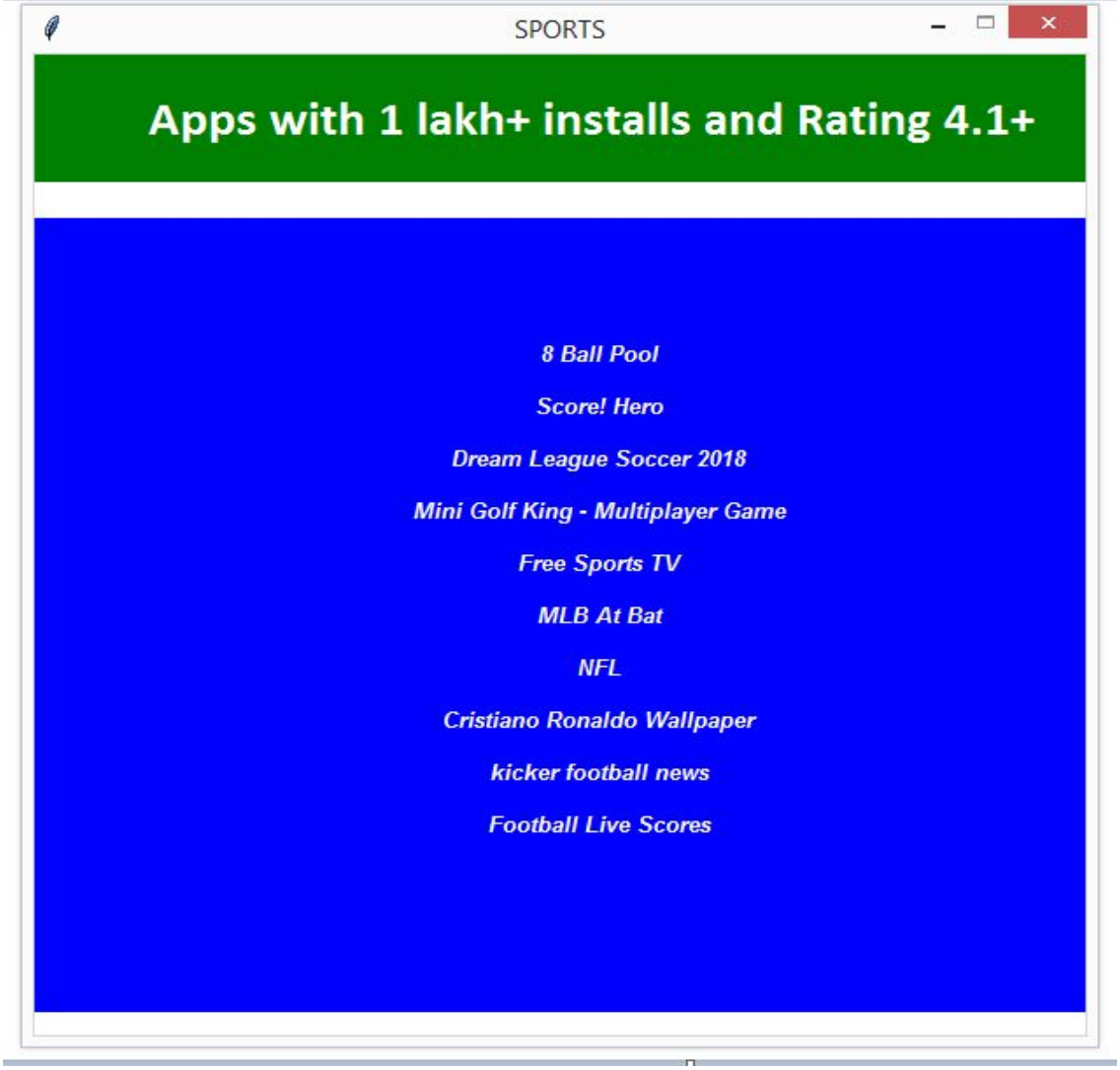
19. Finance



Code:

```
def fin():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("FINANCE")
    adjustWindow(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="Apps with 1 lakh+ installs and Rating 4.1+", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='Nedbank Money\n\nSCB EASY\n\nNubank\n\nBBVA
Spain\n\nVTB-Online\n\n PayPal\n\nGoogle Pay\n\nTransfer\n\nTrueMoney Wallet\n\nWells Fargo
Daily Change\n\n',width='80',height='30' ,font=("Helvetica",10, 'bold', 'italic'), fg='white',
bg='blue').place(x=0, y=100)
```

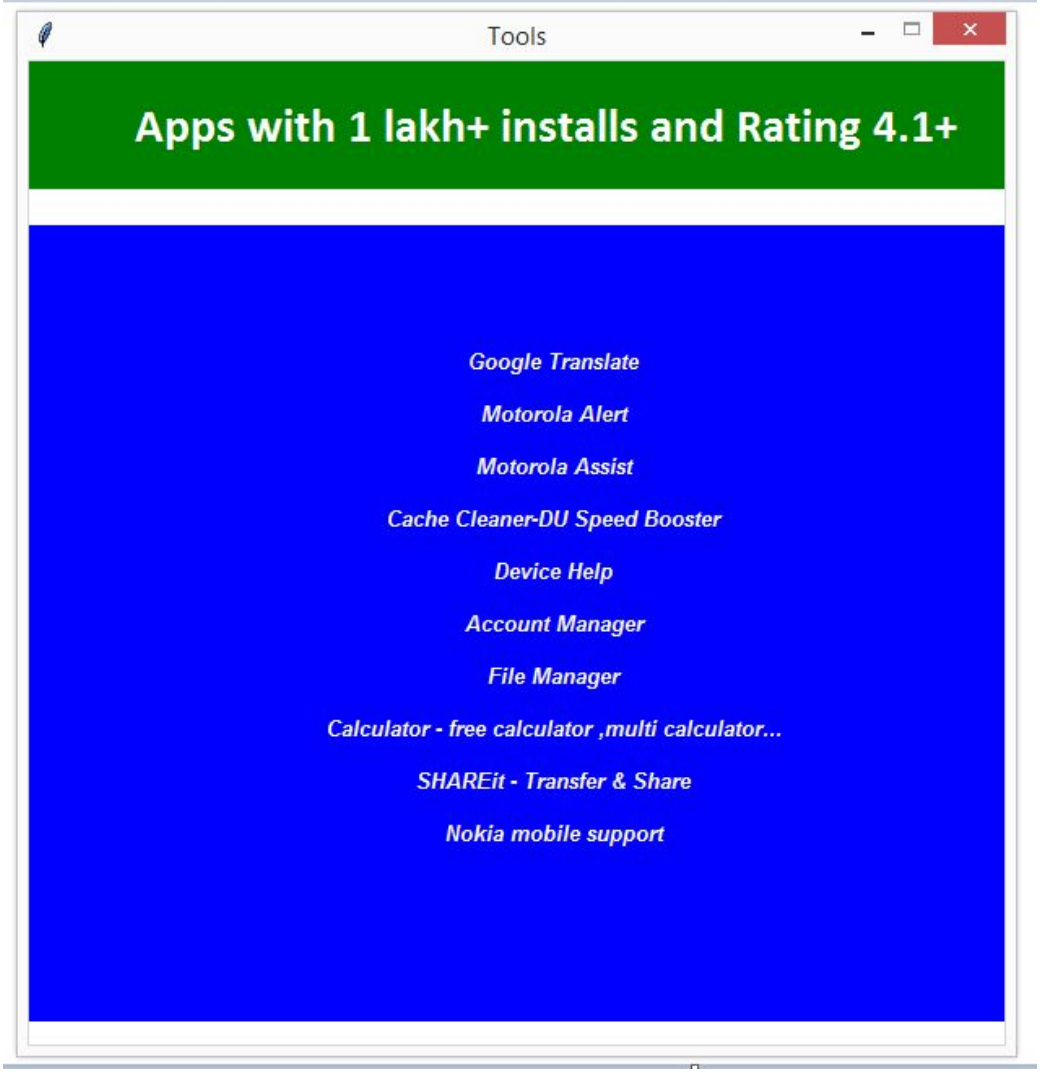
20. Sports



Code:

```
def sport():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("SPORTS")
    adjustWindow(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="Apps with 1 lakh+ installs and Rating 4.1+", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='8 Ball Pool\n\nScore! Hero\n\nDream League Soccer 2018\n\nMini Golf King
- Multiplayer Game\n\nFree Sports TV\n\nMLB At Bat\n\nNFL\n\nCristiano Ronaldo
Wallpaper\n\nkicker football news\n\nFootball Live Scores\n\n',width='80',height='30'
,font=("Helvetica",10, 'bold', 'italic'), fg='white', bg='blue').place(x=0, y=100)
```

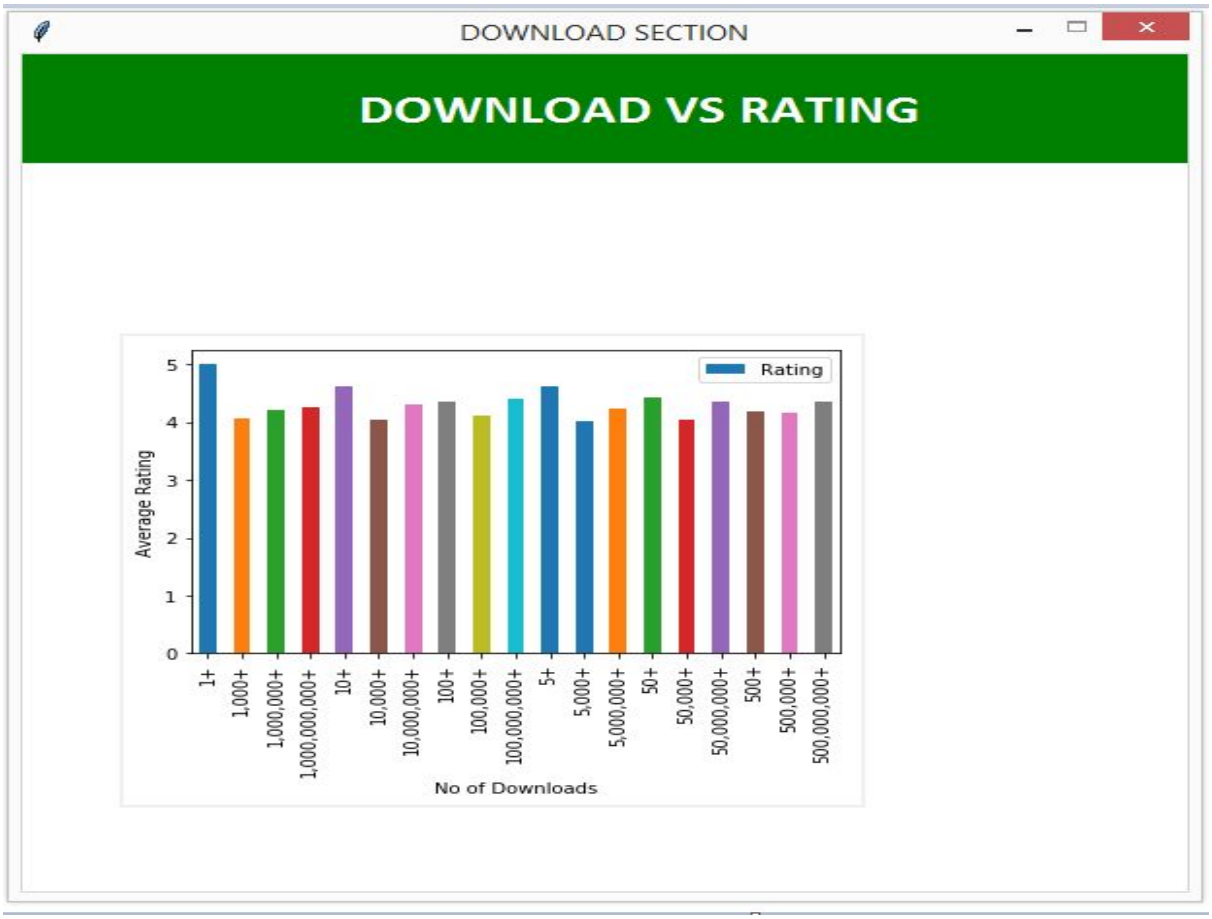
21. Tools



Code:

```
def tool():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("Tools")
    adjustWindow(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="Apps with 1 lakh+ installs and Rating 4.1+", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='Google Translate\n\nMotorola Alert\n\nMotorola Assist\n\nCache
Cleaner-DU Speed Booster\n\nDevice Help\n\nAccount Manager\n\nFile Manager\n\nCalculator -
free calculator ,multi calculator...\n\nSHAREit - Transfer & Share\n\nNokia mobile
support\n\n',width='80',height='30' ,font=("Helvetica",10, 'bold', 'italic'), fg='white',
bg='blue').place(x=0, y=100)
```

22. Download vs Rating



Code for Graph:

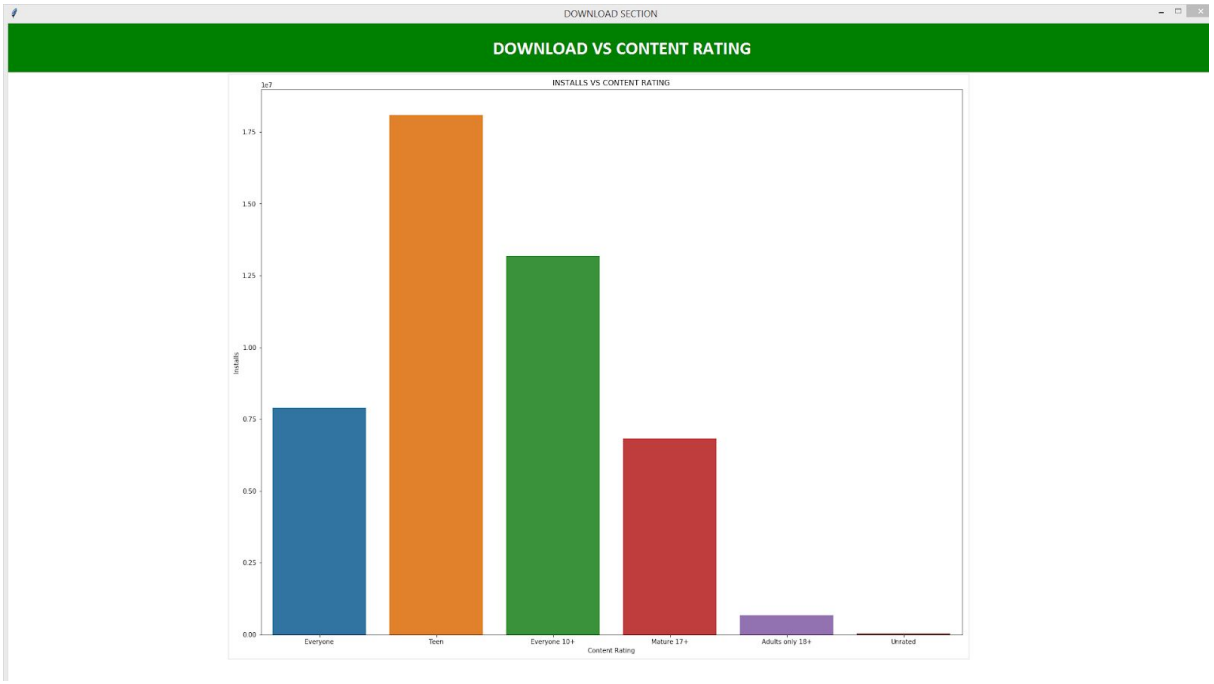
```
df=pd.read_csv('C:\\Users\\Siddhesh\\Desktop\\jayesh\\Database 1')
df = df.dropna()
#df = df.reset_index()
#df= df.sort_index(axis=0,ascending=False)
df_avgRatingGrouped = df.groupby(['Installs'], as_index=False).mean()
df_avgRatingGrouped.plot(kind = 'bar', x = 'Installs', y = 'Rating')
plt.xlabel('No of Downloads')
plt.ylabel('Average Rating')
```

Code for GUI:

```
def download_rating():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("DOWNLOAD SECTION")
    adjustWindow(screen5)
    Label(screen5, text="DOWNLOAD VS RATING", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    photo = PhotoImage(file="dvsr.png") # opening left side image - Note: If image is in same
folderthen no need to mention the full path
    label = Label(screen5, image=photo ,text="") # attaching image to the label
    label.place(x=50, y=200)
```

label.image = photo
#Button(screen5,text="Exit", command=do_exit,bg="yellow",fg="black",font=("Open Sans", 16,"bold")).place(x=0,y=100)

23. Download vs Content Rating



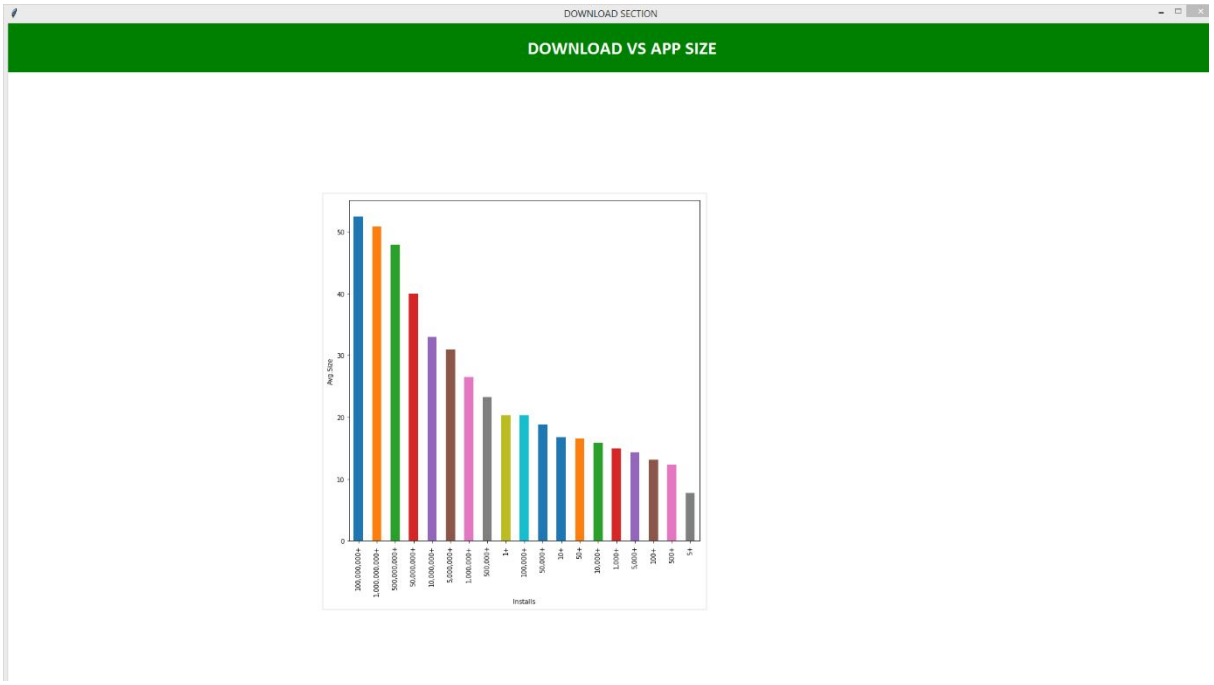
Code for graph:

```
plt.figure(figsize=(20,20))
#df['Categor']=df['Content Rating'][:1000]
sns.barplot(y='Installs',x='Content Rating',data=df,ci=None)
plt.title("INSTALLS VS CONTENT RATING")
plt.show()
```

Code for GUI:

```
def download_content_rating():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("DOWNLOAD SECTION")
    adjustWindow1(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="DOWNLOAD VS CONTENT RATING", width = '130',height ='2', font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    photo = PhotoImage(file="Installs vs Content Rating.png") # opening left side image - Note: If image is in same folderthen no need to mention the full path
    label = Label(screen5, image=photo ,text="") # attaching image to the label
    label.place(x=350, y=80)
    label.image = photo
```

24. Downloads vs App Size



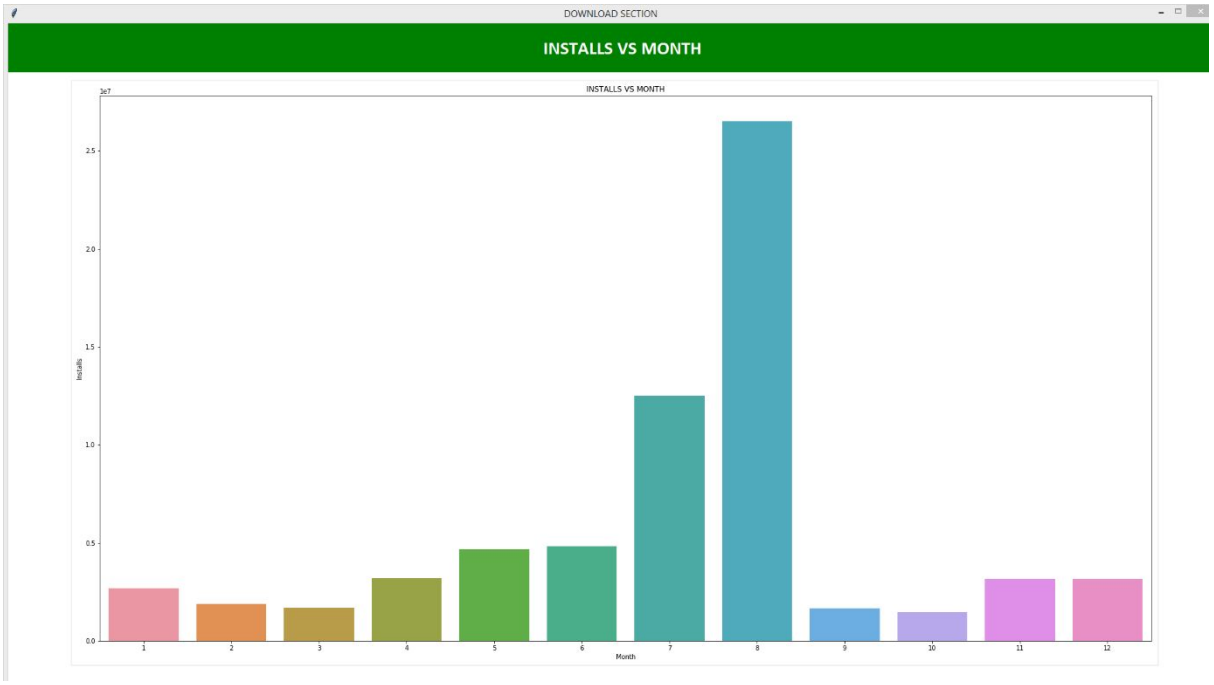
Code for Graph:

```
apps.groupby('Installs')["Size"].mean().sort_values(ascending=False).plot(kind='bar',figsize=(10,10))
plt.ylabel('Avg.Size')
plt.show()
```

Code for GUI:

```
def download_size():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("DOWNLOAD SECTION")
    adjustWindow1(screen5)
    Label(screen5, text="DOWNLOAD VS APP SIZE", width = '130',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    photo = PhotoImage(file="dvsas.png") # opening left side image - Note: If image is in same
folderthen no need to mention the full path
    label = Label(screen5, image=photo ,text="") # attaching image to the label
    label.place(x=500, y=270)
    label.image = photo
```

25. Downloads vs Month



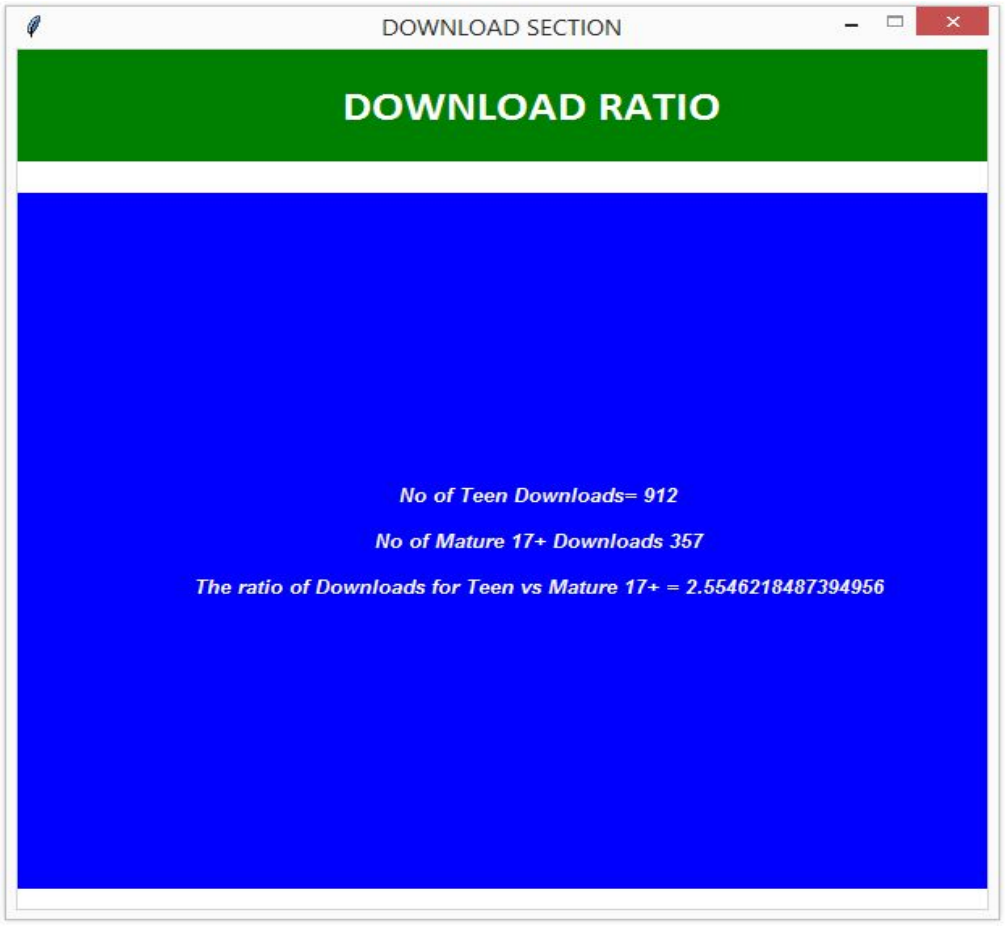
Code for Graph :

```
plt.figure(figsize=(30,20))
sns.barplot(y='Installs',x=df['Month'],data=df,ci=None)
plt.title("INSTALLS VS MONTH")
plt.show()
```

Code for GUI:

```
def installs_month():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("DOWNLOAD SECTION")
    adjustWindow1(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="INSTALLS VS MONTH", width = '130',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    photo = PhotoImage(file="installs vs month.png") # opening left side image - Note: If image is in
same folderthen no need to mention the full path
    label = Label(screen5, image=photo ,text="") # attaching image to the label
    label.place(x=100, y=90)
    label.image = photo
```

26. Download Ratio



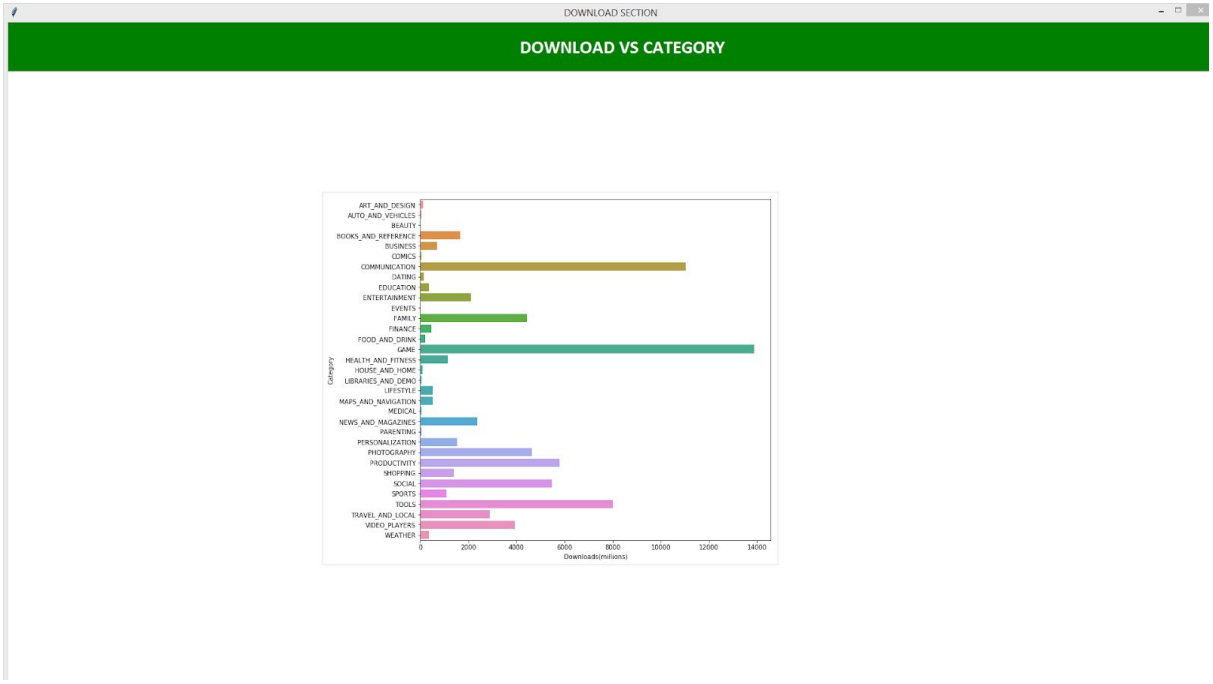
Code for Information:

```
d= df[(df['Content Rating'] == 'Teen')]['Content Rating']
e= df[(df['Content Rating'] == 'Mature 17+')]['Content Rating']
print("No of Teen Downloads=",len(d))
print("No of Mature 17+ Downloads",len(e))
f=len(d)/len(e)
print("The ratio of Downloads for Teen vs Mature 17+ =",f)
```

Code for GUI:

```
def download_ratio():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("DOWNLOAD SECTION")
    adjustWindow(screen5)
   # screen5.resizable(True,True)
    Label(screen5, text="DOWNLOAD RATIO", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='No of Teen Downloads= 912\n\nNo of Mature 17+ Downloads 357\n\nThe
ratio of Downloads for Teen vs Mature 17+ = 2.5546218487394956',width='80',height='30'
,font=("Helvetica",10, 'bold', 'italic'), fg='white', bg='blue').place(x=0, y=100)
```

27. Downloads vs Category



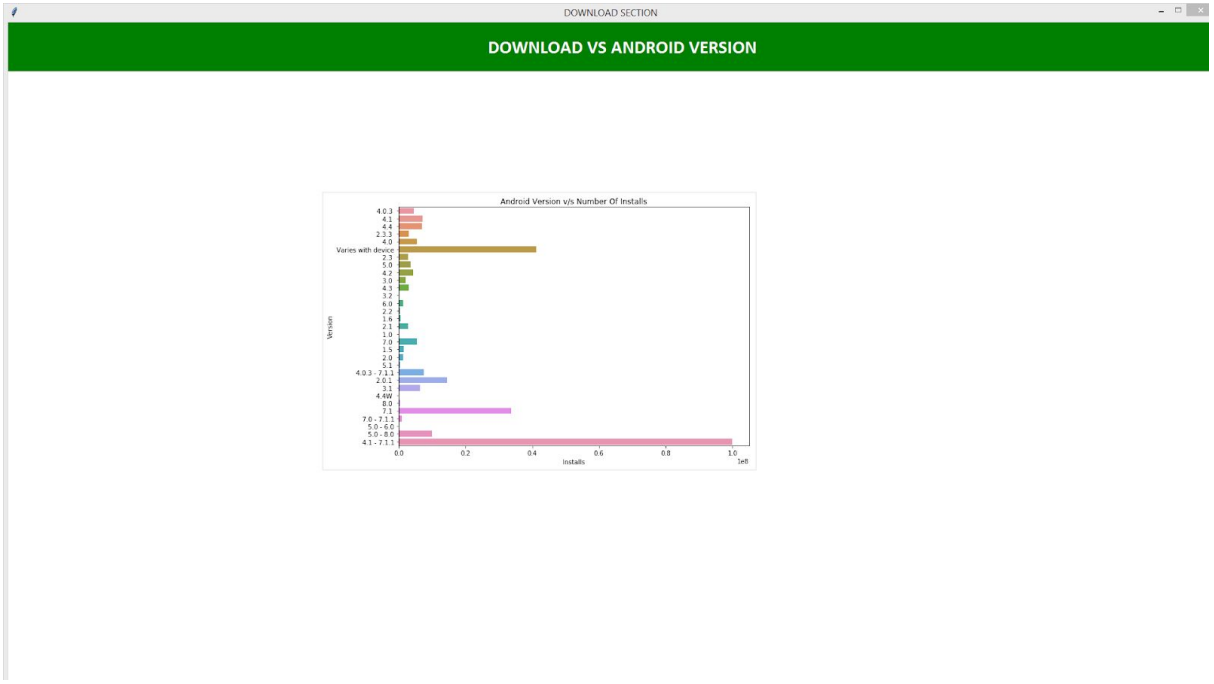
Code for Graph:

```
def appDownloads():
    downloads={}
    for i in df['Category']:
        x=i
        t1=(df[(df.Category==x)].Installs).tolist()
        downloads.update({i:int(sum(t1)/10**6)})
    print(downloads)
    plt.bar(downloads.keys(),downloads.values(),color='red')
    plt.xlabel("Category")
    plt.ylabel("""Number of Downloads
     (in millions)""")
    plt.xticks(rotation=90)
    plt.show()
```

Code for GUI:

```
def download_category1():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("DOWNLOAD SECTION")
    adjustWindow1(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="DOWNLOAD VS CATEGORY", width = '130',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    photo = PhotoImage(file="appdownloads.png") # opening left side image - Note: If image is in
same folderthen no need to mention the full path
    label = Label(screen5, image=photo ,text="") # attaching image to the label
    label.place(x=500, y=270)
    label.image = photo
```

28. Download vs Android Version



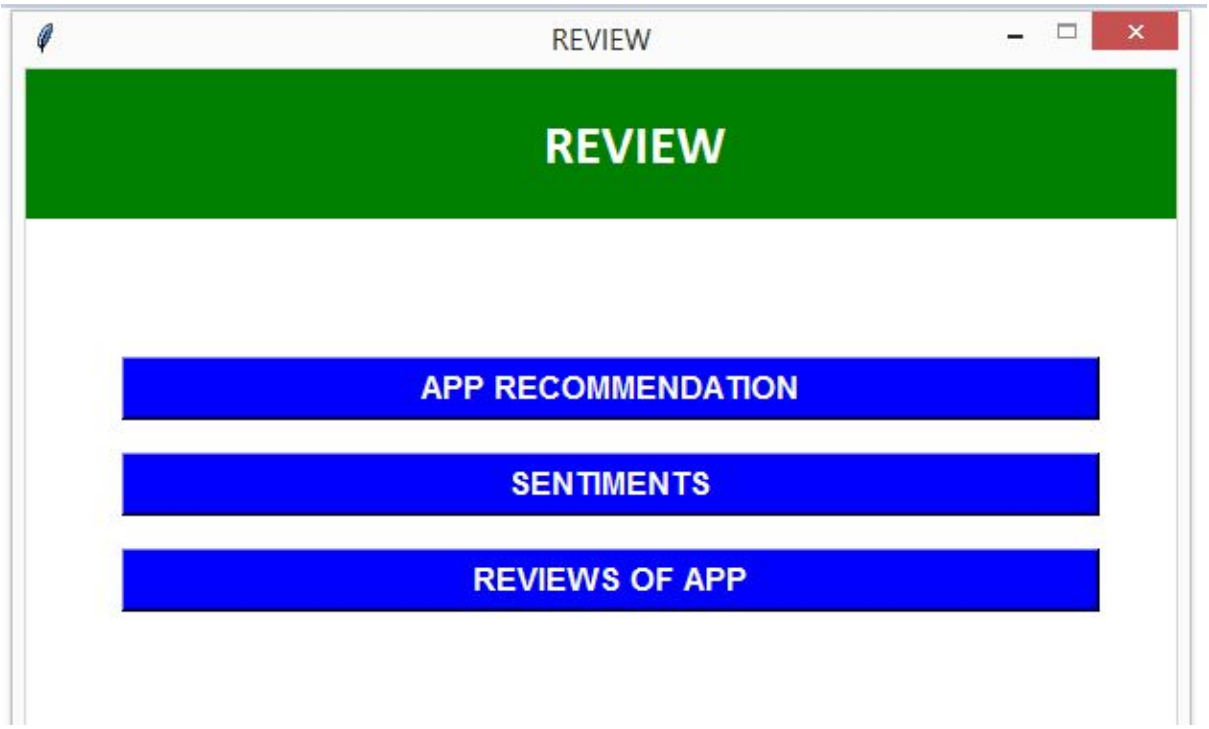
Code for Graph:

```
def versionDownloads():
    plt.figure(figsize=(10,7))
    plt.title('Android Version v/s Number Of Installs')
    an=sns.barplot(y=df['Android Ver'],x=df['Installs'],data=df,ci=None)
    an.set(xlabel='Installs', ylabel='Version')
    plt.show()
    print(df['Android Ver'])
```

Code for GUI:

```
def download_and():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("DOWNLOAD SECTION")
    adjustWindow1(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="DOWNLOAD VS ANDROID VERSION", width = '130',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    photo = PhotoImage(file="versiondownloads.png") # opening left side image - Note: If image is in
same folderthen no need to mention the full path
    label = Label(screen5, image=photo ,text="") # attaching image to the label
    label.place(x=500, y=270)
    label.image = photo
```

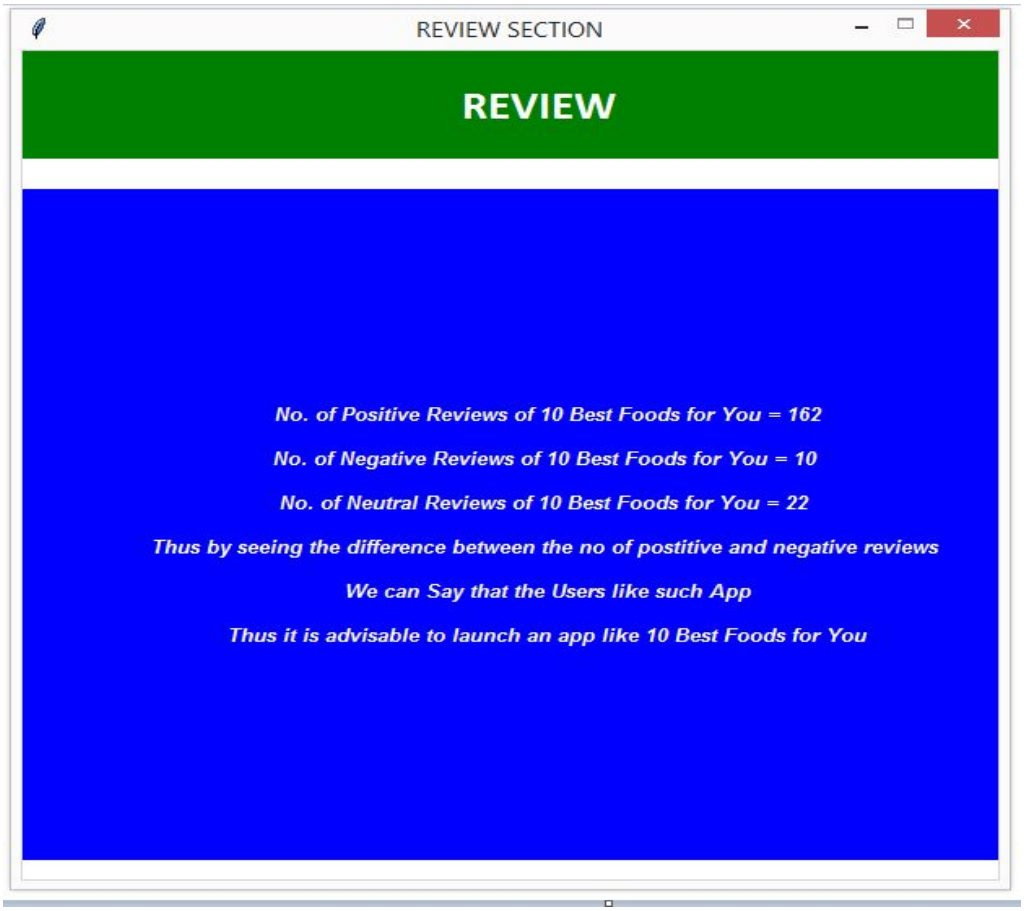
29. Reviews



Code:

```
def analysis2():
    global screen4
    screen4=Toplevel(screen)
    screen4.title("REVIEW")
    adjustWindow(screen4)
    Label(screen4, text="REVIEW",width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Button(screen4, text='APP RECOMMENDATION', width=50, font=("Open Sans", 13,
'bold'),bg='blue', fg='white',command=review).place(x=50, y=150)
    Button(screen4, text='SENTIMENTS', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=sentiment).place(x=50, y=200)
    Button(screen4, text='REVIEWS OF APP', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=reg).place(x=50, y=250)
```

30. Reviews



Code for Information:

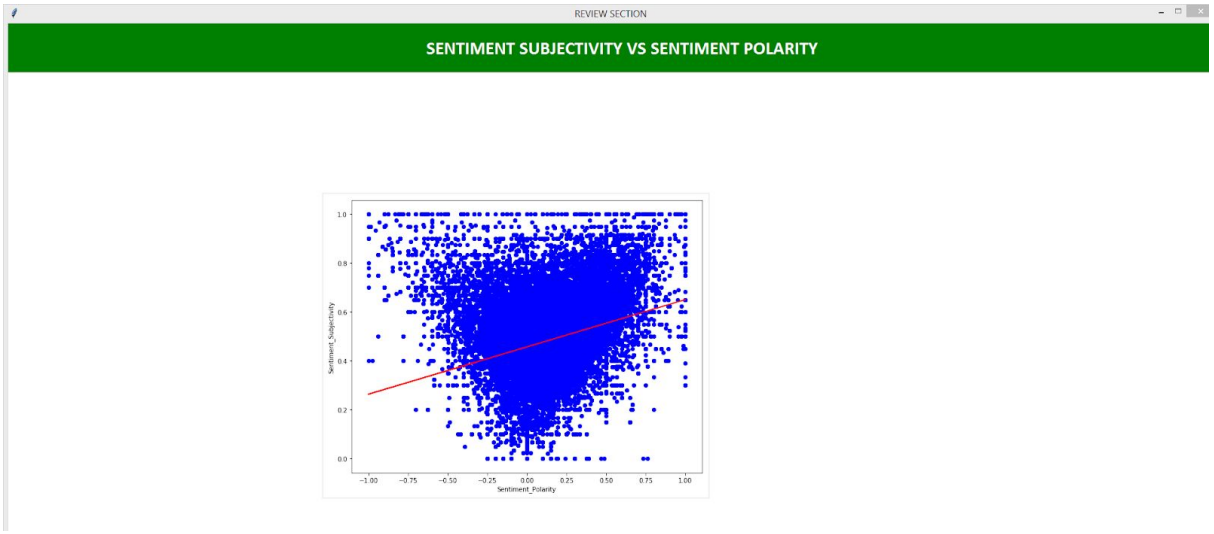
```
d= data[(data['Sentiment'] == 'Positive') & (data['App']=='10 Best Foods for You')]['App']
e= data[(data['Sentiment'] == 'Negative') & (data['App']=='10 Best Foods for You')]['App']
f=data[(data['Sentiment'] == 'Neutral') & (data['App']=='10 Best Foods for You')]['App']
print('No. of Positive Reviews of 10 Best Foods for You =',len(d))
print('No. of Negative Reviews of 10 Best Foods for You =',len(e))
print('No. of Neutral Reviews of 10 Best Foods for You =',len(f))
print("\nThus by seeing the difference between the no of postitive and negative reviews")
print("\nWe can Say that the Users like such App")
print("\nThus it is advisable to launch an app like 10 Best Foods for You")
```

Code for GUI:

```
 def review():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("REVIEW SECTION")
    adjustWindow(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="REVIEW", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='No. of Positive Reviews of 10 Best Foods for You = 162\n\nNo. of Negative
Reviews of 10 Best Foods for You = 10 \n\nNo. of Neutral Reviews of 10 Best Foods for You = 22
\n\nThus by seeing the difference between the no of postitive and negative reviews \n\nWe can Say
```

that the Users like such App\n\nThus it is advisable to launch an app like 10 Best Foods for You',width='80',height='30' ,font=("Helvetica",10, 'bold', 'italic'), fg='white', bg='blue').place(x=0, y=100)

31. Sentiment Subjectivity vs Sentiment Polarity



Code for Graph:

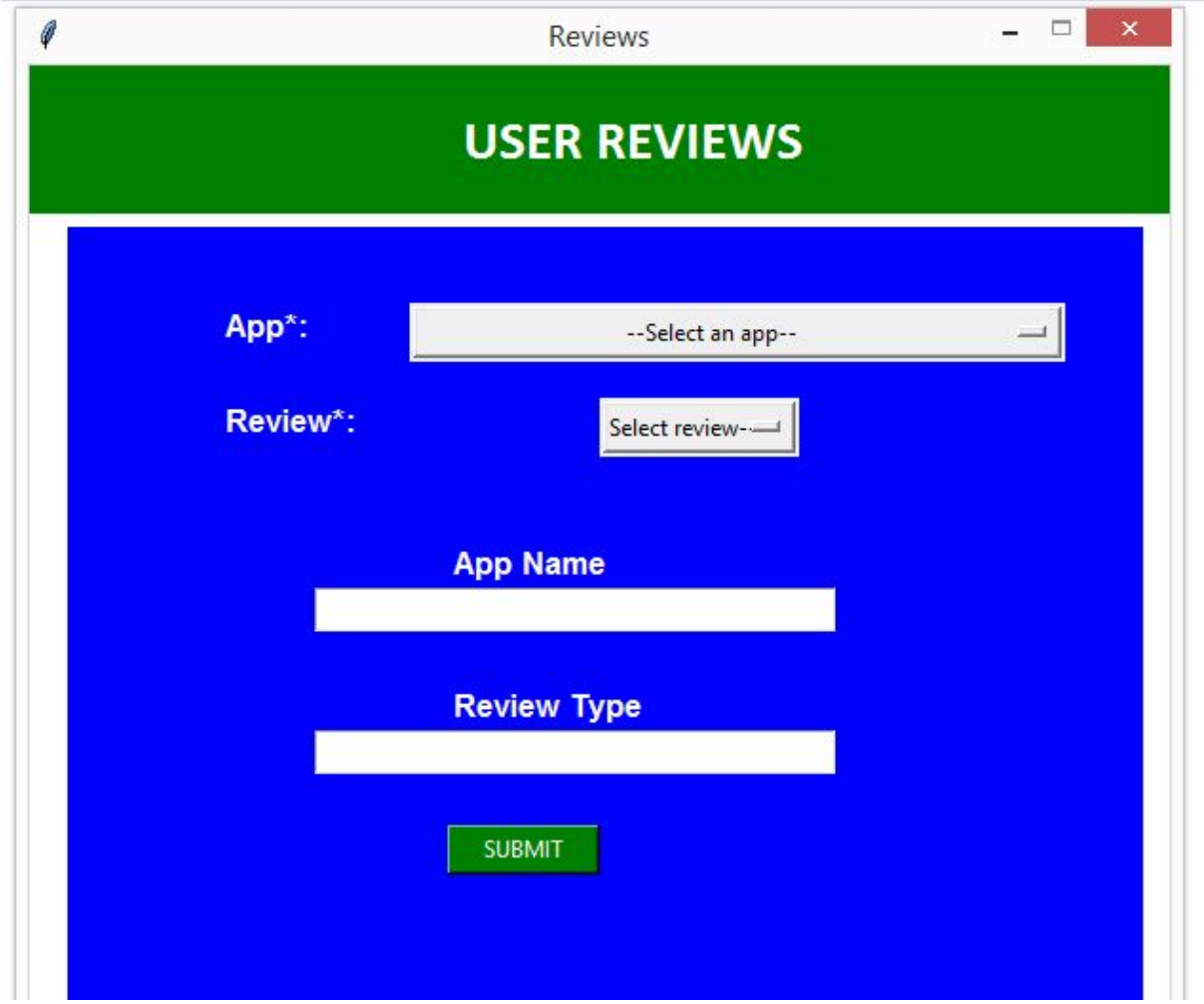
```
def sentimentPolAndSub():
#dropping unnecessary rows
    df2=pd.read_csv("C:\\Users\\Siddhesh\\Desktop\\jayesh\\Database 2 (2)")
    df2=df2.dropna()
#plotting scatterplot of sentiment polarity and sentiment subjectivity
    plt.figure(figsize=(12,6))
    plt.scatter(df2['Sentiment_Polarity'],df2['Sentiment_Subjectivity'],c='blue')
    plt.xlabel("Sentiment_Polarity")
    plt.ylabel("Sentiment_Subjectivity")
    plt.show()
#training data for linear regression
    X=df2['Sentiment_Polarity'].values.reshape(-1,1)
    Y=df2['Sentiment_Subjectivity'].values.reshape(-1,1)
    reg=LinearRegression()
    reg.fit(X,Y)
#plotting regression line
    predictions=reg.predict(X)
    plt.figure(figsize=(10,8))
    plt.scatter(df2['Sentiment_Polarity'],df2['Sentiment_Subjectivity'],c='blue')
    plt.plot(df2['Sentiment_Polarity'],predictions,c='red',linewidth=2)
    plt.xlabel("Sentiment_Polarity")
    plt.ylabel("Sentiment_Subjectivity")
    plt.show()
```

Code for GUI:

```
def sentiment():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("REVIEW SECTION")
    adjustWindow1(screen5)
```

```
  # screen5.resizable(True,True)
    Label(screen5, text="SENTIMENT SUBJECTIVITY VS SENTIMENT POLARITY", width =
'130',height ='2', font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    photo = PhotoImage(file="sentimentpolandsub.png") # opening left side image - Note: If image is
in same folderthen no need to mention the full path
    label = Label(screen5, image=photo ,text="") # attaching image to the label
    label.place(x=500, y=270)
    label.image = photo
```

32. User Reviews



Code:

```
def reg():
    global screen1,screen2,university,review
    screen1 = Toplevel(screen)
    screen1.title("Reviews")
    Label(screen1,text ="",bg="blue", width='80',height='30').place(x=20, y=85)
    Label(screen1, text="USER REVIEWS",width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    university = StringVar()
    review=StringVar()
    adjustWindow(screen1)
    list2=['positive','negative','neutral']
```

```
    droplist = OptionMenu(screen1, university, *list1)
    droplist.config(width=50)
    university.set('--Select an app--')
    Label(screen1,text="App*:",font=("Open Sans", 13, 'bold'),bg='blue',
fg='white').place(x=100,y=125)
    droplist.place(x=200, y=125)
    droplist1 = OptionMenu(screen1, review, *list2)
    droplist1.config(width=10)
    review.set('--Select review--')
    Label(screen1,text="Review*:",font=("Open Sans", 13, 'bold'),bg='blue',
fg='white').place(x=100,y=175)
    droplist1.place(x=300, y=175)
    L1=Label(screen1,text="App Name",font=("Open Sans", 13, 'bold'),bg='blue', fg='white')
    L1.place(x=220,y=250)
    txt1=Entry(screen1,font=("Open Sans", 13, 'bold'),bg='white', fg='black',width=30)
    txt1.place(x=150,y=275)
    L1=Label(screen1,text="Review Type",font=("Open Sans", 13, 'bold'),bg='blue', fg='white')
    L1.place(x=220,y=325)
    txt2=Entry(screen1,font=("Open Sans", 13, 'bold'),bg='white', fg='black',width=30)
    txt2.place(x=150,y=350)
    def retrieve_input():
        inputValue=txt1.get()
        app=inputValue
        inputValue=txt2.get()
        r=inputValue
        data = pd.read_csv("C:\\Users\\Siddhesh\\Desktop\\jayesh\\Database 2")

        def senti(app,r):
            if r=='positive' or r=='POSITIVE' or r=='Positive':
                p=data[(data['Sentiment'] == 'Positive')& (data['App']==app)]['Translated_Review']
                screen2 = Toplevel(screen)
                adjustWindow1(screen2)
                screen2.title("POSITIVE")
               # screen2.resizable(True,True)
                Label(screen2, text="POSITIVE", width = '140',height ='2',
font=('calibri',20,'bold'),fg='white',bg='green').place(x=0,y=0)
                Label(screen2, text=p, width = '200',height ='68',
font=('calibri',9,'bold'),fg='white',bg='blue').place(x=300,y=80)

            if r=='negative' or r=='NEGATIVE' or r=='Negative':
                n= data[(data['Sentiment'] == 'Negative') & (data['App']==app)]['Translated_Review']
                screen2 = Toplevel(screen)
                adjustWindow1(screen2)
                screen2.title("NEGATIVE")
               # screen2.resizable(True,True)
                Label(screen2, text="NEGATIVE", width = '140',height ='2',
font=('calibri',20,'bold'),fg='white',bg='green').place(x=0,y=0)
                Label(screen2, text=n, width = '200',height ='69',
font=('calibri',9,'bold'),fg='white',bg='blue').place(x=300,y=80)
```

```
        if r=='neutral' or r=='NEUTRAL' or r=='Neutral':
            neu= data[(data['Sentiment'] == 'Neutral') & (data['App']==app)]['Translated_Review']
            screen2 = Toplevel(screen)
            adjustWindow1(screen2)
            screen2.title("NEUTRAL")
            #screen2.resizable(True,True)
            Label(screen2, text="NEUTRAL", width = '140',height ='2',
font=('calibri',20,'bold'),fg='white',bg='green').place(x=0,y=0)
            Label(screen2, text=neu, width = '200',height ='69',
font=('calibri',9,'bold'),fg='white',bg='blue').place(x=300,y=80)
        senti(app,r)
    buttonCommit=Button(screen1, height=1, width=10, text="SUBMIT",bg='green',fg='white'
,command=lambda: retrieve_input())#command=lambda: retrieve_input() >>> just means do this
when i press the button
    buttonCommit.place(x=220,y=400)
```

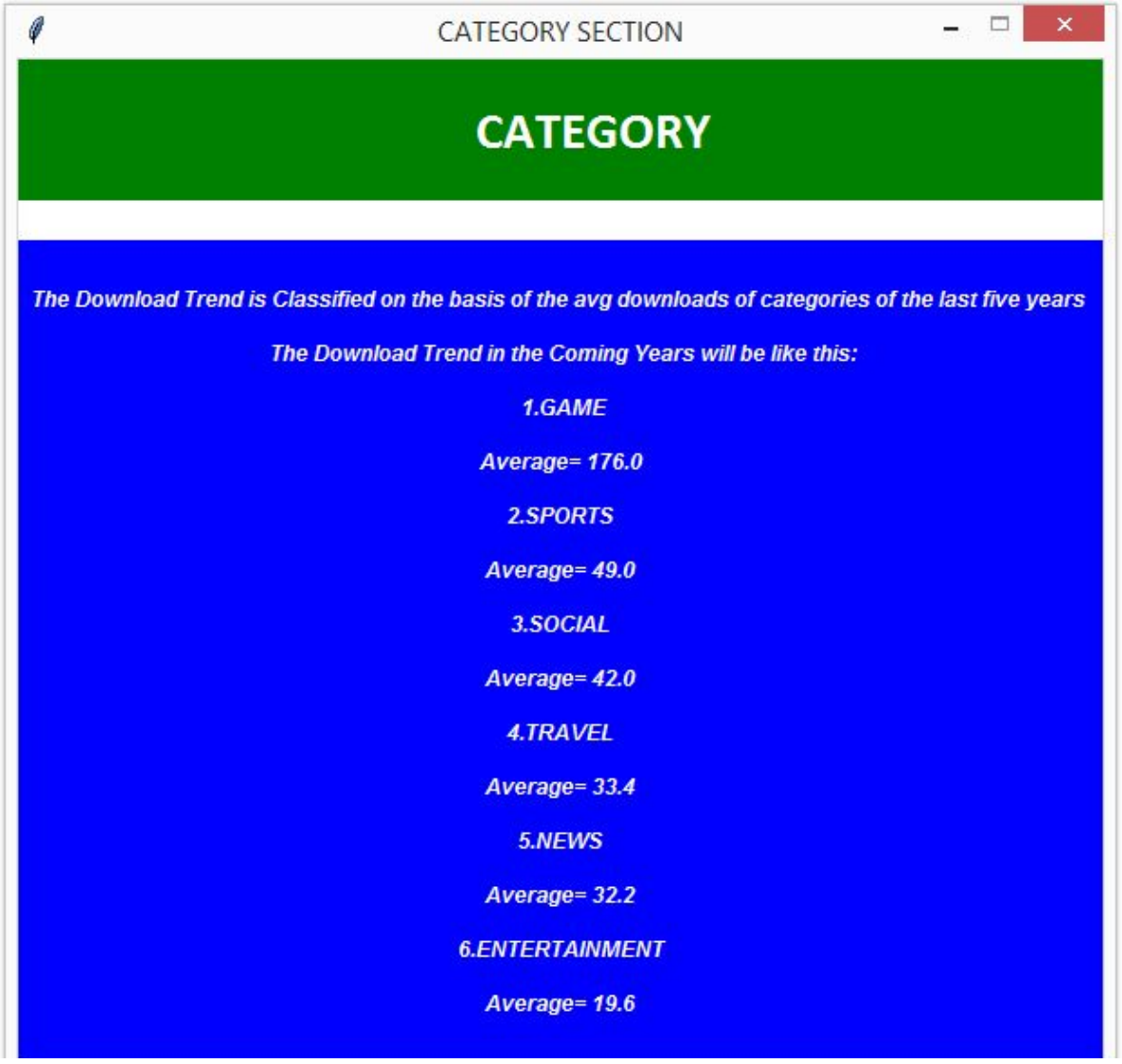
33. Category

Code:

```
def analysis3():
    global screen7
    screen7=Toplevel(screen)
    screen7.title("CATEGORY")
    adjustWindow(screen7)
```

```
    Label(screen7, text="CATEGORY",width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Button(screen7, text='CATEGORY PREDICTION', width=50, font=("Open Sans", 13,
'bold'),bg='blue', fg='white',command=predict).place(x=50, y=100)
    Button(screen7, text='CATEEGORY VS MONTH', width=50, font=("Open Sans", 13,
'bold'),bg='blue', fg='white',command=installs_category).place(x=50, y=150)
    Button(screen7, text='CATEEGORY DOWNLOADS OF 2016', width=50, font=("Open Sans", 13,
'bold'),bg='blue', fg='white',command=downloads_16).place(x=50, y=200)
    Button(screen7, text='CATEEGORY DOWNLOADS OF 2017', width=50, font=("Open Sans", 13,
'bold'),bg='blue', fg='white',command=downloads_17).place(x=50, y=250)
    Button(screen7, text='CATEEGORY DOWNLOADS OF 2018', width=50, font=("Open Sans", 13,
'bold'),bg='blue', fg='white',command=downloads_18).place(x=50, y=300)
    Button(screen7, text='DOWNLOAD HISTORY', width=50, font=("Open Sans", 13,
'bold'),bg='blue', fg='white',command=analysis6).place(x=50, y=350)
    Button(screen7, text='CATEGORY VS AVERAGE RATING', width=50, font=("Open Sans", 13,
'bold'),bg='blue', fg='white',command=cat3).place(x=50, y=400)
```

34. Download Trend



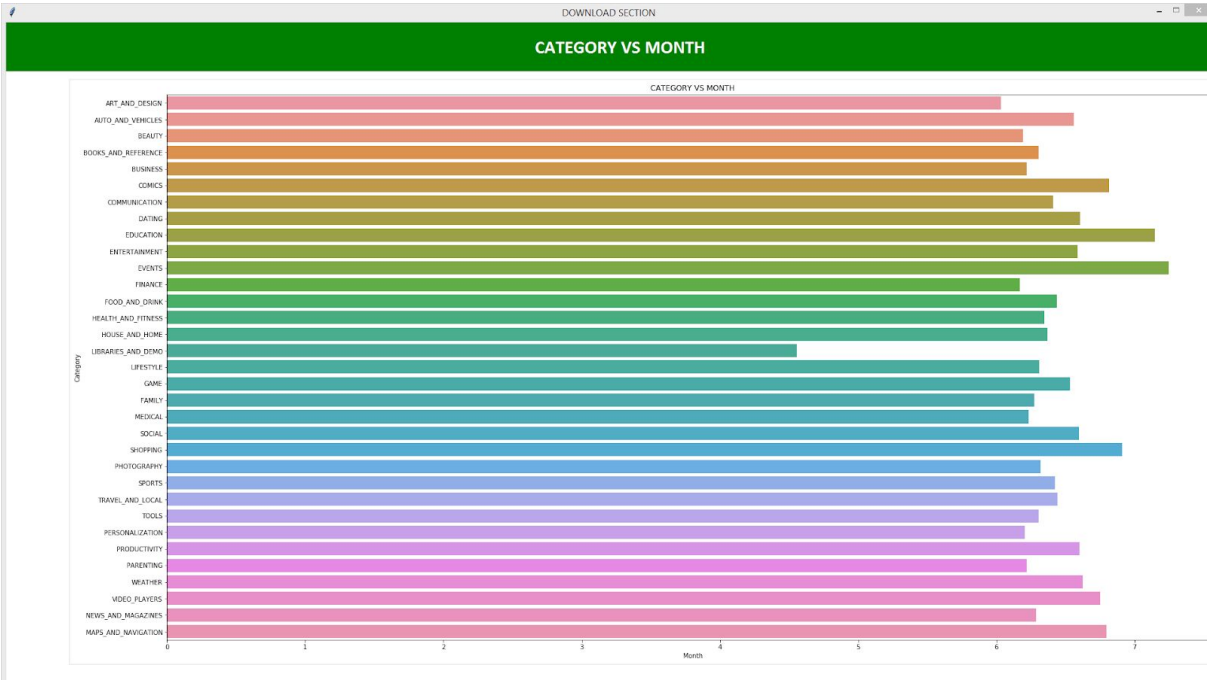
Code for Information:

```
category=['GAME','SPORTS','SOCIAL','EVENTS','TRAVEL_AND_LOCAL','NEWS_AND_MAG
AZINES','ENTERTAINMENT']
year=[2014,2015,2016,2017,2018]
for i in category:
    sum1=0
    avg=0
    count=0
    print("Category=",i)
    for j in year:
        print("Year=",j)
        play = df[(df['Category']==i) &(df['Installs']>='1,000,000,000+') &(df['year'] == j)]['App']
        print(play.count())
        count=play.count()
        sum1=sum1+count
    avg=sum1/5
    print("Average=",avg)
avg=[176.0,49.0,42.0,33.4,32.2,19.6,7.0]
print("The Download Trend is Classified on the basis of the avg downloads of categories of the last
five years")
print("The Download Trend in the Coming Years will be like this:")
print("1.GAME")
print("Average=",avg[0])
print("2.SPORTS")
print("Average=",avg[1])
print("3.SOCIAL")
print("Average=",avg[2])
print("4.TRAVEL")
print("Average=",avg[3])
print("5.NEWS")
print("Average=",avg[4])
print("6.ENTERTAINMENT")
print("Average=",avg[5])
```

Code for GUI:

```
def predict():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("CATEGORY SECTION")
    adjustWindow(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="CATEGORY", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='The Download Trend is Classified on the basis of the avg downloads of
categories of the last five years \n\n The Download Trend in the Coming Years will be like this:\n\n
1.GAME\n\nAverage= 176.0\n\n2.SPORTS\n\nAverage= 49.0\n\n3.SOCIAL\n\nAverage=
42.0\n\n4.TRAVEL\n\nAverage= 33.4\n\n5.NEWS\n\nAverage=
32.2\n\n6.ENTERTAINMENT\n\nAverage= 19.6',width='85',height='30' ,font=("Helvetica",9, 'bold',
'italic'), fg='white', bg='blue').place(x=0, y=100)
```

35. Category vs Month



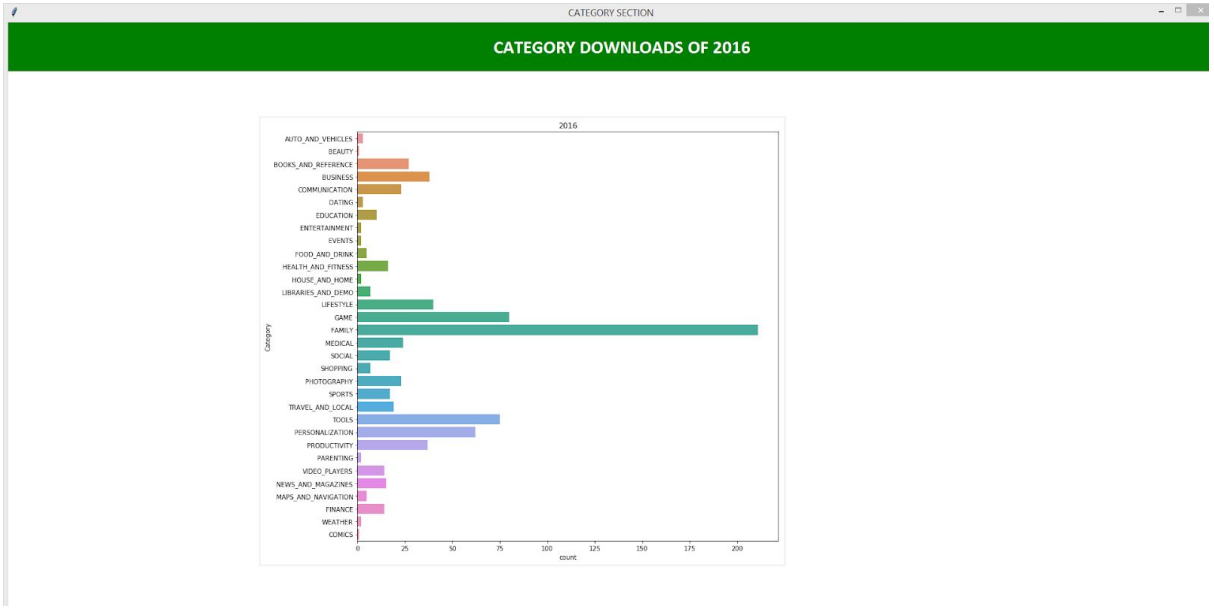
Code for Graph:

```
plt.figure(figsize=(30,16))
sns.barplot(y='Category',x=df['Month'],data=df,ci=None)
plt.title("CATEGORY VS MONTH")
plt.show()
```

Code for GUI:

```
def installs_category():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("DOWNLOAD SECTION")
    adjustWindow1(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="CATEGORY VS MONTH", width = '130',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    photo = PhotoImage(file="Category vs Month.png") # opening left side image - Note: If image is in
same folderthen no need to mention the full path
    label = Label(screen5, image=photo ,text="") # attaching image to the label
    label.place(x=100, y=90)
    label.image = photo
```

36. Category Downloads of 2016



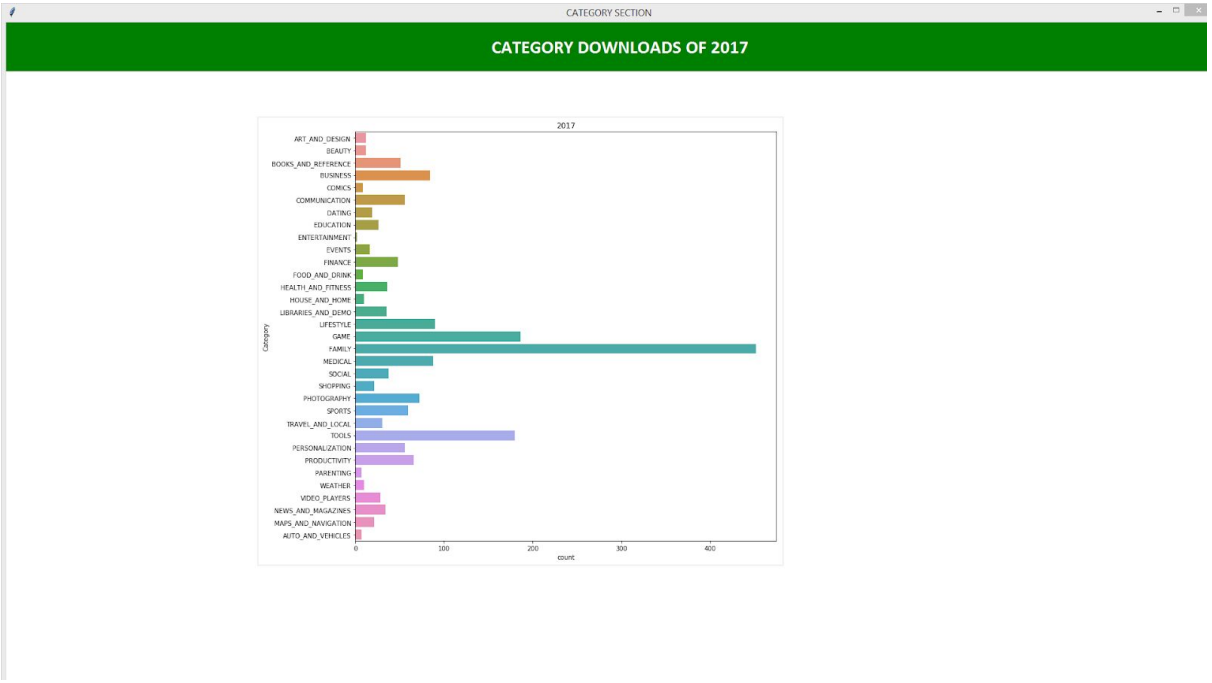
Code for graph:

```
data1 = data[data["Last Updated"].str.contains('2016')]
plt.figure(figsize=(12,6))
sns.countplot(y="Category",data = data1)
plt.title("2016")
plt.show()
```

Code for GUI:

```
def downloads_16():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("CATEGORY SECTION")
    adjustWindow1(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="CATEGORY DOWNLOADS OF 2016", width = '130',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    photo = PhotoImage(file="2016.png") # opening left side image - Note: If image is in same
folderthen no need to mention the full path
    label = Label(screen5, image=photo ,text="") # attaching image to the label
    label.place(x=400, y=150)
    label.image = photo
```

37. Category Downloads of 2017



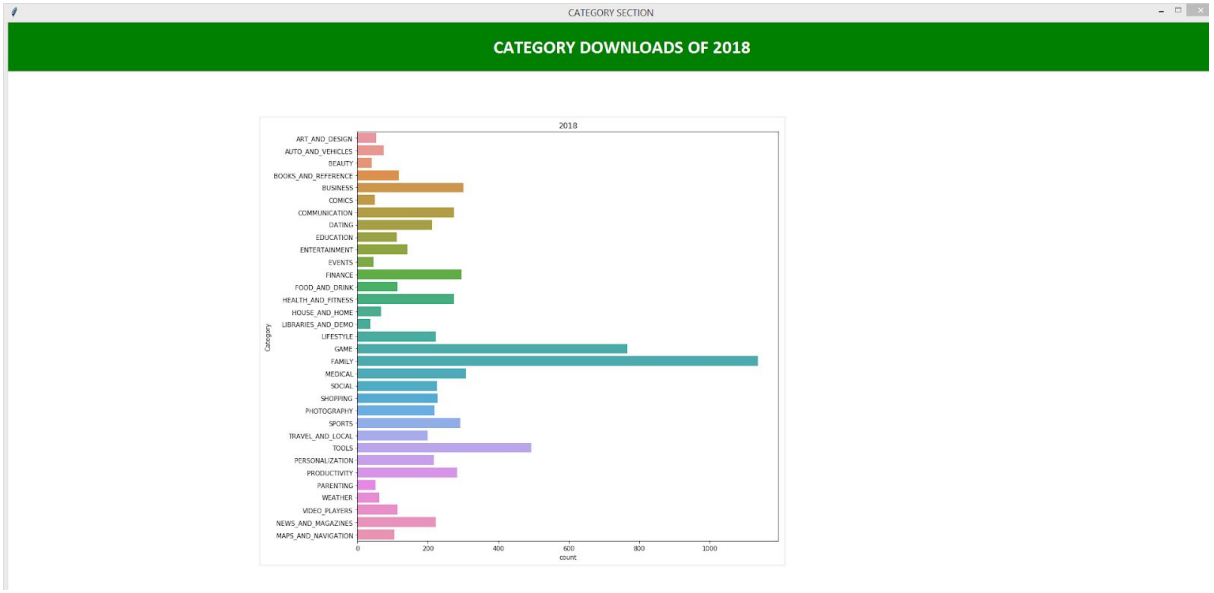
Code for Graph:

```
data2 = data[data["Last Updated"].str.contains('2017')]
plt.figure(figsize=(12,6))
sns.countplot(y="Category",data = data2)
plt.title("2017")
plt.show()
```

Code for GUI:

```
def downloads_17():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("CATEGORY SECTION")
    adjustWindow1(screen5)
  #  screen5.resizable(True,True)
    Label(screen5, text="CATEGORY DOWNLOADS OF 2017", width = '130',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    photo = PhotoImage(file="2017.png") # opening left side image - Note: If image is in same
folderthen no need to mention the full path
    label = Label(screen5, image=photo ,text="") # attaching image to the label
    label.place(x=400, y=150)
    label.image = photo
```

38. Category Downloads of 2018



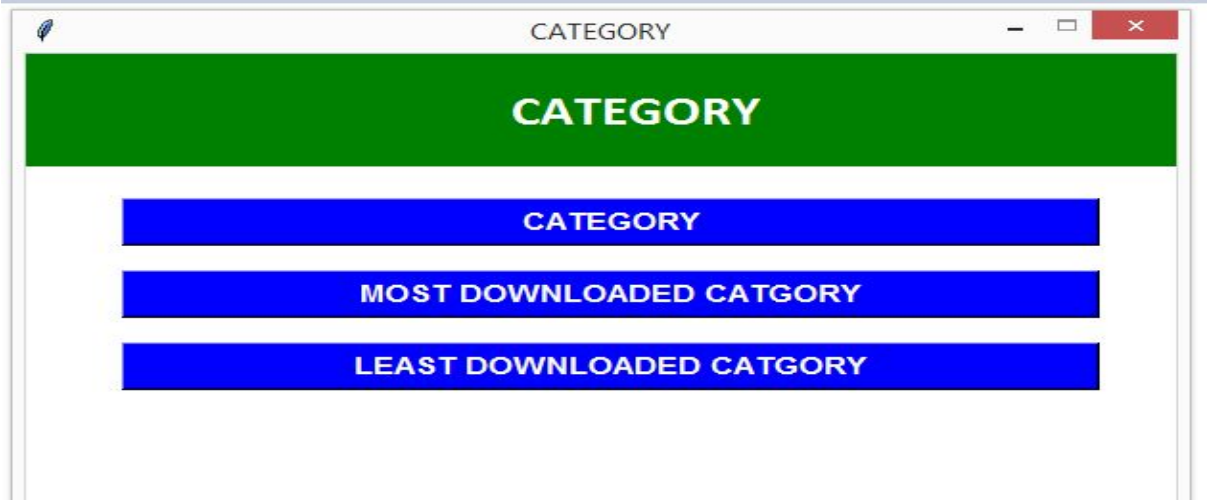
Code for Graph:

```
data3 = data[data["Last Updated"].str.contains('2018')]
plt.figure(figsize=(12,6))
sns.countplot(y="Category",data = data3)
plt.title("2018")
plt.show()
```

Code for GUI

```
def downloads_18():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("CATEGORY SECTION")
    adjustWindow1(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="CATEGORY DOWNLOADS OF 2018", width = '130',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    photo = PhotoImage(file="2018.png") # opening left side image - Note: If image is in same
folderthen no need to mention the full path
    label = Label(screen5, image=photo ,text="") # attaching image to the label
    label.place(x=400, y=150)
    label.image = photo
```

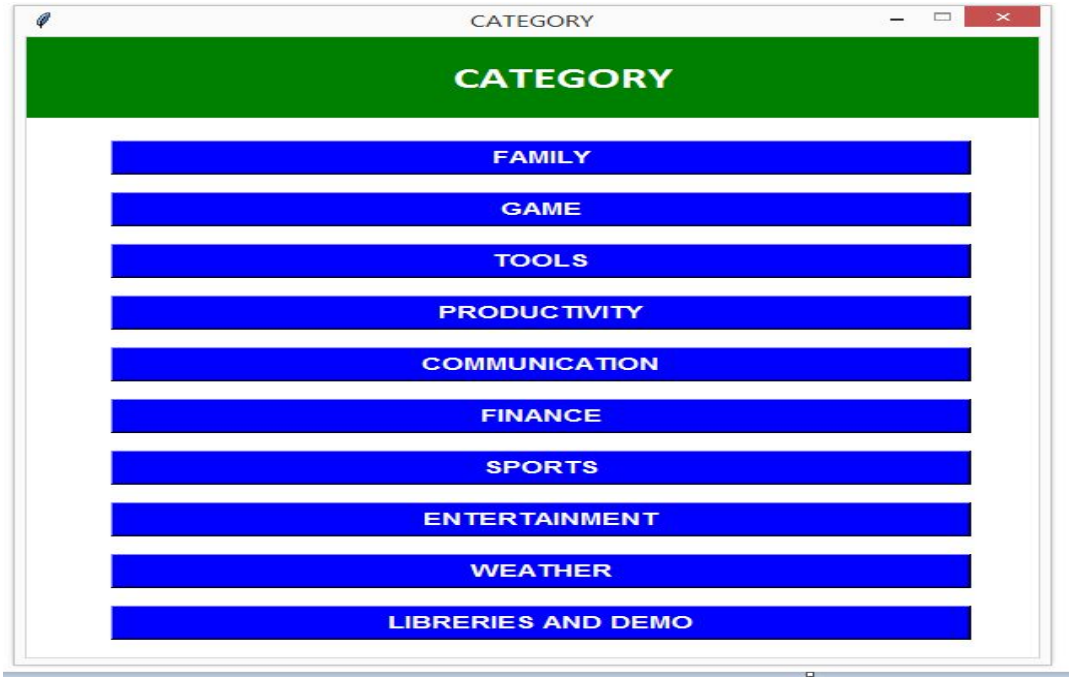
39. Download History



Code:

```
def analysis6():
    global screen7
    screen7=Toplevel(screen)
    screen7.title("CATEGORY")
    adjustWindow(screen7)
    Label(screen7, text="CATEGORY",width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Button(screen7, text='CATEGORY', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=catl).place(x=50, y=100)
    Button(screen7, text='MOST DOWNLOADED CATGORY', width=50, font=("Open Sans", 13,
'bold'),bg='blue', fg='white',command=downloads_amt).place(x=50, y=150)
    Button(screen7, text='LEAST DOWNLOADED CATGORY', width=50, font=("Open Sans", 13,
'bold'),bg='blue', fg='white',command=downloads_amtl).place(x=50, y=200)
```

40. List of Categories



Code for Information:

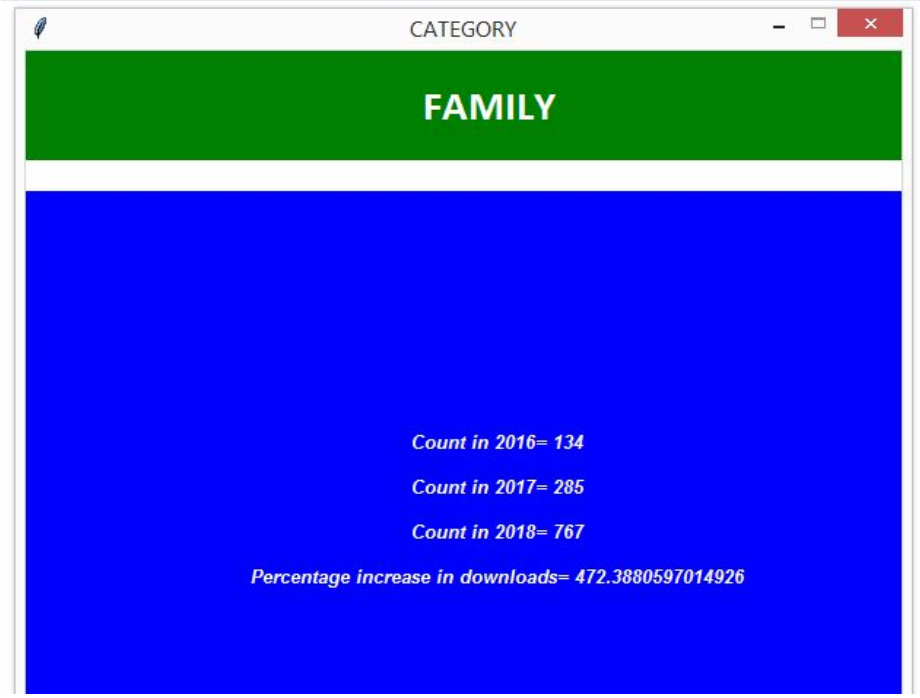
```
df2 = df['Category'].value_counts()
df2 = df2.reset_index()
df2 = df2[df2['Category']>1]
genres=  list(df2['index'])
d = pd.DatetimeIndex(df['Last_Updated'])
df['year'] = d.year
df['month'] = d.month
d=50
e=100
f=100
p=100
s=100
t=100
for i in genres:

    play = df[(df['Category']==i) &(df['Installs']>='1,000,000,000+') &(df['year'] == 2016)]['App']
    play1 = df[(df['Category']==i) & (df['Installs']>='1,000,000,000+') & (df['year'] == 2017)]['App']
    play2 = df[(df['Category']==i) & (df['Installs']>='1,000,000,000+') & (df['year'] == 2018)]['App']
    print(i)
    print("Count in 2016=",play.count())
    print("Count in 2017=",play1.count())
    print("Count in 2018=",play2.count())
    six=play.count()
    sev=play1.count()
    eit=play2.count()
    if eit>six and six!=0:
        one=eit-six
        two=one/six
        three=two*100
        print("Percentage increase in downloads=",three)

    elif six==0:
        six=six+1
        one=eit-six
        two=one/six
        three=(two+1)*100
        print("Percentage increase in downloads=",three)


    else:
        one=six-eit
        two=one/six
        three=two*100
        print("Percentage increase in downloads=",three)
```

Code for GUI:

```
def catl():
    global screen7
    screen7=Toplevel(screen)
```

```
    screen7.title("CATEGORY")
    adjustWindow(screen7)
    Label(screen7, text="CATEGORY",width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Button(screen7, text='FAMILY', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=fam).place(x=50, y=100)
    Button(screen7, text='GAME', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=game).place(x=50, y=150)
    Button(screen7, text='TOOLS', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=tools1).place(x=50, y=200)
    Button(screen7, text='PRODUCTIVITY', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=pro1).place(x=50, y=250)
    #Button(screen7, text='MEDICAL', width=50, font=("Open Sans", 13, 'bold'),bg='brown',
fg='white',command=pro).place(x=50, y=300)
    Button(screen7, text='COMMUNICATION', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=comm).place(x=50, y=300)
    Button(screen7, text='FINANCE', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=fin1).place(x=50, y=350)
    Button(screen7, text='SPORTS', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=sports1).place(x=50, y=400)
    Button(screen7, text='ENTERTAINMENT', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=enter1).place(x=50, y=450)
    Button(screen7, text='WEATHER', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=wet1).place(x=50, y=500)
    Button(screen7, text='LIBRERIES AND DEMO', width=50, font=("Open Sans", 13,
'bold'),bg='blue', fg='white',command=lib1).place(x=50, y=550)
```

41. Family

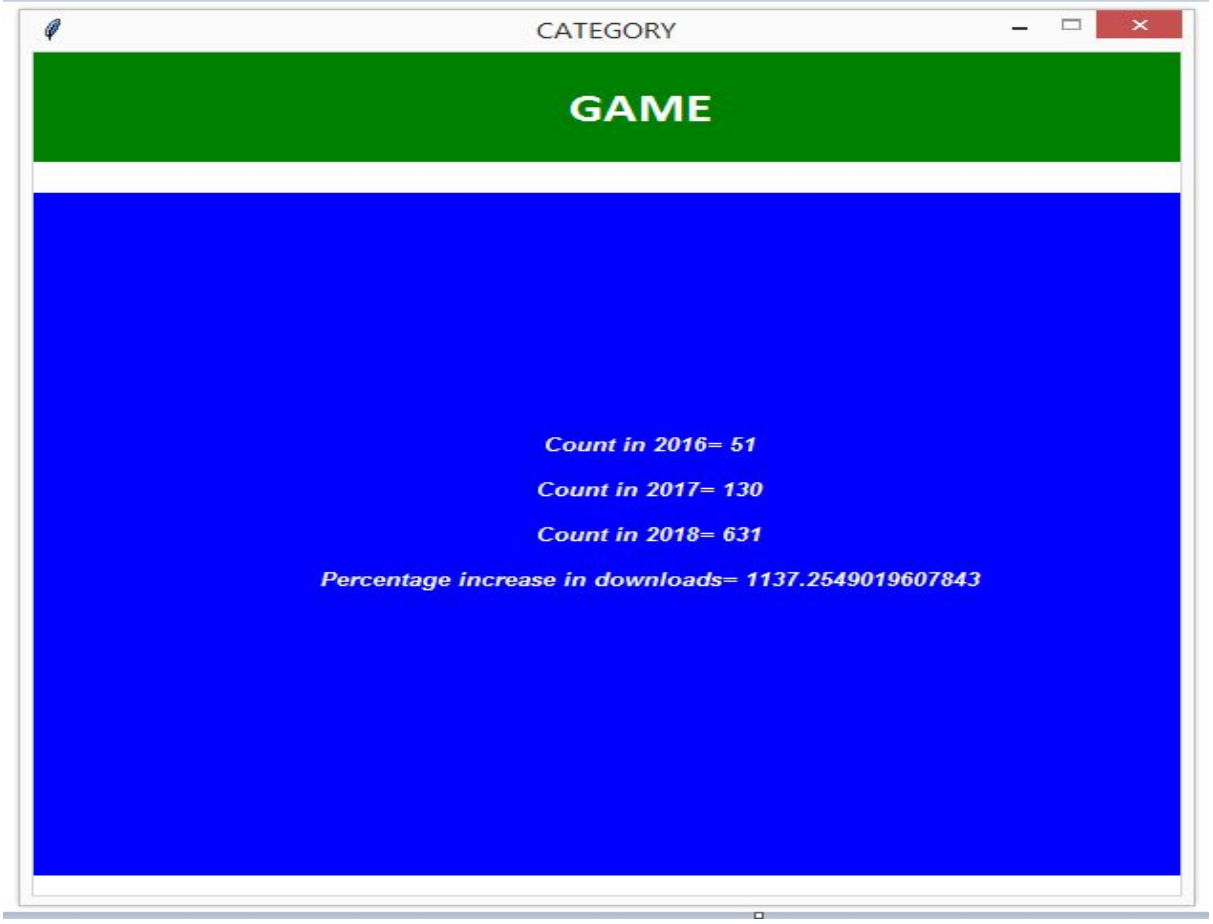


Code:

```
def fam():
    global screen5
```

```
    screen5=Toplevel(screen)
    screen5.title("CATEGORY")
    adjustWindow(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="FAMILY", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='Count in 2016= 134\n\nCount in 2017= 285\n\nCount in 2018=
767\n\nPercentage increase in downloads= 472.3880597014926\n\n',width='80',height='30'
,font=("Helvetica",10, 'bold', 'italic'), fg='white', bg='blue').place(x=0, y=100)
```

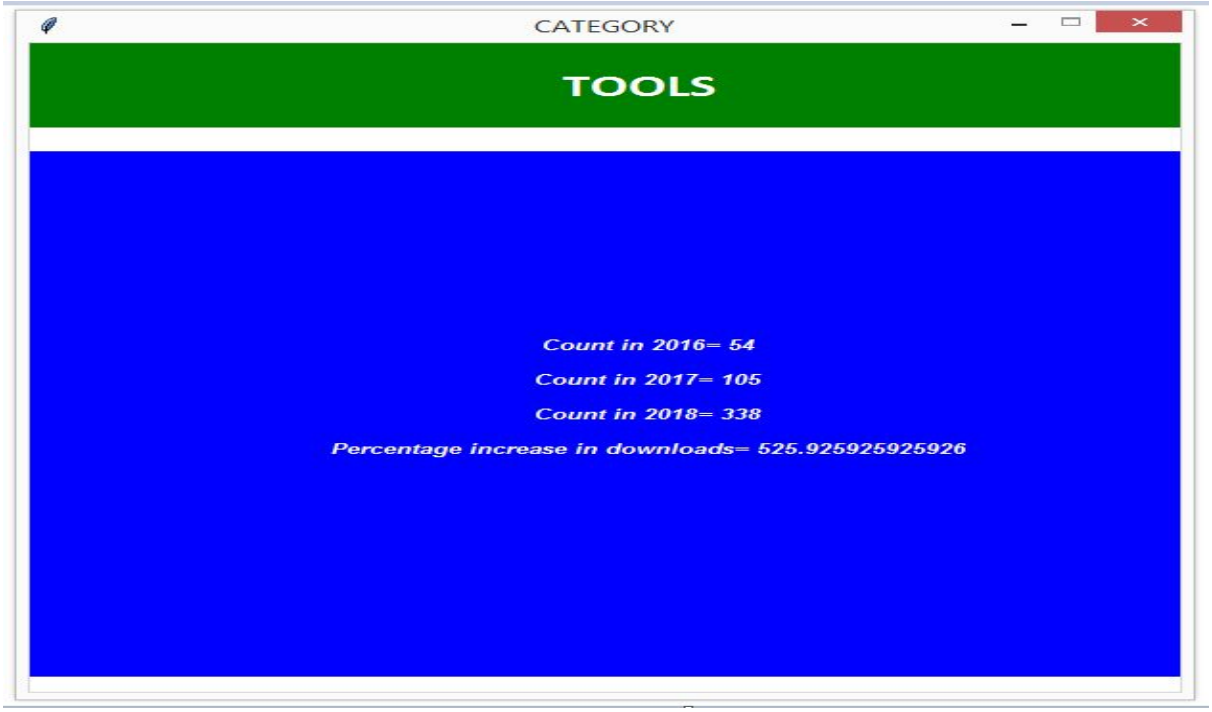
42. Game



Code:

```
def game():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("CATEGORY")
    adjustWindow(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="GAME", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='Count in 2016= 51\n\nCount in 2017= 130\n\nCount in 2018=
631\n\nPercentage increase in downloads= 1137.2549019607843\n\n',width='80',height='30'
,font=("Helvetica",10, 'bold', 'italic'), fg='white', bg='blue').place(x=0, y=100)
```

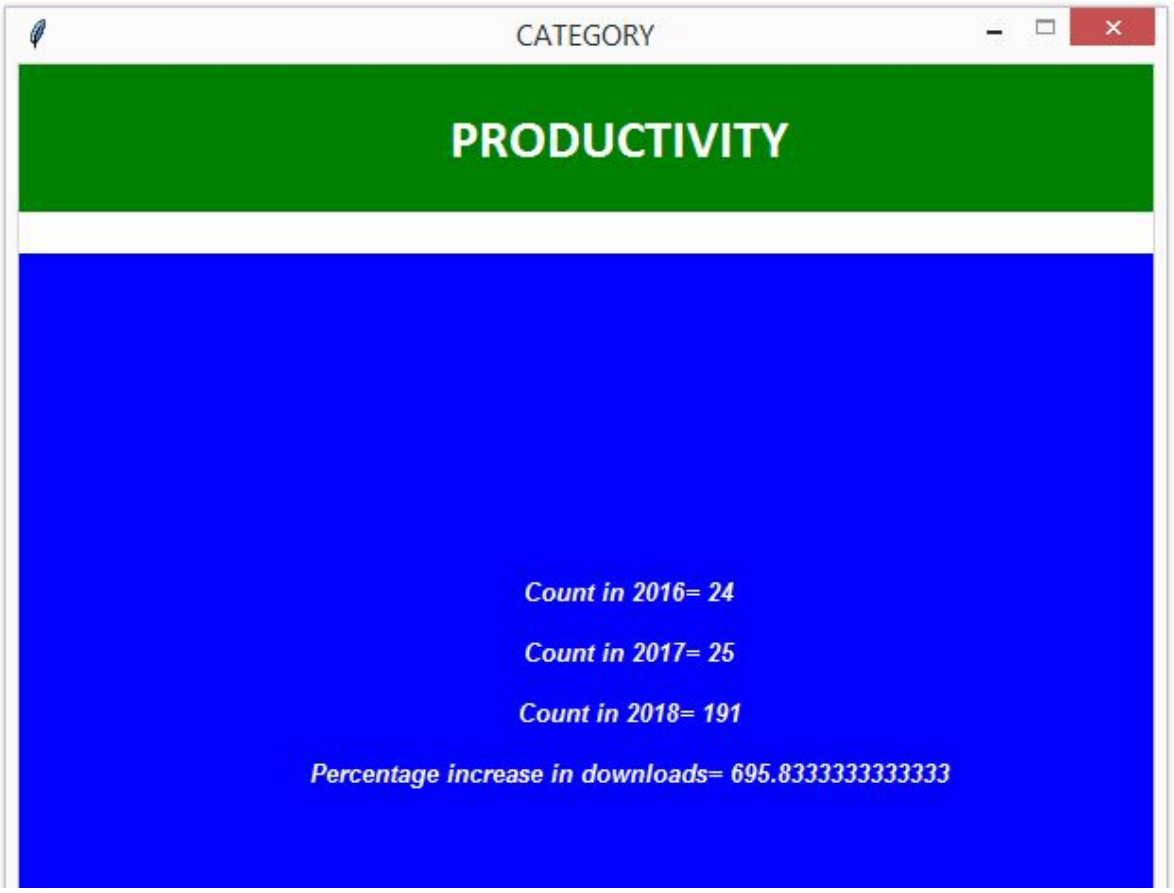
43. Tools



Code:

```
def tools1():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("CATEGORY")
    adjustWindow(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="TOOLS", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='Count in 2016= 54\n\nCount in 2017= 105\n\nCount in 2018=
338\n\nPercentage increase in downloads= 525.925925925926\n\n',width='80',height='30'
,font=("Helvetica",10, 'bold', 'italic'), fg='white', bg='blue').place(x=0, y=100)
```

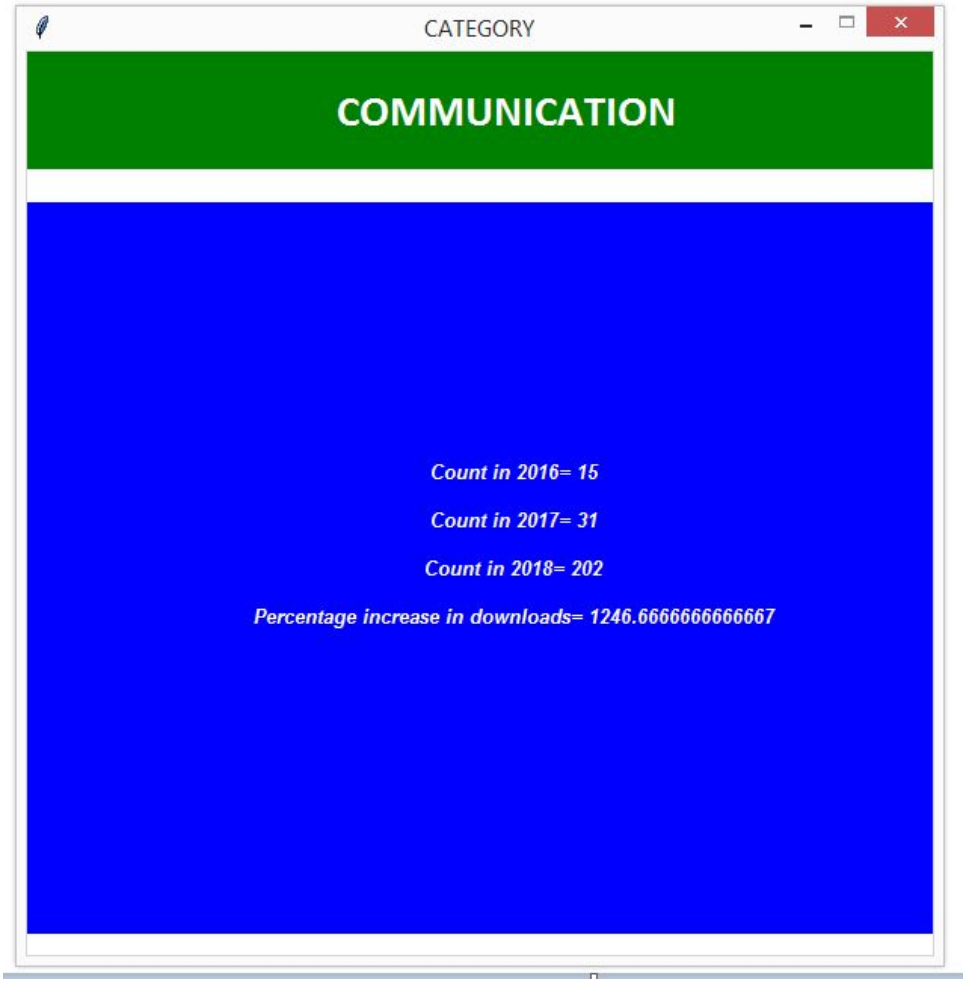
44. Productivity



Code:

```
def pro1():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("CATEGORY")
    adjustWindow(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="PRODUCTIVITY", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='Count in 2016= 24\n\nCount in 2017= 25\n\nCount in 2018=
191\n\nPercentage increase in downloads= 695.8333333333333\n\n'
,width='80',height='30',font=("Helvetica",10, 'bold', 'italic'), fg='white', bg='blue').place(x=0, y=100)
```

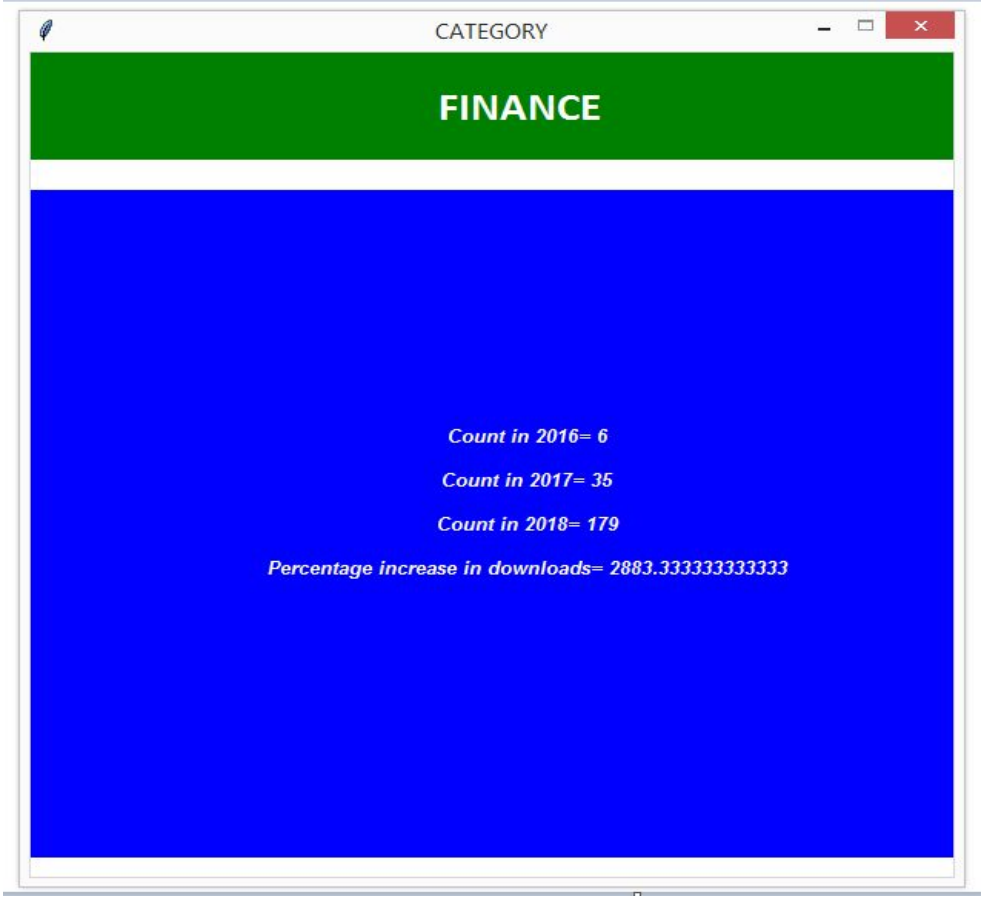
45. Communication



Code:

```
def comm():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("CATEGORY")
    adjustWindow(screen5)
    Label(screen5, text="COMMUNICATION", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='Count in 2016= 15\n\nCount in 2017= 31\n\nCount in 2018=
202\n\nPercentage increase in downloads= 1246.6666666666667\n\n',width='80',height='30'
,font=("Helvetica",10, 'bold', 'italic'), fg='white', bg='blue').place(x=0, y=100)
```

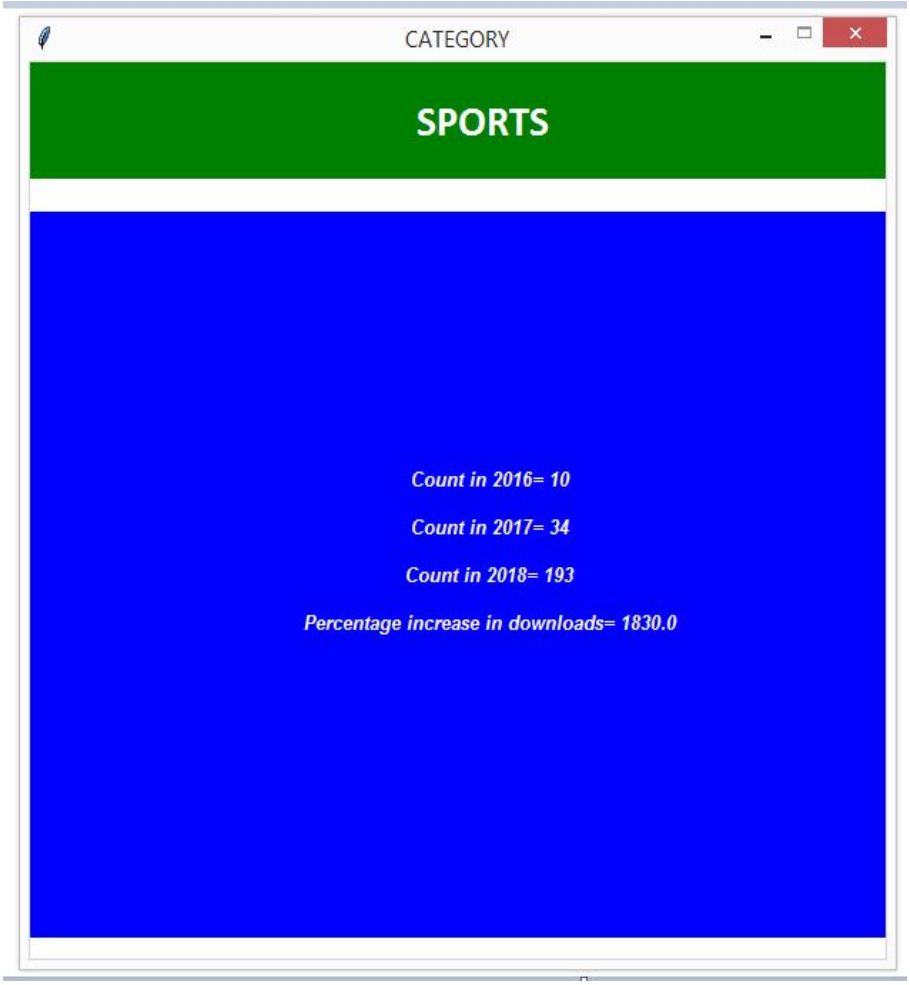
46. Finance



Code:

```
def fin1():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("CATEGORY")
    adjustWindow(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="FINANCE", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='Count in 2016= 6\n\nCount in 2017= 35\n\nCount in 2018=
179\n\nPercentage increase in downloads= 2883.333333333333\n\n',width='80',height='30'
,font=("Helvetica",10, 'bold', 'italic'), fg='white', bg='blue').place(x=0, y=100)
```

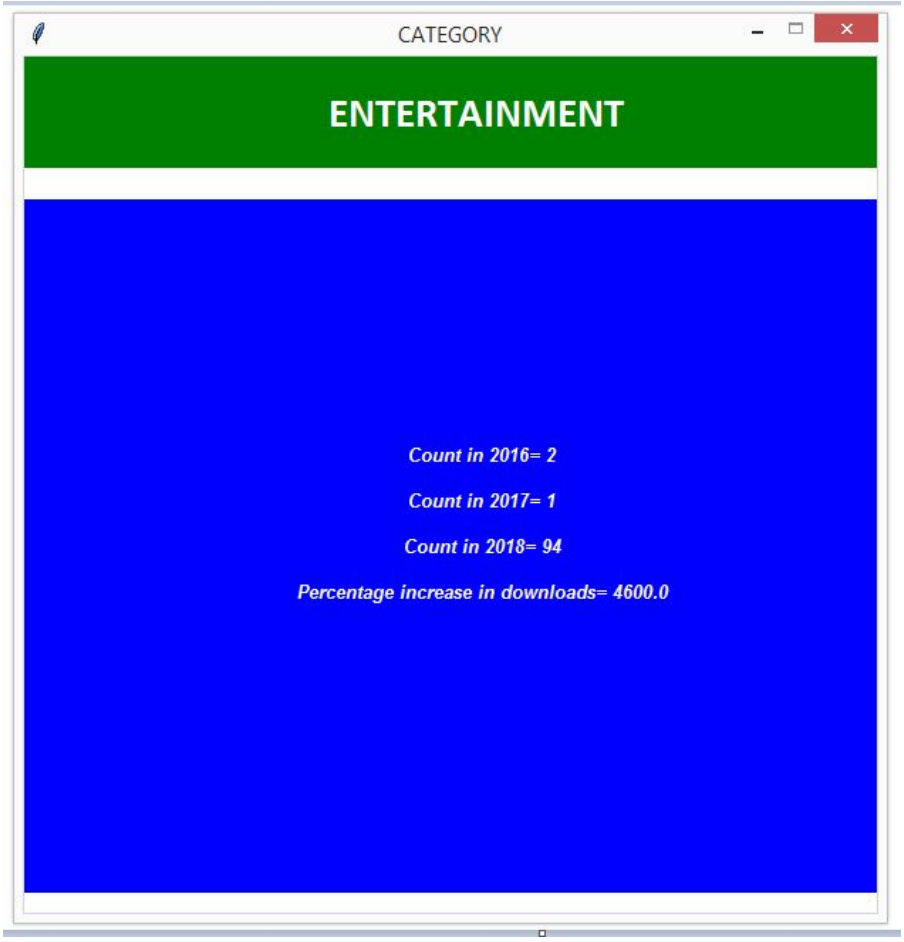
47. Sports



Code:

```
def sports1():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("CATEGORY")
    adjustWindow(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="SPORTS", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='Count in 2016= 10\n\nCount in 2017= 34\n\nCount in 2018=
193\n\nPercentage increase in downloads= 1830.0\n\n' ,width='80',height='30',font=("Helvetica",10,
'bold', 'italic'), fg='white', bg='blue').place(x=0, y=100)
```

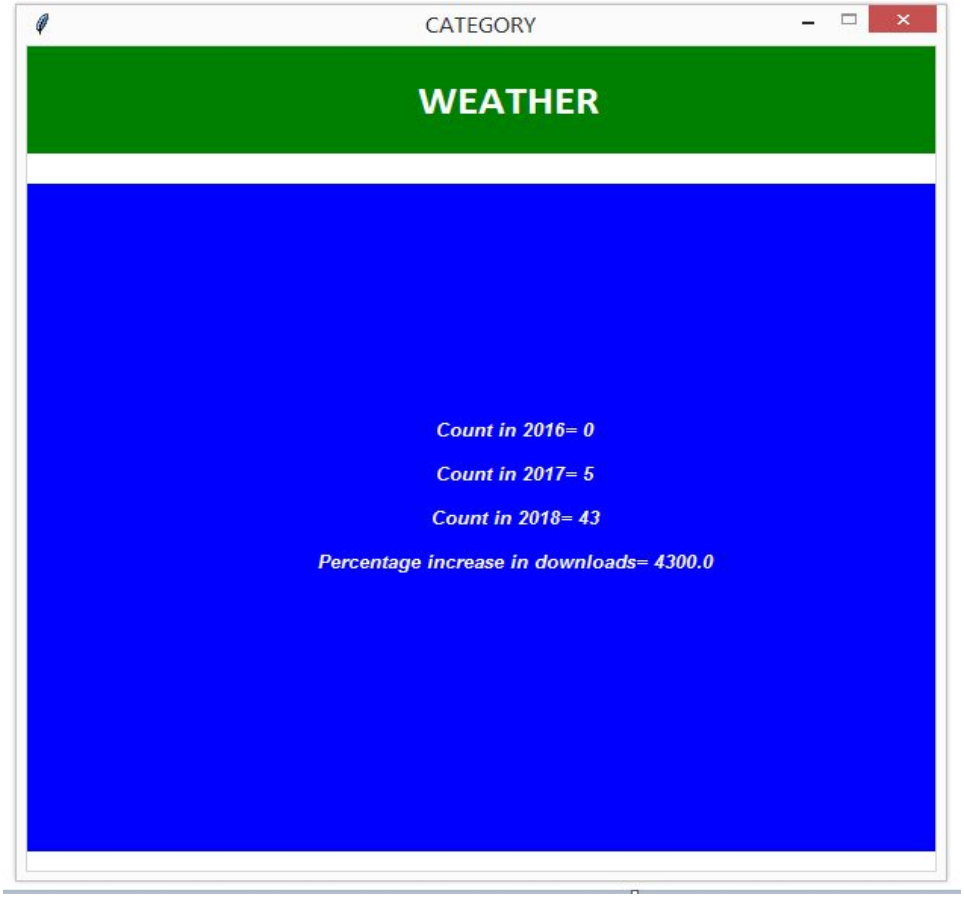
48. Entertainment



Code:

```
def enter1():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("CATEGORY")
    adjustWindow(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="ENTERTAINMENT", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='Count in 2016= 2\n\nCount in 2017= 1\n\nCount in 2018= 94\n\nPercentage
increase in downloads= 4600.0\n\n',width='80',height='30' ,font=("Helvetica",10, 'bold', 'italic'),
fg='white', bg='blue').place(x=0, y=100)
```

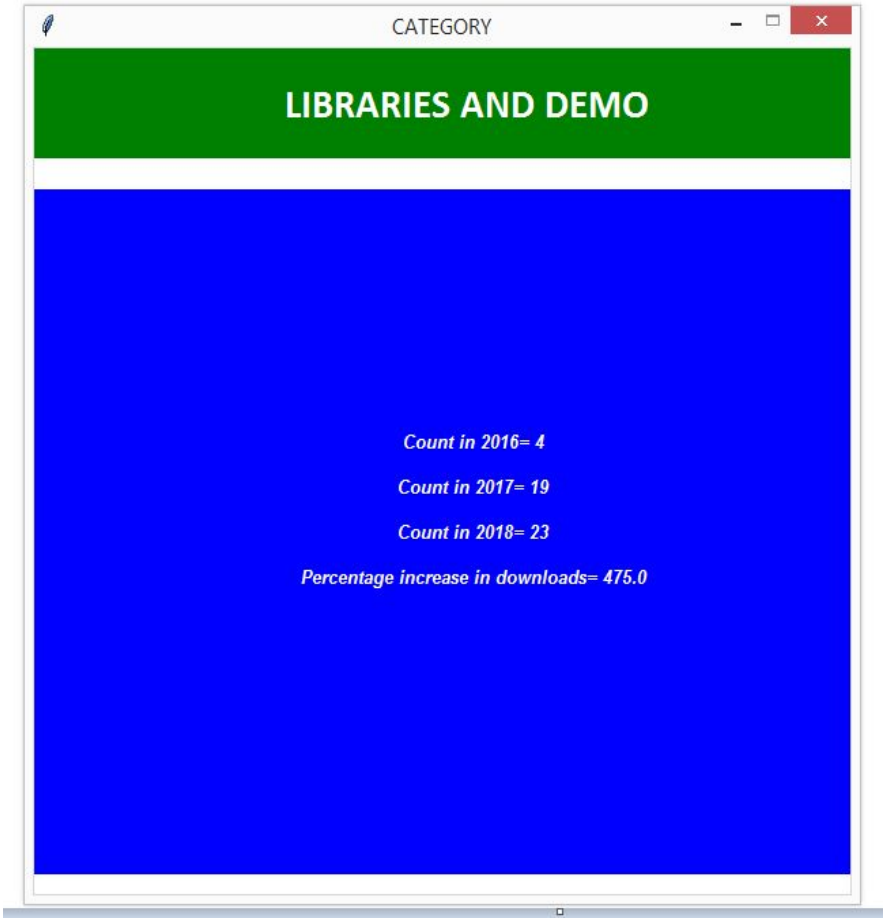
49. Weather



Code:

```
def wet1():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("CATEGORY")
    adjustWindow(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="WEATHER", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='Count in 2016= 0\n\nCount in 2017= 5\n\nCount in 2018= 43\n\nPercentage
increase in downloads= 4300.0\n\n',width='80',height='30' ,font=("Helvetica",10, 'bold', 'italic'),
fg='white', bg='blue').place(x=0, y=100)
```

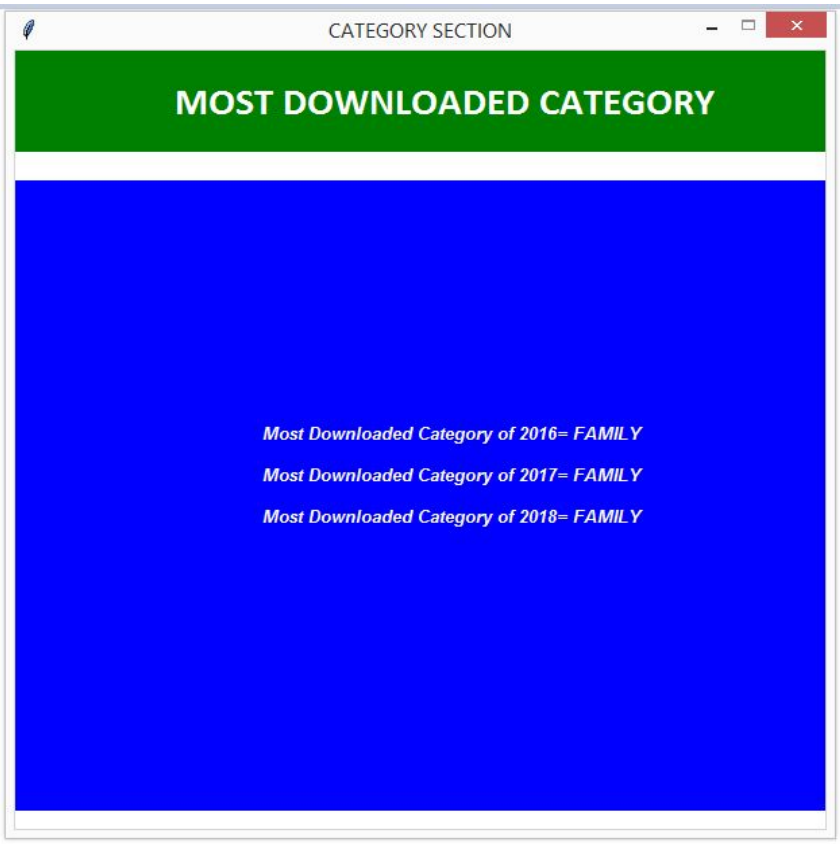
50. Libraries and Demo



Code:

```
def lib1():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("CATEGORY")
    adjustWindow(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="LIBRARIES AND DEMO", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='Count in 2016= 4\n\nCount in 2017= 19\n\nCount in 2018= 23\n\nPercentage
increase in downloads= 475.0\n\n',width='80',height='30' ,font=("Helvetica",10, 'bold', 'italic'),
fg='white', bg='blue').place(x=0, y=100)
```

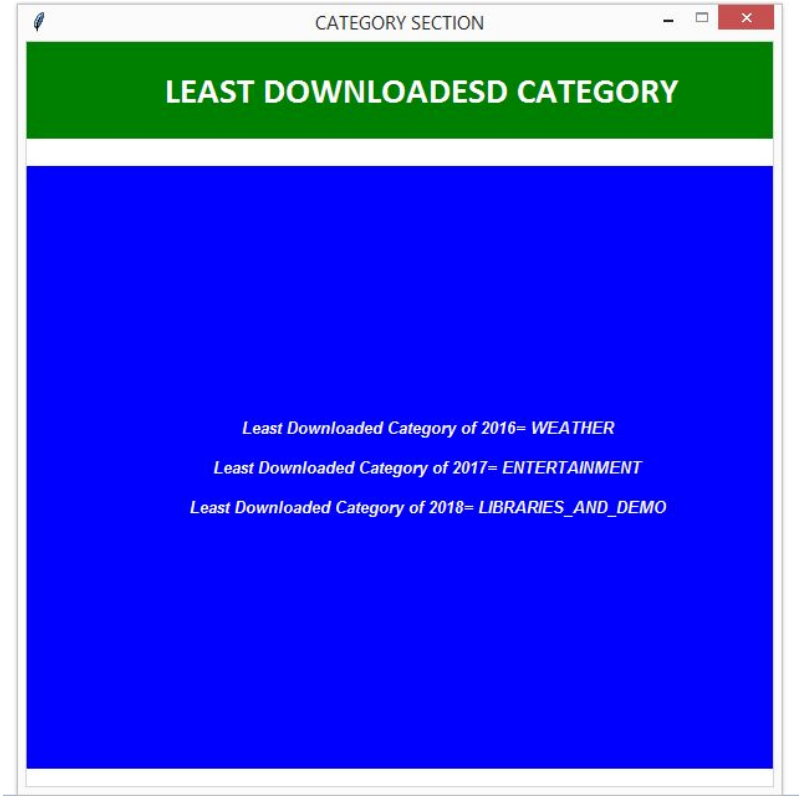
51. Most Downloaded Category



Code for Information :

```
if play.count()>=d:
       d=play.count()
       cat16=i
    if play1.count()>=e:
        e=play1.count()
        cat17=i
    if play2.count()>=f:
        f=play2.count()
        cat18=i
    if play.count()<p:
       p=play.count()
       ca16=i
    if play1.count()<s:
        s=play1.count()
        ca17=i
    if play2.count()<t:
        t=play2.count()
        ca18=i
print("Most Downloaded Category of 2016=",cat16)
print("Most Downloaded Category of 2017=",cat17)
print("Most Downloaded Category of 2018=",cat18)
```

52. Least Downloaded Category



Code for Information:

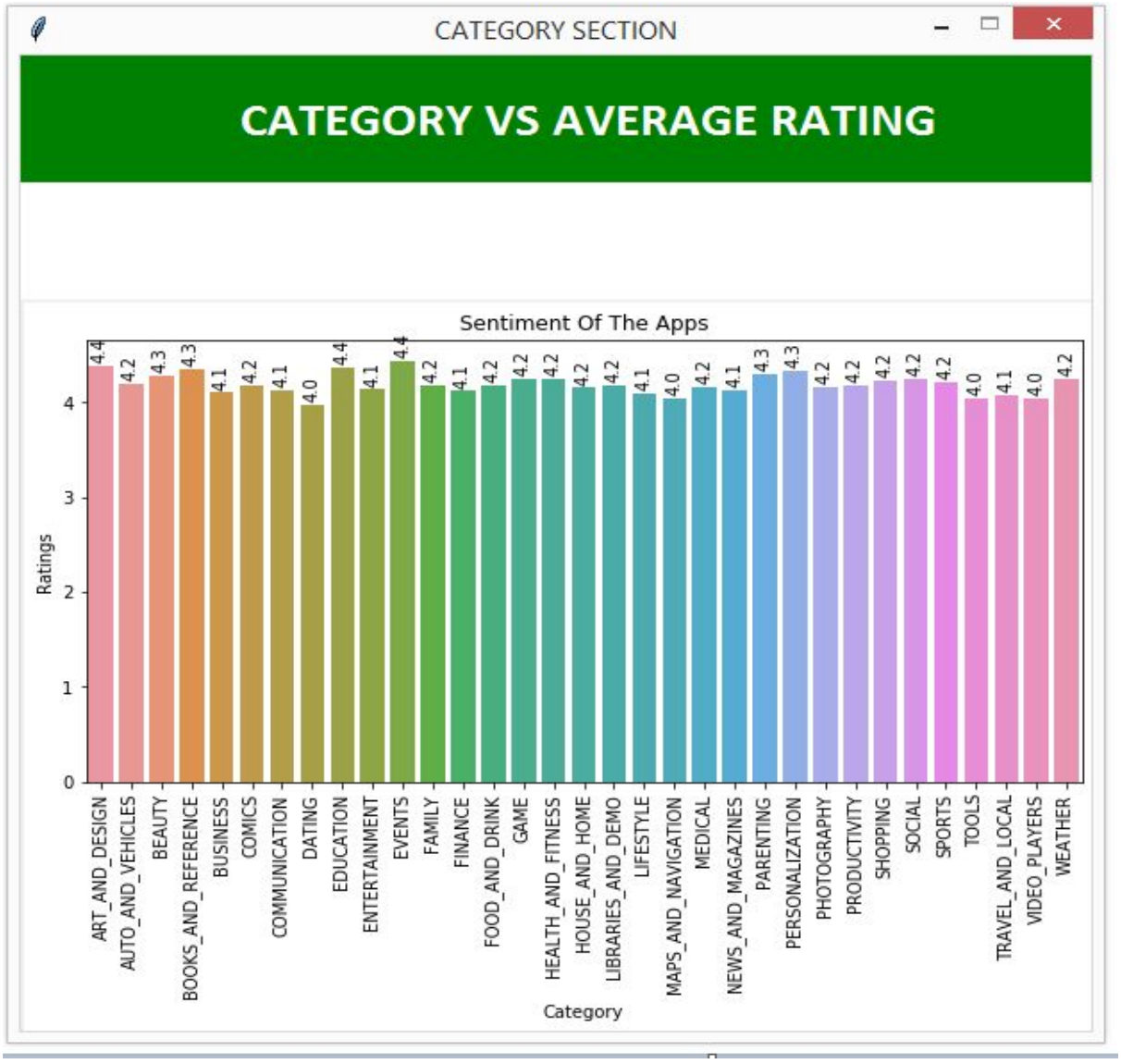
```
if play.count()>=d:
       d=play.count()
       cat16=i
    if play1.count()>=e:
        e=play1.count()
        cat17=i
    if play2.count()>=f:
        f=play2.count()
        cat18=i
    if play.count()<p:
       p=play.count()
       ca16=i
    if play1.count()<s:
        s=play1.count()
        ca17=i
    if play2.count()<t:
        t=play2.count()
        ca18=i
print("Least Downloaded Category of 2016=",ca16)
print("Least Downloaded Category of 2017=",ca17)
print("Least Downloaded Category of 2018=",ca18)
```

Code for GUI:

```
def downloads_amtl():
    global screen5
```

```
    screen5=Toplevel(screen)
    screen5.title("CATEGORY SECTION")
    adjustWindow(screen5)
   # screen5.resizable(True,True)
    Label(screen5, text="LEAST DOWNLOADED CATEGORY", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='Least Downloaded Category of 2016= WEATHER\n\nLeast Downloaded
Category of 2017= ENTERTAINMENT\n\nLeast Downloaded Category of 2018=
LIBRARIES_AND_DEMO',width='80',height='30' ,font=("Helvetica",10, 'bold', 'italic'), fg='white',
bg='blue').place(x=0, y=100)
```

53. Category vs Average Rating



Code for Graph:

```
def avgRatings():
    ratings={}
    for j in df['Category']:
            k=j
```

```
        t2=(df[(df.Category==k)].Rating).tolist()
        ratings.update({j:float(sum(t2))/len(df[df.Category==k])})
    print(ratings)
    key = []
    for i in ratings.keys():
        key.append(i)
    value = []
    for i in ratings.values():
        value.append(i)
    plt.figure(figsize=(7,5))
    plt.ticklabel_format(style='plain', axis='x')
    dft = pd.DataFrame(dict(category=key, rating=value))
    plt.xticks(rotation=90)
    an=sns.barplot("category","rating",data=dft)
    for p in an.patches:
      an.annotate(format(p.get_height(), '.1f'), (p.get_x() + p.get_width() / 2., p.get_height()), ha =
'center', va = 'center', xytext = (0, 12),rotation=90,fontsize=10, textcoords = 'offset points')
    plt.title("Sentiment Of The Apps")
```

Code for GUI:

```
def cat3():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("CATEGORY SECTION")
    adjustWindow(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="CATEGORY VS AVERAGE RATING", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    photo = PhotoImage(file="avgratings (1).png") # opening left side image - Note: If image is in
same folderthen no need to mention the full path
    label = Label(screen5, image=photo ,text="") # attaching image to the label
    label.place(x=0, y=150)
    label.image = photo
```

54. Price

Code for Gui:

```
def analysis4():
    global screen7
```

```
    screen7=Toplevel(screen)
    screen7.title("PRICE")
    adjustWindow(screen7)
    Label(screen7, text="PRICE",width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Button(screen7, text='PRICE', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=price).place(x=50, y=200)
```

55. Price chart

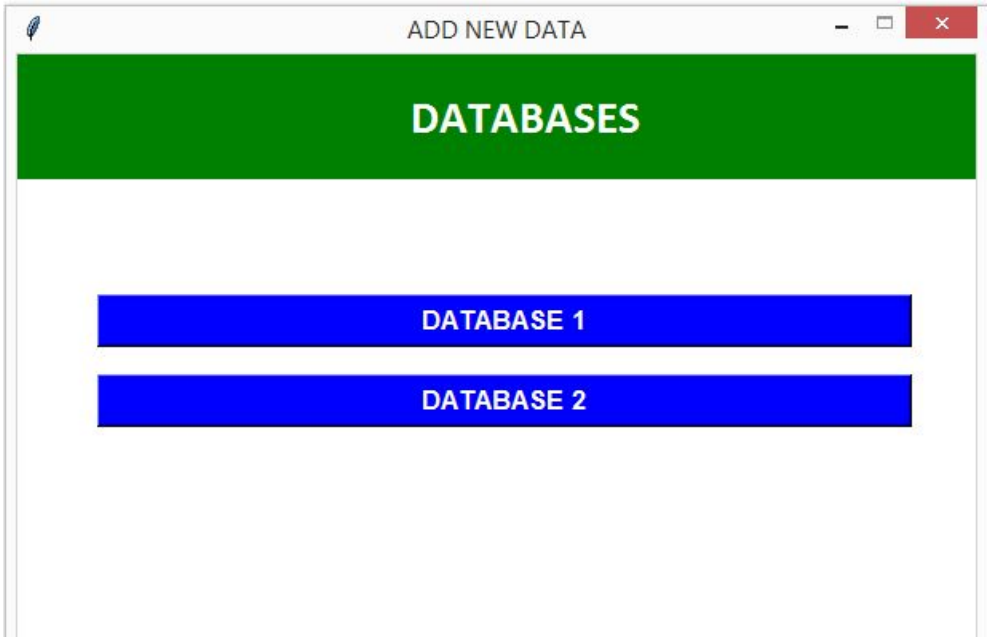
Code for Graph:

```
plt.figure(figsize=(10,10))
fig = sns.countplot(x=df['Type'])
#fig.set_xticklabels(fig.get_xticklabels(),rotation=90)
plt.ylabel("COUNT")
plt.xlabel("TYPE")
plt.show(fig)
```

Code for GUI:

```
def price():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("PRICE SECTION")
    adjustWindow1(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="PRICE", width = '130',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    photo = PhotoImage(file="price.png") # opening left side image - Note: If image is in same
folderthen no need to mention the full path
    label = Label(screen5, image=photo ,text="") # attaching image to the label
    label.place(x=600, y=150)
    label.image = photo
```

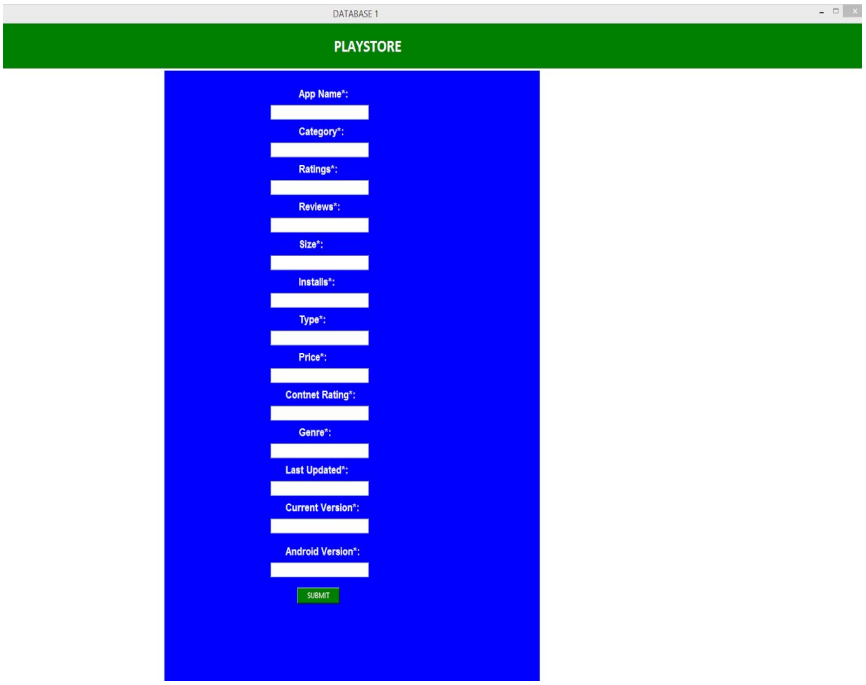
56. Add New Data



Code:

```
def analysis7():
    global screen7
    screen7=Toplevel(screen)
    screen7.title("ADD NEW DATA")
    adjustWindow(screen7)
    Label(screen7, text="DATABASES",width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Button(screen7, text='DATABASE 1', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=analysis8).place(x=50, y=150)
    Button(screen7, text='DATABASE 2', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=analysis9).place(x=50, y=200)
```

57. Entry into Database 1



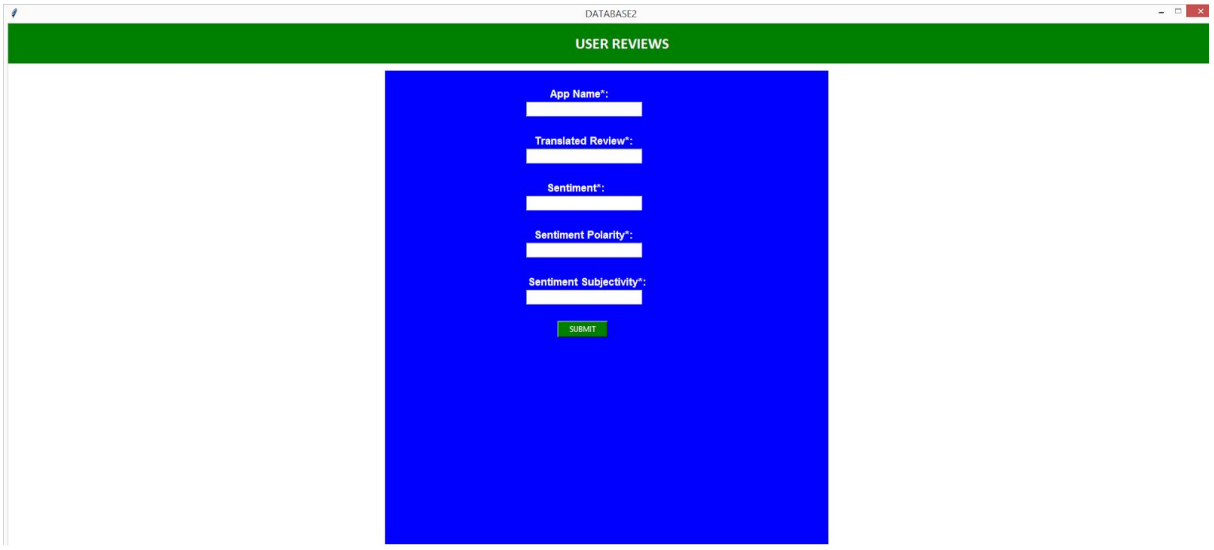
Code:

```
def analysis8():
    global screen7
    screen7=Toplevel(screen)
    screen7.title("DATABASE 1")
    adjustWindow1(screen7)
   # screen7.resizable(True,True)
    Label(screen7, text="PLAYSTORE",width = '140',height ='2',
font=('calibri',20,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen7, text="",bg="blue",width='100',height='700').place(x=600,y=75)
    #Button(screen7, text='CATEGORY', width=50, font=("Open Sans", 13, 'bold'),bg='brown',
fg='white',command=catl).place(x=50, y=100)
    #Button(screen7, text='MOST DOWNLOADED CATGORY', width=50, font=("Open Sans", 13,
'bold'),bg='brown', fg='white',command=downloads_amt).place(x=50, y=150)
    #Button(screen7, text='LEAST DOWNLOADED CATGORY', width=50, font=("Open Sans", 13,
'bold'),bg='brown', fg='white',command=downloads_amtl).place(x=50, y=200)

    L1=Label(screen7,text="App Name*:",font=("Open Sans", 13, 'bold'),bg='blue', fg='white')
    L1.place(x=850,y=100)
    txt1=Entry(screen7,font=("Open Sans", 13, 'bold'),bg='white', fg='black')
    txt1.place(x=800,y=130)
    L1=Label(screen7,text="Category*:",font=("Open Sans", 13, 'bold'),bg='blue', fg='white')
    L1.place(x=850,y=160)
    txt2=Entry(screen7,font=("Open Sans", 13, 'bold'),bg='white', fg='black')
    txt2.place(x=800,y=190)
    L1=Label(screen7,text="Ratings*:",font=("Open Sans", 13, 'bold'),bg='blue', fg='white')
    L1.place(x=850,y=220)
    txt3=Entry(screen7,font=("Open Sans", 13, 'bold'),bg='white', fg='black')
    txt3.place(x=800,y=250)
    L1=Label(screen7,text="Reviews*:",font=("Open Sans", 13, 'bold'),bg='blue', fg='white')
    L1.place(x=850,y=280)
```

```
txt4=Entry(screen7,font=("Open Sans", 13, 'bold'),bg='white', fg='black')
txt4.place(x=800,y=310)
L1=Label(screen7,text="Size*:",font=("Open Sans", 13, 'bold'),bg='blue', fg='white')
L1.place(x=850,y=340)
txt5=Entry(screen7,font=("Open Sans", 13, 'bold'),bg='white', fg='black')
txt5.place(x=800,y=370)
L1=Label(screen7,text="Installs*:",font=("Open Sans", 13, 'bold'),bg='blue', fg='white')
L1.place(x=850,y=400)
txt6=Entry(screen7,font=("Open Sans", 13, 'bold'),bg='white', fg='black')
txt6.place(x=800,y=430)
L1=Label(screen7,text="Type*:",font=("Open Sans", 13, 'bold'),bg='blue', fg='white')
L1.place(x=850,y=460)
txt7=Entry(screen7,font=("Open Sans", 13, 'bold'),bg='white', fg='black')
txt7.place(x=800,y=490)
L1=Label(screen7,text="Price*:",font=("Open Sans", 13, 'bold'),bg='blue', fg='white')
L1.place(x=850,y=520)
txt8=Entry(screen7,font=("Open Sans", 13, 'bold'),bg='white', fg='black')
txt8.place(x=800,y=550)
L1=Label(screen7,text="Contnet Rating*:",font=("Open Sans", 13, 'bold'),bg='blue', fg='white')
L1.place(x=825,y=580)
txt9=Entry(screen7,font=("Open Sans", 13, 'bold'),bg='white', fg='black')
txt9.place(x=800,y=610)
L1=Label(screen7,text="Genre*:",font=("Open Sans", 13, 'bold'),bg='blue', fg='white')
L1.place(x=850,y=640)
txt10=Entry(screen7,font=("Open Sans", 13, 'bold'),bg='white', fg='black')
txt10.place(x=800,y=670)
L1=Label(screen7,text="Last Updated*:",font=("Open Sans", 13, 'bold'),bg='blue', fg='white')
L1.place(x=825,y=700)
txt11=Entry(screen7,font=("Open Sans", 13, 'bold'),bg='white', fg='black')
txt11.place(x=800,y=730)
L1=Label(screen7,text="Current Version*:",font=("Open Sans", 13, 'bold'),bg='blue', fg='white')
L1.place(x=825,y=760)
txt12=Entry(screen7,font=("Open Sans", 13, 'bold'),bg='white', fg='black')
txt12.place(x=800,y=790)
L1=Label(screen7,text="Android Version*:",font=("Open Sans", 13, 'bold'),bg='blue', fg='white')
L1.place(x=825,y=830)
txt13=Entry(screen7,font=("Open Sans", 13, 'bold'),bg='white', fg='black')
txt13.place(x=800,y=860)
def retrieve_input():
    inputValue=txt1.get()
    app=inputValue
    inputValue=txt2.get()
    category=inputValue
    inputValue=txt3.get()
    rating=inputValue
    inputValue=txt4.get()
    reviews=inputValue
    inputValue=txt5.get()
    size=inputValue
    inputValue=txt6.get()
    installs=inputValue
```

```
        inputValue=txt7.get()
        type1=inputValue
        inputValue=txt8.get()
        price=inputValue
        inputValue=txt9.get()
        content_rating=inputValue
        inputValue=txt10.get()
        geners=inputValue
        inputValue=txt11.get()
        last_updated=inputValue
        inputValue=txt12.get()
        current_ver=inputValue
        inputValue=txt13.get()
        and_ver=inputValue
        def get_length(file_path):
            with open("file_path") as csvfile:
                reader=csv.reader(csvfile)
                reader_list=list(reader)
                return len(reader_list)
            return 1
        def
append_data(file_path,app,category,rating,reviews,size,installs,type1,price,content_rating,geners,last_
updated,current_ver,and_ver):

fieldnames=['app','category','rating','reviews','size','installs','type1','price','content_rating','geners','last_
updated','current_ver','and_ver']
            next_id=get_length(file_path)
            '''
            with open(file_path,"a",newline='') as csvfile:
                writer=csv.DictWriter(csvfile,fieldnames=fieldnames)
                writer.writerow({"app":" "})
            '''
            with open(file_path,"a",newline='') as csvfile:
                writer=csv.DictWriter(csvfile,fieldnames=fieldnames)

writer.writerow({"app":app,"category":category,"rating":rating,"reviews":reviews,"size":size,"installs
":installs,"type1":type1,"price":price,"content_rating":content_rating,"geners":geners,"last_updated":l
ast_updated,"current_ver":current_ver,"and_ver":and_ver})

        append_data("C:\\Users\\Siddhesh\\Desktop\\Database
1",app,category,rating,reviews,size,installs,type1,price,content_rating,geners,last_updated,current_ver
,and_ver)
    buttonCommit=Button(screen7, height=1, width=10, text="SUBMIT",bg='green',fg='white'
,command=lambda: retrieve_input())#command=lambda: retrieve_input() >>> just means do this
when i press the button
    buttonCommit.place(x=850,y=900)
```

58. Entry into Database 2



Code:

```
def analysis9():
    global screen7
    screen7=Toplevel(screen)
    screen7.title("DATABASE2")
    adjustWindow1(screen7)
   # screen7.resizable(True,True)
    Label(screen7, text="USER REVIEWS",width = '150',height ='2',
font=('calibri',18,'bold'),fg='white',bg='green').place(x=0,y=0)
    #Button(screen7, text='CATEGORY', width=50, font=("Open Sans", 13, 'bold'),bg='brown',
fg='white',command=catl).place(x=50, y=100)
    #Button(screen7, text='MOST DOWNLOADED CATGORY', width=50, font=("Open Sans", 13,
'bold'),bg='brown', fg='white',command=downloads_amt).place(x=50, y=150)
    #Button(screen7, text='LEAST DOWNLOADED CATGORY', width=50, font=("Open Sans", 13,
'bold'),bg='brown', fg='white',command=downloads_amtl).place(x=50, y=200)
    Label(screen7,text ="",bg="blue", width='100',height='50').place(x=600, y=75)
    L1=Label(screen7,text="App Name*:",font=("Open Sans", 13, 'bold'),bg='blue', fg='white')
    L1.place(x=860,y=100)
    txt1=Entry(screen7,font=("Open Sans", 13, 'bold'),bg='white', fg='black')
    txt1.place(x=825,y=125)
    L1=Label(screen7,text="Translated Review*:",font=("Open Sans", 13, 'bold'),bg='blue', fg='white')
    L1.place(x=835,y=175)
    txt2=Entry(screen7,font=("Open Sans", 13, 'bold'),bg='white', fg='black')
    txt2.place(x=825,y=200)
    L1=Label(screen7,text="Sentiment*:",font=("Open Sans", 13, 'bold'),bg='blue', fg='white')
    L1.place(x=855,y=250)
    txt3=Entry(screen7,font=("Open Sans", 13, 'bold'),bg='white', fg='black')
    txt3.place(x=825,y=275)
    L1=Label(screen7,text="Sentiment Polarity*:",font=("Open Sans", 13, 'bold'),bg='blue', fg='white')
    L1.place(x=835,y=325)
    txt4=Entry(screen7,font=("Open Sans", 13, 'bold'),bg='white', fg='black')
    txt4.place(x=825,y=350)
```

```
   L1=Label(screen7,text="Sentiment Subjectivity*:",font=("Open Sans", 13, 'bold'),bg='blue',
fg='white')
   L1.place(x=825,y=400)
   txt5=Entry(screen7,font=("Open Sans", 13, 'bold'),bg='white', fg='black')
   txt5.place(x=825,y=425)
   def retrieve_input():
      inputValue=txt1.get()
      app=inputValue
      inputValue=txt2.get()
      translated_review=inputValue
      inputValue=txt3.get()
      sentiment=inputValue
      inputValue=txt4.get()
      sentiment_polarity=inputValue
      inputValue=txt5.get()
      sentiment_subjectivity=inputValue

      def get_length1(file_path):
         with open("file_path") as csvfile:
            reader=csv.reader(csvfile)
            reader_list=list(reader)
            return len(reader_list)
         return 1

      def
append_data1(file_path,app,translated_review,sentiment,sentiment_polarity,sentiment_subjectivity):
         fieldnames=['app','translated_review','sentiment','sentiment_polarity','sentiment_subjectivity']
         next_id=get_length1(file_path)
         with open(file_path,"a",newline='') as csvfile:
            writer=csv.DictWriter(csvfile,fieldnames=fieldnames)

writer.writerow({"app":app,"translated_review":translated_review,"sentiment":sentiment,"sentiment_
polarity":sentiment_polarity,"sentiment_subjectivity":sentiment_subjectivity})
      append_data1("C:\\Users\\Siddhesh\\Desktop\\Database
2",app,translated_review,sentiment,sentiment_polarity,sentiment_subjectivity)
   buttonCommit=Button(screen7, height=1, width=10, text="SUBMIT",bg='green',fg='white'
,command=lambda: retrieve_input())#command=lambda: retrieve_input() >>> just means do this
when i press the button
   buttonCommit.place(x=875,y=475)
```

Section 5: Testing

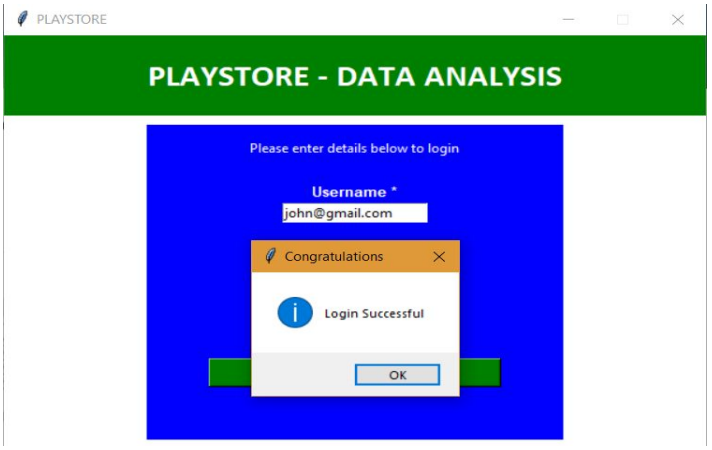
5.1 Graphs

Data is plotted in the form of graphs as it is easy to visualize and understand the characteristics of the data. For this we have used seaborn library which helps in plotting the data in various types of graphs. In order to fit these graphs perfectly in the given screen, we tested various dimensions in the spyder IDE.After some testing we got the ideal dimensions for our plots.

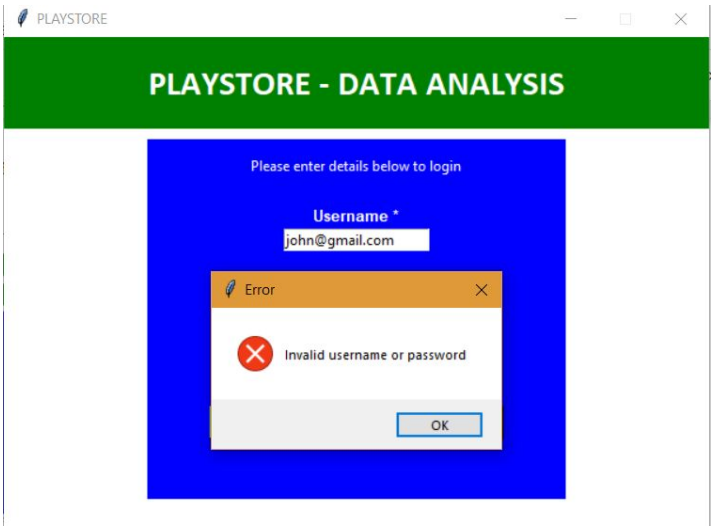
5.2 GUI Framework dimensions

In this project GUI plays a very important part as it acts as a way to interact with the application. GUI is made as simple as possible so that the user can easily understand its functioning and get the required data effectively without wasting a lot of time. The output of the functions are plotted as graphs and those graphs are loaded into the buttons provided in the GUI. All the dimensions of the screens are kept same to ensure uniformity in the application.(600x600)

5.3 Validation of login page



The login page consists of 2 entry fields namely, username and password. When an already existing user enters his username or password, .get() function compares the input to the data in the database. If both the entries are correct, there appears a messagebox stating login successful.



If any one of the entries doesn't match the data in the database then a messagebox stating invalid username or password pops up.

5.4 Validation of registration form

+ Options

	id	fullname	email	password	gender	university
Edit Copy Delete	1	John Doe	john@gmail.com	1234	male	Mumbai University
Edit Copy Delete	2	j	j@g.com	jay	male	Mumbai University
Edit Copy Delete	3	jayesh	j@g.com	jay	male	Mumbai University
Edit Copy Delete	4	jayesh1	ja@g.com	jay	male	Mumbai University
Edit Copy Delete	5	JAYESH	jay@gmail.com	jay	male	Mumbai University
Edit Copy Delete	6	sid	j@g.com	jay	male	Mumbai University
Edit Copy Delete	7	jayesh	jay@gmail.com	jay	male	Mumbai University

Check all *With selected:* Edit Copy Delete Export

When a new user wants to use the application, he/she must fill the registration form before loging in.The form contains various entry fields such as name ,email , password , etc. When the user fills all the required data successfully then a message stating registration successful is displayed.All the entered data is entered in the database and stored for further use. If any field is not filled then an error message pops up.

Section 6: Source Code.

```
from tkinter import *
from tkinter import messagebox
import re
import pymysql
import csv
import pandas as pd

def fam():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("CATEGORY")
    adjustWindow(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="FAMILY", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='Count in 2016= 134\n\nCount in 2017= 285\n\nCount in 2018=
767\n\nPercentage increase in downloads= 472.3880597014926\n\n',width='80',height='30'
,font=("Helvetica",10, 'bold', 'italic'), fg='white', bg='blue').place(x=0, y=100)

def game():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("CATEGORY")
    adjustWindow(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="GAME", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='Count in 2016= 51\n\nCount in 2017= 130\n\nCount in 2018=
631\n\nPercentage increase in downloads= 1137.2549019607843\n\n',width='80',height='30'
,font=("Helvetica",10, 'bold', 'italic'), fg='white', bg='blue').place(x=0, y=100)

def tools1():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("CATEGORY")
    adjustWindow(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="TOOLS", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='Count in 2016= 54\n\nCount in 2017= 105\n\nCount in 2018=
338\n\nPercentage increase in downloads= 525.925925925926\n\n',width='80',height='30'
,font=("Helvetica",10, 'bold', 'italic'), fg='white', bg='blue').place(x=0, y=100)

def pro1():
    global screen5
```

```
    screen5=Toplevel(screen)
    screen5.title("CATEGORY")
    adjustWindow(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="PRODUCTIVITY", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='Count in 2016= 24\n\nCount in 2017= 25\n\nCount in 2018=
191\n\nPercentage increase in downloads= 695.8333333333333\n\n'
,width='80',height='30',font=("Helvetica",10, 'bold', 'italic'), fg='white', bg='blue').place(x=0,
y=100)

def comm():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("CATEGORY")
    adjustWindow(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="COMMUNICATION", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='Count in 2016= 15\n\nCount in 2017= 31\n\nCount in 2018=
202\n\nPercentage increase in downloads= 1246.6666666666667\n\n',width='80',height='30'
,font=("Helvetica",10, 'bold', 'italic'), fg='white', bg='blue').place(x=0, y=100)

def fin1():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("CATEGORY")
    adjustWindow(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="FINANCE", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='Count in 2016= 6\n\nCount in 2017= 35\n\nCount in 2018=
179\n\nPercentage increase in downloads= 2883.333333333333\n\n',width='80',height='30'
,font=("Helvetica",10, 'bold', 'italic'), fg='white', bg='blue').place(x=0, y=100)

def sports1():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("CATEGORY")
    adjustWindow(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="SPORTS", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='Count in 2016= 10\n\nCount in 2017= 34\n\nCount in 2018=
193\n\nPercentage increase in downloads= 1830.0\n\n'
,width='80',height='30',font=("Helvetica",10, 'bold', 'italic'), fg='white', bg='blue').place(x=0,
y=100)
```

```
def enter1():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("CATEGORY")
    adjustWindow(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="ENTERTAINMENT", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='#d9660a').place(x=0,y=0)
    Label(screen5, text='Count in 2016= 2\n\nCount in 2017= 1\n\nCount in 2018=
94\n\nPercentage increase in downloads= 4600.0\n\n',width='80',height='30'
,font=("Helvetica",10, 'bold', 'italic'), fg='white', bg='#174873').place(x=0, y=100)


def wet1():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("CATEGORY")
    adjustWindow(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="WEATHER", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='Count in 2016= 0\n\nCount in 2017= 5\n\nCount in 2018=
43\n\nPercentage increase in downloads= 4300.0\n\n',width='80',height='30'
,font=("Helvetica",10, 'bold', 'italic'), fg='white', bg='blue').place(x=0, y=100)

def lib1():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("CATEGORY")
    adjustWindow(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="LIBRARIES AND DEMO", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='Count in 2016= 4\n\nCount in 2017= 19\n\nCount in 2018=
23\n\nPercentage increase in downloads= 475.0\n\n',width='80',height='30'
,font=("Helvetica",10, 'bold', 'italic'), fg='white', bg='blue').place(x=0, y=100)



def enter():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("ENTERTAINMENT")
    adjustWindow(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="Apps with 1 lakh+ installs and Rating 4.1+", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
```

```
    Label(screen5, text='Netflix\n\nTV+\n\nVigo Lite\n\nHotstar\n\nPeers.TV: broadcast TV channels First, Match T...\n\nH TV\n\nTalking Ginger 2\n\nGirly Lock Screen Wallpaper with Quotes\n\nAmazon Prime Video\n\nIMDb Movies & TV\n\n',width='80',height='30' ,font=("Helvetica",10, 'bold', 'italic'), fg='white', bg='blue').place(x=0, y=100)



def edu():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("EDUCATION")
    adjustWindow(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="Apps with 1 lakh+ installs and Rating 4.1+", width = '42',height ='2', font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='English Communication - Learn English for Chin...\n\nKhan Academy\n\nEnglish Grammar Test\n\nSpeed Reading\n\nLearn English Words Free\n\nEnglish words application mikan\n\nLearn English for beginners\n\nListen and learn English in seven days\n\nLearn English from Persian: Persian to English\n\n English with Lingualeo\n\n',width='80',height='30' ,font=("Helvetica",10, 'bold', 'italic'), fg='white', bg='blue').place(x=0, y=100)

def bus():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("ENTERTAINMENT")
    adjustWindow(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="Apps with 1 lakh+ installs and Rating 4.1+", width = '42',height ='2', font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='Indeed Job Search\n\nADP Mobile Solutions\n\nDocs To Go™ Free Office Suite\n\nGoogle My Business\n\nOfficeSuite : Free Office + PDF Editor\n\nCurriculum vitae App CV Builder Free Resume Maker\n\nPolaris Office for LG\n\nCall Blocker\n\nJobs in Alabama - Jobs in Alba\n\nSquare Point of Sale - POS\n\n',width='80',height='30' ,font=("Helvetica",10, 'bold', 'italic'), fg='white', bg='blue').place(x=0, y=100)



def med():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("MEDICAL")
    adjustWindow(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="Apps with 1 lakh+ installs and Rating 4.1+", width = '42',height ='2', font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
```

```
    Label(screen5, text='Monash Uni Low FODMAP Diet\n\nMedical ID - In Case of
Emergency (ICE)\n\nHuman Anatomy Atlas 2018: Complete 3D Human Body\n\nEssential
Anatomy 3\n\nMuscle Trigger Point Anatomy\n\nEMT Study - NREMT Test Prep\n\nFHR
5-Tier 2.0\n\nAnatomyMapp\n\nMigraine, Headache Diary HeadApp Pro\n\nVisual
Anatomy 2\n\n',width='80',height='30' ,font=("Helvetica",10, 'bold', 'italic'), fg='white',
bg='blue').place(x=0, y=100)


def pro():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("PRODUCTIVITY")
    adjustWindow(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="Apps with 1 lakh+ installs and Rating 4.1+", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='Microsoft Word\n\nAdobe Acrobat Reader\n\nMicrosoft
Outlook\n\nMicrosoft Excel\n\nMicrosoft OneDrive\n\nCalculator - unit
converter\n\nMicrosoft OneNote\n\nGoogle Keep\n\nES File Explorer File
Manager\n\nDropbox\n\n',width='80',height='30' ,font=("Helvetica",10, 'bold', 'italic'),
fg='white', bg='blue').place(x=0, y=100)

def per():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("PERSONALITY")
    adjustWindow(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="Apps with 1 lakh+ installs and Rating 4.1+", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='Nova Launcher\n\nZEDGE™ Ringtones & Wallpapers\n\nXOS -
Launcher,Theme,Wallpaper\n\n3D Live Neon Weed Launcher\n\nEvie Launcher\n\nGolden
Launcher\n\nCM Launcher 3D - Theme, Wallpapers, Efficient\n\n4K Wallpapers and Ultra
HD Backgrounds\n\nZenUI LauncherAPUS Launcher - Theme, Wallpaper, Hide
Apps\n\n',width='80',height='30' ,font=("Helvetica",10, 'bold', 'italic'), fg='white',
bg='blue').place(x=0, y=100)

def life():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("LIFESTYLE")
    adjustWindow(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="Apps with 1 lakh+ installs and Rating 4.1+", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='Dollhouse Decorating Games\n\nEasy Hair Style Design\n\nBlack
Wallpaper, AMOLED, Dark Background: Darkify\n\nChart - Myanmar Keyboard\n\nLive
4D Results ! (MY & SG)\n\nFOSSIL Q: DESIGN YOUR DIAL\n\nKawaii Easy Drawing :
```

```
How to draw Step by Step\n\nSamsung+\n\nBeautiful Design Birthday
Cake\n\nPronunciation and know the name of the caller ...\n\n',width='80',height='30'
,font=("Helvetica",10, 'bold', 'italic'), fg='white', bg='blue').place(x=0, y=100)


def fin():
   global screen5
   screen5=Toplevel(screen)
   screen5.title("FINANCE")
   adjustWindow(screen5)
   #screen5.resizable(True,True)
   Label(screen5, text="Apps with 1 lakh+ installs and Rating 4.1+", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
   Label(screen5, text='Nedbank Money\n\nSCB EASY\n\nNubank\n\nBBVA
Spain\n\nVTB-Online\n\n PayPal\n\nGoogle Pay\n\nTransfer\n\nTrueMoney
Wallet\n\nWells Fargo Daily Change\n\n',width='80',height='30' ,font=("Helvetica",10, 'bold',
'italic'), fg='white', bg='blue').place(x=0, y=100)

def sport():
   global screen5
   screen5=Toplevel(screen)
   screen5.title("SPORTS")
   adjustWindow(screen5)
   #screen5.resizable(True,True)
   Label(screen5, text="Apps with 1 lakh+ installs and Rating 4.1+", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
   Label(screen5, text='8 Ball Pool\n\nScore! Hero\n\nDream League Soccer 2018\n\nMini
Golf King - Multiplayer Game\n\nFree Sports TV\n\nMLB At Bat\n\nNFL\n\nCristiano
Ronaldo Wallpaper\n\nkicker football news\n\nFootball Live
Scores\n\n',width='80',height='30' ,font=("Helvetica",10, 'bold', 'italic'), fg='white',
bg='blue').place(x=0, y=100)

def tool():
   global screen5
   screen5=Toplevel(screen)
   screen5.title("SPORTS")
   adjustWindow(screen5)
   #screen5.resizable(True,True)
   Label(screen5, text="Apps with 1 lakh+ installs and Rating 4.1+", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
   Label(screen5, text='Google Translate\n\nMotorola Alert\n\nMotorola Assist\n\nCache
Cleaner-DU Speed Booster\n\nDevice Help\n\nAccount Manager\n\nFile
Manager\n\nCalculator - free calculator ,multi calculator...\n\nSHAREit - Transfer &
Share\n\nNokia mobile support\n\n',width='80',height='30' ,font=("Helvetica",10, 'bold',
'italic'), fg='white', bg='blue').place(x=0, y=100)
```

```
def downloads_16():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("CATEGORY SECTION")
    adjustWindow1(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="CATEGORY DOWNLOADS OF 2016", width = '130',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    photo = PhotoImage(file="2016.png") # opening left side image - Note: If image is in same
folderthen no need to mention the full path
    label = Label(screen5, image=photo ,text="") # attaching image to the label
    label.place(x=400, y=150)
    label.image = photo


def catl():
    global screen7
    screen7=Toplevel(screen)
    screen7.title("RATINGS")
    adjustWindow(screen7)
    Label(screen7, text="CATEGORY",width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Button(screen7, text='FAMILY', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=fam).place(x=50, y=100)
    Button(screen7, text='GAME', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=game).place(x=50, y=150)
    Button(screen7, text='TOOLS', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=tools1).place(x=50, y=200)
    Button(screen7, text='PRODUCTIVITY', width=50, font=("Open Sans", 13,
'bold'),bg='blue', fg='white',command=pro1).place(x=50, y=250)
    #Button(screen7, text='MEDICAL', width=50, font=("Open Sans", 13, 'bold'),bg='brown',
fg='white',command=pro).place(x=50, y=300)
    Button(screen7, text='COMMUNICATION', width=50, font=("Open Sans", 13,
'bold'),bg='blue', fg='white',command=comm).place(x=50, y=300)
    Button(screen7, text='FINANCE', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=fin1).place(x=50, y=350)
    Button(screen7, text='SPORTS', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=sports1).place(x=50, y=400)
    Button(screen7, text='ENTERTAINMENT', width=50, font=("Open Sans", 13,
'bold'),bg='blue', fg='white',command=enter1).place(x=50, y=450)
    Button(screen7, text='WEATHER', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=wet1).place(x=50, y=500)
    Button(screen7, text='LIBRERIES AND DEMO', width=50, font=("Open Sans", 13,
'bold'),bg='blue', fg='white',command=lib1).place(x=50, y=550)


def listl():
    global screen7
```

```
    screen7=Toplevel(screen)
    screen7.title("RATINGS")
    adjustWindow(screen7)
    Label(screen7, text="GENRE",width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Button(screen7, text='Entertainment', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=enter).place(x=50, y=100)
    Button(screen7, text='Education', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=edu).place(x=50, y=150)
    Button(screen7, text='Business', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=bus).place(x=50, y=200)
    Button(screen7, text='Medical', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=med).place(x=50, y=250)
    Button(screen7, text='Productivity', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=pro).place(x=50, y=300)
    Button(screen7, text='Personalization', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=per).place(x=50, y=350)
    Button(screen7, text='Lifestyle', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=life).place(x=50, y=400)
    Button(screen7, text='Finance', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=fin).place(x=50, y=450)
    Button(screen7, text='Sports', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=sport).place(x=50, y=500)
    Button(screen7, text='Tools', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=tool).place(x=50, y=550)




def downloads_17():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("CATEGORY SECTION")
    adjustWindow1(screen5)
  #  screen5.resizable(True,True)
    Label(screen5, text="CATEGORY DOWNLOADS OF 2017", width = '130',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    photo = PhotoImage(file="2017.png") # opening left side image - Note: If image is in same
folderthen no need to mention the full path
    label = Label(screen5, image=photo ,text="") # attaching image to the label
    label.place(x=400, y=150)
    label.image = photo


def downloads_18():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("CATEGORY SECTION")
```

```
    adjustWindow1(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="CATEGORY DOWNLOADS OF 2018", width = '130',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    photo = PhotoImage(file="2018.png") # opening left side image - Note: If image is in same
folderthen no need to mention the full path
    label = Label(screen5, image=photo ,text="") # attaching image to the label
    label.place(x=400, y=150)
    label.image = photo


def downloads_amt():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("CATEGORY SECTION")
    adjustWindow(screen5)
   # screen5.resizable(True,True)
    Label(screen5, text="MOST DOWNLOADED CATEGORY", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='Most Downloaded Category of 2016= FAMILY\n\nMost
Downloaded Category of 2017= FAMILY\n\nMost Downloaded Category of 2018=
FAMILY\n\n',width='80',height='30' ,font=("Helvetica",10, 'bold', 'italic'), fg='white',
bg='blue').place(x=0, y=100)
    #Button(screen5,text="Exit", command=do_exit,bg="yellow",fg="black",font=("Open
Sans", 16,"bold")).place(x=0,y=100)

def downloads_amtl():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("CATEGORY SECTION")
    adjustWindow(screen5)
   # screen5.resizable(True,True)
    Label(screen5, text="LEAST DOWNLOADESD CATEGORY", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='Least Downloaded Category of 2016= WEATHER\n\nLeast
Downloaded Category of 2017= ENTERTAINMENT\n\nLeast Downloaded Category of
2018= LIBRARIES_AND_DEMO',width='80',height='30' ,font=("Helvetica",10, 'bold',
'italic'), fg='white', bg='blue').place(x=0, y=100)
    #Button(screen5,text="Exit", command=do_exit,bg="yellow",fg="black",font=("Open
Sans", 16,"bold")).place(x=0,y=100)


def download_ratio():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("DOWNLOAD SECTION")
    adjustWindow(screen5)
   # screen5.resizable(True,True)
```

```
    Label(screen5, text="DOWNLOAD RATIO", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='No of Teen Downloads= 912\n\nNo of Mature 17+ Downloads
357\n\nThe ratio of Downloads for Teen vs Mature 17+ =
2.5546218487394956',width='80',height='30' ,font=("Helvetica",10, 'bold', 'italic'), fg='white',
bg='blue').place(x=0, y=100)

def do_exit():

    #Tk.destroy(screen4)
    Tk.destroy(screen)
   # Tk.destroy(screen3)

    #Tk.destroy(screen6)

def predict():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("CATEGORY SECTION")
    adjustWindow(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="CATEGORY", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='The Download Trend is Classified on the basis of the avg downloads
of categories of the last five years \n\n The Download Trend in the Coming Years will be like
this:\n\n 1.GAME\n\nAverage= 176.0\n\n2.SPORTS\n\nAverage=
49.0\n\n3.SOCIAL\n\nAverage= 42.0\n\n4.TRAVEL\n\nAverage=
33.4\n\n5.NEWS\n\nAverage= 32.2\n\n6.ENTERTAINMENT\n\nAverage=
19.6',width='85',height='30' ,font=("Helvetica",9, 'bold', 'italic'), fg='white',
bg='blue').place(x=0, y=100)
    #Button(screen5,text="Exit", command=do_exit,bg="yellow",fg="black",font=("Open
Sans", 16,"bold")).place(x=0,y=100)

def price():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("PRICE SECTION")
    adjustWindow1(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="PRICE", width = '130',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    photo = PhotoImage(file="price.png") # opening left side image - Note: If image is in same
folderthen no need to mention the full path
    label = Label(screen5, image=photo ,text="") # attaching image to the label
    label.place(x=600, y=150)
    label.image = photo
```

```
def review():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("REVIEW SECTION")
    adjustWindow(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="REVIEW", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='No. of Positive Reviews of 10 Best Foods for You = 162\n\nNo. of
Negative Reviews of 10 Best Foods for You = 10 \n\nNo. of Neutral Reviews of 10 Best
Foods for You = 22 \n\nThus by seeing the difference between the no of postitive and
negative reviews \n\nWe can Say that the Users like such App\n\nThus it is advisable to
launch an app like 10 Best Foods for You',width='80',height='30' ,font=("Helvetica",10,
'bold', 'italic'), fg='white', bg='blue').place(x=0, y=100)
    #Button(screen5,text="Exit", command=do_exit,bg="yellow",fg="black",font=("Open
Sans", 16,"bold")).place(x=0,y=100)


def percentage_download():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("DOWNLOAD SECTION")
    adjustWindow1(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="PERCENTAGE DOWNLOAD IN EACH CATEGORY", width =
'130',height ='2', font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    photo = PhotoImage(file="pd.png") # opening left side image - Note: If image is in same
folderthen no need to mention the full path
    label = Label(screen5, image=photo ,text="") # attaching image to the label
    label.place(x=400, y=150)
    label.image = photo
    #Button(screen5,text="Exit", command=do_exit,bg="yellow",fg="black",font=("Open
Sans", 16,"bold")).place(x=0,y=100)

def installs_month():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("DOWNLOAD SECTION")
    adjustWindow1(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="INSTALLS VS MONTH", width = '130',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    photo = PhotoImage(file="installs vs month.png") # opening left side image - Note: If
image is in same folderthen no need to mention the full path
    label = Label(screen5, image=photo ,text="") # attaching image to the label
    label.place(x=100, y=90)
    label.image = photo
```

```
    #Button(screen5,text="Exit", command=do_exit,bg="yellow",fg="black",font=("Open
Sans", 16,"bold")).place(x=0,y=100)


def installs_category():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("DOWNLOAD SECTION")
    adjustWindow1(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="CATEGORY VS MONTH", width = '130',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    photo = PhotoImage(file="Category vs Month.png") # opening left side image - Note: If
image is in same folderthen no need to mention the full path
    label = Label(screen5, image=photo ,text="") # attaching image to the label
    label.place(x=100, y=90)
    label.image = photo
    #Button(screen5,text="Exit", command=do_exit,bg="yellow",fg="black",font=("Open
Sans", 16,"bold")).place(x=0,y=100)



def downloads():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("DOWNLOAD SECTION")
    adjustWindow(screen5)
   # screen5.resizable(True,True)
    Label(screen5, text="NO OF DOWNLOADS", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen5, text='No of apps having downloads between 10 thousand and 50 thousand
are  986 \n\n No of apps having downloads between 50k and 1.5 lakh are  1550 \n\n No of
apps having download between 1.5 lakh and 5 lalkh are  1094 \n\n No apps having
downloads between 5 lakh and 50 lakh are 1917 \n\n No of apps having downloads more than
50 lakhs are 1978',width='80',height='30' ,font=("Helvetica",10, 'bold', 'italic'), fg='white',
bg='blue').place(x=0, y=100)
    #Button(screen5,text="Exit", command=do_exit,bg="yellow",fg="black",font=("Open
Sans", 16,"bold")).place(x=0,y=100)

def download_content_rating():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("DOWNLOAD SECTION")
    adjustWindow1(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="DOWNLOAD VS CONTENT RATING", width = '130',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
```

```
    photo = PhotoImage(file="Installs vs Content Rating.png") # opening left side image -
Note: If image is in same folderthen no need to mention the full path
    label = Label(screen5, image=photo ,text="") # attaching image to the label
    label.place(x=350, y=80)
    label.image = photo
    #Button(screen5,text="Exit", command=do_exit,bg="yellow",fg="black",font=("Open
Sans", 16,"bold")).place(x=0,y=100)



def download_rating():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("DOWNLOAD SECTION")
    adjustWindow(screen5)
    Label(screen5, text="DOWNLOAD VS RATING", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    photo = PhotoImage(file="dvsr.png") # opening left side image - Note: If image is in same
folderthen no need to mention the full path
    label = Label(screen5, image=photo ,text="") # attaching image to the label
    label.place(x=50, y=200)
    label.image = photo
    #Button(screen5,text="Exit", command=do_exit,bg="yellow",fg="black",font=("Open
Sans", 16,"bold")).place(x=0,y=100)


def download_category():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("DOWNLOAD SECTION")
    adjustWindow1(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="DOWNLOAD TREND CATEGORY WISE", width = '130',height
='2', font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    photo = PhotoImage(file="dvsc.png") # opening left side image - Note: If image is in same
folderthen no need to mention the full path
    label = Label(screen5, image=photo ,text="") # attaching image to the label
    label.place(x=250, y=270)
    label.image = photo
   # Button(screen5,text="Exit", command=do_exit,bg="yellow",fg="black",font=("Open
Sans", 16,"bold")).place(x=0,y=100)
def download_category1():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("DOWNLOAD SECTION")
    adjustWindow1(screen5)
    #screen5.resizable(True,True)
```

```
    Label(screen5, text="DOWNLOAD VS CATEGORY", width = '130',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    photo = PhotoImage(file="appdownloads.png") # opening left side image - Note: If image
is in same folderthen no need to mention the full path
    label = Label(screen5, image=photo ,text="") # attaching image to the label
    label.place(x=500, y=270)
    label.image = photo


def download_size():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("DOWNLOAD SECTION")
    adjustWindow1(screen5)
  #  screen5.resizable(True,True)
    Label(screen5, text="DOWNLOAD VS APP SIZE", width = '130',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    photo = PhotoImage(file="dvsas.png") # opening left side image - Note: If image is in
same folderthen no need to mention the full path
    label = Label(screen5, image=photo ,text="") # attaching image to the label
    label.place(x=500, y=270)
    label.image = photo
    #Button(screen5,text="Exit", command=do_exit,bg="yellow",fg="black",font=("Open
Sans", 16,"bold")).place(x=0,y=100)

def download_and():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("DOWNLOAD SECTION")
    adjustWindow1(screen5)
    #screen5.resizable(True,True)
    Label(screen5, text="DOWNLOAD VS ANDROID VERSION", width = '130',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    photo = PhotoImage(file="versiondownloads.png") # opening left side image - Note: If
image is in same folderthen no need to mention the full path
    label = Label(screen5, image=photo ,text="") # attaching image to the label
    label.place(x=500, y=270)
    label.image = photo

def sentiment():
    global screen5
    screen5=Toplevel(screen)
    screen5.title("REVIEW SECTION")
    adjustWindow1(screen5)
   # screen5.resizable(True,True)
    Label(screen5, text="SENTIMENT SUBJECTIVITY VS SENTIMENT POLARITY",
width = '130',height ='2', font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
```

```
    photo = PhotoImage(file="sentimentpolandsub.png") # opening left side image - Note: If
image is in same folderthen no need to mention the full path
    label = Label(screen5, image=photo ,text="") # attaching image to the label
    label.place(x=500, y=270)
    label.image = photo

def analysis1():
    global screen3
    screen3=Toplevel(screen)
    screen3.title("DOWNLOAD SECTION")
    adjustWindow(screen3)
    Label(screen3, text="EVERYTHING ABOUT DOWNLOADS", width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Button(screen3, text='PERCENTAGE DOWNLOAD IN EACH CATEGORY', width=50,
font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=percentage_download).place(x=50, y=100)
    Button(screen3, text='NO OF DOWNLOADS', width=50, font=("Open Sans", 13,
'bold'),bg='blue', fg='white',command=downloads).place(x=50, y=150)
    Button(screen3, text='DOWNLOAD TREND CATEGORY WISE', width=50,
font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=download_category).place(x=50, y=200)
    Button(screen3, text='DOWNLOAD VS RATING', width=50, font=("Open Sans", 13,
'bold'),bg='blue', fg='white',command=analysis5).place(x=50, y=250)
    Button(screen3, text='DOWNLOAD VS CONTENT RATING', width=50, font=("Open
Sans", 13, 'bold'),bg='blue', fg='white',command=download_content_rating).place(x=50,
y=300)
    Button(screen3, text='DOWNLOAD VS APP SIZE', width=50, font=("Open Sans", 13,
'bold'),bg='blue', fg='white',command=download_size).place(x=50, y=350)
    Button(screen3, text='DOWNLOAD VS MONTH', width=50, font=("Open Sans", 13,
'bold'),bg='blue', fg='white',command=installs_month).place(x=50, y=400)
    Button(screen3, text='DOWNLOAD RATIO', width=50, font=("Open Sans", 13,
'bold'),bg='blue', fg='white',command=download_ratio).place(x=50, y=450)
    Button(screen3, text='DOWNLOAD VS CATEGORY', width=50, font=("Open Sans", 13,
'bold'),bg='blue', fg='white',command=download_category1).place(x=50, y=500)
    Button(screen3, text='DOWNLOAD VS ANDROID VERSION', width=50, font=("Open
Sans", 13, 'bold'),bg='blue', fg='white',command=download_and).place(x=50, y=550)
    #Button(screen3, text='PRICE', width=50, font=("Open Sans", 13, 'bold'),bg='brown',
fg='white',command=price).place(x=50, y=450)

      #Button(screen3,text="Exit", command=do_exit,bg="yellow",fg="black",font=("Open
Sans", 16,"bold")).place(x=0,y=100)


def analysis2():
    global screen4
    screen4=Toplevel(screen)
    screen4.title("REVIEW")
    adjustWindow(screen4)
```

```
    Label(screen4, text="REVIEW",width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Button(screen4, text='APP RECOMMENDATION', width=50, font=("Open Sans", 13,
'bold'),bg='blue', fg='white',command=review).place(x=50, y=150)
    Button(screen4, text='SENTIMENTS', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=sentiment).place(x=50, y=200)
    Button(screen4, text='REVIEWS OF APP', width=50, font=("Open Sans", 13,
'bold'),bg='blue', fg='white',command=reg).place(x=50, y=250)
        #Button(screen4, text='RATINGS', width=20, font=("Open Sans", 13,
'bold'),bg='brown', fg='white').place(x=50, y=100)
    #Button(screen4, text='SEMTIMENTS', width=20, font=("Open Sans", 13,
'bold'),bg='brown', fg='white').place(x=50, y=150)
   # Button(screen3,text="Exit", command=do_exit,bg="yellow",fg="black",font=("Open
Sans", 16,"bold")).place(x=0,y=100)
def analysis6():
    global screen7
    screen7=Toplevel(screen)
    screen7.title("CATEGORY")
    adjustWindow(screen7)
    Label(screen7, text="CATEGORY",width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Button(screen7, text='CATEGORY', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=catl).place(x=50, y=100)
    Button(screen7, text='MOST DOWNLOADED CATGORY', width=50, font=("Open
Sans", 13, 'bold'),bg='blue', fg='white',command=downloads_amt).place(x=50, y=150)
    Button(screen7, text='LEAST DOWNLOADED CATGORY', width=50, font=("Open
Sans", 13, 'bold'),bg='blue', fg='white',command=downloads_amtl).place(x=50, y=200)

def analysis3():
    global screen7
    screen7=Toplevel(screen)
    screen7.title("CATEGORY")
    adjustWindow(screen7)
    Label(screen7, text="CATEGORY",width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Button(screen7, text='CATEGORY PREDICTION', width=50, font=("Open Sans", 13,
'bold'),bg='blue', fg='white',command=predict).place(x=50, y=100)
    Button(screen7, text='CATEEGORY VS MONTH', width=50, font=("Open Sans", 13,
'bold'),bg='blue', fg='white',command=installs_category).place(x=50, y=150)
    Button(screen7, text='CATEEGORY DOWNLOADS OF 2016', width=50, font=("Open
Sans", 13, 'bold'),bg='blue', fg='white',command=downloads_16).place(x=50, y=200)
    Button(screen7, text='CATEEGORY DOWNLOADS OF 2017', width=50, font=("Open
Sans", 13, 'bold'),bg='blue', fg='white',command=downloads_17).place(x=50, y=250)
    Button(screen7, text='CATEEGORY DOWNLOADS OF 2018', width=50, font=("Open
Sans", 13, 'bold'),bg='blue', fg='white',command=downloads_18).place(x=50, y=300)

    Button(screen7, text='DOWNLOAD HISTORY', width=50, font=("Open Sans", 13,
'bold'),bg='blue', fg='white',command=analysis6).place(x=50, y=350)
```

```
   #Button(screen7, text='RATINGS LIST', width=50, font=("Open Sans", 13,
'bold'),bg='brown', fg='white',command=analysis5).place(x=50, y=450)



def analysis4():
   global screen7
   screen7=Toplevel(screen)
   screen7.title("PRICE")
   adjustWindow(screen7)
   Label(screen7, text="PRICE",width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
   Button(screen7, text='PRICE', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=price).place(x=50, y=200)


   #Button(screen7, text='CATEEGORY VS MONTH', width=50, font=("Open Sans", 13,
'bold'),bg='brown', fg='white',command=installs_category).place(x=50, y=250)

def analysis5():
   global screen7
   screen7=Toplevel(screen)
   screen7.title("RATINGS")
   adjustWindow(screen7)
   Label(screen7, text="RATINGS",width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
   Button(screen7, text='LIST', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=listl).place(x=50, y=150)
   Button(screen7, text='GRAPH', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=download_rating).place(x=50, y=200)



def analysis7():
   global screen7
   screen7=Toplevel(screen)
   screen7.title("ADD NEW DATA")
   adjustWindow(screen7)
   Label(screen7, text="DATABASES",width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
   Button(screen7, text='DATABASE 1', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=analysis8).place(x=50, y=150)
   Button(screen7, text='DATABASE 2', width=50, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=analysis9).place(x=50, y=200)


def analysis8():
   global screen7
```

```
    screen7=Toplevel(screen)
    screen7.title("DATABASE 1")
    adjustWindow1(screen7)
   # screen7.resizable(True,True)
    Label(screen7, text="PLAYSTORE",width = '140',height ='2',
font=('calibri',20,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen7, text="",bg="blue",width='100',height='700').place(x=600,y=75)
    #Button(screen7, text='CATEGORY', width=50, font=("Open Sans", 13,
'bold'),bg='brown', fg='white',command=catl).place(x=50, y=100)
    #Button(screen7, text='MOST DOWNLOADED CATGORY', width=50, font=("Open
Sans", 13, 'bold'),bg='brown', fg='white',command=downloads_amt).place(x=50, y=150)
    #Button(screen7, text='LEAST DOWNLOADED CATGORY', width=50, font=("Open
Sans", 13, 'bold'),bg='brown', fg='white',command=downloads_amtl).place(x=50, y=200)

    L1=Label(screen7,text="App Name*:",font=("Open Sans", 13, 'bold'),bg='blue',
fg='white')
    L1.place(x=850,y=100)
    txt1=Entry(screen7,font=("Open Sans", 13, 'bold'),bg='white', fg='black')
    txt1.place(x=800,y=130)
    L1=Label(screen7,text="Category*:",font=("Open Sans", 13, 'bold'),bg='blue', fg='white')
    L1.place(x=850,y=160)
    txt2=Entry(screen7,font=("Open Sans", 13, 'bold'),bg='white', fg='black')
    txt2.place(x=800,y=190)
    L1=Label(screen7,text="Ratings*:",font=("Open Sans", 13, 'bold'),bg='blue', fg='white')
    L1.place(x=850,y=220)
    txt3=Entry(screen7,font=("Open Sans", 13, 'bold'),bg='white', fg='black')
    txt3.place(x=800,y=250)
    L1=Label(screen7,text="Reviews*:",font=("Open Sans", 13, 'bold'),bg='blue', fg='white')
    L1.place(x=850,y=280)
    txt4=Entry(screen7,font=("Open Sans", 13, 'bold'),bg='white', fg='black')
    txt4.place(x=800,y=310)
    L1=Label(screen7,text="Size*:",font=("Open Sans", 13, 'bold'),bg='blue', fg='white')
    L1.place(x=850,y=340)
    txt5=Entry(screen7,font=("Open Sans", 13, 'bold'),bg='white', fg='black')
    txt5.place(x=800,y=370)
    L1=Label(screen7,text="Installs*:",font=("Open Sans", 13, 'bold'),bg='blue', fg='white')
    L1.place(x=850,y=400)
    txt6=Entry(screen7,font=("Open Sans", 13, 'bold'),bg='white', fg='black')
    txt6.place(x=800,y=430)
    L1=Label(screen7,text="Type*:",font=("Open Sans", 13, 'bold'),bg='blue', fg='white')
    L1.place(x=850,y=460)
    txt7=Entry(screen7,font=("Open Sans", 13, 'bold'),bg='white', fg='black')
    txt7.place(x=800,y=490)
    L1=Label(screen7,text="Price*:",font=("Open Sans", 13, 'bold'),bg='blue', fg='white')
    L1.place(x=850,y=520)
    txt8=Entry(screen7,font=("Open Sans", 13, 'bold'),bg='white', fg='black')
    txt8.place(x=800,y=550)
```

```
   L1=Label(screen7,text="Contnet Rating*:",font=("Open Sans", 13, 'bold'),bg='blue',
fg='white')
   L1.place(x=825,y=580)
   txt9=Entry(screen7,font=("Open Sans", 13, 'bold'),bg='white', fg='black')
   txt9.place(x=800,y=610)
   L1=Label(screen7,text="Genre*:",font=("Open Sans", 13, 'bold'),bg='blue', fg='white')
   L1.place(x=850,y=640)
   txt10=Entry(screen7,font=("Open Sans", 13, 'bold'),bg='white', fg='black')
   txt10.place(x=800,y=670)
   L1=Label(screen7,text="Last Updated*:",font=("Open Sans", 13, 'bold'),bg='blue',
fg='white')
   L1.place(x=825,y=700)
   txt11=Entry(screen7,font=("Open Sans", 13, 'bold'),bg='white', fg='black')
   txt11.place(x=800,y=730)
   L1=Label(screen7,text="Current Version*:",font=("Open Sans", 13, 'bold'),bg='blue',
fg='white')
   L1.place(x=825,y=760)
   txt12=Entry(screen7,font=("Open Sans", 13, 'bold'),bg='white', fg='black')
   txt12.place(x=800,y=790)
   L1=Label(screen7,text="Android Version*:",font=("Open Sans", 13, 'bold'),bg='blue',
fg='white')
   L1.place(x=825,y=830)
   txt13=Entry(screen7,font=("Open Sans", 13, 'bold'),bg='white', fg='black')
   txt13.place(x=800,y=860)
   def retrieve_input():
      inputValue=txt1.get()
      app=inputValue
      inputValue=txt2.get()
      category=inputValue
      inputValue=txt3.get()
      rating=inputValue
      inputValue=txt4.get()
      reviews=inputValue
      inputValue=txt5.get()
      size=inputValue
      inputValue=txt6.get()
      installs=inputValue
      inputValue=txt7.get()
      type1=inputValue
      inputValue=txt8.get()
      price=inputValue
      inputValue=txt9.get()
      content_rating=inputValue
      inputValue=txt10.get()
      geners=inputValue
      inputValue=txt11.get()
      last_updated=inputValue
      inputValue=txt12.get()
```

```
        current_ver=inputValue
        inputValue=txt13.get()
        and_ver=inputValue
        def get_length(file_path):
            with open("file_path") as csvfile:
                reader=csv.reader(csvfile)
                reader_list=list(reader)
                return len(reader_list)
            return 1
        def
append_data(file_path,app,category,rating,reviews,size,installs,type1,price,content_rating,gen
ers,last_updated,current_ver,and_ver):

fieldnames=['app','category','rating','reviews','size','installs','type1','price','content_rating','gene
rs','last_updated','current_ver','and_ver']
            next_id=get_length(file_path)
            '''
            with open(file_path,"a",newline='') as csvfile:
                writer=csv.DictWriter(csvfile,fieldnames=fieldnames)
                writer.writerow({"app":" "})
            '''
            with open(file_path,"a",newline='') as csvfile:
                writer=csv.DictWriter(csvfile,fieldnames=fieldnames)

writer.writerow({"app":app,"category":category,"rating":rating,"reviews":reviews,"size":size,
"installs":installs,"type1":type1,"price":price,"content_rating":content_rating,"geners":geners,
"last_updated":last_updated,"current_ver":current_ver,"and_ver":and_ver})

        append_data("C:\\Users\\Siddhesh\\Desktop\\Database
1",app,category,rating,reviews,size,installs,type1,price,content_rating,geners,last_updated,cur
rent_ver,and_ver)
    buttonCommit=Button(screen7, height=1, width=10,
text="SUBMIT",bg='green',fg='white' ,command=lambda:
retrieve_input())#command=lambda: retrieve_input() >>> just means do this when i press the
button
    buttonCommit.place(x=850,y=900)

def reg():
    global screen1,screen2,university,review
    screen1 = Toplevel(screen)
    screen1.title("Reviews")
    Label(screen1,text ="",bg="blue", width='80',height='30').place(x=20, y=85)

    Label(screen1, text="USER REVIEWS",width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    university = StringVar()
    review=StringVar()
    adjustWindow(screen1)
```

```
  # screen1.resizable(True,True)
   list2=['positive','negative','neutral']
   list1 = ['10 Best Foods for You', '104 找工作 - 找工作 找打工 找兼職 履歷健檢 履歷診療室',
      '11st', '1800 Contacts - Lens Store',
      '1LINE – One Line with One Touch',

      '21-Day Meditation Experience',
      '2Date Dating App, Love and matching',
      '2GIS: directory & navigator', '2RedBeans',
      '2ndLine - Second Phone Number',
      '30 Day Fitness Challenge - Workout at Home',
      '365Scores - Live Scores', '3D Live Neon Weed Launcher',
      '4 in a Row', '4K Wallpapers and Ultra HD Backgrounds',
      '591房屋交易-租屋、中古屋、新建案、實價登錄、別墅透天、公寓套房、捷運、買房賣房行情、房價房貸查詢', '591房屋交易-香港',
      '7 Cups: Anxiety & Stress Chat', '7 Day Food Journal Challenge',
      '7 Minute Workout', '7 Weeks - Habit & Goal Tracker',
      '8 Ball Pool', '850 Sports News Digest',
      '8fit Workouts & Meal Planner', '95Live -SG#1 Live Streaming App',
      'A Call From Santa Claus!', 'A Word A Day',
      'A&E - Watch Full Episodes of TV Shows',
      'A+ Gallery - Photos & Videos', 'A+ Mobile',
      'ABC Kids - Tracing & Phonics', 'ABC News - US & World News',
      'ABC Preschool Free', 'ABCmouse.com',
      'AC - Tips & News for Android™', 'ACE Elite',
      'AD - Nieuws, Sport, Regio & Entertainment', 'AMC Theatres', 'ANA',
      'AOL - News, Mail & Video', 'AP Mobile - Breaking News',
      'APE Weather ( Live Forecast)',
      'APUS Launcher - Theme, Wallpaper, Hide Apps', 'ARY NEWS',
      'ARY NEWS URDU', 'ASOS', 'ASUS Calling Screen',
      'ASUS Cover for ZenFone 2', 'ASUS Quick Memo',
      'ASUS Sound Recorder', 'ASUS SuperNote',
      'AT&T Navigator: Maps, Traffic', 'AT&T Smart Wi-Fi',
      'AT&T Visual Voicemail',
      'AVG Cleaner – Speed, Battery & Memory Booster',
      'Abs Training-Burn belly fat', 'Account Manager',
      'Accounting App - Zoho Books',
      'AccuWeather: Daily Forecast & Live Weather Reports',
      'Acorn TV: World-class TV from Britain and Beyond',
      'Acorns - Invest Spare Change', 'AdWords Express',
      'Ada - Your Health Guide', 'Add Text To Photo',
      'Adobe Acrobat Reader',
      'Adobe Photoshop Express:Photo Editor Collage Maker',
      'Advanced Task Killer', 'Agar.io', 'Age Calculator',
      'Agoda – Hotel Booking Deals', 'Air Traffic', 'AirAsia',
      'AirBrush: Easy Photo Editor', 'Airbnb',
      'Airport + Flight Tracker Radar',
```

'Airway Ex - Intubate. Anesthetize. Train.', 'Akinator',
'AlReader -any text book reader', 'Alarm Clock',
'Alarm Clock Free', 'Alfred Home Security Camera',
'AliExpress - Smarter Shopping, Better Living',
'All Email Providers', 'All Events in City',
'All Football - Latest News & Videos',
'All Football GO- Live Score, Games',
'All Language Translator Free', 'All Maths Formulas',
'All Mental disorders', 'All Social Networks',
'All Video Downloader 2018',
'All-In-One Toolbox: Cleaner, Booster, App Manager',
'All-in-One Mahjong 3 FREE', 'Allegiant',
'Allrecipes Dinner Spinner', "Alto's Adventure", 'Amazon Drive',
'Amazon FreeTime – Kids' Videos, Books, & TV shows',
'Amazon Kindle', 'Amazon Prime Video', 'Amazon Shopping',
'Amazon for Tablets', 'American Airlines', 'Amex Mobile',
'Amino: Communities and Chats', 'Amtrak',
'Anatomy Learning - 3D Atlas', 'Ancestry',
'AndroZip™ FREE File Manager',
'Android Auto - Maps, Media, Messaging & Voice',
'Android Messages', 'Anger of stick 5 : zombie', 'Angry Birds 2',
'Angry Birds Classic', 'Angry Birds Rio', 'Animal Planet GO',
'Animated Photo Editor',
'Anime Avatar Creator: Make Your Own Avatar',
'Anime Manga Coloring Book', 'Anthem Anywhere',
'Anthem BC Anywhere',
'Any.do: To-do list, Calendar, Reminders & Planner',
'Apartment Decorating Ideas',
'Apartment List: Housing, Apt, and Property Rentals',
'Apartment, Home Rental Search: Realtor.com Rentals',
'Apartments & Rentals - Zillow', 'Apartments.com Rental Search',
'Apex Launcher', 'Apk Installer', 'App vault', 'AppLock',
'AppLock - Fingerprint', 'Apple Daily 蘋果動新聞',
'Aprender inglés con Wlingua', 'Archos File Manager', 'Arrow.io',
'Asana: organize team projects', 'Ascape VR: 360° Virtual Travel',
'Asphalt 8: Airborne', 'Associated Credit Union Mobile',
'Asteroids 3D live wallpaper',
'Atlan3D Navigation: Korea navigator',
'AutoCAD - DWG Viewer & Editor',
'AutoScout24 Switzerland – Find your new car',
'Avakin Life - 3D virtual world', 'Aviary Effects: Classic',
'Aviary Stickers: Free Pack', 'Azar', 'Azpen eReader',
'B612 - Beauty & Filter Camera', 'BBC Media Player', 'BBC News',
'BBC Sport', 'BBM - Free Calls & Messages', 'BBVA Compass Banking',
'BBVA Spain', 'BBW Dating & Curvy Singles Chat- LargeFriends',
'BBW Dating & Plus Size Chat', 'BBWCupid - BBW Dating App',
'BELONG Beating Cancer Together', 'BEST CAR SOUNDS',
'BET NOW - Watch Shows', 'BEYBLADE BURST app', 'BIG Launcher',

'BLK - Swipe. Match. Chat.', 'BZWBK24 mobile',
'BaBe - Baca Berita', 'BaBe Lite - Baca Berita Hemat Kuota',
'BaBe+ - Berita Indonesia', 'Babbel – Learn Languages',
'Babbel – Learn Spanish',
'Baby ABC in box! Kids alphabet games for toddlers!',
'Baby Monitor', 'Baby Name Together', 'Baby Panda Care',
'Baby Panda Learns Shapes', 'Baby Panda Musical Genius',
'Baby Panda's Juice Shop',
'Baby Tiger Care - My Cute Virtual Pet Friend', 'Baby puzzles',
'Baca- Berita Terbaru, Informasi, Gosip dan Politik',
'Backgrounds (HD Wallpapers)', 'Backgrounds HD (Wallpapers)',
'BaconReader for Reddit', 'Bad Piggies',
'Badoo - Free Chat & Dating App', 'Bagan - Myanmar Keyboard',
'Banco Itaú', 'Banco do Brasil', 'Bancomer móvil',
'Banfield Pet Health Tracker', 'Bangla Newspaper – Prothom Alo',
'Banjo', 'Bank of America Mobile Banking', 'BankMobile Vibe App',
'Banorte Movil', 'Banque Populaire', 'Barbie Life™',
'Barbie Magical Fashion', 'Barbie™ Fashion Closet',
'Barclays US for Android', 'Barcode Scanner',
'Baritastic - Bariatric Tracker', 'Baseball Boy!',
'Basketball FRVR - Shoot the Hoop and Slam Dunk!',
'Basketball Stars', 'Bathroom Decorating Ideas',
'Battlelands Royale', 'Be A Legend: Soccer',
'BeSoccer - Soccer Live Score', 'BeWild Free Dating & Chat App',
'Beautiful Design Birthday Cake', 'Beautiful Widgets Free',
'Beautiful Widgets Pro', 'Beauty Camera - Selfie Camera',
'Beauty Makeup Snappy Collage Photo Editor - Lidow',
'BeautyPlus - Easy Photo Editor & Selfie Camera',
'Bed Time Fan - White Noise Sleep Sounds', 'Best Car Wallpapers',
'Best Fiends - Free Puzzle Game',
'Best Ovulation Tracker Fertility Calendar App Glow',
'Best Wallpapers Backgrounds(100,000+ 4K HD)',
'Best Wallpapers QHD',
'BestCam Selfie-selfie, beauty camera, photo editor',
'BetterMe: Weight Loss Workouts', 'Betterment',
'BeyondMenu Food Delivery', 'BeyondPod Podcast Manager', 'Bible',
'Big Days - Events Countdown',
'BigOven Recipes, Meal Planner, Grocery List & More',
'BiggerCity: Chat for gay bears, chubs & chasers',
'Bike Computer - GPS Cycling Tracker', 'Binaural Beats Meditation',
'Binaural Beats Therapy', 'BioLife Plasma Services',
'Birdays – Birthday reminder',
'Birds Sounds Ringtones & Wallpapers',
'Black People Meet Singles Date', 'Block Puzzle',
'Block Puzzle Classic Legend !', 'Blogaway for Android (Blogger)',
'Blogger', "Bloglovin'", 'Blood Donor', 'Blood Pressure',
'Blood Pressure Log - MyDiary', 'Blood Pressure(BP) Diary',
'Bloomberg Professional', 'Bloomberg: Market & Financial News',

'Blossom Blast Saga', 'BluTV', 'Blur Image Background',
'Blur Image Background Editor (Blur Photo Editor)',
'Booking.com Travel Deals', 'Bowmasters', 'Box', 'Boxed Wholesale',
"Boys Photo Editor - Six Pack & Men's Suit",
'Brain Waves - Binaural Beats', 'Branch',
'Brasileirão Pro 2018 - Série A e B',
'Breaking News, Local news, Attacks and Alerts Free',
'Brightest Flashlight Free ®', 'Brightest LED Flashlight',
'Brilliant', 'Brit + Co', 'British Airways', 'Browser 4G',
'Bualuang mBanking', 'Bubble Shooter', 'Bubble Shooter 2',
'Bubble Shooter Genies', 'Bubble Shooter Space',
'Bubble Witch 3 Saga', 'Buienradar - weer', 'Build a Bridge!',
'Bukalapak - Jual Beli Online', 'BukuBayi - Perkembangan Bayi',
'Burner - Free Phone Number', 'Buscapé - Ofertas e descontos',
'Business Calendar 2', 'Butterfly Live Wallpaper',
'Buzz Launcher-Smart&Free Theme', 'BuzzFeed: News, Tasty, Quizzes',
'BÁO MỚI - Đọc Báo, Tin Tức 24h', 'C Programming',
'C++ Programming', 'C++ Tutorials', 'CAIXA',
'CALCU™ Stylish Calculator Free', 'CATS: Crash Arena Turbo Stars',
'CBS News', 'CBS Sports App - Scores, News, Stats & Watch Live',
'CBS Sports Fantasy', 'CDL Practice Test 2018 Edition',
'CIA - Caller ID & Call Blocker',
'CM Browser - Ad Blocker , Fast Download , Privacy',
'CM Flashlight (Compass, SOS)',
'CM Launcher 3D - Theme, Wallpapers, Efficient',
'CM Locker - Security Lockscreen', 'CMB Free Dating App',
'CNBC: Breaking Business News & Live Market Data',
'CNN Breaking US & World News', "COOKING MAMA Let's Cook!",
'CVS Caremark', 'CW Seed', 'CWT To Go',
'Cache Cleaner-DU Speed Booster (booster & cleaner)',
'Caf - Mon Compte',
'Calculator - free calculator, multi calculator app',
'Calculator - unit converter', 'Calculator Plus Free',
'Calculator with Percent (Free)', 'Calendar Widget Month + Agenda',
'Calendar+ Schedule Planner App', 'Call Blocker',
'Call Control - Call Blocker', 'Call of Duty:Black Ops Zombies',
'CallApp: Caller ID, Blocker & Phone Call Recorder', 'Caller ID +',
'Calls & Text by Mo+', 'Calls Blacklist - Call Blocker',
'Calm - Meditate, Sleep, Relax', 'Calorie Counter & Diet Tracker',
'Calorie Counter - EasyFit free', 'Calorie Counter - Macros',
'Calorie Counter - MyFitnessPal', 'Calorie Counter - MyNetDiary',
'Cameringo Lite. Filters Camera', 'Candy Bomb',
'Candy Camera - selfie, beauty camera, photo editor',
'Candy Crush Jelly Saga', 'Candy Crush Saga',
'Candy Crush Soda Saga', 'Candy Day', 'Candy Pop Story',
'Candy Smash', 'Candy selfie - photo editor, live filter camera',
'Canva: Poster, banner, card maker & graphic design',
'Canvas Student', 'Cat Sim Online: Play with Cats',

'Caviar - Food Delivery', 'Chakra Cleansing', 'Championat',
'Chapters: Interactive Stories', 'Chase Mobile',
'Chat Rooms, Avatars, Date - Galaxy', 'ChatVideo Meet new people',
'Cheap Flights & Hotels momondo',
'Cheap hotel deals and discounts — Hotellook',
'CheapTickets – Hotels, Flights & Travel Deals',
'Cheapflights – Flight Search',
'Checkout 51: Grocery coupons', 'Choice Hotels',
'Choices: Stories You Play', 'Christian Dating For Free App',
'Chrome Beta', 'Chrome Dev', 'Cinemark Theatres',
'Cisco Webex Meetings', 'Cisco Webex Teams', 'Citi Mobile®',
'Citibanamex Movil', 'Citizens Bank Mobile Banking',
'CityMaps2Go Plan Trips Travel Guide Offline Maps', 'Claro',
'Clash Royale', 'Clash of Clans', 'Color Flashlight', 'Color Road',
'Color Touch Effects', 'Color by Number - Draw Sandbox Pixel Art',
'Color by Number – New Coloring Book',
'ColorFil - Adult Coloring Book', 'ColorNote Notepad Notes',
'ColorSnap® Visualizer', 'Colorfit - Drawing & Coloring',
'Colorful Glitter Neon Butterfly Keyboard Theme',
'Colorfy: Coloring Book for Adults - Free', 'Coloring & Learn',
'Coloring book moana', 'Comedy Central', 'Common Core',
'Comptia A+ 220-901 & 220-902', 'ConnectLine', 'Contacts',
'Contacts+', 'Content Transfer', 'ConvertPad - Unit Converter',
'Cookbook Recipes', 'Cooking Channel', 'Cooking Fever',
"Cooking Madness - A Chef's Restaurant Games", 'Cookpad',
'Cool Reader',
'Couch to 10K Running Trainer', 'Couch to 5K by RunDouble',
'Couchsurfing Travel App',
'Cougar Dating Life : Date Older Women Sugar Mummy',
'Couple - Relationship App', 'Credit Karma', 'Credit Sesame',
'CreditWise from Capital One',
'Crew - Free Messaging and Scheduling',
'Cricbuzz - Live Cricket Scores & News',
'Cricket Visual Voicemail', 'Crossy Road',
'Crunchyroll - Everything Anime',
'Current debit card and app made for teens',
'Curriculum vitae App CV Builder Free Resume Maker',
'Curso de Ingles Gratis', 'Cut the Rope 2',
'Cut the Rope FULL FREE', 'Cycling - Bike Tracker',
'Cymera Camera- Photo Editor, Filter,Collage,Layout',
'Czech Public Transport IDOS', 'DC Comics', 'DC Super Hero Girls™',
'DEAD TARGET: FPS Zombie Apocalypse Survival Games',
'DEER HUNTER 2018', 'DELISH KITCHEN - 無料レシピ動画で料理を楽しく・簡単に！',
'DINO HUNTER: DEADLY SHORES', 'DIY Garden Ideas',
'DIY On A Budget', 'DMV Permit Practice Test 2018 Edition',
'DRAGON BALL LEGENDS', 'DSLR Camera Hd Ultra Professional',
'DStv Now', 'DU Browser—Browse fast & fun',

'DU Recorder – Screen Recorder, Video Editor, Live',
'Daily Workouts - Exercise Fitness Routine Trainer',
'Daily Yoga - Yoga Fitness Plans',
'Dailyhunt (Newshunt) - Latest News, Viral Videos', 'Dairy Queen',
'Daniel Tiger for Parents', 'DashClock Widget',
'Dashlane Free Password Manager',
'Dating App, Flirt & Chat : W-Match',
'Dating for 50 plus Mature Singles – FINALLY', 'Daum Mail - 다음 메일',
"Davis's Drug Guide", "Davis's Drug Guide for Nurses",
'Debonairs Pizza',
'Delivery Club–Доставка еды:пицца,суши,бургер,салат',
'Delta Dental', 'Despegar.com Hoteles y Vuelos',
'Detector de Radares Gratis', 'Device Help',
'Diabetes & Diet Tracker', 'Diabetes:M',
'Diamond Zipper Lock Screen', 'Diary with lock',
'Diary with lock password', 'Dictionary - Merriam-Webster',
'Die TK-App – alles im Griff', 'Digg', 'DigiCal Calendar Agenda',
'Digit Save Money Automatically', 'Digital Alarm Clock', 'Dil Mil',
'Dino War: Rise of Beasts', 'Dinosaur Simulator: Dino World',
'Discover Mobile', 'Disney Heroes: Battle Mode',
'Disney Magic Kingdoms: Build Your Own Magical Park',
'DisneyNOW – TV Shows & Games', 'Divar',
'Do It Later: Tasks & To-Dos', 'Docs To Go™ Free Office Suite',
'Doctor On Demand', 'Doctor Pets', 'Dog Licks Screen Wallpaper',
'Dog Run - Pet Dog Simulator', 'Dog Sim Online: Raise a Family',
"Domino's Pizza USA",
'Domofond Недвижимость. Купить, снять квартиру.', 'Doodle Jump',
'Door Lock Screen', 'DoorDash - Food Delivery',
'Dosecast - Medication Reminder', 'Down Dog: Great Yoga Anywhere',
'Dr. Oetker Rezeptideen', "Dr. Panda & Toto's Treehouse",
'Dr. Panda Restaurant 3', 'Dr. Panda Town: Vacation',
'DraftKings - Daily Fantasy Sports', 'Dragon Hills',
'Draw A Stickman', 'Draw In', 'Draw Your Game',
'Draw a Stickman: EPIC 2',
'Drawing for Kids Learning Games for Toddlers age 3',
'Dream League Soccer 2018', 'DreamTrips', 'DreamWorks Friends',
'Dresses Ideas & Fashions +3000', 'Droid Zap by Motorola',
'DroidAdmin for Android - Advice', 'Dropbox',
'Drugs.com Medication Guide', 'Dude Perfect 2',
'Dumb Ways to Die 2: The Games',
'Dungeon Hunter Champions: Epic Online Action RPG',
"Dunkin' Donuts", 'Duolingo: Learn Languages Free', 'DuraSpeed',
'E*TRADE Mobile', 'Easy - taxi, car, ridesharing',
'Easy Hair Style Design', 'Easy Healthy Recipes',
'Easy Installer - Apps On SD', 'Easy Makeup Tutorials',
'Easy Origami Ideas', 'Easy Recipes', 'Easy Voice Recorder',
'EasyBib: Citation Generator', 'Eat Fit - Diet and Health Free',
'Eat24 Food Delivery & Takeout', 'EatStreet Food Delivery App',

'Ebates: Cash Back, Coupons, Rewards & Savings', 'Ebook Reader',
'Ecobank Mobile Banking', 'Edmodo', 'Educational Games 4 Kids',
'Educational Games for Kids', 'El tiempo de AEMET',
'English with Lingualeo', 'English-Myanmar Dictionary', 'Entel',
'Enterprise Rent-A-Car', 'Episode - Choose Your Story',
'Epocrates Plus', 'Equestria Girls', 'Essential Anatomy 3',
'Essential Resources', 'Etsy: Handmade & Vintage Goods',
'Etta Homes', 'Eurosport',
'Eve Period Tracker - Love, Sex & Relationships App',
'Even - organize your money, get paid early',
'Evernote – Organizer, Planner for Notes & Memos', 'Evie Launcher',
'ExDialer - Dialer & Contacts',
'Expedia Hotels, Flights & Car Rental Travel Deals',
'Expense IQ Money Manager', 'Experian - Free Credit Report',
'Extreme Car Driving Simulator', 'Extreme Coupon Finder',
'Extreme Match', 'Extreme Racing 2 - Real driving RC cars game!',
'EyeCloud', 'EyeEm - Camera & Photo Filter', 'EzCalculator',
'FBReader: Favorite Book Reader', 'FERZU - Furries Social Network',
'FIFA - Tournaments, Soccer News & Live Scores',
'FINAL FANTASY BRAVE EXVIUS', 'FOSSIL Q: DESIGN YOUR DIAL', 'FOX',
'FOX NOW - On Demand & Live TV',
'FOX Sports: Live Streaming, Scores & News',
'FREEDOME VPN Unlimited anonymous Wifi Security',
'FUN Keyboard – Emoji Keyboard, Sticker,Theme & GIF',
'Fabulous: Motivate Me! Meditate, Relax, Sleep',
'Face Filter, Selfie Editor - Sweet Camera', 'Facebook',
'Facebook Ads Manager', 'Facebook Lite', 'Facebook Local',
'Facebook Pages Manager', 'Facetune - Ad Free',
'Fake Call - Fake Caller ID', 'Fallout Shelter',
'Family Album Mitene: Private Photo & Video Sharing',
'Family Dollar',
'Family GPS Tracker and Chat + Baby Monitor Online',
'Family GPS tracker KidControl + GPS by SMS Locator',
'Family Locator - GPS Tracker', 'FamilySearch Tree',
'FanDuel: Daily Fantasy Sports', 'Fancy Widgets',
'Fandango Movies - Times + Tickets', 'Fantasy Football',
'Fantasy Football & NFL News', 'Fantasy Football Manager (FPL)',
'Farm Fruit Pop: Party Time', 'Farm Heroes Saga',
'Farming Simulator 14', 'Farming Simulator 18', 'Fashion in Vogue',
'Fast News', 'Fast Scanner : Free PDF Scan', 'Fast Secure VPN',
'FastMeet: Chat, Dating, Love',
'Fat Burning Workout - Home Weight lose',
'Fate/Grand Order (English)',
'FidMe Loyalty Cards & Deals at Grocery Supermarket',
'File Browser by Astro (File Manager)', 'File Explorer',
'File Manager',
'File Manager -- Take Command of Your Files Easily',
'Files Go by Google: Free up space on your phone',

'FilterGrid - Cam&Photo Editor', 'Filters for B Live',
'Filters for Selfie', 'Final Fantasy XV: A New Empire',
'Financial Times', 'Find Dining Restaurant Finder',
'Find a Way: Addictive Puzzle', 'Find&Save - Local Shopping',
'FindShip', 'Firefox Browser fast & private',
'Firefox Focus: The privacy browser', 'Fishdom', 'Fitbit',
'Flashlight', 'Flashlight & LED Torch',
'Flashlight - Torch LED Light', 'Flashlight HD LED', 'Flickr',
'Flight & Hotel Booking App - ixigo',
'Flightradar24 Flight Tracker', 'Flights',
'Flip the Gun - Simulator Game', 'FlipaClip - Cartoon animation',
'Flipboard: News For Our Time', 'Flipkart Online Shopping App',
'Flipp - Weekly Shopping',
'Flippy Campus - Buy & sell on campus at a discount',
'Floor Plan Creator',
'Flow Free', 'Flowers Live Wallpaper', 'Fly Delta',
'FollowMyHealth®', 'Font Studio- Photo Texts Image',
'Food Calorie Calculator', 'Food Network',
'Fooducate Healthy Weight Loss & Calorie Counter',
'Foot Mercato : transferts, résultats, news, live',
'Football Live Scores', 'ForecaWeather',
'Fortune City - A Finance App', 'Four In A Line',
'Four In A Line Free', 'Foursquare City Guide',
'Foursquare Swarm: Check In', 'Fox Business',
'Fox News – Breaking News, Live Video & News Alerts',
'Fraction Calculator Plus Free', 'Free & Premium VPN - FinchVPN',
'Free Dating & Flirt Chat - Choice of Love',
'Free Dating App & Flirt Chat - Cheers',
'Free Dating App & Flirt Chat - Match with Singles',
'Free Dating App - Meet Local Singles - Flirt Chat',
'Free Dating App - YoCutie - Flirt, Chat & Meet',
'Free Dating Hook Up Messenger',
'Free Foreclosure Real Estate Search by USHUD.com',
'Free Hypnosis', 'Free Live Talk-Video Call',
'Free Panda Radio Music', 'Free Sports TV',
'Free TV Shows App:News, TV Series, Episode, Movies',
'Free VIN Report for Used Cars', 'Free live weather on screen',
'Free phone calls, free texting SMS on free number',
'FreePrints – Free Photos Delivered',
'Freeletics: Personal Trainer & Fitness Workouts',
'Freeme Launcher—Stylish Theme', 'Fresh EBT - Food Stamp Balance',
'FreshBooks Classic', 'Frontback - Social Photos',
'Frozen Free Fall', 'Fruit Block - Puzzle Legend', 'Fruit Ninja®',
'Fruits Bomb', 'Fuelio: Gas log & costs', 'Full Screen Caller ID',
'Fun Kid Racing', 'Fun Kid Racing - Motocross',
'Funny Alarm Clock Ringtones', 'Funny Pics',
'Fuzzy Seasons: Animal Forest', 'G Cloud Backup',
'GANMA! - オリジナル漫画が全話無料で読み放題',

'GCash - Buy Load, Pay Bills, Send Money', 'GMAT Math Flashcards',
'GMAT Question Bank', 'GMX Mail',
'GO Keyboard - Cute Emojis, Themes and GIFs',
'GO Keyboard - Emoticon keyboard, Free Theme, GIF', 'GO Notifier',
'GO SMS Pro - Messenger, Free Themes, Emoji',
'GO Weather - Widget, Theme, Wallpaper, Efficient', 'GPS Map Free',
'GPS Speedometer and Odometer', 'GPS Speedometer, Distance Meter',
'GPS Status & Toolbox',
'GPS Traffic Speedcam Route Planner by ViaMichelin',
'GRE Flashcards', 'GRE Prep & Practice by Magoosh', 'GRE Tutor',
'GS SHOP', 'GUNSHIP BATTLE: Helicopter 3D',
'Galactic Core Free Wallpaper', 'Galaxy Attack: Alien Shooter',
'Game of Thrones: Conquest™',
'Gametime - Tickets to Sports, Concerts, Theater',
'Garden Coloring Book', 'Garden Fruit Legend',
'Garden Photo Frames - Garden Photo Editor', 'Gardenscapes',
'Garena Free Fire', 'Garmin Connect™', 'GasBuddy: Find Cheap Gas',
'Gay Sugar Daddy Dating & Hookup – Sudy Gay',
'Gboard - the Google Keyboard', 'Gems or jewels ?',
'Genius Scan - PDF Scanner', 'Gmail', 'GoBank',
'GoPro (formerly Capture)', 'Goal Live Scores',
'Goibibo - Flight Hotel Bus Car IRCTC Booking App',
'Goku Wallpaper Art', 'Gold Butterfly Keyboard Theme',
'Golden Dictionary (EN-AR)', 'Golden Launcher',
'Goldstar: Live Event Tickets', 'Golf Channel',
'Golf GPS Rangefinder: Golf Pad', 'Golf GPS by SwingxSwing',
'GolfLogix GPS + Putt Breaks',
'GolfNow: Tee Time Deals at Golf Courses, Golf GPS',
'Golfshot Plus: Golf GPS', 'Golfshot: Golf GPS + Tee Times',
'GoodRx Drug Prices and Coupons', 'Google', 'Google Ads',
'Google Duo - High Quality Video Calls', 'Google Earth',
'Google Fit - Fitness Tracking', 'Google Handwriting Input',
'Google Keep', 'Google My Business', 'Google News',
'Google PDF Viewer', 'Google Pay', 'Google Photos',
'Google Primer', 'Google Slides', 'Google Street View',
'Google Translate', 'Google Trips - Travel Planner',
'Google Voice', 'Google+', 'Granny', 'Graphing Calculator',
'Grim Soul: Dark Fantasy Survival',
'Groovebook Photo Books & Gifts', 'GroupMe',
'Groupon - Shop Deals, Discounts & Coupons',
'Grubhub: Food Delivery', 'Guns of Glory',
'Gyft - Mobile Gift Card Wallet', 'H Pack', 'H TV', 'H&M',
'HBO GO: Stream with TV Package', 'HD Camera',
'HD Camera - Best Cam with filters & panorama',
'HD Camera - Quick Snap Photo & Video',
'HD Camera Pro for Android', 'HD Camera Ultra',
'HD Camera for Android', 'HD Movie Video Player',
'HD Video Player', 'HD Widgets', 'HDFC Bank MobileBanking',

'HESI A2 Pocket Prep',
'HISTORY: Watch TV Show Full Episodes & Specials', 'HTC Calendar',
'HTC File Manager', 'HTC Gallery', 'HTC Help', 'HTC Lock Screen',
'HTC Mail', 'HTC Sense Input', 'HTC Sense Input-AR',
'HTC Service ─ DLNA', 'HTC Service—Video Player',
'HTC Social Plugin - Facebook', 'HTC Speak', 'HTC Weather',
"Hacker's Keyboard", 'Hairstyles step by step',
'Hamilton — The Official App', 'Hangouts', 'Happify',
'Happy Birthday Songs Offline', 'Happy Fruits Bomb - Cube Blast',
'Happy Street', 'Harkins Theatres',
'Harry Potter: Hogwarts Mystery', 'HauteLook', 'Hay Day',
'Haystack TV: Local & World News - Free',
'Headspace: Meditation & Mindfulness',
'Health and Nutrition Guide', 'HealtheLife',
'Healthy Recipes Free', 'Helix Jump', 'Hello Kitty Lunchbox',
'Hello Kitty Nail Salon', 'Hello Stars',
'HelloTalk — Chat, Speak & Learn Foreign Languages',
'Hero Hunters', 'Herpes Dating: 1,000K+ Singles',
'Herpes Positive Singles Dating',
'Hide App, Private Dating, Safe Chat - PrivacyHider',
'Hide Something - Photo, Video', 'Hideman VPN',
'High Blood Pressure Symptoms', 'High-Powered Flashlight',
'High-Speed Camera (GIF,Burst)', 'Hill Climb Racing',
'Hill Climb Racing 2', 'Hily: Dating, Chat, Match, Meet & Hook up',
'Hinge: Dating & Relationships', 'HipChat - Chat Built for Teams',
'Hipmunk Hotels & Flights', 'Hitwe - meet people and chat',
'Hole19: Golf GPS App, Rangefinder & Scorecard',
'Home Decor Showpiece Art making: Medium Difficulty',
'Home Scouting® MLS Mobile',
'Home Security Camera WardenCam - reuse old phones',
'Home Street – Home Design Game', 'Home Workout - No Equipment',
'Home Workout for Men - Bodybuilding',
'Home workouts - fat burning, abs, legs, arms,chest', 'HomeWork',
'Homescapes',
'Homesnap Real Estate & Rentals',
'Homestyler Interior Design & Decorating Ideas',
'Homework Planner', 'Honkai Impact 3rd',
'Hopper - Watch & Book Flights',
'Horoscopes – Daily Zodiac Horoscope and Astrology',
'Horses Live Wallpaper',
'Hostelworld: Hostels & Cheap Hotels Travel App',
'Hot Wheels: Race Off',
'HotelTonight: Book amazing deals at great hotels',
'Hotels Combined - Cheap deals',
'Hotels.com: Book Hotel Rooms & Find Vacation Deals',
'Hotspot Shield Free VPN Proxy & Wi-Fi Security', 'Hotstar',
'Hotwire Hotel & Car Rental App', 'Housing-Real Estate & Property']
droplist = OptionMenu(screen1, university, *list1)

```
    droplist.config(width=50)
    university.set('--Select an app--')
    Label(screen1,text="App*:",font=("Open Sans", 13, 'bold'),bg='blue',
fg='white').place(x=100,y=125)
    droplist.place(x=200, y=125)
    droplist1 = OptionMenu(screen1, review, *list2)
    droplist1.config(width=10)
    review.set('--Select review--')
    Label(screen1,text="Review*:",font=("Open Sans", 13, 'bold'),bg='blue',
fg='white').place(x=100,y=175)
    droplist1.place(x=300, y=175)
    L1=Label(screen1,text="App Name",font=("Open Sans", 13, 'bold'),bg='blue', fg='white')
    L1.place(x=220,y=250)
    txt1=Entry(screen1,font=("Open Sans", 13, 'bold'),bg='white', fg='black',width=30)
    txt1.place(x=150,y=275)
    L1=Label(screen1,text="Review Type",font=("Open Sans", 13, 'bold'),bg='blue',
fg='white')
    L1.place(x=220,y=325)
    txt2=Entry(screen1,font=("Open Sans", 13, 'bold'),bg='white', fg='black',width=30)
    txt2.place(x=150,y=350)
    def retrieve_input():
        inputValue=txt1.get()
        app=inputValue
        inputValue=txt2.get()
        r=inputValue
        data = pd.read_csv("C:\\Users\\Siddhesh\\Desktop\\jayesh\\Database 2")

        def senti(app,r):
            if r=='positive' or r=='POSITIVE' or r=='Positive':
                p=data[(data['Sentiment'] == 'Positive')& (data['App']==app)]['Translated_Review']
                screen2 = Toplevel(screen)
                adjustWindow1(screen2)
                screen2.title("POSITIVE")
               # screen2.resizable(True,True)
                Label(screen2, text="POSITIVE", width = '140',height ='2',
font=('calibri',20,'bold'),fg='white',bg='green').place(x=0,y=0)
                Label(screen2, text=p, width = '200',height ='68',
font=('calibri',9,'bold'),fg='white',bg='blue').place(x=300,y=80)

            if r=='negative' or r=='NEGATIVE' or r=='Negative':
                n= data[(data['Sentiment'] == 'Negative') &
(data['App']==app)]['Translated_Review']
                screen2 = Toplevel(screen)
                adjustWindow1(screen2)
                screen2.title("NEGATIVE")
               # screen2.resizable(True,True)
                Label(screen2, text="NEGATIVE", width = '140',height ='2',
font=('calibri',20,'bold'),fg='white',bg='green').place(x=0,y=0)
```

```
            Label(screen2, text=n, width = '200',height ='69',
font=('calibri',9,'bold'),fg='white',bg='blue').place(x=300,y=80)


          if r=='neutral' or r=='NEUTRAL' or r=='Neutral':
            neu= data[(data['Sentiment'] == 'Neutral') &
(data['App']==app)]['Translated_Review']
            screen2 = Toplevel(screen)
            adjustWindow1(screen2)
            screen2.title("NEUTRAL")
            #screen2.resizable(True,True)
            Label(screen2, text="NEUTRAL", width = '140',height ='2',
font=('calibri',20,'bold'),fg='white',bg='green').place(x=0,y=0)
            Label(screen2, text=neu, width = '200',height ='69',
font=('calibri',9,'bold'),fg='white',bg='blue').place(x=300,y=80)


       senti(app,r)
   buttonCommit=Button(screen1, height=1, width=10,
text="SUBMIT",bg='green',fg='white' ,command=lambda:
retrieve_input())#command=lambda: retrieve_input() >>> just means do this when i press the
button
   buttonCommit.place(x=220,y=400)
   #var=str(droplist.grab_current())
   #print(var)

def analysis9():
   global screen7
   screen7=Toplevel(screen)
   screen7.title("DATABASE2")
   adjustWindow1(screen7)
  # screen7.resizable(True,True)
   Label(screen7, text="USER REVIEWS",width = '150',height ='2',
font=('calibri',18,'bold'),fg='white',bg='green').place(x=0,y=0)
   #Button(screen7, text='CATEGORY', width=50, font=("Open Sans", 13,
'bold'),bg='brown', fg='white',command=catl).place(x=50, y=100)
   #Button(screen7, text='MOST DOWNLOADED CATGORY', width=50, font=("Open
Sans", 13, 'bold'),bg='brown', fg='white',command=downloads_amt).place(x=50, y=150)
   #Button(screen7, text='LEAST DOWNLOADED CATGORY', width=50, font=("Open
Sans", 13, 'bold'),bg='brown', fg='white',command=downloads_amtl).place(x=50, y=200)
   Label(screen7,text ="",bg="blue", width='100',height='50').place(x=600, y=75)
   L1=Label(screen7,text="App Name*:",font=("Open Sans", 13, 'bold'),bg='blue',
fg='white')
   L1.place(x=860,y=100)
   txt1=Entry(screen7,font=("Open Sans", 13, 'bold'),bg='white', fg='black')
   txt1.place(x=825,y=125)
   L1=Label(screen7,text="Translated Review*:",font=("Open Sans", 13, 'bold'),bg='blue',
fg='white')
```

```
    L1.place(x=835,y=175)
    txt2=Entry(screen7,font=("Open Sans", 13, 'bold'),bg='white', fg='black')
    txt2.place(x=825,y=200)
    L1=Label(screen7,text="Sentiment*:",font=("Open Sans", 13, 'bold'),bg='blue', fg='white')
    L1.place(x=855,y=250)
    txt3=Entry(screen7,font=("Open Sans", 13, 'bold'),bg='white', fg='black')
    txt3.place(x=825,y=275)
    L1=Label(screen7,text="Sentiment Polarity*:",font=("Open Sans", 13, 'bold'),bg='blue',
fg='white')
    L1.place(x=835,y=325)
    txt4=Entry(screen7,font=("Open Sans", 13, 'bold'),bg='white', fg='black')
    txt4.place(x=825,y=350)
    L1=Label(screen7,text="Sentiment Subjectivity*:",font=("Open Sans", 13,
'bold'),bg='blue', fg='white')
    L1.place(x=825,y=400)
    txt5=Entry(screen7,font=("Open Sans", 13, 'bold'),bg='white', fg='black')
    txt5.place(x=825,y=425)
    def retrieve_input():
        inputValue=txt1.get()
        app=inputValue
        inputValue=txt2.get()
        translated_review=inputValue
        inputValue=txt3.get()
        sentiment=inputValue
        inputValue=txt4.get()
        sentiment_polarity=inputValue
        inputValue=txt5.get()
        sentiment_subjectivity=inputValue

        def get_length1(file_path):
            with open("file_path") as csvfile:
                reader=csv.reader(csvfile)
                reader_list=list(reader)
                return len(reader_list)
            return 1

        def
append_data1(file_path,app,translated_review,sentiment,sentiment_polarity,sentiment_subjec
tivity):

fieldnames=['app','translated_review','sentiment','sentiment_polarity','sentiment_subjectivity']
            next_id=get_length1(file_path)
            with open(file_path,"a",newline='') as csvfile:
                writer=csv.DictWriter(csvfile,fieldnames=fieldnames)

writer.writerow({"app":app,"translated_review":translated_review,"sentiment":sentiment,"se
ntiment_polarity":sentiment_polarity,"sentiment_subjectivity":sentiment_subjectivity})
```

```
        append_data1("C:\\Users\\Siddhesh\\Desktop\\Database
2",app,translated_review,sentiment,sentiment_polarity,sentiment_subjectivity)
    buttonCommit=Button(screen7, height=1, width=10,
text="SUBMIT",bg='green',fg='white' ,command=lambda:
retrieve_input())#command=lambda: retrieve_input() >>> just means do this when i press the
button
    buttonCommit.place(x=875,y=475)

def adjustWindow(window):

    w = 600
    h = 600
    ws = screen.winfo_screenwidth()
    hs = screen.winfo_screenheight()
    x = (ws/2)-(w/2)
    y = (hs/2)-(h/2)
    window.geometry('%dx%d+%d+%d'%(w,h,x,y))
    window.resizable(False,False)
    c=window
   # if window==screen6:
    #   window.resizable(True,True)
    window.configure(background='white')

def adjustWindow1(window):

    w = 1920
    h = 1080
    ws = screen.winfo_screenwidth()
    hs = screen.winfo_screenheight()
    x = (ws/2)-(w/2)
    y = (hs/2)-(h/2)
    window.geometry('%dx%d+%d+%d'%(w,h,x,y))
    window.resizable(False,False)
    c=window
   # if window==screen6:
    #   window.resizable(True,True)
    window.configure(background='white')


def welcome_page(student_info):

    global screen2
    screen2 = Toplevel(screen)
    screen2.title("Welcome")
    adjustWindow(screen2)
    Label(screen2, text="Welcome " +student_info[0][1], width = '42',height ='2',
font=('calibri',22,'bold'),fg='white',bg='green').place(x=0,y=0)
    Label(screen2, text="",bg="blue",width='30',height='50').place(x=0,y=76)
```

```
    Message(screen2, text="'Welcome to the Data Analysis of Playstore."\n\n - By Trio
Techies',width='180', font=("Helvetica",10, 'bold', 'italic'), fg='white', bg='blue', anchor =
CENTER).place(x=10, y=100)

    photo = PhotoImage(file="play.png") # opening left side image - Note: If image is in same
folderthen no need to mention the full path
    label = Label(screen2, image=photo ,text="") # attaching image to the label
    label.place(x=10, y=270)
    label.image = photo
  # it is necessary in Tkinter to keep a instance of image to displayimage in labe
    #label1 = Label(screen2, text="") # attaching image to the label
    #label1.place(x=200, y=78)

    Button(screen2, text='DOWNLOADS', width=20, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=analysis1).place(x=270, y=100)
    Button(screen2, text='REVIEW', width=20, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=analysis2).place(x=270, y=200)
    Button(screen2, text='CATEGORY', width=20, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=analysis3).place(x=270, y=300)
    Button(screen2, text='PRICE', width=20, font=("Open Sans", 13, 'bold'),bg='blue',
fg='white',command=analysis4).place(x=270, y=400)
    Button(screen2, text='ADD NEW DATA', width=20, font=("Open Sans", 13,
'bold'),bg='blue', fg='white',command=analysis7).place(x=270, y=500)
    #Button(screen2,text="Exit", command=do_exit,bg="yellow",fg="black",font=("Open
Sans", 16,"bold")).place(x=0,y=100)



def login_verify():
    global studentID
    connection = pymysql.connect(host='localhost',user='root',passwd="",database='edumate')
    cursor = connection.cursor()
    select_query = "SELECT * FROM student_details where email = '" +
username_verify.get() + "' AND password = '" + password_verify.get() + "';"
    cursor.execute(select_query)
    student_info = cursor.fetchall()
    connection.commit()
    connection.close()
    if student_info:
        messagebox.showinfo("Congratulations","Login Successful")
        studentID = student_info[0][0]
        welcome_page(student_info)
    else:
        messagebox.showerror("Error","Invalid username or password")



def register_user():
```

```
   if (fullname.get() and email.get() and password.get() and repassword.get() and
gender.get()):
       if (university.get() == "--select your university--"):
           Label(screen1,text="Please select your
university",fg="red",font=('Calibri',11),width='30',anchor=W,bg='white').place(x=0,y=570)
           return
       else:
           if (tnc.get()):
               if (re.match("^.+@(\[?)[a-zA-Z0-9-.]+.([a-zA-Z]{2,3}|[0-9]{1,3})(]?)$",
email.get())):
                   if (password.get() == repassword.get()):
                       gender_value = 'male'
                       if (gender.get()==2):
                           gender_value='female'
                       connection =
pymysql.connect(host='localhost',user='root',passwd="",database='edumate')
                       cursor = connection.cursor()
                       insert_query = "INSERT INTO
student_details(fullname,email,password,gender,university)VALUES('"+ fullname.get() + "',
'"+ email.get() + "', '"+ password.get() + "', '"+ gender_value + "', '"+ university.get() + "' );"
                       cursor.execute(insert_query)
                       connection.commit()
                       connection.close()
                       Label(screen1,text="Registration
Success",fg='green',font=('calibri',10),width='30',anchor=W,bg='white').place(x=0,y=570)
                       Button(screen1,text='Proceed to Login ->', width=20,font=('open
sans',10,'bold'),bg='brown',fg='white',command=screen1.destroy).place(x=170,y=576)
                   else:
                       Label(screen1, text="Password does not match", fg="red", font=("calibri",
11), width='30', anchor=W, bg='white').place(x=0, y=570)
                       return
               else:
                   Label(screen1, text="Please enter valid email id", fg="red", font=("calibri", 11),
width='30', anchor=W, bg='white').place(x=0, y=570)
                   return
           else:
               Label(screen1, text="Please accept the agreement", fg="red",font=("calibri", 11),
width='30', anchor=W, bg='white').place(x=0, y=570)
               return
   else:
       Label(screen1, text="Please fill all the details", fg="red",font=("calibri", 11), width='30',
anchor=W, bg='white').place(x=0, y=570)
       return



def register():
   global screen1, fullname, email, password, repassword,university,gender,tnc
```

```
    fullname = StringVar()
    email = StringVar()
    password = StringVar()
    repassword = StringVar()
    university = StringVar()
    gender = IntVar()
    tnc = IntVar()
    screen1 = Toplevel(screen)
    screen1.title("Registration")
    adjustWindow(screen1)
    Label(screen1, text = "REGISTRATION FORM", width = '50', height='2',
font=("Calibri",22,'bold'),fg='white',bg='green').pack()
    Label(screen1,text ="",bg="blue", width='72',height='30').place(x=45, y=120)
    Label(screen1, text="Full Name*:",font=("Open
Sans",11,'bold'),fg='white',bg='blue',anchor=W).place(x=150,y=160)
    Entry(screen1,textvar=fullname).place(x=300,y=160)
    Label(screen1, text="Email ID*:",font=("Open
Sans",11,'bold'),fg='white',bg='blue',anchor=W).place(x=150,y=210)
    Entry(screen1,textvar=email).place(x=300,y=210)
    Label(screen1, text="Gender*:",font=("Open
Sans",11,'bold'),fg='white',bg='blue',anchor=W).place(x=150,y=260)

Radiobutton(screen1,text="male",variable=gender,value=1,bg='blue',fg='red').place(x=300,y
=260)

Radiobutton(screen1,text="female",variable=gender,value=2,bg='blue',fg='red').place(x=370,
y=260)
    Label(screen1, text="University*:", font=("Open Sans", 11, 'bold'), fg='white', bg='blue',
anchor=W).place(x=150, y=310)
    list1 = ['Mumbai University', 'Savitribai Phule Pune Univeristy','Gujarat Technological
University', 'JNTU Kakinada', 'University of Delhi', 'Anna University']
    droplist = OptionMenu(screen1, university, *list1)
    droplist.config(width=17)
    university.set('--select your university--')
    droplist.place(x=300, y=305)
    Label(screen1, text="Password*:",font=("Open
Sans",11,'bold'),fg='white',bg='blue',anchor=W).place(x=150,y=360)
    Entry(screen1, textvar=password, show="*").place(x=300, y=360)
    Label(screen1, text="Re-Password*:", font=("Open Sans", 11, 'bold'), fg='white', bg='blue',
anchor=W).place(x=150, y=410)
    entry_4 = Entry(screen1, textvar=repassword, show="*")
    entry_4.place(x=300, y=410)
    Checkbutton(screen1, text="I accept all terms and conditions", variable=tnc, bg='blue',
font=("Open Sans", 9, 'bold'), fg='red').place(x=175, y=450)
    Button(screen1, text='Submit', width=20, font=("Open Sans", 13, 'bold'), bg='green',
fg='white',command=register_user).place(x=170, y=490)
```

```
def main_screen():
    global screen, username_verify,password_verify
    screen=Tk()
    username_verify = StringVar()
    password_verify = StringVar()
    screen.title("PLAYSTORE")
    adjustWindow(screen)
    Label(screen,text="PLAYSTORE - DATA ANALYSIS", width="500",
height="2",font=("Calibri",22,'bold'),fg='white',bg='green').pack()
    Label(text="",bg='white').pack()
    Label(screen, text="", bg='blue',width='50',
height='20').place(relx=0.5,rely=0.4,anchor=CENTER)
    Label(screen, text="Please enter details below to login", bg='blue', fg='white').pack()
    Label(screen,text="",bg="blue").pack()
    Label(screen,text="Username *",font=("Open
Sans",10,'bold'),bg="blue",fg='white').pack()
    Entry(screen, textvar=username_verify).pack()
    Label(screen, text="", bg='blue').pack()
    Label(screen, text="Password * ", font=("Open Sans", 10, 'bold'), bg='blue',
fg='white').pack()
    Entry(screen, textvar=password_verify, show="*").pack()
    Label(screen, text="", bg='blue').pack()
    Button(screen, text="LOGIN", bg="green", width=15, height=1, font=("Open Sans", 10,
'bold'), fg='white', command=login_verify).pack()
    Label(screen, text="", bg='blue').pack()
    Button(screen, text="New User? Register Here", height="1", width="30", bg='green',
font=("Open Sans", 10, 'bold'), fg='white', command=register ).pack()
    screen.mainloop()
   # Button(screen,text="Exit", command=do_exit,bg="yellow",fg="black",font=("Open
Sans", 16,"bold")).place(x=0,y=100)

main_screen()
```