TV SHOW POPULARITY ANALYSIS

END TERM REPORT

by

AISHWARYA, RAMAN SINGH, SWETA, HARSHINI
Section: K18SJ
Roll Numbers: 48,49,50,53

Department of Intelligent Systems
School of Computer Science Engineering
Lovely Professional University, Jalandhar

April-2020

STUDENT DECLARATION

This is to declare that this report has been written by us. No part of the report is copied from other sources. All information included from other sources have been duly acknowledged. We aver that if any part of the report is found to be copied, we are shall take full responsibility for it.

BY: Aishwarya Dhiman -Roll number: 48
: Raman Singh - Roll number: 49
: Sweta Saha - Roll number: 50
: Harshini - Roll number: 53

PLACE: **LPU**
DATE**: 1ST April,2020**

TABLE OF CONTENTS

BONAFIDE CERTIFICATE

Certified that this project report “TV SHOW POPULARITY ANALYSIS “ is the bonafide work of”Aishwarya,Raman,Sweta,Harshini” who carried out the project work under my supervision.

Dipen Saini
Assistant Professor
UID: 23681
Department of Intelligent Systems

BACKGROUND AND OBJECTIVES OF THE PROJECT ASSIGNED.

TV SHOW POPULARITY ANALYSIS Project refers to a system which display the current popularity of any tv show present in the database according to the votes and ratings.

Most of the television shows which are being telecast nowadays are reality shows specializing in dancing, singing, and acting. We conclude to build such a system that will recognize people's sentimental comments on TV shows. The vote from the viewer will be extracted along with the viewer details such as gender, location, etc…The ratings will be gathered from various sources and the entry will be maintained into the excel sheet. The excel file will contain peoples name, vote and rating etc. Based on people's comment and sentiments, the TV Show popularity will be rated accordingly. Admin will Login into the system and can perform task such as Adding pages, maintaining entries, viewing graphs and printing the graphs. System allows admin to add pages by defining the name of the page and link of that page. All the entries from people are maintained by the admin in an excel sheet. The entries may contain name, email id, age, gender, location, likes-dislikes and their sentimental comment. Based on the people's comment, a graph will be generated by the system which will be categorized as age, gender, location and god or bad comments. Admin can also print the system generated graph for maintaining a hard copy records. Visitors can view TV show popularity data in a graphical representation in pie charts and bar charts. Visitor can also view the popular show rating as well as the top show in a country.

People can rely on this application with great ease in order to know about the popularity of the TV shows. This will be one of the applications that the final year students can work on in real time world. The user interface will be simple and easy to understand. The features that can be included in the TV show popularity analysis using data mining application are as follows:

- **Comment analysis:** The comments whether it is good or bad can be predicted through the use of this application.
- **Trending shows**: Easy prediction of the trending shows can be obtained through this application.
- **Graphical representation**: Graphical representation of the TV shows related to the analysis done can be given with great ease through this application.
- **Easy import and export:** Through the use of this application import and export of the data will be easier without any difficulty.
- **Popular shows**: The visitors will come to know about the popular shows with great ease.
- **Easy access**: This application can be accessed anytime and anywhere from the world.
- **User friendly**: This application will be user friendly since the user interface will be simple and easy to understand even by the common man.

DESCRIPTION OF WORK DIVISION IN TERMS OF ROLES AMONG STUDENTS.

AISHWARYA DHIMAN	The login page was built using gui by her, report is made by her and linking of all modules in the project the project was done by her
HARSHINI	She prepared and collected all the data for excel sheet and helped to link the database with it.
RAMAN SINGH	He organised all the facts and information provided by excel sheet into the pie chart and hence depicting the popularity of different tv-shows
SWETA SAHA	She helped in abstracting knowledge from the data and linking the login page with the main file

CODE EXECUTED IN THE PROJECT

In [3]:

```
from tkinter import Toplevel, Button, Tk, Menu
from tkinter import *
from tkinter import messagebox
import csv
import numpy as np
import pandas as pd
from collections import Counter
from  matplotlib import pyplot as plt
top = Tk()

top.geometry('1000000x10000000')
def display():
 {
plt.style.use("fivethirtyeight")


data=pd.read_csv('tvshow.csv')
print(data )
sno=list(data['sno.'])
showname=list(data['show name'])
rating=list(data['rating'])
votes=list(data['votes'])


print(sno,showname,rating,votes,sep='\n')

plt.barh(showname[0:30],rating[0:30],label='show popularity',)
plt.legend(loc='best',)
plt.ylabel('showname')
plt.xlabel('rating')
plt.axis(xmax=10)
plt.subplots_adjust(left=0.27,right=0.96,bottom=0.08,top=1)

plt.show()
plt.pie(rating[0:20],labels=showname[0:20],autopct='%1.1f%%',rotatelabels=False)
plt.subplots_adjust(left=0.03,right=0.96,bottom=0,top=1)
```

```
plt.show()


import requests
import pandas as pd
from bs4 import BeautifulSoup
import pickle


url = "http://www.imdb.com/chart/toptv/"
r = requests.get(url)
html = r.text
html[0:200]

# Use Beautiful soup to extract the imdb numbers from the webpage
soup = BeautifulSoup(html, "lxml")

#soup = BeautifulSoup(r.content, features="html")

# Scrape the IMDb numbers for the 250 top rated shows

show_list = []
for tbody in soup.findAll('tbody', class_='lister-list'):
    for title in tbody.findAll('td', class_='titleColumn'):
        show_list.append(str(title.findAll('a')).split("/")[2])

print(show_list)




DO_NOT_RUN = True     # Do not run when notebook is loaded to avoid unnecessary calls to the API

if not DO_NOT_RUN:
    shows = pd.DataFrame()
    for show_id in show_list:
```

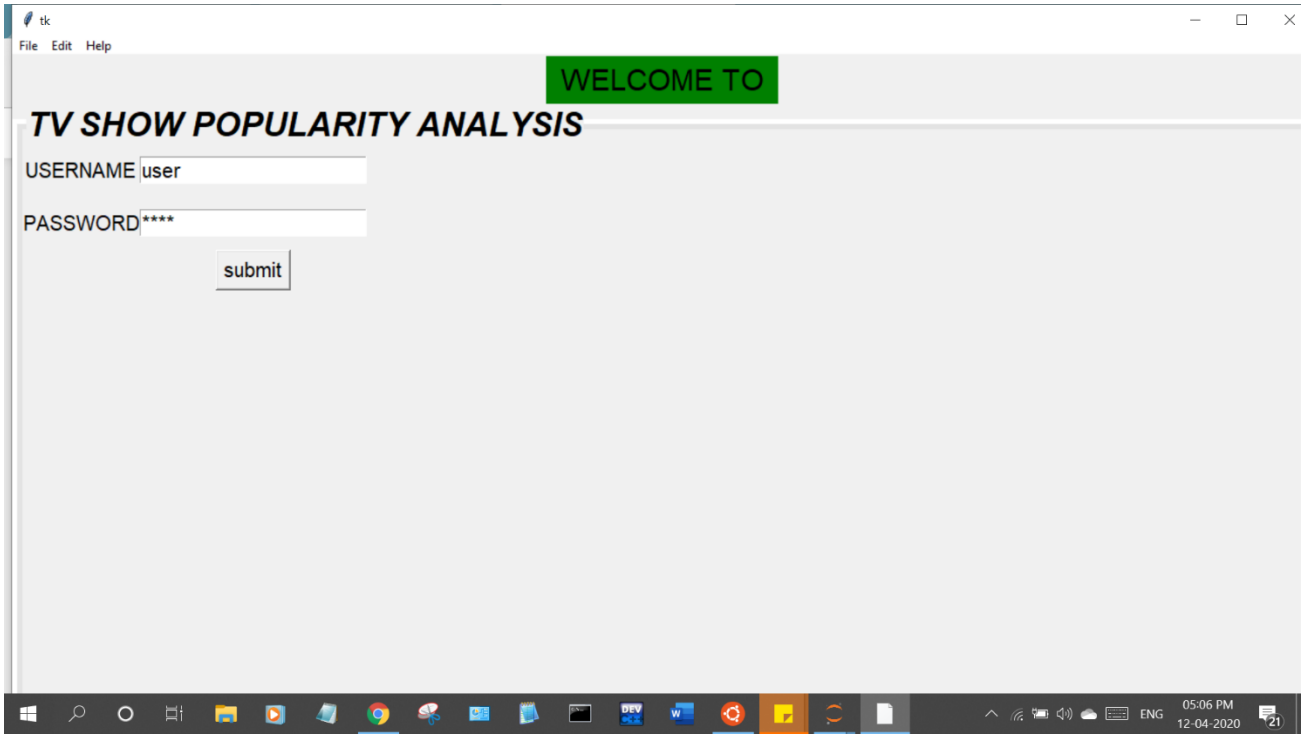
```
DO_NOT_RUN = True     # Do not run when notebook is loaded to avoid unnecessary calls to the API

if not DO_NOT_RUN:
    shows = pd.DataFrame()
    for show_id in show_list:
            try:
                print (show_id)
                # Get the tv show info from the api
                url = "http://api.tvmaze.com/lookup/shows?imdb=" + show_id
                r = requests.get(url)

                # convert the return data to a dictionary
                json_data = r.json()

                # load a temp dataframe with the dictionary, then append to the composite dataframe
                temp_df = pd.DataFrame.from_dict(json_data, orient='index', dtype=None)
                ttemp_df = temp_df.T     # Was not able to load json in column orientation, so must transpose
                shows = shows.append(ttemp_df, ignore_index=True)
            except:
                print (show_id)# " could not be retrieved from api"

    shows.head()
 }

msg = Message(top, text="WELCOME TO",width=5000,font=('arial',22))
msg.config(bg='green')
msg.pack()
menubar = Menu(top)
file = Menu(menubar, tearoff=0)
file.add_command(label="New")
file.add_command(label="Open")
file.add_command(label="Save")
file.add_command(label="Save as...")
file.add_command(label="Close")
```

```
file.add_separator()

file.add_command(label="Exit", command=top.quit)

menubar.add_cascade(label="File", font=('arial',15),menu=file)
edit = Menu(menubar, tearoff=0)
edit.add_command(label="Undo")

edit.add_separator()

edit.add_command(label="Cut")
edit.add_command(label="Copy")
edit.add_command(label="Paste")
edit.add_command(label="Delete")
edit.add_command(label="Select All")

menubar.add_cascade(label="Edit", menu=edit)
help = Menu(menubar, tearoff=0)
help.add_command(label="About")
menubar.add_cascade(label="Help", menu=help)




f1=LabelFrame(top,text="TV SHOW POPULARITY ANALYSIS",font=('arial','25','bold italic'),bd=5,borderwidth=10,relief=RAISED)
f1.pack(side=LEFT,fill=BOTH,expand=1)


user=Label(f1,text="USERNAME",font='25',width='10',height='2').grid(row=0,column=1)
password=Label(f1,text="PASSWORD",font='25',width='10',height='2').grid(row=1,column=1)
uservalue=StringVar()
passwordvalue=StringVar()


ue=Entry(f1,textvariable=uservalue,width='20',font=('arial',15)).grid(row=0,column=2)
```

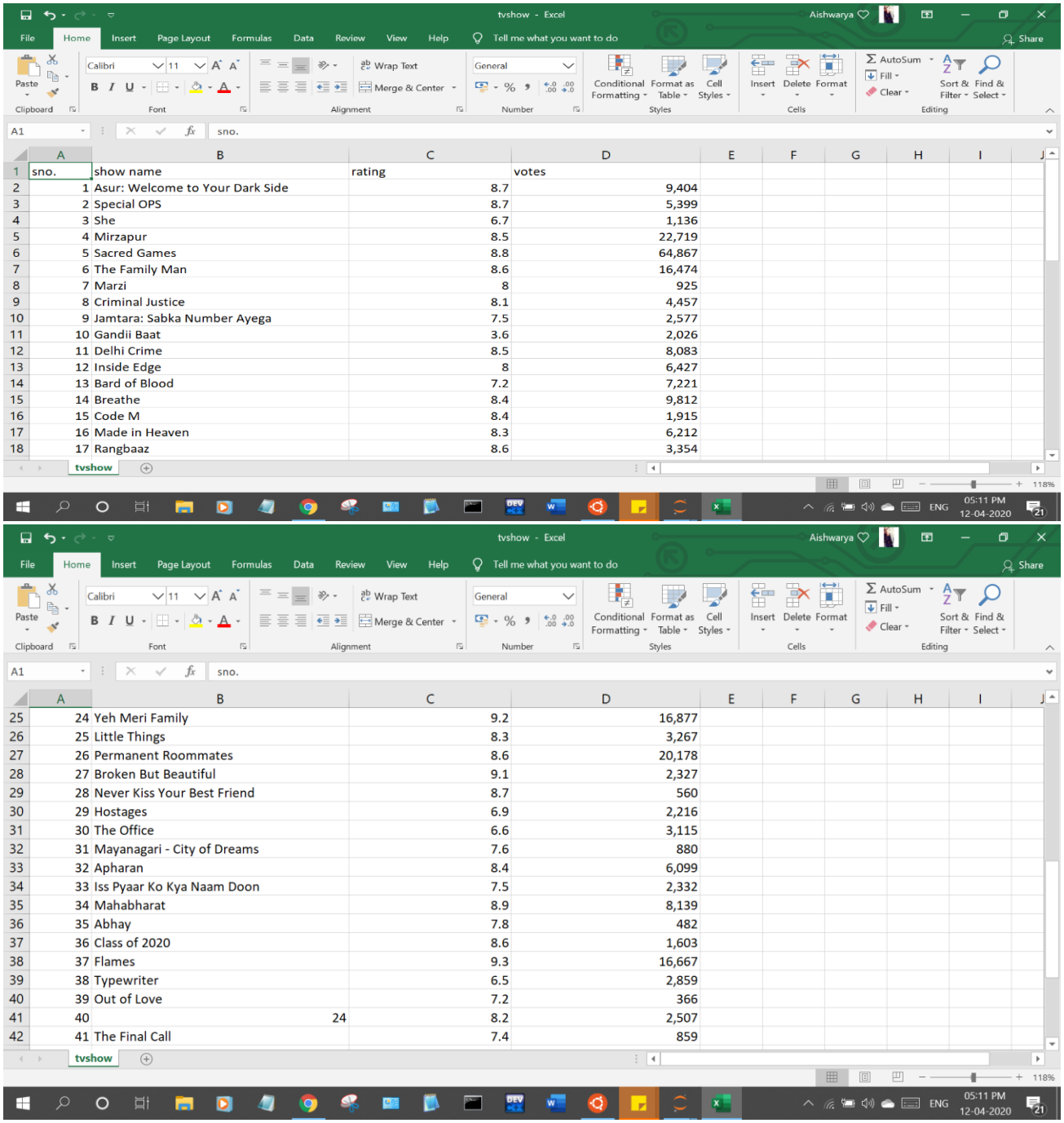
```
menubar.add_cascade(label="File", font=('arial',15),menu=file)
edit = Menu(menubar, tearoff=0)
edit.add_command(label="Undo")

edit.add_separator()

edit.add_command(label="Cut")
edit.add_command(label="Copy")
edit.add_command(label="Paste")
edit.add_command(label="Delete")
edit.add_command(label="Select All")

menubar.add_cascade(label="Edit", menu=edit)
help = Menu(menubar, tearoff=0)
help.add_command(label="About")
menubar.add_cascade(label="Help", menu=help)




f1=LabelFrame(top,text="TV SHOW POPULARITY ANALYSIS",font=('arial','25','bold italic'),bd=5,borderwidth=10,relief=RAISED)
f1.pack(side=LEFT,fill=BOTH,expand=1)


user=Label(f1,text="USERNAME",font='25',width='10',height='2').grid(row=0,column=1)
password=Label(f1,text="PASSWORD",font='25',width='10',height='2').grid(row=1,column=1)
uservalue=StringVar()
passwordvalue=StringVar()


ue=Entry(f1,textvariable=uservalue,width='20',font=('arial',15)).grid(row=0,column=2)
pe=Entry(f1,textvariable=passwordvalue,width='20',font=('arial',15),show="*").grid(row=1,column=2)

b1=Button(f1,text='submit',command=display,relief=RAISED,font=('arial',15)).grid(row=2,column=2)

top.config(menu=menubar)
top.mainloop()
```

IMPLEMENTATION OF SCHEDULED WORK OF PROJECT

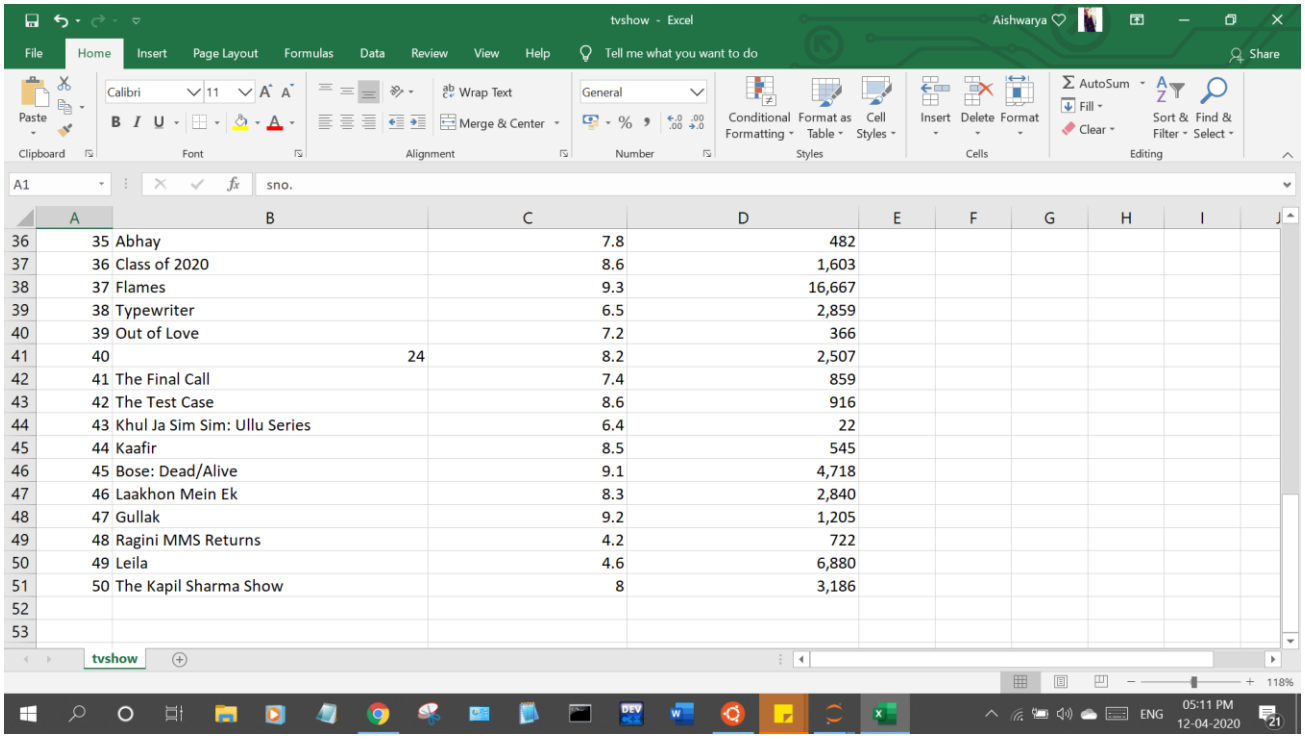
LOGIN PAGE:



DATA IN EXCEL SHEET:

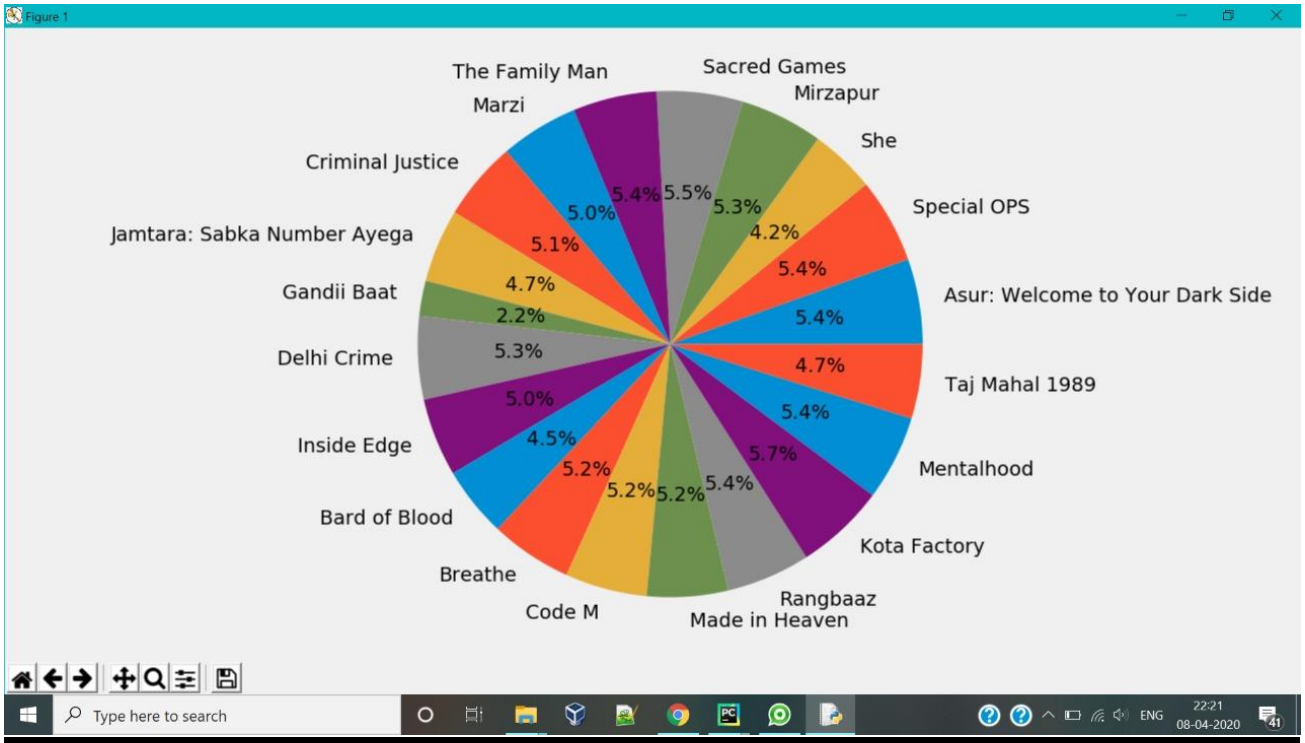


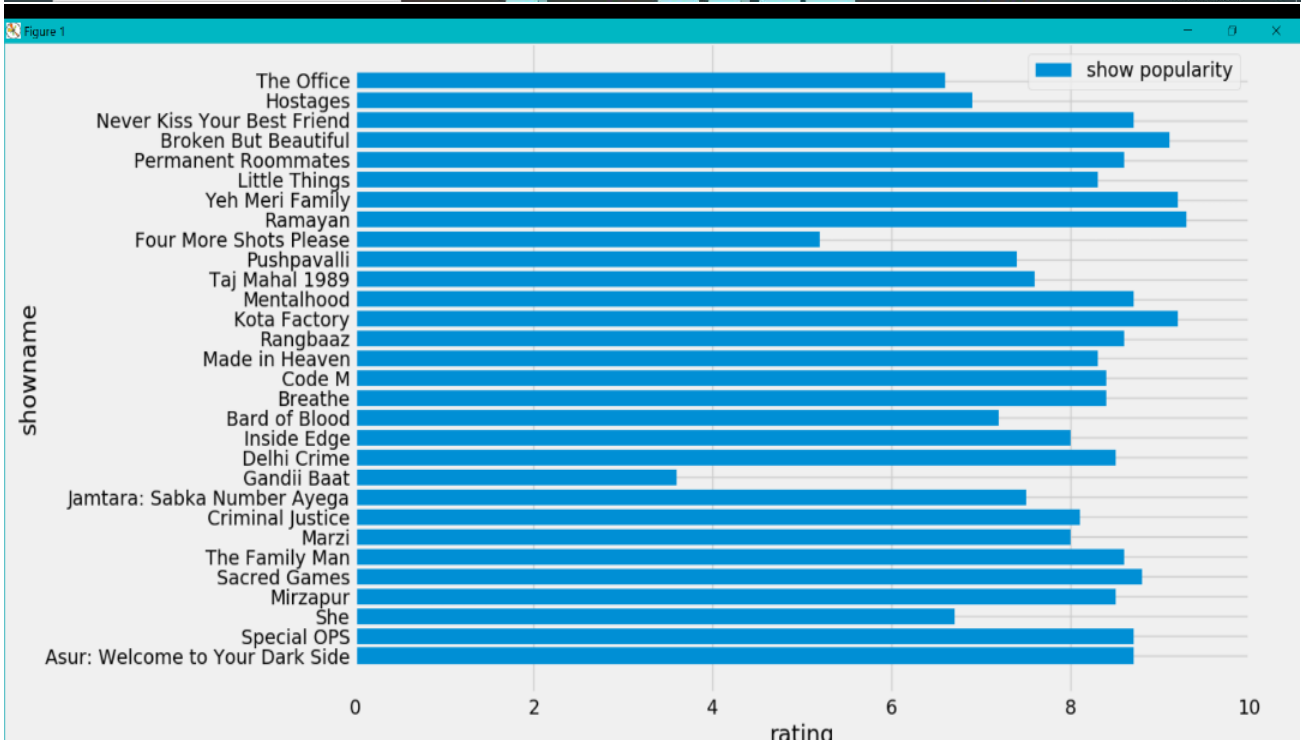
sno.	show name	rating	votes
1	Asur: Welcome to Your Dark Side	8.7	9,404
2	Special OPS	8.7	5,399
3	She	6.7	1,136
4	Mirzapur	8.5	22,719
5	Sacred Games	8.8	64,867
6	The Family Man	8.6	16,474
7	Marzi	8	925
8	Criminal Justice	8.1	4,457
9	Jamtara: Sabka Number Ayega	7.5	2,577
10	Gandii Baat	3.6	2,026
11	Delhi Crime	8.5	8,083
12	Inside Edge	8	6,427
13	Bard of Blood	7.2	7,221
14	Breathe	8.4	9,812
15	Code M	8.4	1,915
16	Made in Heaven	8.3	6,212
17	Rangbaaz	8.6	3,354
24	Yeh Meri Family	9.2	16,877
25	Little Things	8.3	3,267
26	Permanent Roommates	8.6	20,178
27	Broken But Beautiful	9.1	2,327
28	Never Kiss Your Best Friend	8.7	560
29	Hostages	6.9	2,216
30	The Office	6.6	3,115
31	Mayanagari - City of Dreams	7.6	880
32	Apharan	8.4	6,099
33	Iss Pyaar Ko Kya Naam Doon	7.5	2,332
34	Mahabharat	8.9	8,139
35	Abhay	7.8	482
36	Class of 2020	8.6	1,603
37	Flames	9.3	16,667
38	Typewriter	6.5	2,859
39	Out of Love	7.2	366
40	24	8.2	2,507
41	The Final Call	7.4	859



	A	B	C	D	E	F	G	H	I
36	35	Abhay	7.8	482					
37	36	Class of 2020	8.6	1,603					
38	37	Flames	9.3	16,667					
39	38	Typewriter	6.5	2,859					
40	39	Out of Love	7.2	366					
41	40	24	8.2	2,507					
42	41	The Final Call	7.4	859					
43	42	The Test Case	8.6	916					
44	43	Khul Ja Sim Sim: Ullu Series	6.4	22					
45	44	Kaafir	8.5	545					
46	45	Bose: Dead/Alive	9.1	4,718					
47	46	Laakhon Mein Ek	8.3	2,840					
48	47	Gullak	9.2	1,205					
49	48	Ragini MMS Returns	4.2	722					
50	49	Leila	4.6	6,880					
51	50	The Kapil Sharma Show	8	3,186					
52									
53									

RESULTING GRAPH:





TECHNOLOGIES AND FRAMEWORK USED.

GUI

GUI stands for **Graphical User Interface.** GUI permits users to use the graphics to interact with an operating system. In graphical user interface, menus are provided such as : windows, scrollbars, buttons, wizards, painting pictures, alternative icons etc. It's intuitive, simple to find out and reduces psychological feature load. In GUI, the information is shown or presented to the user in any form such as: plain text, videos, images, etc.

PYTHON GUI – TKINTER

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

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.

PICKLE

Python pickle module is used for serializing and de-serializing a Python object structure. Any object in Python can be pickled so that it can be saved on disk. What pickle does is that it "serializes" the object first before writing it to file. Pickling is a way to convert a python object (list, dict, etc.) into a character stream. The idea is that this character stream contains all the information necessary to reconstruct the object in another python script.

BEAUTIFUL SOAP

Beautiful Soup is a Python library for pulling data out of HTML and XML files. It works with your favorite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree. It

commonly saves programmers hours or days of work. Beautiful Soup is a Python library for pulling data out of HTML and XML files. It works with your favorite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree. It commonly saves programmers hours or days of work.

REQUEST MODULE

The requests module allows you to send HTTP requests using Python.
The HTTP request returns a Response Object with all the response data (content, encoding, status, etc).

SWOT ANALYSIS ACHIEVED IN PROJECT

Strengths:

It includes simple modules which can easily be edited or changed if needed.

Weaknesses:

The admin portal lacks differentiation between the roles to play.

Opportunities :

If developed in a more better way, can become a good software which can be used in industries.

Threats :

The username and password is not secure enough to sustain privacy