

THE INTELLIGENT RECOMMENDER SYSTEM

END TERM REPORT

by

NAME OF THE CANDIDATE(S)

NAYANIKA DEB (A01)

VAISHNAVI JAISWAL(A17)

RESHAM CHANEY(A27)

DIKSHA SHARMA(B66)

Section : K18SJ



**Department of Intelligent Systems
School of Computer Science and 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 has been duly acknowledged. We aver that if any part of the report is found to be copied, we shall take full responsibility for it.

Resham Chaney
Rollnumber: 27

Diksha Sharma
Rollnumber: 66

Vaishnavi Jaiswal
Rollnumber: 17

Nayanika Deb
Rollnumber: 01

Date
13-04-20

TABLE OF CONTENTS

<i>TITLE</i> _____	<i>PAGE NUMBER</i>
1.BONAFIDE CERTIFICATE_____	04
2.BACKGROUND AND OBJECTIVE OF THE PROJECT_____	5-7
3.DESCRPTION OF THE PROJECT_____	07
4.WORK DIVISION_____	08
5.IMPLEMENTATION OF THE SCHEDULED PROJECT_____	9-11
6.TECHNOLOGIES AND THE FRAMEWORK USED_____	11-15
7.SWOT ANALYSIS_____	15-16
8.CODE_____	18-23

BONAFIDE CERTIFICATE

Certified that this project report “**The Intelligent Recommender System**” is the bonafide work of “**Nayanika Deb, Vaishnavi Jaiswal, Resham Chaney, Diksha Sharma.**” who carried out the project work under my supervision.

<<Signature of the Supervisor>>

<<Name of supervisor>>

<<Academic Designation>>

<<ID of Supervisor>>

<<Department of Supervisor>>

BACKGROUND AND OBJECTIVE OF THE PROJECT

BACKGROUND OF THE PROJECT:

The huge amount of data that is available on the Internet has led to the development of recommendation systems in the world. This project proposes the use of soft computing techniques to develop recommendation systems. Recommendation systems are basically carrying the idea of observing people's opinions and choices and using it to guide other users. It addresses the limitations of current algorithms used to implement recommendation systems, evaluation of experimental results, and conclusion. This report provides a detailed summary of the project "Recommendation System". This report includes a description of the topic, system architecture, and provides a detailed description of the work done till point. Included in the report are the detailed descriptions of the work done: snapshots of the implementations, various approaches, and tools used so far. The report also includes the project schedule and deliverables.

Recommendation system is an information filtering technique, which provides users with information, in which he/she may be interested in depending on some criteria used to collect data from several users.

MOTIVATION OF THE PROJECT:

The explosive growth of the World Wide Web (www), the emerging popularity of e-commerce and social networks have provided access to a large quantity of information, which was previously inaccessible. Gathering data is not a problem anymore, but the extraction of useful information and its presentation to the user in a relevant way is. Recommender systems have been developed to help fill the gap between information collection and analysis, by filtering all available information and presenting the most relevant items to the user. The recommender system helps enhance the capacity and efficiency of this process. The biggest challenge of this type of system is finding the perfect

match between those recommending and those receiving the recommendation; that is, defining and discovering the relation between their interests.

Recommender systems are widely used in several different domains for the recommendation of articles, music, movies, and even people. Portals such as Amazon, Flipkart use recommendation systems to suggest products to their customers. Meanwhile, social networks such as LinkedIn and Facebook use them to suggest new contacts. To accomplish that, the most used techniques employed in recommender systems are the collaborative filtering and content-based systems. The collaborative filtering does not take into account the type of items, nor their attributes. It takes exclusively into account the expressed opinion about the other items in order to make recommendations. Meanwhile, content-based filtering uses the knowledge it has of the items and their attributes to make recommendations.

In collaborative filtering recommendations are made based on predictions of user preferences resulting in interactions between other users. This type of filtering usually offers a higher degree of surprise to the user with good recommendations and, in some cases, may offer totally irrelevant contents.

The content based filtering is based on the premise that the user would like to see similar items as to those previously seen by him. With information on a specific content and data about a specific user that can be related to this information, it is possible to define the relation between user and content.

OUTCOME OF THE PROJECT:

The main objective of designing a recommendation system is to facilitate the user with the organised contents and the content in which the user is interested. So if the service can offer the personalized feed (built according to the correlation with the recent purchases) and understands the user's needs (from similar look-a-like audience), it tends him to buy more what lead both to the customer satisfaction and increase in sales. When a user sees the personalized feed, generated by the

recommendation system: no matter we are talking about music, books, news, movies or e-commerce – he feels less stress and more connected with the service.

Therefore the main outcome of our project is to provide user with the content they are interested in.

DESCRIPTION OF THE PROJECT

The project designed by us tells the user among the best available options say mobile phones. The user chooses among the different types of recommendation systems available.

The project searches for the best options in the database of some sites and represents the results obtained using graphs. The project tells the user about the names of the product and the prices along with the ratings given to the product.

Our project will provide the user with the top ten recommendations for the product and will suggest the user for the best ones with the help of some graphs.

The project also uses a part of GUI to give it an interactive look.

WORK DIVISION

Nayanika Deb: Some of the GUI part was made by her. She made the GUI using tkinter. The search engine was made by her. Upon searching a result in the search engine and clicking the search button, three list boxes display the details of the result. That was done by her using frames ,listboxes and buttons . In the report she designed the certificate, declaration and work division part.

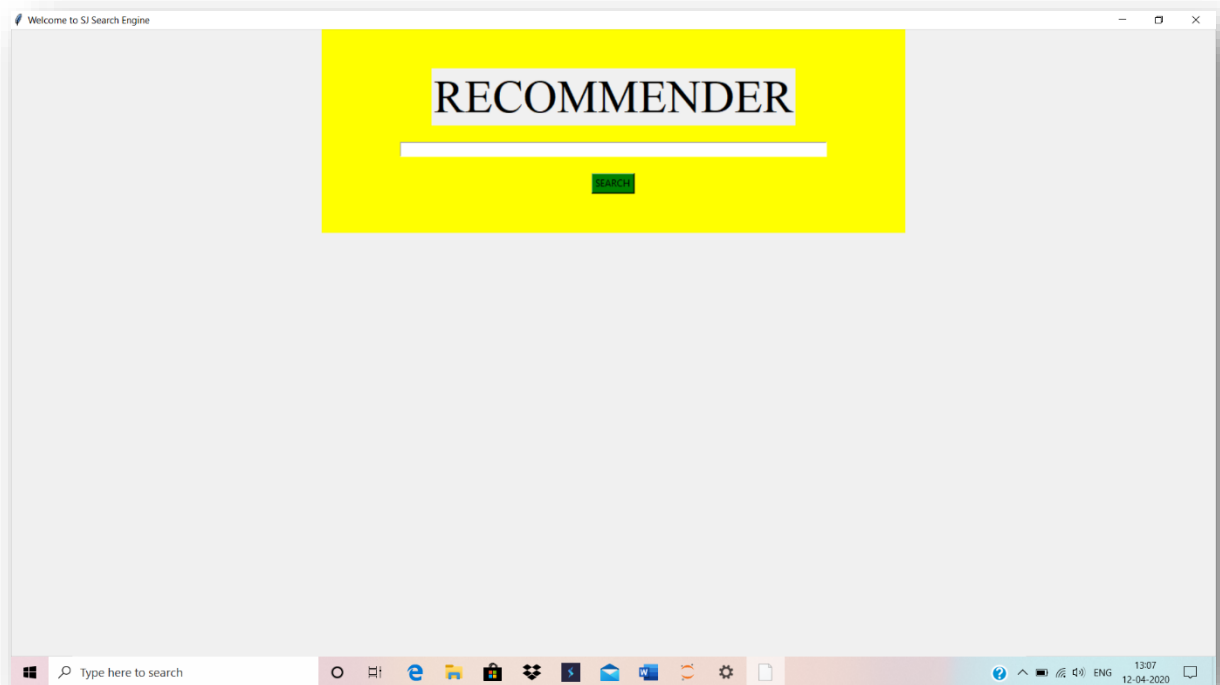
Vaishnavi Jaiswal: She played a role in the backend work by fetching the data from the internet to make the recommender system. She helped in fetching data regarding based on ranking on internet along with price and ranking number. For the report she designed the cover page, along with the SWOT Analysis and technologies and framework used.

Resham Chaney: Major part of the GUI was designed by her. She made the graphs using matplotlib to display the results and also incorporated the graphs in the GUI. She compiled the entire project to incorporate everything into the GUI. The frontend part was majorly done by her. In the report too, she extended her help in the implementation pictorial representation.

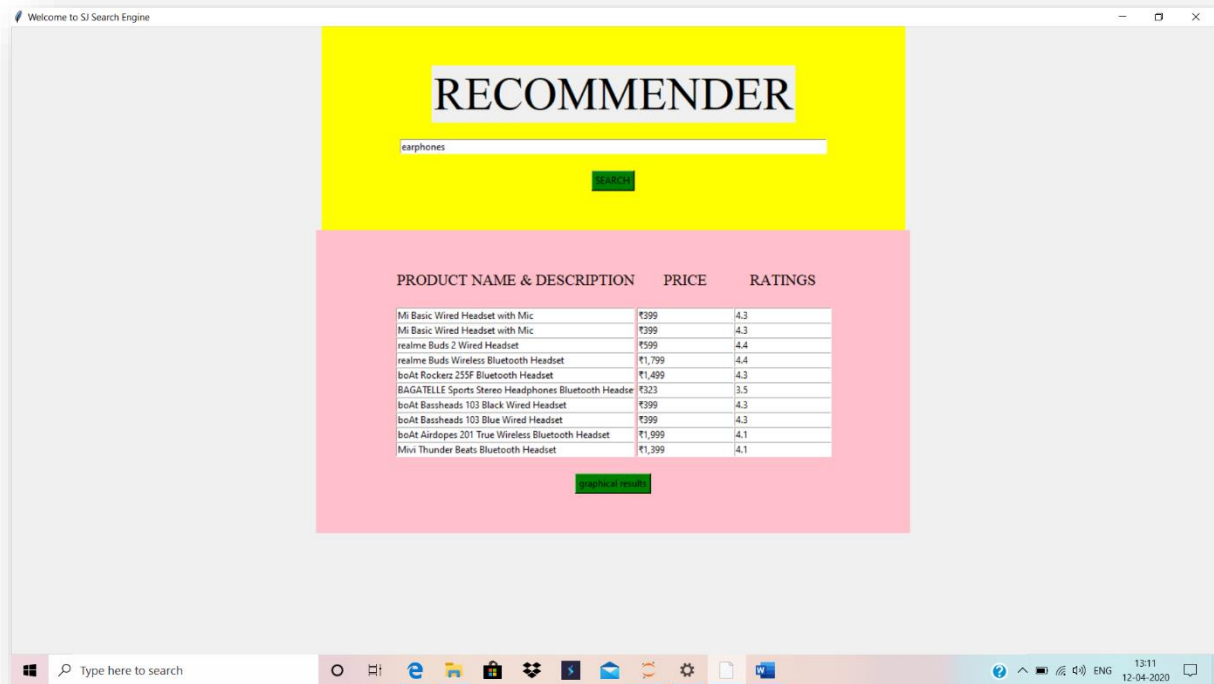
Diksha Sharma: She worked along side Vaishnavi in the backend part. She worked in fetching data and with the libraries to incorporate the details received by the internet in the project. She also helped in fetching the price from the internet of the products. In the report, she extended her valuable help in writing the background and description of the project.

IMPLEMENTATION OF THE SCHEDULED PROJECT

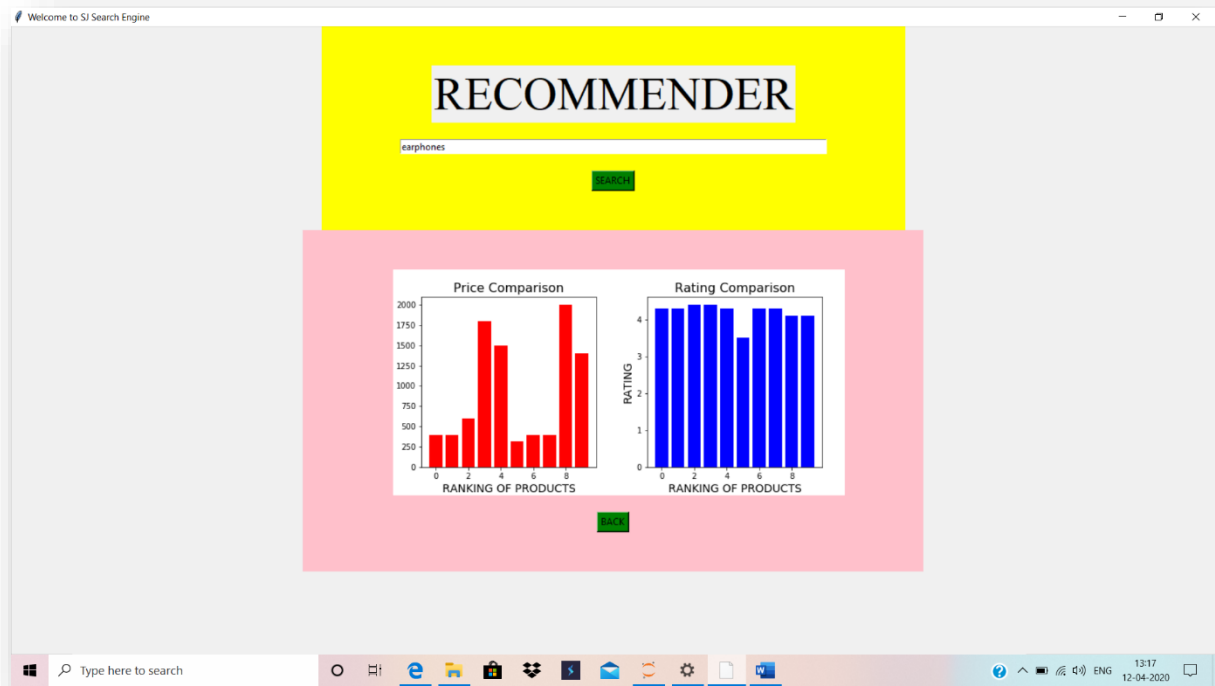
In this section, we present how our project looks like ,when we work on the front end.



When we will run the code , the above box is displayed on the screen. It has a search box where the user can enter whatever he/she wants recommendation for and then click on search.



When the user enters the product example “earphones”, the system generates the results with the prices as well as the ratings of the various options available. There is also an option for viewing the graphical results.



On clicking on the graphical results , there are graphs displayed according to the ratings and the prices compared to the ranking. Here , the user can easily choose the product comparing the price and the rating of the given product. The back option takes us back to the preceding page.

TECHNOLOGIES AND THE FRAMEWORK TO BE USED

BEAUTIFULSOUP:

Beautiful Soup is a library that makes it easy to scrape information from web pages. It sits atop an HTML or XML parser, providing Pythonic idioms for iterating, searching, and modifying the parse tree.

Here , we are pulling out data from ‘Flipkart’ which helps us to recommend products easily.

Code snippet from project :

```
# Transform our code in well understand format
```

```
soup = BeautifulSoup(fhand, 'html.parser')
```

URLLIB:

urllib is a package that collects several modules for working with URLs.

urllib.request is a Python module for fetching URLs (Uniform Resource Locators). It offers a very simple interface, in the form of the *urlopen* function. This is capable of fetching URLs using a variety of different protocols. It also offers a slightly more complex interface for handling common situations - like basic authentication, cookies, proxies and so on. These are provided by objects called handlers and openers.

Code snippet from project :

```
# Fetch data
```

```
fhand=urllib.request.urlopen('https://www.flipkart.com/search?q='+product,context=ctx).read()
```

JSON :

Python has a built-in package called **json**, which can be used to work with JSON data.

JSON is a syntax for storing and exchanging data. JSON is text, written with JavaScript object notation.

Code snippet from project :

```
# Parsing Code into json structure and iterate through each item found
for i,item in enumerate(json.loads(final.text)['itemListElement']):
    print(i+1,item['name'], price[i].text, rate[i].text)
name.append(item['name'])
y.append(price[i].text)
y1.append(rate[i].text)
```

SSL :

SSL stands for **Secure Sockets Layer** and is designed to create secure connection between client and server. Secure means that connection is encrypted and therefore protected from eavesdropping. It also allows to validate server identity.

Code snippet from project :

```
# Ignore SSL Certificate error
ctx = ssl.create_default_context()
ctx.check_hostname = False
ctx.verify_mode = ssl.CERT_NONE
```

REGEX MODULE :

Python has a built-in package called re, which can be used to work with Regular Expressions. A RegEx, or Regular Expression, is a sequence of characters that forms a search pattern.

RegEx can be used to check if a string contains the specified search pattern.

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

Code snippet from project :

```
#GRAPHICAL input  
window = Tk()  
window.title("Welcome to SJ Search Engine")  
window.geometry('1200x800')
```

Python Numpy :

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

It is the fundamental package for scientific computing with Python.

Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data.

MATPLOTLIB :

matplotlib.pyplot is a plotting library used for *visualisation* 2D graphics in python programming language. It can be used in python scripts, shell, web application servers and other graphical user interface toolkits.

One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc.

from matplotlib import pyplot as plt

pylab combines pyplot with numpy into a single namespace. This is convenient for interactive work, but for programming it is recommended that the namespaces be kept separate.

from matplotlib.figure import Figure

The whole figure is regarded as the figure object. It is necessary to explicitly use plt.figure() when we want to tweak the size of the figure and when we want to add multiple Axes objects in a single figure.

Code snippet from project :

```
fig = Figure(figsize=(4,4))
pr = fig.add_subplot(111)
pr.bar(x,y,color='red')
pr.set_title ("Price Comparison", fontsize=16)
pr.set_ylabel("PRICE", fontsize=14)
pr.set_xlabel("RANKING OF PRODUCTS", fontsize=14)
```

from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg

This statement is used to embed a Matplotlib graph to a Tkinter GUI.

Code snippet from project :

```
fig = Figure(figsize=(4,4))
pr = fig.add_subplot(111)
pr.bar(x,y,color='red')
pr.set_title ("Price Comparison", fontsize=16)
pr.set_ylabel("PRICE", fontsize=14)
pr.set_xlabel("RANKING OF PRODUCTS", fontsize=14)
```

```
canvas = FigureCanvasTkAgg(fig, master=frame3)
canvas.get_tk_widget().grid(row=1,column=1)
canvas.draw()
```

SWOT ANALYSIS

STRENGTHS :

- Interactive and easy to use.
- Comparison of ratings graph and price graph can be easily done
- Since we are fetching the data from internet, getting recommendations on each product is seconds work.
- The various libraries makes the code short enough to understand in a go.

WEAKNESS :

- Does not work when the internet is unavailable or slow.
- Whenever it receives a bulk of data , it displays error and asks the user to try again
- When the user types another item to search in the same window , the backend fetches the correct data but the front end does not display it. So one has to close the window and try again.
- The graph is not marking the products and it is a bar graph . So, it becomes difficult to comprehend it when the values are approximately same.

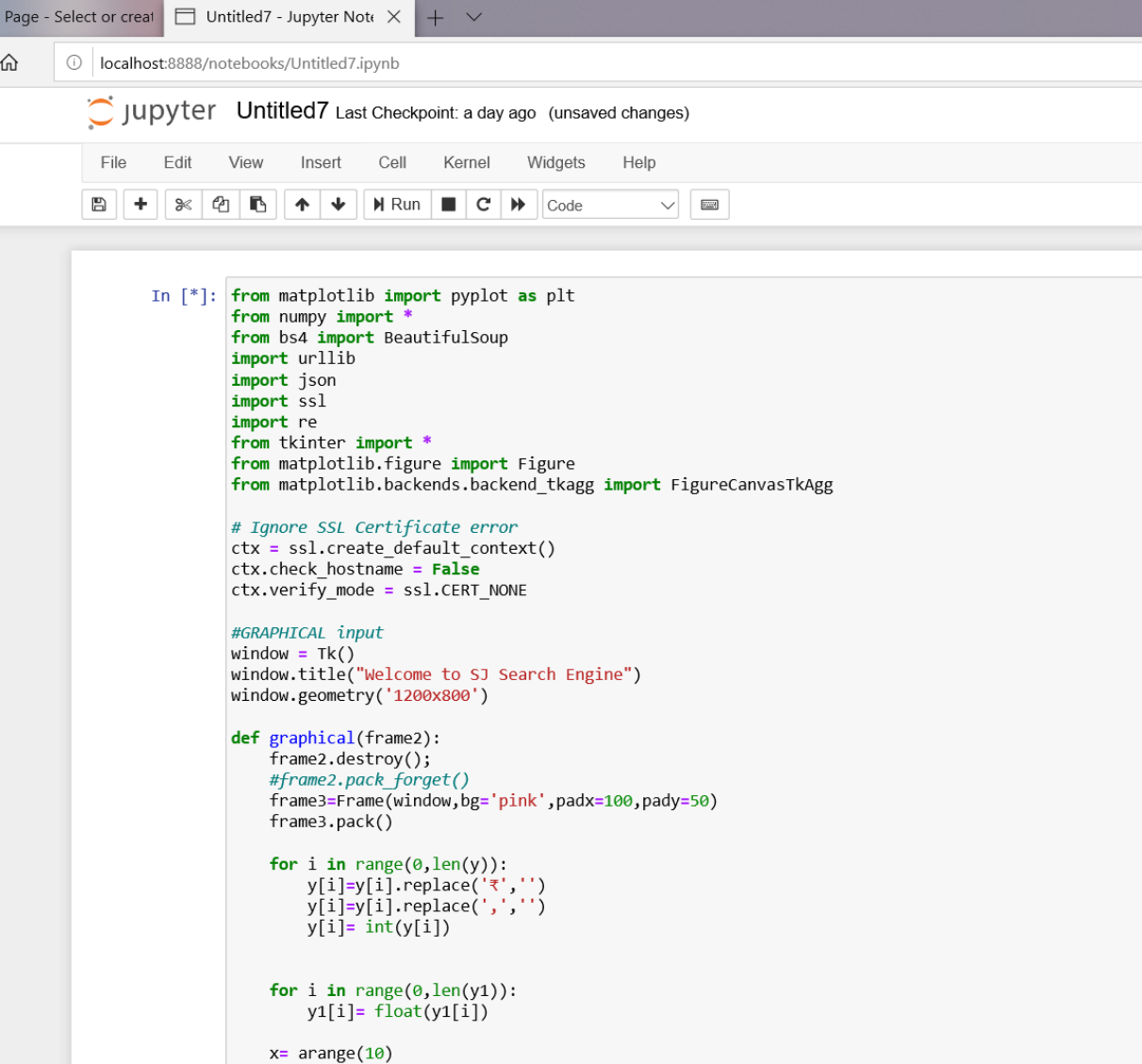
OPPURTUNITIES :

- The recommender system can be varied from person to person depending on his or her choices of various products.
- It can be made more interactives where we can easily handle bulk data and represent it.
- All the above weakness can be removed given some time and effort except the need for an internet connection.

THREAT :

- The main threat is the market where everyday a new system is developed and used. Keeping up to date is not easy.
- The competition is quite high when comes to recommendation system as every company tries to make it the best way they can . So having new approaches and idea every time to update the project is not possible.

CODE



The screenshot shows a Jupyter Notebook window titled "Untitled7 - Jupyter Notebooks". The browser address bar indicates the notebook is running on localhost:8888. The notebook interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for saving, adding cells, undo, redo, and running code. The code cell contains the following Python code:

```
In [*]: from matplotlib import pyplot as plt
        from numpy import *
        from bs4 import BeautifulSoup
        import urllib
        import json
        import ssl
        import re
        from tkinter import *
        from matplotlib.figure import Figure
        from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg

        # Ignore SSL Certificate error
        ctx = ssl.create_default_context()
        ctx.check_hostname = False
        ctx.verify_mode = ssl.CERT_NONE

        #GRAPHICAL input
        window = Tk()
        window.title("Welcome to SJ Search Engine")
        window.geometry('1200x800')

        def graphical(frame2):
            frame2.destroy();
            #frame2.pack_forget()
            frame3=Frame(window,bg='pink',padx=100,pady=50)
            frame3.pack()

            for i in range(0,len(y)):
                y[i]=y[i].replace('₹','')
                y[i]=y[i].replace(',','')
                y[i]=int(y[i])

            for i in range(0,len(y1)):
                y1[i]=float(y1[i])

            x= arange(10)
```



```
fig = Figure(figsize=(4,4))
pr = fig.add_subplot(111)
pr.bar(x,y,color='red')
pr.set_title ("Price Comparison", fontsize=16)
pr.set_ylabel("PRICE", fontsize=14)
pr.set_xlabel("RANKING OF PRODUCTS", fontsize=14)

canvas = FigureCanvasTkAgg(fig, master=frame3)
canvas.get_tk_widget().grid(row=1,column=1)
canvas.draw()

fig2 = Figure(figsize=(4,4))
ra = fig2.add_subplot(111)
ra.bar(x,y1,color='blue')
ra.set_title ("Rating Comparison", fontsize=16)
ra.set_ylabel("RATING", fontsize=14)
ra.set_xlabel("RANKING OF PRODUCTS", fontsize=14)

canvas = FigureCanvasTkAgg(fig2, master=frame3)
canvas.get_tk_widget().grid(row=1,column=2)
canvas.draw()

sp6=Label(frame3,text=" ",bg="pink")
sp6.grid(row=2)

btn3 = Button(frame3, text="BACK",bg="green", command= lambda :details(frame3))
btn3.grid(row=3,columnspan=3)

def details(frame3):
    frame3.destroy()
    frame2=Frame(window,bg='pink',width=120,padx=100,pady=50)
    frame2.pack()

    l1=Label(frame2,text="PRODUCT NAME & DESCRIPTION",font=("Times new roman", 15),bg="pink")
    l2=Label(frame2,text="PRICE",font=("Times new roman", 15),bg="pink")
    l3=Label(frame2,text="RATINGS",font=("Times new roman", 15),bg="pink")

    l1.grid(row=0, column=0)
```

```
File Edit View Insert Cell Kernel Widgets Help
[Icons] [Code]
12.grid(row=0, column=1)
13.grid(row=0, column=2)

sp1=Label(frame2,text=" ",bg="pink")
sp1.grid(row=1)

height = 10
width = 3
x1=[]
for i in range(height): #Rows
    x1.append([])
    for j in range(width): #Columns
        a=StringVar()
        a.set("")
        x1[i].append(a)
        if(j==0):
            b = Entry(frame2,width=50, textvariable=x1[i][j])
            b.grid(row=i+2, column=j)
        else:
            b = Entry(frame2, textvariable=x1[i][j])
            b.grid(row=i+2, column=j)

for j in range(height):
    x1[j][0].set(name[j])

for j in range(height):
    x1[j][1].set(y[j])

for j in range(height):
    x1[j][2].set(y1[j])

sp5=Label(frame2,text=" ",bg="pink")
sp5.grid(row=12)

btn2 = Button(frame2, text="graphical results",bg="green", command= lambda: graphical(frame2))
btn2.grid(row=13, columnspan=3)

def li():

    # Ask Prouct to Search
```

```

# Ask Product to Search
#product = input("Enter Product For Search: ")
product=entry.get()
product.replace(' ','+')
print("product : ",product)

# Try to fetch data
try:
    print("Trying to fetch data...")

    # Fetch data
    fhand = urllib.request.urlopen('https://www.flipkart.com/search?q='+product, context=ctx).read()

    # Transform our code in well understand format
    soup = BeautifulSoup(fhand, 'html.parser')

    #Extract Price
    price=soup.find_all('div', attrs={"class" : "_1vC40E"})
    #Extract Price
    rate=soup.find_all('div', attrs={"class" : "hGSR34"})

    # Extracting the Script
    final=soup.find('script', type="application/ld+json")

    print("Recommended Products on",product.replace(' ',' '))

    # Parsing Code into json structure and iterate through each item found
    for i,item in enumerate(json.loads(final.text)['itemListElement']):
        print(i+1,item['name'], price[i].text, rate[i].text)
        name.append(item['name'])
        y.append(price[i].text)
        y1.append(rate[i].text)

    details(Frame())

except:
    print("Error in fetching the data. Try Again!")
  
```

```

File Edit View Insert Cell Render Widgets Help
[Icons] [Run] [Code]

print("Error in fetching the data. Try Again!")

frame=Frame(window,bg='yellow',width=120,padx=100,pady=50)
frame.pack()

head=Label(frame,text="RECOMMENDER")
head.config(font=("Times new roman", 44))
head.grid(row=1)

sp1=Label(frame,text=" ",bg="yellow")
sp1.grid(row=2)

txt=StringVar()
entry=Entry(frame,width=90,textvariable=txt)
entry.grid(row=3)

sp2=Label(frame,text=" ",bg="yellow")
sp2.grid(row=4)

name=list()
y=list()
y1=list()

btn = Button(frame, text="SEARCH",bg="green", command=li)
btn.grid(row=5)

window.mainloop()

product : EARPHONES
Trying to fetch data...
Recommended Products on EARPHONES
1 Mi Basic Wired Headset with Mic ₹399 4.3
2 Mi Basic Wired Headset with Mic ₹399 4.3
3 realme Buds 2 Wired Headset ₹599 4.4
4 realme Buds Wireless Bluetooth Headset ₹1,799 4.4

```

```
txt=StringVar()
entry=Entry(frame,width=90,textvariable=txt)
entry.grid(row=3)

sp2=Label(frame,text=" ",bg="yellow")
sp2.grid(row=4)

name=list()
y=list()
y1=list()

btn = Button(frame, text="SEARCH",bg="green", command=li)
btn.grid(row=5)

window.mainloop()

product :  EARPONES
Trying to fetch data...
Recommended Products on EARPONES
1 Mi Basic Wired Headset with Mic ₹399 4.3
2 Mi Basic Wired Headset with Mic ₹399 4.3
3 realme Buds 2 Wired Headset ₹599 4.4
4 realme Buds Wireless Bluetooth Headset ₹1,799 4.4
5 boAt Rockerz 255F Bluetooth Headset ₹1,499 4.3
6 BAGATELLE Sports Stereo Headphones Bluetooth Headset ₹323 3.5
7 boAt Bassheads 103 Black Wired Headset ₹399 4.3
8 boAt Bassheads 103 Blue Wired Headset ₹399 4.3
9 boAt Airdopes 201 True Wireless Bluetooth Headset ₹1,999 4.1
10 Mivi Thunder Beats Bluetooth Headset ₹1,399 4.1
```

In []:

OUTPUT

