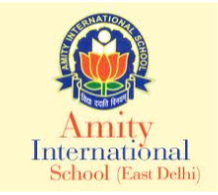
**AMITY INTERNATIONAL SCHOOL, MAYUR VIHAR**

***COMPUTER SCIENCE PROJECT***

Amazon vs Flipkart: Wishlist





**By:**

**Class:**

**Roll No.**

**Acknowledgement**

*I acknowledge the valuable contribution of Mrs.\_\_\_\_\_\_\_\_\_\_\_ in providing me the proper guidance to complete the computer science project. The project would not have been completed without her support and kind help.*

(Name of student)

**CERTIFICATE**

AMITY INTERNATIONAL SCHOOL,

MAYUR VIHAR

This is to certify that \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Roll number:

of Class XII \_\_\_ has successfully completed

the CBSE class XII Computer Science Project:

“Amazon vs Flipkart: Wishlist” under my supervision, according to the guidelines laid down by CBSE.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ms.\_\_\_\_\_

**Index**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Title** | **Page No.** |
| 1 | Introduction | 5 |
| 2 | Modules Used | 6 |
| 3 | Source Code | 7-19 |
| 4 | Output Screens | 20-23 |
| 5 | Future Expansion | 24 |
| 6 | Bibliography | 25 |

**Introduction**

Since Amazon and Flipkart are one of the leading online e-commerce websites, customers are often left to choose between the two. This program, Amazon vs Flipkart: Wishlist compares prices of the same product across the two sites and returns the name of the site which sells the product at a lower cost, thus helping the customer get a better deal.

This program creates a database in MySQL. For every product that the customer searches, a new row is added to the table. The columns in the table show the respective Amazon and Flipkart cost of the product. This database is visible to the customer as the “Wishlist” in the window created using Tkinter. The program then compares the two prices and returns the name and web link of the site selling the product at a lower cost.

User must enter the name of the product along with its appropriate specifications to get a better result.

This project has been written in Python. This project also makes use of Python and MySQL connection. I have used most of the concepts taught over the past two years to successfully bring this project to completion.

**Modules Used**

**BeautifulSoup4:** It is a library that makes it easy to scrape information from web pages and pulling data out of HTML and XML files. It is used for web scraping and parsing.

**Tkinter:** It is Python's standard GUI (Graphical User Interface) package. It allows programmers to create a window and place desirable widgets such as Label, Button, Radiobutton, Listbox, Textbox, OptionMenu, etc in it.

**PyMySQL:** It is a Python library which is used to connect with a MySQL database.

**Urllib:** It is a package which is a collection of several modules like request, error, parse, etc used for working with URLs. urllib.request is primarily used for opening and fetching URLs.

**PIL or Python Image Library:** It is a [free](https://en.wikipedia.org/wiki/Free_and_open_source_software) [library](https://en.wikipedia.org/wiki/Library_(computing)) for the [Python programming language](https://en.wikipedia.org/wiki/Python_(programming_language)) that adds support for opening, manipulating, and saving many different image file formats like PNG,JPEG,GIF,PPM,BMP,TIFF. It can be used for resizing images, masking and transparency handling, image filtering, image enhancing, adding text to images, etc.

**PrettyTable:** It is a Python library for generating simple ASCII tables. It is used to control many aspects of a table, such as the width of the column, padding, the alignment of text, table border and to sort data. PrettyTable can read data from MySQL and output data in ASCII or HTML.

**Textwrap:** It can be used for wrapping and formatting plain text in the output. This module provides formatting of text by adjusting the line breaks in the input paragraph. It is used for making the text output of a code more presentable.

**Source Code**

1. Code for Web-Scraping

# -\*- coding: utf-8 -\*-

"""

Created on Fri Dec 13 16:45:28 2019

@author: Administrator

"""

import bs4 as bs

import urllib.request as urlr

class WebScraper():

alink='https://www.amazon.in/s?k='

flink='https://www.flipkart.com/search?q='

def \_\_init\_\_(self):

pass

def inputtoqry(self,exp):

srchqry=''

wordlist=exp.split()

for word in wordlist:

srchqry+=str(word)+'+'

return srchqry

def search(self,qstr,found,prodnames,prodprices):

def wordtosrch(wrdlist):

def strtolower(st1):

resultstr=''

for char in st1:

if char.isalnum():

resultstr+=char.lower()

else:

continue

return resultstr

wrdlist2=[]

for i in range(len(wrdlist)):

wrdlist2.append(strtolower(wrdlist[i]))

return wrdlist2

if found==0:

return ['Not Available',0]

elif found==1:

wordlist=qstr.split()

wdlst= wordtosrch(wordlist)

matchlist=[]

matchplist=[]

matchind=[]

numberofmatch=[]

cnt=0

for name in prodnames:

nmchk=name.split()

nmlst=wordtosrch(nmchk)

nmstr=''

for elem in nmlst:

nmstr+=elem

for word in wdlst:

if word in nmstr:

cnt+=1

nameind=prodnames.index(name)

if wdlst.index(word)==len(wdlst)-1:

if cnt==len(wdlst):

matchind.append(nameind)

numberofmatch.append(cnt)

for i in range (len(numberofmatch)):

if numberofmatch[i]==len(wdlst):

reqind=matchind[i]

price=prodprices[reqind]

matchlist.append(prodnames[reqind])

matchplist.append(price)

nmstr=''

cnt=0

else:

cnt=0

nmstr=''

if matchlist == [] or matchplist== []:

return [qstr+' is Not Available','0']

else:

resultset=[matchlist[0],matchplist[0]]

return resultset

print()

def flipkart\_search(self,query):

srchurl=self.flink+self.inputtoqry(query)

hdr = { 'User-Agent' : 'Chrome (Windows NT 6.1; Win64; x64)' }

req=urlr.Request(srchurl,headers=hdr)

sauce=urlr.urlopen(req).read()

soup=bs.BeautifulSoup(sauce,'lxml')

body=soup.body

fnamelist=[]

fpricelist=[]

for para in body.find\_all('div', class\_ = "\_3O0U0u"):

if para is None:

fmatch=0

continue

elif para.find\_all('div', style = 'width:25%'):

reslt=para.find\_all('div', style ='width:25%')

for child in reslt:

d=child.find('a', class\_ = "\_2cLu-l")

product\_name=(d.contents)[0]

fnamelist.append(product\_name)

e=child.find('div', class\_ = "\_1vC4OE")

product\_price=(e.contents)[0]

fpricelist.append(product\_price)

fmatch=1

else:

b=para.find('div', class\_ = "\_3wU53n")

name=list(b.contents)[0]

fnamelist.append(name)

i=0

for parent in para.parents:

if i==0:

if parent is None:

print(parent)

elif parent.find('div', class\_ ="col col-5-12 \_2o7WAb"):

f=parent.find('div', class\_ = "\_1vC4OE \_2rQ-NK")

price=list(f.contents)[0]

fpricelist.append(price)

i=1

fmatch=1

return(self.search(query,fmatch,fnamelist,fpricelist))

def amazon\_search(self,query):

srchurl=self.alink+self.inputtoqry(query)

hdr = { 'User-Agent' : 'Chrome (Windows NT 6.1; Win64; x64)' }

req=urlr.Request(srchurl,headers=hdr)

sauce=urlr.urlopen(req).read()

soup=bs.BeautifulSoup(sauce,'lxml')

body=soup.body

anamelist=[]

apricelist=[]

match=0

for para in body.find\_all('span', class\_ ="a-size-medium a-color-base a-text-normal"):

if para is None:

match=0

else:

name=list(para.contents)[0]

anamelist.append(name)

i=0

for parent in para.parents:

if i==0:

if parent is None:

print(parent)

elif parent.find('span', class\_ ='a-price-whole'):

b=parent.find('span', class\_ ='a-price-whole')

price=list(b.contents)[0]

apricelist.append(price)

i=1

match=1

return(self.search(query,match,anamelist,apricelist))

1. Code for GUI

import tkinter as tk

from tkinter import messagebox

from PIL import ImageTk,Image

import Project\_Functions as funcs

import Project\_WebScraper\_search as flip

import webbrowser as browser

web=funcs.Functions()

scraper=flip.WebScraper()

font1=("Helvetica",36)

font2=("Helvetica",12)

font3=("Helvetica",18)

font4=("Helvetica",26)

class Parent\_Project(tk.Tk):

def \_\_init\_\_(self):

tk.Tk.\_\_init\_\_(self)

container=tk.Frame(self)

container.pack(side="top",fill="both",expand=True)

container.grid\_rowconfigure(0,weight=1)

container.grid\_columnconfigure(0,weight=1)

self.title("Amazon Vs Flipkart Wishlist: Project")

self.frames={}

for f in [Welcome\_Page,Search\_Page,Wishlist\_Page]:

frame=f(container,self)

self.frames[f]=frame

frame.grid(row=0,column=0,sticky="nsew")

self.show\_frame(Welcome\_Page)

def show\_frame(self,cont):

frame=self.frames[cont]

frame.tkraise()

class Welcome\_Page(tk.Frame):

def \_\_init\_\_(self,parent,controller):

tk.Frame.\_\_init\_\_(self,parent)

self.grid\_columnconfigure(0)

self.grid\_rowconfigure(0)

#Background Color

self.configure(background="#ede59a")

#Title

title=tk.Label(self,text="Amazon Vs Flipkart: Wishlist",font=font1,bg="#fddb3a")

title.pack(pady=8)

#Instructions

instructions\_0=tk.Label(self,text='Welcome To The Wishlist:',font=font3,bg="#ede59a")

instructions\_0.pack(side='top',padx=10,anchor='nw')

instructions\_1=tk.Label(self,text="Instructions:",font=font3,bg="#ede59a")

instructions\_1.pack(side='top',padx=10,anchor='nw')

instructions\_2=tk.Label(self,text='1.You can compare prices of your desired products on Flipkart and Amazon',font=font3,bg="#ede59a")

instructions\_2.pack(side='top',padx=100,anchor='nw')

#Instruction to search

instructions\_3=tk.Label(self,text='2.Click search to go to Search Page',font=font3,bg="#ede59a")

instructions\_3.pack(side='top',padx=100,anchor='nw')

#Instruction to view

instructions\_3=tk.Label(self,text='3.Click view to see current wishlist',font=font3,bg="#ede59a")

instructions\_3.pack(side='top',padx=100,anchor='nw')

#Image

img\_avf=ImageTk.PhotoImage(Image.open(r'C:\Users\SCL1\Desktop\CS Project\pics\amazon\_vs\_flipkart.png'))

imglbl=tk.Label(self,image=img\_avf)

imglbl.image=img\_avf

imglbl.pack(side='top',anchor='center',pady=20)

test=tk.Frame(self,bg='#ede59a')

test.pack(side="bottom",fill='none')

#Button to search page

img\_search=ImageTk.PhotoImage(Image.open(r'C:\Users\SCL1\Desktop\CS Project\pics\search.png'))

def gotosearch():

global b1

global b2

b1=tk.Button()

b2=tk.Button()

controller.show\_frame(Search\_Page)

button1=tk.Button(test,text='Search',image=img\_search,command=lambda: gotosearch())

button1.image=img\_search

button1.pack(side="right",anchor="se",padx=10,pady=10)

#Button to view page

img\_view=ImageTk.PhotoImage(Image.open(r'C:\Users\SCL1\Desktop\CS Project\pics\view.jpg'))

button2=tk.Button(test,text='View',image=img\_view,command=lambda: controller.show\_frame(Wishlist\_Page))

button2.image=img\_view

button2.pack(side="left",anchor="sw",padx=10,pady=10)

class Search\_Page(tk.Frame):

def \_\_init\_\_(self,parent,controller):

tk.Frame.\_\_init\_\_(self,parent)

self.grid\_columnconfigure(0)

self.grid\_rowconfigure(0)

#Background Color

self.configure(background="#ede59a")

#Title

title=tk.Label(self,text="Amazon Vs Flipkart: Wishlist",font=font1,bg="#fddb3a")

title.pack(pady=8)

title1=tk.Label(self,text="Search",font=font4,bg="#AEE100")

title1.pack(pady=8)

#Frame to contain content

f2=tk.Frame(self,bg='#ede59a')

f2.pack(side='top')

#Name of product input

name\_label=tk.Label(f2,text='Enter the name of the product: ',font=font3,bg='#ede59a')

name\_label.grid(row=0,column=0,sticky='w')

input\_box1=tk.Entry(f2, width=30,bg='#fddb3a',font=font3)

input\_box1.grid(row=0,column=1,padx=3)

#Specification input

spec\_label=tk.Label(f2,text="Enter the specification: ",font=font3,bg='#ede59a')

spec\_label.grid(row=1,column=0,sticky='w')

input\_box2=tk.Entry(f2, width=30,bg='#fddb3a',font=font3)

input\_box2.grid(row=1,column=1,padx=3,pady=5)

#Search Button

img\_search=ImageTk.PhotoImage(Image.open(r'C:\Users\SCL1\Desktop\CS Project\pics\search.png'))

button1=tk.Button(f2,text='Search',image=img\_search,command=lambda: execute\_all())

button1.image=img\_search

button1.grid(row=2,column=1,pady=10)

#Spacing

l1=tk.Label(f2,text='',bg="#ede59a",height=2)

l1.grid(row=3,columnspan=2)

l2=tk.Label(f2,text='',bg="#ede59a",height=2)

l2.grid(row=4,columnspan=2)

#Functionality

'''Variables'''

best\_site=''

best\_price=''

def execute\_all():

global best\_site

global best\_price

result\_flip=scraper.flipkart\_search(input\_box1.get()+' '+input\_box2.get())

result\_amz=scraper.amazon\_search(input\_box1.get()+' '+input\_box2.get())

best\_site=web.get\_best\_site(web.string\_to\_int(result\_flip[1]),web.string\_to\_int(result\_amz[1]))

best\_price=web.get\_best\_price(web.string\_to\_int(result\_flip[1]),web.string\_to\_int(result\_amz[1]))

best\_price=chr(8377)+best\_price

web.add\_new(result\_flip,result\_amz)

messagebox.showinfo('Result','Added to the database:\nFlipkart\nName "{}" | Price: "{}"\nAmazon\nName "{}" | Price: "{}"'.format(result\_flip[0],result\_flip[1],result\_amz[0],result\_amz[1]))

post\_site()

post\_price()

link()

#Output Frame

f3=tk.Frame(self,bg='#ede59a')

f3.pack(side='top')

#Outputs

img\_site=ImageTk.PhotoImage(Image.open(r'C:\Users\SCL1\Desktop\CS Project\pics\site.png'))

deal\_label=tk.Label(f3,text="You will get better deal on: ",image=img\_site,compound='right',bg='#ede59a',font=font3)

deal\_label.image=img\_site

deal\_label.grid(row=0,column=0,pady=10)

img\_price=ImageTk.PhotoImage(Image.open(r'C:\Users\SCL1\Desktop\CS Project\pics\price.png'))

price\_label=tk.Label(f3,text="Price : ",image=img\_price,compound='right',bg='#ede59a',font=font3)

price\_label.image=img\_price

price\_label.grid(row=0,column=2,padx=5,pady=10,sticky='w')

outsite=tk.Text(f3,bg='#fddb3a',width=20,height=1,font=font2)

outsite.grid(row=0,column=1,padx=5,pady=10,sticky='w')

outprice=tk.Text(f3,bg='#fddb3a',width=20,height=1,font=font2)

outprice.grid(row=0,column=3,padx=5,pady=10)

def post\_site():

global best\_site

outsite.delete(0.0,'end')

outsite.insert(0.0,best\_site)

def post\_price():

global best\_price

outprice.delete(0.0,'end')

outprice.insert(0.0,best\_price)

#Link

def link():

global best\_site

global b1

global b2

global best\_price

if best\_site=='Flipkart':

bestsite\_link=scraper.inputtoqry(scraper.flink+input\_box1.get()+' '+input\_box2.get())

b1=tk.Button(f3,bg="#ede59a",text="Click to visit Flipkart link",font=font2,command=lambda: browser.open(bestsite\_link),cursor='hand2')

b1.grid(row=2,column=1,sticky='w',pady=5)

elif best\_site=="Amazon":

bestsite\_link=scraper.inputtoqry(scraper.alink+input\_box1.get()+' '+input\_box2.get())

b2=tk.Button(f3,bg="#ede59a",text='Click to visit Amazon link',font=font2,command=lambda: browser.open(bestsite\_link),cursor='hand2')

b2.grid(row=2,column=1,sticky='w',pady=5)

elif best\_site=='Either':

if web.string\_to\_int(best\_price) != 0:

bestsite\_link\_a=scraper.inputtoqry(scraper.alink+input\_box1.get()+' '+input\_box2.get())

bestsite\_link\_f=scraper.inputtoqry(scraper.flink+input\_box1.get()+' '+input\_box2.get())

b1=tk.Button(f3,bg="#ede59a",text='Click to visit Amazon link',font=font2,command=lambda: browser.open(bestsite\_link\_a),cursor='hand2')

b1.grid(row=2,column=1,sticky='w',pady=5)

b2=tk.Button(f3,bg="#ede59a",text='Click to visit Flipkart link',font=font2,command=lambda: browser.open(bestsite\_link\_f),cursor='hand2')

b2.grid(row=3,column=1,sticky='w',pady=5)

else:

pass

#Back Button

test=tk.Frame(self,bg='#ede59a')

test.pack(side="bottom",fill='none')

def back():

global best\_site

global best\_price

global b1

global b2

outsite.delete(0.0,'end')

outprice.delete(0.0,'end')

input\_box1.delete(0,'end')

input\_box2.delete(0,'end')

best\_site=''

best\_price=''

b1.destroy()

b2.destroy()

controller.show\_frame(Welcome\_Page)

img\_back=ImageTk.PhotoImage(Image.open(r'C:\Users\SCL1\Desktop\CS Project\pics\back.jpg'))

button2=tk.Button(test,text='Back',image=img\_back,command=lambda: back())

button2.image=img\_back

button2.pack(side="right",anchor="se",padx=10,pady=10)

class Wishlist\_Page(tk.Frame):

def \_\_init\_\_(self,parent,controller):

tk.Frame.\_\_init\_\_(self,parent)

self.grid\_columnconfigure(0)

self.grid\_rowconfigure(0)

#Background Color

self.configure(background="#ede59a")

#Title

title=tk.Label(self,text="Amazon Vs Flipkart: Wishlist",font=font1,bg="#fddb3a")

title.pack(pady=8)

title1=tk.Label(self,text="Wishlist",font=font4,bg="#AEE100")

title1.pack(pady=8)

#Frame to contain output

f1=tk.Frame(self,bg='#ede59a')

f1.pack(side='top')

#Output Box

output\_box=tk.Text(f1,bg="#d5c455",width=86)

output\_box.grid(row=0,column=0)

def post\_output():

output\_box.delete(0.0,'end')

output\_box.insert(0.0,web.view())

post\_output()

#Scrollbar

sc=tk.Scrollbar(f1)

sc.grid(row=0,column=1,sticky='ns')

output\_box.config(yscrollcommand=sc.set)

sc.config(command=output\_box.yview)

#Back Button

test=tk.Frame(self,bg='#ede59a')

test.pack(side="bottom",fill='none',anchor='e')

img=ImageTk.PhotoImage(Image.open(r'C:\Users\SCL1\Desktop\CS Project\pics\back.jpg'))

button1=tk.Button(test,text='Back',image=img,command=lambda: controller.show\_frame(Welcome\_Page))

button1.image=img

button1.pack(side="right",anchor="se",padx=10,pady=10)

#Update Button

img\_update=ImageTk.PhotoImage(Image.open(r'C:\Users\SCL1\Desktop\CS Project\pics\up.png'))

button2=tk.Button(test,text='Update List',image=img\_update,command=lambda: post\_output())

button2.image=img\_update

button2.pack(side='left',anchor='sw',padx=10,pady=10)

gui=Parent\_Project()

gui.mainloop()

1. Code for MySQL Connection

import pymysql as sql

import prettytable

from textwrap import fill

mycon=sql.connect(host='localhost',user='root',password='amity',database='Project')

cursor=mycon.cursor()

class Functions():

def string\_to\_int(self,price):

new\_int=''

nums='0123456789'

for i in price:

if i in nums:

new\_int+=i

return int(new\_int)

def add\_new(self,fliplist,amzlist):

addlist=[fliplist[0],self.string\_to\_int(fliplist[1]),amzlist[0],self.string\_to\_int(amzlist[1])]

insert\_string='insert into newtable values("{}",{},"{}",{})'.format(addlist[0],addlist[1],addlist[2],addlist[3])

cursor.execute(insert\_string)

mycon.commit()

def get\_best\_site(self,price\_flip,price\_amz):

if price\_flip==0 and price\_amz !=0:

return "Amazon"

elif price\_amz==0 and price\_flip!=0:

return "Flipkart"

elif price\_flip>price\_amz:

return "Amazon"

elif price\_flip<price\_amz:

return "Flipkart"

else:

return "Either"

def get\_best\_price(self,price\_flip,price\_amz):

if price\_flip==0:

return str(price\_amz)

elif price\_amz==0:

return str(price\_flip)

elif price\_flip>price\_amz:

return str(price\_amz)

else:

return str(price\_flip)

def view(self):

cursor.execute("select \* from newtable")

data=cursor.fetchall()

hdr=["S.No",'Flipkart\_Name',"Flipkart\_Price",'Amazon\_Name',"Amazon\_Price"]

table=prettytable.PrettyTable(hdr)

count=0

for i in data:

count+=1

table.add\_row([count,fill(i[0],width=20),chr(8377)+str(i[1]),fill(i[2],width=20),chr(8377)+str(i[3])])

table.add\_row(['','','','',''])

return table.get\_string()

**Output Screens**

1. Welcome Page



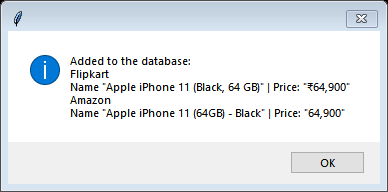
1. Search Page



1. Search Page with Input Query 1 and final output



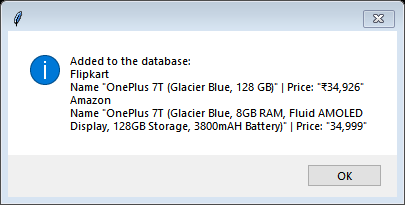
1. Message Box alerting the user about addition of entry in the database



1. Search Page with Input Query 2 and final result



1. Message Box alerting the user about addition of entry in the database



1. Wishlist Page



**Future Expansion**

* After inspecting the codes of some other e-commerce websites, this program could be modified to compare costs of products over a large number of e-commerce websites.
* In case a particular product is unavailable, the program could suggest similar available products.
* The program could be modified to allow users to delete particular entries in their respective “Wishlist”.
* The appearance of the program could be improved by adding colour to the text, adding more images to the background, etc.
* The program could be made available on the internet for everybody to use.

**Bibliography**

* Computer Science with Python for class 11 by Sumita Arora
* Computer Science with Python for class 12 by Sumita Arora
* Github (https://www.github.com)
* Stack exchange (https://www.stackexchange.com)
* Stack overflow (https://www.stackoverflow.com)
* Wikipedia (https://www.wikipedia.org)
* YouTube (https://www.youtube.com)