**PRICE COMPARISION WEBSITE**

INT 213

PYTHON PROJECT



SUBMITTED BY: ROLL NO

ANSHUL KUMAR SINGH 11814863 38

SHUBHAM VERMA 11814867 39

HARSH JAISWAL 11808377 40

SUBMITTED TO: MS. UPINDER KAUR

**INTRODUCTION**

In this project an attempt will be made to make Price Comparison Website using graphical user interface and a combination of basic concepts of Python i.e. - *Dictionary, GUI, List etc.*

In the contemporary arena, Internet has become an inseparable and indispensable part of an average household. People are extensively using Internet to gather information about products and services they are willing to purchase or pay for.

People tend to spend hours together to browse through a whole host of websites to gather particulars about a product, before going in for its purchase.The search down various websites especially holds true for product and / or service being purchased for the very first time, or being technological in nature needs updating of general knowledge, or being a rare item that has to be first found available and then compared as regards its prices on offer.The modern Internet savvy users prefer Internet to make purchases rather than plod across offline stores.

It is simply a matter of convenience and a feasible alternative to utter lack of time faced by them.

Price Comparison websites are also known as "Aggregators" because they aggregate information on a product from many websites offering online shopping of the product. These sites render you easy and lucid comparisons of the prices on offer of a product under consideration, without any toiling involved, on a single screen in a matter of a few clicks of the mouse! .These sites manage to save a substantial amount of effort, time and money of the prospective shopper, and above all, take away the headache of compiling information.

### OBJECTIVES

### 1- Briefly, they can save you time and money.

### 2- Instead of going from store to store, hunting in the Yellow Pages or even bringing up individual merchant sites on a search engine, you can visit a comparison shopping site and view even more information than these other methods would have yielded in a matter of minutes.

### 3- To save money is the primary reason most shoppers rely on the cost-comparison sites.

### 4- The sites offer an alternative to buying the first product you see at the first price you see.

### 5- Price comparison sites can collect data directly from merchants.

### 6-  Retailers who want to list their products on the website then supply their own lists of products and prices, and these are matched against the original database.

### FUTURE SCOPE

Considering the fact that *Indian Consumers* are very much price sensitive and hence price plays a very important role in Indian economy.

Since the number of ecommerce websites in India is quite a big in number, customers do need a platform where they can compare the prices of products over different websites and chose the cheapest one with good customer service.

The future for an e-commerce business is the integration of online stores with price comparison websites and companies sending and delivering goods around the world.

Ceneo is currently the largest and most popular price comparison engine on the Polish market.

I believe that Price comparison websites have a bright future because users want to save time and money by listing their products and chose their best price.

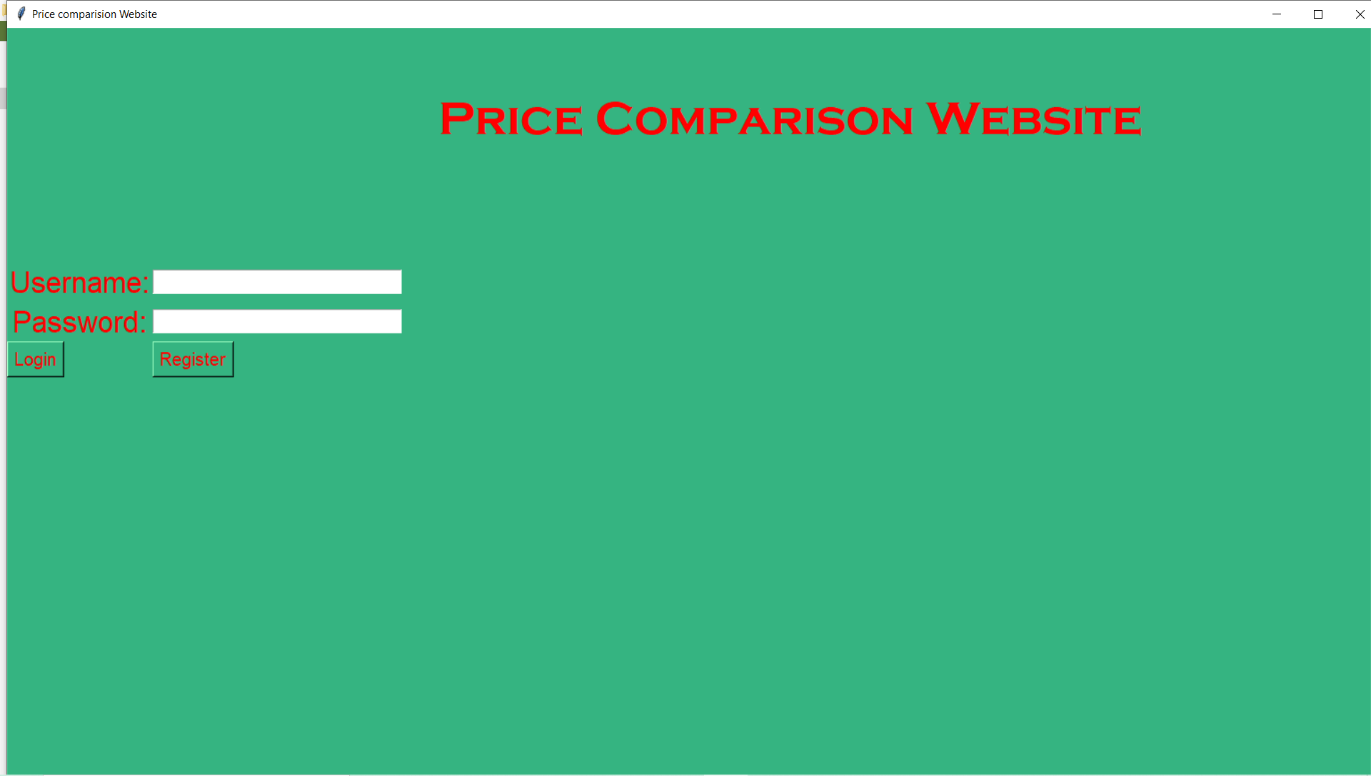
Price comparison websites are very important to every shopping sites because it sale their products at the low price compared to all other sites and give the best price to the buyer.

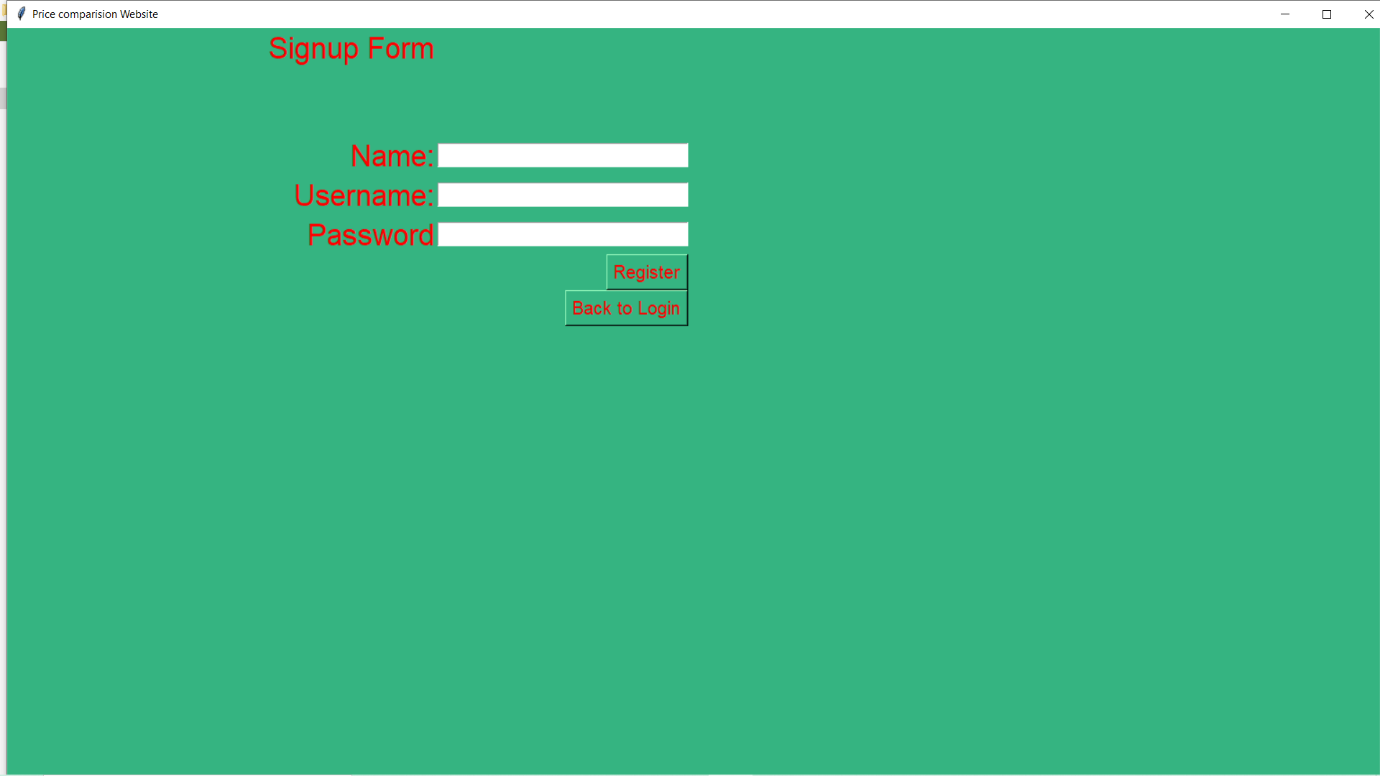
**DATA FLOW DIAGRAM**

PORTAL DATABASE

USER

SCREEN SHOTS:









**WORKING**

Consumers use cost-comparison sites for two reasons: either to locate the lowest possible price on a specific item from a merchant with a good reputation, or to access information and reviews about a range of products in order to make a selection.

Pricing is the focus of most current shopping engines, although some small start-up companies are specializing in presenting product information to help buyers choose the right product, according to the SmartMoney website.

Even if the item is not purchased online, the consumer gains some perspective on whether the prices displayed in stores are good ones, factoring in what shipping costs might have been.

**CODE:**

from tkinter import \*

from tkinter import ttk

import os

import pathlib #for making a new dir

class comparison():

root = Tk()

root.title("Price comparision Website")

root.geometry('2024x2024')

def home(self):

self.fg\_color='red'

self.bk\_color='#35B481'

self.root.configure(background=self.bk\_color)

self.label1=Label(self.root,text="\t Price Comparison Website\n",font=("Copperplate Gothic Bold",40),fg=self.fg\_color,bg=self.bk\_color)

self.label1.grid(row=1,column=1)

self.dummy=Label(self.root,text="",bg=self.bk\_color)

self.dummy.grid(row=1,columnspan=3,sticky=E+W+N+S,pady=120)

self.label2=Label(self.root,text="Username:",font=("",25),fg=self.fg\_color,bg=self.bk\_color)

self.label2.grid(row=2,column=0,sticky=E+W+N+S,padx=0)

self.entry1=Entry(self.root,font=("",15),width=25)

self.entry1.grid(row=2,column=1,sticky=W,padx=0)

self.label3=Label(self.root,text="Password:",font=("",25),fg=self.fg\_color,bg=self.bk\_color)

self.label3.grid(row=3,column=0,sticky=E+W+N+S)

self.entry2=Entry(self.root,font=("",15),width=25)

self.entry2.grid(row=3,column=1,sticky=W)

self.button1=Button(self.root,text="Login",font=("",15),fg=self.fg\_color,bg=self.bk\_color,command=self.login)

self.button1.grid(row=4,column=0,columnspan=2,sticky=W)

self.button2=Button(self.root,text="Register",font=("",15),fg=self.fg\_color,bg=self.bk\_color,command=self.Register)

self.button2.grid(row=4,column=1,columnspan=2,sticky=W)

self.label4=Label(self.root,font=("",25),bg=self.bk\_color)

self.label4.grid(row=6,column=0,sticky=W+E)

self.root.mainloop()

def destroy\_home(self):

self.label1.destroy()

self.label2.destroy()

self.label3.destroy()

self.entry1.destroy()

self.entry2.destroy()

self.button1.destroy()

self.button2.destroy()

self.label4.destroy()

self.dummy.destroy()

def Register(self):

self.destroy\_home()

self.rlabel1=Label(self.root,text="\t\tSignup Form\n\n",font=("",25),fg=self.fg\_color,bg=self.bk\_color)

self.rlabel1.grid(row=0,column=0,sticky=E+W)

self.rname=Label(self.root,text="Name:",font=("",25),fg=self.fg\_color,bg=self.bk\_color)

self.rname.grid(row=1,column=0,sticky=E)

self.rname\_entry=Entry(self.root,font=("",15),width=25)

self.rname\_entry.grid(row=1,column=1,sticky=W)

self.rlabel2=Label(self.root,text="Username:",font=("",25),fg=self.fg\_color,bg=self.bk\_color)

self.rlabel2.grid(row=2,column=0,sticky=E)

self.rentry1=Entry(self.root,font=("",15),width=25)

self.rentry1.grid(row=2,column=1,sticky=W)

self.rlabel3=Label(self.root,text="Password",font=("",25),fg=self.fg\_color,bg=self.bk\_color)

self.rlabel3.grid(row=3,column=0,sticky=E)

self.rentry2=Entry(self.root,font=("",15),width=25)

self.rentry2.grid(row=3,column=1,sticky=W)

self.rbutton1=Button(self.root,text="Register",font=("",15),fg=self.fg\_color,bg=self.bk\_color,command=self.register\_sucess)

self.rbutton1.grid(row=4,columnspan=2,sticky=E)

self.rbutton2=Button(self.root,text="Back to Login",font=("",15),fg=self.fg\_color,bg=self.bk\_color,command=self.destroy\_signup)

self.rbutton2.grid(row=5,columnspan=2,sticky=E)

self.rsucess=Label(self.root,font=("calibri",30),bg=self.bk\_color)

self.rsucess.grid(row=6,columnspan=3,sticky=E+W)

def destroy\_signup(self):

self.rlabel1.destroy()

self.rlabel2.destroy()

self.rlabel3.destroy()

self.rentry1.destroy()

self.rentry2.destroy()

self.rname.destroy()

self.rbutton1.destroy()

self.rbutton2.destroy()

self.rsucess.destroy()

self.rname\_entry.destroy()

self.home()

def register\_sucess(self):

username\_info=self.rentry1.get()

password\_info=self.rentry2.get()

if(username\_info ==''):

self.rsucess.config(text="Invalid Username")

elif(password\_info ==''):

self.rsucess.config(text="Invalid Password")

else:

fname=username\_info+".txt"

file=open(fname,"w+")

file.write(username\_info+"\n")

file.write(password\_info)

file.close()

self.rentry1.delete(0,END)

self.rentry2.delete(0,END)

self.rname\_entry.delete(0,END)

self.rsucess.config(text="Register Sucess")

def login(self):

username1=self.entry1.get()

password1=self.entry2.get()

fname=username1+".txt"

self.entry1.delete(0,END)

self.entry2.delete(0,END)

list\_of\_files=os.listdir()

if(len(username1)==0):

self.label4.config(text="Enter Credential")

elif(len(username1)==0):

self.label4.config(text="Enter Credential")

elif fname in list\_of\_files:

file1=open(fname,"r")

verify=file1.read().splitlines()

if password1 in verify:

self.front()

else:

self.label4.config(text="Wrong Password")

else:

self.label4.config(text="User not found")

def front(self):

self.destroy\_home()

self.flabel=Label(self.root,text="\t\tWELCOME TO PRICE COMPARISION WEBSITE \n\n",font=("",20),fg=self.fg\_color,bg=self.bk\_color)

self.flabel.grid(row=0,column=0)

self.product\_name=Label(self.root,text="PRODUCT NAME:",font=("",20),fg=self.fg\_color,bg=self.bk\_color)

self.product\_name.grid(row=1,column=0,sticky=E)

self.product\_name\_entry=Entry(self.root,font=("",15),width=25)

self.product\_name\_entry.grid(row=1,column=1,sticky=W)

self.submit=Button(self.root,text="Submit",font=("",15),fg=self.fg\_color,bg=self.bk\_color,command=self.product\_detail)

self.submit.grid(row=2,columnspan=2,sticky=E)

self.flipkart\_label=Label(self.root,font=("",20),fg=self.fg\_color,bg=self.bk\_color)

self.flipkart\_label.grid(row=3,column=0)

self.amazon\_label=Label(self.root,font=("",20),fg=self.fg\_color,bg=self.bk\_color)

self.amazon\_label.grid(row=3,column=1)

def product\_detail(self):

product=self.product\_name\_entry.get()+".txt"

path1="C:/Users/ANSHUL SINGH/Desktop/anshul/Flipkart/"

cpath1=path1+product

entries1=os.listdir(path1)

if product in entries1:

file1=open(cpath1,"r")

flipkart\_data=file1.read()

path2="C:/Users/ANSHUL SINGH/Desktop/anshul/Amazon/"

cpath2=path2+product

entries2=os.listdir(path2)

if product in entries2:

file2=open(cpath2,"r")

amazon\_data=file2.read()

if(flipkart\_data!='' and amazon\_data!=''):

self.flipkart\_label.config(text=("FLIPKART\n"+flipkart\_data))

self.amazon\_label.config(text=("AMAZON\n"+amazon\_data))

else:

self.flipkart\_label.config(text="DATA NOT FOUND")

if \_\_name\_\_=="\_\_main\_\_":

user=comparison() #making an object for user

user.home()