INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, NAGPUR

IT Workshop 1
PROJECT REPORT

Currency Converter

Submitted To: Mayuri Digalwar

20th November 2020

Submitted by:-
Dhananjay Mishra (BT19CSE135)
Nischay Yadav(BT19CSE136)
Shrungi Katre(BT19CSE132)
Debrupa Nag(BT19CSE131)
Eeshan Wairagade(BT19CSE134)

ACKNOWLEDGEMENT:-

We express our deep sense of gratitude to our guide professor, **Dr. Mayuri Digalwar**, for her valuable guidance and encouragement. We would also like to express our gratitude to all those who were involved directly or indirectly with the completion of this project

Index

Pg No.

Credits

Introduction:

Working of the Project:

1. **Nischay Yadav(BT19CSE136)**
2. **Shrungi Katre(BT19CSE132)**
3. **Dhananjay Mishra(BT19CSE135)**
4. **Debrupa Nag(BT19CSE131)**

Input and Output:

1. **Eeshan Wairagade(BT19CSE134)**

Conclusion:

1.

Introduction

Problem Statement:
To design a currency converter system in which the user can interconvert currency from different countries.

Contemporary Currency Converters:
These days the currency converters are designed in a very regular style in which the systems take all the input from the user including the amount to be converted and between which two countries they wish to interchange and finally the probable output is displayed on the screen.

Novelty in our Project:
Our project's key idea is focused on the concept of currency conversions. We all are accustomed to the basic systems available to us but here we tried to introduce a new feature in which we shortlisted the most frequently used currency so that there's no need for users to search for new countries again and again rather the user gets to experience all frequently used conversions at one glance. Besides, we made this list customisable for future changes.
We built our project using python as our base language. We used basic file handling and OOP concepts. For real-time exchange rates we used online resources to make our system precise and accurate. Along with this we introduced a graphic interface using the python built-in GUI Tkinter module.
Hence making our project user friendly and more user interactive.

Working of the Project

In this project we have to make a currency calculator using python as a programming language. The currency converter project in python requires you to have basic knowledge of python programming like file handling, Dictionary, list,etc.

Before proceeding ahead, let's discuss what are the main steps required to build this project from scratch. They are as follow:-

- ❏ Real-time Exchange rates
- ❏ Most frequent used Exchange Rates
- ❏ Import required Libraries(if GUI)
- ❏ Opening a Text File
- ❏ CurrencyConverter Function
- ❏ Main Function
- ❏ UI for CurrencyConverter (if Made)

So let us Discuss each and every point in detail

1) **Real-time Exchange rates:-** So in this part we copy the content i.e. rates from the website (Click here for website / Check the reference section). It is in base US Dollar it means we have our base currency USD. which means to convert any currency we have to first convert it to USD then from USD, we will convert it in whichever currency we want. Copied content was stored in file CurrencyRate.txt. CurrencyRate.txt will be used while opening a file for modification.

2) **Most frequent used Exchange Rates:-** This section is the same as above. In this section the currency used is frequently used. Our project user has 2 options which contain default calculator and manual calculator which will be discussed in CurrencyConverter function. In default calculator we used mostfrequentlyused.txt(this file contain most frequently used currency rates in base USD) is used.
3) …..

4) Opening a Text File :-

```
with open("CurrencyRate.txt") as file1:
    lines=file1.readlines()

currencyDict={}
for line in lines:
    passed=line.split("\t")
    currencyDict[passed[0]]=passed[1]
```

This block of code is used to open the file named as CurrencyRate.txt. So data of CurrencyRate.txt is stored in the list named as lines. Then we traverse over the list and split the string if blank space("**\t")** is present and splited value stored in the new passed list. passed List contains currency name, currency rate in base-USD.We have created an empty dictionary, there we store currency name as key and currency rate as corresponding value. For example:
Let line contain this string :- "Indian Rupee 74.550914 0.013414"
When we split this string using "**\t"** then passed[0] = Indian Rupee, passed[1] = 74.550914, passed[2] = 0.013414(1 INR how many dollar)
Then we Added this value in currencyDict. If we call currencyDict["Indian Rupee"] then it will give INR corresponding to 1 USD i.e. 74.550914

```
with open("mostfrequentlyused.txt") as file2:
    lines=file2.readlines()

freqDict={}
for line in lines:
    passed=line.split("\t")
    freqDict[passed[0]]=passed[1]
```

This block does the same work as explained above. But here it store most frequently used currency

5) CurrencyConverter Function:-

```
def defaultCal():
    print("\nEnter the Amount")
    try:
        amount=int(input())
        for i in freqDict.keys():
            print(str(amount)+" US Dollor is equal to "+str(amount*float(currencyDict[i]))+" "+str(i))
    except:
        print("\nPlease Enter the amount correctly")
```

This block of code is for default Calculator. In the Default Calculator Most frequently used currency is used. In this function first we take input from the user which should be integer and store it in variable amount. If a user gives a special keyword or a string then it will go in the except block because amount should be always in integer.After taking the amount we traverse in the freqDict Dictionary. While traversing in Dictionary we multiply the value with corresponding currency and also print the same. In this block users put the amount and get converted in frequently used currency. In this block users have no choice for choosing currency.

```
def manualCal():
    print("\nPlease Select the name of currency using only the options Available")
    [print(item)for item in currencyDict.keys()]
    flag=0
    currency1=input("Please Enter which currency you want\n")
    for i in currencyDict.keys():
        if i==currency1:
            flag=1
    if flag==0:
        print("\nPlease Select the currency from the give list")
        return
    flag =0
    currency2=input("Please Enter in which Currency you want to convert\n")
    for i in currencyDict.keys():
        if i==currency2:
            flag=1
    if flag==0:
        print("\nPlease Select the currency from the give list")
        return
    flag =0
    print("\nEnter the Amount")
```

```
try:
    amount=int(input())
    print(str(amount)+" "+str(currency1)+" is equal to "+str(amount*float(currencyDict[currency2])/float(currencyDict[currency1])))
except:
    print("\nPlease Enter the amount correctly")
```

This block of code is for Manual Calculator. In the Manual Calculator Mostly all used currency is used. In this function first we display all available currency which users can use for conversion. Then we ask users to give input which currency they want. If a user gives a currency name which is available then we ask users to give in which currency they want to convert. The program will run normally if both currencies are present else it will stop working and will show respective errors. We handle this Error using the flag variable. We initialize the flag variable as zero assuming currency is not present and if currency is found then we raise the flag which means we assign flag value as 1. If flag is 1 then no error is occurred and then we take input from the user which should be integer and store it in variable amount. If a user gives a special keyword or a string then it will go in the except block because amount should be

always in integer.After taking the amount we traverse in the currencyDict Dictionary. While traversing in Dictionary if we found that key in which the user wants to convert then we simply apply the formula.

6) Main Function:-

```
print("\n\t\t\tCURRENCY CALULATOR")
print("\nDo you want Default Currency Calculator or you want manual Currency Calculator")
print("In Default mode you will get most frequently used results and it's expected that you have Enter in US Dollar\n")
print("\nPress\n1 for Default Calculator \n2 for Manual Calculator\n")
choice=int(input())
if choice == 1:
    defaultCal()
else:
    manualCal()
```

This is the main program of the code of the **Currency Calculator** where you will get 2 options. First Option contains default calculator and Second option contains the manual calculator. Both of the options are discussed in the **CurrencyConverter Function** part; you can refer to this section for more detail.

7) UI for Currency Converter:-

For the making of the GUI, the following widgets were used:

Frame Widget:

The Frame widget is very important for the process of grouping and organizing other widgets in a somehow friendly way. It works like a container, which is responsible for arranging the position of other widgets. It uses rectangular areas in the screen to organize the layout and to provide padding of these widgets.

Syntax:

The syntax to use the checkbutton is given below.

w = frame(master, options)

```
35      # initiate frame
36    mainframe = ttk.Frame(root, padding="3 3 12 12")
37    mainframe.pack(fill=BOTH, expand=1)
```

Entry Widget:

The Entry Widget is a Tkinter Widget used to Enter or display a single line of text.

Syntax :

entry = tk.Entry(parent, options)

For the input field:

```
48      # Add input field
49    in_field = ttk.Entry(mainframe, width=20, textvariable=in_amt)
50    in_field.grid(row=1, column=2, sticky=(W, E))
```

For the output field :

```
    # Add output field and drop-down
ttk.Entry(mainframe, textvariable=out_amt, state="readonly").grid(column=2, row=3, sticky=(W, E))
```

This entry field is in read only state

OptionMenu:

The **OptionMenu** class is a helper class that creates a popup menu, and a button to display it.To get the currently selected value from an option menu, you have to pass in a Tkinter variable

```
    # Add drop-down for input unit
in_select = OptionMenu(mainframe, in_unit, "US Dollar", "Euros", "Indian Rupees", "Swedish krona", "Swiss franc", "Arab Emira

```

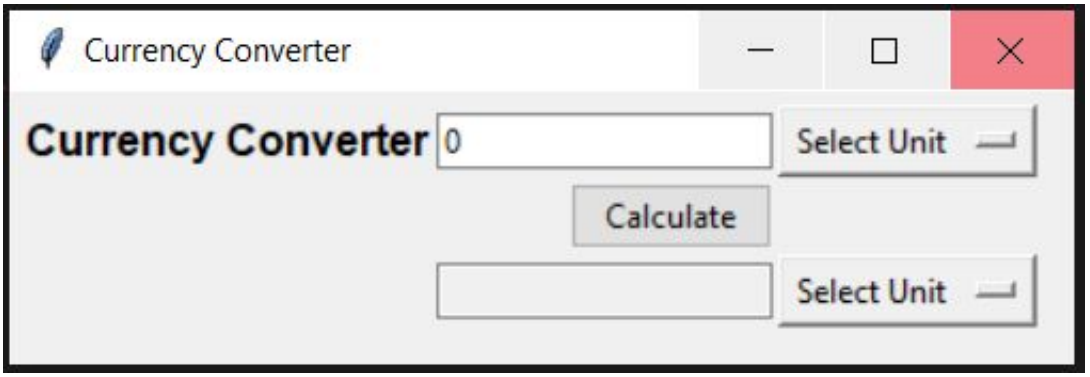
Button:

The Button widget is used to add buttons in a Python application. These buttons can display text or images that convey the purpose of the buttons. We can attach a function or a method to a button which is called automatically when you click the button. In this case, the function callback is called upon clicking the button

```
calc_button = ttk.Button(mainframe, text="Calculate",command=callback).grid(column=2, row=2, sticky=E)

```

Final GUI:



Working of the code:

The pivotal function of the entire code is the Convert() function (computes the new amount in the desired currency by reading the rates from the msf.txt file) and the Callback() function catches any kind of exception.

Callback function:

The exceptions that are checked for are

1. If in_field.get() gives a Value Error exception the output is set to 'Invalid input' else amt variable stores the value of the first field
2. If the in_unit.get() or the out_unit.get() is not selected and the default text of the button i.e 'Select Unit' is returned, the output field is set to ' Input or Output unit not chosen'
3. Else if all the variables possess ideal values, the Convert function is called, passing the input from the first drop down menu in frm variable and second drop down menu input in the to variable along with the amt variable

```
16  def callback():
17      try:
18          amt = float(in_field.get())
19
20      except ValueError:
21          out_amt.set('Invalid input')
22          return None
23      if in_unit.get() == 'Select Unit' or out_unit.get() == 'Select Unit':
24          out_amt.set('Input or output unit not chosen')
25          return None
26      else:
27          frm = ids[in_unit.get()]
28          to = ids[out_unit.get()]
29          out_amt.set(convert(amt, frm, to))
30
31
```

Convert function:

This function reads the file msf.txt and computes the output result using the following formula

`amt*float(freqDict[to])/float(freqDict[frm])`

The lines in the file are read and turned into a dictionary with keys being the first column of the respective Currencies and the value being the second column of their rates(i.e freqDict[passed[0]]=passed[1]

```
14  freqDict={}
15  for line in lines:
16      passed=line.split("\t")
17      freqDict[passed[0]]=passed[1]
18
19  def convert(amt, frm, to):
20      return amt*float(freqDict[to])/float(freqDict[frm])
21
22
```

msf.txt file is as follows:

msf.txt

```
1   USD 1   1
2   SEK 8.690475    0.115069
3   CHF 0.912508    1.095881
4   JPY 104.655006  0.009555
5   INR 74.550914   0.013414
6   CNY 6.604539    0.151411
7   GBP 0.758157    1.318988
8   EUR 0.844946    1.183507
```

Input and Output

```
PS C:\Users\ASUS\Desktop\currency converter> & C:/Users/ASUS/AppData/Local/Programs/Python/Python39/python.exe "c:/Users/ASUS/Desktop/currency converter/CurrencyConverter.py"

                    CURRENCY CALULATOR

Do you want Default Currency Calculator or you want manual Currency Calculator
In Default mode you will get most frequently used results and it's expected that you have Enter in US Dollar


Press
1 for Default Calculator
2 for Manual Calculator
```

On running the currency calculator program the program displays the heading (Currency calculator) a menu and a few lines before it which specify the operations performed by the options given in the menu.

```
1 for Default Calculator
2 for Manual Calculator

1

Enter the Amount
1000
1000 US Dollor is equal to 844.946 Euro
1000 US Dollor is equal to 758.1569999999999 British Pound
1000 US Dollor is equal to 74550.914 Indian Rupee
1000 US Dollor is equal to 1375.2459999999999 Australian Dollar
1000 US Dollor is equal to 1313.3790000000001 Canadian Dollar
1000 US Dollor is equal to 1348.112 Singapore Dollar
1000 US Dollor is equal to 912.508 Swiss Franc
1000 US Dollor is equal to 4122.561 Malaysian Ringgit
1000 US Dollor is equal to 104655.006 Japanese Yen
1000 US Dollor is equal to 6604.539 Chinese Yuan Renminbi
PS C:\Users\ASUS\Desktop\currency converter>
```

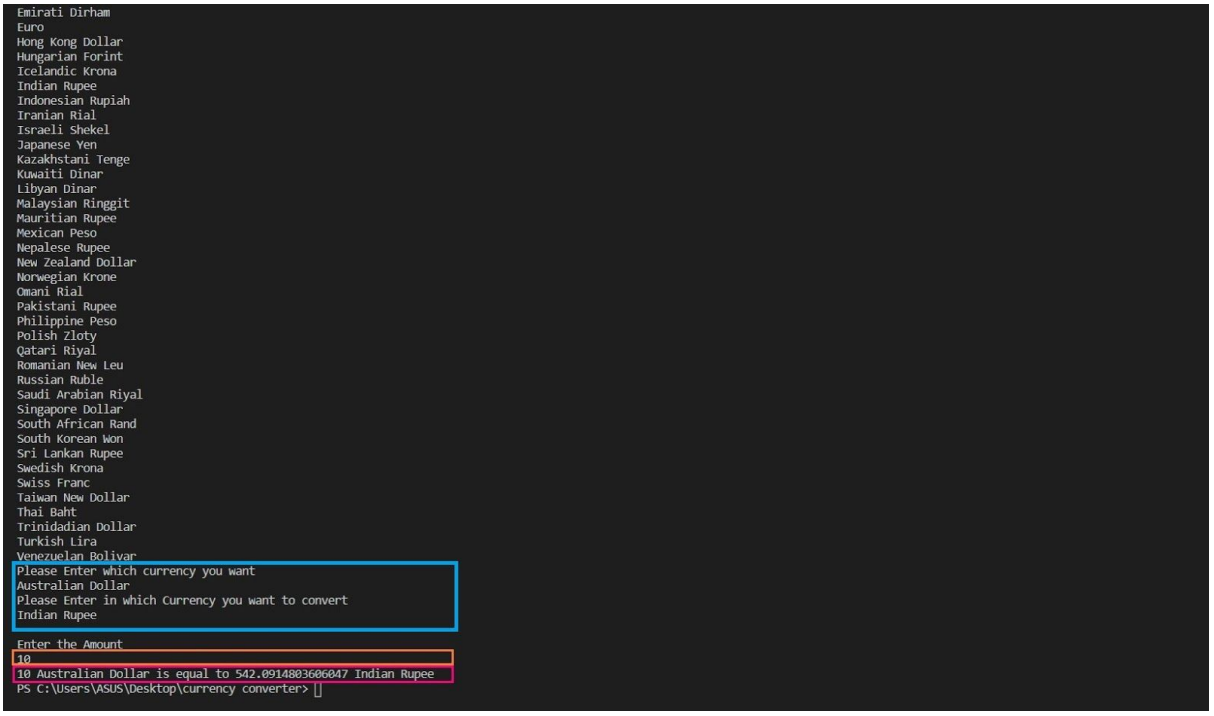
First let us try the default calculator option. So we give input as 1(highlighted in red). The calculator then asks the user to enter the amount (in USD). This amount (highlighted in yellow) is then converted to the most frequently used currency conversions like the Euro, British Pound, Indian Rupee, etc.

```
Press
1 for Default Calculator
2 for Manual Calculator

2

Please Select the name of currency using only the options Available
Argentine Peso
Australian Dollar
Bahraini Dinar
Botswana Pula
Brazilian Real
British Pound
Bruneian Dollar
Bulgarian Lev
Canadian Dollar
Chilean Peso
Chinese Yuan Renminbi
Colombian Peso
Croatian Kuna
Czech Koruna
Danish Krone
Emirati Dirham
Euro
Hong Kong Dollar
Hungarian Forint
Icelandic Krona
Indian Rupee
Indonesian Rupiah
Iranian Rial
Israeli Shekel
Japanese Yen
Kazakhstani Tenge
Kuwaiti Dinar
Libyan Dinar
Malaysian Ringgit
Mauritian Rupee
Mexican Peso
Nepalese Rupee
New Zealand Dollar
Norwegian Krone
Omani Rial
Pakistani Rupee
Philippine Peso
```

To try out the manual calculator the user has to give input 2 (highlighted in red in above picture). This generates a list of all available currencies.



The program then asks the user the currency which is to be converted and the currency to which the previous currency is to be converted to as shown in the blue box.
Then the program asks for the amount to be converted (highlighted in orange), and shows the final conversion (highlighted in pink).

Conclusion

We implemented the converter system using python language and further to make it more user friendly and interactive we used a graphic user interface as Tkinter.

References

- https://www.x-rates.com/table/?from=USD&amount=1
- https://www.geeksforgeeks.org/python-tkinter-tutorial/#basic
- https://www.geeksforgeeks.org/tkinter-read-only-entry-widget/
- https://www.geeksforgeeks.org/python-tkinter-entry-widget/
- https://www.geeksforgeeks.org/python-menu-widget-in-tkinter/
- OtionMenu in Tkinter
- https://www.geeksforgeeks.org/python-tkinter-frame-widget/
- https://www.journaldev.com/33500/python-valueerror-exception-handling-examples