Currency Convertor

Pooja Chauhan SYBSCIT A003 – 45207220011

Abstract/Synopsis:

This project is about a currency convertor which can convert any type of currency to another currency within seconds. This programs used functions, loops and strings. Tkinter is the main highlight of the program to create a GUI based convertor. This is just a demo program and hence the exchange rates of the currency are not stored and have to be typed out by the user itself. This program consists of 100 lines of code and was created using PyCharm and IDLE od Python.

Keywords:

Some of the keywords related to the program are:

tkinter, root, my_notebook, def lock(), def unlock(), def convert(), def clear(), root.mainloop, root.geometry, conversion_entry, conversion_label, etc.

Objective and Scope:

The main objective of the program is to provide the user with a user-friendly interface where he/she can convert currencies to any other currency in the world. This program fulfils all the needs of the scope and the objective of the project.

Detailed Working:

1. **def lock():**

This is the 1st function used in the program. This function helps us to lock the values that we enter under the currency tab (can be seen in the 1st and 3rd screenshot provided). If the user does not enter any input under the currency tab then a message will appear which is, "WARNING!", "Fill Out All The Fields". Under this function we alos use the if-else loop. In this function we basically lock the values entered and then proceed to the next tab.

2. **def unlock():**

This is the 2nd function used in this program. This function enables the text boxes in the convert tab for the user to enter the values and convert the currency. This function even disables the text boxes in the currency tab so that the program can run based on the values entered.

3. **def convert():**

This is the 3rd function of the program. This function enables the clear button for the user to clear the values entered in the convert tab. This function even accepts the values from the users for the amount to convert to. The output shown will be rounded off and have 2 decimal places. It will even add commas to the final output.

4. **def clear():**

This is the 4th function of the program and just has a simple job of clearing the values ended by the user.

Use and Purpose:

The main use of this code is basically viewing the converted currency and also helping the user to enter the desired exchange rate of the currency to convert the amount. This program can also be used to convert the currency based on an exchange rate dated years back and compare the prices.

Merits and Demerits:

Merits:

- Any type of currency can be converted to any other type of currency.
- It is easy to use, understand and is user-friendly.

Demerits:

- The values/exchange rate of currencies cannot be stored.
- User will have to enter the exchange rates every time he/she wants to convert a currency.

Future Enhancements:

- User can store the converted rates in a database.
- Exchange rates will be stored in a database and can then be accessed via the program.
- A graph can also be showed for the variations of each currency overtime.

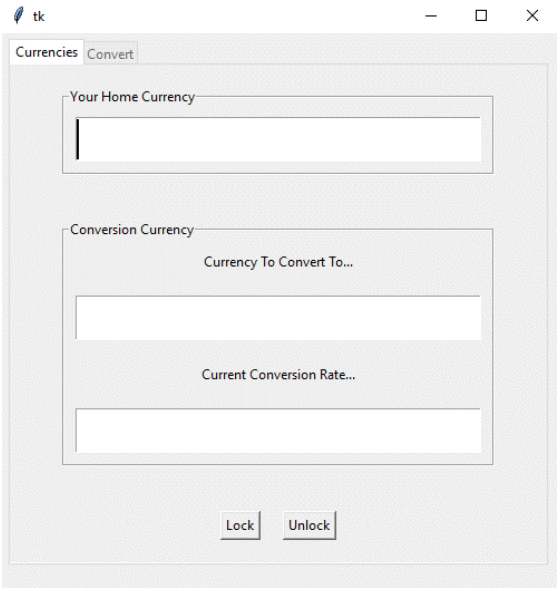
References:

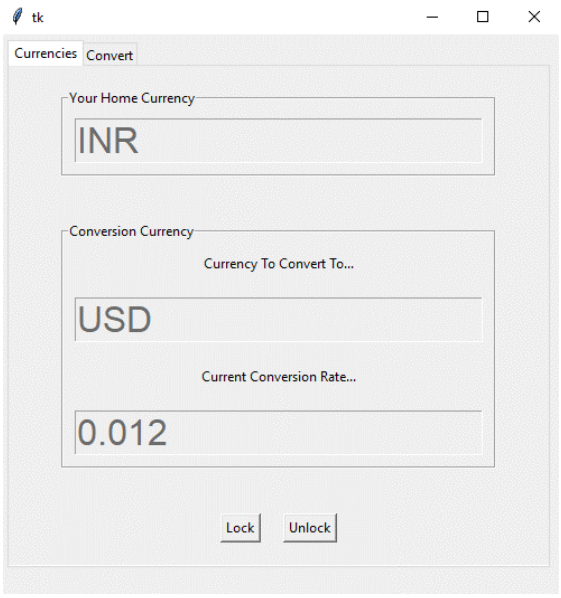
- Think Python (1st Edition) by Allen Downey
- Object-oriented Programming in Python (1st Edition) by Michael H. Goldwasser & David Letscher

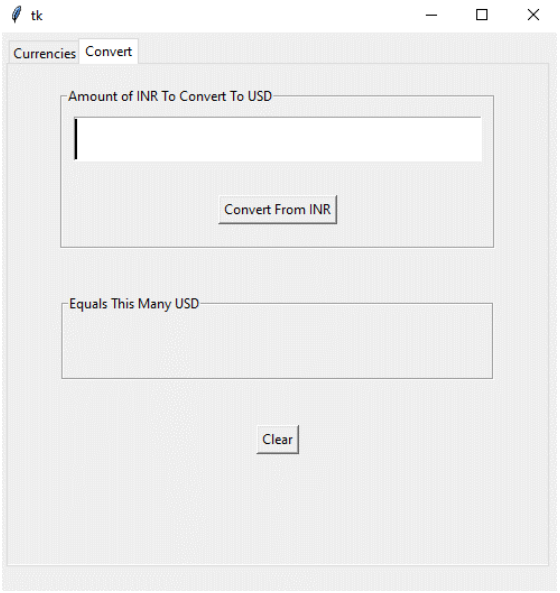
Web References:

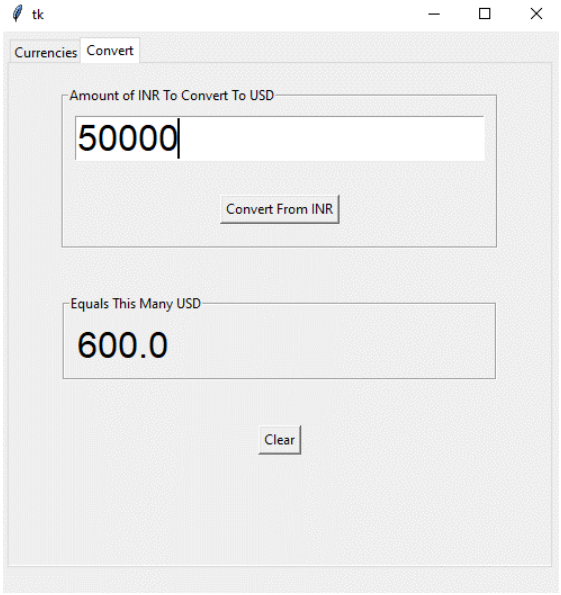
- www.w3schools.com
- www.javatpoint.com
- www.stackoverflow.com
- www.github.com
- www.geeksforgeeks.com

SCREENSHOTS:









SOURCE CODE:

```
import tkinter as tk
from tkinter import ttk
from tkinter import messagebox

root = tk.Tk()
root.geometry("500x500")
root.option_add("*TButton*Font", "Times 12")

my_notebook = ttk.Notebook(root)
my_notebook.pack(pady=5)

currency_frame = tk.Frame(my_notebook, width=480, height=480)
conversion_frame = tk.Frame(my_notebook, width=480, height=480)

currency_frame.pack(fill="both", expand=1)
conversion_frame.pack(fill="both", expand=1)

my_notebook.add(currency_frame, text="Currency")
my_notebook.add(conversion_frame, text="Convert")

my_notebook.tab(1, state='disabled')

def lock():
    if not home_entry.get() or not conversion_entry.get() or not
rate_entry.get():
        messagebox.showwarning("WARNING!", "Fill Out All The Fields")
    else:
        home_entry.config(state="disabled")
        conversion_entry.config(state="disabled")
        rate_entry.config(state="disabled")
        my_notebook.tab(1, state='normal')
        amount_label.config(text=f'Amount of {home_entry.get()} To Convert To
{conversion_entry.get()}')
        converted_label.config(text=f'Converted Rate
{conversion_entry.get()}')
        convert_button.config(text=f'Convert')

def unlock():
    home_entry.config(state="normal")
    conversion_entry.config(state="normal")
    rate_entry.config(state="normal")
    my_notebook.tab(1, state='disabled')

home = tk.LabelFrame(currency_frame, text="Currency To Convert From")
home.pack(pady=20)

home_entry = tk.Entry(home, font=("Times New Roman", 24))
```

```
home_entry.pack(pady=10, padx=10)

conversion = tk.LabelFrame(currency_frame, text="Conversion Currency")
conversion.pack(pady=20)

conversion_label = tk.Label(conversion, text="Currency To Convert To")
conversion_label.pack(pady=10)

conversion_entry = tk.Entry(conversion, font=("Times New Roman", 24))
conversion_entry.pack(pady=10, padx=10)

rate_label = tk.Label(conversion, text="Current Exchange Rate of Currency")
rate_label.pack(pady=10)

rate_entry = tk.Entry(conversion, font=("Times New Roman", 24))
rate_entry.pack(pady=10, padx=10)

button_frame = tk.Frame(currency_frame)
button_frame.pack(pady=20)

lock_button = tk.Button(button_frame, text="Lock", command=lock, font=("Times
New Roman", 12))
lock_button.grid(row=0, column=0, padx=10)

unlock_button = tk.Button(button_frame, text="Unlock", command=unlock,
font=("Times New Roman", 12))
unlock_button.grid(row=0, column=1, padx=10)

def convert():
    converted_entry.delete(0, tk.END)
    try:
        conversion = float(rate_entry.get()) * float(amount_entry.get())
        converted_entry.insert(0, f'{conversion}')
    except ValueError:
        messagebox.showerror("Error", "Invalid input")

def clear():
    amount_entry.delete(0, tk.END)
    converted_entry.delete(0, tk.END)

amount_label = tk.LabelFrame(conversion_frame, text="Amount To Convert")
amount_label.pack(pady=20)

amount_entry = tk.Entry(amount_label, font=("Times New Roman", 24))
amount_entry.pack(pady=10, padx=10)

convert_button = tk.Button(amount_label, text="Convert", command=convert,
font=("Times New Roman", 12))
```

```
convert_button.pack(pady=20)

converted_label = tk.LabelFrame(conversion_frame, text="Converted Currency")
converted_label.pack(pady=20)

converted_entry = tk.Entry(converted_label, font=("Times New Roman", 24),
bd=0, bg="systembuttonface")
converted_entry.pack(pady=10, padx=10)

clear_button = tk.Button(conversion_frame, text="Clear", command=clear,
font=("Times New Roman", 12))
clear_button.pack(pady=20)

spacer = tk.Label(conversion_frame, text="", width=68)
spacer.pack()

root.mainloop()
```