# # Python Project on Currency Converter

import requests

from tkinter import \*

import tkinter as tk

from tkinter import ttk

class RealTimeCurrencyConverter():

    def \_\_init\_\_(self,url):

            self.data = requests.get(url).json()

            self.currencies = self.data['rates']

    def convert(self, from\_currency, to\_currency, amount):

        initial\_amount = amount

        if from\_currency != 'USD' :

            amount = amount / self.currencies[from\_currency]

        # limiting the precision to 4 decimal places

        amount = round(amount \* self.currencies[to\_currency], 4)

        return amount

class App(tk.Tk):

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

        tk.Tk.\_\_init\_\_(self)

        self.title = 'Currency Converter'

        self.currency\_converter = converter

        #self.configure(background = 'blue')

        self.geometry("500x200")

        # Label

        self.intro\_label = Label(self, text = 'Welcome to Real Time Currency Convertor',  fg = 'blue', relief = tk.RAISED, borderwidth = 3)

        self.intro\_label.config(font = ('Courier',15,'bold'))

        self.date\_label = Label(self, text = f"1 Indian Rupee equals = {self.currency\_converter.convert('INR','USD',1)} USD \n Date : {self.currency\_converter.data['date']}", relief = tk.GROOVE, borderwidth = 5)

        self.intro\_label.place(x = 10 , y = 5)

        self.date\_label.place(x = 160, y= 50)

        # Entry box

        valid = (self.register(self.restrictNumberOnly), '%d', '%P')

        self.amount\_field = Entry(self,bd = 3, relief = tk.RIDGE, justify = tk.CENTER,validate='key', validatecommand=valid)

        self.converted\_amount\_field\_label = Label(self, text = '', fg = 'black', bg = 'white', relief = tk.RIDGE, justify = tk.CENTER, width = 17, borderwidth = 3)

        # dropdown

        self.from\_currency\_variable = StringVar(self)

        self.from\_currency\_variable.set("INR") # default value

        self.to\_currency\_variable = StringVar(self)

        self.to\_currency\_variable.set("USD") # default value

        font = ("Courier", 12, "bold")

        self.option\_add('\*TCombobox\*Listbox.font', font)

        self.from\_currency\_dropdown = ttk.Combobox(self, textvariable=self.from\_currency\_variable,values=list(self.currency\_converter.currencies.keys()), font = font, state = 'readonly', width = 12, justify = tk.CENTER)

        self.to\_currency\_dropdown = ttk.Combobox(self, textvariable=self.to\_currency\_variable,values=list(self.currency\_converter.currencies.keys()), font = font, state = 'readonly', width = 12, justify = tk.CENTER)

        # placing

        self.from\_currency\_dropdown.place(x = 30, y= 120)

        self.amount\_field.place(x = 36, y = 150)

        self.to\_currency\_dropdown.place(x = 340, y= 120)

        #self.converted\_amount\_field.place(x = 346, y = 150)

        self.converted\_amount\_field\_label.place(x = 346, y = 150)

        # Convert button

        self.convert\_button = Button(self, text = "Convert", fg = "black", command = self.perform)

        self.convert\_button.config(font=('Courier', 10, 'bold'))

        self.convert\_button.place(x = 225, y = 135)

    def perform(self):

        amount = float(self.amount\_field.get())

        from\_curr = self.from\_currency\_variable.get()

        to\_curr = self.to\_currency\_variable.get()

        converted\_amount = self.currency\_converter.convert(from\_curr,to\_curr,amount)

        converted\_amount = round(converted\_amount, 2)

        self.converted\_amount\_field\_label.config(text = str(converted\_amount))

    def restrictNumberOnly(self, action, string):

        regex = re.compile(r"[0-9,]\*?(\.)?[0-9,]\*$")

        result = regex.match(string)

        return (string == "" or (string.count('.') <= 1 and result is not None))

if \_\_name\_\_ == '\_\_main\_\_':

    url = 'https://api.exchangerate-api.com/v4/latest/USD'

    converter = RealTimeCurrencyConverter(url)

    App(converter)

    mainloop()