ABSTRACT

The Currency converter In Python is a simple project developed using Python. This project is a GUI application which converts Currency from one unit to another (euros and pounds). Also, this app is capable of handling all types of exceptions. This project is an interesting useful project.

This Currency Converter App is in Python. Talking about the features of this system, this python application is designed to convert entered numbers from one system/unit to other system/unit and it is also capable of handling all types of exceptions. Module Used – Tkinter()-It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter outputs the fastest and easiest way to create the GUI applications. Also, the design of this system is pretty simple so that the user won't get any difficulties while working on it.

III

Table of Contents

Chapter-1

Introduction

The Currency converter In Python is a simple project developed using Python. This project is a GUI application which converts Currency from one unit to another (euros and pounds). Also, this app is capable of handling all types of exceptions. This project is an interesting useful project.

This Currency Converter App is in Python. Talking about the features of this system, this python application is designed to convert entered numbers from one system/unit to other system/unit and it is also capable of handling all types of exceptions. Module Used – Tkinter()-It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter outputs the fastest and easiest way to create the GUI applications. Also, the design of this system is pretty simple so that the user won't get any difficulties while working on it.

1.1 Problem Statement

The reason why we selected Currency Coverter is because it is a basic necessity in everyday life,in terms of foreign exchange,stock market etc.

1.2 Objectives:

- To convert one currency form to another based on current currency prices.
- Helps the user to convert any currency without any difficulty in the process at a single location.

1.2 Scope:

This Currency Converter system provides the simplest currency convertion based on current market prices at a single interface. In short, this projects mainly focus on adding and calculating results. There's no external database connection file used in this mini project to save user's data permanently. The main aim of the entire activity is to automate the process of currency convertion.

Chapter-2

Code Snippets

```
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)
```

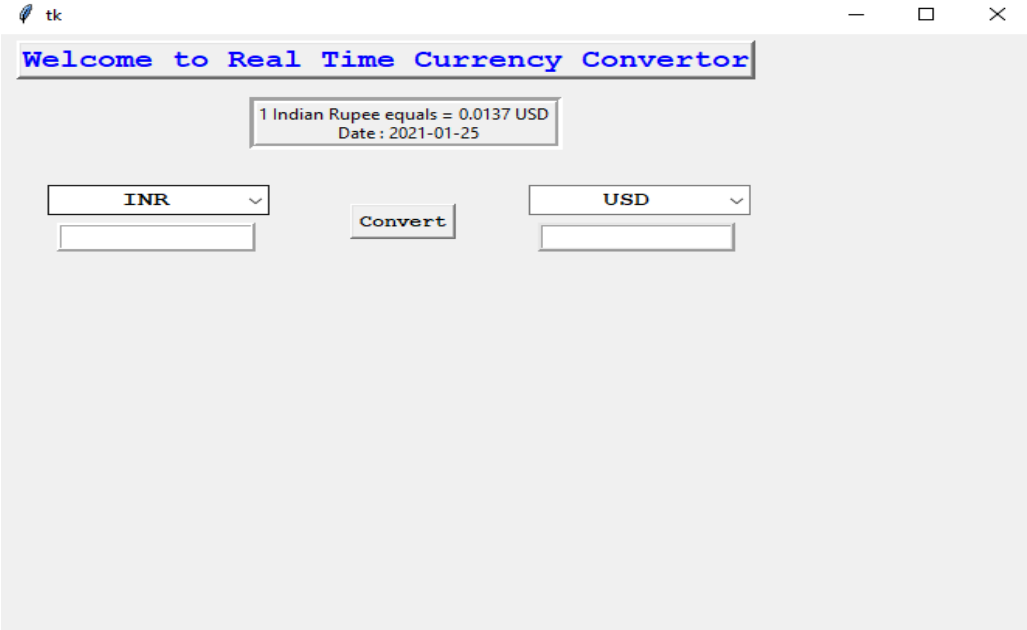
```
self.converted_amount_field_label = Label(self, text = '', fg = 'black', bg = 'white', relief = tk.RIDGE, justify = tk.CENTER, width = 17, borderwidth = 3)


# 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)

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();
```

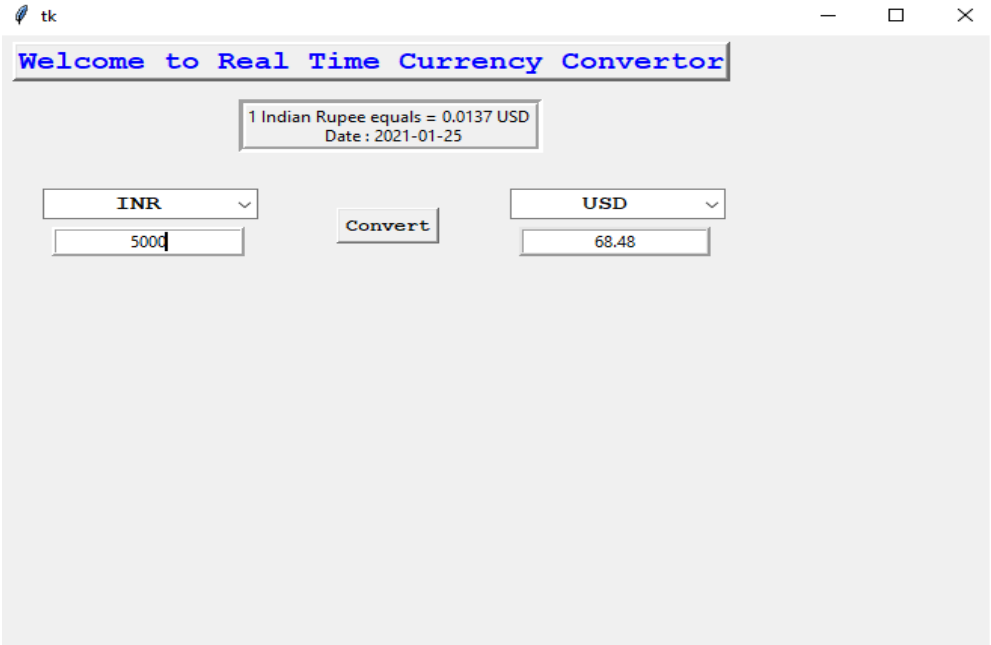
Chapter 3

Snapshots

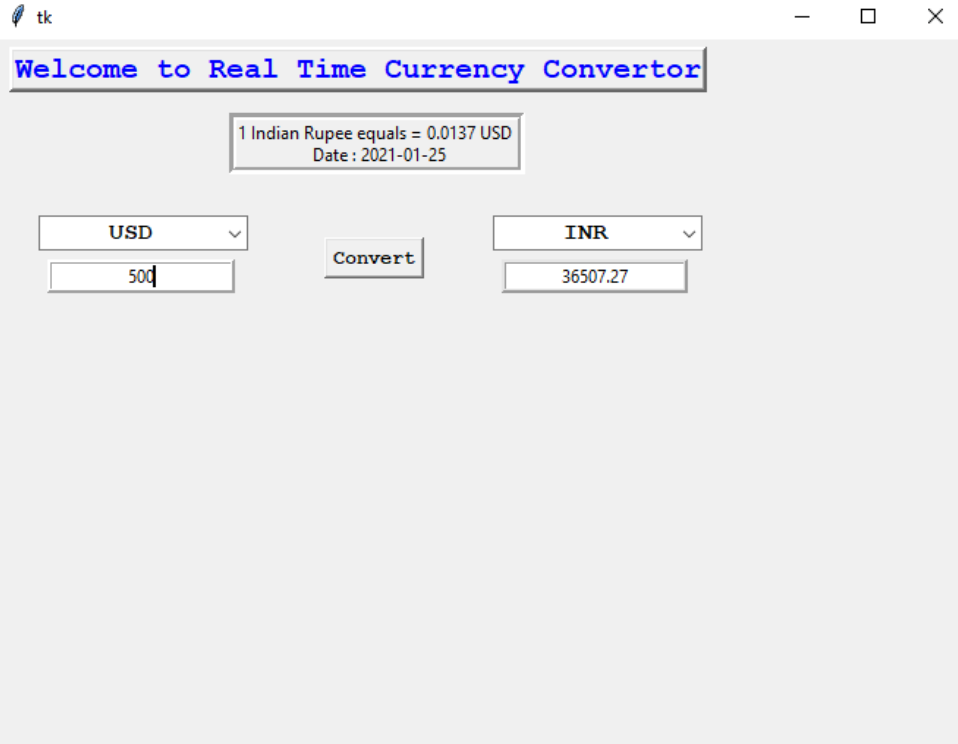
3.1 Currency Converter Layout



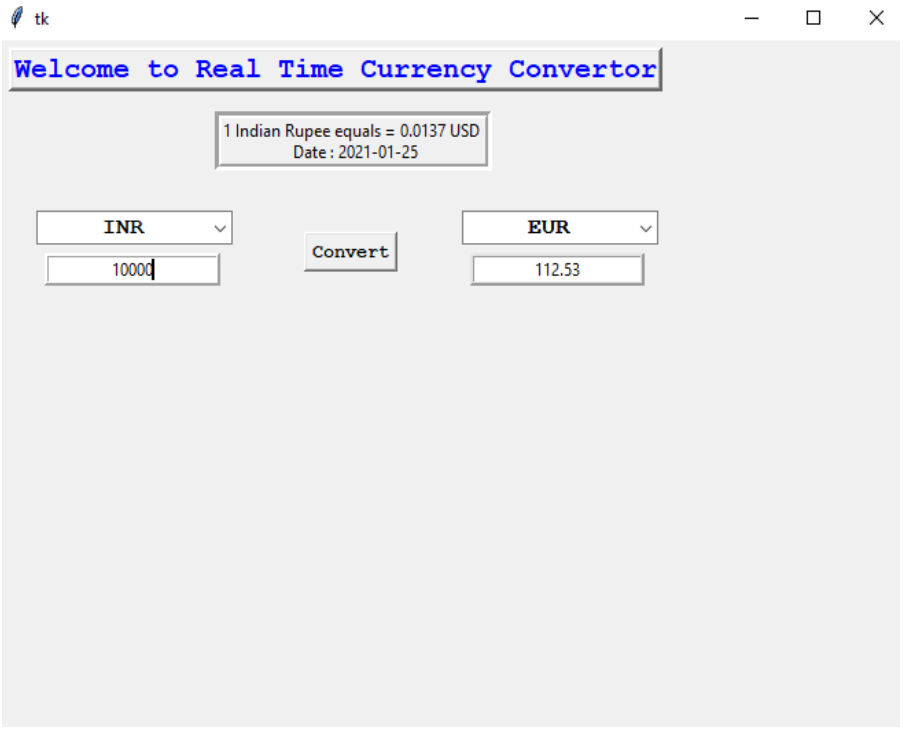
3.2 Indian Rupee To US Dollar



3.3 US Dollar to Indian Rupee



3.3 Indian Rupee to Euro



Conclusion

What we have here is easy to use, highly beneficial website. The design of the website is done by taking in to the consideration of all the user's needs, boxing them and presenting it in the best way possible. Using the website saves the much needed time and helps streamline the entire process. The website has been completed successfully and tested with suitable test cases. It is user friendly and contains suitable options for all users. This project is part of the spearhead the pierces the veil of redundancy and creates a future where everything is on the web and easily accessible.

Currency Converter , is a project aimed to faster convertion of any currency to any other currency at the standard international rates.

References

[1]. W3 Schools (Python reference) – https://www.w3schools.com/

[2]. Stack Overflow – https://stackoverflow.com/

[3]. Stack Exchange – https://stackexchange.com/

[4]. Wikipedia – https://www.wikipedia.org/