14759912

Abstract

THIS PORTFOLIO IS A SYSTEMATIC PRESENTATION OF SOFTWARE BEHIND CURRENCY CALCULATOR AND HOW IT CAN BE OF A GREAT HELP TO PEOPLE.

PROBLEM SOLVING aND PROGRAMMING

A PROFESSIONAL PORTFOLIO SHOWCASING MY PROGRAMMING SKILLS



Contents

[Table of Figures 2](#_Toc168567485)

[Introduction 3](#_Toc168567486)

[a) Overview of the problem 3](#_Toc168567487)

[b) Purpose of the problem 3](#_Toc168567488)

[c) Goals of a currency calculator 3](#_Toc168567489)

[Task 1: Problem solving techniques 4](#_Toc168567490)

[1.1 Real world problem 4](#_Toc168567491)

[1.2 Analysis of the problem 4](#_Toc168567492)

[1.3 Designing 4](#_Toc168567493)

[Task 2: 4](#_Toc168567494)

[2.1 List of steps to solve problem 4](#_Toc168567495)

[2.2 Psuedocode 5](#_Toc168567496)

[2.3 Flowchart 6](#_Toc168567497)

[2.4 Explanation of the flowchart 7](#_Toc168567498)

[2.5 Complexity of algorithm table 8](#_Toc168567499)

[**Complexity Table** 8](#_Toc168567500)

[Task 3: Programming language techniques 8](#_Toc168567501)

[3.1 Syntax elements for psuedocode 8](#_Toc168567502)

[Task 4: Software Development 10](#_Toc168567503)

[4.1 Software development life cycle 10](#_Toc168567504)

[4.2 Challenges encountered 13](#_Toc168567505)

[4.3 Video Demonstration 14](#_Toc168567506)

[Conclusion 14](#_Toc168567507)

[References 14](#_Toc168567508)

[Appendix 15](#_Toc168567509)

CURRENCY CALCULATOR

# Introduction

## a) Overview of the problem

The currency calculator is a practical example of how to build an interactive Python application. It allows users to perform currency conversions between GBP and INR and handles user input in a clear and structured manner. This program serves as a solid foundation for further enhancements, such as integrating real-time exchange rates

## b) Purpose of the problem

The currency calculator built using Python serves multiple purposes:

a) Educational: It helps beginners learn and apply core programming concepts and practices.

b) Practical: It provides a functional tool for converting currency, useful in various real-world scenarios.

c) Developmental: It offers a foundation for building more complex applications and adding advanced features.

d) Community: It encourages sharing and collaboration, fostering a community of learning and development.

## c) Goals of a currency calculator

a) Provide quick and responsive performance, with minimal delay in fetching exchange rates and calculating conversions.

b) Design a user-friendly interface that allows users to easily input data and understand the conversion results.

c) Aim for a high level of user satisfaction by providing a reliable and efficient tool for currency conversion.

# Task 1: Problem solving techniques

## 1.1 Real world problem

In this task I’m going to be explaining the software behind the "Currency Calculator" from GBP to INR and vice versa. This is helpful to the people who are calculating prices for things like transport, money exchange, groceries , rental etc

Pound sterling /Quid/ Nicker (£) is the currency in United Kingdom and Rupee (₹) in India.

## 1.2 Analysis of the problem

~First, we need to note down each of the currency values in each other countries and compare them with mathematical values.

~Now, note the value u want to convert from and note the value u want to convert to and then note the amount of money u want to convert. These will be the input values for further process.

~Showcase a button to trigger the conversion.

## 1.3 Designing

Designing: From establishing the program structure to putting the main feature into practice, there are various important tasks involved in designing a currency calculator in Python.

Step 1: Specify the structure of the program. The user will have the option to convert currencies or close the program through a menu-driven interface.

Step 2: Establish Conversion Rates. Specify the conversion rates between INR and GBP.

Step 3: Put Functions Into Practice. Provide the menu's display, conversion, and user input handling capabilities. Execute the Main Loop in Step Four and make a loop that does not stop the application from operating until the user decides to stop it. Add error handling in step five (optional). Improve the software by using error.

# Task 2:

## 2.1 List of steps to solve problem

a) Requirements Gathering and Analysis. Identify the primary features needed and Input field for the amount.

b) System Design

c) Setting Up the Development Environment

d) Implementing the Frontend, Implementing the Backend, Connecting Frontend and Backend

e) Testing

f) Deployment

g) Maintenance and Updates. Monitor the application for any issues or bugs.

## 2.2 Psuedocode

# Define the function convert currency (amount, from currency, to currency) to convert currencies:

# Dictionary to record exchange rates in relation to a reference currency (GBP, for example)

EXCHANGE\_RATES = { "GBP": 1.0, "INR":106.27,etc # Add more currencies and their rates as necessary

#Transform the amount into GBP pounds (the base currency) using the formula amount in GBP = amount / EXCHANGE\_RATES [from currency].

#Transfer money between the base and destination currencies: converted amount = amount in GBP \* EXCHANGE\_RATES [to currency] RETURN the amount converter.

#Principal application

PRINT "Currency Calculator!" is the output of function main (). Now, get the amount entered by the user (INPUT). Enter the desired conversion amount by typing "from currency = INPUT. "Enter the original currency (GBP, INR etc.) that you are converting from: " to currency = INPUT "Input the target currency (GBP, INR etc.): "

## 2.3 Flowchart

Figure : Flowchart for currency calculator

Start

Display menu options User Input: Choice (1,2,3)

Select an option

Prompt user: Enter GBP Amount

Print ”Exiting the program.”

Prompt User: Enter INR Amount

Exit

Convert GBP to INR (GBP to INR)

Convert INR to GBP (INR to GBP)

Display Result:“X INR is equal to Y GBP”

Display Result: "X GBP is equal to Y INR"

Loop Back to Display Menu Options

## 2.4 Explanation of the flowchart

This flowchart represents a simple interactive program that allows the user to perform currency conversions between GBP and INR, with the ability to exit the program at any time.

~Display menu function does the following steps:-

a) Shows she user the menu selections and asks the user to select one of three options (1, 2, or 3).

b) Gives the user's selection back.

c) Takes the conversion rate and the amount in GBP and gives back the same amount in Indian Rupees.

d) Takes the conversion rate and the amount in Indian rupees and gives back the same amount in British pounds.

~Principal function:

It establishes sample exchange rates between GBP and INR and starts a loop that doesn't end until the user decides to stop it.

~Considering the user's selection:

a)If the answer is '1', it asks for the amount in GBP, converts it to INR, and shows the outcome.

b)If the answer is '2', it asks for the INR amount, translates it to GBP, and shows the outcome.

c)If '3', the loop is just ended.

## 2.5 Complexity of algorithm table

Table 1: Complexity Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Component** | **Description** | **Operation Count** | **Complexity** |
| Menu Display | Displaying menu options | Fixed number of print ops | O(1) |
| User Input Handling | Taking user inputs for choice and amounts | Fixed number of inputs | O(1) |
| Currency Conversion | Arithmetic operations for conversion | Fixed number of operations | O(1) |
| Loop for Repeated Operations | Repeatedly showing menu until user exits | Depends on user interactions | O(n) |

# Task 3: Programming language techniques

## 3.1 Syntax elements for psuedocode

Required variables along with Inputs and Outputs:

Print: To print various statements on the application, we utilize the word "print." The syntax "print" symbol's assigned symbol is (). For instance, you would type print "statement" to tell a statement. and for output you will write ("Convert the amount").

Function: A function is a code that will only run when it is called. Making it readable to the user very easily is the ultimate purpose of function. It can be used at limitless times after it is defined thoroughly.

~The function definition is initiated with the use of the term def. example: 'def total amount'.

~Conditional statement:-- If condition 1 is true, the following code block will be executed:(Shaw, Z. A. 2013)

~Pass elif condition 2: # Code block to run in the event that condition 2 is true: pass else:

~ Code block to run in the event that none of the aforementioned conditions hold true pass.

# Task 4: Software Development

## 4.1 Software development life cycle

This method is used in building a currency calculator.

**1. Requirements Gathering and Analysis:**

~Objective: Identify and document the requirements for the currency calculator.

Identify who will use the currency calculator (e.g., travelers, finance professionals).

Understand their needs: converting between multiple currencies, using real-time exchange rates, simple and intuitive UI.

~Requirements Documentation:

Functional Requirements: Input field for entering the amount.

Dropdown menus for selecting source and target currencies.

Button to trigger the conversion.

Display area for showing the conversion result.

Integration with an API for real-time exchange rates.

Non-Functional Requirements:

User-friendly and intuitive interface.

Fast and accurate conversion.

Error handling for invalid inputs and network issues.

**2. System Design:**

Objective: Plan the architecture and design of the currency calculator.

High-Level Design:

User Interface (UI) Design:

Sketch the layout: input fields, currency dropdowns, convert button, result display.

Backend Design: Decide on the API for fetching real-time exchange rates

Plan the logic for fetching exchange rates and performing the conversion.

**Error Handling:**

Design mechanisms for handling invalid inputs and network/API errors.

Detailed Design, UI Elements

Define properties and behaviour of UI components: labels, entry fields, buttons, result labels etc

Conversion Logic: Detail the algorithm for currency conversion using fetched exchange rates.

**3. Implementation:**

Objective: Write the code based on the detailed design.

Frontend Development: Use Tkinter to create the user interface with labels, entry fields, combo boxes, buttons, and result labels.

Example code snippet: Python

Copy code and import tkinte as tk , from tkinter import ttk

# Initialize main window

root = tk. Tk(). root. title("Currency Calculator")

# Amount entry

tk. Label(root, text="Amount:").grid(row=0, column=0, padx=10, pady=10). Amount entry = tk. Entry(root). Amount entry. grid(row=0, column=1, padx=10, pady=10)

# From currency dropdown

tk.Label(root, text="From:").grid(row=1, column=0, padx=10, pady=10)

from\_currency\_combobox = ttk.Combobox(root, values=['GBP,INR'])

from\_currency\_combobox.grid(row=1, column=1, padx=10, pady=10)

from\_currency\_combobox.current(0)

# To currency dropdown: tk.Label(root, text="To:").grid(row=2, column=0, padx=10, pady=10). to\_currency\_combobox = ttk.Combobox(root, values=['GBP,INR']). to\_currency\_combobox.grid(row=2, column=1, padx=10, pady=10). to\_currency\_combobox.current(1)

# Convert button

def convert\_currency():

# Placeholder function for conversion logic

convert\_button = tk.Button(root, text="Convert", command=convert\_currency)

convert\_button.grid(row=3, column=0, columnspan=2, pady=10)

# Result label: result\_label = tk.Label(root, text="", font=('Helvetica', 14))

result\_label.grid(row=4, column=0, columnspan=2, pady=10). root.mainloop()

Backend Development: Write the logic to fetch exchange rates from the chosen API. Now, Implement the conversion calculation using the fetched exchange rates.

Example code snippet: Python and copy the code and import requests.

def get\_exchange\_rate(from\_currency, to\_currency): # Fetch real-time exchange rates and response = requests.get(url). Now, data = response .json () return data['rates'][to currency.

**4. Testing Objective:**

Unit Testing: Test individual components: UI elements, conversion logic. Validate conversion calculations with known rates.

Integration Testing: Test the interaction between the frontend UI and backend logic. Ensure input values are correctly passed to the conversion logic and results are accurately displayed.

System Testing: Test the entire application to ensure all parts work together as expected. Simulate user interactions and verify correct behaviour.

User Acceptance Testing (UAT):Have end users test the application.

Collect feedback and make necessary adjustments.

**5. Deployment Objective:**

Make the currency calculator available to users. Packaging: Prepare the application for distribution, such as creating an executable file or deploying it as a web application.

Installation: Provide clear installation instructions for users. Ensure the application can be easily installed and used.

**6. Maintenance:**

Objective:- Keep the currency calculator functional and up-to-date.

Bug Fixes: Address issues reported by users post-deployment. Updates: Add new features or improve existing ones based on user feedback. Ensure the application continues to fetch accurate exchange rates and perform optimally.

Monitoring: Regularly check the application’s performance and accuracy of exchange rates. Make necessary adjustments to handle changes in API endpoints or data structures.

By following these steps using the waterfall method, you can ensure a well-structured and organized development process for the currency calculator. Each phase has clear objectives and deliverables, leading to a functional and reliable application.

## 4.2 Challenges encountered

1. The first challenge is ensuring that the currency rates used are current and correct. Solution: To retrieve real-time currency rates, integrate with a dependable API (such Open Exchange Rates, Currency Layer, or Forex).
2. The challenge of API integration: managing API keys, responding to requests for and from the program, and making sure the application can withstand any outages or rate restrictions imposed by the external service. Resolution: Set up reliable systems for handling errors, cache data, and respect API usage limitations.
3. Overcoming the Challenge of Various Currencies: Encouraging a large variety of currencies, each with distinct formatting guidelines, decimal places, and symbols. Solution: Make use of libraries that handle many of these complexities, such as currency converter or forex-python.
4. User Input Validation Challenge: Making sure that user inputs are accurate and thoroughly cleaned to prevent security breaches.

## 4.3 Video Demonstration

The video demonstration is in AULA Handin.

# Conclusion

In summary,

Using Python to create a currency calculator is a great method to practice applying basic programming ideas like loops, functions, conditional statements, and user input/output. Based on user input, the program we've been discussing offers a simple and interactive command-line interface that enables users to convert between GBP and INR. The following are some important lessons learned:-

Interface that's Easy to Use: Users may easily explore the software and conduct desired operations without confusion thanks to the menu-driven approach.

Usability: The application efficiently demonstrates the usage of conversion factors and fundamental arithmetic operations by converting monetary amounts based on predetermined exchange rates. Reusability and Modularity of Code: The code stays structured and is simple to update or expand by utilizing functions for managing user input, converting currencies, and displaying the menu.

# References

<https://www.learningmilestone.com/post/currency-converter-python>, Accessed on 04 June 2024

<https://data-flair.training/blogs/currency-converter-python>, Accessed on 04 June 2024

### **Eric Matthes**

* **Book**: "Python Crash Course"

# Appendix

Code

#  According to the problem,the function display\_menu should not return any value.It should just display the menu

def display\_menu():

  print("1. Convert GBP to Rupees(INR)")

  print("2. Convert INR to British Pound Sterling (GBP)")

  print("3. Invalid")

#The other four functions that convert currencies accept a value through an argument and return the converted value

def GBP\_to\_INR(value):

  return value\*106.27

def INR\_to\_GBP(value):

  return value\*0.0094

while True:

  display\_menu()

  choice=int(input())

  if choice==3:

    print("Invalid!")

    break

  else:

    amount=float(input("Enter an amount in GBP Pounds: "))

    if choice==1:

      print(amount,"GBP",GBP\_to\_INR(amount),"INR")

    elif choice==2:

      print(amount,"INR",INR\_to\_GBP(amount),"GBP")