# Currency Converter Application

A Python Project using Tkinter

**Submitted by:** [Student Name]

**Semester:** 1st Sem (CS)

# Introduction & Problem Statement

- **The Problem:** Manual currency conversion is time-consuming and prone to mathematical errors. Users often need to check multiple websites for rates.
- **The Objective:** To create a simple, user-friendly Desktop GUI application that allows users to convert amounts between different global currencies instantly.
- **Scope:** The project covers 18 major currencies and uses a defined set of exchange rates for demonstration purposes.

# Tools & Technologies

## Hardware Requirements

- Standard PC/Laptop
- Minimum 4GB RAM

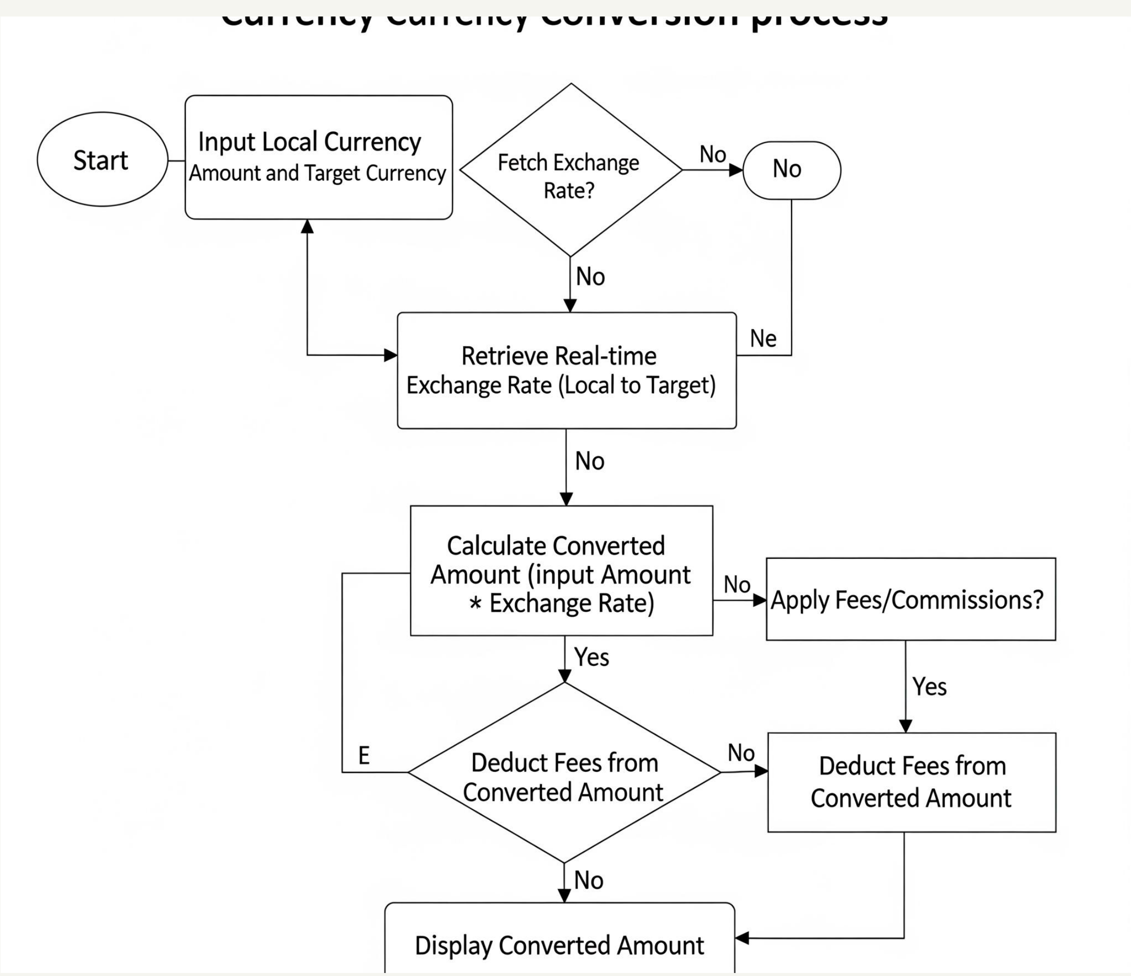
## Software Requirements

- **Language:** Python 3.x
- **IDE:** VS Code / PyCharm
- **Libraries:**
    - tkinter (For GUI)
    - ttk (For themed widgets)

Python

# Algorithm Flowchart

1. **Start** the application.
2. **Input** the amount in the entry field.
3. **Select** "From" currency and "To" currency.
4. **Process:** Fetch exchange rates from Dictionary.
5. **Calculate:** Apply conversion formula.
6. **Display** result on the Label.
7. **Stop.**


Currency Currency Conversion process
Start
Input Local Currency
Amount and Target Currency
Fetch Exchange Rate?
No
No
No
Retrieve Real-time Exchange Rate (Local to Target)
Ne
No
Calculate Converted Amount (input Amount * Exchange Rate)
No
Apply Fees/Commissions?
Yes
Yes
E
Deduct Fees from Converted Amount
No
Deduct Fees from Converted Amount
No
Display Converted Amount

# The Logic (Methodology)

We use **USD (United States Dollar)** as the base currency for all calculations to simplify the logic.

### Step 1: Convert to Base (USD)

```
Amount_in_USD = Input_Amount / From_Rate
```

### Step 2: Convert to Target

```
Final_Output = Amount_in_USD * To_Rate
```

*Example*: Converting 100 EUR to INR

(100 / 0.92) * 83.50 = 9076.08 INR

# GUI Implementation (Tkinter)

The interface is built using the standard tkinter library widgets:

| Widget | Purpose in Project |
|---|---|
| Tk() | Creates the main application window. |
| Entry() | Accepts user input (Amount). |
| Combobox() | Dropdown menu for selecting currencies. |
| Button() | Triggers the convert() function. |
| Label() | Displays static text and the final result. |

# Source Code: Conversion Function

```
def convert(): # 1. Fetch User Input amount = float(entry_amount.get()) from_curr = combo_from.get()
```

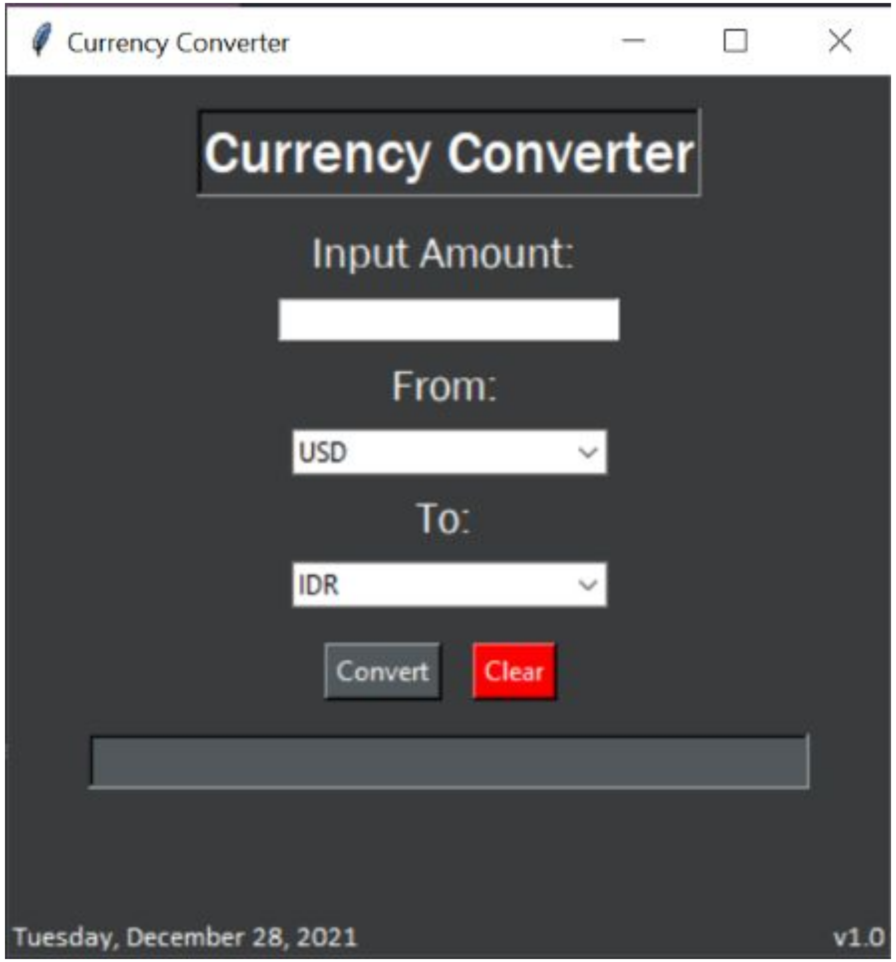
This function is bound to the "Convert" button command.

# Output Screen

## Successful Execution

The image shows the final application running on Windows.

- Window Title: "Simple Converter"
- Geometry: 250x250 pixels
- Theme: Default Tkinter System Theme



Currency Converter

Currency Converter

Input Amount:

From: USD

To: IDR

Tuesday, December 28, 2021

v1.0

# Limitations & Future Scope

## Current Limitations:

- Exchange rates are **static** (hardcoded) and do not update automatically.
- Basic error handling (program might crash if text is entered).

## Future Enhancements:

- **API Integration:** Fetch live rates using *exchangerate-api*.
- **History Feature:** Save previous conversions to a text file.
- **Reset Button:** To clear all inputs instantly.

# Conclusion

In conclusion, this project successfully demonstrates the use of Python for building GUI applications.

We learned how to handle user inputs, perform arithmetic operations using backend logic, and update the frontend dynamically.

This project serves as a strong foundation for understanding Event-Driven Programming.

## Thank You!

Questions?