DEPARTMENTOFELECTRONICSANDCOMMUNICATIONENGINEERING

PROJECTREPORT

ESDX001-PYTHON PROGRAMMING

B.Tech.,(Electronics and Computer Engineering)

2025

B.S. Abdur Rahman ™

Crescent

Institute of Science & Technology

Deemed to be University u/s 3 of the UGC Act, 1956

CURRENCY CALCULATOR USING PYTHON

Team Details:

SHAIK MOHAMMAD KHAJA GOUSE(230201601047)
SUHAIL.M (230201601054)
ABDUL HAMEED(230201601001)

Problem Statement:

Currency exchange rates fluctuate continuously, and manual calculation of exchange rates is prone to error and time-consuming. This project aims to create a Python-based currency converter that allows users to convert between different currencies accurately and quickly using real-time or static exchange rates.

Abstract:

The Currency Converter project is designed to convert one currency to another using real-time exchange rates. The project uses Python programming language and leverages an API to fetch the latest currency rates. The user inputs the source currency, target currency, and the amount to be converted. The program calculates the equivalent amount and displays the result. The project provides an easy-to-use interface and ensures high accuracy by using live market rates.

Objective:

- To develop a Python application that converts one currency to another using ap re defined exchange rate or an online API.
- To create a user-friendly interface for easy interaction.
- To ensure the accuracy and speed of currency conversion.

Component Selection &Feasibility

Components Used:

1. **Python Libraries:**
   - `tkinter`–For GUI development
   - `requests`–For fetching real-time exchange rates
   - `json`–For handling data responses
2. **API:**
   - Free exchange rate API (e.g.,exchangerate-api.com)
3. **Development Tools:**
   - Python3.x
   - IDE(e.g.,VSCode,PyCharm)

Feasibility Analysis:

- **Cost-effectiveness :** Free AP Iand open-source libraries
- **Availability:** Python and libraries are widely available
- **Power Consumption :** Minimalas it's a light weight desktop-based program.

Overall Working of the Project:

1. **Input:**
    - User inputs the amount to be converted.
    - User selects the source and target currencies.
2. **Processing:**
    - The programs ends a request to the currency exchange API.
    - The API returns the current exchange rate.
    - The program calculates the converted value using the formula:

      CONVERTEDAMOUNT=INPUTAMOUNT×EXCHANGERATE

3. **Output:**
    - The converted amount is displayed to the user.
    - Error handling is implemented for invalid inputs and connection issues.

Explanation:

1. **Importing Libraries:**
    - `Tkinter` for the graphical user interface.
    - `Requests` for API requests to get exchange rates.
2. **convert_currency ()Function:**
    - Takes input for currencies and amount.
    - Makes an API request to fetch exchange rates.
    - Converts the amount and displays the result.
3. **GUI Setup:**
    - `Label` and `Entry` widgets for input.
    - `Button` for triggering conversion.
    - `Label` to display the result.
4. **Error Handling:**
    - Displays an error message if the input is incorrect or API fails.

Project Plan &Timeline

Task	Duration	Status
Requirements Gathering	1week	Completed

Task	Duration	Status
Coding	2weeks	In Progress
Testing	1week	Pending
Documentation	1week	Pending

Source codes:

(PYTHON)

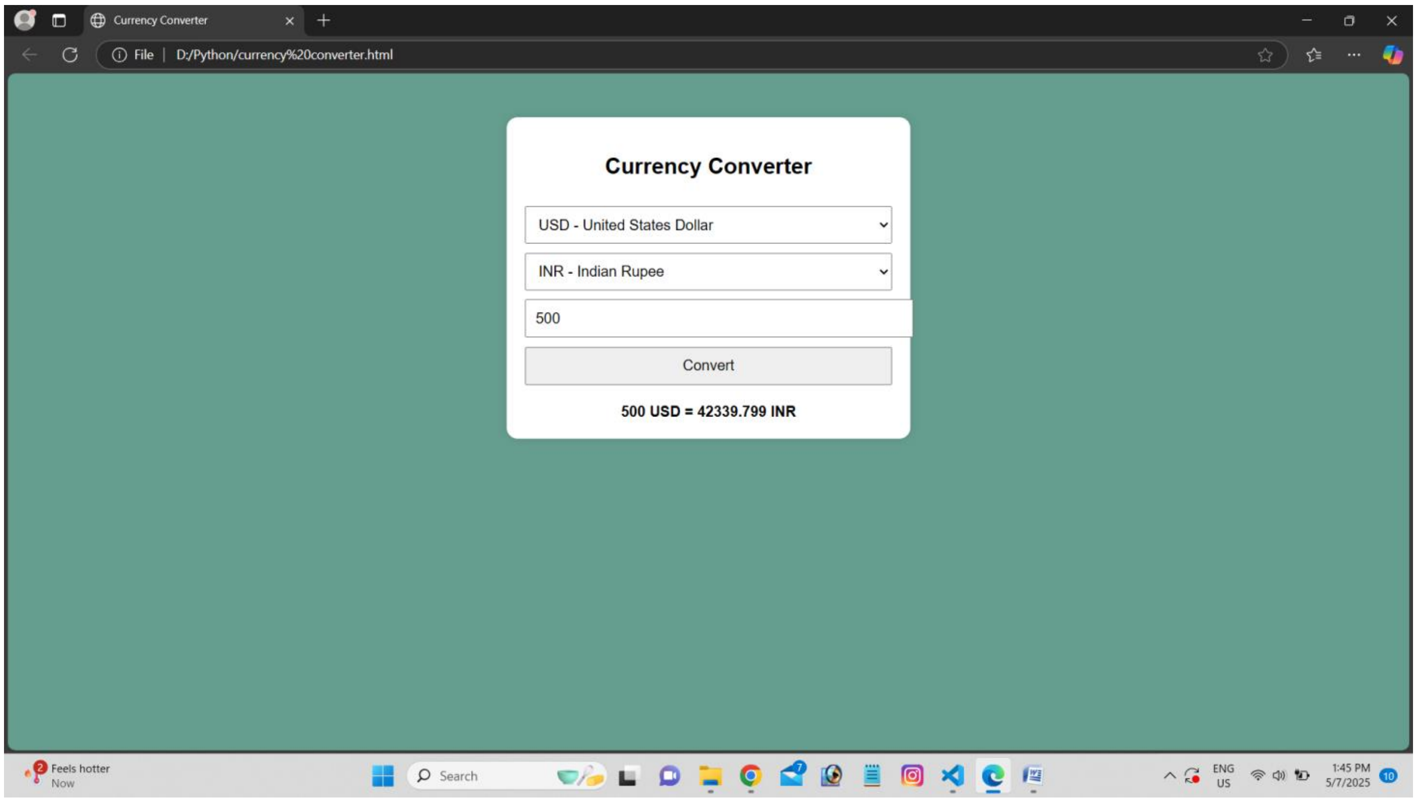
```python
import requests
import tkinter as tk
from tkinter import ttk, messagebox

def fetch_currencies():
    url = "https://open.er-api.com/v6/latest/USD"  # Free endpoint
    try:
        response = requests.get(url)
        data = response.json()

        # Debug output
        print("API response for currencies:", data)

        if 'rates' not in data:
            messagebox.showerror("API Error", "No 'rates' found in API response.")
            return []

        return list(data['rates'].keys())
    except Exception as e:
        messagebox.showerror("Connection Error", f"Unable to fetch currencies: {e}")
        return []

def convert_currency():
    from_curr = from_currency.get()
    to_curr = to_currency.get()
    amount_val = amount.get()

    if not from_curr or not to_curr or not amount_val:
        messagebox.showerror("Input Error", "Please fill in all fields.")
        return

    try:
        float_amount = float(amount_val)
    except ValueError:
        messagebox.showerror("Input Error", "Amount must be a number.")
        return

    url = f"https://open.er-api.com/v6/latest/{from_curr}"
    try:
        response = requests.get(url)
        data = response.json()

        print("Conversion data:", data)

        if 'rates' in data and to_curr in data['rates']:
            rate = data['rates'][to_curr]
            converted = float_amount * rate
            result_label.config(text=f"{amount_val} {from_curr} = {converted:.2f} {to_curr}")
        else:
            messagebox.showerror("Conversion Error", "Currency not found in exchange rates.")
    except Exception as e:
        messagebox.showerror("Error", f"Conversion failed: {e}")

# GUI setup
root = tk.Tk()
root.title("Currency Converter")
root.geometry("400x250")

currencies = fetch_currencies()
```

```
from_currency = ttk.Combobox(root, values=currencies)
from_currency.set("From Currency")
from_currency.pack(pady=10)

to_currency = ttk.Combobox(root, values=currencies)
to_currency.set("To Currency")
to_currency.pack(pady=10)

amount = tk.Entry(root)
amount.insert(0, "Enter amount")
amount.pack(pady=10)

convert_btn = tk.Button(root, text="Convert", command=convert_currency)
convert_btn.pack(pady=10)

result_label = tk.Label(root, text="")
result_label.pack(pady=10)

root.mainloop()
```

(HTML)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Currency Converter</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      text-align: center;
      padding: 40px;
      background: #659d8e;
    }
    .converter {
      background: #ffffff;
      padding: 20px;
      border-radius: 12px;
      box-shadow: 0 0 10px rgba(0,0,0,0.1);
      max-width: 400px;
      margin: auto;
    }
    select, input, button {
      width: 100%;
      padding: 10px;
      margin-top: 10px;
      font-size: 16px;
    }
    .result {
      margin-top: 20px;
      font-weight: bold;
    }
  </style>
</head>
<body>
  <div class="converter">
    <h2>Currency Converter</h2>
    <select id="from"></select>
    <select id="to"></select>
    <input type="number" id="amount" placeholder="Enter amount" />
    <button onclick="convert()">Convert</button>
    <div class="result" id="result">Result will appear here</div>
  </div>

  <script>
    const accessKey = 'K7uxuRwXO82sSTuJXbngHixcXRRjOKFM'; // Replace with your real API key
    const fromSelect = document.getElementById('from');
    const toSelect = document.getElementById('to');

    async function fetchCurrencies() {
      const res = await fetch(`https://api.apilayer.com/exchangerates_data/symbols`, {
        headers: { apikey: accessKey }
      });
      const data = await res.json();
      if (!data.symbols) {
        alert('Failed to fetch currency symbols');
        return;
      }

      for (let key in data.symbols) {
        const option1 = new Option(`${key} - ${data.symbols[key]}`, key);
        const option2 = new Option(`${key} - ${data.symbols[key]}`, key);
        fromSelect.add(option1);
        toSelect.add(option2);
      }

      fromSelect.value = 'USD';
      toSelect.value = 'EUR';
    }

    async function convert() {
      const from = fromSelect.value;
      const to = toSelect.value;
      const amount = document.getElementById('amount').value;

      const res = await fetch(`https://api.apilayer.com/exchangerates_data/convert?from=${from}&to=${to}&amount=${amount}`,
{
        headers: { apikey: accessKey }
      });
      const data = await res.json();
      document.getElementById('result').textContent = `${amount} ${from} = ${data.result} ${to}`;
    }

    fetchCurrencies();
  </script>
</body>
</html>
```

OUTPUT:



Currency Converter

USD - United States Dollar

INR - Indian Rupee

500

Convert

500 USD = 42339.799 INR

Summary

The **Currency Calculator Using Python** is a user-friendly application developed to simplify and automate the process of currency conversion. Built using Python and real-time exchange rate APIs, the system allows users to convert amounts between different currencies accurately and quickly. The project includes both a desktop version with a graphical user interface (using Tkinter) and a web version (using HTML, CSS, and JavaScript).

Users input the source and target currencies along with the amount, and the application fetches the latest exchange rates via an API to perform the conversion. The result is displayed instantly, and robust error handling ensures reliability even in the case of incorrect inputs or connection issues.

This project showcases real-time data integration, efficient GUI development, and practical problem-solving using open-source tools. Future improvements may include historical data tracking, offline support, and graphical trend analysis.

Improvements & Future Enhancements

To make the Currency Converter more robust, interactive, and widely usable, the following improvements are proposed for future versions:

- **Historical Exchange Rate Support:** Allow users to check past exchange rates and perform historical conversions.
- **Offline Mode:** Enable conversion using the most recently cached exchange rates when an internet connection is unavailable.
- **Graphical Data Visualization:** Integrate graphs to display currency trends and rate fluctuations over time.
- **Multi-Currency Conversion:** Support simultaneous conversion to multiple currencies.
- **Dark Mode & Theme Customization:** Enhance user experience with customizable UI themes.
- **Mobile App Version:** Develop a responsive mobile application version using frameworks like Kivy or Flutter.
- **Voice Input Integration:** Add voice command functionality for better accessibility.

References & Acknowledgments

References:

- https://www.exchangerate-api.com – For real-time exchange rate API used in the project.
- https://apilayer.com – Alternative API service used for currency symbols and conversion data.
- Python Official Documentation – https://docs.python.org/3/
- Tkinter Documentation – https://docs.python.org/3/library/tkinter.html
- W3Schools – For guidance on HTML, CSS, and JavaScript used in the web version.

Acknowledgments:

We extend our sincere thanks to:

- Our faculty **RAMESH SIR** for their guidance and support.
- The developers and maintainers of the open-source APIs and Python libraries used.
- Online coding communities such as Stack Overflow for troubleshooting assistance.
- Our teammates for their dedicated collaboration:
  - **SHAIK MOHAMMAD KHAJA GOUSE**
  - **SUHAIL M**
  - **ABDUL HAMEED**

Conclusion

The currency converter application effectively converts between different currencies using real-time exchange rates. The project showcases the integration of GUI with real-time API data retrieval ,ensuring accurate and fast conversions . Future improvements can make the application more robust and feature-rich.