

CURRENCY CONVERTER USING PYTHON

Project submitted to the
SRM University – AP, Andhra Pradesh
for the partial fulfillment of the requirements to award the degree
of

Bachelor of Technology

In

**Computer Science and Engineering
School of Engineering and Sciences**

Submitted by

Aman Kumar Thakur (AP23110011682)

Bibek Kumar Sah (AP23110011548)

Sameer Bachhar (AP23110011496)

Bikash Kumar Mahato (AP23110011681)



Under the Guidance of

Ms. V. Veda Sri

Lecturer, Department of CSE

SRM University-AP

Neerukonda, Mangalagiri, Guntur

Andhra Pradesh – 522 240

APRIL 2025

Certificate

Date: 04-2025

This is to certify that the work present in this Project entitled **“CURRENCY CONVERTER USING PYTHON”** has been carried out by **Aman Kumar Thakur (AP23110011682), Bibek Kumar Sah (AP23110011548), Sameer Bachhar (AP23110011496), Bikash Kumar Mahato (AP23110011681)**, under my/our supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor of Technology/Master of Technology in **School of Engineering and Sciences**.

Supervisor

(Signature)

Prof. / Dr. Karnena Kavitha Rani

Designation,

Affiliation.

Acknowledgement

The satisfaction that accompanies the successful completion of any task would be incomplete without introducing the people who made it possible and whose constant guidance and encouragement crowns all efforts with success.

I am extremely grateful and express my profound gratitude and indebtedness to my project guide, **Dr. Karnena Kavitha Rani**, Department of Computer Science & Engineering, SRM University, Andhra pradesh, for her kind help and for giving me the necessary guidance and valuable suggestions in completing this project work.

1. Table of Contents

Certificate	i
Acknowledgements	ii
Table of Contents	iii
Abstract	iv
1. Introduction	1-2
1.1 Objective	
1.1.1 Scope	
2. Methodology	3
2.1 Software Requirements Specifications	
2.1.1 Tools & Development Approach	
3. Implementation	4
4. Result and Analysis	5
5. Discussion and Conclusion	6
6. Future Work	7-8
7. References	9

Abstract

The Currency Converter project focuses on building a user-friendly and visually appealing desktop application that enables users to convert monetary values between a wide range of international currencies. Developed using Python, the application leverages a modern GUI framework (customtkinter) to offer an interactive and clean user experience.

To ensure accurate and up-to-date conversions, the system integrates with a real-time currency exchange rate API. Users can input an amount, select source and target currencies, and receive immediate conversion results. The application includes features like input validation, theme customization, and a responsive layout, making it suitable for both novice users and professionals.

This tool is particularly useful for travelers, financial analysts, students, and individuals managing international transactions. Its lightweight nature allows it to run smoothly on local machines without the need for extensive system resources or web-based platforms. In addition, the modular code structure ensures that the application is customizable and extensible for future enhancements such as historical data tracking, offline mode, or graphical analysis.

Overall, the project demonstrates how desktop applications can provide a practical solution for global financial tasks in a simple, effective, and elegant manner.

1. Introduction

Currency exchange has become a vital aspect of day-to-day life in our increasingly interconnected global economy. With the rise of globalization, international tourism, remote work, and cross-border e-commerce, there is a growing need for a tool that can convert currencies accurately and efficiently.

This project presents a simple yet powerful **Currency Converter** application developed using Python. The tool enables users to convert amounts between various global currencies using **real-time exchange rates** fetched from a public API. The program combines clean design with accuracy, making it highly suitable for casual users and professionals alike.

The graphical user interface (GUI) is designed using the (*customtkinter*) library, providing a modern and responsive user experience. The tool is lightweight, easy to install, and capable of functioning smoothly on Windows and Linux platforms.

This converter is particularly useful for:

- **Tourists** managing foreign expenses
- **Business analysts** evaluating multi-currency transactions
- **Students and learners** understanding currency dynamics

- **Freelancers and remote workers** receiving payments in different currencies

1.1 Objective

The main objective of this project is to design and develop a desktop-based crosscurrency calculator that:

- **Is fast and reliable.**
- **Utilizes real-time exchange rate APIs.**
- **Offers a clean, intuitive, and user-friendly interface.**
- **Can be extended or integrated into other financial or budgeting tools.**

1.1.1 Scope

The application is specifically designed for **desktop environments** (Windows/Linux).

It can be integrated into:

- Personal budgeting software
- Finance management tools
- Educational or business-use dashboards

2. Methodology

The project development followed these steps:

- Selection of Python for its robust libraries and UI capabilities.
- Use of customtkinter for modern GUI design.
- API integration for real-time exchange rates.
- Implementation of error handling and input validations.

2.1 Software Requirements Specifications:

- **Language:** Python 3.x
- **Libraries:**tkinter/customtkinter, requests
- **Platform:** Windows
- **API Used:** Exchange rate API (exchangerate-api.com)

2.1.1 Tools & Development Approach

- **IDEs Used:** VS Code / PyCharm
- **Design Pattern:** Modular scripting for separation of GUI and logic
- **Testing:** Manual testing for GUI, API calls, and input validations

3. Implementation

The project is structured as follows:

- **main.py:** Main file handling UI and logic
- **config.py:** Stores supported currencies
- **requirements.txt:** Lists dependencies

Design Flow:

1. User inputs amount and selects currencies.
2. On clicking convert, the app fetches exchange rate.
3. Result is displayed in the output field.

Sample Code Snippet:

```
1 import requests
2 url = f"https://api.exchangerate-api.com/v4/latest/{from_currency}"
3 data = requests.get(url).json()
4 rate = data['rates'][to_currency]
5 converted_amount = amount * rate
```


4. Result and Analysis

Input Currency	Output Currency	Input Amount	Converted Amount
USD	INR	1	85.48
EUR	JPY	1	160.95

Screenshot:



The screenshot shows a web application titled "Currency Converter" with a dark blue background. The "From:" dropdown is set to "USD - United States Dollar" and the "To:" dropdown is set to "INR - Indian Rupee". The "Amount:" input field contains the value "1". A blue "Convert" button is visible. Below the button, the result is displayed in yellow text: "1.0 USD = 85.48 INR".



The screenshot shows the same "Currency Converter" application. The "From:" dropdown is now set to "EUR - Euro" and the "To:" dropdown is set to "JPY - Japanese Yen". The "Amount:" input field still contains "1". The blue "Convert" button is present. The result is displayed in yellow text: "1.0 EUR = 160.95 JPY".

5. Discussion and Conclusion

The development of the Currency Converter project successfully met its initial objectives. The application proved to be **efficient, userfriendly, and practical**, especially for users who need quick and accurate currency conversions.

The use of **Python** and the *customtkinter* library enabled the creation of a modern and responsive GUI. The integration of a real-time exchange rate API allowed the application to fetch the most recent conversion values, making the tool reliable and up to date. The lightweight nature of the project ensures it runs smoothly on most desktop environments without the need for complex dependencies or installations.

Key Achievements:

- A clean and intuitive GUI for seamless user interaction.
- Real-time conversion powered by API integration.
- Compatibility across major operating systems (Windows/Linux).
- A modular and easily extendable codebase for future enhancements.

Conclusion

This project demonstrates how Python can be used to build a functional desktop application with real-world use cases. With a few improvements – such as offline support, graphical features, and extended analytics – it can evolve into a more powerful financial utility.

6.Future Work

While the current version of the Currency Converter application meets its core objectives, there is ample scope for enhancement and future development. The following features can be considered to make the tool more powerful, flexible, and usercentric:

- **Historical Exchange Rate Analysis**
Future versions of the application could support historical data lookup, allowing users to view and analyze currency trends over specific time periods. This would be particularly useful for financial analysts, traders, and researchers who need to evaluate past currency behavior.
- **Graphing and Plotting Tools**
By integrating graphing libraries such as matplotlib or plotly, the app can visually represent exchange rate trends and comparisons. This would enhance the analytical capabilities of the tool and provide better insights to users through interactive charts and graphs.
- **Dark/Light Theme Support**
Adding multiple UI themes such as dark mode and light mode would significantly improve the user experience. This feature would not only make the application visually appealing but also provide comfort during extended usage, especially in low-light environments.
- **Offline Mode with Cached Data**
Incorporating offline functionality using cached exchange

rate data would ensure usability even when the user lacks an internet connection. The system could store the most recent exchange rates locally and use them for conversions during offline sessions, notifying the user about the last sync time.

- **Multi-language Support**

To make the tool accessible to a broader audience, multi-language support can be added. This would enable users from different regions to interact with the app in their native language.

- **Mobile App Integration**

Expanding the application to mobile platforms (Android/iOS) would make it more versatile and accessible for users who travel frequently or need to convert currencies on the go.

7. References

1. **exchangerate-api.com** - API for currency exchange rates.
2. **PythonOfficialDocumentation-**
<https://docs.python.org>
3. **CustomTkinterGitHubRepository-**
<https://github.com/TomSchimansky/CustomTkinter>