PROJECT ON

Currency Converter

(WITH AI PREDITOR)

**BACHELOR OF TECHNOLOGY**

 (Artificial Intelligence and Machine Learning)

**SUBMITTED BY:**

Palak (2420574)

Payal Mishra (2420577)

Riya (2420593)

**Under the guidance of**

Mr. Rajiv Kumar

**Department of Computer Science and Engineering**

**CHANDIGARH ENGINEERING COLLEGE JHANJERI, MOHALI**

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Sno.** | **Contents** | **Pages** |
| **1.** | **Introduction** | **1** |
| **2.** | **Brief Literature Survey** | **2** |
| **3.** | **Problem formulation** | **3-4** |
| **4.** | **Objective** | **5** |
| **5.** | **Methodology/Planning of work** | **6** |
| **6.** | **Facilities required for proposed work** | **7** |
| **7.** | **References** | **8** |
| **9.** | **Conclusion** | **10** |

**Introduction**

In the era of globalization, international transactions, travel, and business have become part of everyday life, making currency conversion a necessary and recurring activity for individuals, students, tourists, and enterprises. The frequent necessity to convert one currency to another—whether for shopping online through international merchants or for planning travel budgets—demands fast, accurate, and user-friendly solutions. Fluctuations in global currency markets can significantly impact the cost of goods, services, and investments, so the ability to access real-time exchange rates becomes indispensable .

Traditionally, users have depended on financial news sites, banks, or dedicated web applications to fetch exchange rates and manually perform conversions. While such sources provide up-to-date information, they may have usability limitations, lack advanced features, or require multiple manual steps which can lead to errors or inefficiencies. Furthermore, these solutions often do not offer any insight or guidance about how currency values might change in the near future, leaving users without support for planning purposes. As a result, the need for smart, desktop-based, and highly interactive applications has grown—especially in the context of educational environments where learning and experimentation are equally important alongside practical usage .

The project "Currency Converter with AI Prediction" aims to bridge these gaps by providing a Python-based desktop application that integrates real-time currency conversion with an AI-powered next-day rate prediction feature. This tool is specifically designed to combine ease of use, accuracy, and a learning component for users who want both timely conversions and an introduction to artificial intelligence applications in finance .

The application's backbone is a reliable online API (Frankfurter), which delivers up-to-the-minute currency exchange rates from trusted sources. Users simply select the source and target currencies, enter their amount, and receive instant conversion results. The application supports a variety of the world's most widely used currencies, including USD, INR, EUR, GBP, JPY, CAD, and AUD, making it versatile and relevant for a broad audience .

A distinctive feature of this project is the AI-based prediction of the following day's exchange rate. While financial forecasting itself can be exceedingly complex and may involve advanced data analytics, time series modelling , and deep learning, this application demonstrates a simplified approach using simulated AI logic to introduce users to the principles of prediction and uncertainty in currency markets. The core objective is to not only serve the practical needs of users but also to provide educational value by showing how algorithms can estimate future exchange rates and why such predictions require robust data and techniques .

The software further distinguishes itself through its clean, intuitive graphical user interface, implemented using Python's Tkinter library. The UI is designed to be modern and responsive, allowing users of all skill levels to perform conversions, view predictions, and understand the workflow at a glance. Comprehensive error handling ensures that users receive clear notifications in the case of connectivity issues, invalid inputs, or unavailable API services .

Beyond its utility for end-users, the project offers a strong academic component with opportunities for future expansion. Students and educators can use the code to understand API integration, GUI development, exception handling, and foundational concepts in AI-based forecasting. Developers may further extend the software to support tax or transaction fee breakdowns, advanced machine learning for improved prediction accuracy, or integration with data visualization tools for currency trend analysis.

The project thus serves as both a real-world tool and a foundation for continued learning and innovation .In conclusion, the "Currency Converter with AI Prediction" project is a comprehensive, practical, and educational solution for modern currency conversion needs. By merging the immediacy of live data with the forward-looking aspect of AI simulation, it equips users with a powerful tool for everyday use while fostering better understanding and experimentation with artificial intelligence in financial contexts.

**Brief Literature Survey**

The need for accurate currency conversion has grown significantly in the context of expanding globalization, international trade, travel, and borderless e-commerce. Over the past decades, the approach to currency exchange has evolved from manual conversion tables and bank-driven rates to advanced, instant, internet-enabled software utilities. The academic and technical literature on currency conversion tools, exchange rate prediction, and the application of artificial intelligence in finance is both rich and diverse.

**[2.1]Evolution of Currency Conversion Technology**

In the early years, foreign exchange (forex) conversion was handled through fixed printed tables distributed by banks and money changers. Users referred to these tables for basic cross-currency conversion using fixed rates. Fluctuations in daily forex rates required regular table updates, often causing inconvenience and inaccuracies for end-users, especially travelers and importers/exporters. With the rise of computers and basic programming, the first generation of software tools—typically spreadsheets and desktop calculators—enabled more flexible, user-defined conversions but still relied on manually updated rate lists .

The advent of the internet brought a radical transformation. It became feasible to fetch near-real-time exchange rates directly from financial data providers, such as XE, OANDA, Bloomberg, and Yahoo Finance. APIs (Application Programming Interfaces) offered by these providers enabled developers to automatically integrate current rates into their software, web pages, or mobile apps. This increased both the accuracy and speed with which users could convert currencies for personal and business purposes.

**[2.2] Desktop, Web & Mobile Currency Converter Solutions**

In the early 2000s, many major financial portals launched online currency converters through web interfaces or APIs. Users could access websites to select source and target currencies, enter the desired amount, and receive instant results. With the spread of smartphones, currency converter mobile apps also became popular, often bundling offline rate storage, graphs, and batch conversion features.

Desktop applications, while effective, were relatively fewer and often lacked the seamless interface and up-to-the-minute accuracy now made possible by internet integration. Notably, software such as "Currency", "Universal Currency Converter" and various Excel add-ons were used; however, they often suffered from complex interfaces or manual update requirements.

**[2.3] Live Data APIs & Python-Based Solutions**

Python, due to its simplicity and strong ecosystem, has become a popular choice for both academic and commercial projects involving financial data. APIs such as Frankfurter, Open Exchange Rates, and Forex Python have made it trivial to obtain live rates for all major world currencies. Python also offers robust libraries for data handling, GUI development, and error management.

**Problem formulation**

Currency conversion is a fundamental need for millions of people worldwide involved in cross-border financial activities such as trade, tourism, education, and investment. While several online platforms and mobile applications offer currency conversion services by fetching live data from financial agencies, there remains a real-world need for a simple, user-friendly desktop application that can serve users who prefer offline accessibility combined with occasional live updates.

The traditional methods of currency conversion—manual look-ups, static tables, or visiting websites—are inefficient and prone to errors. Users often get overwhelmed by complex interfaces or a lack of consolidated features, such as multi-currency support or predictive insight into future currency trends. Moreover, real-time rates are highly dynamic due to geopolitical and economic factors, necessitating an application that adjusts seamlessly to changing data without demanding technical intervention from users.

Another major challenge is the absence of accessible forecasting tools for smaller-scale users. Advanced financial analysts and banks use sophisticated statistical and AI-powered models to predict currency movements, but these are not usually integrated into consumer-friendly applications. Hence, there is a gap in providing lightweight, understandable, and demonstrative AI prediction features within a simple currency converter app, which could serve both educational and practical purposes.

The problem can thus be stated as:

How to effectively combine real-time currency conversion with AI-driven next-day rate prediction in a single, comprehensive, and easy-to-use desktop application?

How to design a Graphical User Interface (GUI) that allows for straightforward input, clear outputs, error handling, and multi-currency support without complicating the user experience?

How to simulate predictive analytics in currency conversion without relying on large-scale datasets or complex machine learning infrastructures, making AI accessible and comprehensible for learners?

The proposed solution must be lightweight, reliable, and capable of running on common personal computers or laptops. It must fetch live currency data through an API, accurately perform conversions, and produce intuitive predictions based on simplified AI logic.

Overall, the project aims to bridge the gap between practical currency conversion tools and educational AI experiments, enabling users to conveniently perform conversions and gain preliminary insights into exchange rate forecasts. This formulation guides the development path, encompassing API integration, UI/UX design, and AI simulation models.

**Objective**

1.To develop a Python-based desktop application that performs real-time currency conversion using live exchange rates from an online API.

2.To provide support for multiple major currencies including USD, INR, EUR, GBP, JPY, CAD, and AUD to enhance usability.

3.To build an interactive and user-friendly Graphical User Interface (GUI) using Python’s

library.

4.To integrate a simplified AI module that predicts the next day’s currency exchange rate for educational and demonstrative purposes.

5.To ensure robust error handling and input validation for smooth, error-free user experience.

6.To present clear, immediate output of conversion results and AI predictions within the GUI.

7.To create a scalable foundation that can be extended in the future for features such as detailed tax breakdowns, charts, and advanced machine learning predictions.

**Methodology / Working Principle**

1.Requirements Gathering

* Understand user requirements: multilingual support, real-time conversion, AI prediction.
* Gather technological specifications: Python 3.x, Tkinter for GUI, requests library for API access.

2. Design Phase

* Design the GUI layout: dropdowns for 'From' and 'To' currencies, input box for amount, buttons for 'Convert' and 'Predict Rate'.
* Develop the flowcharts for user interaction: input → process → output.
* Sketch the user interface (UI) elements visually for clarity.

3. API Integration

* Select a reliable API for real-time exchange rates, e.g., Frankfurter or Open Exchange Rates.
* Implement API request function using requests library to fetch latest currency exchange data.
* Handle network errors, invalid responses, or unavailable API gracefully.

4. GUI Development

* Build the interface using Tkinter:
* Dropdown menus for currency options.
* Entry box for the amount.
* Buttons for conversion and prediction.
* Labels for displaying results.
* Add input validation to check:
* Numeric validity of amount.
* Selection of currencies.
* Display error messages or prompts if inputs are invalid or connection issues.

5. Conversion Functionality

* On pressing 'Convert', trigger a function:
* Fetch the latest rate from API.
* Calculate the converted amount.
* Display the result in the GUI.
* Update the UI dynamically for immediate feedback.

6. AI Prediction Module

* Implement a simple AI simulation for future currency rates:
* Use random variation around the current rate.
* For more precise models, prepare historical rate data and train predictive models (optional).
* On pressing 'Predict Rate', generate and display the predicted value.
* Show the prediction in an intuitive format, e.g., "Next day’s predicted rate: X".

7. Error Handling & Validation

* Wrap API request calls inside try/except blocks:
* Handle connection timeouts, failures, and invalid responses.
* Validate user inputs:
* Non-numeric amounts.
* Empty fields.
* Unsupported currency selections.
* Show appropriate error messages or prompts to guide the user.

8. Testing and Validation

* Conduct unit testing to verify individual functions.
* Perform integration testing:
* Check the entire user flow.
* Test under various conditions: network failure, invalid inputs.
* Collect user feedback and optimize the interface and functionalities.

9. Documentation & Deployment

* Document key code sections and user instructions.
* Prepare a user manual describing how to run and use the program.
* Package the application into an executable if desired.

**Facilities required for purposed work**

1.Hardware:

A personal computer or laptop with a minimum of 4GB RAM and adequate storage to support Python development and execution.

Internet connectivity to fetch live currency rates from online APIs during runtime.

2.Software:

Python 3.x installed on the system.

Python libraries such as tkinter

A text editor or IDE (like VSCode, PyCharm, or even Notepad++) to develop the project code.

3.API Access:

Access to a free or paid currency exchange rate API such as Frankfurter API, which provides up-to-date financial data necessary for live currency conversion.

4.Operating System:

Windows, Linux, or macOS to run Python and associated libraries without compatibility issues. 5.Development Environment:

Stable internet connection for downloading APIs and libraries. Proper Python environment setup for package management.

6.Testing Facilities:

Ability to test the application on different network conditions and multiple currency conversion scenarios.

Tools or manual verification for cross-checking live rates with trusted financial websites.

**References**

1.Frankfurter API Documentation:

https://www.frankfurter.app

Used for real-time currency exchange rate data retrieval via HTTP requests.

2.DataFlair:

"Currency Converter Python Project Tutorial"

A detailed tutorial showing Python Tkinter GUI and API integration for currency conversion.

3.Python Official Documentation:

Tkinter module (for GUI creation)Requests module (for HTTP requests)

4.GeeksForGeeks:

"Currency Converter in Python."

A practical guide for building currency converters with live rates in Python.

**Conclusion**

The "Currency Converter with AI Prediction" project successfully addresses the need for a user-friendly desktop application that combines real-time currency conversion with AI-based forecasting features. By integrating live exchange rate data from reliable online sources and using Python's Tkinter library to build an interactive GUI, the project provides an accessible and efficient solution for users requiring prompt currency calculations.

The additional AI prediction functionality, albeit simplified for demonstration, introduces the concept of forecasting future currency values, exposing users to practical applications of artificial intelligence in financial domains. This layer of functionality not only enhances user experience but also serves as a learning tool for students and enthusiasts aiming to understand predictive modeling.

The project underscores the value of integrating data APIs, GUI development, and machine learning simulation into cohesive software, capable of being further extended with advanced forecasting, tax calculations, and graphical data representation. Overall, this tool contributes significantly to both practical currency conversion needs and educational pursuits in AI, marking a stepping stone for future innovations in personal and professional finance technologies.