**Currency Converter Using AI**

**INTRODUCTION:**

The **Currency Converter Using AI** is a software application that allows users to convert between different currencies using real-time exchange rates. It integrates **API-based real-time exchange rate retrieval** and employs **machine learning (Linear Regression)** to predict future currency exchange rates. Implemented in **Python**, this project provides an accurate and user-friendly way to handle currency conversion efficiently. The system's robust architecture ensures reliability, making it suitable for financial forecasting, e-commerce, and international transactions.

**SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC)**

The development of the **Currency Converter Using AI** follows the **Software Development Life Cycle (SDLC)** framework, ensuring a structured and efficient process through the following phases:

**1. Planning**

The goal of the project was defined as designing a **currency converter** with real-time exchange rates and AI-powered prediction. The feasibility, scope, and expected outcomes were determined, ensuring the project's alignment with financial forecasting needs.

**2. Defining Requirements**

The software and hardware requirements were established, including:

* **Operating System:** Windows/Linux/macOS
* **Programming Language:** Python
* **Development Environment:** Jupyter Notebook / Visual Studio Code
* **Libraries Used:** requests, pandas, scikit-learn, matplotlib, tkinter (for GUI)
* **API Service:** Currency exchange rate provider (e.g., Open Exchange Rates, Forex API)

**3. Designing**

The software was designed to handle real-time exchange rates and predictive modeling. Key design aspects included:

* **API Integration:** Fetching live exchange rates.
* **Machine Learning Model:** Linear Regression for predicting future rates.
* **User Interface:** A simple GUI for user-friendly interaction.
* **Data Storage:** SQLite for storing historical exchange rates.

**4. Building (Implementation)**

The program was implemented using Python, integrating real-time API calls and machine learning. Key components included:

* **Fetching live exchange rates via API.**
* **Storing past exchange rates for ML training.**
* **Training a Linear Regression model for future predictions.**
* **Developing a simple GUI using Tkinter.**

**5. Testing**

The software underwent rigorous testing, including:

* **Unit Testing:** Checking API response handling and currency conversion accuracy.
* **Edge Case Testing:** Testing invalid inputs, unsupported currencies, and API failures.
* **Performance Testing:** Evaluating response time and prediction accuracy.

**6. Deployment**

Once tested, the program was deployed for use. Future deployment considerations include integrating the system into mobile apps and cloud platforms for wider accessibility.

**PROCEDURE AND METHODS USED:**

1. **Input Representation:** User selects input and target currency.
2. **API Integration:** Fetch real-time exchange rates from an API.
3. **Currency Conversion:** Convert input currency using fetched exchange rates.
4. **Machine Learning Prediction:** Train a Linear Regression model on past exchange rates to predict future trends.
5. **Output Representation:** Display converted amount and future predictions.

**ALGORITHM:**

1. Start the program.
2. Fetch live exchange rates using API.
3. Take user input for the source and target currency.
4. Convert the currency using the fetched exchange rate.
5. Store exchange rate data for machine learning training.
6. Train a **Linear Regression model** on past exchange rate trends.
7. Predict future exchange rates based on the trained model.
8. Display the converted amount and future exchange rate predictions.
9. End the program.

**FUTURE SCOPE:**

* **Enhanced Prediction Models:** Use advanced ML models like LSTMs for better accuracy.
* **Multi-Currency Support:** Extend to cryptocurrencies and commodities.
* **Mobile Application Integration:** Develop Android/iOS applications.
* **Cloud Integration:** Deploy on cloud platforms for global access.

**FUTURE ENHANCEMENTS:**

* **Deep Learning Integration:** Improve accuracy using neural networks.
* **Historical Trend Analysis:** Provide graphical representations of currency fluctuations.
* **Voice Input Feature:** Enable hands-free currency conversion using voice commands.

**ADVANTAGES:**

* Provides real-time exchange rate conversions.
* Predicts future exchange rates for informed decision-making.
* User-friendly interface with cross-platform compatibility.
* Supports financial analysis and investment planning.

**REFERENCES:**

* API Documentation of Open Exchange Rates, Forex API.
* Linear Regression concepts in Machine Learning.
* Python requests and Tkinter library documentation.