******

**School of Mechanical & Manufacturing Engineering (SMME),**

**National University of Science and Technology (NUST),**

**Sector H-12, Islamabad**

|  |
| --- |
| **Engineering Thermodynamics LAB**  Program: BE-Aerospace Section: AE-01  Session: Fall 2024 Semester: 3rd |

**PROJECT REPORT**

|  |  |  |
| --- | --- | --- |
| **NAME** | **CMS ID** |  |
| Ali Tawasal | 455925 |  |

AEROSPACE ENGINEERING | SMME

GROUP: 2

**Abstract**

The Currency Converter Application is a user-friendly software tool developed using Python's tkinter library. It allows users to convert amounts between different currencies based on predefined exchange rates, record transaction history, and update conversion rates dynamically. This application aims to provide an intuitive and efficient solution for currency conversion needs without relying on external resources or complex calculations.

**Objective**

The primary objective of this project is to develop a graphical user interface (GUI) application that simplifies currency conversion. The application should allow users to:

1. Convert amounts between various currencies.
2. View a history of past transactions.
3. Update or add new currency conversion rates dynamically.
4. Enhance user interaction through a simple and effective interface.

**Introduction**

Currency conversion is an essential tool in a globally connected world. Whether for business, travel, or personal finance, individuals frequently need to convert one currency into another. Manual calculations are prone to errors, and real-time converters often require internet connectivity.

This project leverages Python to create a standalone, offline application that provides currency conversion functionalities. It incorporates an intuitive GUI using thinker, making the tool accessible to users with minimal technical expertise.

**Theory**

The project is based on the following theoretical concepts:

1. **Currency Conversion**: Currency conversion involves multiplying an amount in a source currency by a conversion rate to obtain its equivalent in a target currency.

Converted Amount=Amount×Conversion Rate

1. **Graphical User Interface (GUI)**: A GUI provides a visual platform for users to interact with software. It simplifies complex operations into intuitive tasks using buttons, input fields, and labels.
2. **Data Management**: The application stores predefined conversion rates and transaction history in memory. It provides methods for accessing and updating this data dynamically during runtime.

**Explanation of Project Components**

**1. Application Design**

The Currency Converter consists of three main components:

* **RateRecorder**: Manages currency conversion rates.
* **HistoryRecorder**: Keeps a record of past currency conversion transactions.
* **CurrencyConverterApp**: The main GUI component that connects user inputs to business logic.

**2. Implementation**

**RateRecorder Class**

* Handles predefined exchange rates.
* Allows querying and updating conversion rates for different currency pairs.

**HistoryRecorder Class**

* Records each transaction, including the amount, source currency, target currency, and the converted amount.
* Provides a method to retrieve and display transaction history.

**CurrencyConverterApp Class**

* Implements the main user interface.
* Provides input fields for amount, source currency, and target currency.
* Offers buttons for converting currencies, viewing transaction history, and updating rates.
* Uses thinker message boxes to display errors, results, and history.

**3. Application Workflow**

1. **Currency Conversion**:
   * The user enters the amount, source currency, and target currency.
   * The convert\_currency method retrieves the conversion rate and computes the converted amount.
   * The result is displayed in the GUI, and the transaction is recorded in history.
2. **View History**:
   * The user can click the "View History" button to see a record of past transactions.
   * This history is displayed in a message box.
3. **Update Rates**:
   * Users can add or modify conversion rates using the "Update Rates" button.
   * A new window allows the user to specify the source and target currencies and input a new conversion rate.

**4. Error Handling**

* The application includes robust error handling to manage invalid inputs, such as non-numeric amounts or unsupported currency pairs.
* Appropriate error messages are displayed via message boxes.

**Results and Discussion**

The application successfully meets its objectives by providing a seamless currency conversion experience. Users can easily convert currencies, view transaction history, and update rates. The GUI is intuitive, making it accessible to users without a technical background.

The project demonstrates the effective use of Python for building standalone applications. The modular design ensures that each component operates independently, which improves maintainability and scalability.

**Conclusion**

The Currency Converter Application is a practical tool for offline currency conversion. It highlights the potential of Python and thinker in developing lightweight, user-friendly applications. Future enhancements could include real-time exchange rate fetching, multi-language support, and improved transaction history visualization.