INTERNSHIP PROJECT REPORT

CURRENCY CONVERTER APPLICATION

PROJECT - I

Submitted by

ANUPAM KANOONGO (TEAM LEAD)
SUMAN M
SHAHANA NARGEES

To

INFOTACT SOLUTIONS

(Electronics City Phase 1, Electronic City, Bengaluru, Karnataka 560100)

JANUARY 2025

ABSTRACT

CURRENCY CONVERTER APPLICATION

This project is designed to develop a comprehensive desktop application for currency conversion that prioritizes both functionality and user experience. The application uses real-time exchange rates, fetched through an API, to provide accurate and instantaneous currency conversions. To enhance security, the app integrates a secure user authentication system, ensuring that all passwords are encrypted for safe login. Additionally, it incorporates interactive data visualization through matplotlib, offering users valuable insights into historical and current currency trends. Built with Python and essential libraries such as Tkinter for the graphical user interface (GUI), requests for API integration, and matplotlib for visual representation, the application combines convenience, security, and data analysis in a single platform, making it a useful tool for individuals and businesses dealing with currency exchange.

TABLE OF CONTENTS

1.INTRODUCTION

1.1 PROBLEM STATEMENT CONTEXT

The need for efficient currency management is a common challenge for individuals and businesses engaging in global transactions. Existing solutions often require users to rely on multiple platforms to access real-time exchange rates and analyze currency trends, leading to inefficiency and inconvenience. Moreover, many tools lack a secure and user-friendly interface, making the process cumbersome and less reliable. The Currency Converter Application addresses these issues by providing a single, secure desktop platform that combines real-time currency conversion, historical trend analysis, and an interactive user experience, simplifying currency management and empowering users to make informed financial decisions.

1.2 OBJECTIVES

- Develop a real-time currency converter that integrates live exchange rate APIs to provide accurate and instantaneous currency conversion for the users.
- Implement secure user authentication using encrypted passwords and advanced security protocols to ensure the privacy and safety of user data.
- Incorporate data visualization features to display historical and current currency trends, enabling users to analyze the patterns and make well-informed financial decisions.
- Design a modern, responsive desktop GUI with intuitive navigation and aesthetically appealing styling to enhance user experience and accessibility across different devices.

2. TECHNOLOGIES USED

1. **PYTHON - Python with Tkinter for GUI development** to create an interactive and user-friendly desktop interface, allowing users to easily navigate and perform currency conversions.

2. **APIs - Requests library for API integration** to fetch live exchange rates from the Exchange Rate API, ensuring real-time accuracy for currency conversions.

3. **SQLITE - SQLite database** to securely store user credentials and maintain a history of currency conversions for future reference and analysis.

4. **PIL - Pil (Pillow) for image processing** to incorporate visually appealing icons and graphics, enhancing the overall user experience of the application.

5. **MATPLOTLIB - Matplotlib for data visualization** to plot and display historical currency trends, aiding users in analyzing market patterns for informed decision-making.

6. **PANDAS - Pandas library for data manipulation** to efficiently manage and process currency data retrieved from APIs or the database, ensuring seamless functionality.

7. **SECURIPY - SecuriPy library for password encryption** to provide robust security for user authentication and protect sensitive user data.

3. FEATURES AND FUNCTIONALITIES

3.1 USER AUTHENTICATION

- **Secure login and account creation** ensures user data is protected by encrypting passwords using advanced security algorithms, preventing unauthorized access.
- The **login screen includes fields for username and password**, with real-time validation to alert users of errors such as incorrect credentials or missing inputs.
- Intuitive buttons for **logging in or creating new accounts** guide users through a seamless authentication process, enhancing usability and security.

3.2 CURRENCY CONVERSION

- **Real-time exchange rates** are retrieved dynamically using the Exchange Rate API, ensuring accurate and up-to-date conversion values.
- Users can **input the amount and select source and target currencies** from dropdown menus for easy customization of conversions.
- The **conversion result is displayed prominently** on the interface, providing clarity and quick access to essential information.

3.3 DATA VISUALIZATION

- **Graphical representation of currency rate trends** is achieved using Matplotlib, allowing users to visually analyze the fluctuations of exchange rates over time.
- Users can **click the "Visualize" button** to trigger the visualization process, enabling them to explore currency trends in a dedicated graphical interface.
- The separate window **displays detailed plots** of exchange rate trends, providing an intuitive way to interpret historical data for informed decision-making.

3.4 DATA VISUALIZATION

- The **fullscreen GUI** offers a sleek, modern design with responsive elements that adapt to different screen sizes, providing a seamless user experience.
- **Dropdown menus and background images** are integrated for enhanced visual appeal, creating an aesthetically pleasing interface while maintaining functional clari

4. DATABASE SETUP

1. SQLite DATABASE FOR USER CREDENTIALS

- The SQLite database is used to securely store user information, with a dedicated **user table** that holds essential credentials.
- The **user table** consists of two primary columns: USERNAME and ENCRYPTED_PW. The passwords are encrypted using secure algorithms, ensuring that sensitive data is protected.
- **Encryption methods** such as hashing (e.g., bcrypt or SHA-256) are implemented to make the passwords unreadable, even if the database is compromised.
- Each user's login attempt is validated by comparing the entered username and password against the stored encrypted credentials, ensuring only authorized access.

2. CONVERSION HISTORY TABLE FOR VISUALIZATION

- The SQLite database is used to securely store user information, with a dedicated **user table** that holds essential credentials.
- The **user table** consists of two primary columns: USERNAME and ENCRYPTED_PW. The passwords are encrypted using secure algorithms, ensuring that sensitive data is protected.
- **Encryption methods** such as hashing (e.g., bcrypt or SHA-256) are implemented to make the passwords unreadable, even if the database is compromised.
- Each user's login attempt is validated by comparing the entered username and password against the stored encrypted credentials, ensuring only authorized access.

5. USER INTERFACE DESIGN

LOGIN SCREEN

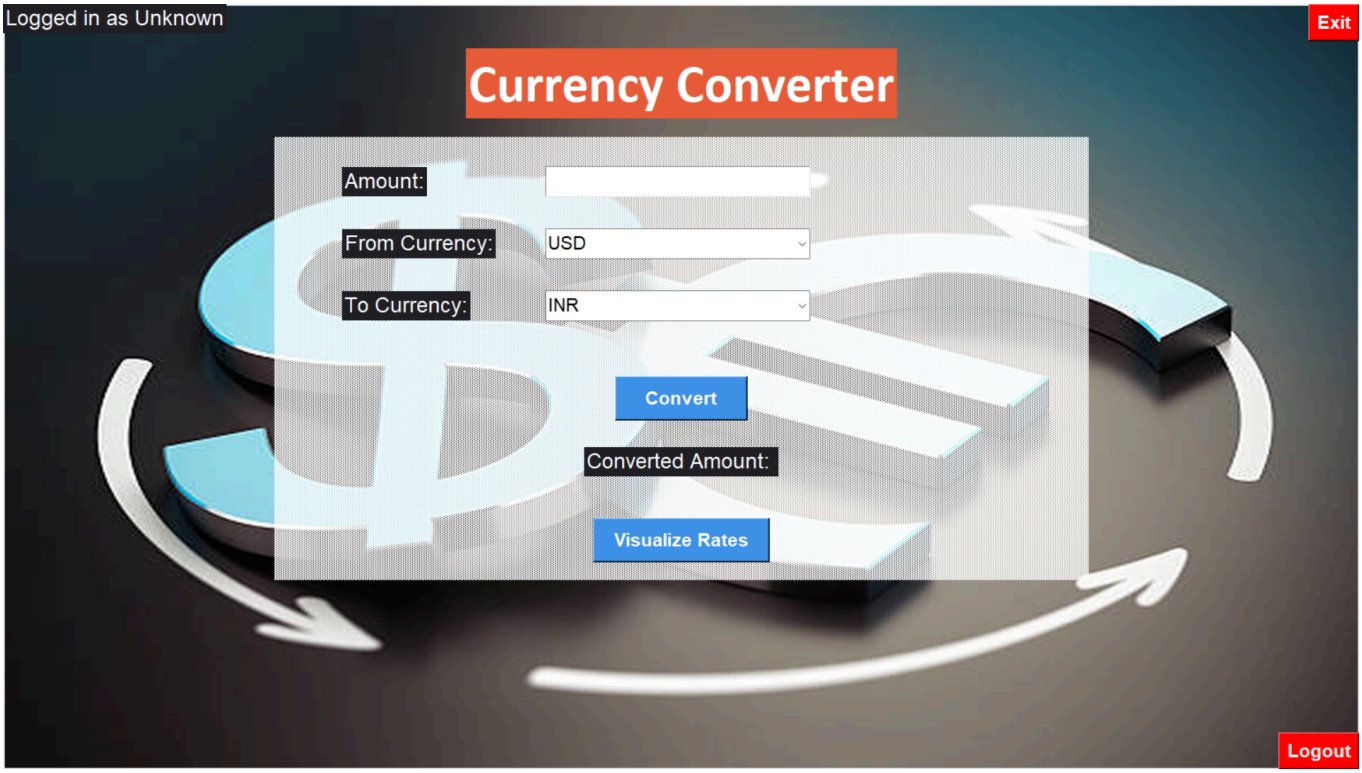
The screen design focuses on secure user authentication by including fields for **username** and **password**. These fields are essential components for a login system, ensuring that only authorized users can access the application. Users are required to input their credentials into these fields, which will then be verified by the system to grant access.

To enhance user interaction, the interface provides two key buttons:

1. **Login**: This button is intended for users who have already registered. After entering the correct username and password, they can click this button to log in to their accounts. The system verifies their credentials to ensure secure access.
2. **Create Account**: This option is designed for first-time users. Clicking this button redirects them to a registration process, where they can create a new account by providing necessary details like a unique username and password. Once registered, they can use their credentials to log in.

This dual-functionality interface streamlines the process for both new and existing users, maintaining security and usability. It ensures that sensitive data is protected while allowing easy access for authenticated users.

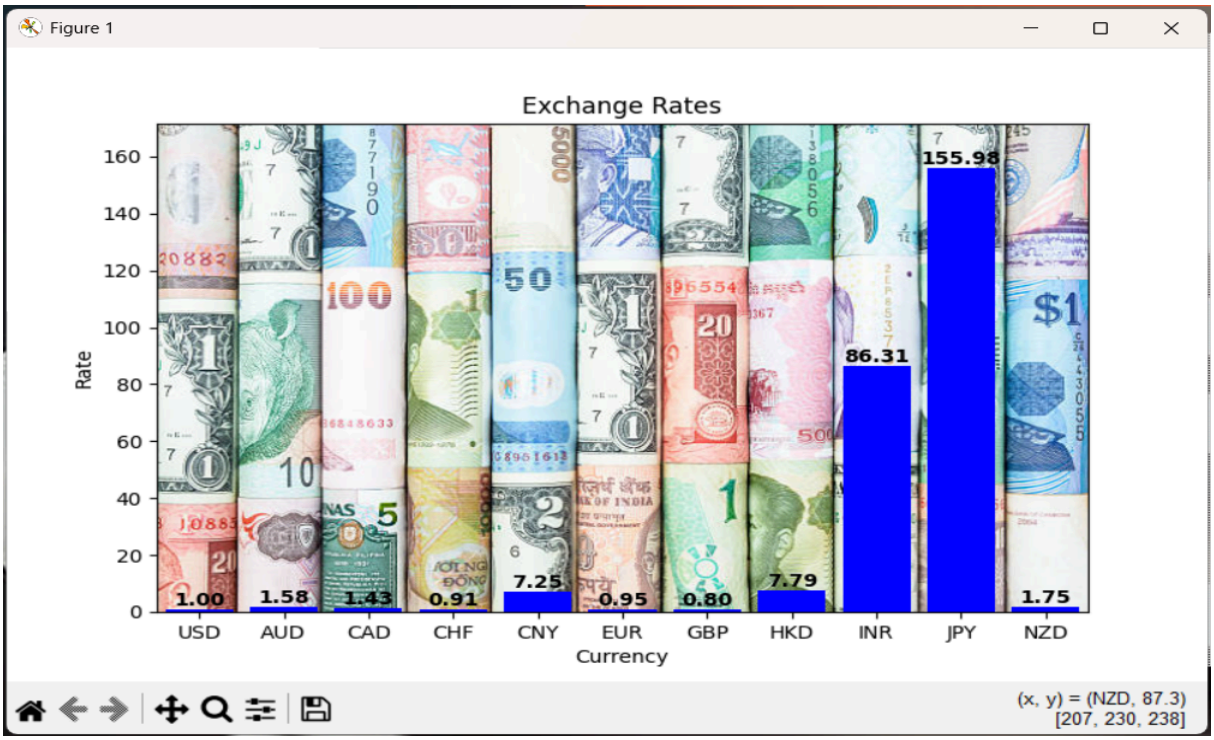
MAIN INTERFACE



The main interface of a Currency Converter application, designed to perform real-time currency conversions. The layout is user-friendly and includes the following elements:

1. **Input Fields:**
   - An input box labeled **Amount**, where users can enter the monetary value they want to convert.
2. **Dropdown Menus:**
   - From Currency: A dropdown menu for selecting the source currency (e.g., USD).
   - To Currency: A dropdown menu for selecting the target currency (e.g., INR).
3. **Convert Button:**
   - This button initiates the conversion process. Once the user fills in the required information and clicks this button, the system calculates the converted amount based on the latest exchange rates.
4. **Converted Amount Display:**
   - A dedicated area labeled Converted Amount displays the output, showing the equivalent value in the target currency after conversion.
5. **Visualize Rates Button:**
   - This button is likely intended to provide users with a graphical or tabular representation of current exchange rates for better insights.
6. **Logout and Exit:**
   - The Logout button allows the user to securely log out of their session.
   - An Exit button at the top-right corner closes the application.

VISUALIZATION OF EXCHANGE RATES



The visualization of exchange rates, created using matplotlib. It appears after clicking the "Visualize Rates" button in the Currency Converter application.

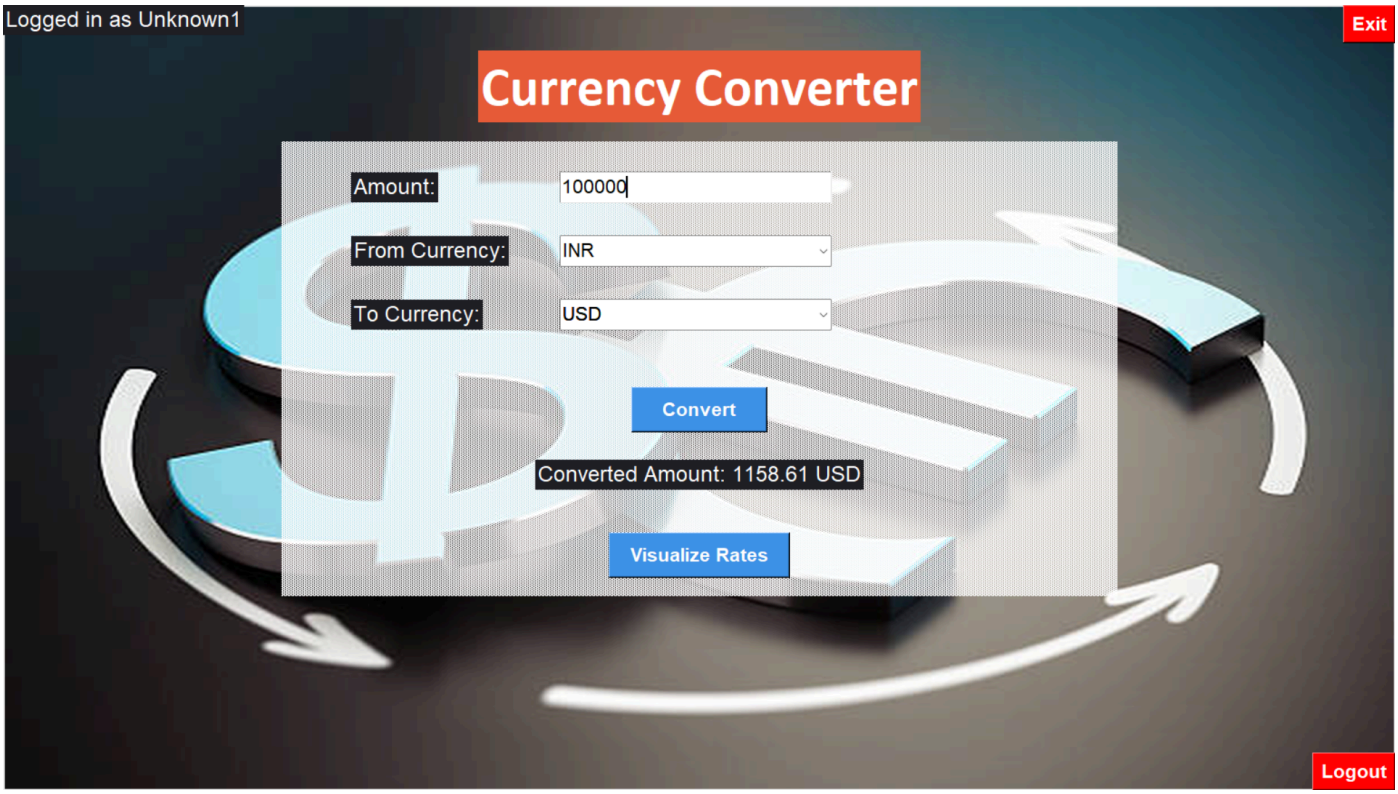
1. **Bar Chart:**
    - The chart plots different currencies **(USD, AUD, CAD, CHF, CNY, EUR, GBP, HKD, INR, JPY, NZD)** on the x-axis.
    - The exchange rates corresponding to each currency are plotted on the y-axis, with the rates represented as bars in blue.
2. **Currency Backgrounds:**
    - Behind the bars, the visualization features banknotes of the respective currencies, enhancing the chart's aesthetic and emphasizing the financial theme.
3. **Data Points:**
    - The exact exchange rates are shown above some of the bars, making it easy for users to read and compare the rates visually.
4. **Interactive Controls:**
    - The window includes interactive tools at the bottom-left corner, such as zoom, pan, and save, allowing users to explore the chart and save the data if required.

This feature enhances the application's usability by complementing the numeric conversion tool with graphical insights.

6. OUTPUT IMAGES

CREATED ACCOUNT:

CURRENCY CONVERSION:



CURRENCY VISUALIZATION:

7. CONCLUSION

In conclusion, the Currency Converter Application effectively leverages a combination of Python libraries, APIs, and robust security features to deliver an intuitive and secure platform for currency conversion. By integrating real-time exchange rate data through APIs and offering interactive data visualization with Matplotlib, the application not only simplifies the process of currency conversion but also provides valuable insights into currency trends. The secure user authentication ensures that sensitive data is protected, while the user-friendly interface enhances accessibility and ease of use. Overall, this application meets the practical needs of users by providing a comprehensive, efficient, and secure solution for managing currency conversions and analyzing exchange rate patterns.

REFERENCES:

- **Python Tkinter Documentation**
- **Exchange Rate API Documentation**
- **SQLite Official Documentation**
- **Matplotlib Documentation**
- **SecuriPy Encryption Module Documentation**

THANK YOU