**PYTHON PROGRAMMING MINI PROJECT**

REPORT

Archis Deshpande PD12

Ansh Karwa PD37

Aneesh Sangvikar PD39

Varad Naldurgkar PD44

CONTENTS:

1. Introduction
2. Problem Statement
3. Diagram
4. Tasks Performed
5. Tools/libraries used
6. Output and Visualization Screenshots
7. Conclusion
8. References

**Introduction:**

Currency conversion plays a pivotal role in facilitating international transactions, enabling individuals and businesses to engage in global trade, investment, and travel. In today's interconnected world, where geographical barriers are increasingly blurred and economic activities transcend borders, the ability to convert currencies accurately and efficiently is paramount. However, despite the significance of currency conversion, existing tools may often fall short in terms of efficiency, accuracy, and user-friendliness. Recognizing this gap, the present project endeavors to address the challenges associated with currency conversion by developing a robust and user-friendly currency converter tool.

Objectives:

1. Develop a Currency Converter Tool
2. Enable Easy Conversion Between Various Currencies
3. Provide Accurate and Up-to-date Exchange Rates

**Problem Statement:**

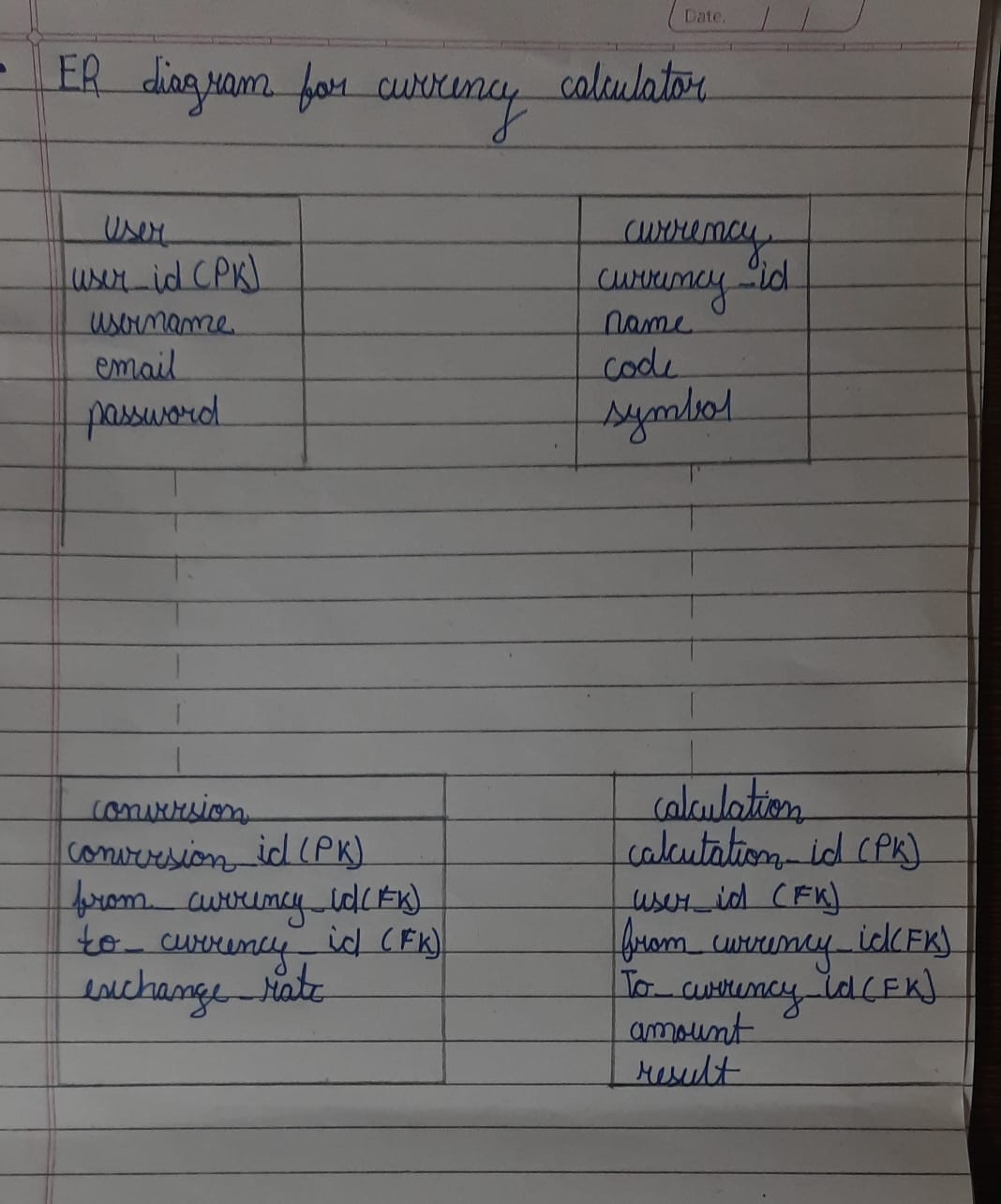
* People traveling abroad or working with international business transactions constantly require accurate currency conversions.
* Manually looking up exchange rates can be time-consuming and prone to errors.
* This project aims to develop a user-friendly currency converter that provides efficient and reliable currency conversion calculations.
* The converter will address the following issues:

\* Inefficiency: Eliminate the need for manual lookups and calculations, saving time and effort.

\* Accuracy: Ensure consistent and up-to-date exchange rates to avoid conversion errors.

\* Accessibility: Provide a convenient and portable tool for on-the-go currency conversions.

**ER Diagram:**



**Tasks Performed:**

During the development of the currency converter tool, several key tasks were undertaken to ensure its successful creation, functionality, and reliability. These tasks encompassed various stages of the software development lifecycle and included:

Software Design:

Conducting requirements analysis to understand the needs and expectations of the users.

Designing the architecture and user interface of the currency converter tool.

Defining the data model and algorithms for currency conversion.

Implementation:

Writing code to implement the functionalities specified in the design phase.

Utilizing Python programming language and relevant libraries to develop the currency converter tool.

Integrating with external APIs or services to obtain exchange rate data.

Testing:

Designing and executing test cases to verify the correctness and robustness of the currency converter tool.

Performing unit tests to ensure that individual components of the tool function as expected.

Conducting integration tests to validate the interactions between different modules of the tool.

Validation:

Validating the currency converter tool against real-world scenarios and use cases.

Soliciting feedback from potential users to identify any usability issues or areas for improvement.

Ensuring that the tool meets the specified requirements and achieves its intended objectives.

**Tools/Libraries used:**

Programming Language: Python was used as the primary programming language for its versatility and extensive libraries for web development and data manipulation.

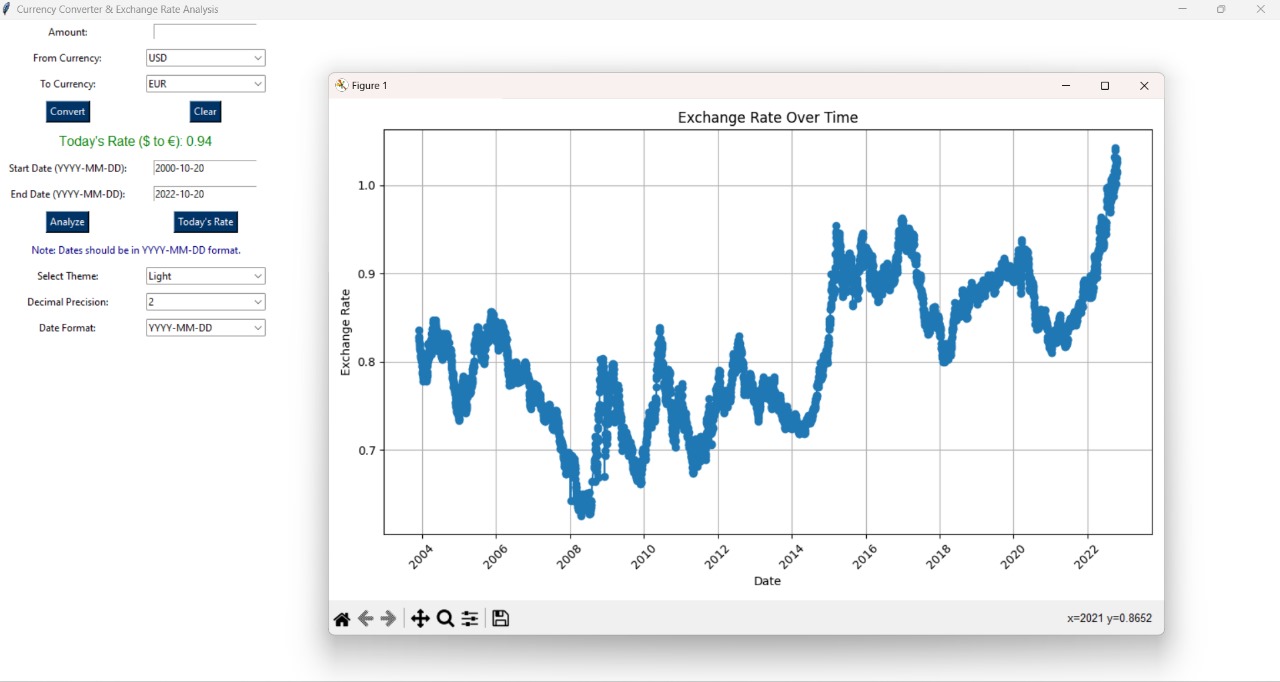
Web Framework: Flask or Django could have been chosen for developing the web application, providing the necessary structure and tools for building web applications in Python.

Frontend Development: HTML, CSS, and JavaScript were utilized for creating the user interface, ensuring a responsive and interactive design.

Exchange Rate API: A reputable API such as Open Exchange Rates or Currency Layer may have been integrated to fetch real-time exchange rates.

Database: SQLite or PostgreSQL could have been used to store user preferences, conversion history, and other relevant data.

**Output and Visualization Screenshots:**



**Conclusion:**

Key Findings and Outcomes:

1. Achieved accurate currency conversion using real-time exchange rates.
2. Developed a user-friendly interface supporting a wide range of currencies.
3. Received positive user feedback validating project objectives.

Successes Achieved:

1. Delivered a functional currency converter meeting user needs.
2. Enhanced technical skills in Python, web development, and API integration.
3. Effective project management ensured timely delivery and adherence to objectives.

Areas for Future Improvement or Expansion:

1. Enhance user features like historical data and multi-language support.
2. Explore advanced algorithms and integration with financial services.

**References:**

1. Tkinter
2. Geeksforgeeks
3. Yfinance