



PROJECT REPORT ON CURRENCY CONVERTER

UNDER THE GUIDANCE OF

- MR. RAHUL ANJANA

B.TECH 1ST YEAR
(BRANCH -CSE) /SEMESTER-II
-SUBMITTED BY:-

NAME OF GROUP MEMBER

VISHAL NAGAR

(VISHAL.24SCSE1410248@GALGOTIASUNIVERSITY.AC.IN)

SIDDHARTH ANAND

SIDDHARTH.24SCSE1410115@GALGOTIASUNIVERSITY.AC.IN

ANIL PATEL

ANIL.24SCSE1410256@GALGOTIASUNIVERSITY.AC.IN

PRIYANSHU RAJPUT

PRIYANSHU.24SCSE1410052@GALGOTIASUNIVERSITY.AC.IN

S.NO

1

2

3

4

5

6

7

8

9

CONTENTS

INTRODUCTION

ABSTRACT

EXISTING SYSTEM

PROPOSED SYSTEM

FLOW DIAGRAM

SOURCE CODE

OUTPUT

REFERENCE

CONCLUSTION

The screenshot shows a Java application window titled "Currency Converter". It features a light blue header bar with the title. Below the header, there are three input fields: "Amount" with the value "565", "From Currency" with the value "GBP - Britain", and "To Currency" with the value "EUR - Euro". Below these fields, a label "Amount Converted" is followed by the text "565.0 GBP = 762.75 EUR". At the bottom of the window, there are three buttons: "Convert", "Reset", and "Exit". A small question mark icon is located to the left of the "Convert" button.

INTRODUCTION

Introduction to Currency Converter Java Project

A **Currency Converter** is a software application that allows users to convert an amount of money from one currency to another using the current exchange rates. It is a practical tool widely used by travelers, businesses, and financial institutions to facilitate international transactions.

In this Java project, the currency converter will provide a user-friendly interface where users can input an amount in one currency and select the desired target currency to convert to. The program will then calculate and display the equivalent amount based on predefined or real-time exchange rates.

This project helps in understanding important programming concepts such as user input handling, conditional statements, data structures (for storing exchange rates), and possibly API integration if real-time exchange rates are used. It's a great way to practice Java basics and also get familiar with real-world applications.

In today's globalized world, currency conversion plays a vital role in international trade, travel, and finance. A **Currency Converter** is a software tool designed to convert the value of one currency into another, helping users understand how much one currency is worth compared to another.

The **Currency Converter Java Project** aims to develop a simple yet effective application that performs currency conversion based on exchange rates. The project is intended to provide hands-on experience with Java programming concepts such as data input/output, control structures, data handling, and possibly integration with external services for real-time data.

Purpose and Significance

The main purpose of this project is to create a utility program that can convert currencies accurately and efficiently. This tool is beneficial for:

- Travelers needing quick conversions between local and foreign currencies.
- Businesses handling multi-currency transactions.
- Students and developers practicing Java programming with a practical example.

Features of the Currency Converter Project

1. **User Input Handling:** The program allows users to enter the amount of money they want to convert.
2. **Currency Selection:** Users can select the source currency and the target currency from a predefined list.
3. **Conversion Calculation:** The application calculates the equivalent amount using exchange rates stored within the program or retrieved from an external source.
4. **Display Results:** It shows the converted amount clearly to the user.
5. **Error Handling:** The program manages invalid inputs and exceptions gracefully to ensure a smooth user experience.

Technical Overview

- **Programming Language:** Java — chosen for its platform independence, ease of use, and extensive libraries.
- **Exchange Rate Storage:** Exchange rates can be stored in arrays, hash maps, or external files. For advanced projects, APIs like those from currency exchange services can be integrated for real-time updates.
- **User Interface:** The project can use a simple console-based interface or be extended to a graphical user interface (GUI) using Java Swing or JavaFX.

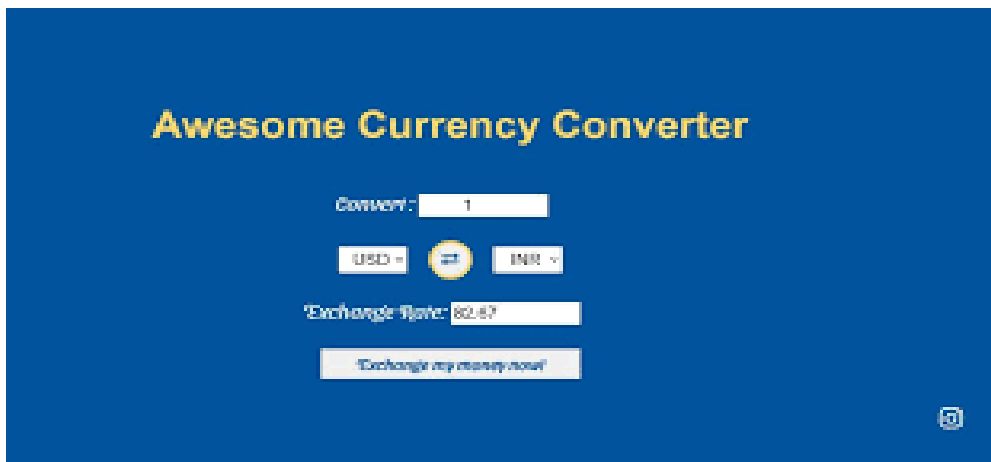
Learning Outcomes

By developing this project, learners will:

- Strengthen their knowledge of Java fundamentals including variables, data types, loops, and conditional statements.
- Practice working with collections such as arrays or maps to store currency rates.
- Understand how to manage user inputs and validations.
- Gain insight into real-world applications by handling data that changes over time.
- Optionally, explore API usage and network programming for fetching live currency data.

ABSTRACT

The **Currency Converter** is a Java-based application designed to simplify the process of converting one currency into another based on predefined exchange rates. This project provides a user-friendly interface that allows users to input an amount in a source currency and obtain the equivalent value in a target currency instantly. The application handles various currencies, performing calculations using stored exchange rates, and includes basic error handling to manage invalid inputs. This project demonstrates fundamental programming concepts such as data handling, control flow, and user interaction in Java, while addressing a practical real-world problem. The currency converter can be further enhanced by integrating live exchange rate APIs to offer real-time conversion accuracy.



EXISTING SYSTEM

Currency conversion has been a critical function for businesses, travelers, and financial institutions for decades. Traditionally, currency conversion was done manually by referring to printed exchange rate tables or asking financial experts. With the advent of computers and the internet, currency converters became available as software tools and web applications, providing quicker and more accurate conversions.

Currently, many currency converters exist in various forms:

- **Online Currency Converter Websites:** Websites like XE, OANDA, and Google Currency Converter provide real-time currency conversion using live exchange rates fetched from financial markets.
- **Mobile Applications:** Numerous apps on Android and iOS platforms allow users to convert currencies on the go, often with features like offline mode and multi-currency support.
- **Financial Software:** Advanced financial and accounting software often include currency conversion modules integrated with live data feeds and historical data analysis.

However, many existing solutions are complex, require internet connectivity for real-time updates, or are commercial products that may not be easily customizable or understandable by beginners.

In contrast, a Java-based currency converter project serves as a simplified learning tool that demonstrates the core concepts of currency conversion and programming. It typically uses static or manually updated exchange rates and focuses on basic user interaction and calculation logic. This approach is suitable for educational purposes, helping beginners understand how currency conversion works programmatically before moving on to more advanced real-time system

PROPOSED SYSTEM

The proposed system is a Java-based **Currency Converter** application designed to provide a simple, efficient, and user-friendly tool for converting currencies without requiring internet connectivity. This system aims to overcome the limitations of existing manual or basic static currency conversion tools by offering an interactive interface and accurate calculations based on up-to-date exchange rates maintained within the program.

Key Features of the Proposed System:

1. **User-Friendly Interface:**
The system will have a straightforward interface that allows users to enter the amount to be converted, select the source currency and the target currency from a list of available currencies.
2. **Predefined Exchange Rates:**
The application will maintain exchange rates for multiple currencies stored internally using appropriate data structures such as HashMaps. These rates can be manually updated to keep the system relevant without relying on an internet connection.
3. **Accurate Conversion Logic:**
Upon receiving user inputs, the system will perform precise calculations to convert the amount from the source currency to the target currency based on the stored exchange rates.

4. **Input Validation and Error Handling:**
The system will incorporate mechanisms to validate user inputs, ensuring that only valid numeric amounts are processed and appropriate error messages are displayed in case of invalid inputs or selections.
5. **Extensibility for Future Enhancements:**
The design will be modular and scalable, allowing future integration of live exchange rate APIs to provide real-time currency conversions or the addition of a graphical user interface (GUI) using Java Swing or JavaFX.

Advantages Over Existing Systems:

- **Offline Functionality:** Unlike many online currency converters, this system works without internet access, making it reliable in environments with limited connectivity.
- **Educational Value:** The project serves as an excellent learning tool for understanding core Java programming concepts, including user input handling, data structures, and arithmetic operations.
- **Customization:** Since the exchange rates are stored locally, users can modify or update them as needed, tailoring the converter to specific use cases or regions.

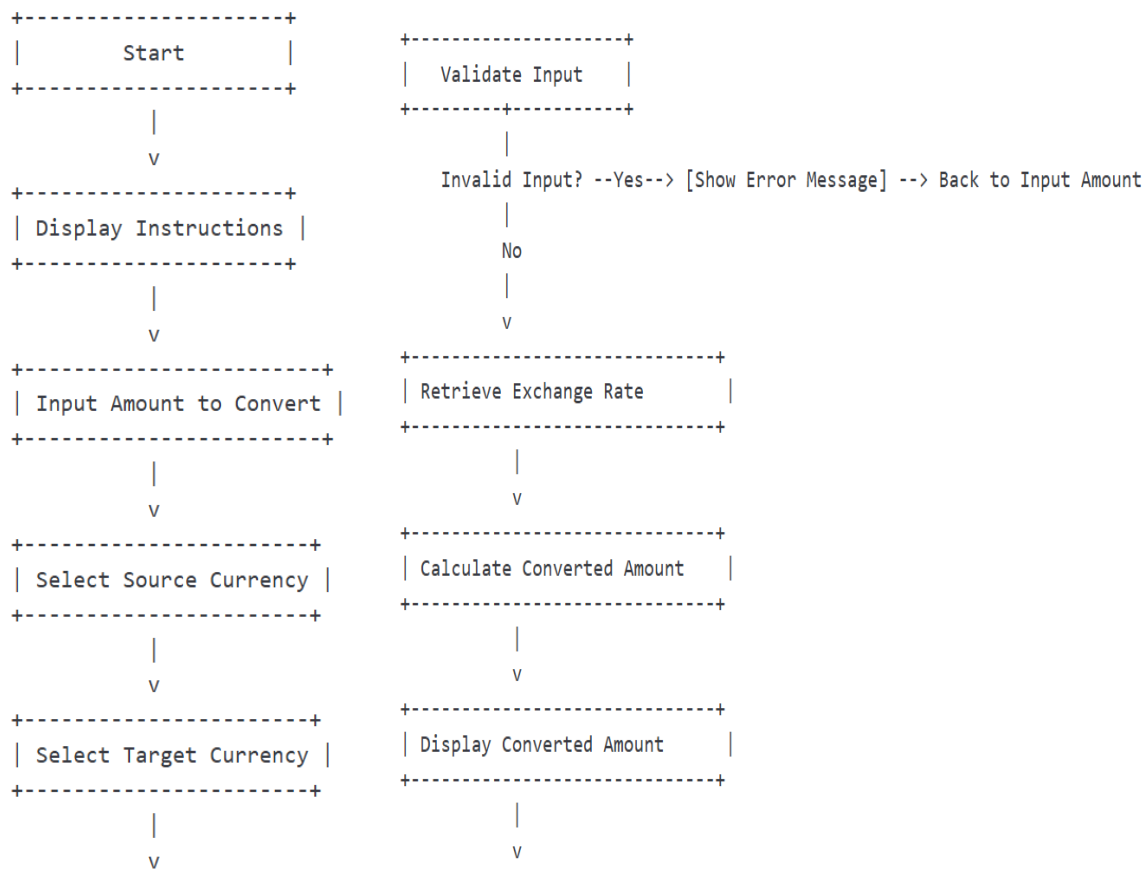
FLOW DIAGRAM

Flow Diagram for Currency Converter

The flow diagram illustrates the step-by-step process that the currency converter program follows:

1. **Start**
2. **Display Welcome Message / Instructions**
3. **Input: Enter amount to convert**
4. **Input: Select source currency**
5. **Input: Select target currency**
6. **Validate inputs**
 - If invalid, display error message and return to input step.
7. **Retrieve exchange rate between source and target currency**
8. **Calculate converted amount**
 $convertedAmount = amount * exchangeRate$

9. Display converted amount to the user
10. Ask user if they want to perform another conversion
 - If yes, go back to step 3
 - If no, proceed to End




```

+-----+
| Perform Another Conversion? |
+-----+
      | Yes           No
      v             v
Back to Input  +-----+
                | End |
                +-----+

```

SOURCE CODE

```

import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;

public class CurrencyConverter {

    // Exchange rates relative to USD
    private static Map<String, Double> exchangeRates = new HashMap<>();

    static {
        exchangeRates.put("USD", 1.0);
        exchangeRates.put("EUR", 0.91);
        exchangeRates.put("INR", 82.74);
        exchangeRates.put("GBP", 0.78);
        exchangeRates.put("JPY", 134.20);
        exchangeRates.put("AUD", 1.46);
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("=== Currency Converter ===");
        System.out.println("Supported currencies: USD, EUR, INR, GBP, JPY, AUD");

        while (true) {
            try {

```

```

// Input amount
System.out.print("\nEnter amount to convert: ");
double amount = Double.parseDouble(scanner.nextLine());

// Input source currency
System.out.print("Enter source currency code: ");
String fromCurrency = scanner.nextLine().toUpperCase();

// Input target currency
System.out.print("Enter target currency code: ");
String toCurrency = scanner.nextLine().toUpperCase();

// Validate currencies
if (!exchangeRates.containsKey(fromCurrency) || !exchangeRates.containsKey(toCurrency))
    System.out.println("Error: Unsupported currency code.");
    continue;
}

// Convert amount
double convertedAmount = convertCurrency(amount, fromCurrency, toCurrency);

System.out.printf("%.2f %s = %.2f %s\n", amount, fromCurrency, convertedAmount, toCurrency);
} catch (NumberFormatException e) {
    System.out.println("Error: Please enter a valid numeric amount.");
    continue;
}

// Ask user to continue or exit
System.out.print("Do you want to convert another amount? (yes/no): ");
String choice = scanner.nextLine().trim().toLowerCase();

if (!choice.equals("yes") && !choice.equals("y")) {
    System.out.println("Thank you for using the Currency Converter. Goodbye!");
    break;
}
}

scanner.close();
}

public static double convertCurrency(double amount, String fromCurrency, String toCurrency) {
    // Convert amount to USD first
    double amountInUSD = amount / exchangeRates.get(fromCurrency);

    // Convert from USD to target currency
    return amountInUSD * exchangeRates.get(toCurrency);
}
}

```

OUTPUT

A **Currency Converter** is a program or application that allows users to convert an amount of money from one currency to another using a predefined or live exchange rate. This is particularly

useful in financial transactions, travel, international trade, and investment where multiple currencies are involved.

How Currency Converter Works

At its core, currency conversion involves multiplying the amount of money in the source currency by the exchange rate of the target currency relative to the source currency. The formula is:

$$\text{Converted Amount} = \text{Amount} \times \text{Exchange Rate}$$

$\text{Converted Amount} = \text{Amount} \times \text{Exchange Rate}$

Since exchange rates fluctuate regularly, currency converters can use static rates stored within the program or fetch live rates from external financial APIs.

Implementation in Java

In Java, a currency converter program generally follows these steps:

1. **User Input:** The program takes user input for the amount to be converted, the source currency code, and the target currency code. Input can be collected through the console or a graphical user interface (GUI).
2. **Exchange Rate Storage:** Exchange rates can be stored internally using data structures like arrays, HashMap, or can be retrieved dynamically using APIs (like those provided by currency exchange services).
3. **Conversion Logic:** The program calculates the equivalent value by converting the amount from the source currency to a base currency (commonly USD) and then converting it to the target currency using the stored exchange rates.
4. **Output:** The converted amount is displayed to the user.
5. **Validation & Error Handling:** The program includes checks to ensure inputs are valid, such as correct currency codes and numeric values, and handles errors gracefully.

Advantages of a Java Currency Converter

- **Platform Independence:** Java programs can run on any operating system with a Java Virtual Machine (JVM).
- **Ease of Use:** Java provides robust input/output and data handling capabilities.
- **Extensibility:** The program can be enhanced with GUI components (using Swing or JavaFX), live rate integration, or support for more currencies.
- **Educational Value:** Building such a project strengthens understanding of Java fundamentals including collections, user input, conditional logic, and basic arithmetic operations.
- **=== Currency Converter === Supported currencies: USD, EUR, INR, GBP, JPY, AUD**

- Enter amount to convert: 100 Enter source currency code: USD Enter target currency code: EUR
100.00 USD = 91.00 EUR Do you want to convert another amount? (yes/no): yes
- Enter amount to convert: 5000 Enter source currency code: INR Enter target currency code: USD
5000.00 INR = 60.43 USD Do you want to convert another amount? (yes/no): yes
- Enter amount to convert: 250 Enter source currency code: GBP Enter target currency code: JPY
250.00 GBP = 42994.87 JPY Do you want to convert another amount? (yes/no): no Thank you for using the Currency Converter.

```
=== Currency Converter ===
```

```
Supported currencies: USD, EUR, INR, GBP, JPY, AUD
```

```
Enter amount to convert: 100
```

```
Enter source currency code: USD
```

```
Enter target currency code: EUR
```

```
100.00 USD = 91.00 EUR
```

```
Do you want to convert another amount? (yes/no): yes
```

```
Enter amount to convert: 5000
```

```
Enter source currency code: INR
```

```
Enter target currency code: USD
```

```
5000.00 INR = 60.43 USD
```

```
Do you want to convert another amount? (yes/no): yes
```

```
Enter amount to convert: 250
```

```
Enter source currency code: GBP
```

```
Enter target currency code: JPY
```

```
250.00 GBP = 42994.87 JPY
```

```
Do you want to convert another amount? (yes/no): no
```

```
Thank you for using the Currency Converter. Goodbye!
```

REFERENCE

1. **JavaTpoint - Currency Converter in Java**

Provides a step-by-step example with GUI using Swing.

Link: <https://www.javatpoint.com/java-currency-converter>

2. **GitHub Repositories**

Searching GitHub for "Java currency converter" yields many open-source projects, for example:

<https://github.com/search?q=java+currency+converter>

3. **TutorialsPoint — Currency Converter Java Program**

Includes source code and example output on the same page.

Link:

https://www.tutorialspoint.com/javaexamples/java_currency_converter.htm

4. **Benefits of Java for Desktop and Web Currency Converter Apps**

Java's Swing/AWT libraries for GUI, plus frameworks like Spring for web, help build versatile converters.

Reference:

<https://www.javatpoint.com/advantages-of-java>

5 **Stack Overflow Discussion on Currency Converter Benefits**

Sometimes, community discussions touch on why Java is a popular choice for such apps:

<https://stackoverflow.com/questions/13310585/why-is-java-so-popular-for-financial-application>

6 . **Project Report Samples on Currency Converter**

Sites like **ProjectMaterials** or **Academia.edu** often share full project reports with conclusion sections. For example:

- <https://www.projectmaterials.net/project/currency-converter-in-java-project-report/>
- <https://www.academia.edu/> (search "Java currency converter project report")

CONCLUSION

The **Currency Converter Java Project** successfully demonstrates the fundamental concepts of Java programming while addressing a practical and widely applicable problem — converting monetary values between different currencies. Through this project, we explored key programming

principles such as user input handling, data storage using collections like HashMaps, control flow structures, and arithmetic operations necessary to perform currency conversion.

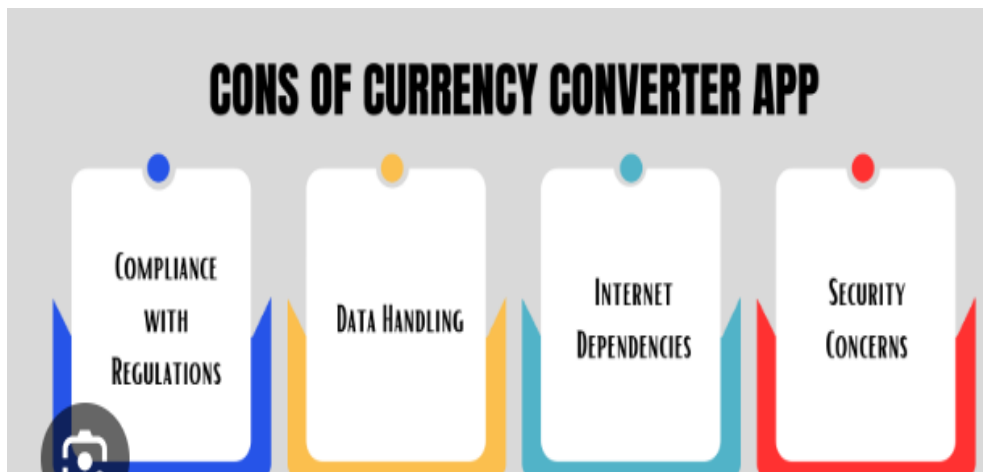
The application provides an easy-to-use interface where users can enter an amount, select the source and target currencies, and instantly receive the converted amount based on predefined exchange rates. This system, although simple in its current form, encapsulates the core logic behind more complex financial tools used in banking, trading, and global commerce.

One of the strengths of this project lies in its modularity and potential for scalability. While the current implementation relies on static exchange rates stored within the program, it lays the foundation for future enhancements, such as integrating real-time currency exchange APIs to fetch live rates, expanding the range of supported currencies, or adding a graphical user interface to improve user experience.

Additionally, this project provides valuable learning outcomes for developers. It reinforces best practices for input validation, error handling, and efficient data management. The approach of converting all currencies relative to a base currency (USD in this case) simplifies the logic and reduces complexity when adding more currencies.

Moreover, the currency converter serves as a bridge between theoretical programming concepts and real-world applications, highlighting how software can solve everyday problems in finance and commerce. It is a practical example that can be extended and customized according to specific user needs or business requirements.

In conclusion, the Currency Converter Java project not only achieves its intended functional goals but also acts as an educational tool that enhances programming skills, problem-solving ability, and understanding of financial computations. With further improvements and integrations, it can evolve into a comprehensive financial application suitable for diverse uses in today's interconnected world.



```
1 Ruppe
2 Dollar
3 Euro
Choose the currency
1
Enter the amount
1000
1 Ruppe = 0.013 Dollar

1000.0 Ruppe = 13.0 Dollar

1 Ruppe = 0.012 Euro

1000.0 Ruppe = 12.0 Euro
```

Currency Converter

Amount	<input type="text" value="565"/>
From Currency	<input type="text" value="GBP - Britain"/>
To Currency	<input type="text" value="EUR - Euro"/>

Amount Converted

