A

PROJECT REPORT ON

"Currency Converter "

<u>SUBMITTED BY:</u>

Mr. Kartik Vitthal Unde(2124UCEM1057)

<u>SUBJECT:</u>

Programming in Problem Solving using C++

<u>Under the guidance of</u>

Miss. Ishwari Tirse

Department of Computer Science and Engineering
Sanjivani Rural Education Society's

SANJIVANI UNIVERSITY
KOPARGAON – 423603, DIST : AHMEDNAGAR

2024-2025

INDEX

INTRODUCTION

A currency converter system is essential in today's globalized world for several reasons. Firstly, it facilitates international trade by allowing businesses to quickly and accurately convert currencies, helping them understand costs, pricing, and profitability. This is particularly important for invoicing and payments in different currencies. Additionally, for travelers, a currency converter aids in budgeting and understanding purchasing power in foreign countries, enabling informed spending decisions.

Moreover, currency converters serve as valuable educational tools for students and researchers studying economics and finance, while also enabling analysts to observe market trends. The convenience of quick access to real-time exchange rates helps users make informed decisions on-the-go, whether for travel, shopping, or investments. Additionally, businesses can utilize these converters for risk management strategies to mitigate potential losses from currency fluctuations.

CODE

```cpp
#include <iostream>
using namespace std;
int main() {
 double amount;
 int choice;

 cout << "Currency Converter\n";
 cout << "Select a currency to convert from:\n";
 cout << "1. USD (US Dollar)\n";
 cout << "2. EUR (Euro)\n";
 cout << "3. JPY (Japanese Yen)\n";
 cout << "4. GBP (British Pound)\n";
 cout << "5. AUD (Australian Dollar)\n";
 cout<< "6. INR (Indian Rupee)\n";
 cout << "Enter your choice (1-6): ";
 cin >> choice;
 cout << "Enter the amount: ";
 cin >> amount;
 switch (choice) {

 case 1: // USD
 cout << amount << " USD = " << amount * 0.85 << " EUR\n";
cout << amount << " USD = " << amount * 110.0 << " JPY\n";
cout << amount << " USD = " << amount * 0.75 << " GBP\n";
cout << amount << " USD = " << amount * 1.35 << " AUD\n";
cout << amount << " USD = " << amount * 74.0 << " INR\n";
break;
```

```cpp
    case 2: // EUR cout << amount << " EUR = " << amount *
1.18 << " USD\n";
cout << amount << " EUR = " << amount * 129.0 <<"JPY\n";
cout << amount << " EUR = " << amount * 0.88 << " GBP\n";
cout << amount << " EUR = " << amount * 1.59 << " AUD\n";
cout << amount << " EUR = " << amount * 91.17 <<"INR\n";
break;

    case 3: // JPY
 cout << amount << " JPY = " << amount * 0.0091 << "
USD\n";
cout << amount << " JPY = " << amount * 0.0078 << " EUR\n";

 cout << amount << " JPY = " << amount * 0.0069 << "GBP\n";
cout << amount << " JPY = " << amount * 0.0123 << " AUD\n";
cout<< amount << " JPY = " << amount * 0.56 << "INR\n";

 break;


    case 4: // GBP
 cout << amount << " GBP = " << amount * 1.33 << " USD\n";
cout << amount << " GBP = " << amount * 1.14 << " EUR\n";
cout << amount << " GBP = " << amount * 145.0 << " JPY\n";
cout << amount << " GBP = " << amount * 1.78 << " AUD\n";
cout << amount <<" GBP = " << amount * 109.73<< "INR\n";
break;
```

```cpp
case 5: // AUD
 cout << amount << " AUD = " << amount * 0.74 << "USD\n";
cout << amount << " AUD = " << amount * 0.63 << " EUR\n";
cout << amount << " AUD = " << amount * 0.56 << " GBP\n";
cout << amount << " AUD = " << amount * 81.0 << " JPY\n";
cout << amount << " AUD = " << amount * 56.43 << "INR\n";
break;


case 6:// INR
 cout << amount << " INR = " << amount * 0.012 << " USD\n";
cout << amount << " INR = " << amount * 0.11 << " EUR\n";
cout << amount << " INR = " << amount * 0.0091 << " GBP\n";
cout << amount << " INR = " << amount * 1.78 << " JPY\n";
cout << amount << " INR = " << amount * 0.018 <<" AUD\n";
default:
 cout << "Invalid choice.\n";
 }
 return 0;
}
```

Output

```
Currency Converter
Select a currency to convert from:
1. USD (US Dollar)
2. EUR (Euro)
3. JPY (Japanese Yen)
4. GBP (British Pound)
5. AUD (Australian Dollar)
6. INR (Indian Rupee)
Enter your choice (1-6): 2
Enter the amount: 25000
25000 EUR = 29500 USD
25000 EUR = 3.225e+06JPY
25000 EUR = 22000 GBP
25000 EUR = 39750 AUD
25000 EUR = 2.27925e+06INR
```

```
Currency Converter
Select a currency to convert from:
1. USD (US Dollar)
2. EUR (Euro)
3. JPY (Japanese Yen)
4. GBP (British Pound)
5. AUD (Australian Dollar)
6. INR (Indian Rupee)
Enter your choice (1-6): 6
Enter the amount: 25000
25000 INR = 300 USD
25000 INR = 2750 EUR
25000 INR = 227.5 GBP
25000 INR = 44500 JPY
25000 INR = 450 AUD
Invalid choice.


=== Code Execution Successful ===
```

```
Currency Converter
Select a currency to convert from:
1. USD (US Dollar)
2. EUR (Euro)
3. JPY (Japanese Yen)
4. GBP (British Pound)
5. AUD (Australian Dollar)
6. INR (Indian Rupee)
Enter your choice (1-6): 3
Enter the amount: 25000
25000 JPY = 227.5 USD
25000 JPY = 195 EUR
25000 JPY = 172.5 GBP
25000 JPY = 307.5 AUD
25000 JPY = 14000INR
```

```
Currency Converter
Select a currency to convert from:
1. USD (US Dollar)
2. EUR (Euro)
3. JPY (Japanese Yen)
4. GBP (British Pound)
5. AUD (Australian Dollar)
6. INR (Indian Rupee)
Enter your choice (1-6): 1
Enter the amount: 2500
2500 USD = 2125 EUR
2500 USD = 275000 JPY
2500 USD = 1875 GBP
2500 USD = 3375 AUD
2500 USD = 185000 INR
```

```
Currency Converter
Select a currency to convert from:
1. USD (US Dollar)
2. EUR (Euro)
3. JPY (Japanese Yen)
4. GBP (British Pound)
5. AUD (Australian Dollar)
6. INR (Indian Rupee)
Enter your choice (1-6): 5
Enter the amount: 25000
25000 AUD = 18500USD
25000 AUD = 15750 EUR
25000 AUD = 14000 GBP
25000 AUD = 2.025e+06 JPY
25000 AUD = 1.41075e+06INR
```

```
Currency Converter
Select a currency to convert from:
1. USD (US Dollar)
2. EUR (Euro)
3. JPY (Japanese Yen)
4. GBP (British Pound)
5. AUD (Australian Dollar)
6. INR (Indian Rupee)
Enter your choice (1-6): 4
Enter the amount: 25000
25000 GBP = 33250 USD
25000 GBP = 28500 EUR
25000 GBP = 3.625e+06 JPY
25000 GBP = 44500 AUD
25000 GBP = 2.74325e+06INR
```

CONCLUSION

The Currency Converter project showcases fundamental C++ programming skills, including functions, loops, conditionals, and user input handling. Through this project, we learned to:

- Define and use functions for modularity.
- Utilize data structures like map for easy lookups.
- Implement user-friendly interfaces for better interaction.

This mini project can be further enhanced by integrating real-time exchange rate APIs, adding error handling, and expanding the list of supported currencies. Overall, it serves as a solid foundation for more complex programming challenges in C++.