# Currency Converter Project Report

## Description

The Currency Converter project is a simple and efficient application developed using the GCC compiler. This project allows users to convert amounts between various currencies with real-time accuracy. The code has been implemented in C programming language and is executed in a DOS-based environment, making it lightweight and portable.

## Features

- Supports conversions between major international currencies, such as USD, EUR, INR, GBP, JPY, and more.

- Simple text-based interface for easy navigation.

- Utilizes predefined conversion rates to ensure accurate calculations.

- Fast and efficient currency conversion.

- Includes validation for incorrect inputs and displays appropriate error messages.

- Easy to add support for additional currencies.

## Terms and Details

Source Code: Written in C programming language, compiled using GCC.

Compiler: The project uses the GCC (GNU Compiler Collection), ensuring compatibility and performance across platforms.

Execution Environment: Designed for DOS-based systems, making it lightweight and accessible on older machines or in terminal-based environments.

Predefined Conversion Rates: Rates are hardcoded in the program and can be updated manually as needed.

## Steps to Execute

- Install GCC Compiler: Ensure that the GCC compiler is installed on your system. You can download it from GCC official website (https://gcc.gnu.org/).

- Compile the Program: Open the terminal or command prompt and navigate to the directory containing the source code file. Use the following command to compile:  
 gcc -o currency\_converter currency\_converter.c

- Run the Program: After successful compilation, run the program using the command:  
 ./currency\_converter

- Follow On-Screen Instructions: Enter the amount and select the source and target currencies as prompted by the program.

- View Results: The program will display the converted amount.

## Future Enhancements

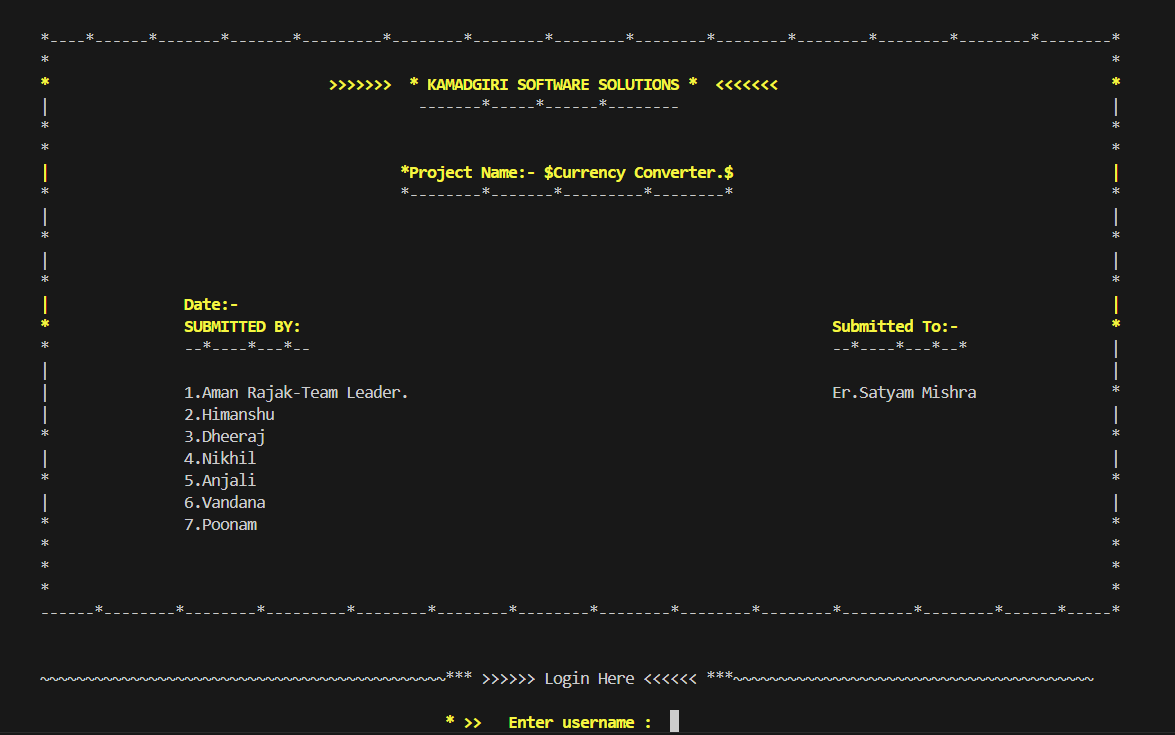
- Integration of live exchange rates using APIs.

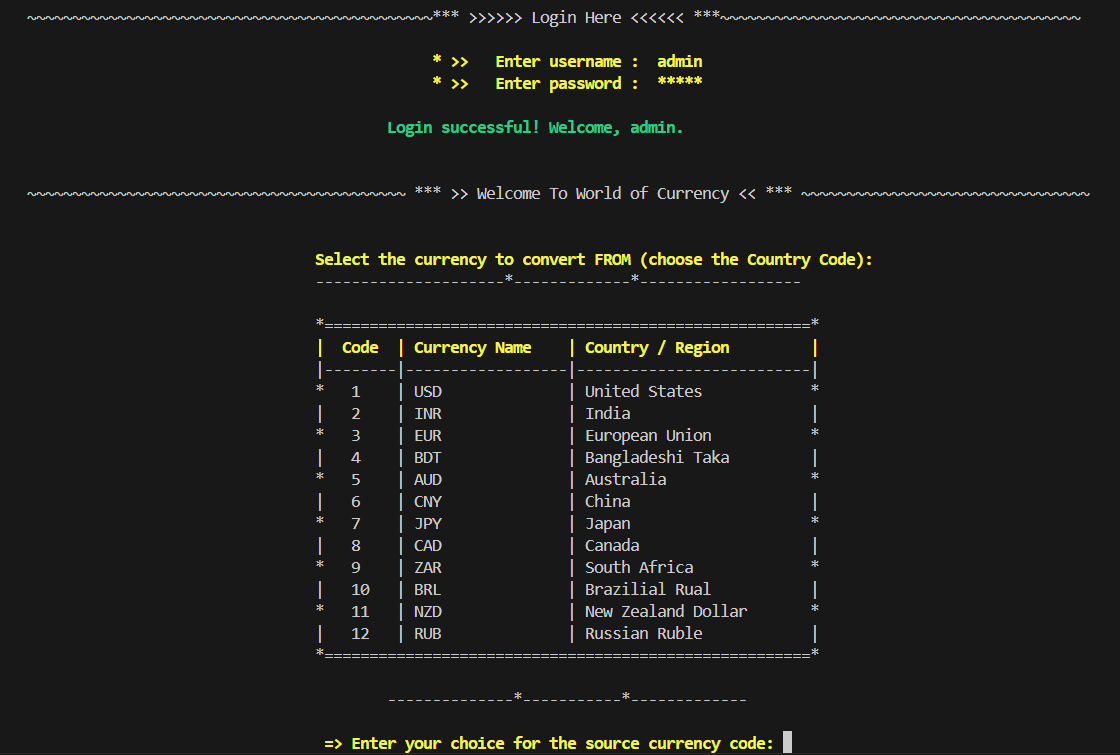
- Graphical User Interface (GUI) for enhanced user experience.

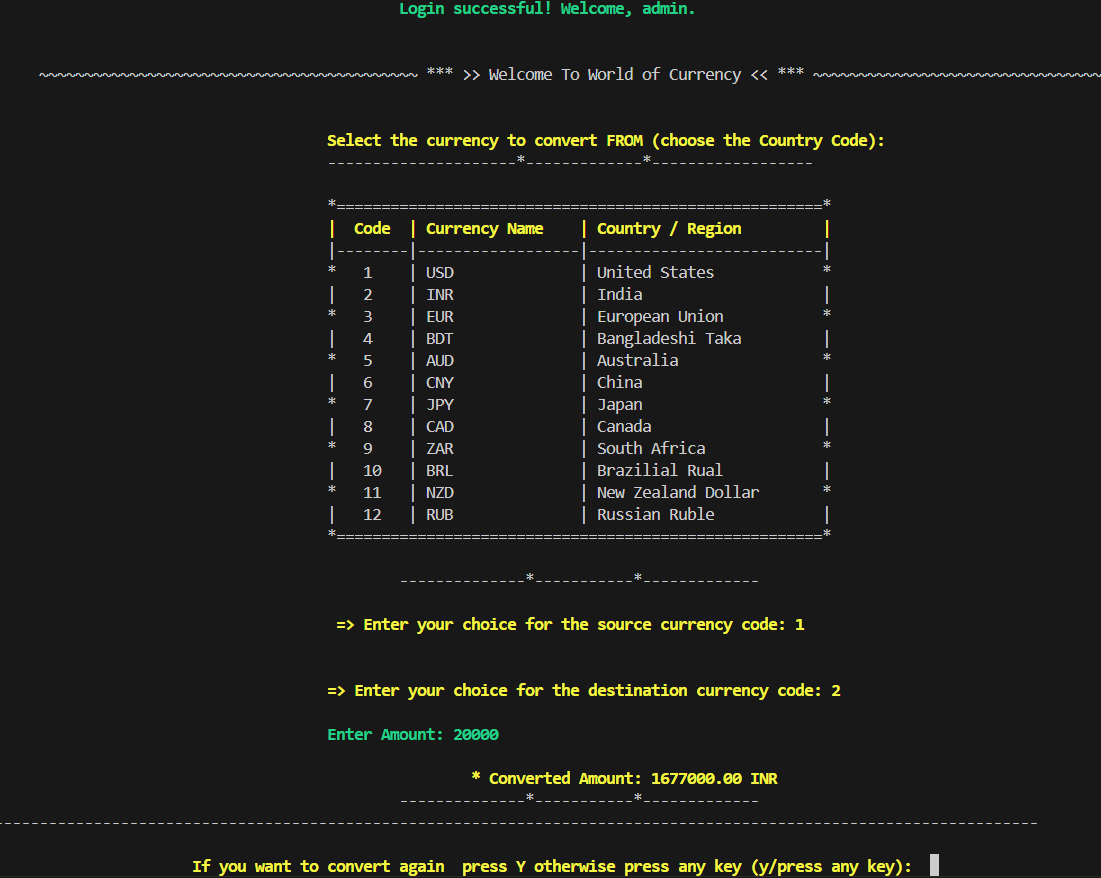
- Support for additional languages and regional currencies.

## Conclusion

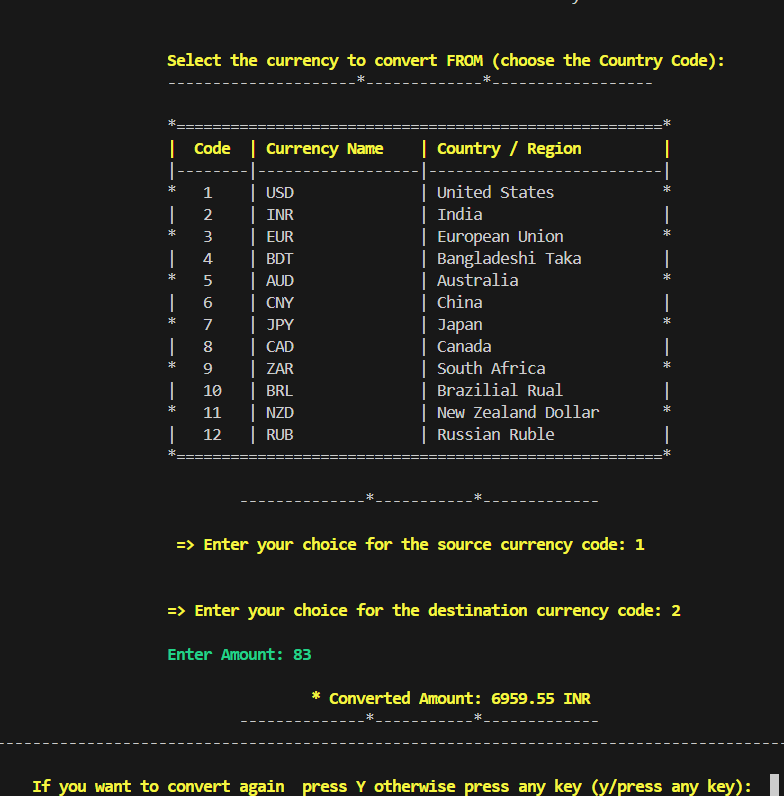
The Currency Converter project is a functional and practical application for basic currency conversions. With its lightweight design and GCC compatibility, it serves as an excellent learning tool for understanding C programming concepts and real-world application development.

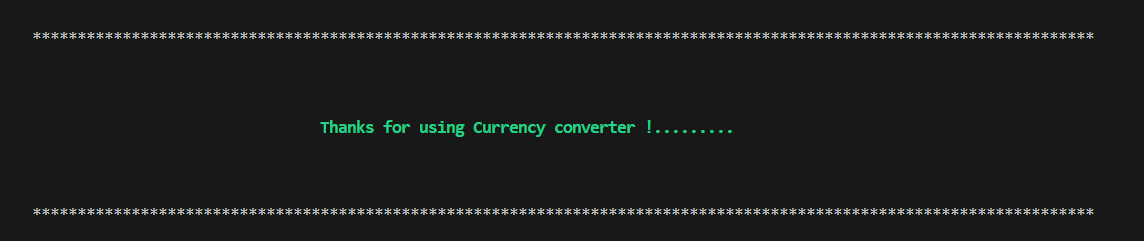












SOURCE CODE

#include <stdio.h>

#include <conio.h>

#include <stdlib.h>

#include <string.h>

void printWelcomePage();

int login();

float getConversionRate();

void printCurrencyList();

int getValidatedInput();

int currencyConverter();

float getValidatedAmount();

// Function  welcome page

void printWelcomePage()

{

printf("\n\n\n\n\t\033[0m\*----\*------\*-------\*-------\*---------\*--------\*--------\*--------\*--------\*--------\*--------\*--------\*--------\*--------\*\n");

printf("\t\*                                                                                                                      \*\n");

printf("\t\033[1;33m\*\t\t\t\t>>>>>>>  \* KAMADGIRI SOFTWARE SOLUTIONS \*  <<<<<<<\t\t\t\t       \*\n");

printf("\t\033[0m|\t\t\t\t\t  -------\*-----\*------\*--------\t\t\t\t\t\t       |\n");

printf("\t\*                                                                                                                      \*\n");

printf("\t\*                                                                                                                      \*\n");

printf("\t\033[1;33m|\t\t\t\t\t\*Project Name:- $Currency Converter.$\t\t\t\t\t       |\n");

printf("\t\033[0m\*\t\t\t\t\t\*--------\*-------\*---------\*--------\*\t\t\t\t\t       \*\n");

printf("\t|                                                                                                                      |\n");

printf("\t\*                                                                                                                      \*\n");

printf("\t|                                                                                                                      |\n");

printf("\t\*                                                                                                                      \*\n");

printf("\t\033[1;33m|\t\tDate:-                                                                                                 |\n");

printf("\t\033[1;33m\*\t\tSUBMITTED BY:\t\t\t\t\t\t\t\tSubmitted To:-\t\t       \*\n");

printf("\t\033[0m\*\t\t--\*----\*---\*--\t\t\t\t\t\t\t\t--\*----\*---\*--\*\t\t       |\n");

printf("\t|                                                                                                                      |\n");

printf("\t\033[0m|\t\t1.Aman Rajak-Team Leader.\t\t\t\t\t\tEr.Satyam Mishra \t       \*\n");

printf("\t|\t\t2.Himanshu                                                                                             |\n");

printf("\t\*\t\t3.Dheeraj                                                                                              \*\n");

printf("\t|\t\t4.Nikhil                                                                                               |\n");

printf("\t\*\t\t5.Anjali                                                                                               \*\n");

printf("\t|\t\t6.Vandana                                                                                              |\n");

printf("\t\*\t\t7.Poonam                                                                                               \*\n");

printf("\t\*                                                                                                                      \*\n");

printf("\t\*                                                                                                                      \*\n");

printf("\t\*                                                                                                                      \*\n");

printf("\t------\*--------\*--------\*---------\*--------\*--------\*--------\*--------\*--------\*--------\*--------\*--------\*------\*-----\*\n\n");

}

//  login ke liye

int login()

{

char correctUsername[] = {"admin"};

char correctPassword[] = {"12345"};

char inputUsername[20];

char inputPassword[20];

int loginSuccess = 0, len;

char retryChoice;

int i;

do

{

printf("\n\t\033[0m~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\*\*\* >>>>>> Login Here <<<<<< \*\*\*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n\n");

// User name

printf("\t\t\t\t\t\t \033[1;33m    \* >>   Enter username :  ");

gets(inputUsername);

len = strlen(correctPassword);

// usere password

printf(" \t\t\t\t\t             \* >>   Enter password :  ");

for ( i = 0; i < len; i++)

{

inputPassword[i] = getch();

putch('\*'); // change Enter Password to \* symbol.

}

inputPassword[len] = '\0'; //

// Check user name or password

if (strcmp(inputUsername, correctUsername) == 0 && strcmp(inputPassword, correctPassword) == 0)

{

loginSuccess = 1;

}

//  system("cls"); // screen clear krne ke liye

if (loginSuccess)

{

printf("\n\n\t\t\t\t\t\t\033[1;32mLogin successful! Welcome, %s.\n\n", inputUsername);

// printf("\033[0m\n\t\t\t\t\t\tPress Any Key to Continue....");

// getchar();

}

else

{

printf("\n\n\t\t\t\t\t\033[1;31mLogin failed. Invalid username or password.\n\n");

printf("\033[0m~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Access denied ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n");

printf("\033[1;33m\t\tDo you want to try logging  again?: press Y otherwise press any key (y/any key) ");

retryChoice = getch();

printf("%c\n", retryChoice);

}

} while (!loginSuccess && (retryChoice == 'y' || retryChoice == 'Y'));

if (!loginSuccess)

{

printf("\n\033[1;31m\t\tAccess denied. You chose not to retry.\n");

}

return loginSuccess;

}

// Function to get the conversion rate between two selected currencies using switch-case

float getConversionRate(int from, int to)

{

float rate = 1;

switch (from)

{

case 1: // USD to others

switch (to)

{

case 2:

rate = 83.85;

break; // USD to INR

case 3:

rate = 0.90;

break; // USD to EUR

case 4:

rate = 119.56;

break; // USD to BDT

case 5:

rate = 1.48;

break; // USD to AUD

case 6:

rate = 1.36;

break; // USD to CAD

case 7:

rate = 140.75;

break; // USD to JPY

case 8:

rate = 7.09;

break; // USD to CNY

case 9:

rate = 17.62;

break; // USD to ZAR

case 10:

rate = 5.52;

break; // USD to BRL

case 11:

rate = 1.62;

break; // USD to NZD

case 12:

rate = 91.40;

break; // USD to RUB

}

break;

case 2: // INR to others

switch (to)

{

case 1:

rate = 0.012;

break; // INR to USD

case 3:

rate = 0.011;

break; // INR to EUR

case 4:

rate = 1.43;

break; // INR to BDT

case 5:

rate = 0.018;

break; // INR to AUD

case 6:

rate = 0.016;

break; // INR to CAD

case 7:

rate = 1.68;

break; // INR to JPY

case 8:

rate = 0.085;

break; // INR to CNY

case 9:

rate = 0.21;

break; // INR to ZAR

case 10:

rate = 0.066;

break; // INR to BRL

case 11:

rate = 0.019;

break; // INR to NZD

case 12:

rate = 1.09;

break; // INR to RUB

}

break;

case 3: // EUR to others

switch (to)

{

case 1:

rate = 1.11;

break; // EUR to USD

case 2:

rate = 93.30;

break; // EUR to INR

case 4:

rate = 133.04;

break; // EUR to BDT

case 5:

rate = 1.65;

break; // EUR to AUD

case 6:

rate = 1.51;

break; // EUR to CAD

case 7:

rate = 156.53;

break; // EUR to JPY

case 8:

rate = 7.89;

break; // EUR to CNY

case 9:

rate = 19.61;

break; // EUR to ZAR

case 10:

rate = 6.14;

break; // EUR to BRL

case 11:

rate = 1.80;

break; // EUR to NZD

case 12:

rate = 101.70;

break; // EUR to RUB

}

break;

case 4: // BDT to Other

switch (to)

{

case 1:

rate = 0.0084;

break; // BDT to USD

case 2:

rate = 0.70;

break; // BDT to INR

case 3:

rate = 0.0075;

break; // BDT to INR

case 5:

rate = 0.012;

break; // BDT to AUD

case 6:

rate = 0.011;

break; // BDT to CAD

case 7:

rate = 1.18;

break; // BDT to JPY

case 8:

rate = 0.059;

break; // BDT to CNY

case 9:

rate = 0.15;

break; // BDT to ZAR

case 10:

rate = 0.047;

break; // BDT to BRL

case 11:

rate = 0.014;

break; // BDT to NZD

case 12:

rate = 0.76;

break; // BDT to RUB

}

break;

case 5: // AUD to other

switch (to)

{

case 1:

rate = 0.67;

break; // AUD to USD

case 2:

rate = 56.24;

break; // AUD to INR

case 3:

rate = 0.61;

break; // AUD to EUR

case 4:

rate = 80.14;

break; // AUD to BDT

case 6:

rate = 0.91;

break; // AUD to CAD

case 7:

rate = 94.59;

break; // AUD to JPY

case 8:

rate = 4.76;

break; // AUD to CNY

case 9:

rate = 11.90;

break; // AUD to ZAR

case 10:

rate = 3.73;

break; // AUD to BRL

case 11:

rate = 1.09;

break; // AUD to NZD

case 12:

rate = 61.64;

break; // AUD to RUB

}

break;

case 6: // CAD to USD

switch (to)

{

case 1:

rate = 0.74;

break; // CAD to USD

case 2:

rate = 61.69;

break; // CAD to INR

case 3:

rate = 0.66;

break; // CAD to EUR

case 4:

rate = 87.92;

break; // CAD to BDT

case 5:

rate = 1.10;

break; // CAD to AUD

case 7:

rate = 103.63;

break; // CAD to JPY

case 8:

rate = 5.22;

break; // CAD to CNY

case 9:

rate = 13.06;

break; // CAD to ZAR

case 10:

rate = 4.09;

break; // CAD to BRL

case 11:

rate = 1.19;

break; // CAD to NZD

case 12:

rate = 67.23;

break; // CAD to RUB

}

break;

case 7: // JPY to Other

switch (to)

{

case 1:

rate = 0.0071;

break; // JPY to USD

case 2:

rate = 0.60;

break; // JPY to INR

case 3:

rate = 0.0064;

break; // JPY to EUR

case 4:

rate = 0.85;

break; // JPY to BDT

case 5:

rate = 0.011;

break; // JPY to AUD

case 6:

rate = 0.0096;

break; // JPY to CAD

case 8:

rate = 0.050;

break; // JPY to CNY

case 9:

rate = 0.13;

break; // JPY to ZAR

case 10:

rate = 0.039;

break; // JPY to BRL

case 11:

rate = 0.011;

break; // JPY to NZD

case 12:

rate = 0.65;

break; // JPY to RUB

}

break;

case 8: // CNY to other

switch (to)

{

case 1:

rate = 0.14;

break; // CNY to USD

case 2:

rate = 11.83;

break; // CNY to INR

case 3:

rate = 0.13;

break; // CNY to EUR

case 4:

rate = 16.85;

break; // CNY to BDT

case 5:

rate = 0.21;

break; // CNY to AUD

case 6:

rate = 0.19;

break; // CNY to CAD

case 7:

rate = 19.85;

break; // CNY to JPY

case 9:

rate = 2.50;

break; // CNY to ZAR

case 10:

rate = 0.78;

break; // CNY to BRL

case 11:

rate = 0.23;

break; // CNY to NZD

case 12:

rate = 12.89;

break; // CNY to RUB

}

break;

case 9: // ZAR to Other

switch (to)

{

case 1:

rate = 0.056;

break; // ZAR to USD

case 2:

rate = 4.73;

break; // ZAR to INR

case 3:

rate = 0.051;

break; // ZAR to EUR

case 4:

rate = 6.73;

break; // ZAR to BDT

case 5:

rate = 0.084;

break; // ZAR to AUD

case 6:

rate = 0.77;

break; // ZAR to CAD

case 7:

rate = 7.93;

break; // ZAR to JPY

case 8:

rate = 0.40;

break; // ZAR to CNY

case 10:

rate = 6.73;

break; // ZAR to BDT

case 11:

rate = 0.092;

break; // ZAR to NZD

case 12:

rate = 5.19;

break; // ZAR to RUB

}

break;

case 10: // BRL to Other

switch (to)

{

case 1:

rate = 0.18;

break; // BRL to USD

case 2:

rate = 15.07;

break; // BRL to INR

case 3:

rate = 0.16;

break; // BRL to  EUR

case 4:

rate = 21.48;

break; // BRL to BDT

case 5:

rate = 0.27;

break; // BRL to AUD

case 6:

rate = 0.24;

break; // BRL to CAD

case 7:

rate = 25.31;

break; // BRL to JPY

case 8:

rate = 1.27;

break; // BRL to CNY

case 9:

rate = 3.19;

break; // BRL to ZAR

case 11:

rate = 0.29;

break; // BRL to NZD

case 12:

rate = 16.56;

break; // BRL to RUB

}

break;

case 11: // NZD to Other

switch (to)

{

case 1:

rate = 0.62;

break; // NZD to USD

case 2:

rate = 51.84;

break; // NZD to INR

case 3:

rate = 0.56;

break; // NZD to  EUR

case 4:

rate = 73.92;

break; // NZD to BDT

case 5:

rate = 0.32;

break; // NZD to AUD

case 6:

rate = 0.24;

break; // NZD to CAD

case 7:

rate = 87.08;

break; // NZD to JPY

case 8:

rate = 4.39;

break; // NZD to CNY

case 9:

rate = 10.90;

break; // NZD to ZAR

case 10:

rate = 3.41;

break; // NZD to BRL

case 12:

rate = 56.53;

break; // NZD to RUB

}

break;

case 12: // RUB to Other

switch (to)

{

case 1:

rate = 0.011;

break; // RUB to USD

case 2:

rate = 0.92;

break; // RUB to INR

case 3:

rate = 0.0098;

break; // RUB to  EUR

case 4:

rate = 1.31;

break; // RUB to BDT

case 5:

rate = 0.016;

break; // RUB to AUD

case 6:

rate = 0.015;

break; // RUB to CAD

case 7:

rate = 1.54;

break; // RUB to JPY

case 8:

rate = 0.078;

break; // RUB to CNY

case 9:

rate = 0.19;

break; // RUB to ZAR

case 10:

rate = 0.060;

break; // RUB to BRL

case 11:

rate = 0.018;

break; // RUB to NZD

}

break;

default:

printf("Invalid currency selection.\n");

return -1;

}

return rate;

}

// Function to print the currency list

void printCurrencyList()

{

printf("\n\n\033[0m\t\t\t\t\t\*======================================================\*\n");

printf("\033[1;33m\t\t\t\t\t|  Code  | Currency Name    | Country / Region         |\n");

printf("\033[0m\t\t\t\t\t|--------|------------------|--------------------------|\n");

printf("\t\t\t\t\t\*   1    | USD              | United States            \*\n");

printf("\t\t\t\t\t|   2    | INR              | India                    |\n");

printf("\t\t\t\t\t\*   3    | EUR              | European Union           \*\n");

printf("\t\t\t\t\t|   4    | BDT              | Bangladeshi Taka         |\n");

printf("\t\t\t\t\t\*   5    | AUD              | Australia                \*\n");

printf("\t\t\t\t\t|   6    | CNY              | China                    |\n");

printf("\t\t\t\t\t\*   7    | JPY              | Japan                    \*\n");

printf("\t\t\t\t\t|   8    | CAD              | Canada                   |\n");

printf("\t\t\t\t\t\*   9    | ZAR              | South Africa             \*\n");

printf("\t\t\t\t\t|   10   | BRL              | Brazilial Rual           |\n");

printf("\t\t\t\t\t\*   11   | NZD              | New Zealand Dollar       \*\n");

printf("\t\t\t\t\t|   12   | RUB              | Russian Ruble            |\n");

printf("\033[0m\t\t\t\t\t\*======================================================\*\n\n");

}

// amount validation ke liye

float getValidatedAmount()

{

float amount;

char term;

while (1)

{

printf("\n\033[1;32m\t\t\t\t\tEnter Amount: ");

// Validate floating point input

if (scanf("%f%c", &amount, &term) == 2 && term == '\n')

{

return amount; // valid input, return amount

}

else

{

printf("\033[1;31m\t\t\t\t\t Invalid input. Please enter a valid amount.\n");

while (getchar() != '\n'); // clear input buffer

}

}

}

// Function to validate numeric input

int getValidatedInput(int min, int max, const char \*massage)

{

int input;

char term;

while (1)

{

printf("\n");

// printf("\033[1;31m\t\t\t\t\tEnter your choice for the source currency: ");

printf("\033[1;31m\t\t\t\t\t %s", massage);

if (scanf("%d%c", &input, &term) == 2 && term == '\n' && input >= min && input <= max)

{

break; // valid input, exit loop

}

else

{

printf("\033[1;31m\t\t\t\t\t Invalid input. Please enter a valid number between %d and %d.\n", min, max);

while (getchar() != '\n')

; // clear input buffer

}

}

return input;

}

// Currency Converter functionality

int currencyConverter()

{

int fromChoice, toChoice;

float amount, conversionRate, convertedAmount;

char continueFlag = 'y';

char \*currencies[12] = {

"USD ", "INR ", "EUR ", "BDT ", "AUD ",

"CAD ", "JPY ", "CNY ", "ZAR ", "BRL ", "NZD", "RUB"};

while (continueFlag == 'y' || continueFlag == 'Y')

{

printf("\033[0m\n\t~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ \*\*\* >> Welcome To World of Currency << \*\*\* ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n\n");

// Display the currency choices

printf("\033[1;33m\n\t\t\t\t\tSelect the currency to convert FROM (choose the Country Code):\n");

printf("\033[0m\t\t\t\t\t---------------------\*-------------\*------------------");

// Display the currency choices

printCurrencyList();

// Get validation for source currency

printf("\033[0m\t\t\t\t\t\t--------------\*-----------\*-------------\n");

fromChoice = getValidatedInput(1, 12, "\033[1;33m=> Enter your choice for the source currency code: ");

// Get validation for destination currency

toChoice = getValidatedInput(1, 12, "\033[1;33m\n\t\t\t\t\t=> Enter your choice for the destination currency code: ");

amount= getValidatedAmount();

// Get conversion rate and check if fromChoice and toChoice are the same

if (fromChoice == toChoice)

{

printf("\033[1;31m\n\n\t\t\t\t\t\t Same Currency no need to  convert..!  \n");

}

else

{

conversionRate = getConversionRate(fromChoice, toChoice);

if (conversionRate != -1)

{

// formula kaise kam krega

convertedAmount = amount \* conversionRate;

printf("\033[1;33m\n\t\t\t\t\t\t\t\* Converted Amount:\033[1;33m %.2f %s\n", convertedAmount, currencies[toChoice - 1]);

printf("\033[0m\t\t\t\t\t\t--------------\*-----------\*-------------");

}

}

printf("\033[0m\n ----------------------------------------------------------------------------------------------------------------------\n\n"); // Print border

printf("\033[1;33m\t\t\t If you want to convert again  press Y otherwise press any key (y/press any key):\033[1;31m  ");

scanf(" %c", &continueFlag);

}

system("cls");

printf("\033[0m\n\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n\n");

printf("\033[1;32m\n\t\t\t\t\tThanks for using Currency converter !.........\n\n\n");

printf("\033[0m\n\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");

}

int main()

{

//  Display the Welcome Page

printWelcomePage();

//  Perform User Login

if (login())

{

// If login successful

currencyConverter();

}

else

{

printf("\033[1;0m\n\t\t\t\t\tExiting program due to failed login.\n");

}

return 0;

}