CURRENCY CONVERTER

Project report submitted in partial fulfilment of the requirements for the Award of the degree of   
BACHELOR OF TECHNOLOGY  
In

COMPUTER SCIENCE AND ENGINEERING  
By

M. Manasa 24KB1A05AU

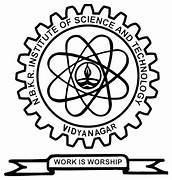
P. Ruchika 24KB1A05EL

P. Amrutha Varshini 24KB1A05EG

N. Tejaswi 24KB1A05BW

Under the Guidance of

P. Suneetha



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

N.B.K.R.IST

N.B.K.R. INSTITUTE OF SCIENCE AND TECHNOLOGY

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

A black and white logo

AI-generated content may be incorrect.

**CERTIFICATE**

This is to certify that the project report entitled

CURRENCY CONVERTER being submitted by

M. Manasa 24KB1A05AU

P. Ruchika 24KB1A05EL

P. Amrutha Varshini 24KB1A05EG

N. Tejaswi 24KB1A05BW

In partial fulfilment for the award of the degree of Bachelor of technology in computer science and engineering to the N.B.K.R.IST is a record of Bonafede work carried out under my Guidance and supervision.

P. Suneetha Dr. A. Raja Sekhar Reddy

Designation M. Tech, Ph. D

Head of the department.

DECLARATION

I hereby declare that the dissertation entitled currency converter submitted for the B. Tech degree is my original work and the dissertation has not formed the basis for the award of any degree, associateship, fellow or any other similar titles.

Place: Vidyanagar

Date:7 may 2025

M. Manasa 24KB1A05AU

P. Ruchika 24KB1A05EL

P. Amrutha Varshini 24KB1A05EG

N. Tejaswi 24KB1A05BW

**Acknowledgment:**

I would like to acknowledge the effort and learning that went into developing this Currency Converter program in C. This project provided practical experience with fundamental programming concepts including structures, arrays, string manipulation, dynamic memory allocation, and linked lists. It also helped reinforce skills in creating menu-driven interfaces and handling user input. The implementation of a conversion history feature added value by demonstrating how data can be stored and retrieved dynamically during program execution. This project served as an excellent opportunity to apply theoretical knowledge to a functional and user-friendly application.

**Abstract:**

This project presents a console-based Currency Converter application developed in the C programming language. The primary objective of the program is to facilitate currency conversions between five predefined currencies—USD, EUR, JPY, INR, and GBP—using static exchange rates. The system allows users to input source and target currency codes, enter the amount to be converted, and receive an immediate conversion result. An additional feature maintains a dynamic linked list to store and display a history of all conversions performed during the session. This project showcases practical applications of core C programming concepts including structures, arrays, string handling, functions, pointers, dynamic memory allocation, and linked lists. It is designed for ease of use with a menu-driven interface, providing a functional and educational example of data processing and memory management in C.

**Introduction:**

This project is a console-based Currency Converter developed in C, designed to convert amounts between five predefined currencies: USD, EUR, JPY, INR, and GBP, using static exchange rates. It features a simple menu-driven interface and maintains a session-based conversion history using a linked list. The program demonstrates key C programming concepts such as structures, arrays, dynamic memory allocation, and pointers. It serves as both a practical tool and a learning exercise in implementing real-world functionality with core C features.

**Literature Survey / Existing System:**

Currency converters are widely used in financial applications, travel services, and e-commerce platforms to provide real-time currency exchange values. Existing systems, such as online converters (e.g., XE, OANDA, Google Finance), use APIs to fetch live exchange rates and offer multi-currency support with advanced features like historical trends, charts, and mobile integration.

**Software Requirements and Analysis:**

Functional Requirements:

* Input source and target currency codes
* Input the amount to be converted
* Perform currency conversion using static exchange rates
* Display conversion result
* Store and display conversion history

Non-Functional Requirements:

* User-friendly, menu-based interface
* Fast and responsive input/output operations
* Efficient memory usage with dynamic allocation
* Portable across platforms with a C compiler

**Software Design:**

The Currency Converter is designed using a modular, procedural approach in C. It includes three main modules: a menu-driven main module, a conversion module using static rates, and a history module using a linked list for storing past conversions. Data is structured using struct types, and memory is dynamically managed with malloc. The user interface is simple and text-based, providing clear options for conversion, history viewing, and exit.

**Proposed System:**

The proposed system is an offline, console-based currency converter developed in C. It allows users to convert amounts between five predefined currencies using static exchange rates. The system also maintains a conversion history during runtime using a linked list. Unlike online converters, this tool focuses on implementing core C programming concepts such as structures, dynamic memory allocation, and menu-driven interaction, making it ideal for learning and demonstration purposes.

**Coding:**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX\_CURRENCIES 5

// Currency Rate Structure

typedef struct {

char code[4];

float rate\_to\_usd;

} Currency;

// Conversion History Node

typedef struct HistoryNode {

char from\_currency[4];

char to\_currency[4];

float amount;

float result;

struct HistoryNode\* next;

} HistoryNode;

// Currency Rates Array

Currency currencies[MAX\_CURRENCIES] = {

{"USD", 1.0},

{"EUR", 0.91},

{"JPY", 155.12},

{"INR", 83.33},

{"GBP", 0.78}

};

// History Head Pointer

HistoryNode\* history\_head = NULL;

// Function to find currency rate

float get\_rate(char code[]) {

for (int i = 0; i < MAX\_CURRENCIES; i++) {

if (strcmp(currencies[i].code, code) == 0)

return currencies[i].rate\_to\_usd;

}

return -1; // Not found

}

// Add to conversion history

void add\_to\_history(char from[], char to[], float amount, float result) {

HistoryNode\* new\_node = (HistoryNode\*)malloc(sizeof(HistoryNode));

strcpy(new\_node->from\_currency, from);

strcpy(new\_node->to\_currency, to);

new\_node->amount = amount;

new\_node->result = result;

new\_node->next = history\_head;

history\_head = new\_node;

}

// Display conversion history

void print\_history() {

HistoryNode\* temp = history\_head;

printf("\nConversion History:\n");

while (temp != NULL) {

printf("%.2f %s = %.2f %s\n", temp->amount, temp->from\_currency, temp->result, temp->to\_currency);

temp = temp->next;

}

}

// Convert currency

float convert(char from[], char to[], float amount) {

float from\_rate = get\_rate(from);

float to\_rate = get\_rate(to);

if (from\_rate == -1 || to\_rate == -1) {

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

return -1;

}

float usd\_amount = amount / from\_rate;

return usd\_amount \* to\_rate;

}

// Main Program

int main() {

char from[4], to[4];

float amount, result;

int choice;

do {

printf("\nCurrency Converter\n");

printf("1. Convert Currency\n");

printf("2. View Conversion History\n");

printf("3. Exit\n");

printf("Enter choice: ");

scanf("%d", &choice);

switch(choice) {

case 1:

printf("Enter from currency code (e.g., USD): ");

scanf("%s", from);

printf("Enter to currency code (e.g., EUR): ");

scanf("%s", to);

printf("Enter amount: ");

scanf("%f", &amount);

result = convert(from, to, amount);

if (result != -1) {

printf("Converted: %.2f %s = %.2f %s\n", amount, from, result, to);

add\_to\_history(from, to, amount, result);

}

break;

case 2:

print\_history();

break;

case 3:

printf("Exiting...\n");

break;

default:

printf("Invalid choice.\n");

}

} while (choice != 3);

return 0;

}

**Output:**

A screenshot of a computer

AI-generated content may be incorrect.

**Conclusion:**

The Currency Converter project successfully demonstrates the implementation of a functional and user-friendly console application using the C programming language. It provides accurate currency conversions between five predefined currencies and maintains a session-based history using a linked list. This project highlights key programming concepts such as structures, pointers, dynamic memory allocation, and modular design. Overall, it serves as a practical and educational tool for understanding how real-world applications can be built using fundamental C techniques.

**References:**

1. Programming in ANSI C by E. Balagurusamy – McGraw Hill Education

2. Let Us C by Yashavant Kanetkar – BPB Publications

3. TutorialsPoint – C Programming Language: https://www.tutorialspoint.com/cprogramming

4. GeeksforGeeks – C Programming: https://www.geeksforgeeks.org/c-programming-language

5. Currency exchange rate sources for reference (for static rates):

XE Currency Converter: https://www.xe.com

OANDA Currency Converter: https://www.oanda.com

THANK YOU