**CREDIT CARD BILL PAYMENT SYSTEM**

***Github Link*:** https://github.com/Yashitha-Gillella/projects.git

***DONE BY:* Yashitha,Gnana Priya,Neharika,**

**Sowjanya,Bhavitha.**

**Contents**

1. **Introduction**

* **Title**
* **Objective**
* **Purpose**
* **Significance**

1. **System Design and Work flow**

* **Program Structure**
* **System Flow**
* **Key components**
* **Interface Description**

1. **Programming Language**

* **Input Fields**
* **Core Variables**
* **Output Description**

1. **Functional and Non-Functional Features**

* **Functional Features**
* **Non-Functional Features**
* **System Environment**

1. **Project Evaluation**

* **Evaluation**
* **Educational Values**

1. **Conclusion**

**CREDIT CARD BILL PAYMENT SYSTEM**

**Language**: C  
**Platform**: Console (Command Line Interface)  
**Pages**: 5

**Introduction**

**1.1 Title**

**Credit Card Bill Payment System Using C Programming**

**1.2 Objective**

The objective of this project is to simulate a credit card bill payment system using the C programming language. It enables users to input their card and billing details and receive a calculated final balance after making a payment. The program mimics a basic financial transaction and displays a formatted receipt to the user.

**1.3 Purpose**

This project aims to provide an understanding of how simple transaction-based systems work. It helps learners practice core programming concepts such as data collection, arithmetic operations, and formatted output. The application is designed to resemble real-world usage but in a controlled, academic environment.

**1.4 Significance**

The significance of this program lies in its practical approach to teaching basic C programming. It offers a hands-on simulation of a financial task, reinforcing programming logic, and giving students an opportunity to build foundational skills that can be extended to more complex systems.

Page 3

**System Design and Workflow**

**2.1 Program Structure**

The program follows a linear and procedural design. The structure is divided into three primary phases:

* **Input Phase**: Collects the necessary information from the user.
* **Processing Phase**: Performs the payment calculation.
* **Output Phase**: Displays a formatted transaction summary or receipt.

**2.2 System Flow**

1. User is welcomed and prompted to enter details such as credit card number, CVV, outstanding amount, cardholder name, date, and payment amount.
2. The system processes this data to compute the remaining balance.
3. A transaction summary is displayed, presenting all user input and calculated information in an organized format.

**2.3 Key Components**

* **User Input Interface**: Handles the collection of input data.
* **Calculation Module**: Computes the final balance after deduction.
* **Report Generator**: Produces a structured and readable payment statement.

**2.4 Interface Description**

The program runs in a console window, interacting with the user through clear prompts and text-based menus. Receipt is printed using formatted text output, mimicking a digital or paper receipt format.

**Page 4**

**3.1 Programming Language**

The application is written in C, making use of standard library functions for input/output operations, data formatting, and arithmetic processing.

**3.2 Input Fields**

The following fields are collected from the user:

* **Credit Card Number**: Numeric input typically consisting of 14 digits.
* **CVV**: A 4-digit security code.
* **Current Outstanding Amount**: The total due before payment.
* **Card Holder Name**: The name of the user making the payment.
* **Payment Date**: Entered in string format (e.g., DD-MM-YYYY).
* **Payment Amount**: Amount the user intends to pay.

**3.3 Core Variables (Logical Description)**

* A string to hold the cardholder's name.
* A string for the date of payment.
* Floating-point numbers to store outstanding amount, deducted amount, and final balance.
* Long integers to store credit card number and CVV code.

**3.4 Output Description**

After processing the inputs, the system prints a formatted bill containing all the entered details along with the final amount due. The design imitates a standard financial transaction receipt.

Page 5

**4.Functional and Non-Functional Features**

**4.1 Functional Features**

* **Input Handling**: Prompt and accept the required payment information from the user.
* **Balance Calculation**: Subtract the entered payment from the outstanding balance.
* **Formatted Output**: Print a clearly structured bill receipt with all the necessary transaction details.

**4.2 Non-Functional Features**

* **Usability**: The program should be easy to operate and understand for users with basic computing skills.
* **Performance**: The system must handle inputs and display results quickly for a single user.
* **Maintainability**: The code should be modular and easy to expand for future features.
* **Portability**: The program should run on any C-compatible environment across Windows, Linux, or macOS platforms.

**4.3 System Environment**

* **Operating System**: Windows/Linux/macOS
* **Compiler**: GCC, Turbo C++, or any ANSI C-compatible compiler
* **Interface**: Console/Terminal

Page 6

**5 Project Evaluation**

**5.1 Evaluation**

The project successfully meets its goal of simulating a basic credit card bill payment system. It handles the full cycle of a financial transaction: input collection, processing, and result display. The output is neatly formatted to resemble a printed bill, enhancing the user experience within a console environment.

**5.2 Educational Value**

This project is particularly useful for beginners to learn about:

* Input/output operations in C
* Handling numeric and text data
* Implementing basic transaction logic
* Using formatted print statements to structure console output

**6 Conclusion**

The Credit Card Bill Payment System in C is a valuable academic project that illustrates the intersection of practical logic with programming fundamentals. It provides learners a base to explore transaction-based application development and fosters a deeper understanding of data processing and structured output in console applications.

Page 7

**Final code:**

#include<stdio.h>

int main()

{

char name[50],date[20];

int opt; //name=account holder name ; date=bill payment date

double amt,ded,coa; //ded=amount deducted; amt=final amount after dedcting; coa=current outstanding amount

long long int cn,cvv; //cn=card number; cvv=cvv number

printf("\t\t----------------Credit Card bill payment -------------------\n");

printf(" Enter your credit card number(14 digits) : ");

scanf("%lld",&cn);

printf(" cvv(4 digits) : ");

scanf("%lld",&cvv);

printf(" Current outstanding amount[maximum amount is 1 lakh] : ");

scanf("%lf",&coa);

printf(" Enter card holder name : "); //printing account holder details

scanf("%s",&name);

printf("\n\t\t card holder name : %s \n",name);

printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n" );

printf(" credit card details |\n");

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

printf("|Card Number : %lld |\n",cn);

printf("|CVV : %lld |\n",cvv);

printf("|Current Outstanding Amount : %.2lf |\n",coa);

page 8

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

printf("Check whether the details given above are correct[type 1 for yes , 0 for no] :");

scanf("%d",&opt);

if(opt==1)

{

printf("\n Enter today date: ");

scanf("%s",date);

printf("How much amount do you want to pay[minimum amount is 10000]? ");

scanf("%lf",&ded);

amt=coa-ded; //calculating amount deducted

printf("Payment Successful!!!!\n"); //printing final bill statement

printf(" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf(" -----------------------Final Bill Statement ----------------------|\n",name);

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

printf("|Card Number : %lld |\n",cn);

printf("|CVV : %lld |\n",cvv);

printf("|Bill Date : %s |\n",date);

printf("|Current Outstanding Amount : %.2lf |\n",coa);

printf ("|Deducted Amount : %.2lf |\n",ded);

printf("|Final Balance : %.2lf |\n",amt);

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

}

else if(opt==0)

{

printf("please enter your details again :");

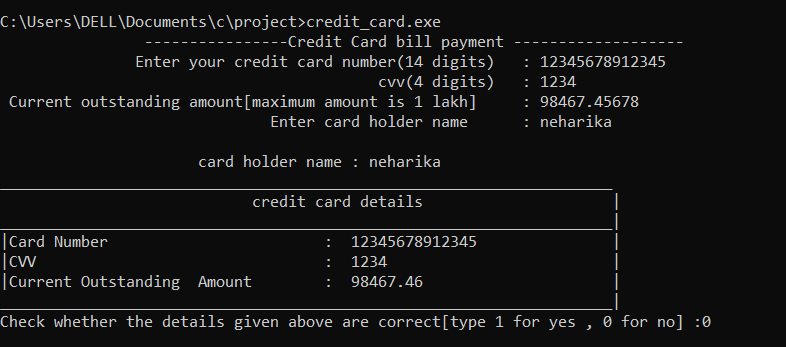
}

return 0;

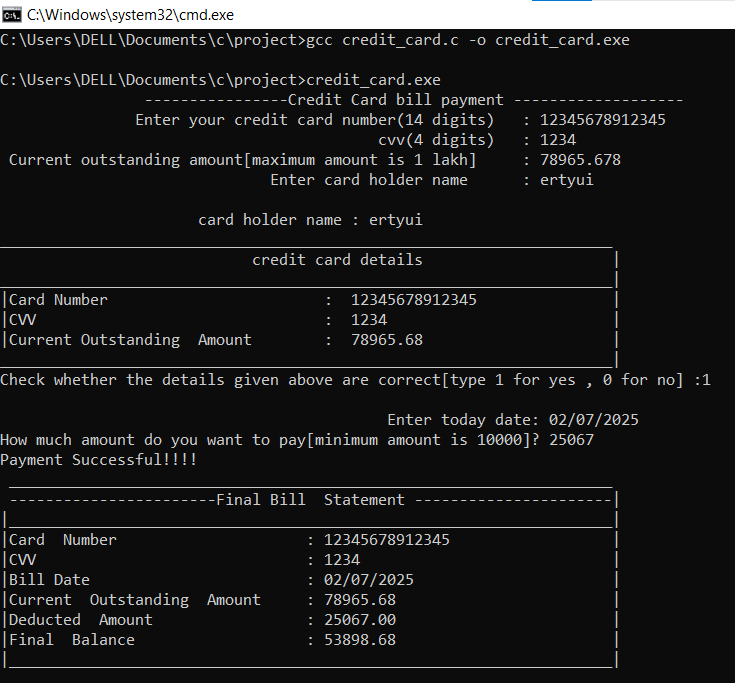
}

Page 9

Sample output 1:



Sample output 2:



Page 10