

**FINAL PROJECT**

**SEMESTER 1st**

**BANK MANAGEMENT SYSTEM**

**BY**

**ROHAN BALAMI**

**BACHELOR OF INFORMATION AND COMMUNICATION**

**TECHNOLOGY SCHOOL OF SCIENCE AND TECHNOLOGY**

**ASIA e UNIVERSITY**

**This report is prepared to fulfil the requirement of CPP203 Final Project**

**BY**

**ROHAN BALAMI**

**SCHOOL OF SCIENCE AND TECHNOLOGY ASIA e UNIVERSITY**

**SEMESTER 1ST**

**DISCLAIMER**

**CCP203 – FINAL PROJECT**

I am responsible for the accuracy of all opinions, technical comments, factual report, data, figures, illustrations and photographs highlighted in this report. I bear full responsibility that the report submitted has been reviewed and subject to copyright or ownership rights. Asia e University will not bear any liability for the accuracy of any comment, report and other technical and factual information, and the copyright or ownership right claims.

ROHAN BALAMI

C30101230038

**ACKNOWLEDGEMENT**

I would like to express my heartfelt gratitude to Mr. Roshan KC, the Coordinator of Virinchi College, for allowing us permission to carry out on Bank Management System project. Er. Animesh Regmi is sincerely appreciated for his tremendous direction, supervision, and support during the project. I am also grateful to my friends, family, and others for their ongoing assistance, cooperation, and support in completing this project in such a short period of time. This project has been a rewarding learning experience as well as a useful asset to my future profession. Mr. Animesh Regmi, our guide and mentor, deserves special recognition for pushing us to seek solutions and for consistently editing and redesigning our report. I am also appreciative to my classmates for their insightful comments.

Thank you very much!

**ABSTRACT**

The Bank Management System is a software solution designed to address the challenges faced by banks in efficiently managing their operations while ensuring data security. The system offers features such as customer account management and transaction tracking. By replacing manual paperwork with digital records, the software streamlines the entire banking process, reducing the chances of errors and improving overall efficiency. Moreover, it provides an additional layer of security through login verification, ensuring authorized access to sensitive banking information. The Bank Management System employs file-based data storage, eliminating the need for complex database management systems. This design choice simplifies the installation and maintenance process while maintaining data integrity and reliability. By embracing technology and automating core banking processes, the Bank Management System offers a reliable and efficient solution to modernize the banking industry. This project contributes to the digital transformation of banking operations, leading to enhanced productivity, streamlined workflows, and improved data security.

**TABLE OF CONTENTS**

1. **INTRODUCTION ...................................................................................................... 1**
2. **MOTIVATION ........................................................................................................... 2**
3. **OBJECTIVE............................................................................................................... 3**
4. **FEATURE OF PROJECT ......................................................................................... 4**
5. **LITERATURE REVIEW .......................................................................................... 5**
6. **PROBLEM STATEMENT ........................................................................................ 6**
7. **ALGORITHM ............................................................................................................ 7**
8. **FLOWCHART ........................................................................................................... 8**
9. **IMPLEMENTATION ................................................................................................ 9**
10. **FUTURE ENHANCEMENTS ................................................................................. 10**
11. **OUTPUT SCREEN .................................................................................................. 11**
12. **SOURCE CODE ........................................................................................................ 19**
13. **REFERENCE ............................................................................................................. 46**

**1. INTRODUCTION**

The Bank Management System project aims to revolutionize the traditional manual methods used in banks by implementing an efficient and secure software solution developed in the C programming language. This project leverages the capabilities of C to create a comprehensive system that addresses the needs of both bank administrators and customers.

Similar to the Other Management System, which streamlines book management in libraries, the Bank Management System provides a digital platform for storing and managing crucial banking information. By digitizing data storage and eliminating the reliance on handwritten documents, this system significantly improves the efficiency and accuracy of banking operations. The Bank Management System consists of two primary modules: the Admin/Staff module and the User module. The Admin/Staff module serves as the backbone of the system, enabling administrators to oversee and manage the overall banking processes. On the other hand, the User module empowers customers to access and utilize various banking services offered by the system.

To ensure the security and integrity of the data, the system incorporates a login functionality, preventing unauthorized access and manipulation of sensitive information. This feature safeguards the confidentiality and reliability of the stored banking data. With the implementation of this Bank Management System, the administration can easily and efficiently manage book details, including customer accounts and transactions. Customers can also benefit from the system by accessing their accounts and viewing available banking services.

Overall, the Bank Management System represents a significant step towards digital transformation in the banking sector. It aims to optimize operations, enhance customer satisfaction, and improve data security. Through the utilization of C programming, this project provides a reliable and efficient solution for modernizing bank management processes.

**2. MOTIVATION**

The primary motivation behind selecting the Bank Management System project stems from the existing situation in the banking industry. Recognizing the challenges faced by banks in Nepal, it became evident that there is a lack of efficient bank management systems, particularly for smaller banks. Manual paperwork remains the predominant method for carrying out various tasks, leading to an increased risk of errors and raising significant concerns.

With numerous small banks across the country encountering similar issues, the absence of a systematic and organized management system has emerged as a critical problem that needs immediate attention. To address this pressing issue, the decision was made to develop a Bank Management System software solution.

The proposed software aims to revolutionize the way banks operate by introducing a comprehensive database for storing and managing crucial banking information. By automating data storage and entry processes, the system ensures enhanced accuracy and efficiency in managing bank operations. The user-friendly interface, equipped with clear labels for all essential features, ensures ease of use for bank administrators and customers alike.

The motivation behind this project also stems from the realization that local bank owners in Nepal still heavily rely on outdated methods, such as handwritten documents, to track their banking activities. This approach is not only inconvenient and time-consuming but also poses security risks. Therefore, there is a pressing need to overcome these challenges and improve the effectiveness and security of bank record-keeping.

By developing this Bank Management System software, the project aims to streamline banking operations, making them easier, faster, more efficient, and more secure. The software provides a digital solution that replaces the inefficient and insecure process of storing banking records at the local level.

Overall, the motivation behind this project lies in addressing the current limitations and challenges faced by banks in Nepal. By leveraging C programming, the Bank Management System aims to bring about a transformative change in the way banking operations are carried out, leading to increased efficiency, accuracy, and security.

# 3. OBJECTIVE

The objective of this C programming project on Bank Management System is to design and develop software that enhances the efficiency of user and administrative processes, reduces dependency on paper-based methods, and ultimately improves customer conversion rates while addressing security concerns. The project aims to achieve the following key objectives:

* **Enhance Organizational Management**

Develop a comprehensive software solution that facilitates seamless management of various banking operations, including customer accounts, transactions, to enhance overall organizational efficiency and effectiveness.

* **Reduce Paper Usage**

Implement digital record-keeping mechanisms to minimize reliance on traditional paper-based processes, thereby reducing costs, improving data accuracy, and promoting eco-friendly practices within the banking environment.

* **Optimize Time and Manpower Utilization**

Automate manual tasks, such as data entry and report generation, to streamline operations and reduce the time and effort required by bank staff, thereby optimizing resource allocation and improving productivity.

By achieving these objectives, this project aims to revolutionize bank management processes, resulting in enhanced operational efficiency, improved customer satisfaction, reduced costs, and heightened data security.

# 4. FEATURE OF PROJECT

This project offers the following features:

* It uses C programming language for its completion.
* It has Admin and User section.
* It can create and stores record of customer details in a file.
* It is banking system so; user can transfer an amount of balance to another person using account number and vice-versa.
* It has Admin and User login option for security measures.
* It also provides ATM system for User in which user can deposit and withdraw money.
* It overcomes the traditional method for keeping the record and It is fast, easy and effective

**5. LITERATURE REVIEW**

Banking institutions are the backbone of economic systems, and effective management is critical to their success. With advancements in technology and evolving customer expectations, the implementation of robust bank management systems has become imperative. This literature review explores key themes related to bank management systems, including efficiency, security, customer experience, and regulatory compliance.

Efficiency is a key consideration in bank management systems. Researchers have highlighted the importance of automating routine tasks, streamlined processes, and optimized resource allocation. Studies have shown that implementing efficient bank management systems leads to improved operational effectiveness, reduced costs, and enhanced customer satisfaction. This efficiency can be achieved through features such as automated transaction tracking, streamlined loan management, and accurate report generation.

Security is another crucial aspect of bank management systems. With the increasing risk of cyber threats and fraudulent activities, ensuring robust security measures is paramount. Research emphasizes the importance of implementing secure login verification, encryption techniques, and data backup systems to safeguard sensitive customer information and maintain the integrity of the system.

Customer experience is a key driver of success in the banking industry. Studies highlight the significance of user-friendly interfaces, clear navigation, and personalized services in enhancing customer satisfaction and loyalty. Bank management systems should prioritize providing a seamless and intuitive experience for customers, enabling them to easily access their accounts, make transactions, and retrieve relevant information

Regulatory compliance is a critical concern in the banking sector. Researchers emphasize the need for bank management systems to adhere to various regulations, such as anti-money laundering (AML) and know-your-customer (KYC) requirements. Effective bank management systems should incorporate functionalities that facilitate compliance monitoring, reporting, and auditing processes to ensure adherence to regulatory standards

In conclusion, the literature highlights the importance of efficient, secure, customer-centric, and compliant bank management systems.

# 6. PROBLEM STATEMENT

The existing bank management system relies heavily on manual paperwork, resulting in laborious efforts and a higher probability of errors. The need for extensive physical file storage complicates the maintenance of data related to various banking operations. The limitations of the traditional system have been identified, including the excessive use of manual labor, repetitive procedures, inadequate security measures, difficulties in handling data, lack of efficient data backup mechanisms, and challenges in managing and organizing information.

The traditional system also exhibits low levels of security, leaving sensitive customer information vulnerable to unauthorized access and potential breaches. In addition, the difficulties in handling and organizing data in a paper-based environment hinder quick and efficient retrieval of information, impacting customer service and decision-making processes

Furthermore, the lack of effective data backup mechanisms increases the risk of data loss in case of unforeseen events such as natural disasters or system failures. The current information management practices within the bank management system are challenging and chaotic, impeding the seamless flow of information and hindering effective decision-making.

Therefore, the need arises for a comprehensive Bank Management System software solution that addresses these challenges and provides a more efficient, secure, and streamlined approach to managing banking operations. By developing a software system that automates processes, improves security measures, ensures reliable data backup, and facilitates efficient information management, these identified issues can be effectively resolved, resulting in enhanced operational efficiency, improved customer satisfaction, and better overall management of banking processes.

**7. ALGORITHM**

The algorithm of the entire system is given below:

|  |  |
| --- | --- |
| **Steps** | **Work** |
|  Step 1 | Start |
|  Step 2 | Input username and password |
|  Step 3 | Check if username and password is belonging to admin or customers. |
|  Step 4 | If it is belonging to customers then, move to main menu. Else, back to login page. |
|  Step 5 | If it is belonging to admin then, move to main menu. Else, back to login page. |
|  Step 5 | Press the options as you required. |
|  Step 6 | Display the desire output as the user presses the option accordingly. |
|  Step 7 | Exit |

*Figure 1: Algorithm of program*

**8. Flowchart**

Input Username and password

if username & password = customer's

if username & password = admin's

False False

Output " You are not a valid User "

True True

Welcome Customers

Welcome Admin

If press =

1

1

Show Records

Add Records

Transactions

2

Update Records

2

3

3

Press = 1

ATM

Delete Records

If press 1 or 0

4

Exit

4

Check all Records

5

Exit

Press = 0

*Figure 2: Flowchart of Library Management System*

**9. IMPLEMENTATION**

We execute our program with the help of string and i/o libraries. To implement he goals, following methodology are used:

1. In start of this system with login authentication is done with string and file handling functions.
2. Use of structure while writing code.
3. It uses the concept of file handling for storing, reading and writing data in file.

**10. FUTURE ENHANCEMENTS**

As the field of banking and financial services continues to evolve, the integration of emerging technologies becomes crucial to stay competitive and provide innovative solutions. One potential future enhancement for the Bank Management System project is the integration of blockchain technology.

* Enhanced Security:

Blockchain provides a high level of security by utilizing cryptographic algorithms and distributed consensus mechanisms. Transactions recorded on the blockchain are resistant to tampering, making it extremely difficult for malicious actors to alter or manipulate data. This ensures the integrity and trustworthiness of the banking system.

* Improved Transparency:

The transparent nature of blockchain allows all participants in the banking system, including customers, administrators, and regulators, to have real-time visibility into transactions. This promotes transparency and accountability, reducing the risk of fraudulent activities.

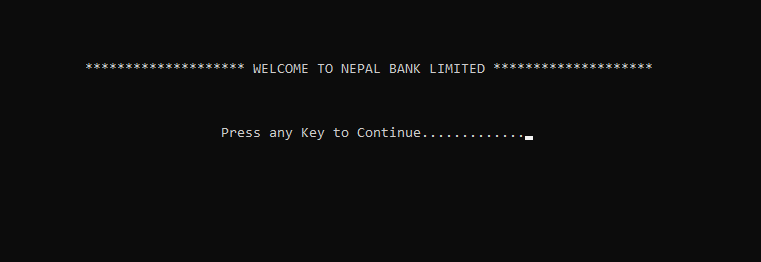
* Streamlined Transactions:

Blockchain enables faster and more efficient transaction processing. With the use of smart contracts, which are self-executing agreements written in code, various banking operations such as fund transfers, loan approvals, and account management can be automated and executed seamlessly, eliminating the need for manual intervention.

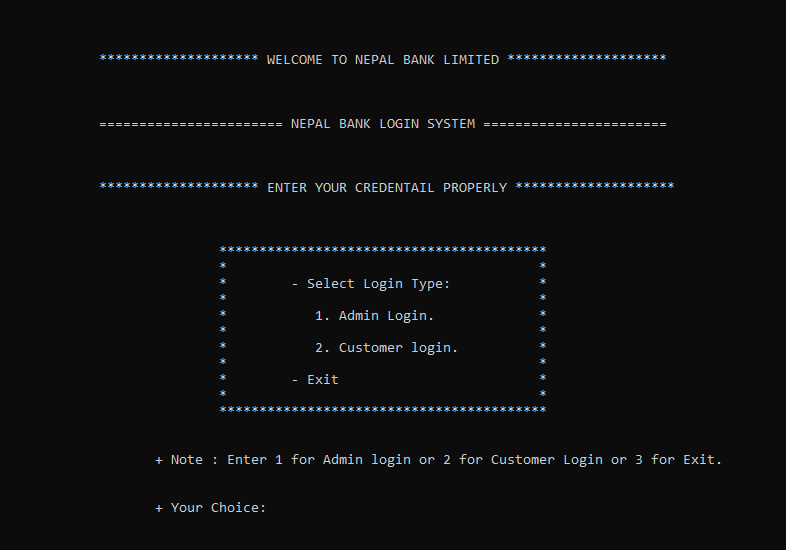
By incorporating blockchain technology into the Bank Management System, these enhancements can bring about a more secure, transparent, and efficient banking experience for customers while streamlining administrative processes. However, it is important to consider the technical complexities and potential challenges associated with implementing blockchain, including scalability, interoperability, and regulatory considerations. Thorough research, analysis, and collaboration with experts in blockchain technology would be required to ensure a successful integration in the future.

# 11. OUTPUT SCREEN

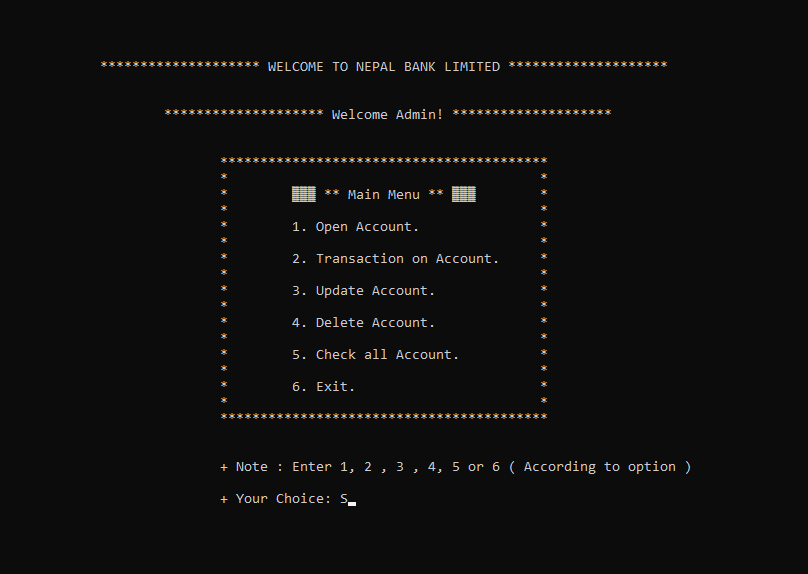
The output screen of some features are provided below:



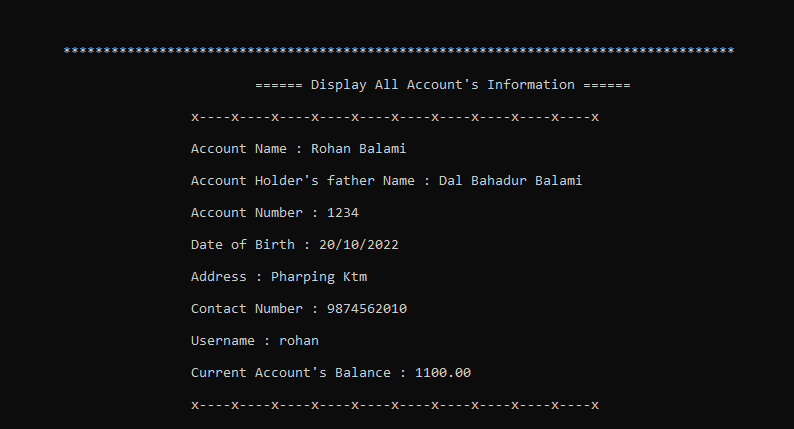
*Figure 3: Welcome Screen*



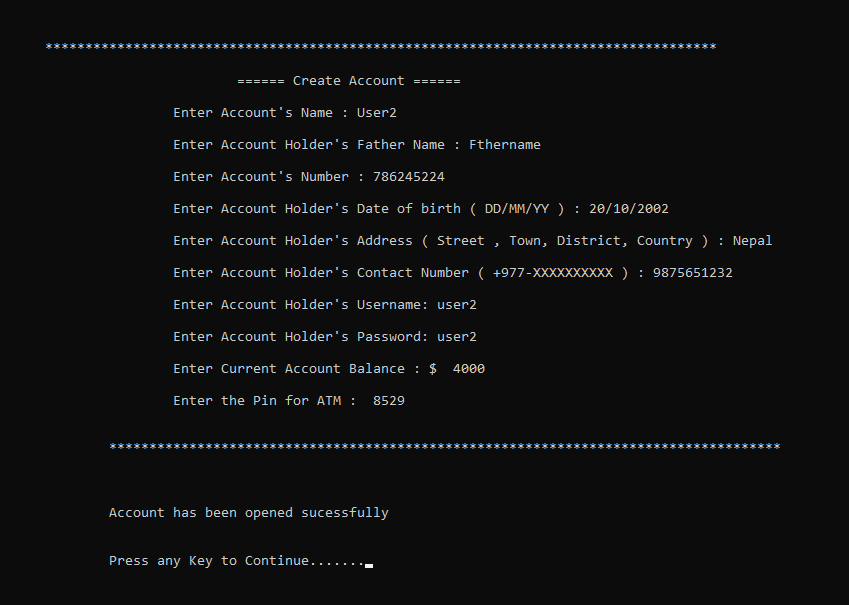
*Figure 4: Login option*



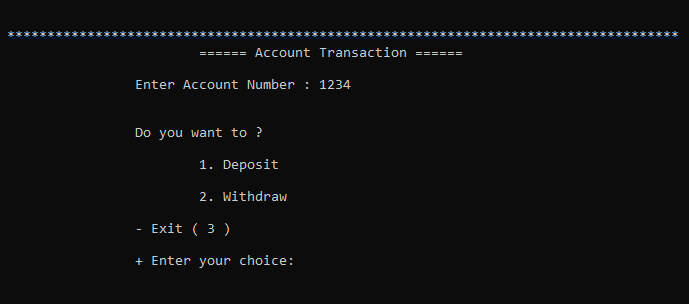
*Figure 5: Admin menu*

****

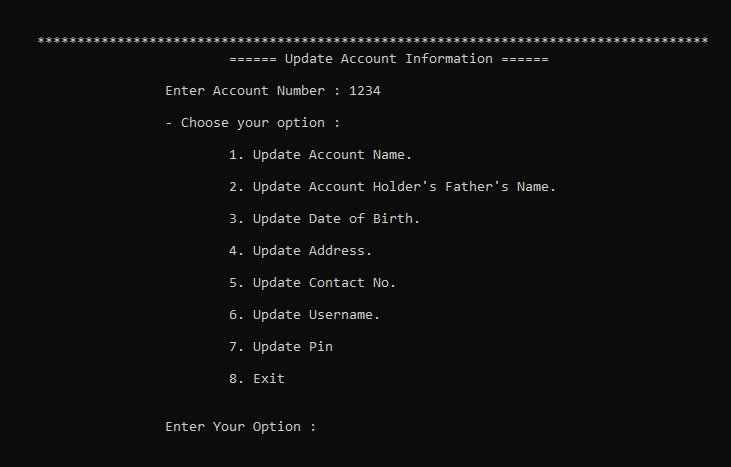
*Figure 6: Viewing all account information of customer.*



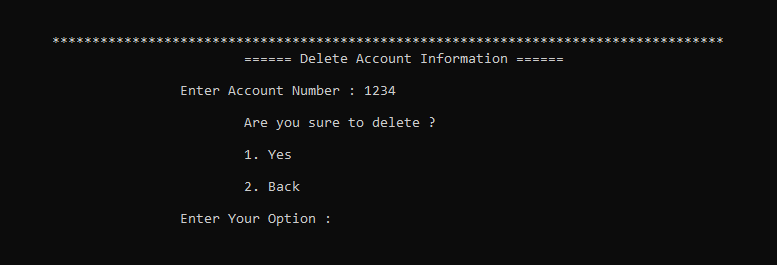
*Figure 7: Creating or adding record of new customers.*



*Figure 8: Balance withdrawal and Deposit by Admin.*



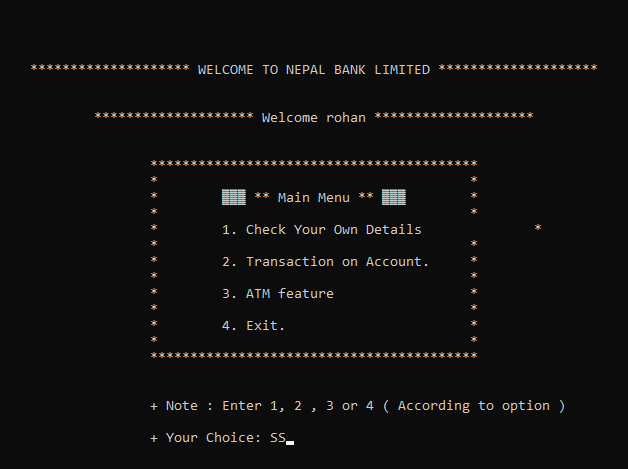
*Figure 9: Updating the customers information by admin*



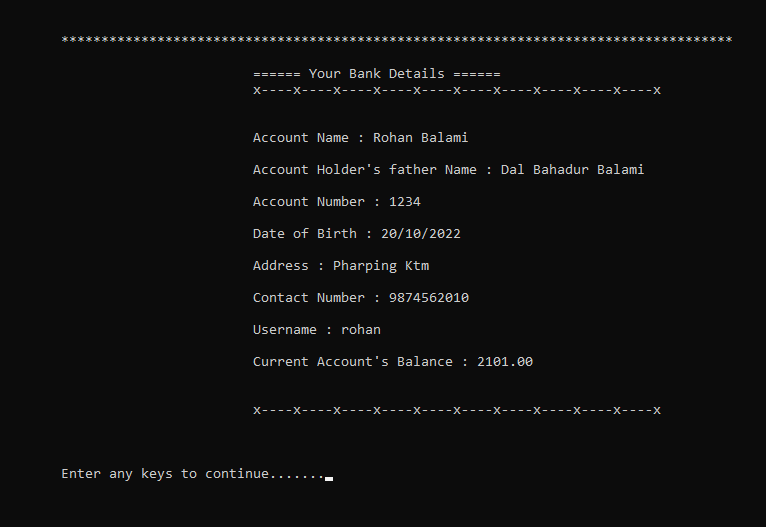
*Figure 10: Deleting Account Information*



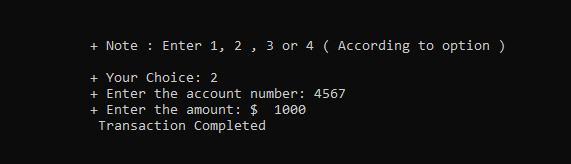
*Figure 11: Login panel for both admin and customers*



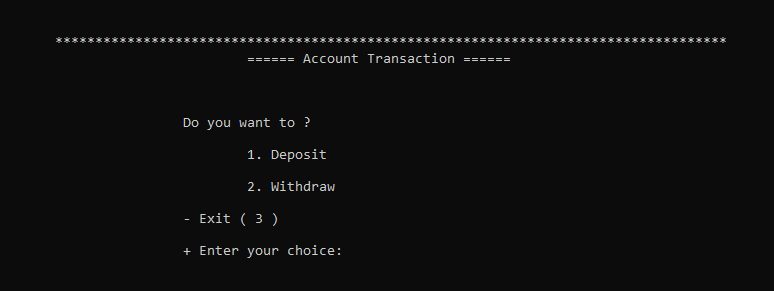
*Figure 12: Customer Menus*



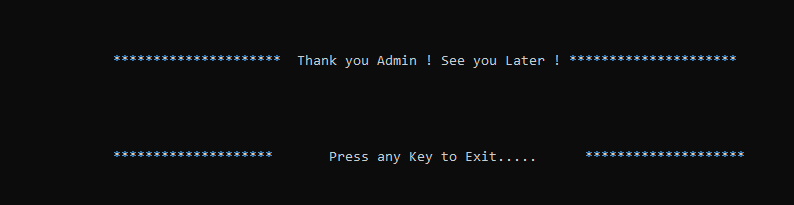
*Figure 13: Checking Customers Detail by itself*



*Figure 14: Transferring balance from one to another*



*Figure 15: ATM Features*



*Figure 16: Exiting both as a customer and a staff*

1. **SOURCE CODE**

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

#include<string.h>

#include<time.h>

#define MAX\_USERNAME\_LEN 20

#define MAX\_PASSWORD\_LEN 20

#define MAX\_LEN 20

void auth();

int adminlogin();

int customerlogin();

void admin\_function();

int customer\_function(char\* accountNum, char\* cus\_usr, char\* atmpin);

void acc\_open();

void trac\_admin();;

void update\_acc();

void delete\_acc();

void check\_acc();

int cus\_check\_details(char\* accountNum);

void cus\_trans(char\* accountNum);

int atm(char\* accountNum, char\* atmpin);

typedef struct Account

{

char accountName[40];

char acctholfathernam[40];

char accountNumber[20];

char DateOfBirth[15];

char address[50];

char contactNum[15];

char username[40];

char upassword[40];

float accountBalance;

char atmPin[04];

} Account;

void main () {

system("CLS");

printf("\n\n\n\n\t\t\t\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* WELCOME TO NEPAL BANK LIMITED \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \t\t\t\t\t\t\t\t\n\n\n");

printf("\t\t\t\t\t\t\t Press any Key to Continue.............");

getch();

auth();

}

// Function to handle user to choose admin or customer login

void auth() {

int choice;

// Prompt user to choose admin or customer login

system("CLS");

printf("\n\n\n\n\t\t\t\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* WELCOME TO NEPAL BANK LIMITED \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \t\t\t\t\t\t\t\t\n\n\n");

printf("\t\t\t\t\t ======================= NEPAL BANK LOGIN SYSTEM ======================= \t\t\t\t\t\t\t\t\n\n\n");

printf("\t\t\t\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ENTER YOUR CREDENTAIL PROPERLY \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \t\t\t\t\t\t\t\t\n\n\n");

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

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

printf("\t\t\t\t\t\t\t\*\t - Select Login Type: \t\t\*\n");

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

printf("\t\t\t\t\t\t\t\*\t 1. Admin Login. \t\t\*\n");

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

printf("\t\t\t\t\t\t\t\*\t 2. Customer login. \t\t\*\n");

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

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

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

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

printf("\t\t\t\t\t\t+ Note : Enter 1 for Admin login or 2 for Customer Login or 3 for Exit. \n\n\n");

printf("\t\t\t\t\t\t+ Your Choice: ");

scanf("%d", &choice);

// Call login function based on user's choice

switch (choice)

{

case 1:

adminlogin(); // Admin login

break;

case 2:

customerlogin(); // Customer login

break;

case 3:

system("CLS"); // Exit

printf("\n\n\n\n\t\t\t\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Thank you Admin ! See you Later ! \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \t\t\t\t\t\t\t\t\n");

printf("\n\n\n\n\t\t\t\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Press any Key to Exit..... \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

getch();

exit(0);

default:

// Invalid login function, calling auth() again.

system("CLS");

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

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

printf("\t\t\t\t\t\t\t\*\t Invalid choice ! \t\t\t\*\n");

printf("\t\t\t\t\t\t\t\*\t Sorry please try again ! \t\t\*\n");

printf("\t\t\t\t\t\t\t\*\t Note: You are only allowed to choose: \t\*\n");

printf("\t\t\t\t\t\t\t\*\t 1 or 2 or 3 \t\t\t\*\n");

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

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

printf("\t\t\t\t\t\t\t\*\t Press any Key to continue.....");

getch();

auth();

}

}

// Function to handle admin login option

int adminlogin(){

// Declare variables for username and password

char username[MAX\_USERNAME\_LEN];

char password[MAX\_PASSWORD\_LEN];

system("CLS");

// Get username and password from user input

printf("\n\n\n\n\t\t\t\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* WELCOME TO NEPAL BANK LIMITED \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \t\t\t\t\t\t\t\t\n\n\n");

printf("\t\t\t\t\t ======================= NEPAL BANK LOGIN SYSTEM ======================= \t\t\t\t\t\t\t\t\n\n\n");

printf("\t\t\t\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ENTER YOUR CREDENTAIL PROPERLY \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \t\t\t\t\t\t\t\t\n\n\n");

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

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

printf("\t\t\t\t\t\t\t\* Enter your Username: ( Maximum 20 ) \t\*\n ");

printf("\t\t\t\t\t\t\t\* => ");

scanf("%s", username);

printf("\t\t\t\t\t\t\t\* Enter your Password: ( Maximum 20 ) \t\*\n ");

printf("\t\t\t\t\t\t\t\* => ");

int i=0;

char ch;

while ((ch = \_getch()) != 13) {

// ch = \_getch(); // read a character from the user input

if (ch == '\n') {

password[i] = '\0';

break;

} else if (ch == '\b') { // backspace key

if (i > 0) {

i--;

printf("\b \b"); // erase the last character on the screen

}

} else if ( ch != '\n' ) {

password[i] = ch;

i++;

printf("\*"); // display \* on the screen to mask the password

}

password[i] = '\0';

}

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

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

// Define the valid usernames and passwords for the system

char valid\_admin\_username[20];

char valid\_admin\_password[20];

// file

FILE \*admin\_file;

char a\_user[MAX\_LEN], a\_password[MAX\_LEN];

char admin[] = "Admin/admin.txt";

// Admin Username and Password

admin\_file = fopen(admin, "r");

if (admin\_file == NULL) {

printf("Unable to valid the User ! Please try again !");

return 1;

}

fgets(a\_user, MAX\_LEN, admin\_file);

strtok(a\_user, "\n"); // remove newline character

fgets(a\_password, MAX\_LEN, admin\_file);

strtok(a\_password, "\n"); // remove newline character

fclose(admin\_file);

strcpy(valid\_admin\_username, a\_user);

strcpy(valid\_admin\_password, a\_password);

// Check if the user is an admin

if (strcmp(username, valid\_admin\_username) == 0 && strcmp(password, valid\_admin\_password) == 0)

{

admin\_function();

}

else {

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

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

printf("\t\t\t\t\t\t\t\*\t Invalid Username or Password ! \t\*\n");

printf("\t\t\t\t\t\t\t\*\t Sorry Please Try Again ! \t\t\*\n");

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

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

printf("\t\t\t\t\t\t\t Press any Key to continue..... ");

getch();

auth();

}

return 0;

}

// Function to handle customer login option

int customerlogin(){

// Declare variables for username and password

char username[MAX\_USERNAME\_LEN];

char password[MAX\_PASSWORD\_LEN];

Account accountInfo;

int found = 0;

system("CLS");

// Get username and password from user input

printf("\n\n\n\n\t\t\t\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* WELCOME TO NEPAL BANK LIMITED \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \t\t\t\t\t\t\t\t\n\n\n");

printf("\t\t\t\t\t ======================= NEPAL BANK LOGIN SYSTEM ======================= \t\t\t\t\t\t\t\t\n\n\n");

printf("\t\t\t\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ENTER YOUR CREDENTAIL PROPERLY \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \t\t\t\t\t\t\t\t\n\n\n");

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

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

printf("\t\t\t\t\t\t\t\* Enter your Username: ( Maximum 20 ) \t\*\n ");

printf("\t\t\t\t\t\t\t\* => ");

scanf("%s", username);

printf("\t\t\t\t\t\t\t\* Enter your Password: ( Maximum 20 ) \t\*\n ");

printf("\t\t\t\t\t\t\t\* => ");

int i=0;

char ch;

while ((ch = \_getch()) != 13) {

// ch = \_getch(); // read a character from the user input

if (ch == '\n') {

password[i] = '\0';

break;

} else if (ch == '\b') { // backspace key

if (i > 0) {

i--;

printf("\b \b"); // erase the last character on the screen

}

} else if ( ch != '\n' ) {

password[i] = ch;

i++;

printf("\*"); // display \* on the screen to mask the password

}

password[i] = '\0';

}

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

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

// file

FILE \*customer\_file;

// Customer Username and Password from accountinfo.bin

customer\_file = fopen("Sys/accountInfo.bin", "rb");

if ( customer\_file == NULL ) {

printf("Unable to valid the User ! Please try again !");

return 0;

}

char \*accountNumber = NULL;

char \*cus\_username = NULL;

char \*atmpin = NULL;

// Search for the account with the matching username and password

while (fread(&accountInfo, sizeof(struct Account), 1, customer\_file) == 1) {

if (strcmp(username, accountInfo.username) == 0 &&

strcmp(password, accountInfo.upassword) == 0) {

found = 1;

accountNumber = accountInfo.accountNumber;

cus\_username = accountInfo.username;

atmpin = accountInfo.atmPin;

break;

}

}

if (found) {

customer\_function(accountNumber, cus\_username, atmpin);

return 0;

} else {

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

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

printf("\t\t\t\t\t\t\t\*\t Invalid Username or Password ! \t\*\n");

printf("\t\t\t\t\t\t\t\*\t Sorry Please Try Again ! \t\t\*\n");

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

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

printf("\t\t\t\t\t\t\t Press any Key to continue..... ");

getch();

auth();

return 0;

}

return 0;

}

// Function to handle admin login

void admin\_function()

{

// Admin Main Menu

int choice\_admin;

lineno219:

system("CLS");

printf("\n\n\n\n\t\t\t\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* WELCOME TO NEPAL BANK LIMITED \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \t\t\t\t\t\t\t\t\n");

printf("\n\t\t\t\t\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Welcome Admin! \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \t\t\t\t\t\t\t\t\n\n\n");

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

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

printf("\t\t\t\t\t\t\t\*\t \xB2\xB2\xB2 \*\* Main Menu \*\* \xB2\xB2\xB2 \t\*\n");

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

printf("\t\t\t\t\t\t\t\*\t 1. Open Account. \t\t\*\n");

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

printf("\t\t\t\t\t\t\t\*\t 2. Transaction on Account.\t\*\n");

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

printf("\t\t\t\t\t\t\t\*\t 3. Update Account. \t\t\*\n");

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

printf("\t\t\t\t\t\t\t\*\t 4. Delete Account. \t\t\*\n");

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

printf("\t\t\t\t\t\t\t\*\t 5. Check all Account. \t\t\*\n");

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

printf("\t\t\t\t\t\t\t\*\t 6. Exit. \t\t\t\*\n");

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

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

printf("\t\t\t\t\t\t\t+ Note : Enter 1, 2 , 3 , 4, 5 or 6 ( According to option ) \n\n");

printf("\t\t\t\t\t\t\t+ Your Choice: ");

scanf("%d", &choice\_admin);

// Call login function based on admin's choice

switch (choice\_admin)

{

case 1:

acc\_open(); // Account Open

goto lineno219;

break;

case 2:

trac\_admin(); // Transaction on Account

goto lineno219;

break;

case 3:

update\_acc(); // Update Account

goto lineno219;

break;

case 4:

delete\_acc(); // Delete Account

goto lineno219;

break;

case 5:

check\_acc(); // Check all Account

goto lineno219;

break;

case 6:

system("CLS"); // Exit

printf("\n\n\n\n\t\t\t\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Thank you Admin ! See you Later ! \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \t\t\t\t\t\t\t\t\n");

printf("\n\n\n\n\t\t\t\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Press any Key to Exit...... \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \t\t\t\t\t\t\t\t\n");

getch();

auth();

default:

// Invalid option, calling auth() again.

system("CLS");

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

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

printf("\*\t Invalid choice ! \t\t\t\*\n");

printf("\*\t Sorry please try again ! \t\t\*\n");

printf("\*\t Note: You are only allowed to choose: \t\*\n");

printf("\*\t 1, 2, 3, 4, 5, or 6 \t\t\t\*\n");

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

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

printf("\t\t\t\t\t\t\t Press any Key to Continue..... \t\t\t\t\t\t\t\t");

getch();

goto lineno219;

break;

}

}

// Function to handle admin Account Opening

void acc\_open()

{

FILE \*fileopen = fopen("Sys/accountInfo.bin", "ab+");

if (fileopen == NULL)

{

printf("\n\t\t\t Error in Creating Account\n");

}

Account accountInformation;

system("CLS");

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

printf("\t\t\t\t\t\t\t\t====== Create Account ======\n");

printf("\n\t\t\t\t\t\t\tEnter Account's Name : ");

getchar();

gets(accountInformation.accountName);

printf("\n\t\t\t\t\t\t\tEnter Account Holder's Father Name : ");

gets(accountInformation.acctholfathernam);

printf("\n\t\t\t\t\t\t\tEnter Account's Number : ");

gets(accountInformation.accountNumber);

printf("\n\t\t\t\t\t\t\tEnter Account Holder's Date of birth ( DD/MM/YY ) : ");

gets(accountInformation.DateOfBirth);

printf("\n\t\t\t\t\t\t\tEnter Account Holder's Address ( Street , Town, District, Country ) : ");

gets(accountInformation.address);

printf("\n\t\t\t\t\t\t\tEnter Account Holder's Contact Number ( +977-XXXXXXXXXX ) : ");

gets(accountInformation.contactNum);

printf("\n\t\t\t\t\t\t\tEnter Account Holder's Username: ");

gets(accountInformation.username);

printf("\n\t\t\t\t\t\t\tEnter Account Holder's Password: ");

gets(accountInformation.upassword);

printf("\n\t\t\t\t\t\t\tEnter Current Account Balance : Rs ");

scanf("%f", &accountInformation.accountBalance);

getchar();

printf("\n\t\t\t\t\t\t\tEnter the Pin for ATM : ");

gets(accountInformation.atmPin);

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

fwrite(&accountInformation, sizeof(accountInformation), 1, fileopen);

printf("\n\n\t\t\t\t\t\tAccount has been opened sucessfully\n");

printf("\n\n\t\t\t\t\t\tPress any Key to Continue.......");

getch();

fclose(fileopen);

}

// Function to handle admin Transaction

void trac\_admin()

{

FILE \*fileOne = fopen("Sys/accountInfo.bin", "rb");

FILE \*temp = fopen("Sys/temp.bin", "wb");

Account accountInformation, temp\_Info;

int op, flag = 0;

if (fileOne == NULL || temp == NULL)

{

printf("\n\t\t\tError !\n");

}

system("CLS");

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

printf("\t\t\t\t\t\t\t\t\t\t====== Account Transaction ======\n\n");

printf("\t\t\t\t\t\t\t\t\tEnter Account Number : ");

getchar();

gets(temp\_Info.accountNumber);

while (fread(&accountInformation, sizeof(accountInformation), 1, fileOne) == 1)

{

if (strcmp(accountInformation.accountNumber, temp\_Info.accountNumber) == 0)

{

flag++;

printf("\n\n\t\t\t\t\t\t\t\t\tDo you want to ?\n\n\t\t\t\t\t\t\t\t\t\t1. Deposit\n\n\t\t\t\t\t\t\t\t\t\t2. Withdraw\n\n\t\t\t\t\t\t\t\t\t- Exit ( 3 )\n\n\t\t\t\t\t\t\t\t\t+ Enter your choice: ");

scanf("%d", &op);

if (op == 1)

{

system("CLS");

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

printf("\t\t\t\t\t\t\t\t\t\t====== Diposit ======\n\n");

printf("\n\t\t\t\t\t\t\t\t\t+ Current Balance:Rs %.2f", accountInformation.accountBalance);

printf("\n\n\t\t\t\t\t\t\t\t\t+ Enter the amount you want to deposit:Rs ");

scanf("%f", &temp\_Info.accountBalance);

accountInformation.accountBalance += temp\_Info.accountBalance;

printf("\n\t\t\t\t\t\t\t\t\t+ Current Balance after Deposit:Rs %.2f", accountInformation.accountBalance);

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

fwrite(&accountInformation, sizeof(accountInformation), 1, temp);

printf("\n\n\t\t\t\t\t\t\t\t\tDeposited successfully!\n\n");

}

else if (op == 2 )

{

system("CLS");

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

printf("\t\t\t\t\t\t\t\t\t\t====== Withdraw ======\n\n");

printf("\n\t\t\t\t\t\t\t\t\t+ Current Balance:Rs %.2f", accountInformation.accountBalance);

printf("\n\n\t\t\t\t\t\t\t\t\t+ Enter the amount you want to withdraw:Rs ");

scanf("%f", &temp\_Info.accountBalance);

if (accountInformation.accountBalance < temp\_Info.accountBalance)

{

printf("\n\t\t\t\t\t\t\t\t\t Unsufficient Balance!\n\n");

fwrite(&accountInformation, sizeof(accountInformation), 1, temp);

}

else

{

accountInformation.accountBalance -= temp\_Info.accountBalance;

printf("\n\t\t\t\t\t\t\t\t\tCurrent Balance after Withdraw:Rs %.2f", accountInformation.accountBalance);

fwrite(&accountInformation, sizeof(accountInformation), 1, temp);

printf("\n\n\t\t\t\t\t\t\t\t\tWithdraw Successfully!\n\n");

}

}

else if (op == 3 )

{

fwrite(&accountInformation, sizeof(accountInformation), 1, temp);

}

else

{

printf("\n\n\t\t\t\t\t\t\t Invalid Option");

getch();

fwrite(&accountInformation, sizeof(accountInformation), 1, temp);

}

}

else

{

fwrite(&accountInformation, sizeof(accountInformation), 1, temp);

}

}

if (flag == 0)

{

printf("\n\t\t\t\t\t\t\t\t\tAccount is not found");

}

fclose(fileOne);

fclose(temp);

remove("Sys/accountInfo.bin");

rename("Sys/temp.bin", "Sys/accountInfo.bin");

printf("\n\n\t\t\t\t\t\t\t\t\tPress any Key to Continue.......");

getch();

}

// Function to handle admin Update Account

void update\_acc()

{

FILE \*fileOne = fopen("Sys/accountInfo.bin", "rb");

FILE \*temp = fopen("Sys/temp.bin", "wb");

Account accountInformation, tempInformation;

int choice, flag = 0;

if (fileOne == NULL || temp == NULL)

{

printf("\n\t\t\tError !\n");

}

system("CLS");

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

printf("\t\t\t\t\t\t\t\t\t\t====== Update Account Information ======\n");

printf("\n\t\t\t\t\t\t\t\t\tEnter Account Number : ");

getchar();

gets(tempInformation.accountNumber);

while (fread(&accountInformation, sizeof(accountInformation), 1, fileOne) == 1)

{

if (strcmp(accountInformation.accountNumber, tempInformation.accountNumber) == 0)

{

flag++;

printf("\n\t\t\t\t\t\t\t\t\t- Choose your option :\n\n\t\t\t\t\t\t\t\t\t\t1. Update Account Name.\n\n\t\t\t\t\t\t\t\t\t\t2. Update Account Holder's Father's Name.\n\n\t\t\t\t\t\t\t\t\t\t3. Update Date of Birth.\n\n\t\t\t\t\t\t\t\t\t\t4. Update Address.\n\n\t\t\t\t\t\t\t\t\t\t5. Update Contact No.\n\n\t\t\t\t\t\t\t\t\t\t6. Update Username.\n\n\t\t\t\t\t\t\t\t\t\t7. Update Pin\n\n\t\t\t\t\t\t\t\t\t\t8. Exit");

printf("\n\n\n\t\t\t\t\t\t\t\t\tEnter Your Option : ");

scanf("%d", &choice);

if (choice == 1)

{

system("cls");

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

printf("\t\t\t\t\t\t\t\t\t\t====== Update Account Information ======\n\n");

printf("\n\t\t\t\t\t\t\t\t\tEnter Account's Name to Update: ");

getchar();

gets(tempInformation.accountName);

strcpy(accountInformation.accountName, tempInformation.accountName);

fwrite(&accountInformation, sizeof(accountInformation), 1, temp);

printf("\n\n\t\t\t\t\t\t\t\t\tUpdated successfully!\n\n");

}

else if (choice == 2)

{

system("cls");

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

printf("\t\t\t\t\t\t\t\t\t\t====== Update Account Information ======\n\n");

printf("\n\t\t\t\t\t\t\t\t\tEnter Account Holder's Father's Name : ");

getchar();

gets(tempInformation.acctholfathernam);

strcpy(accountInformation.acctholfathernam, tempInformation.acctholfathernam);

fwrite(&accountInformation, sizeof(accountInformation), 1, temp);

printf("\n\n\t\t\t\t\t\t\t\t\tUpdated successfully!\n\n");

}

else if (choice == 3)

{

system("cls");

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

printf("\t\t\t\t\t\t\t\t\t\t====== Update Account Information ======\n\n");

printf("\n\t\t\t\t\t\t\t\t\tEnter Date of Birth to Update: ");

getchar();

gets(tempInformation.DateOfBirth);

strcpy(accountInformation.DateOfBirth, tempInformation.DateOfBirth);

fwrite(&accountInformation, sizeof(accountInformation), 1, temp);

printf("\n\n\t\t\t\t\t\t\t\t\tUpdated successfully!\n\n");

}

else if (choice == 4)

{

system("cls");

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

printf("\t\t\t\t\t\t\t\t\t\t====== Update Account Information ======\n\n");

printf("\n\t\t\t\t\t\t\t\t\tEnter Address to Update: ");

getchar();

gets(tempInformation.address);

strcpy(accountInformation.address, tempInformation.address);

fwrite(&accountInformation, sizeof(accountInformation), 1, temp);

printf("\n\n\t\t\t\t\t\t\t\t\tUpdated successfully!\n\n");

}

else if (choice == 5)

{

system("cls");

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

printf("\t\t\t\t\t\t\t\t\t\t====== Update Account Information ======\n\n");

printf("\n\t\t\t\t\t\t\t\t\tEnter Contact No. to Update: ");

getchar();

gets(tempInformation.contactNum);

strcpy(accountInformation.contactNum, tempInformation.contactNum);

fwrite(&accountInformation, sizeof(accountInformation), 1, temp);

printf("\n\n\t\t\t\t\t\t\t\t\tUpdated successfully!\n\n");

}

else if (choice == 6)

{

system("cls");

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

printf("\t\t\t\t\t\t\t\t\t\t====== Update Account Information ======\n\n");

printf("\n\t\t\t\t\t\t\t\t\tEnter New Username to Update: ");

getchar();

gets(tempInformation.username);

strcpy(accountInformation.username, tempInformation.username);

fwrite(&accountInformation, sizeof(accountInformation), 1, temp);

printf("\n\n\t\t\t\t\t\t\t\t\tUpdated successfully!\n\n");

}

else if (choice == 7)

{

system("cls");

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

printf("\t\t\t\t\t\t\t\t\t\t====== Update Account Information ======\n\n");

printf("\n\t\t\t\t\t\t\t\t\tEnter New ATM Pin to Update: ");

getchar();

gets(tempInformation.atmPin);

strcpy(accountInformation.atmPin, tempInformation.atmPin);

fwrite(&accountInformation, sizeof(accountInformation), 1, temp);

printf("\n\n\t\t\t\t\t\t\t\t\tUpdated successfully!\n\n");

}

else if (choice == 8)

{

fwrite(&accountInformation, sizeof(accountInformation), 1, temp);

}

else

{

printf("\n\t\t\t\t\t\t\t\t\tInvalid Option.");

fwrite(&accountInformation, sizeof(accountInformation), 1, temp);

}

}

else

{

fwrite(&accountInformation, sizeof(accountInformation), 1, temp);

}

}

fclose(fileOne);

fclose(temp);

remove("Sys/accountInfo.bin");

rename("Sys/temp.bin", "Sys/accountInfo.bin");

if (flag == 0)

{

printf("\n\n\t\t\t\t\t\t\t\t\tAccount is not found");

}

printf("\n\n\t\t\t\t\t\t\t\t\tPress any key to Continue.......");

getch();

}

// Function to handle admin Delete Account

void delete\_acc()

{

FILE \*fileOne = fopen("Sys/accountInfo.bin", "rb");

FILE \*temp = fopen("Sys/temp.bin", "wb");

Account accountInformation, tempInformation;

int choice, flag = 0;

if (fileOne == NULL || temp == NULL)

{

printf("\n\t\t\tError !\n");

}

system("CLS");

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

printf("\t\t\t\t\t\t\t\t\t\t====== Delete Account Information ======\n");

printf("\n\t\t\t\t\t\t\t\t\tEnter Account Number : ");

getchar();

gets(tempInformation.accountNumber);

while (fread(&accountInformation, sizeof(accountInformation), 1, fileOne) == 1)

{

if (strcmp(accountInformation.accountNumber, tempInformation.accountNumber) == 0)

{

flag++;

printf("\n\t\t\t\t\t\t\t\t\t\tAre you sure to delete ?\n\n\t\t\t\t\t\t\t\t\t\t1. Yes\n\n\t\t\t\t\t\t\t\t\t\t2. Back\n\n\t\t\t\t\t\t\t\t\tEnter Your Option : ");

scanf("%d", &choice);

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

if (choice == 1)

{

printf("\n\n\t\t\t\t\t\t\t\t\tAccount has been deleted successfully!\n\n");

}

else if (choice == 2)

{

fwrite(&accountInformation, sizeof(accountInformation), 1, temp);

}

else

{

printf("\n\n\t\t\t\t\t\t\t\t\tInvalid Option");

fwrite(&accountInformation, sizeof(accountInformation), 1, temp);

}

}

else

{

fwrite(&accountInformation, sizeof(accountInformation), 1, temp);

}

}

fclose(fileOne);

fclose(temp);

remove("Sys/accountInfo.bin");

rename("Sys/temp.bin", "Sys/accountInfo.bin");

if (flag == 0)

{

printf("\n\n\t\t\t\t\t\t\t\t\tAccount is not found\n\n");

}

printf("\n\t\t\t\t\t\t\t\t\tPress any key to Continue.......");

getch();

}

// Function to handle admin Check Account

void check\_acc()

{

FILE \*fileOne = fopen("Sys/accountInfo.bin", "rb");

if (fileOne == NULL)

{

printf("\n\t\t\tError !\n");

}

Account accountInformation;

system("CLS");

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

printf("\t\t\t\t\t\t\t\t\t\t====== Display All Account's Information ======\n\n");

while (fread(&accountInformation, sizeof(accountInformation), 1, fileOne) == 1)

{

printf("\t\t\t\t\t\t\t\t\tx----x----x----x----x----x----x----x----x----x----x\n\n");

printf("\t\t\t\t\t\t\t\t\tAccount Name : %s\n\n\t\t\t\t\t\t\t\t\tAccount Holder's father Name : %s\n\n\t\t\t\t\t\t\t\t\tAccount Number : %s\n\n\t\t\t\t\t\t\t\t\tDate of Birth : %s\n\n\t\t\t\t\t\t\t\t\tAddress : %s\n\n\t\t\t\t\t\t\t\t\tContact Number : %s\n\n\t\t\t\t\t\t\t\t\tUsername : %s\n\n\t\t\t\t\t\t\t\t\tCurrent Account's Balance : %.2f\n\n", accountInformation.accountName, accountInformation.acctholfathernam, accountInformation.accountNumber, accountInformation.DateOfBirth, accountInformation.address, accountInformation.contactNum, accountInformation.username, accountInformation.accountBalance);

printf("\t\t\t\t\t\t\t\t\tx----x----x----x----x----x----x----x----x----x----x\n\n");

}

fclose(fileOne);

printf("\n\n\t\t\t\t\t\t\t\t\tPress any key to Continue.......");

getch();

}

// Function to handle customer login

int customer\_function(char\* accountNum, char\* cus\_usr, char\* atmpin)

{

// Customer Main Menu

int choice\_customer;

lineno752:

system("CLS");

printf("\n\n\n\n\t\t\t\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* WELCOME TO NEPAL BANK LIMITED \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \t\t\t\t\t\t\t\t\n");

printf("\n\t\t\t\t\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Welcome %s \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \t\t\t\t\t\t\t\t\n\n\n", cus\_usr);

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

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

printf("\t\t\t\t\t\t\t\*\t \xB2\xB2\xB2 \*\* Main Menu \*\* \xB2\xB2\xB2 \t\*\n");

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

printf("\t\t\t\t\t\t\t\*\t 1. Check Your Own Details \t\t\*\n");

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

printf("\t\t\t\t\t\t\t\*\t 2. Transaction on Account.\t\*\n");

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

printf("\t\t\t\t\t\t\t\*\t 3. ATM feature \t\t\*\n");

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

printf("\t\t\t\t\t\t\t\*\t 4. Exit. \t\t\t\*\n");

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

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

printf("\t\t\t\t\t\t\t+ Note : Enter 1, 2 , 3 or 4 ( According to option ) \n\n");

printf("\t\t\t\t\t\t\t+ Your Choice: ");

scanf("%d", &choice\_customer);

// Call login function based on user's choice

switch (choice\_customer)

{

case 1:

cus\_check\_details(accountNum); // customers check details

goto lineno752;

break;

case 2:

cus\_trans(accountNum ); // Customer Transaction

goto lineno752;

break;

case 3:

atm(accountNum, atmpin); // ATM feautres

goto lineno752;

break;

case 4:

system("CLS"); // Exit

printf("\n\n\n\n\t\t\t\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Thank you Customers ! See you Later ! \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \t\t\t\t\t\t\t\t\n");

printf("\n\n\n\n\t\t\t\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Press any Key to Exit..... \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

getch();

exit(0);

default:

// Invalid login function, calling auth() again.

system("CLS");

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

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

printf("\t\t\t\t\t\t\t\*\t Invalid choice ! \t\t\t\*\n");

printf("\t\t\t\t\t\t\t\*\t Sorry please try again ! \t\t\*\n");

printf("\t\t\t\t\t\t\t\*\t Note: You are only allowed to choose: \t\*\n");

printf("\t\t\t\t\t\t\t\*\t 1 or 2 or 3 or 4 \t\t\t\*\n");

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

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

printf("\t\t\t\t\t\t\t\*\t Press any Key to continue.....");

getch();

auth();

}

return 0;

}

// Function to handle customer Check Account

int cus\_check\_details(char\* accountNum) {

FILE \*detailsfile = fopen("Sys/accountInfo.bin", "rb");

Account acc\_details;

int acc\_found = 0;

if (detailsfile == NULL )

{

printf("\n\t\t\tError !\n");

}

system("CLS");

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

printf("\t\t\t\t\t\t\t\t====== Your Bank Details ======\n");

while (fread(&acc\_details, sizeof(acc\_details), 1, detailsfile) == 1) {

if (strcmp(acc\_details.accountNumber, accountNum) == 0) {

acc\_found = 1 ;

break;

}

}

fclose(detailsfile);

// Print details of account if found

if (acc\_found) {

printf("\t\t\t\t\t\t\t\tx----x----x----x----x----x----x----x----x----x----x\n\n");

printf("\n\t\t\t\t\t\t\t\tAccount Name : %s\n\n\t\t\t\t\t\t\t\tAccount Holder's father Name : %s\n\n\t\t\t\t\t\t\t\tAccount Number : %s\n\n\t\t\t\t\t\t\t\tDate of Birth : %s\n\n\t\t\t\t\t\t\t\tAddress : %s\n\n\t\t\t\t\t\t\t\tContact Number : %s\n\n\t\t\t\t\t\t\t\tUsername : %s\n\n\t\t\t\t\t\t\t\tCurrent Account's Balance : %.2f\n\n", acc\_details.accountName, acc\_details.acctholfathernam, acc\_details.accountNumber, acc\_details.DateOfBirth, acc\_details.address, acc\_details.contactNum, acc\_details.username, acc\_details.accountBalance);

printf("\n\t\t\t\t\t\t\t\tx----x----x----x----x----x----x----x----x----x----x\n\n");

} else {

printf("\t\t\t\t\t\t\t\tAccount not found\n");

}

printf("\n\n\t\t\t\t\tEnter any keys to continue.......");

getch();

}

void cus\_trans(char\* accountNum) {

char recevier[20];

float amount;

// open account info file

FILE \*file = fopen("Sys/accountInfo.bin", "r+b");

if (file == NULL) {

printf("\t\t\t\t\t\t\t Error: Could not open account info file.\n");

return;

}

printf("\t\t\t\t\t\t\t+ Enter the account number: ");

scanf("%s", recevier );

// printf("Sender %s , Receiver %s, Amount %f", accountNum, recevier, amount );

Account from\_account, to\_account;

// find 'Sender' account

int found\_from\_account = 0, found\_to\_account = 0;

while (fread(&from\_account, sizeof(Account), 1, file) == 1) {

if (strcmp(from\_account.accountNumber, accountNum ) == 0) {

found\_from\_account = 1;

// printf("\t\t\t\t\t\t\t Sender Account found");

printf("\t\t\t\t\t\t\t+ Enter the amount: Rs ");

scanf("%f", &amount );

if (from\_account.accountBalance < amount) {

printf("\t\t\t\t\t\t\tError: Insufficient balance.\n");

getch();

fclose(file);

return;

}

from\_account.accountBalance -= amount;

fseek(file, (long)(-sizeof(Account)), SEEK\_CUR);

fwrite(&from\_account, sizeof(Account), 1, file);

printf("\t\t\t\t\t\t\t Transaction Completed");

getch();

break;

}

}

// find 'to' account

fseek(file, 0, SEEK\_SET);

while (fread(&to\_account, sizeof(Account), 1, file) == 1) {

if (strcmp(to\_account.accountNumber, recevier) == 0) {

found\_to\_account = 1;

to\_account.accountBalance += amount;

fseek(file, (long)(-sizeof(Account)), SEEK\_CUR);

fwrite(&to\_account, sizeof(Account), 1, file);

break;

}

}

// close file and print message

fclose(file);

if (!found\_from\_account) {

printf("Error: From account not found.\n");

return;

}

if (!found\_to\_account) {

printf("Error: To account not found.\n");

return;

}

printf("Transaction successful.\n");

}

int atm(char\* accountNum, char\* atmpin) {

int right\_pin = atoi(atmpin);

int enteredpin, count;

int continuetransaction = 1;

time\_t now;

time(&now);

system("CLS");

printf("\n\n");

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

printf("\t\t\t\t\t\t\t\t====== Welcome to ATM System ======\n\n");

printf("\t\t\t\t\t\t\t\t %s", ctime(&now));

while (right\_pin != enteredpin )

{

system("CLS");

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

printf("\t\t\t\t\t\t\t\t====== Welcome to ATM System ======\n");

printf("\n\n\n\n\t\t\t\t\t Enter your Pin : ");

scanf("%d", &enteredpin);

}

if ( enteredpin != right\_pin ) {

system("CLS");

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

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

printf("\t\t\t\t\t\t\t\*\t Invalid choice ! \t\t\t\*\n");

printf("\t\t\t\t\t\t\t\*\t Sorry please try again ! \t\t\*\n");

printf("\t\t\t\t\t\t\t\*\t Note: You are only allowed to choose: \t\*\n");

printf("\t\t\t\t\t\t\t\*\t 1 or 2 or 3 or 4 \t\t\t\*\n");

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

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

printf("\t\t\t\t\t\t\t\*\t Press any Key to continue.....");

getch();

}

count++;

if( count == 3 && right\_pin != enteredpin ) {

system("CLS");

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

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

printf("\t\t\t\t\t\t\t\*\t No more allowed ! \t\t\t\*\n");

printf("\t\t\t\t\t\t\t\*\t Sorry please try again later ! \t\t\*\n");

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

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

printf("\t\t\t\t\t\t\t\*\t Press any Key to continue.....");

getch();

}

while ( continuetransaction != 0)

{

if ( enteredpin = right\_pin )

{

FILE \*fileOne = fopen("Sys/accountInfo.bin", "r+b");

// FILE \*temp = fopen("Sys/temp.bin", "w+b");

Account accountInformation, temp\_Info;

int op, flag = 0;

if (fileOne == NULL)

{

printf("\n\t\t\tError !\n");

}

system("CLS");

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

printf("\t\t\t\t\t\t\t\t\t\t====== Account Transaction ======\n\n");

strcpy(temp\_Info.accountNumber, accountNum);

while (fread(&accountInformation, sizeof(accountInformation), 1, fileOne) == 1)

{

if (strcmp(accountInformation.accountNumber, temp\_Info.accountNumber) == 0)

{

flag++;

printf("\n\n\t\t\t\t\t\t\t\t\tDo you want to ?\n\n\t\t\t\t\t\t\t\t\t\t1. Deposit\n\n\t\t\t\t\t\t\t\t\t\t2. Withdraw\n\n\t\t\t\t\t\t\t\t\t- Exit ( 3 )\n\n\t\t\t\t\t\t\t\t\t+ Enter your choice: ");

scanf("%d", &op);

if (op == 1)

{

system("CLS");

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

printf("\t\t\t\t\t\t\t\t\t\t====== Diposit ======\n\n");

printf("\n\t\t\t\t\t\t\t\t\t+ Current Balance:Rs %.2f", accountInformation.accountBalance);

printf("\n\n\t\t\t\t\t\t\t\t\t+ Enter the amount you want to deposit:Rs ");

scanf("%f", &temp\_Info.accountBalance);

accountInformation.accountBalance += temp\_Info.accountBalance;

printf("\n\t\t\t\t\t\t\t\t\t+ Current Balance after Deposit:Rs %.2f", accountInformation.accountBalance);

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

fseek(fileOne, (long)(-sizeof(Account)), SEEK\_CUR);

fwrite(&accountInformation, sizeof(Account), 1, fileOne);

printf("\n\n\t\t\t\t\t\t\t\t\tDeposited successfully!\n\n");

}

else if (op == 2 )

{

system("CLS");

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

printf("\t\t\t\t\t\t\t\t\t\t====== Withdraw ======\n\n");

printf("\n\t\t\t\t\t\t\t\t\t+ Current Balance:Rs %.2f", accountInformation.accountBalance);

printf("\n\n\t\t\t\t\t\t\t\t\t+ Enter the amount you want to withdraw:Rs ");

scanf("%f", &temp\_Info.accountBalance);

if (accountInformation.accountBalance < temp\_Info.accountBalance)

{

printf("\n\t\t\t\t\t\t\t\t\t Unsufficient Balance!\n\n");

fwrite(&accountInformation, sizeof(accountInformation), 1, fileOne);

}

else

{

accountInformation.accountBalance -= temp\_Info.accountBalance;

printf("\n\t\t\t\t\t\t\t\t\tCurrent Balance after Withdraw:Rs %.2f", accountInformation.accountBalance);

fseek(fileOne, (long)(-sizeof(Account)), SEEK\_CUR);

fwrite(&accountInformation, sizeof(Account), 1, fileOne);

printf("\n\n\t\t\t\t\t\t\t\t\tWithdraw Successfully!\n\n");

}

}

else if (op == 3 )

{

printf("\n\n\t\t\t\t\t\t\t\t\t Thank you ! Visit Again.......\n\n");

return 0;

}

else

{

printf("\n\n\t\t\t\t\t\t\t Invalid Option");

getch();

fwrite(&accountInformation, sizeof(accountInformation), 1, fileOne);

}

}

else

{

fwrite(&accountInformation, sizeof(accountInformation), 1, fileOne);

}

}

if (flag == 0)

{

printf("\n\t\t\t\t\t\t\t\t\tAccount is not found");

}

fclose(fileOne);

// fclose(temp);

remove("Sys/accountInfo.bin");

rename("Sys/temp.bin", "Sys/accountInfo.bin");

printf("\n\n\t\t\t\t\t\t\t\t\tPress any Key to Continue.......");

getch();

}

}

printf("\t\t\t\t\t\t\t\t + Do You want to Continue ? ");

printf("\t\t\t\t\t\t\t\t + Press 1 for Continue or 0 for Exit ");

scanf("%d",&continuetransaction);

return 0;

}

1. **REFERENCES**

*Bank Account Management System - javatpoint*. (n.d.). www.javatpoint.com. <https://www.javatpoint.com/bank-account-management-system>

*Bank management system in C - TAE*. (n.d.). https://www.tutorialandexample.com/bank-management-system-in-c

*Bank Management System Project Using C Language - Studytonight*. (n.d.). https://www.studytonight.com/c-projects/bank-management-system-project-using-c-language

Evangelista, A. (2022). Bank Management System in C with Source Code. *Itsourcecode.com*. https://itsourcecode.com/free-projects/c-projects/bank-management-system-in-c-with-source-code/

GeeksforGeeks. (2022). Bank account system in C using File handling. *GeeksforGeeks*. https://www.geeksforgeeks.org/bank-account-system-in-c-using-file-handling/

*Introducing ChatGPT*. (n.d.). https://openai.com/blog/chatgpt

Lemonaki, D. (2021). What is The C Programming Language? A Tutorial for Beginners. *freeCodeCamp.org*. https://www.freecodecamp.org/news/what-is-the-c-programming-language-beginner-tutorial/