Final Project Report — Programming Fundamentals

University Name: FAST NUCES Karachi

Department: Department of Ai and Data Science

Course: Programming Fundamentals

Project Title: Bank Management System

Submitted By: Naresh Kumar (Roll No. 25K-2501) and Syed Zaid Khalid (Roll No. 25K-2522)

Submitted To: Sir Jahanzaib

Semester: Fall 2025

Date: 21-11-2025

Abstract

This Banking System is a graphic user interface (GUI) program developed using C language. It essentially is comparable to a full-scale banking website that allows the user to perform a plethora of tasks such as registering and creating a new account at our bank, if they already have an account then checking balance, adding/ withdrawing money, updating information in that already existing account, checking their transaction history and if they wish to, they also have the ability to delete their bank account. The system can compute many processes for the user per his desire as previously mentioned, it can help him update information of his bank account, it can help him add money, create/ delete his account and much more. The project uses many programming concepts such as loops, arrays, functions, pointers, string manipulation and file handling for permanent memory. The project demonstrates a fully functional banking system recording and appending records of users per their demand.

1. Introduction

Since the inception of modern banking as well as the technological revolution, humans have slowly but drastically moved many of their day-to-day tasks towards automation. The banking industry was no different in terms of becoming part of this global trend. Keeping that in mind we made a small banking system demonstrating the effectiveness and power of computation and automata. Making this project helped strengthen our understanding of variables, strings, arrays, conditionals, loops, file handling, and modular programming techniques.

2. Objectives

- To provide a digitalized system to users so he can manage his multiple accounts more efficiently, if he has many.
- To allow users to perform essential banking operations such as account creation, update, deletion, and secure login.
- To enable accurate storage and display of deposits, withdrawals, balance inquiries and transactional history.
- To reinforce programming concepts like loops, arrays, pointers, file handling, and functions.
- To provide a simple, user-friendly menu-based interface.

3. System Design

System Overview

Flow of the program:

Start → Display login page → If user chooses to “Create Account”, allow him to Input his Details → If user chooses login, allow him to enter his userID and password and log him in → Display menu with multiple options for the user to decide → Once user decides what he wishes to do, computing that option → continue the program until users decides to Exit

Algorithm

1. Start the program.
2. Prompt users to either create account, login or exit.

Once in, prompt users to either (1) Check Balance, (2) Update Information, (3) Delete Account, (4) Deposit Money, (5) Withdraw Money, (6) View Transaction money or (7) Log out.

3. Compute what user wishes to do.
4. End

Input & Output

Input: If creating account then Name, Father name, mobile number, address, and new password. If logging in, then mobile number and password.

Output: Depending on what user chooses, it can output their bank balance, withdrawn money and transaction money.

4. Implementation

Language: C

Compiler/IDE: Code: Blocks / Dev C++ / GCC

Key Features

- If the user creates a new account, the new data he provides is permanently stored in our login file using file handling which later used to verify his ID the next time he tries to log in.
- When user creates his account (if he chooses that option), the program does not end after he is finished setting up his account and he then must rerun it, rather it takes him ,back to login page to now login with his new account.
- The program does not end after user performs an operation provided in User menu rather when he is done performing that operation, he is taken back to user menu to perform further action.
- Even if the user deletes his account or logs out, the program does not end, rather he is taken to login page to do further actions. This means that program is never ending as well as the user can log in as many times as he wants with as many accounts he wants.
- Allow user to delete his account if he wishes to, if he does his login records are permanently deleted from our login file making it impossible for him to login again with the same mobile number and password.
- Provides user with a multitude of options mimicking an actual full-scale banking system, from withdrawing/ depositing money to checking his transaction history.
- Whenever user makes a withdrawal, his transaction is stored in our Transaction file which allows us to keep track of his transactions.
- Separate function for every task to be performed ensuring clarity in code and execution.
- Simple user menu for navigation

Code Snippets:

(1) Create account:

```
void createAccount()
```

This function is used to make account, when it is called it requests user to enter his name, father name, mobile number, address, and password. Once done user's account is done and his credentials are stored in login file to be used when he tries to login.

(2) Login:

```
Void login()
```

This function is called if the user chooses to login at login page. It requires user to enter his number and password. Once the user enters it, it is cross checked from entries stored in login database file, if found, the user is logged in.

(3) User menu:

```
void userMenu(Account *user)
```

Once the user logs in, this function is called. This function displays 7 operations users has at his disposal to execute per his liking.

(4) Checking balance:

```
void checkBalance(Account *user)
```

If the user chooses the first option at user menu, this function is called. This function basically accesses the file that stores the specific user's Balance and displays it.

(5) Update Information:

```
Void updateInformation(Account *user)
```

If the user chooses the second option on the user menu, this function is called. This function basically allows user to update his information if he wishes to, it allows user to change his name, father name, address, and password. When done the data is permanently altered in login file.

(6) Deleting account:

```
void deleteAccount(Account *user)
```

If the user chooses the third option on the user menu, this function is called. This function allows user to permanently delete his account.

(7) Deposit Money:

```
void depositMoney(Account *user)
```

If the user chooses the fourth option on the user menu, this function is called. This function allows users to deposit money in his account.

(8) Withdraw Money:

```
void withdrawMoney(Account *user)
```

If the user chooses the fifth option at user menu, this function is called. This function allows user to withdraw desired amount only if his account has bigger balance.

(9) View Transaction history:

```
void viewTransactionHistory(Account *user)
```

If the user chooses the sixth option on the user menu, this function is called. This function allows users to check his transaction history which is constantly stored during transactions.

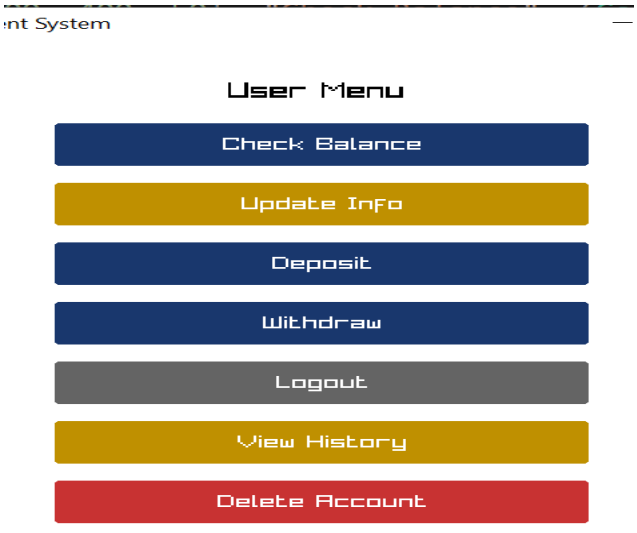
5. Testing & Results

Test No	Input	Expected Output	Actual Output	Status
1	Login Number: Password:	Successfully logged in/ User Menu	Successfully logged in/ User Menu	✅
2	Create account: Name, father name, 11-digit mobile number, address, password	Information is added to file and "Accounted created successfully displayed"	Information is added to file and "Accounted created successfully displayed"	✅
3	Check balance: Click Check balance button at User Menu	User balance displayed.	User balance displayed.	✅
4	Update Information: Click Update info button at User Menu	Allows user the option to update his/her name, father name, address, number, and password and Information updated in file.	Allows user the option to update his/her name, father name, an address, number, and password and Information updated at file.	✅
5	Deleting account: Click Deleting button at User Menu.	User record from Login file deleted and message "Account deleted" displayed.	User record from Login file deleted and message "Account deleted" displayed.	✅
6	Deposit money: Click Deposit button at User Menu.	Money deposited and the message "Deposited amount".	Money deposited and the message "Deposited amount".	✅

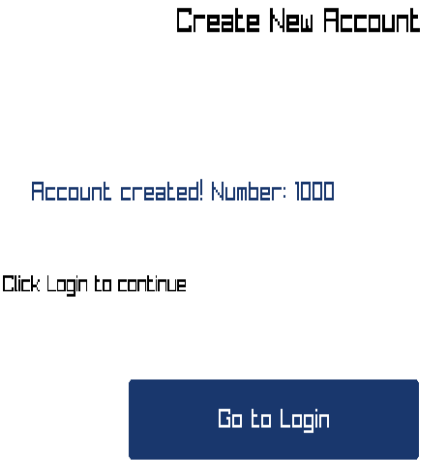
7	Withdraw money: Click Withdraw at User Menu.	User requested to enter amount to be withdrawn and that amount subtracted from his account balance.	User requested to enter amount to be withdrawn and that amount subtracted from his account balance.	✅
8	Transaction History: Click Transaction History at User Menu.	Transaction history displayed.	Transaction history displayed.	✅
9	Log out: Click Logout at User Menu.	User logged out and taken back to Log in page.	Users logged out and taken back to login page.	✅

Results:

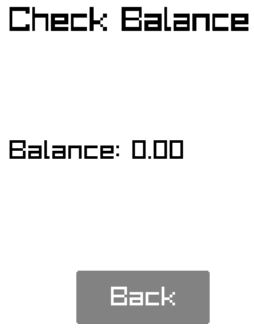
- **User Menu**



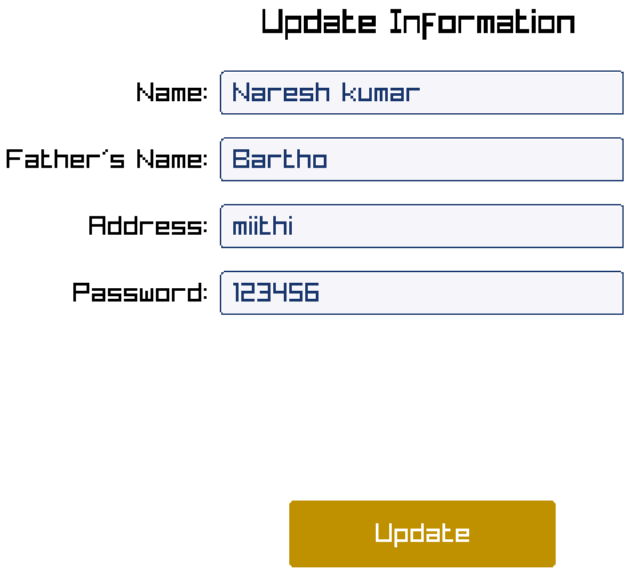
- **Account Created**

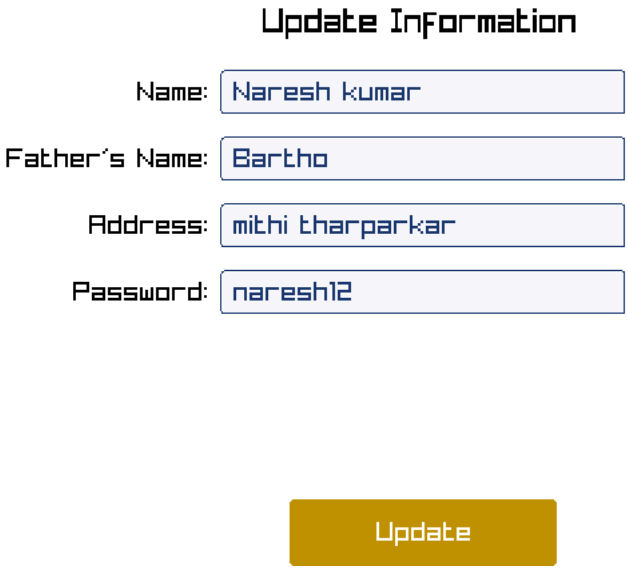


- **Check Balance**



- **Update Information** **Before Update** **After Update**



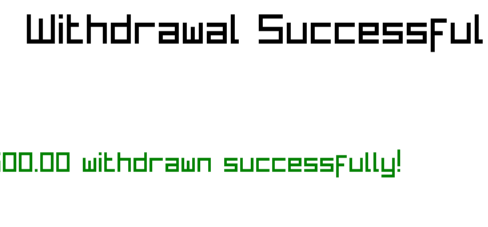


- **Delete Account**

Account deleted.

- **Deposit Money**

- **Withdraw Money**



- **If withdrawing amount 50000 will ask random questions**

- **If answer wrong :**

Withdrawal Failed

Incorrect answer!

Withdrawal cancelled.

- **If answer correct:**

Withdrawal Successful

Amount 51000.00 withdrawn successfully!

- **Logout**

Thank You!

Thanks for visiting our bank

- **Transaction History**

Transaction History

17/11/2025 19:51:17: Deposit 5000.00, Balance: 5000.00
17/11/2025 19:52:04: Withdraw 500.00, Balance: 4500.00
17/11/2025 19:53:56: Withdraw 500.00, Balance: 4000.00

The program performs all its 9 intended actions adequately and allows user to easily and with no hurdle do what he wishes to do with the provided options. The program is also very nicely in sync with the permanent memory at backend in the form of files which are accessed using techniques of file handling and appends/ updates date records in these files consistently. The program also has a very straightforward and easy-to-follow interface making the whole process not just smooth computation-wise but also for the user.

6. Conclusion, Limitations & References

Conclusion

The Banking System successfully demonstrates the application of basic programming principles. It adequately computes all the options available in a standard professional banking system successfully imitating it. The project strengthened our understanding of arrays, loops, conditional statements, file handling, string manipulation, pointers, and functions.

Limitations

- No Encryption
- No checks in place of what will be done to money left if the user decides to delete his bank account but there is still money left in his account
- No 2-step verification for logging in to enhance security.
- When a user deletes his account, his record is permanently deleted, making the recovery of his account, if he decides to undo his action, impossible.
- Allows user to make as many accounts as he wants using create account option.

Future Enhancements

- Add 2-step verification for login and transaction for enhanced security.
- Restrict account deletion by requiring the user to first empty his bank balance.
- Instead of instantly deleting a user's record when he deletes his account, instead store his record in another file for at least 30-60 days and let user know this is the time frame in which he can recover his deleted account if he changes his mind in the future. Once that time frame passes, then permanently delete his record.
- Place a max count on the number of accounts a single user can make.

References

- Let Us C by Yashavant P. Kanetkar
- https://www.geeksforgeeks.org/c-programming-language/
- Stack Overflow