**The British College**

**KATHMANDU**

**Coursework Submission Coversheet**

A Project Report on

**C++ programming**

Subject:

**Object Oriented Programming (OOP)**

Submitted by:

**Tarun Agarwal, 7542, FACULTY BSc. (Hons.) Computing,2022**

Name of Instructor:

**Kumar Lohala**

Submission Date:

09/03/2022

**ACKNOWLEDGEMENT**

I consider it to be in my best interest to express my gratitude and regard to each and every one of the people involved, as the satisfaction that comes with completing a task successfully would be lacking without mentioning the individuals whose constant support and encouragement helped me to complete my project.

I would want to specifically thank my OOP (Object Oriented Programming) instructor, Thank you to Mr . Kumar Lohala for providing me with all the crucial information I needed for this project.

**ABSTRACT**

IT, framework, and product improvement are put to the test when it comes to creating and managing requirement. To make sure they are meeting the client's needs, displaying consistency, staying on schedule, and confirming compliance, an association must fully describe and manage key conditions. The definition and management of necessities is a strategy that can result in a significant, immediate measurable profit.

The project "BANK MANAGEMENT SYSTEM" is built on current technology. The creation of software for a bank administration system is the main goal of this project. This assignment was developed to complete the procedures efficiently and quickly, which is impossible using the manual's organizational structure and is accomplished by this product. This application was written in the C++ programming language and ran on Visual Studio Code As a result, it provides a complete solution for the present management system.

Table of Contents

[1. Introduction 6](#_Toc113119710)

[1.1 Project Description 6](#_Toc113119713)

[1.2 Motivation / Problem Statements 6](#_Toc113119714)

[1.3 Objectives 6](#_Toc113119715)

[2. Literature / Technical Review. 7](#_Toc113119716)

[3. Methodology 8](#_Toc113119717)

[3.1 Detailed Design: 8](#_Toc113119718)

[3.2 List of Classes 9](#_Toc113119719)

[3.3 Objective of each function 9](#_Toc113119720)

[3.4 Variables defined in the functions 11](#_Toc113119721)

[3.5 Hardware / Software Requirements 12](#_Toc113119722)

[4. Output 13](#_Toc113119723)

[5. Conclusion 14](#_Toc113119724)

[6. References 15](#_Toc113119725)

[7. Appendix 16](#_Toc113119726)

[7.1 Screenshots 16](#_Toc113119727)

[7.2 User Manual 22](#_Toc113119728)

Table of Figures

[Figure 1: Main Menu 16](#_Toc113120069)

[Figure 2: Creating an Account 16](#_Toc113120070)

[Figure 3: Displaying the account details 17](#_Toc113120071)

[Figure 4: Transaction Menu 17](#_Toc113120072)

[Figure 5: Displaying the withdrawal menu 18](#_Toc113120073)

[Figure 6: Displaying the withdrawal amount and balance 18](#_Toc113120074)

[Figure 7: Displaying the Deposited Amount and balance 19](#_Toc113120075)

[Figure 8 Displaying individual account information (inactive) 19](#_Toc113120076)

[Figure 9 Displaying the Deposited Amount and balance 20](#_Toc113120077)

[Figure 10 Displaying individual account information (active) 20](#_Toc113120078)

[Figure 11 Displaying all account information 21](#_Toc113120079)

[Figure 12 Deleting an Account 21](#_Toc113120080)

# Introduction

## A bank's secured website, which may be a virtual bank, a depository financial institution, or a savings and loan association, allows consumers to execute financial transactions.

## For many years, banks have used software to offer their services to clients electronically. In the past, these apps allowed users to phone their bank straight from their computer, but banks are now increasingly hesitant to provide their consumers online banking due to security concerns. 2020 (Ramayah)

## 1.1 Project Description

For this project, I developed a menu-driven software that bank employees may use to assist consumers with financial transactions including opening accounts, making deposits, and withdrawing cash in response to inquiries. Concepts like encapsulation, which implies the ability to hide data, inheritance, which means a child class inherits its parent class's properties, and polymorphism, which means the capacity to take an object in a variety of forms, Type conversion, which is described in the supplied project in several formats to use, entails converting one type of data into another type of data as well as operator overloading. The basic ideas in the C++ final project are that the file handling case should be used, which means opening the file in read and write mode, making sure to employ particular criteria, and then closing the file.

## 1.2 Motivation / Problem Statements

I had a lot of difficulty doing the assigned schoolwork since it was challenging. It took additional time because of the length of the question and the small subtleties that could not be ignored. Not only that, but the deadline I had to meet was also highly constrained. The flaws were difficult to spot since the code was so long. It took a long time to fix several mistakes. It was able to finish the working project on time with the aid of certain references, friends' assistance in identifying errors, and effective time management.

## 1.3 Objectives

Making a straightforward menu-driven software and learning about Object-Oriented programming concepts like encapsulation, inheritance, polymorphism, and file handling are the major goals of this project.

* The first objective to develop a program for managing a checking account which will evaluate saving and checking accounts deposit / withdrawals.
* To store all the information permanently made by user like number of deposits and withdraws, ending balance after deducting service fee.
* To notify the user if the account turns active or inactive after having low balance in his / her account.
* Provides security from unauthorized access, authorized users are access granted to the system as password protected and up thus far records of the purchasers are maintained by the authority.
* Creates a user-friendly environment, where a standard user can access through all the advantages of the system.
* Increases efficiency, saves time and extremely quick access of saved data inside the system by the user.

# 2. Literature / Technical Review.

During the time of auditing the code of the project of “Bank Management System” there were some different between the project code and code from the source. From sample I got idea about how to enhance the capability in this project by adding the multiple different functionalities needed to complete this project which could be very helpful to use by their customers.

In this program a base class Account is created in which in the protected section includes float (balance- account holder balance, annual\_intrest\_rate- annual interest rate, service\_charges- service charges, start\_balance- beginning balance, wit\_charges- withdrawal charges, stat), char (name, pass), int (no. of dep, no of wit). In Public section it includes an structure of account holder information as wit\_charges – withdrawal charges, service\_charge- Service charges, stat- account active or not, balance – account balance, annual\_interest\_rate- annual interest rate which is 6.5, start\_balance- set to 0; number of deposit and withdrawal. Including several other functions Then, another class Saving is derived from the base class and calls the member function from the base class and adds its own new characters. Lastly, another class Checking is derived from the base class and calls the member function from the base class and adds its own new characters.

# 

# 3. Methodology

|  |
| --- |
| SavingAccount |
| Status(float)  Make\_withdrawl(float, float)  Make\_deposit(float, float)  MonthlyProc(float, float) |

|  |
| --- |
| CheckingAccount  makeWithdrawal(float, float)  monthlyProc(float)  account\_check() |

## 3.1 Detailed Design:

|  |
| --- |
| Account |
| balance  Annual\_intrest\_rate  Service\_charge  Start\_balance  Wit\_charge  Stat  Name  Pass  No\_of\_dep  No\_of\_wit  No\_of\_deposite  No\_of\_withdrwal |
| Account()  createAccount()  get\_balance()  get\_name()  get\_pass()  set\_stat(bool)  get\_stat()  set\_bal(float)  get\_bal()  makeDeposite(float, float)  makeWithdrawl(float, float)  clacInt(float)  monthlyProc(float, float)  increment\_wit()  diplayData()  delRecord()  display()  menu()  createAcc()  makingDeposite\_Withdraw()  viewRecord()  viewData()  delRec()  main() |

## 3.2 List of Classes

|  |
| --- |
| Account(), createAccount(), get\_balance(), get\_name(), get\_pass(), set\_stat(bool), get\_stat(), set\_bal(float), get\_bal(), makeDeposite(float, float), makeWithdrawl(float, float), clacInt(float), monthlyProc(float, float), increment\_wit(), diplayData(), delRecord(), display() menu(), createAcc(), makingDeposite\_Withdraw(), viewRecord(), viewData(), delRec(), main() |

* **Class Account and their function**
* **Class Saving and their function**

|  |
| --- |
| Status(float), Make\_withdrawl(float, float), Make\_deposit(float, float), MonthlyProc(float, float) |

* **Class Checking and their function**

|  |
| --- |
| makeWithdrawal(float, float), monthlyProc(float), account\_check() |

## 3.3 Objective of each function

**In Account class as base class**

* Account(): structure for customer account.
* createAccount(): creates customer account with unique username, password and initial balance .
* get\_balance(): returns balance .
* get\_name(): returns copy of name.
* get\_pass(): returns copy of password .
* set\_stat(bool): applying bool to stat as st.
* get\_stat(): returns stat.
* set\_bal(float): applying float to balance as bal .
* get\_bal(): returns bal.
* makeDeposite(float, float): returns balance after deposit .
* makeWithdrawl(float, float): returns balance after withdrawal.
* calcInt(float): returns interest added balance after 12 months as bal.
* monthlyProc(float, float): returns balance after processing monthly charges as bal.
* increment\_wit(): increases number of withdrawal.
* diplayData(): diplays all the data from the file account.txt of all existing users after verifying admin password.
* delRecord(): deletes any account requested after verifying password and username.
* display(): displays specific user data when requested after verifying password and username.
* menu(): diplays home menu
* createAcc(): upload the createAccount function in file
* makingDeposite\_Withdraw(): calls class Checking\_account.
* viewRecord(): calls display from account class.
* viewData(): calls displayData from account class.
* delRec(): calls delRecord from account class.
* main(): main function with switch case

**In Saving class which has public access to account class**

* Status(float): checks if the account is active or inactive with the criteria of minimum $25 available in the account.
* Make\_withdrawl(float, float): Returns the balance processed from account class makeWithdrawl function along with increment in the no of withdrawal
* Make\_deposit(float, float): Returns the balance processed from account class makeDeposite function along with changing stat to active if the account fulfills the criteria.
* MonthlyProc(float, float) checks the total no of withdrawals made and if it is more than 4 service charge of $1 is added to each withdrawals and base class monthlyProc() is called.

**In Checking Class which has public access to account class**

* makeWithdrawal(float, float): checks if the account has enough balance to withdraw if not it displays an message and returns and if the amount has enough amount then it checks if the account still has the minimum balance i.e. $25 the amount will be withdrawn yet still if the remaining balance is less than $25 the account stat will be changed to inactive.
* monthlyProc(float): adds the monthly fee of $5 plus $0.10 per withdrawal to the base class variable that holds the monthly service charges.
* account\_check(): the function processes the entire transaction method including the stat, deposit, withdrawal, charges, etc. and returns the final balance

## 3.4 Variables defined in the functions

The different type of the variables is defined in different function order to complete this project. Some of them are as mentioned below:

**In class Account as base class**

* Account(): int (wit\_charge, service\_charge, stat, balance, Annual\_intrest\_rate, start\_balance, no\_of\_wit, no\_of\_withdrawl, no\_of\_dep, no\_of\_deposite)
* createAccount(): it has string name, pass and float balance.
* get\_balance(): it has float balance
* get\_name(): it has string name
* get\_pass(): it has string pass
* set\_stat(bool): it has bool type stat, st
* get\_stat(): it has int type stat
* set\_bal(float): it has float type balance, bal
* get\_bal(): it has float type bal
* makeDeposite(float, float): it has no\_of\_dep as int and a float bal.
* makeWithdrawl(float, float): it has no\_of\_wit as int and a float bal
* clacInt(float): it has float montly interest rate, annual intrest rate as float
* monthlyProc(float, float) it has float service charges
* increment\_wit(): it has int no\_of\_wit
* diplayData(): it has password as string,
* delRecord(): it has name as string and password
* display(): it has name as string
* menu(): it has system as function
* createAcc(): it has file as fstream
* makingDeposite\_Withdraw(): it has ca as checking account
* viewRecord(): it has a as account class
* viewData(): it has a as display function in account class
* delRec(): it has a as delRecord in account class
* main(): it has switch case

**In class Saving which has public access to account class**

* Status(float): it has bal as float
* Make\_withdrawl(float, float): it has wit and bal as float
* Make\_deposit(float, float): it has dep and bal as float
* MonthlyProc(float, float): it has ch and bal as float

**In class Checking which has public access to account class**

* makeWithdrawal(float, float): it has wit and bal as float
* monthlyProc(float): it has bal as float
* account\_check(): it has file as fstream

## 3.5 Hardware / Software Requirements

**Hardware requirement:**

* 512MB of Ram or higher
* 800MHz processor or above
* 20Mb of hard disk space

**Software Requirements:**

* Windows, mac or Linux any operating system.
* Visual Studio Code program need to be installed.

# 4. Output

The Output of the following code as per the project topic and the guidelines provided by the instructor are:

* **Main menu**: Contains 6 options to create account, transaction, view individual banking data, view all account data, delete an account, exit. The output is shown in Figure 1.
* **Creating new account**: All the requirements have to be filled with proper identification such as unique username, password and initial balance. The output is expressed in Figure **2**.
* **Transaction:** Unique ID and password should be correctly inserted to access through existing customer otherwise this facility is not executable. Figure **3** shows the output. Herein, customer want to deposit or withdraw and show the current balance after executing customer’s requirement. The output can be seen in Figure **4 to 7 and 9.**
* **View your banking Information:** Unique ID and password should be correctly inserted to access through existing customer otherwise this facility is not executable. Figure **8 and 10.** Herein, customer can view their Starting balance, current balance, number of withdrawals.
* **View all banking Information:** Admin password should be correctly inserted to access through existing customer otherwise this facility is not executable. Figure **10 and 11.** Herein, admin can view their Starting balance, current balance, number of withdrawals.
* **Close your account:** Unique ID and password should be correctly inserted to access through existing customer otherwise this facility is not executable. Figure **12.** Herein, customer can delete their account.

**Picture references are given in 7.1 Screenshots.**

# 5. Conclusion

Although I had a difficult time finishing the final coursework, the OOP ideas enabled me to make our application more practical and successful.

The final project encompasses all of the C++ programming fundamentals together with object-oriented programming, which promotes increased coding experience and the creation of practical programs. I became more familiar with the project base program thanks to principles like inheritance, polymorphisms, encapsulation, constructors, destructors, and file handling. Although game developers utilize C++ primarily, it may also be employed in other project-based problem-solving tasks like final projects and other projects.

Because it is an extension of the C programming language and because it incorporates notions from Object-oriented programming languages rather than Procedure-oriented programming, C++ is extremely popular.

# 6. References

* Ramayah, T., 2020. *Classifying Users and Non-Users of Internet Banking In Northern Malaysia*. [online] Icommercecentral.com. Available at: <http://www.icommercecentral.com/open-access/classifying-users-and-nonusers-of-internet-banking-in-northern-malaysia.php?aid=38582> [Accessed 9 September 2020].
* Softwaretestinghelp.com. 2020. *C Vs C++: 39 Main Differences Between C And C++ With Examples*. [online] Available at: <https://www.softwaretestinghelp.com/c-vs-cpp/> [Accessed 13 September 2020].

# 7. Appendix

## 7.1 Screenshots

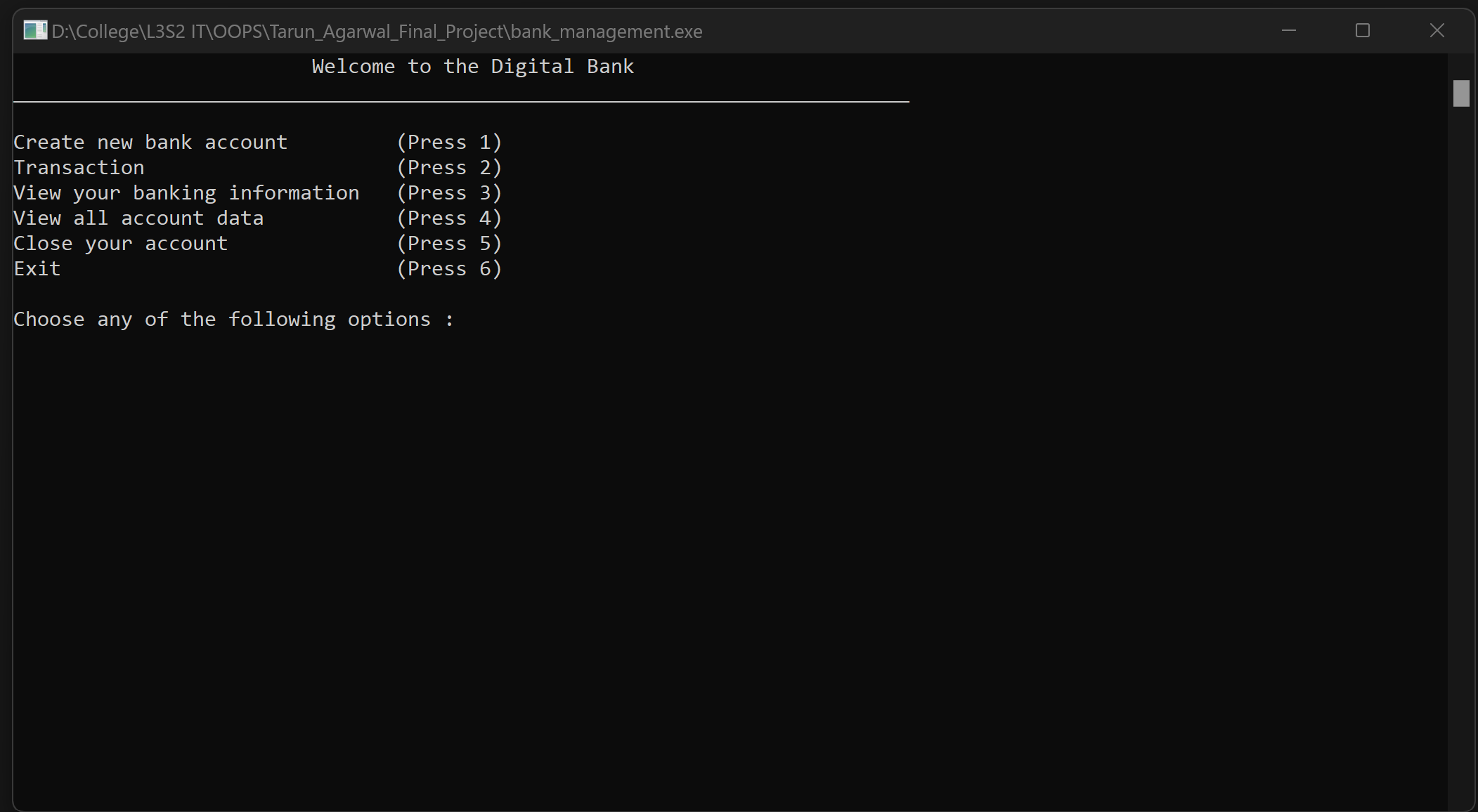


Figure 1: Main Menu

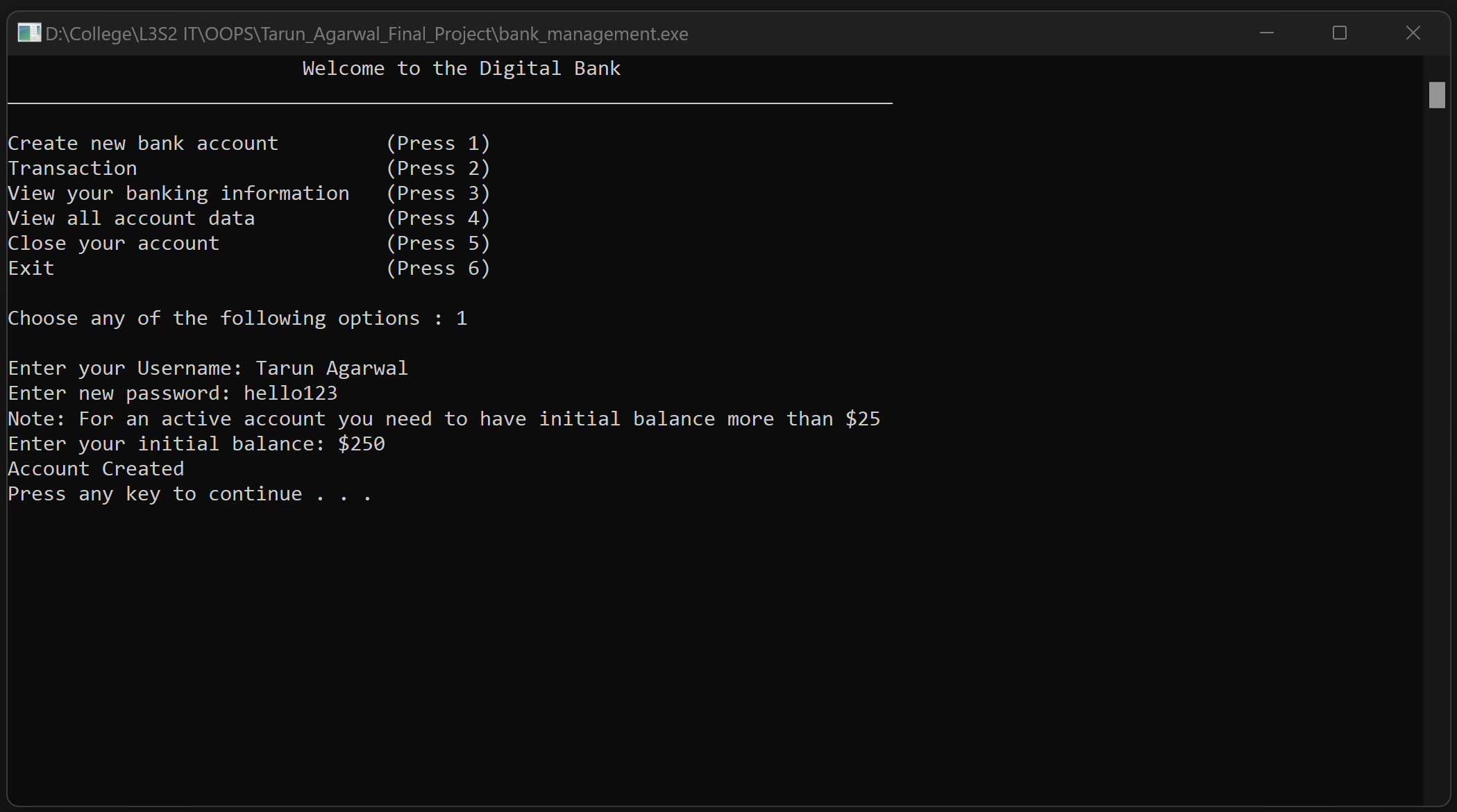


Figure 2: Creating an Account

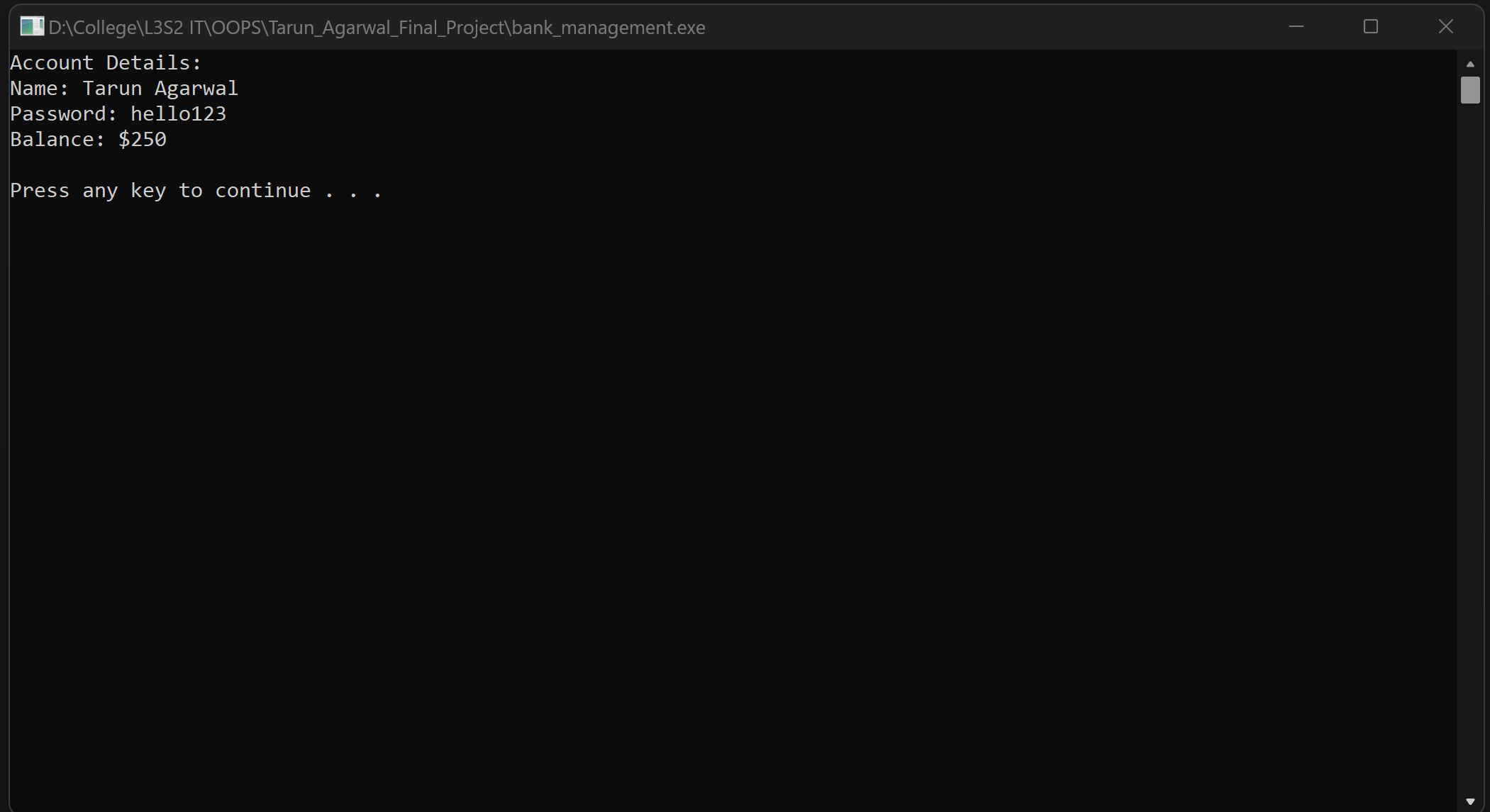


Figure 3: Displaying the account details

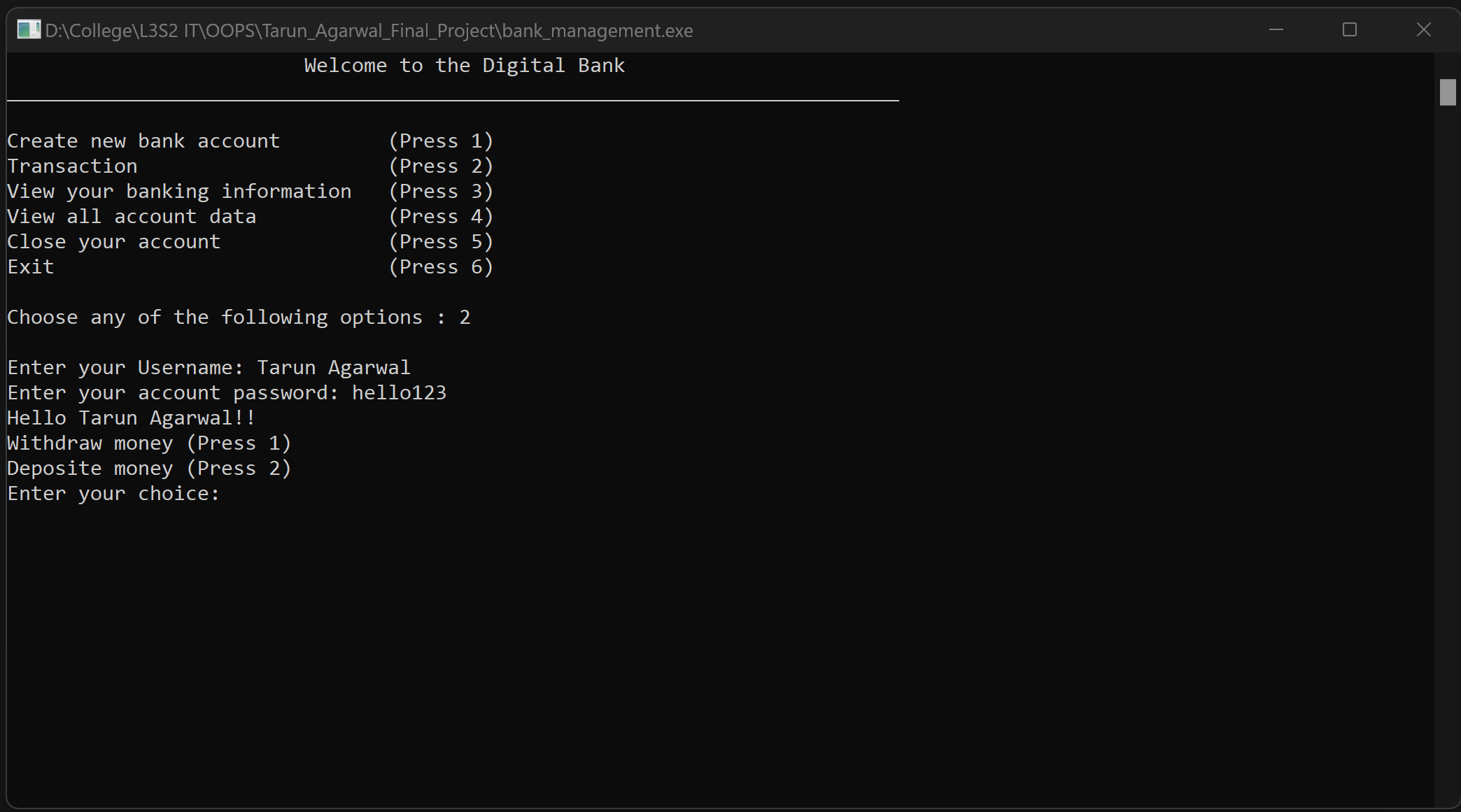


Figure 4: Transaction Menu

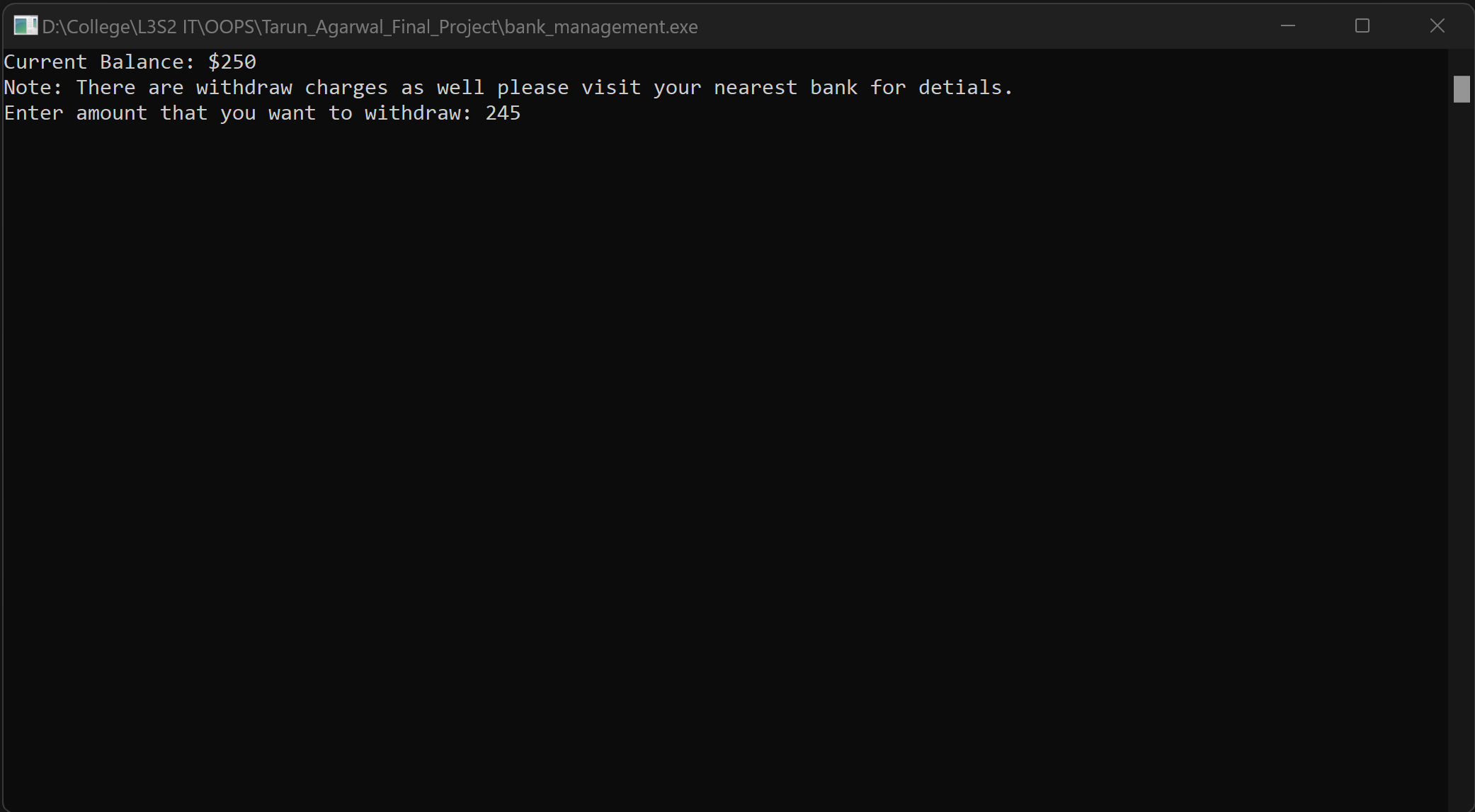


Figure 5: Displaying the withdrawal menu



Figure 6: Displaying the withdrawal amount and balance

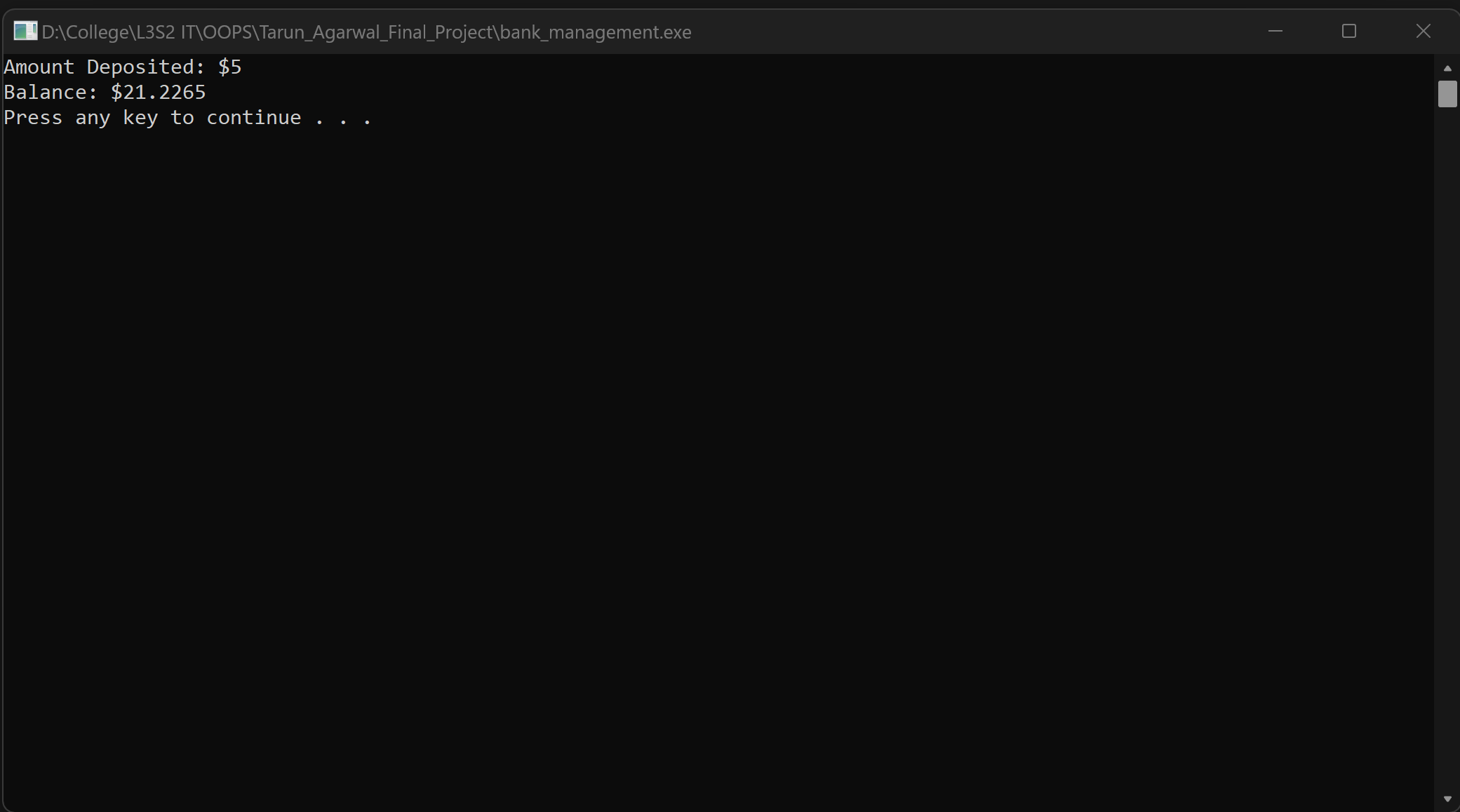


Figure 7: Displaying the Deposited Amount and balance

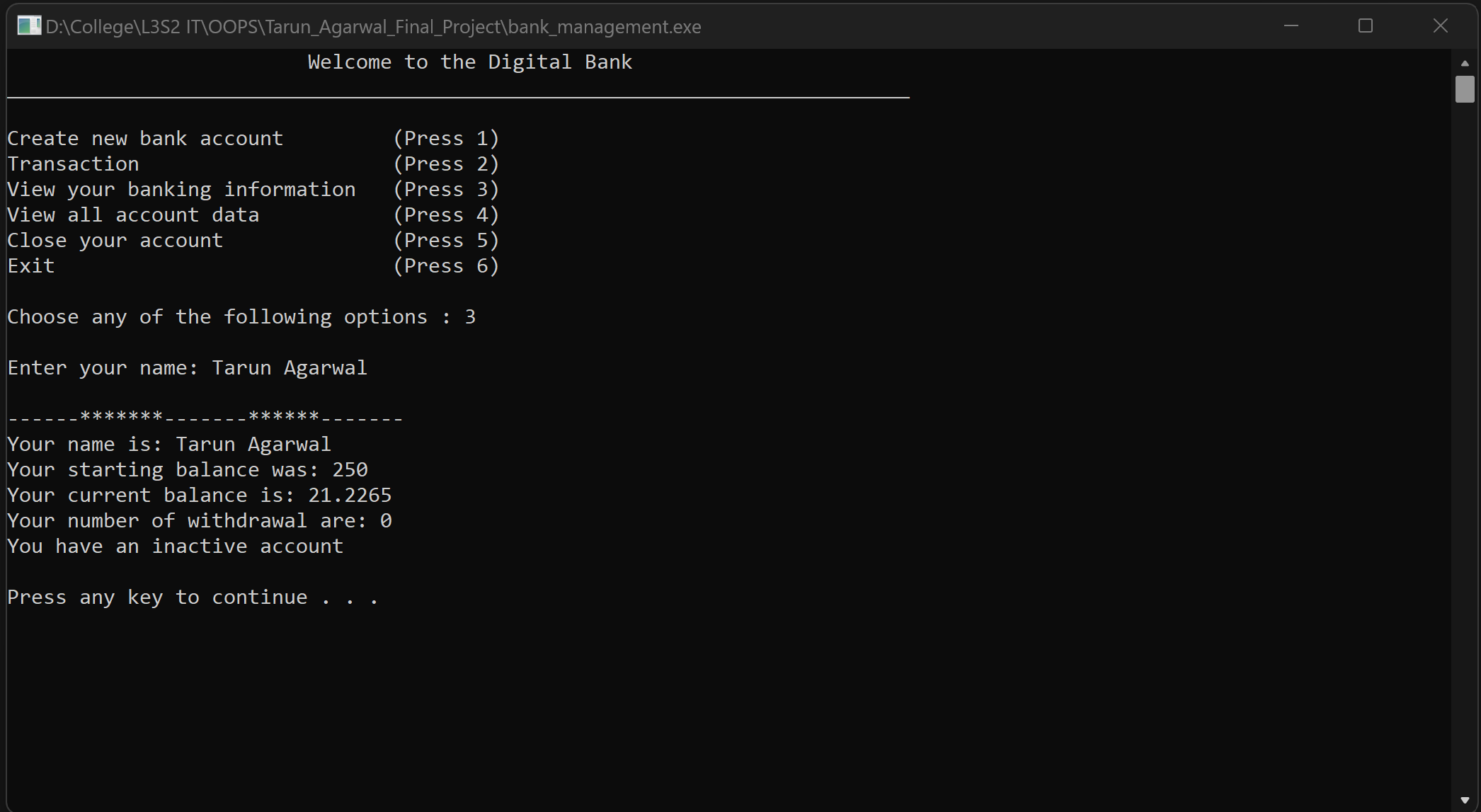


Figure Displaying individual account information (inactive)

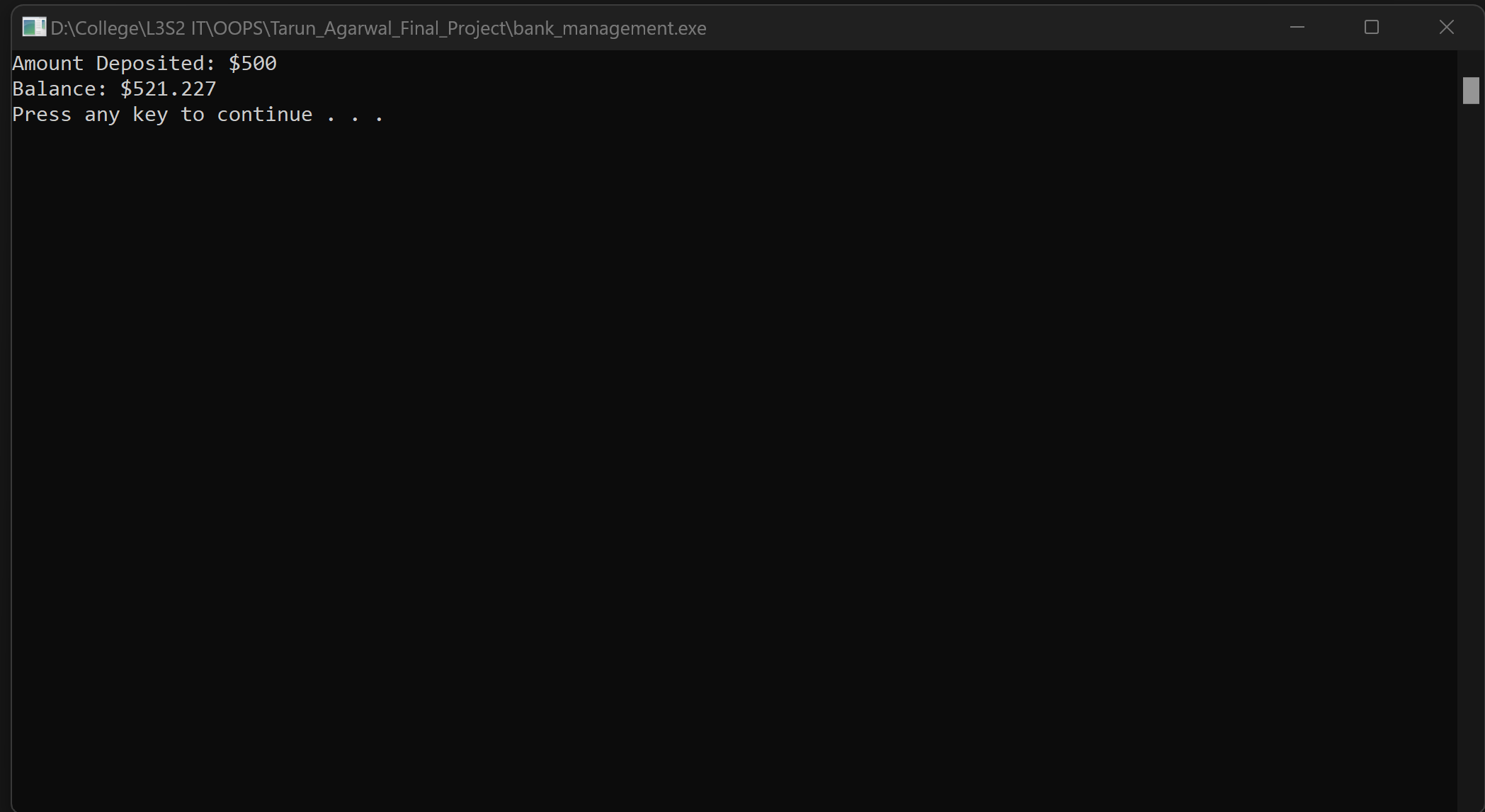


Figure Displaying the Deposited Amount and balance

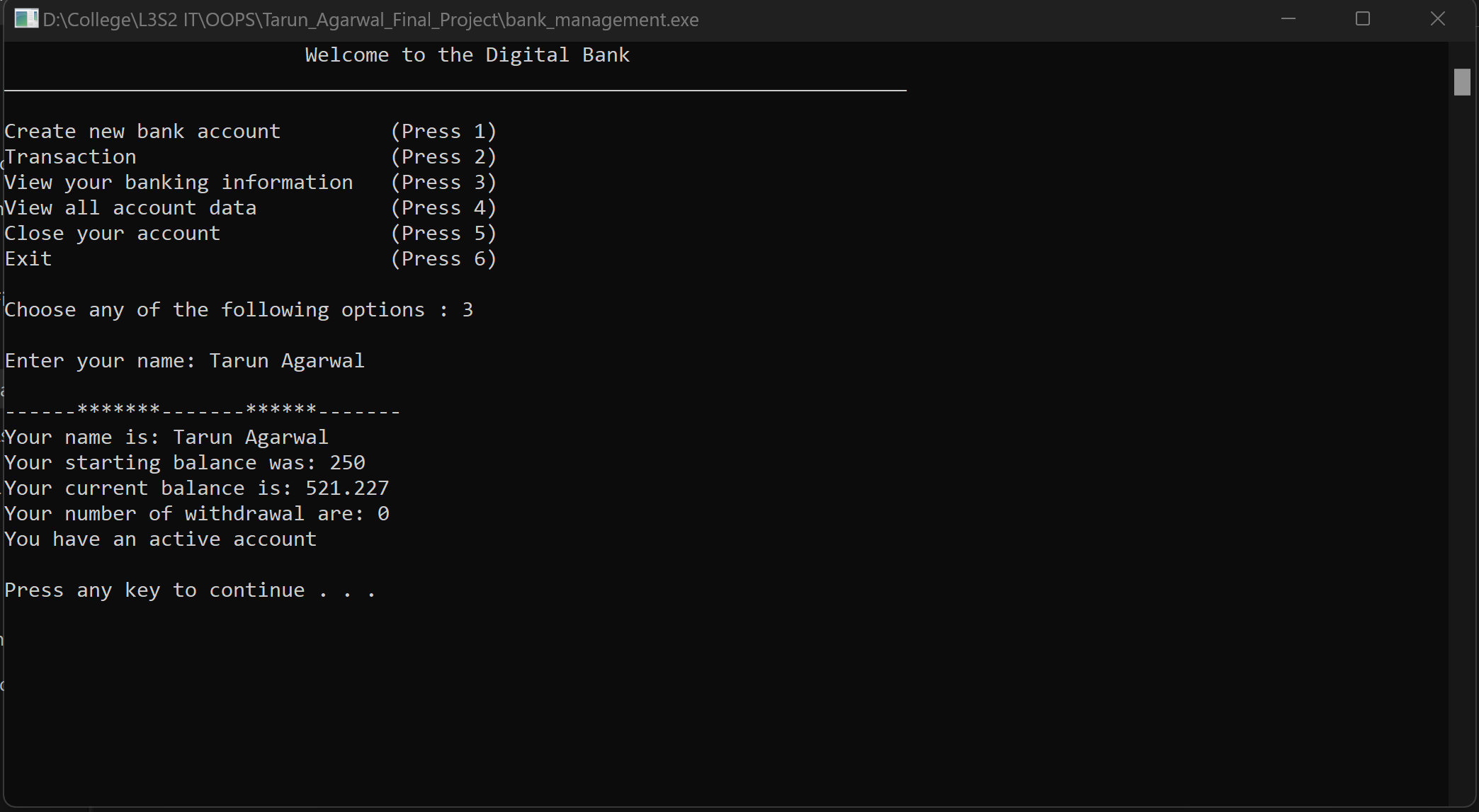


Figure Displaying individual account information (active)

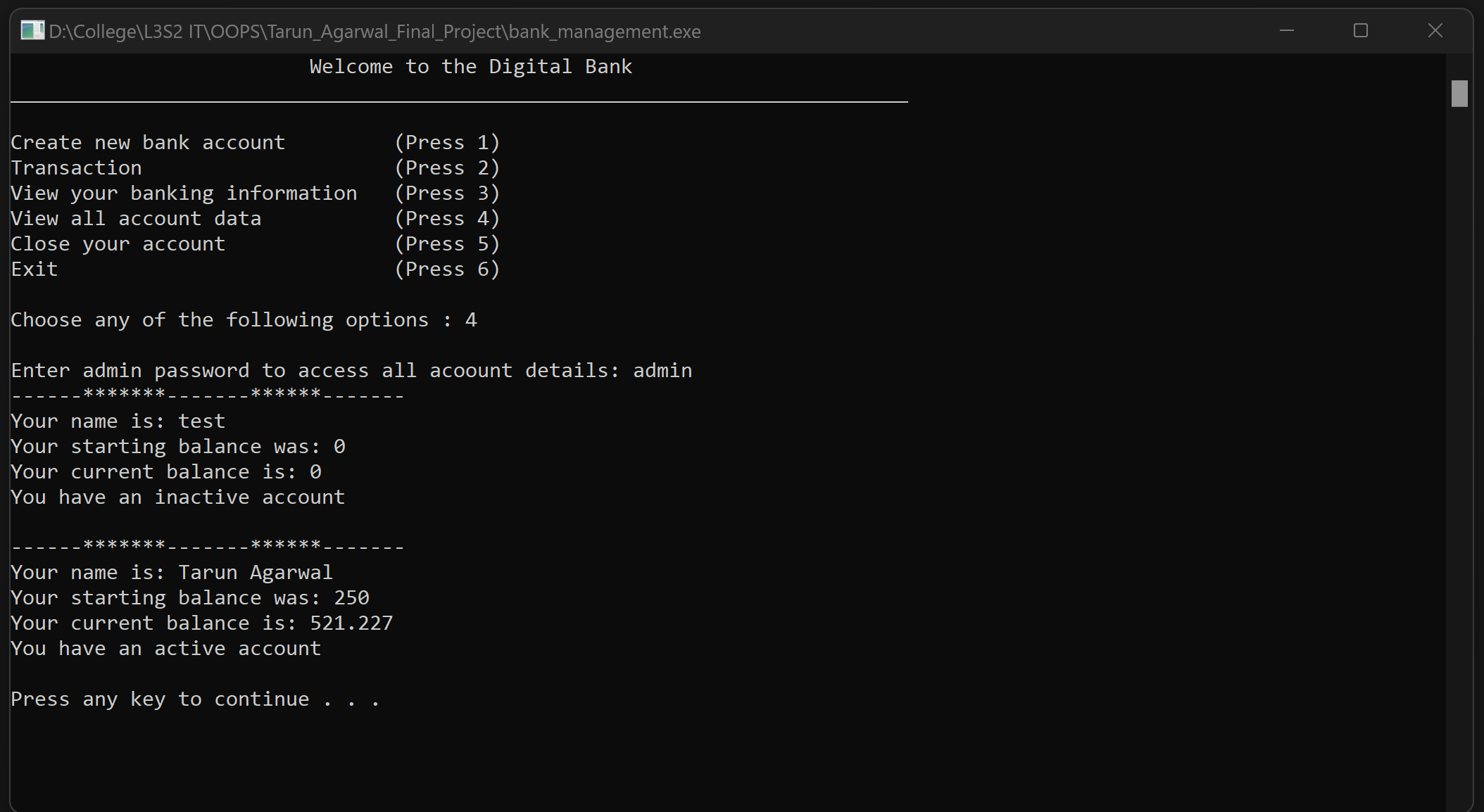


Figure Displaying all account information

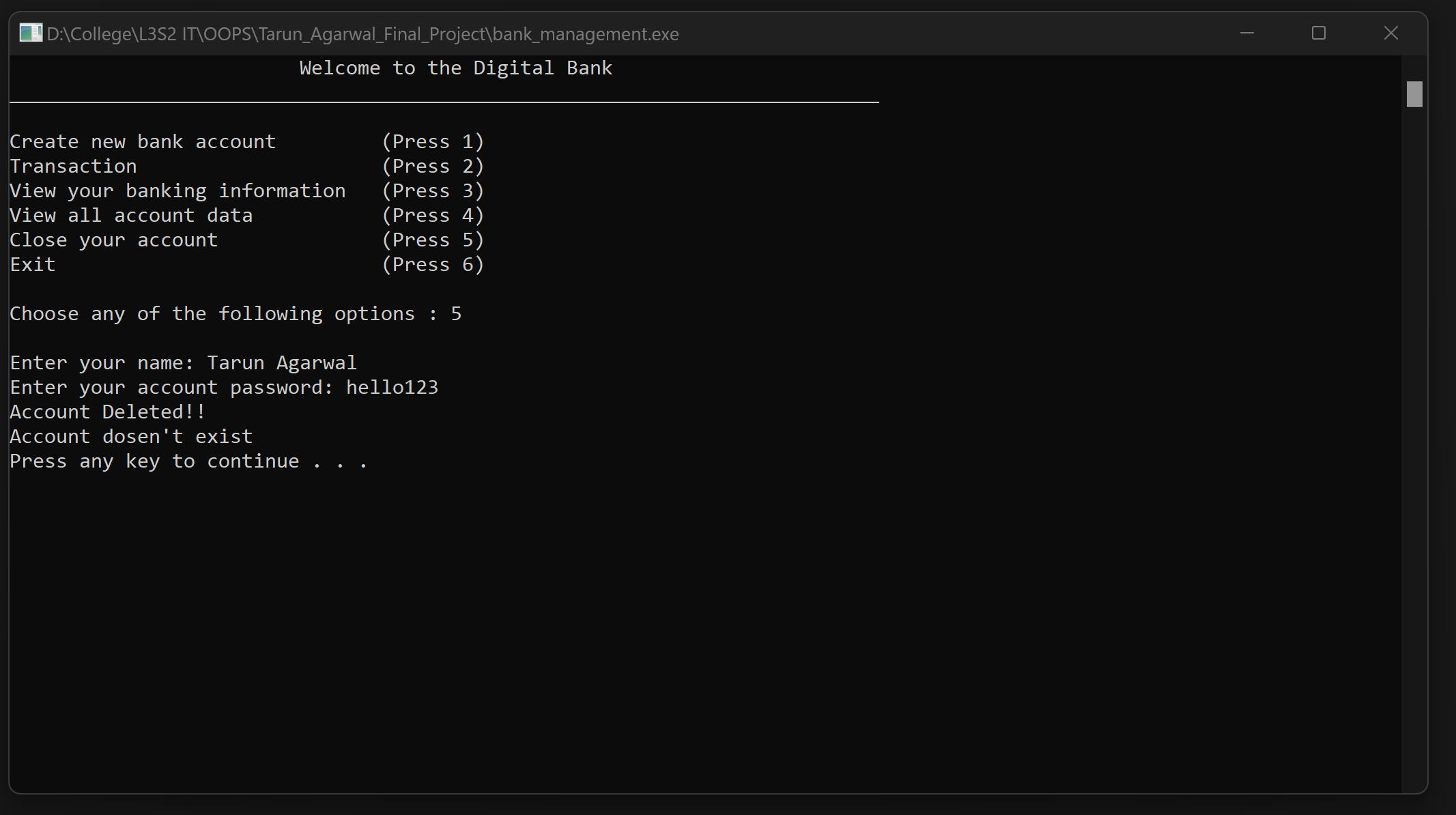


Figure Deleting an Account

## 7.2 User Manual

Once we open the program it will cast six options.

* Create Account, Transaction, View your banking information, view all banking information, close your account and Exit are shown in Main Menu.
* If we choose Create account by pressing ‘1’ , the program will provide us with the options asks the entire detail of the customer like Account Holder’s Name, Password, Initial Balance.
* After filling that and pressing enter, you will again enter into Main Menu.
* If you want to deposit money or withdraw money then press ‘2’ , press ‘3’ for displaying information for your bank account, press ‘4’ for all bank account information, press ’5’ to close your account and exit out of the main menu press ‘6’.
* While depositing and withdrawing amount your user ID and password is necessary. All the amount are deposited in dollar.
* While accessing all account information you will need admin password.
* While closing your account you will need your account password and username.

**Admin Credentials:** admin