BANKING MANAGEMENT SYSTEM

A

Mini Project Report

Submitted in partial fulfilment of the

Requirements for the award of the Degree of

BACHELOR OF ENGINEERING

IN

INFORMATION TECHNOLOGY

By

S. SANTOSH KUMAR - 1602-19-737-164

AKSHAT SHANDILYA - 1602-19-737-126

Department of Information Technology

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Ibrahim Bagh, Hyderabad-31

2020

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Hyderabad-500 031

Department of Information Technology

DECLARATION BY THE CANDIDATE

We, S.SANTOSH KUMAR and AKSHAT SHANDILYA bearing hall ticket numbers, 1602-19-737-164 and 1602-19-737-126, hereby declare that the project report entitled "BANKING MANAGEMENT SYSTEM" is submitted in partial fulfilment of the requirement for the award of the degree of Bachelor of Engineering in Information Technology.

This is a record of bonafide work carried out by us and the results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

S. SANTOSH KUMAR
1602-19-737-164
AKSHAT SHANDILYA
1602-19-737-126

(Faculty In-Charge) (Head.Dept IT)

ACKNOWLEDGEMENTS

We would like to express our special thanks of gratitude to Ms. Divya Lingineni, Assistant Professor, Vasavi College of Engineering who gave us the golden opportunity to do this wonderful project on “Banking Management System” based on C programming, which also helped us in doing a lot of Research and we got to learn about so many new things. We are really obliged to get this opportunity.

We are over helmed in all humbleness and gratefulness to acknowledge our depth to all those who have helped me to put these ideas, well above the level of simplicity and into something concrete.

Any attempt at any level can't be satisfactorily completed without the support and guidance of our friends.

We would like to thank our friends who helped us a lot in gathering different information, collecting data and helping us from time to time in making this project, they gave me different ideas in making this project unique.

S. SANTOSH KUMAR
1602-19-737-164
AKSHAT SHANDILYA
1602-19-737-126

Date-18-12-2020

ABSTRACT

Banking is a profit and service oriented institution. A bank is a place where every person keeps his trust and saves what all he has. The bank in turn tries for costumer friendly services to withstand that trust. Organization need to effectively define and manage requirements to ensure that they are meeting the needs of their customers.

The “BANKING MANAGEMENT SYSTEM” is an application for maintaining a person's account and transactions in a bank. This is an approach to provide an opportunity to the customers to do important transactions easily. This project is to carry out these processes easily and quickly, which is not possible with the manual process. The design and development of this project provides a secured process in managing costumer’s information. The interesting part is our project has access to both customers and employees in which they can perform their respective tasks and implement banking services and ATM services. This uses a multi-level security for bank and costumer safety. Also this project contains many more features when compared to a conventional banking system.

TABLE OF CONTENTS

1. INTRODUCTION

.1. Problem Domain in general

During the past several decades, personnel function has been transformed from a relatively obscure record keeping staff to central and top level management function. There are many factors that have influenced this transformation like technological advances, professionalism, and general recognition of human beings as most important resources.

A computer based management system is designed to handle all the primary information required to keep track of costumer transactions. Separate database is maintained to handle all the details required for calculating tasks and generating details. In the bank, customers deposit and withdraw their money. Transaction of money also is a part where customer takes shelter of the bank. Now to keep the belief and trust of customers, there is the positive need for management of the bank, which can handle all this with comfort and ease. Smooth and efficient management affects the satisfaction of the customers and staff members, indirectly. And of course, it encourages management committee in taking some needed decision for future enhancement of the bank. Now-a-days, managing a bank is tedious job up to certain limit. So software that reduces the work is essential. Also today's world is a genuine computer world and is getting faster and faster day-by-day. Thus, considering above necessities, the application for bank management became necessary which would be useful in managing the bank more efficiently.

1.2 Project Introduction

This mini project using C programming has been designed, which intends to introduce more user friendliness in the various activities such as record updation, maintenance, searching and transactions. The searching of record has been made quite simple as all the details of the customer can be obtained by simply keying the account

number of that customer. Similarly, record maintenance and updation can also be accomplished by using the account number with all the details being automatically generated. These details are also being promptly automatically updated in the master file thus keeping the record absolutely up-to-date.

The entire information has maintained in the files and whoever wants to retrieve can't retrieve, only authorization user can retrieve the necessary information which can be easily be accessible from the file. This system provides fast, efficient, reliable and User friendly interfaces in banking and has no chance of losing data while processing of user data i.e., customer account transactions. This application provides a good user interface such that a user of basic computer knowledge can operate the application. It also reduces effort done by the accountant and also reduces the load of real time computation. This application enables faster transaction like new account creation, withdrawal of cash from the account, deposit of cash to the account, checking account balance of the account holder even if there are large amount of data in the system. This application also enables user to update their details if they are incorrect, with some restrictions on immutable details and some fields only the employee can update them.

This application also performs various activities that help the employees in handling costumer records and also maintaining the banking services of enabling and disabling the transactions like deposit/withdraw for costumers.

1.3 List of features in the project

This Project takes the two-ended feature of employee end and Account holder end.

Features of Account Holder

- Create New Account
- Login, View, Update details in existing account
- View Account Balance and Transaction History
- Perform transactions such as Deposit and Withdraw
- Change Account PIN for securing account
- Close Bank Account

Features of Employee

- Login and View list of Accounts in bank
- View specific account holder details and update them
- Delete account of a costumer
- Manage ATM services such as Adding money in ATM, disable ATM services i.e., deposit & withdrawals.

2. TECHNOLOGY

All computer software needs certain hardware components and also other software resources to be present, in order for computers to be used efficiently. These pre-requisites are known as System Requirements. System Requirements are of two types – Software Requirements and Hardware Requirements.

2.1. Software Requirements

Software Requirements deal with defining the software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. These preconditions are generally not included in the software package and need to be installed separately.

In order to use Banking Management System, one should have the following software requirements:

- Operating System: Windows 7 and above
- C compiler: GNU Compiler Collection (GCC).
- Editor: Any Text Editor that supports C language

2.2. Hardware Requirements

Hardware Requirements refer to the common set of requirements defined by any operating system or software application and are usually the physical computer resources. In this we look into the architecture, processing power, memory, secondary memory, display adapter and peripherals.

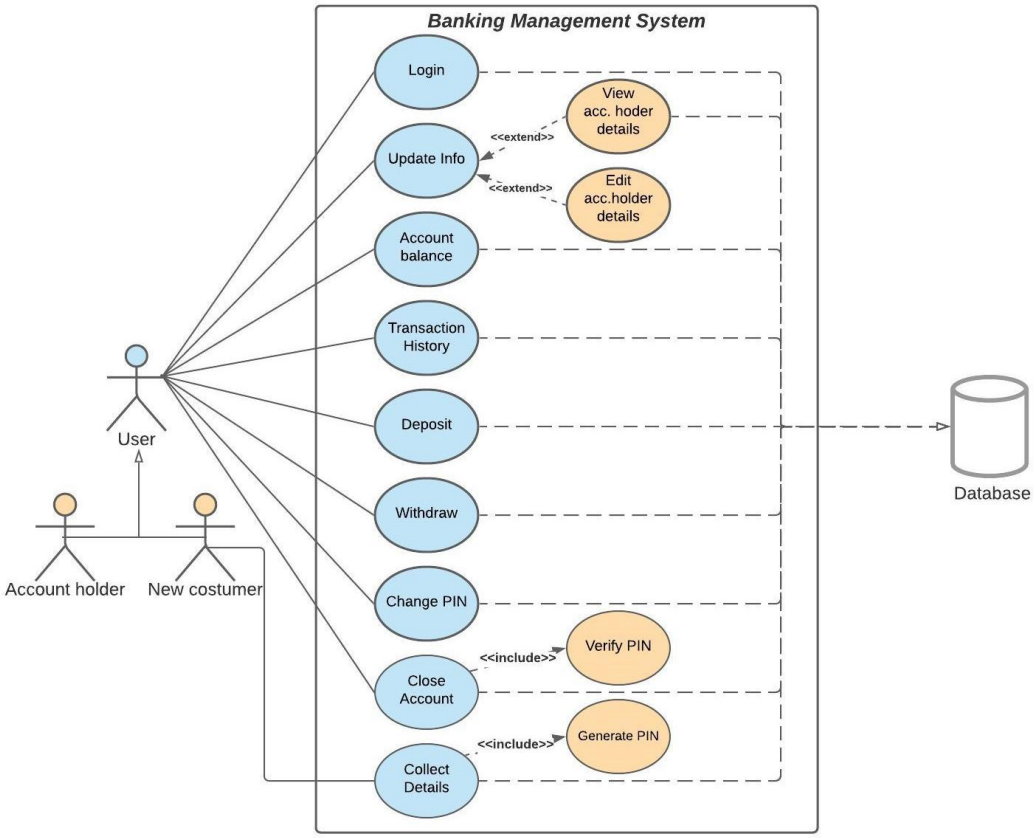
In order to use Banking Management System, one should have the following hardware requirements:

- Processor: Intel Core i5 and above
- Memory: 4 GB RAM and above

3. PROPOSED WORK

3.1. Design

3.1.1. Account Holder Use Cases



3.1.1.1. New Account Holder

Any user can create an account in Banking Management System. For this, the user has to select the option to create new account and the process proceeds as below.

3.1.1.1.1. Collect Details

First, the system prompts the user to enter list of details (full name, father name, age, gender, DOB, phone no etc.) While entering, it checks whether the details are in correct format. If it is not entered correctly, it displays an error message and asks to re-enter until he/she enters in correct format. Once it is entered correctly it asks to enter another detail.

3.1.1.1.2. Generate PIN

If all the details are entered correctly, then the system generates a unique Account number and PIN and displays it on the screen for the user to note them down. Finally, an account is successfully created for the user and his/her details are stored in the database.

3.1.1.2. Login

The system prompts the user to enter account number and PIN. Once the user enters the details, it checks whether the account number and PIN are entered correctly and they match with the database. If the details are matched, the user is logged in and they are redirected to User Menu where they can perform their banking tasks. If the credentials are invalid, it displays error message. If the entered PIN is incorrect, it prompts the user to try again or go back to Main Menu. If the account number itself is wrong, then the page is automatically redirected to Main Menu.

3.1.1.3. View Account Details

When the user selects this option, the system retrieves the data from the file and displays his details on the screen. The user can view the details and can go back to menu after the task is completed.

3.1.1.4. Update Account Details

When the user selects this option, the existing account details of the user are displayed with serial no. Then the system asks to enter the serial number of the detail to update with new data. If the entered input is valid, then he/she can enter the updated details and the details will be successfully updated in database. User can’t edit all the fields of details (like aadhaar number). So, if the input is invalid (Invalid Sl.no or Sl.no of immutable detail is entered) then an error is displayed.

3.1.1.5. View Account Balance

The account balance of the account holder is retrieved from the file and displayed on the screen. Here the user is provided with two options to navigate. First, to view Transaction History and second is to go back to main menu. User has to input his choice and that action is performed. It displays an error message when an invalid input is entered.

3.1.1.6. View Transaction History

When the user chooses this option, it retrieves data from the file and displays his/her transaction history on the screen. This contains the transaction type (deposit or withdrawal) along with date and time of transaction. If the user didn't perform any transactions, it shows an message "No Transactions Found".

3.1.1.7. Deposit

Deposit amount feature allows the users to deposit the amount into his account. For this, the user has to enter the amount twice, one time to process and another time to verify the amount again. If the verification is failed, then it displays an error message and he is redirected to user menu. If the ATM Amount limit (10 lakhs) is reached, then it shows a message that the transaction can't be processed. If the verification is correct and ATM limit is not reached then the transaction is processed and it displays a message that amount is successfully deposited. At last, the user is shown a menu where he can navigate to view account balance or view Transaction History or to go back to user menu.
The input is taken from the user and respective task is performed.

3.1.1.8. Withdraw

Withdraw amount feature allows the users to withdraw the amount from their account. For this, the user has to enter the amount twice, one time to process and another time to verify the amount again. Here the system checks many exceptions and displays an error message

- If the verification is failed
- If the entered amount is greater than account balance
- If there is no enough cash in the ATM
- If the requested withdrawal amount is high(>50,000)

If all the above test cases are passed then the transaction will be successfully processed and it displays a message on the screen.

At last, the user is shown a menu where he can navigate to view account balance or view Transaction History or to go back to user menu. The input is taken from the user and respective task is performed.

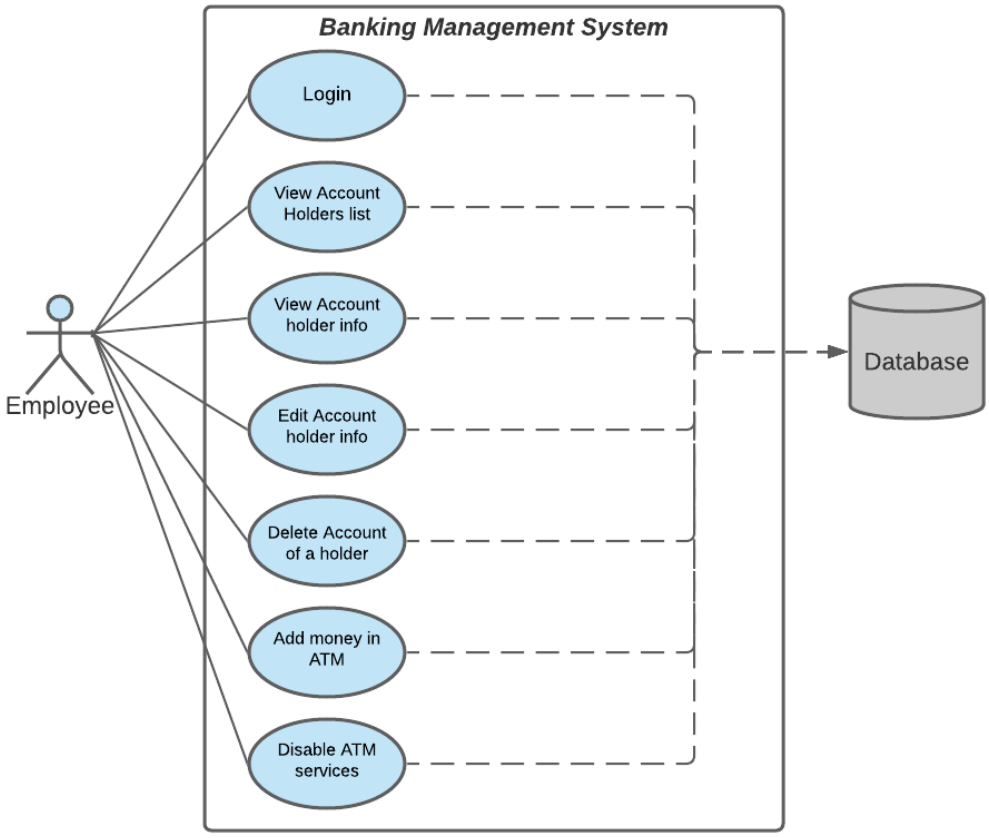
3.1.1.9. Change PIN

If the user has a doubt of account privacy that their password is known to other people or if they find any irregular activities from their account then they can change their account PIN. For this, when the user selects this option, it displays a verification page where he/she is supposed to verify their present account number and password. If the entered credentials are correct, then it allows him to enter new PIN. Finally, the PIN is updated in the data base, the user is logged out from the account and redirected to Main Menu.

3.1.1.10. Close Account

If the user wishes to close his bank account, he/she is supposed to verify their present account number and password. If the entered credentials are correct, then the task will be processed and his data will be deleted from the File. If the account details are wrong then it displays an error message and he has the option to try again or go back to user menu.

3.1.2. Employee Use Cases



3.1.2.1. Admin Login

This login in interface of the employee is hard-coded. An Employee needs to login to get access to these features and perform the tasks. This is secured with a password. On entering wrong password, the system displays an error message and redirects to main menu.

3.1.2.2. View Account Holders List

When the employee chooses this option, list of account holders along with their account number and phone number is displayed on the screen. The details are stored in a file, which are retrieved when this function is called.

3.1.2.3. View Account Holder Details

Employee can also view all the details of a specific account holder by simply keying the account number. The system checks if there is an account with specified account number. If present, then all the details are displayed on the screen, else it displays an error message “Invalid Account Details”.

3.1.2.4. Edit Account Holder Details

When the employee chooses this option, the system asks for the account number of the costumer whose account details he want to update. Unlike edit option for costumer, employee can edit all fields of details except account number and account balance which are immutable. This process is same as mentioned in account holder use case where the system asks for serial number and user can edit if the input is valid.

3.1.2.5. Delete An Account

Employee also has the option to delete account of any account holder. Employee has to input the account number of the costumer. If the account number is valid, then the account is deleted from the database else it shows an error message.

3.1.2.6. Add Money to ATM

When the ATM runs out of cash, then the withdrawal process can’t be performed for the users. At such times, the only way is employee has to add money into the ATM. To do this, employee has to verify the login again to make the process multi-secure. If the login verification is correct then the employee can add money into the ATM else it displays an error message. Also, while adding money employee can add amount less than 10 lakhs. If the amount is invalid or out of range, then it displays an error message. If the process is completed successfully, then the ATM cash is updated and a message is displayed on the screen.

3.1.2.7. Disable ATM Services

Employee can also disable the ATM services temporarily if any irregularities are found in the application and if they are needed to be updated. During this time, all other functions are performed normally without any effect. When this function is called first time, the ATM Services are disabled and can be enabled back only after performing this same task again. Before performing this operation, employee has to verify his login credentials again. So this can be ensured as a multi-level secured process.

3.2. IMPLEMENTATION

3.2.1. Module-Wise Code

3.2.1.1. Display functions

- **Function Name: void LoadingPage()**

```
void loadingPage(char message[])

{

   system("cls");

   gotoxy(40,10);
printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\
xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xD
B\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\
xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xD
B\xDB\xDB\xDB\xDB\xDB");

   menuBorder();

   int y=16,y1=18;

   for(int i=0;i<9;i++)

   {

      gotoxy(51,y++);


printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB
2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\
xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");

   }

   for(int i=0;i<5;i++)

   {
```

```
        gotoxy(52,y1++);
        printf("                                ");
    }
    if(strcmp(message,"WELCOME TO VCE STUDENT BANK.") == 0)
    {
        gotoxy(58,y1-3);
    }
    else if(strcmp(message,"THANK YOU") == 0)
    {
        gotoxy(68,y1-3);
    }
    for(int i=0;i<strlen(message);i++)
    {
        printf("%c",message[i]);
        Sleep(70);
    }
    Sleep(500);
}
```

- **Function Name : void DisplayWelcomePage()**

```
void displayWelcomePage()
{
    int choice;
```

```
   clrscr();

   gotoxy(40,10);

   printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\tWELCOME TO
BANKING MANAGEMENT
SYSTEM\t\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB");

   menuBorder();

   gotoxy(40,14);      printf("\xDB\t\xB2\xB2\xB2 1. LOGIN TO YOUR
ACCOUNT");

   gotoxy(40,16);      printf("\xDB\t\xB2\xB2\xB2 2. CREATE AN ACCOUNT");

   gotoxy(40,18);      printf("\xDB\t\xB2\xB2\xB2 3. TO LOGIN AS
EMPLOYEE");

   gotoxy(40,20);      printf("\xDB\t\xB2\xB2\xB2 4. EXIT");

   gotoxy(42,26);      printf("Enter your Choice : ");       scanf("%d",&choice);
fflush(stdin);

   switch(choice)

   {

      case 1:     loginPageAccountHolder();       break;

      case 2:     displayNewAccountPage();        break;

      case 3:     loginPageEmployee();            break;

      case 4:     break;

      default:    printf("\033[1;31m");

                  gotoxy(64,28);      printf("INVALID CHOICE!!");

                  printf("\033[0m");

                  Sleep(1500);        displayWelcomePage();

                  break;          }         }
```

- **Function Name : void displayAccountHolderMenu(long int)**

```
void displayAccountHolderMenu(long int accountNo)
{
   clrscr();
   int option;
   gotoxy(40,10);
printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\
xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\tMENU\t\xDB\xD
B\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\
xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB");
   menuBorder();
   gotoxy(40,13);  printf("\xDB\t1. VIEW ACCOUNT DETAILS ");
   gotoxy(40,14);  printf("\xDB\t2. UPDATE ACCOUNT DETAILS ");
   gotoxy(40,15);  printf("\xDB\t3. VIEW ACCOUNT BALANCE ");
   gotoxy(40,16);  printf("\xDB\t4. VIEW TRANSACTION HISTORY ");
   gotoxy(40,17);  printf("\xDB\t5. DEPOSIT ");
   gotoxy(40,18);  printf("\xDB\t6. WITHDRAW ");
   gotoxy(40,19);  printf("\xDB\t7. CHANGE PIN ");
   gotoxy(40,20);  printf("\xDB\t8. CLOSE ACCOUNT ");
   gotoxy(40,21);  printf("\xDB\t9. LOGOUT ");
   gotoxy(42,26);  printf("Enter your Choice : ");     scanf("%d",&option);
   switch(option)
   {
      case 1:     viewAccDetails(accountNo,1);         break;
      case 2:     updateAccDetails(accountNo);         break;
      case 3:     viewAccountBalance(accountNo);       break;
```

```
        case 4:     viewTransactionHistory(accountNo);   break;
        case 5:     depositAmount(accountNo);            break;
        case 6:     withdrawAmount(accountNo);           break;
        case 7:     changePin(accountNo);                break;
        case 8:     closeAccount(accountNo);             break;
        case 9:     displayWelcomePage();                break;
        default:    printf("\033[1;31m");
                    gotoxy(64,27);  printf("INVALID ENTRY!!!");     Sleep(1000);
                    printf("\033[0m");
                    displayAccountHolderMenu(accountNo);     break;
    }
}
```

- **Function Name : void displayEmployeeMenu()**

```
void displayEmployeeMenu()
{
    clrscr();
    int option;
    gotoxy(40,10);
printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\tCHOOSE ANY
OPTION\t\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB");
    menuBorder();
```

```
   gotoxy(40,14);  printf("\xDB\t1. VIEW ACCOUNT HOLDERS LIST ");

   gotoxy(40,15);  printf("\xDB\t2. VIEW ACCOUNT HOLDER DETAILS");

   gotoxy(40,16);  printf("\xDB\t3. EDIT ACCOUNT HOLDER DETAILS ");

   gotoxy(40,17);  printf("\xDB\t4. DELETE AN ACCOUNT ");

   gotoxy(40,18);  printf("\xDB\t5. ADD MONEY TO ATM ");

   gotoxy(40,19);  printf("\xDB\t6. DISABLE ATM SERVICES ");

   gotoxy(40,20);  printf("\xDB\t7. LOGOUT");

   gotoxy(40,21);  printf("\xDB\t8. EXIT");

   gotoxy(42,25);  printf("Enter your Choice : ");        scanf("%d",&option);
fflush(stdin);

   switch(option)

   {

      case 1:     AccountsList();                  break;

      case 2:     viewAccHolderDetails();           break;

      case 3:     editAccHolderDetails();           break;

      case 4:     closeAccHolderAccount();          break;

      case 5:     addMoneyToATM();                  break;

      case 6:     disableATMServices();             break;

      case 7:     displayWelcomePage();             break;

      case 8:     break;
```

```
        default:    gotoxy(65,26);  printf("INVALID ENTRY!!!");     Sleep(1000);
                    displayEmployeeMenu();     break;
    }
}
```

- **Function Name : void MenuBorder()**

```
void menuBorder()
{
   gotoxy(40,11);  printf("\xDB");     gotoxy(105,11);  printf("\xDB");
   gotoxy(40,12);  printf("\xDB");     gotoxy(105,12);  printf("\xDB");
   gotoxy(40,13);  printf("\xDB");     gotoxy(105,13);  printf("\xDB");
   gotoxy(40,14);  printf("\xDB");     gotoxy(105,14);  printf("\xDB");
   gotoxy(40,15);  printf("\xDB");     gotoxy(105,15);  printf("\xDB");
   gotoxy(40,16);  printf("\xDB");     gotoxy(105,16);  printf("\xDB");
   gotoxy(40,17);  printf("\xDB");     gotoxy(105,17);  printf("\xDB");
   gotoxy(40,18);  printf("\xDB");     gotoxy(105,18);  printf("\xDB");
   gotoxy(40,19);  printf("\xDB");     gotoxy(105,19);  printf("\xDB");
   gotoxy(40,20);  printf("\xDB");     gotoxy(105,20);  printf("\xDB");
   gotoxy(40,21);  printf("\xDB");     gotoxy(105,21);  printf("\xDB");
   gotoxy(40,22);  printf("\xDB");     gotoxy(105,22);  printf("\xDB");
   gotoxy(40,23);  printf("\xDB");     gotoxy(105,23);  printf("\xDB");
```

```
    gotoxy(40,24);  printf("\xDB");     gotoxy(105,24);  printf("\xDB");

    gotoxy(40,25);  printf("\xDB");     gotoxy(105,25);  printf("\xDB");

    gotoxy(40,26);  printf("\xDB");     gotoxy(105,26);  printf("\xDB");

    gotoxy(40,27);  printf("\xDB");     gotoxy(105,27);  printf("\xDB");

    gotoxy(40,28);  printf("\xDB");     gotoxy(105,28);  printf("\xDB");

    gotoxy(40,29);  printf("\xDB");     gotoxy(105,29);  printf("\xDB");

    gotoxy(40,30);
  printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\
  xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xD
  B\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\
  xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xD
  B\xDB\xDB\xDB\xDB\xDB\n");

  }
```

3.2.1.2. Account Holder use case functions

- **Function Name : void displayNewAccountPage()**

```
void displayNewAccountPage()

{

   clrscr();

   int valid,ok;

   ptr = fopen("records.txt","a+");
```

```
    gotoxy(40,10);

printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\tCREATE AN
ACCOUNT\t\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB");

    menuBorder();

    gotoxy(45,12);  printf("FULL NAME: ");

    gotoxy(45,13);  printf("FATHER NAME: ");

    gotoxy(45,14);  printf("AGE: ");

    gotoxy(45,15);  printf("GENDER (M/F): ");

    gotoxy(45,16);  printf("NATIONALITY: ");

    gotoxy(45,17);  printf("DATE OF BIRTH(DDMMYYYY): ");

    gotoxy(45,18);  printf("PHONE NO: ");

    gotoxy(45,19);  printf("AADHAAR NUMBER: ");

    gotoxy(45,20);  printf("MARITAL STATUS(Y/N): ");

    gotoxy(45,21);  printf("EDUCATION LEVEL: ");

    gotoxy(45,22);  printf("OCCUPATION: ");

    gotoxy(45,23);  printf("Initial Deposit: ");


    Account.AccNo = generateAccNo();
```

```
Account.PIN = (rand() % 9000)+1000 ;
fullName:
gotoxy(75,12);      gets(Account.fullName);
gotoxy(45,27);      printf("                                        ");
if(strlen(Account.fullName)>25||strlen(Account.fullName)<5)
      {
   printf("\033[1;31m");
            gotoxy(50,27);  printf("FullName should be less than 25 characters !");
   printf("\033[0m");
   gotoxy(75,12);  printf("                        ");
            goto fullName;
      }
else
      {
            for (int b=0;b<strlen(Account.fullName);b++)
            {
                  if (isalpha(Account.fullName[b]) || Account.fullName[b] == ' ')
                  {
                        valid=1;
```

```
                }
                else
                {
                        valid=0;
                        break;
                }
        }
        if(!valid)
        {
    printf("\033[1;31m");
                gotoxy(49,27);  printf("FullName can contain only Alphabets and
spaces !");
    printf("\033[0m");
    gotoxy(75,12);  printf("                    ");
                goto fullName;
        }
    }
  gotoxy(45,27);    printf("                                    ");


  fatherName:
```

```
gotoxy(75,13);      gets(Account.fatherName);
gotoxy(45,27);      printf("                                        ");
if(strlen(Account.fatherName)>25||strlen(Account.fatherName)<5)
      {
   printf("\033[1;31m");
            gotoxy(48,27);  printf("FatherName should be less than 25 characters !");
   printf("\033[0m");
   gotoxy(75,13);  printf("                    ");
            goto fatherName;
      }
else
      {
            for (int b=0;b<strlen(Account.fatherName);b++)
            {
            if (isalpha(Account.fatherName[b]) || Account.fatherName[b] == ' ')
                  {
                        valid=1;
                  }
                  else
```

```
                {
                        valid=0;
                        break;
                }
        }
        if(!valid)
        {
    printf("\033[1;31m");
                gotoxy(47,27);  printf("FatherName can contain only Alphabets and spaces !");
    printf("\033[0m");
    gotoxy(75,13);  printf("                    ");
                goto fatherName;
        }
    }
  gotoxy(45,27);    printf("                                ");


  age:
  gotoxy(75,14);    scanf("%d",&(Account.age));    fflush(stdin);
  gotoxy(45,27);    printf("                                ");
```

```
    if(Account.age >100 || Account.age<18)
    {
       printf("\033[1;31m");
       gotoxy(56,27);   printf("Age limit should be 18-100 !");
       printf("\033[0m");
       gotoxy(75,14);   printf("     ");
       goto age;
    }
    gotoxy(45,27);      printf("                                        ");
    gender:
    gotoxy(75,15);      gets(Account.gender);
    gotoxy(45,27);      printf("                                        ");
    for(int b=0;b<strlen(Account.gender);b++)
    {
       if(toupper(Account.gender[b])=='M'|| toupper(Account.gender[b])=='F' ||
toupper(Account.gender[b])=='O')
       {
          ok = 1;
       }
       else
```

```
    {
        ok = 0;
        break;
    }
}
if(!ok)
{
    printf("\033[1;31m");
    gotoxy(56,27);   printf("Gender can contain only M/F/O !");
    printf("\033[0m");
    gotoxy(75,15);   printf("     ");
    goto gender;
}
gotoxy(45,27);      printf("                                        ");
nationality:
gotoxy(75,16);      gets(Account.nationality);
gotoxy(45,27);      printf("                                        ");
if(strlen(Account.nationality)>25||strlen(Account.nationality)<5)
        {
```

```
    printf("\033[1;31m");
                gotoxy(48,27);  printf("Nationality should be less than 25 characters !");
    printf("\033[0m");
    gotoxy(75,16);  printf("                        ");
                goto nationality;
        }
else
        {
                for (int b=0;b<strlen(Account.nationality);b++)
                {
                        if (isalpha(Account.nationality[b]) || Account.nationality[b] == ' ')
                        {
                                valid=1;
                        }
                        else
                        {
                                valid=0;
                                break;
                        }
```

```
                }
                if(!valid)
                {
        printf("\033[1;31m");
                        gotoxy(46,27);  printf("Nationality can contain only Alphabets
and spaces !");
        printf("\033[0m");
        gotoxy(75,16);  printf("                        ");
                        goto nationality;
                }
        }
    gotoxy(45,27);      printf("                                        ");


    gotoxy(75,17);      scanf("%ld",&(Account.DOB));
    gotoxy(45,27);      printf("                                        ");
    phoneNo:
    gotoxy(75,18);      scanf("%lld",&(Account.phoneNo));   fflush(stdin);
    char phNo[20];
    itoa(Account.phoneNo,phNo,10);
    if(strlen(phNo)>10||strlen(phNo)!=10)
```

```
{
   printf("\033[1;31m");
   gotoxy(56,27);       printf("PhoneNo should be 10 digits !");
   printf("\033[0m");
   gotoxy(75,18);      printf("                    ");
   goto phoneNo;
}
gotoxy(45,27);      printf("                                    ");
AadhaarNo:
gotoxy(75,19);      gets(Account.AadhaarNo);
gotoxy(45,27);      printf("                                    ");
char AadNo[20];
strcpy(AadNo,Account.AadhaarNo);
if(strlen(AadNo)>12 || strlen(AadNo)!=12)
{
   printf("\033[1;31m");
   gotoxy(56,27);       printf("AADHAAR NO should be 16 digits !");
   printf("\033[0m");
   gotoxy(75,19);   printf("                    ");
```

```
        goto AadhaarNo;
    }
    gotoxy(45,27);      printf("                                        ");
    MaritalStatus:
    gotoxy(75,20);      gets(Account.MaritalStatus);
    gotoxy(45,27);      printf("                                        ");
    for(int b=0;b<strlen(Account.MaritalStatus);b++)
    {
        if(toupper(Account.MaritalStatus[b])=='Y'||
toupper(Account.MaritalStatus[b])=='N')
        {
            ok = 1;
        }
        else
        {
            ok = 0;
            break;
        }
    }
    if(!ok)
```

```
{
   printf("\033[1;31m");
   gotoxy(54,27);   printf("MaritalStatus can contain only Y/N !");
   printf("\033[0m");
   gotoxy(75,20);   printf("     ");
   goto MaritalStatus;
}
gotoxy(45,27);     printf("                                    ");
gotoxy(75,21);     gets(Account.EducationLevel);
gotoxy(45,27);     printf("                                    ");
gotoxy(75,22);     gets(Account.occupation);
gotoxy(45,27);     printf("                                    ");
accBal:
gotoxy(75,23);     scanf("%d",&Account.accBal);
gotoxy(45,27);     printf("                                    ");
if(Account.accBal<0 || Account.accBal>50000)
{
   printf("\033[1;31m");
   gotoxy(52,27);   printf("Acc Bal should in range of 0 - 50,000 !");
```

```
        printf("\033[0m");
        gotoxy(75,23);   printf("         ");
        goto accBal;
    }
    gotoxy(45,27);      printf("                                        ");
    fprintf(ptr,"%ld %d %s %s %d %s %s %ld %lld %s %s %s %s
%d\n",Account.AccNo,Account.PIN,Account.fullName,Account.fatherName,Account.a
ge,Account.gender,Account.nationality,Account.DOB,Account.phoneNo,Account.Aadh
aarNo,Account.MaritalStatus,Account.EducationLevel,Account.occupation,Account.acc
Bal);
    fclose(ptr);
    gotoxy(56,25);      printf("***ACCOUNT SUCCESSFULLY CREATED***");
Sleep(500);
    gotoxy(60,26);      printf("YOUR ACCOUNT NUMBER :%ld",Account.AccNo);
    gotoxy(61,27);      printf("YOUR PASSWORD: %d",Account.PIN);
    gotoxy(43,29);      printf("PRESS ANY KEY TO GO BACK : ");    getch();
    displayWelcomePage();
}
```

- **Function Name : loginPageAccountPage()**

```
void loginPageAccountHolder()
{
```

```
    clrscr();
    long int Ano;
    int pw,flag=0;
    gotoxy(40,10);

printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\tLOGIN TO YOUR ACCOUNT\t\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB");

    menuBorder();
    gotoxy(40,14);      printf("\xDB\t\xB2\xB2\xB2 ENTER ACCOUNT NUMBER : ");
    gotoxy(40,16);      printf("\xDB\t\xB2\xB2\xB2 ENTER PIN : ");
    gotoxy(75,14);      scanf("%d",&Ano);   gotoxy(75,16);      scanf("%d",&pw);

    ptr = fopen("records.txt","r");
    while(fscanf(ptr,"%ld %d %s %s %d %s %s %ld %lld %s %s %s %s %d\n",&traverse.AccNo,&traverse.PIN,traverse.fullName,traverse.fatherName,&traverse.age,traverse.gender,traverse.nationality,&traverse.DOB,&traverse.phoneNo,traverse.AadhaarNo,traverse.MaritalStatus,traverse.EducationLevel,traverse.occupation,&traverse.accBal) != EOF)
    {
       if(traverse.AccNo == Ano)
       {
```

```
        flag=1;
        if(traverse.PIN == pw)
        {
           gotoxy(58,23);     printf("******PASSWORD MATCHED******");
Sleep(500);
           gotoxy(62,25);     printf("-----LOGGING IN-----");          Sleep(1000);
           fclose(ptr);
           displayAccountHolderMenu(Ano);
        }
        else
        {
           fclose(ptr);
           gotoxy(57,23);     printf("WARNING : WRONG PASSWORD!!!!!");
Sleep(1200);
           gotoxy(47,25);     printf("ENTER 1 TO TRY AGAIN..0 TO RETURN TO
MAIN MENU..");
           int opt;    gotoxy(60,27);   printf("ENTER YOUR CHOICE: ");
scanf("%d",&opt);
           if(opt==1)
           {
              loginPageAccountHolder();
```

```
            }
            else
            {
               displayWelcomePage();
            }
          }
       }
   }
   if(flag != 1)
   {
      fclose(ptr);
      gotoxy(60,23);     printf("INVALID ACCOUNT DETAILS!!!");   Sleep(1000);
      gotoxy(61,25);     printf("RETURNING TO MAINMENU!!!");     Sleep(1500);
displayWelcomePage();
   }
}
```

- **Function Name : viewAccDetails(long int, int)**

```
int viewAccDetails(long int accountNo, int flagVar)
{
   clrscr();
```

```
    int choice,flag=0;

    gotoxy(40,10);

printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xD
B\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\t   VIEW
DETAILS\t\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\
xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB");

    menuBorder();

    ptr = fopen("records.txt","r");

    while(fscanf(ptr,"%ld %d %s %s %d %s %s %ld %lld %s %s %s %s
%d\n",&traverse.AccNo,&traverse.PIN,traverse.fullName,traverse.fatherName,&travers
e.age,traverse.gender,traverse.nationality,&traverse.DOB,&traverse.phoneNo,traverse.
AadhaarNo,traverse.MaritalStatus,traverse.EducationLevel,traverse.occupation,&Accou
nt.accBal) != EOF)

    {

        if(accountNo == traverse.AccNo)

        {

            flag = 1;

            gotoxy(58,22);  printf("GENERATING ACCOUNT DETAILS...");

            Sleep(2000);

            clrscr();

            gotoxy(40,10);
```

```
printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xD
B\xDB\xDB\xDB\t ACCOUNT DETAILS
\t\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB
\xDB");

    menuBorder();

    gotoxy(45,12);  printf("1. ACCOUNT NO: ");          gotoxy(75,12);
printf("%ld",traverse.AccNo);

    gotoxy(45,13);  printf("2. FULL NAME: ");           gotoxy(75,13);
printf("%s",traverse.fullName);

    gotoxy(45,14);  printf("3. FATHER NAME: ");         gotoxy(75,14);
printf("%s",traverse.fatherName);

    gotoxy(45,15);  printf("4. AGE: ");                 gotoxy(75,15);
printf("%d",traverse.age);

    gotoxy(45,16);  printf("5. GENDER (M/F): ");        gotoxy(75,16);
printf("%s",traverse.gender);

    gotoxy(45,17);  printf("6. NATIONALITY: ");         gotoxy(75,17);
printf("%s",traverse.nationality);

    gotoxy(45,18);  printf("7. DATE OF BIRTH(DDMMYYYY): ");
gotoxy(75,18);    printf("%ld",traverse.DOB);

    gotoxy(45,19);  printf("8. PHONE NO: ");            gotoxy(75,19);
printf("%lld",traverse.phoneNo);

    gotoxy(45,20);  printf("9. AADHAAR NUMBER: ");      gotoxy(75,20);
printf("%s",traverse.AadhaarNo);
```

```
            gotoxy(45,21);  printf("10. MARITAL STATUS(Y/N): ");       gotoxy(75,21);
printf("%s",traverse.MaritalStatus);

            gotoxy(45,22);  printf("11. EDUCATION LEVEL: ");          gotoxy(75,22);
printf("%s",traverse.EducationLevel);

            gotoxy(45,23);  printf("12. OCCUPATION: ");               gotoxy(75,23);
printf("%s",traverse.occupation);

            gotoxy(45,24);  printf("13. ACCOUNT BALANCE: ");          gotoxy(75,24);
printf("%d",traverse.accBal);

            break;

        }

    }

    fclose(ptr);

    if(flag != 1)

    {

        gotoxy(60,21);      printf("INVALID ACCOUNT DETAILS!!!");   Sleep(1500);

        gotoxy(61,22);      printf("RETURNING TO MAINMENU!!!");     Sleep(2000);

        return -1;

    }

    if(flagVar == 1)

    {

        gotoxy(43,29);   printf("PRESS ANY KEY TO GO BACK : ");    getch();
```

```
        displayAccountHolderMenu(accountNo);
    }
}
```

- **Function Name : void updateAccDetails(long int)**

```
void updateAccDetails(long int accountNo)
{
    clrscr();
    int opt;
    int x = viewAccDetails(accountNo,0);
    if( x == -1)
    {
        displayAccountHolderMenu(accountNo);
    }
    printf("\033[01;33m");
    gotoxy(42,29);      printf("Enter 0 to SKIP");
    printf("\033[0m");
    gotoxy(45,27);      printf("Enter the option number to Edit: ");     scanf("%d",&opt);
fflush(stdin);
    gotoxy(45,27);      printf("                              ");
    if((opt==1) || (opt==9)){
```

```
        printf("\033[1;31m");

        gotoxy(44,27);      printf("Acc.No and Aadhaar.No can't be changed by Account holder");     Sleep(1500);

        printf("\033[0m");

        updateAccDetails(accountNo);

    }

    else if(opt==13){

        printf("\033[1;31m");

        gotoxy(60,28);      printf("Acc Bal is just to view.");       Sleep(1500);

        printf("\033[0m");

        updateAccDetails(accountNo);

    }

    else if(opt == 0)

    {

        //do nothing

    }

    else{

        ReadAndUpdate(accountNo,opt);

    }

    displayAccountHolderMenu(accountNo);
```

```
}
```

- **Function Name : void viewAccountBalance(long int)**

```
void viewAccountBalance(long int accountNo)

{

   clrscr();

   int choice,flag=0;

   gotoxy(40,10);

printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\t   ACCOUNT
BALANCE\t\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB");

   menuBorder();

   ptr = fopen("records.txt","r");

   while(fscanf(ptr,"%ld %d %s %s %d %s %s %ld %lld %s %s %s %s
%d\n",&traverse.AccNo,&traverse.PIN,traverse.fullName,traverse.fatherName,&travers
e.age,traverse.gender,traverse.nationality,&traverse.DOB,&traverse.phoneNo,traverse.
AadhaarNo,traverse.MaritalStatus,traverse.EducationLevel,traverse.occupation,&traver
se.accBal) != EOF)

   {

      if(accountNo == traverse.AccNo)

      {

         flag = 1;
```

```
gotoxy(45,14);      printf("ACCOUNT NUMBER: %ld",traverse.AccNo);

gotoxy(45,15);      printf("NAME: %s",traverse.fullName);

gotoxy(60,19);      printf("ACCOUNT BALANCE: %d",traverse.accBal);

gotoxy(45,24);      printf("1. TO CHECK TRANSACTION HISTORY");

gotoxy(45,25);      printf("2. TO GO TO MENU");

gotoxy(45,26);      printf("ENTER YOUR CHOICE: ");

fclose(ptr);

scanf("%d",&choice);

if(choice == 1)

{

   viewTransactionHistory(accountNo);

}

else if(choice == 2)

{

   displayAccountHolderMenu(accountNo);

}

else

{

   gotoxy(45,23);      printf("            ");
```

```
            gotoxy(45,24);      printf("                          ");
            gotoxy(45,25);      printf("                ");
            gotoxy(45,26);      printf("                    ");
            printf("\033[1;31m");
            gotoxy(64,25); printf("INVALID ENTRY!!!");      Sleep(1500);
            printf("\033[0m");
            displayAccountHolderMenu(accountNo);
          }
        }
    }
    if(flag != 1)
    {
        fclose(ptr);
        gotoxy(58,25);      printf("UNABLE TO LOAD ACC BALANCE.");
Sleep(1500);
        displayAccountHolderMenu(accountNo);
    }
}
```

- **Function Name : viewTransactionHistory(long int)**

```
void viewTransactionHistory(long int accountNo)
{
   clrscr();
   FILE *fp;
   int choice, flag = 0,y = 14,i=0;
   char *token,str[1000],str1[1000],amount[10],ANo[10];
   const char delim[3]="*";
   gotoxy(40,10);

printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\tTRANSACTION
HISTORY\t\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB");
   gotoxy(40,11);     printf("\xDB");     gotoxy(105,11);     printf("\xDB");
   fp = fopen("transactions.txt","r");
   while(fgets(str,1000,fp) != NULL)
   {
      gotoxy(40,12);     printf("\xDB");     gotoxy(105,12);     printf("\xDB");
      gotoxy(40,13);     printf("\xDB");     gotoxy(105,13);     printf("\xDB");
```

```
gotoxy(40,14);     printf("\xDB");     gotoxy(105,14);     printf("\xDB");
strcpy(str1,str);
itoa(accountNo,ANo,10);
token = strtok(str1,delim);
gotoxy(45,12);     printf("ACCOUNT NO : ");
gotoxy(45,14);     printf("TRANSACTIONS : ");
if(strcmp(token,ANo) == 0)
{
   flag = 1;
   gotoxy(60,12);     printf("%s",token);
   token = strtok(NULL,delim);
   while( token != NULL )
   {
      gotoxy(40,y);     printf("\xDB");     gotoxy(105,y);     printf("\xDB");
      gotoxy(60,y++);      printf("%s",token);
      token = strtok(NULL,delim);
   }
   gotoxy(40,y);     printf("\xDB");     gotoxy(105,y++);     printf("\xDB");
   gotoxy(40,y);     printf("\xDB");     gotoxy(105,y++);     printf("\xDB");
```

```c
        }

    }

    fclose(fp);

    if(flag != 1)

    {

        menuBorder();

        gotoxy(60,12);      printf("%s",ANo);

        gotoxy(60,21);      printf("NO TRANSACTIONS FOUND!!!");     Sleep(500);

        gotoxy(62,24);      printf("RETURNING TO MAINMENU");        Sleep(2000);

        displayAccountHolderMenu(accountNo);

    }

    else

    {

        gotoxy(40,y);      printf("\xDB");     gotoxy(105,y++);     printf("\xDB");

        gotoxy(40,y);      printf("\xDB");     gotoxy(105,y++);     printf("\xDB");

        gotoxy(40,y);
printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xD
B\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB
\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\
xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\x
DB\xDB\xDB\n");
```

```
        gotoxy(43,y-2);   printf("PRESS ANY KEY TO GO BACK : ");    getch();

        displayAccountHolderMenu(accountNo);

    }

}
```

- **FunctionName : void depositAmount(long int)**

```
void depositAmount(long int accountNo)

{

    clrscr();

    int choice,transRes = 0;

    long long int amt=0,verify=0;

    long long int AtmAmount = 0;

    char transType[] = "DEPOSITED - ";

    char trans = 'D';

    gotoxy(40,10);

printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\t   DEPOSIT
AMOUNT\t\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB");

    menuBorder();

    if(ATMServices == TRUE)

    {
```

```
gotoxy(45,13);	printf("ENTER AMOUNT TO DEPOSIT: ");
scanf("%lld",&amt);

gotoxy(45,15);	printf("VERIFY AMOUNT: ");	scanf("%lld",&verify);
fflush(stdin);

if(amt==verify)

{

ptr = fopen("records.txt","r");

ptr1 = fopen("new.txt","w+");

while(fscanf(ptr,"%ld %d %s %s %d %s %s %ld %lld %s %s %s %s
%d\n",&traverse.AccNo,&traverse.PIN,traverse.fullName,traverse.fatherName,&travers
e.age,traverse.gender,traverse.nationality,&traverse.DOB,&traverse.phoneNo,traverse.
AadhaarNo,traverse.MaritalStatus,traverse.EducationLevel,traverse.occupation,&traver
se.accBal) != EOF)

{

if(accountNo != traverse.AccNo)

{

fprintf(ptr1,"%ld %d %s %s %d %s %s %ld %lld %s %s %s %s
%d\n",traverse.AccNo,traverse.PIN,traverse.fullName,traverse.fatherName,traverse.age,
traverse.gender,traverse.nationality,traverse.DOB,traverse.phoneNo,traverse.AadhaarN
o,traverse.MaritalStatus,traverse.EducationLevel,traverse.occupation,traverse.accBal);

}

else

{
```

```
AtmAmount = checkATMAmount();
if(AtmAmount+amt > 1000000)
{
   gotoxy(58,20);     printf("--ATM AMOUNT LIMIT REACHED--");
   gotoxy(60,21);     printf("CAN'T PROCESS DEPOSITS!!");
   transRes = 1;
}
if(transRes == 1)
{
   fprintf(ptr1,"%ld %d %s %s %d %s %s %ld %lld %s %s %s %s
%d\n",traverse.AccNo,traverse.PIN,traverse.fullName,traverse.fatherName,traverse.age,
traverse.gender,traverse.nationality,traverse.DOB,traverse.phoneNo,traverse.AadhaarN
o,traverse.MaritalStatus,traverse.EducationLevel,traverse.occupation,traverse.accBal);
}
else
{
   fprintf(ptr1,"%ld %d %s %s %d %s %s %ld %lld %s %s %s %s
%d\n",traverse.AccNo,traverse.PIN,traverse.fullName,traverse.fatherName,traverse.age,
traverse.gender,traverse.nationality,traverse.DOB,traverse.phoneNo,traverse.AadhaarN
o,traverse.MaritalStatus,traverse.EducationLevel,traverse.occupation,traverse.accBal+a
mt);
   updateATMAmount(trans,verify);
```

```
                addToTransactions(accountNo,transType,amt);

                gotoxy(57,21);      printf("AMOUNT SUCCESFULLY
DEPOSITED!!");

            }

        }

    }

    fclose(ptr);    fclose(ptr1);

    remove("records.txt");

    if(rename("new.txt","records.txt") == 0)

    {

        gotoxy(45,24);      printf("1. CHECK ACCOUNT BALANCE");

        gotoxy(45,25);      printf("2. CHECK TRANSACTION HISTORY");

        gotoxy(45,26);      printf("3. GO BACK");

        gotoxy(62,27);      printf("ENTER YOUR CHOICE: ");
scanf("%d",&choice);    fflush(stdin);

        if(choice == 1){

            viewAccountBalance(accountNo);

        }

        else if(choice == 2){

            viewTransactionHistory(accountNo);
```

```
        }
        else if(choice == 3){
            displayAccountHolderMenu(accountNo);
        }
        else{
            gotoxy(45,27);      printf("                              ");
            printf("\033[1;31m");
            gotoxy(64,27);      printf("INVALID CHOICE!");
            printf("\033[0m");
            Sleep(1500);
            displayAccountHolderMenu(accountNo);
        }
    }
    else{
        gotoxy(45,27);
        printf("ERROR OCCURRED..TRY AGAIN LATER!");     Sleep(1500);
        displayAccountHolderMenu(accountNo);
    }
}
```

```
        else
        {
            printf("\033[1;31m");
            gotoxy(60,25);  printf("VERIFICATION FAILED!!!");   Sleep(500);
            printf("\033[0m");
            gotoxy(62,26);  printf("RETURNING TO MENU");        Sleep(1500);
            Sleep(500);
            displayAccountHolderMenu(accountNo);
        }
    }
    else
    {
        printf("\033[1;31m");
        gotoxy(45,18);
        printf("OH SORRY!! ATM SERVICES ARE TEMPORARILY DISABLED!!");
Sleep(500);
        gotoxy(60,20);      printf("PLEASE TRY AGAIN LATER!!");     Sleep(1500);
        printf("\033[0m");
        gotoxy(62,26);  printf("RETURNING TO MENU");    Sleep(2000);
        displayAccountHolderMenu(accountNo);
```

```
    }
}
```

- **Function Name : void withdrawAmount(long int)**

```
void withdrawAmount(long int accountNo)
{
    int amt=0,verify=0,choice;
    long long int AtmAmount;
    char transType[] = "WITHDRAWN - ";
    char trans = 'W';
    clrscr();
    gotoxy(40,10);

printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\t   WITHDRAW
AMOUNT\t\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB");
    menuBorder();
    if(ATMServices == TRUE)
    {
        gotoxy(45,13);      printf("ENTER AMOUNT TO WITHDRAW: ");
scanf("%d",&amt);
```

```
gotoxy(45,15);      printf("VERIFY AMOUNT: ");      scanf("%d",&verify);

if(amt==verify)

{

   ptr = fopen("records.txt","r");

   ptr1 = fopen("new.txt","w+");

   while(fscanf(ptr,"%ld %d %s %s %d %s %s %ld %lld %s %s %s %s
%d\n",&traverse.AccNo,&traverse.PIN,traverse.fullName,traverse.fatherName,&travers
e.age,traverse.gender,traverse.nationality,&traverse.DOB,&traverse.phoneNo,traverse.
AadhaarNo,traverse.MaritalStatus,traverse.EducationLevel,traverse.occupation,&traver
se.accBal) != EOF)

   {

      if(accountNo != traverse.AccNo)

      {

         fprintf(ptr1,"%ld %d %s %s %d %s %s %ld %lld %s %s %s %s
%d\n",traverse.AccNo,traverse.PIN,traverse.fullName,traverse.fatherName,traverse.age,
traverse.gender,traverse.nationality,traverse.DOB,traverse.phoneNo,traverse.AadhaarN
o,traverse.MaritalStatus,traverse.EducationLevel,traverse.occupation,traverse.accBal);

      }

      else

      {

         AtmAmount = checkATMAmount();

         if(amt>traverse.accBal)
```

```
{
    fprintf(ptr1,"%ld %d %s %s %d %s %s %ld %lld %s %s %s %s
%d\n",traverse.AccNo,traverse.PIN,traverse.fullName,traverse.fatherName,traverse.age,
traverse.gender,traverse.nationality,traverse.DOB,traverse.phoneNo,traverse.AadhaarN
o,traverse.MaritalStatus,traverse.EducationLevel,traverse.occupation,traverse.accBal);
    printf("\033[1;31m");
    gotoxy(63,22);  printf("PROCESS TERMINATED !");    Sleep(500);
    gotoxy(60,23);  printf("ERROR: INSUFFICIENT BALANCE");
Sleep(2000);
    printf("\033[0m");
}
else if(amt>AtmAmount)
{
    fprintf(ptr1,"%ld %d %s %s %d %s %s %ld %lld %s %s %s %s
%d\n",traverse.AccNo,traverse.PIN,traverse.fullName,traverse.fatherName,traverse.age,
traverse.gender,traverse.nationality,traverse.DOB,traverse.phoneNo,traverse.AadhaarN
o,traverse.MaritalStatus,traverse.EducationLevel,traverse.occupation,traverse.accBal);
    printf("\033[1;31m");
    gotoxy(61,22);  printf("PROCESS TERMINATED!!!");    Sleep(500);
    gotoxy(51,23);  printf("ERROR : SORRY...INSUFFICIENT CASH IN
ATM!!");  Sleep(2000);
    printf("\033[0m");
}
```

```
else if(amt>50000)

{

   fprintf(ptr1,"%ld %d %s %s %d %s %s %ld %lld %s %s %s %s
%d\n",traverse.AccNo,traverse.PIN,traverse.fullName,traverse.fatherName,traverse.age,
traverse.gender,traverse.nationality,traverse.DOB,traverse.phoneNo,traverse.AadhaarN
o,traverse.MaritalStatus,traverse.EducationLevel,traverse.occupation,traverse.accBal);

   printf("\033[1;31m");

   gotoxy(64,22);  printf("LIMIT EXCEEDED!!");    Sleep(500);

   gotoxy(52,23);  printf("ERROR : CAN'T WITHDRAW MORE THAN
50,000.");  Sleep(2000);

   printf("\033[0m");

}

else

{

   fprintf(ptr1,"%ld %d %s %s %d %s %s %ld %lld %s %s %s %s
%d\n",traverse.AccNo,traverse.PIN,traverse.fullName,traverse.fatherName,traverse.age,
traverse.gender,traverse.nationality,traverse.DOB,traverse.phoneNo,traverse.AadhaarN
o,traverse.MaritalStatus,traverse.EducationLevel,traverse.occupation,traverse.accBal-
amt);

   updateATMAmount(trans,verify);

   addToTransactions(accountNo,transType,amt);

   gotoxy(59,21);      printf("PLEASE COLLECT THE MONEY!!");

}
```

```
    }
  }
  fclose(ptr);    fclose(ptr1);
  remove("records.txt");
  if(rename("new.txt","records.txt") == 0)
    {
      gotoxy(45,24);      printf("1. CHECK ACCOUNT BALANCE");
      gotoxy(45,25);      printf("2. CHECK TRANSACTION HISTORY");
      gotoxy(45,26);      printf("3. GO BACK");
      gotoxy(62,28);      printf("ENTER YOUR CHOICE: ");
scanf("%d",&choice);
      if(choice == 1){
        viewAccountBalance(accountNo);
      }
      else if(choice == 2){
        viewTransactionHistory(accountNo);
      }
      else if(choice == 3){
        displayAccountHolderMenu(accountNo);
      }
```

```
                else{
                    gotoxy(45,27);      printf("                              ");
                    printf("\033[1;31m");
                    gotoxy(64,27);      printf("INVALID CHOICE!");
                    printf("\033[0m");
                    Sleep(1500);
                    displayAccountHolderMenu(accountNo);
                }
            }
        else
            {
                gotoxy(45,27);      printf("ERROR OCCURRED..TRY AGAIN LATER!");
Sleep(1500);
                displayAccountHolderMenu(accountNo);
            }
    }
    else
    {
        printf("\033[1;31m");
        gotoxy(60,25);  printf("VERIFICATION FAILED!!!");
```

```
            printf("\033[0m");
            gotoxy(62,26);  printf("RETURNING TO MENU");
            Sleep(1000);
            displayAccountHolderMenu(accountNo);
        }
    }
    else
    {
        printf("\033[1;31m");
        gotoxy(45,18);
        printf("OH SORRY!! ATM SERVICES ARE TEMPORARILY DISABLED!!");
Sleep(500);
        gotoxy(60,20);      printf("PLEASE TRY AGAIN LATER!!");     Sleep(1500);
        printf("\033[0m");
        gotoxy(62,26);      printf("RETURNING TO MENU");    Sleep(2000);
        displayAccountHolderMenu(accountNo);
    }
}
```

- **Function Name : void changePin(long int)**

```
void changePin(long int accountNo)

{

    clrscr();

    long int Ano;   int pw,flag=0;

    gotoxy(40,10);


printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xD
B\xDB\xDB\xDB\tVERIFY YOUR
ACCOUNT\t\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xD
B\xDB\xDB\xDB");

    menuBorder();

    gotoxy(40,14);      printf("\xDB\t\xB2\xB2\xB2 ENTER ACCOUNT NUMBER: ");

    gotoxy(40,16);      printf("\xDB\t\xB2\xB2\xB2 ENTER PIN: ");

    gotoxy(75,14);      scanf("%d",&Ano);   gotoxy(75,16);      scanf("%d",&pw);

    ptr = fopen("records.txt","r");

    ptr1 = fopen("new.txt","w+");

    while(fscanf(ptr,"%ld %d %s %s %d %s %s %ld %lld %s %s %s %s
%d\n",&traverse.AccNo,&traverse.PIN,traverse.fullName,traverse.fatherName,&travers
e.age,traverse.gender,traverse.nationality,&traverse.DOB,&traverse.phoneNo,traverse.
AadhaarNo,traverse.MaritalStatus,traverse.EducationLevel,traverse.occupation,&traver
se.accBal) != EOF)

    {
```

```
if(accountNo != traverse.AccNo)
{
   fprintf(ptr1,"%ld %d %s %s %d %s %s %ld %lld %s %s %s %s
%d\n",traverse.AccNo,traverse.PIN,traverse.fullName,traverse.fatherName,traverse.age,
traverse.gender,traverse.nationality,traverse.DOB,traverse.phoneNo,traverse.AadhaarN
o,traverse.MaritalStatus,traverse.EducationLevel,traverse.occupation,traverse.accBal);
}
else
{
   flag=1;
   if(traverse.PIN == pw)
   {
      gotoxy(58,20);     printf("******PASSWORD MATCHED******");
Sleep(1000);
      gotoxy(62,22);     printf("Enter New PIN: ");
      int newPw;         scanf("%d",&newPw);     fflush(stdin);
      fprintf(ptr1,"%ld %d %s %s %d %s %s %ld %lld %s %s %s %s
%d\n",traverse.AccNo,newPw,traverse.fullName,traverse.fatherName,traverse.age,trave
rse.gender,traverse.nationality,traverse.DOB,traverse.phoneNo,traverse.AadhaarNo,trav
erse.MaritalStatus,traverse.EducationLevel,traverse.occupation,traverse.accBal);
      gotoxy(59,25);     printf("PIN CHANGED SUCCESSFULLY!!!");
Sleep(1000);
```

```
            gotoxy(60,27);      printf("REDIRECTING TO LOGIN MENU");
Sleep(1500);

            fclose(ptr);    fclose(ptr1);

            remove("records.txt");    rename("new.txt","records.txt");

            displayWelcomePage();

        }

        else

        {

            gotoxy(57,21);      printf("WARNING : WRONG PASSWORD!!!!!");
Sleep(1000);

            gotoxy(50,22);      printf("ENTER 1 TO TRY AGAIN..0 TO RETURN TO
MENU..");

            int opt;    gotoxy(60,24);   printf("ENTER YOUR CHOICE: ");
scanf("%d",&opt);

            if(opt==1)

            {

               fclose(ptr);    fclose(ptr1);

               remove("records.txt");    rename("new.txt","records.txt");

               changePin(accountNo);

            }

            else
```

```
            {
               fclose(ptr);    fclose(ptr1);
               remove("records.txt");    rename("new.txt","records.txt");
               displayAccountHolderMenu(accountNo);
            }
         }
      }
   }
   if(flag != 1)
   {
      gotoxy(60,21);      printf("INVALID ACCOUNT DETAILS!!!");   Sleep(1500);
      gotoxy(61,22);      printf("RETURNING TO MAINMENU!!!");     Sleep(1500);
displayAccountHolderMenu(accountNo);
   }
}
```

- **Function Name : void closeAccount(long int)**

```
void closeAccount(long int accountNo)
{
   clrscr();
   long int Ano;   int pw,flag=0;
```

```
   gotoxy(40,10);

printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\tVERIFY YOUR
ACCOUNT\t\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB");

   menuBorder();

   gotoxy(40,14);     printf("\xDB\t\xB2\xB2\xB2 ENTER ACCOUNT NUMBER: ");

   gotoxy(40,15);     printf("\xDB\t\xB2\xB2\xB2 ENTER PIN: ");

   gotoxy(75,14);     scanf("%d",&Ano);   gotoxy(75,15);     scanf("%d",&pw);

   ptr = fopen("records.txt","r");

   ptr1 = fopen("new.txt","w+");

   while(fscanf(ptr,"%ld %d %s %s %d %s %s %ld %lld %s %s %s %s
%d\n",&traverse.AccNo,&traverse.PIN,traverse.fullName,traverse.fatherName,&travers
e.age,traverse.gender,traverse.nationality,&traverse.DOB,&traverse.phoneNo,traverse.
AadhaarNo,traverse.MaritalStatus,traverse.EducationLevel,traverse.occupation,&traver
se.accBal) != EOF)

   {

      if(accountNo != traverse.AccNo)

      {

         fprintf(ptr1,"%ld %d %s %s %d %s %s %ld %lld %s %s %s %s
%d\n",traverse.AccNo,traverse.PIN,traverse.fullName,traverse.fatherName,traverse.age,
traverse.gender,traverse.nationality,traverse.DOB,traverse.phoneNo,traverse.AadhaarN
o,traverse.MaritalStatus,traverse.EducationLevel,traverse.occupation,traverse.accBal);
```

```
    }
    else
    {
        flag=1;
        if(traverse.PIN == pw)
        {
            gotoxy(58,21);      printf("******PASSWORD MATCHED******");
Sleep(1000);
            gotoxy(54,24);      printf("!!! ACCOUNT SUCCESSFULLY DELETED !!!");
Sleep(1000);
            gotoxy(60,27);      printf("REDIRECTING TO LOGIN MENU");
Sleep(1500);
            fclose(ptr);    fclose(ptr1);
            remove("records.txt");    rename("new.txt","records.txt");
            displayWelcomePage();
        }
        else
        {
            fprintf(ptr1,"%ld %d %s %s %d %s %s %ld %lld %s %s %s %s
%d\n",traverse.AccNo,traverse.PIN,traverse.fullName,traverse.fatherName,traverse.age,
traverse.gender,traverse.nationality,traverse.DOB,traverse.phoneNo,traverse.AadhaarN
o,traverse.MaritalStatus,traverse.EducationLevel,traverse.occupation,traverse.accBal);
```

```
            gotoxy(57,21);      printf("WARNING : WRONG PASSWORD!!!!!");
Sleep(1000);

            gotoxy(50,22);      printf("ENTER 1 TO TRY AGAIN..0 TO RETURN TO
MENU..");

            int opt;    gotoxy(60,24);   printf("ENTER YOUR CHOICE: ");
scanf("%d",&opt);

            if(opt==1)

            {

               fclose(ptr);    fclose(ptr1);

               remove("records.txt");    rename("new.txt","records.txt");

               closeAccount(accountNo);

            }

            else

            {

               fclose(ptr);    fclose(ptr1);

               remove("records.txt");    rename("new.txt","records.txt");

               displayAccountHolderMenu(accountNo);

            }

         }

      }

   }
```

```
    if(flag != 1)
    {
        fclose(ptr);    fclose(ptr1);
        gotoxy(60,21);      printf("INVALID ACCOUNT DETAILS!!!");   Sleep(1500);
        gotoxy(61,22);      printf("RETURNING TO MAINMENU!!!");     Sleep(1500);
        displayAccountHolderMenu(accountNo);
    }
}
```

3.2.1.3. Employee use case functions

- **Function Name : void loginPageEmployee()**

```
void loginPageEmployee()
{
    clrscr();
    int y = 75,i=0;
    char username[20],pw[20],ch;
    gotoxy(40,10);
printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\t   ADMIN
LOGIN\t\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB");
```

```
menuBorder();
gotoxy(40,14);      printf("\xDB\t\xB2\xB2\xB2 ENTER USERNAME: ");
gotoxy(40,16);      printf("\xDB\t\xB2\xB2\xB2 ENTER PASSWORD: ");
gotoxy(72,14);      gets(username);     gotoxy(72,16);
while(ch != 13)
{
   ch = getch();
   if(ch != 13 && ch!= 8)
   {
      printf("*");
      pw[i] = ch;
      i++;
   }
   else if(ch == 8)
   {
      if(i>0)
      {
         i--;
         pw[i] = '\0';
```

```
            printf("\b \b");
        }
    }
    if(ch == 13)
    {
        break;
    }
  }
  if((strcmp(username,"bankingsystem@miniproject")==0) &&
(strcmp(pw,"vcestudentbank") == 0))
  {
    gotoxy(58,21);    printf("******PASSWORD MATCHED******");
Sleep(500);
    gotoxy(62,23);    printf("-----LOGGING IN-----");        Sleep(1000);
    displayEmployeeMenu();
  }
  else
  {
    gotoxy(51,22);    printf("WARNING : WRONG USERNAME OR
PASSWORD!!!!!"); Sleep(2000);
```

```
        gotoxy(49,23);      printf("ENTER 1 TO TRY AGAIN..0 TO RETURN TO MAIN
MENU..");

        int opt;    gotoxy(60,25);   printf("ENTER YOUR CHOICE: ");  scanf("%d",&opt);

        if(opt==1){

                fflush(stdin);

                loginPageEmployee();

            }

        else{

                displayWelcomePage();

            }

    }

}
```

- **Function Name : void AccountsList()**

```
void AccountsList()

{

    clrscr();

    int y=14,slNo=1;

    gotoxy(40,10);


printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xD
B\xDB\xDB\xDB\xDB\tLIST OF
```

```
ACCOUNTS\t\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\x
DB\xDB\xDB\xDB");

   gotoxy(40,11);   printf("\xDB");    gotoxy(105,11);     printf("\xDB");

   gotoxy(40,12);   printf("\xDB");    gotoxy(42,12);      printf("Sl.No");

   gotoxy(50,12);      printf("--Acc No--");       gotoxy(65,12);      printf("--NAME--");

   gotoxy(85,12);      printf("--Phone No--");     gotoxy(105,12);     printf("\xDB");

   gotoxy(40,13);   printf("\xDB");    gotoxy(105,13);      printf("\xDB");

   ptr = fopen("records.txt","r");

   while(fscanf(ptr,"%ld %d %s %s %d %s %s %ld %lld %s %s %s %s
%d\n",&traverse.AccNo,&traverse.PIN,traverse.fullName,traverse.fatherName,&travers
e.age,traverse.gender,traverse.nationality,&traverse.DOB,&traverse.phoneNo,traverse.
AadhaarNo,traverse.MaritalStatus,traverse.EducationLevel,traverse.occupation,&traver
se.accBal) != EOF)

   {

      gotoxy(40,y);   printf("\xDB");

      gotoxy(44,y);   printf("%d",slNo);

      gotoxy(53,y);   printf("%ld",traverse.AccNo);

      gotoxy(66,y);   printf("%s",traverse.fullName);

      gotoxy(87,y);   printf("%lld",traverse.phoneNo);

      gotoxy(105,y);   printf("\xDB");

      y++;       slNo++;
```

```
    }

    fclose(ptr);

    gotoxy(40,y);     printf("\xDB");     gotoxy(105,y);     printf("\xDB");

    gotoxy(40,y+1);   printf("\xDB");     gotoxy(105,y+1);   printf("\xDB");

    gotoxy(40,y+2);   printf("\xDB");     gotoxy(105,y+2);   printf("\xDB");

    gotoxy(40,y+3);
printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xD
B\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB
\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\
xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\x
DB\xDB\xDB");

    gotoxy(43,y+2);   printf("PRESS ANY KEY TO GO BACK : ");    getch();

    displayEmployeeMenu();

}
```

- **Function Name : void viewAccHolderDetails()**

```
void viewAccHolderDetails()

{

    clrscr();

    long int accountNo;          int opt;

    gotoxy(40,10);


printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xD
```

```
B\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\t   VIEW
DETAILS\t\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\
xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB");

   menuBorder();

   gotoxy(45,12);      printf("Enter Account Number: ");

   scanf("%ld",&accountNo);

   int x = viewAccDetails(accountNo,0);

   if(x == -1){

      displayEmployeeMenu();

   }

   gotoxy(43,29);   printf("PRESS ANY KEY TO GO BACK : ");    getch();

   displayEmployeeMenu();

}
```

- **Function Name : void editAccHolderDetails()**

```
void editAccHolderDetails()

{

   clrscr();

   long int accountNo;

   int opt;

   gotoxy(40,10);
```

```
printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xD
B\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\t   VIEW
DETAILS\t\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\
xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB");

   menuBorder();

   gotoxy(45,12);      printf("Enter Account Number: ");

   scanf("%ld",&accountNo);

   int x = viewAccDetails(accountNo,0);

   if(x == -1)

   {

      displayEmployeeMenu();

   }

   gotoxy(45,26);      printf("Enter the option number to Edit: ");     scanf("%d",&opt);
fflush(stdin);

   Sleep(100);

   if((opt==1) || (opt==13)){

      gotoxy(43,25);      printf("                                                ");

      gotoxy(43,26);      printf("                                                ");

      gotoxy(55,27);      printf("Acc.No and Acc.Bal can't be changed");

   }
```

```
    else{
        gotoxy(43,25);      printf("                                        ");
        gotoxy(45,26);      printf("                                        ");
        ReadAndUpdate(accountNo,opt);
    }
    gotoxy(43,29);   printf("PRESS ANY KEY TO GO BACK : ");    getch();
    displayEmployeeMenu();
}
```

- **Function Name : void closeAccHolderAccount()**

```
void closeAccHolderAccount()
{
    clrscr();
    int flag=0;
    long int accountNo;
    gotoxy(40,10);

printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\t DELETE
ACCOUNT\t\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB");

    menuBorder();
```

```
    gotoxy(45,12);      printf("Enter Account Number: ");

    scanf("%ld",&accountNo);

    ptr = fopen("records.txt","r");

    ptr1 = fopen("new.txt","w+");

    while(fscanf(ptr,"%ld %d %s %s %d %s %s %ld %lld %s %s %s %s
%d\n",&traverse.AccNo,&traverse.PIN,traverse.fullName,traverse.fatherName,&travers
e.age,traverse.gender,traverse.nationality,&traverse.DOB,&traverse.phoneNo,traverse.
AadhaarNo,traverse.MaritalStatus,traverse.EducationLevel,traverse.occupation,&traver
se.accBal) != EOF)

    {

        if(accountNo != traverse.AccNo)

        {

            fprintf(ptr1,"%ld %d %s %s %d %s %s %ld %lld %s %s %s %s
%d\n",traverse.AccNo,traverse.PIN,traverse.fullName,traverse.fatherName,traverse.age,
traverse.gender,traverse.nationality,traverse.DOB,traverse.phoneNo,traverse.AadhaarN
o,traverse.MaritalStatus,traverse.EducationLevel,traverse.occupation,traverse.accBal);

        }

        else

        {

            flag=1;

            gotoxy(55,24);      printf("!!! ACCOUNT SUCCESSFULLY DELETED !!!");
Sleep(500);

            gotoxy(55,25);      printf("RETURNING TO MENU!!!");     Sleep(500);
```

```
        }
    }
    fclose(ptr);    fclose(ptr1);
    remove("records.txt");    rename("new.txt","records.txt");
    if(flag != 1)
    {
        gotoxy(60,21);      printf("INVALID ACCOUNT DETAILS!!!");   Sleep(1000);
        gotoxy(61,22);      printf("RETURNING TO MAINMENU!!!");     Sleep(1500);
displayEmployeeMenu();
    }
    displayEmployeeMenu();
}
```

- **Function Name : void addMoneyToATM()**

```
void addMoneyToATM()
{
    clrscr();
    int choice;
    long long int AtmAmount;
    char trans = 'D';
    gotoxy(40,10);
```

```
printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xD
B\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB
\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\
xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\x
DB\xDB\xDB");

   menuBorder();

   gotoxy(58,20);      printf("GENERATING ATM AMOUNT STATUS");

   for(int i=0;i<5;i++)

   {

      Sleep(500);  printf(".");

   }



   clrscr();

   gotoxy(40,10);


printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xD
B\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB
\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\
xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\x
DB\xDB\xDB");

   menuBorder();

   AtmAmount = checkATMAmount();
```

```
    gotoxy(60,15);      printf("REMAINING ATM AMOUNT : %lld",AtmAmount);

    gotoxy(53,28);      printf("Enter 1 to PROCEED and 0 to GO BACK : ");
scanf("%d",&choice);     fflush(stdin);

    if(choice == 1)

    {

        gotoxy(43,28);       printf("                                                  ");

        gotoxy(60,17);      printf("ENTER AMOUNT TO ADD : ");

        long long int add;        scanf("%lld",&add);

        if(AtmAmount+add > 1000000)

        {

            gotoxy(63,25);      printf("PROCESS FAILED...");        Sleep(500);

            gotoxy(54,26);      printf("ATM CAN'T HAVE MORE THAN 10 LAKHS...");
Sleep(2000);

        }

        else

        {

            updateATMAmount(trans,add);

            gotoxy(50,27);      printf("MONEY ADDED SUCCESSFULLY...RETURNING
TO MENU!!!");      Sleep(2000);

        }

        displayEmployeeMenu();
```

```
    }
    else if(choice == 0)
    {
        displayEmployeeMenu();
    }
    else
    {
        gotoxy(50,19);      printf("INVALID ENTRY...RETURNING TO MENU!!!");
Sleep(1500);
        displayEmployeeMenu();
    }
}
```

- **Function Name : void disableATMServices()**

```
void disableATMServices()
{
    clrscr();
    int i=0;
    char username[20],pw[20],option[5],ch;
    gotoxy(40,10);
```

```
printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\t  VERIFY LOGIN\t\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB");

   menuBorder();

   gotoxy(40,14);      printf("\xDB\t\xB2\xB2\xB2 ENTER USERNAME : ");

   gotoxy(40,16);      printf("\xDB\t\xB2\xB2\xB2 ENTER PASSWORD : ");

   gotoxy(75,14);      gets(username);   gotoxy(75,16);


   while(ch != 13)

   {

      ch = getch();

      if(ch != 13 && ch!= 8)

      {

         printf("*");

         pw[i] = ch;

         i++;

      }

      else if(ch == 8)

      {
```

```
            if(i>0)
            {
               i--;
               pw[i] = '\0';
               printf("\b \b");
            }
         }
         if(ch == 13)
         {
            break;
         }
      }
      if((strcmp(username,"bankingsystem@miniproject")==0) &&
(strcmp(pw,"vcestudentbank") == 0))
      {
         gotoxy(65,25);      printf("LOGIN VERIFIED");   Sleep(1000);
         gotoxy(65,25);      printf("                ");
         if(ATMServices == TRUE)
         {
            printf("\033[1;31m");
```

```
        gotoxy(82,29);     printf("NOTE : CASE SENSITIVE");

        printf("\033[0m;");

        gotoxy(50,22);      printf("DISABLE ATM SERVICES (YES/NO) : ");
scanf("%s",option);

        if(strcmp(option,"YES") == 0)

        {

           gotoxy(50,24);      printf("PRESS 'YES' TO CONFIRM : ");
scanf("%s",option);

           gotoxy(82,29);     printf("                    ");

           if(strcmp(option,"YES") == 0)

           {

              ATMServices = FALSE;

              gotoxy(47,26);      printf("ATM SERVICES ARE DISABLED UNTIL
YOU ENABLE BACK!!!");    Sleep(2000);

              gotoxy(43,29);   printf("PRESS ANY KEY TO GO BACK : ");    getch();

           }

        }

     }

     else

     {
```

```
        gotoxy(50,22);      printf("ENABLE ATM SERVICES (YES/NO) : ");
scanf("%s",option);

        if(strcmp(option,"YES") == 0)

        {

           gotoxy(50,24);      printf("PRESS 'YES' TO CONFIRM : ");
scanf("%s",option);

           if(strcmp(option,"YES") == 0)

           {

              ATMServices = TRUE;

              gotoxy(51,26);      printf("ATM SERVICES ARE ENABLED BACK FOR
USERS!!");

              gotoxy(43,29);   printf("PRESS ANY KEY TO GO BACK : ");    getch();

           }

        }

     }

     displayEmployeeMenu();

  }

  else

  {

     gotoxy(55,22);      printf("WARNING : WRONG USERNAME OR
PASSWORD!!!!!"); Sleep(2000);
```

```
        gotoxy(47,23);      printf("ENTER 1 TO TRY AGAIN..0 TO RETURN TO MAIN
MENU..");

        int opt;    gotoxy(60,24);   printf("ENTER YOUR CHOICE: ");  scanf("%d",&opt);

        if(opt==1){

              fflush(stdin);

              disableATMServices();

           }

        else{

              fflush(stdin);

              displayEmployeeMenu();

           }

    }

}
```

3.2.1.4. Miscellaneous function

- **Function Name : int main()**

```
int main()

{

   char welMsge[] = "WELCOME TO VCE STUDENT BANK.";

   char tqMsge[] = "THANK YOU";
```

```
    srand(time(NULL));
    loadingPage(welMsge);
    displayWelcomePage();
    Sleep(1000);
    clrscr();
    loadingPage(tqMsge);
}
```

- **Function Name : void ReadAndUpdate(long int, int)**

```
void ReadAndUpdate(long int accountNo, int opt)
{
    long long int updatedData;
    char updatedInfo[20];
    gotoxy(43,25);      printf("                                        ");
    gotoxy(45,26);      printf("ENTER UPDATED INFO: ");
    ptr = fopen("records.txt","r");
    ptr1 = fopen("new.txt","w+");
    while(fscanf(ptr,"%ld %d %s %s %d %s %s %ld %lld %s %s %s %s
%d\n",&traverse.AccNo,&traverse.PIN,traverse.fullName,traverse.fatherName,&travers
e.age,traverse.gender,traverse.nationality,&traverse.DOB,&traverse.phoneNo,traverse.
```

```
AadhaarNo,traverse.MaritalStatus,traverse.EducationLevel,traverse.occupation,&traverse.accBal) != EOF)
   {
      if(accountNo != traverse.AccNo)
      {
         fprintf(ptr1,"%ld %d %s %s %d %s %s %ld %lld %s %s %s %s %d\n",traverse.AccNo,traverse.PIN,traverse.fullName,traverse.fatherName,traverse.age,traverse.gender,traverse.nationality,traverse.DOB,traverse.phoneNo,traverse.AadhaarNo,traverse.MaritalStatus,traverse.EducationLevel,traverse.occupation,traverse.accBal);
      }
      else
      {
         switch(opt)
         {
            case 2:
               gotoxy(65,26);   gets(updatedInfo);
               fprintf(ptr1,"%ld %d %s %s %d %s %s %ld %lld %s %s %s %s %d\n",traverse.AccNo,traverse.PIN,updatedInfo,traverse.fatherName,traverse.age,traverse.gender,traverse.nationality,traverse.DOB,traverse.phoneNo,traverse.AadhaarNo,traverse.MaritalStatus,traverse.EducationLevel,traverse.occupation,traverse.accBal);
               break;
```

```
            case 3:

                gotoxy(65,26);   gets(updatedInfo);

                fprintf(ptr1,"%ld %d %s %s %d %s %s %ld %lld %s %s %s %s
%d\n",traverse.AccNo,traverse.PIN,traverse.fullName,updatedInfo,traverse.age,traverse
.gender,traverse.nationality,traverse.DOB,traverse.phoneNo,traverse.AadhaarNo,travers
e.MaritalStatus,traverse.EducationLevel,traverse.occupation,traverse.accBal);

                break;



            case 4:

                gotoxy(65,26);   scanf("%lld",&updatedData);

                fprintf(ptr1,"%ld %d %s %s %lld %s %s %ld %lld %s %s %s %s
%d\n",traverse.AccNo,traverse.PIN,traverse.fullName,traverse.fatherName,updatedData
,traverse.gender,traverse.nationality,traverse.DOB,traverse.phoneNo,traverse.AadhaarN
o,traverse.MaritalStatus,traverse.EducationLevel,traverse.occupation,traverse.accBal);

                break;

              case 5:

                gotoxy(65,26);   gets(updatedInfo);

                fprintf(ptr1,"%ld %d %s %s %d %s %s %ld %lld %s %s %s %s
%d\n",traverse.AccNo,traverse.PIN,traverse.fullName,traverse.fatherName,traverse.age,
updatedInfo,traverse.nationality,traverse.DOB,traverse.phoneNo,traverse.AadhaarNo,tr
averse.MaritalStatus,traverse.EducationLevel,traverse.occupation,traverse.accBal);
break;
```

```
            case 6:

                gotoxy(65,26);   gets(updatedInfo);

                fprintf(ptr1,"%ld %d %s %s %d %s %s %ld %lld %s %s %s %s
%d\n",traverse.AccNo,traverse.PIN,traverse.fullName,traverse.fatherName,traverse.age,
traverse.gender,updatedInfo,traverse.DOB,traverse.phoneNo,traverse.AadhaarNo,traver
se.MaritalStatus,traverse.EducationLevel,traverse.occupation,traverse.accBal);

                break;

            case 7:

                gotoxy(65,26);   scanf("%lld",&updatedData);

                fprintf(ptr1,"%ld %d %s %s %d %s %s %ld %lld %s %s %s %s
%d\n",traverse.AccNo,traverse.PIN,traverse.fullName,traverse.fatherName,traverse.age,
traverse.gender,traverse.nationality,updatedData,traverse.phoneNo,traverse.AadhaarNo,
traverse.MaritalStatus,traverse.EducationLevel,traverse.occupation,traverse.accBal);

                break;

            case 8:

                gotoxy(65,26);   scanf("%lld",&updatedData);

                fprintf(ptr1,"%ld %d %s %s %d %s %s %ld %lld %s %s %s %s
%d\n",traverse.AccNo,traverse.PIN,traverse.fullName,traverse.fatherName,traverse.age,
traverse.gender,traverse.nationality,traverse.DOB,updatedData,traverse.AadhaarNo,trav
erse.MaritalStatus,traverse.EducationLevel,traverse.occupation,traverse.accBal);

                break;

            case 9:

                gotoxy(65,26);  gets(updatedInfo);
```

```
            fprintf(ptr1,"%ld %d %s %s %d %s %s %ld %lld %s %s %s %s
%d\n",traverse.AccNo,traverse.PIN,traverse.fullName,traverse.fatherName,traverse.age,
traverse.gender,traverse.nationality,traverse.DOB,traverse.phoneNo,updatedData,traver
se.MaritalStatus,traverse.EducationLevel,traverse.occupation,traverse.accBal);

            break;

        case 10:

            gotoxy(65,26);   gets(updatedInfo);

            fprintf(ptr1,"%ld %d %s %s %d %s %s %ld %lld %s %s %s %s
%d\n",traverse.AccNo,traverse.PIN,traverse.fullName,traverse.fatherName,traverse.age,
traverse.gender,traverse.nationality,traverse.DOB,traverse.phoneNo,traverse.AadhaarN
o,updatedInfo,traverse.EducationLevel,traverse.occupation,traverse.accBal);

            break;

        case 11:

            gotoxy(65,26);   gets(updatedInfo);

            fprintf(ptr1,"%ld %d %s %s %d %s %s %ld %lld %s %s %s %s
%d\n",traverse.AccNo,traverse.PIN,traverse.fullName,traverse.fatherName,traverse.age,
traverse.gender,traverse.nationality,traverse.DOB,traverse.phoneNo,traverse.AadhaarN
o,traverse.MaritalStatus,updatedInfo,traverse.occupation,traverse.accBal);

            break;

        case 12:

            gotoxy(65,26);   gets(updatedInfo);

            fprintf(ptr1,"%ld %d %s %s %d %s %s %ld %lld %s %s %s %s
%d\n",traverse.AccNo,traverse.PIN,traverse.fullName,traverse.fatherName,traverse.age,
```

```
traverse.gender,traverse.nationality,traverse.DOB,traverse.phoneNo,traverse.AadhaarNo,traverse.MaritalStatus,traverse.EducationLevel,updatedInfo,traverse.accBal);
                break;
            }
        }
    }
    fclose(ptr);    fclose(ptr1);
    gotoxy(43,26);      printf("                                                    ");
    remove("records.txt");
    if(rename("new.txt","records.txt") == 0)
    {
        gotoxy(65,27);      printf("CHANGES SAVED!!");      Sleep(2000);
    }
    else
    {
        gotoxy(55,27);      printf("ERROR OCCURRED..TRY AGAIN LATER!");
Sleep(2000);
    }
}
```

- **Function Name : void updateATMAmount(char, long long int)**

```
void updateATMAmount(char transType, long long int amount)
{
   long long int amt;
   FILE *fp = fopen("AtmBalance.txt","a+");
   while(fscanf(fp,"%lld",&amt) != EOF)
   {
      //do nothing;
   }
   FILE *fp1 = fopen("newBal.txt","w+");
   if(transType == 'D')
   {
      amt = amt + amount;
      fprintf(fp1,"%lld",amt);
   }
   else if(transType == 'W')
   {
      amt = amt - amount;
      fprintf(fp1,"%lld",amt);
```

```
    }
    else
    {
       //
    }
    fclose(fp); fclose(fp1);
    remove("AtmBalance.txt");
    rename("newBal.txt","AtmBalance.txt");
}
```

- **Function Name : void addToTransactions(long int, char, long long int)**

```
void addToTransactions(long int accountNo, char transType[],long long int amt)
{
    time_t t;
    time(&t);
    FILE *fp,*fp1;
    char *token,str[1000],str1[1000],str2[1000],amount[60],ANo[10],time[50];
    fp = fopen("transactions.txt","a+");
    fp1 = fopen("newtrans.txt","a+");
    int i=0,flag=0;
```

```
const char delim[3]="*";
for(int i = 0;i<strlen(ctime(&t))-1;i++)
{
   time[i] = ctime(&t)[i];
}
while(fgets(str,1000,fp) != NULL)
{
   strcpy(str1,str);
   token = strtok(str1,delim);
   itoa(amt,amount,10);
   itoa(accountNo,ANo,10);
   if(strcmp(token,ANo) == 0)
   {
      flag = 1;
      for(i=0;i<strlen(str)-1;i++)
      {
         str2[i]=str[i];
      }
      str2[i] = '\0';
```

```
        strcat(amount,"(");
        strcat(amount,time);
        strcat(amount,")");
        strcat(strcat(str2,delim),strcat(transType,amount));
        fputs(str2,fp1);
        fprintf(fp1,"\n");
    }
    else
    {
        fputs(str,fp1);
    }
}
if(flag != 1)
{
    itoa(accountNo,ANo,10);
    itoa(amt,amount,10);
    strcat(amount,"(");
    strcat(amount,time);
    strcat(amount,")");
```

```
        strcat(transType,amount);
        fprintf(fp1,"%s%s%s\n",ANo,delim,transType);
    }
    fclose(fp);    fclose(fp1);
    remove("transactions.txt");
    if(rename("newtrans.txt","transactions.txt") == 0)
    {
        gotoxy(61,20);  printf("TRANSACTION SUCCESSFUL");
    }
}
```

- **Function Name : int generateAccNo()**

```
int generateAccNo()
{
    long int r = rand();
    ptr1 = fopen("records.txt","r");
    while(fscanf(ptr1,"%ld %d %s %s %d %s %s %ld %lld %s %s %s %s
%d\n",&traverse.AccNo,&traverse.PIN,traverse.fullName,traverse.fatherName,&travers
e.age,traverse.gender,traverse.nationality,&traverse.DOB,&traverse.phoneNo,traverse.
AadhaarNo,traverse.MaritalStatus,traverse.EducationLevel,traverse.occupation,&traver
se.accBal) != EOF)
    {
```

```
        if(traverse.AccNo == r)
        {
            r = generateAccNo();
        }
    }
    fclose(ptr1);
    return r;
}
```

- **Function Name : long long int checkATMAmount()**

```
long long int checkATMAmount()
{
    FILE *fp = fopen("AtmBalance.txt","r");
    long long int AtmBal;
    while(fscanf(fp,"%lld",&AtmBal) != EOF)
    {
        //do nothing;
    }
    fclose(fp);
    return AtmBal;        }
```

- **Function Name : void gotoxy(int, int)**

```
void gotoxy(int x,int y)

{

   coord.X=x + 15;

   coord.Y=y;

   SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE),coord);

}
```

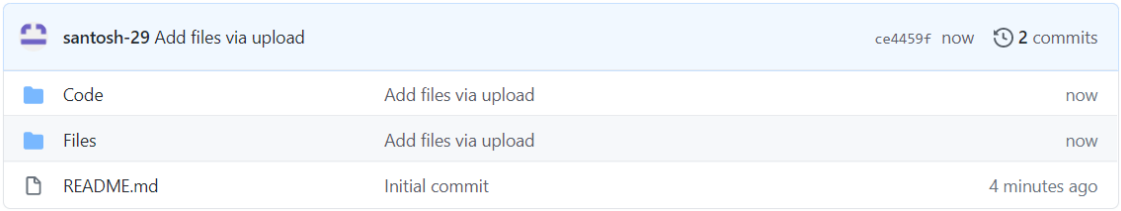
3.2.2. GitHub Folder/ Structure

We have segregated the files of our project into folders namely - Code and Files. Code folder contains the main C file and the Files folder contains the data files (.txt format) which we used in our project. Our repository also contains a README file which has a brief description of our project. We have also uploaded report of our project on our github repository.

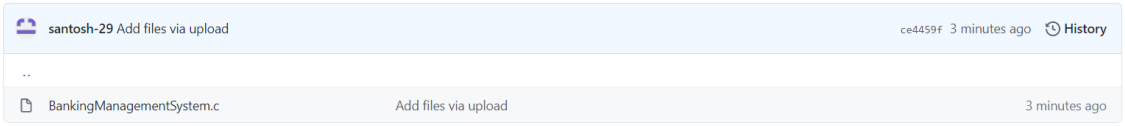
GitHub Repository Link:

https://github.com/santosh-29/BankingManagementSystem

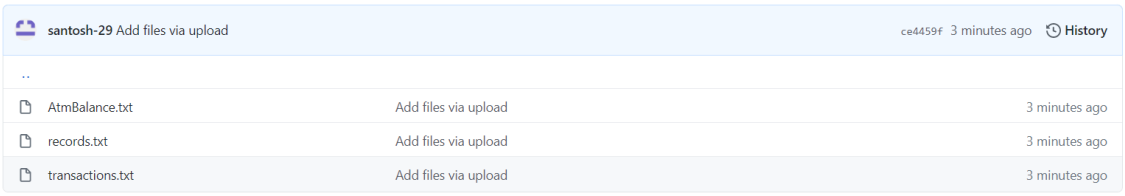
https://github.com/akshatshandilya/BankingManagementSystem



- Code Folder :



- Files Folder :



3.3. TESTING

3.3.1. Account Holder Test Cases

- LOGIN

Test case ID: TC01		Use case ID: **UC01**
Test case title: Login (Account Holder)		
Test case description: User has to enter account number and PIN		
Test steps	**Expected result**	**Actual result**
System prompts the user to enter account no and PIN. If incorrect account number or PIN is entered.	An error message should be displayed on the screen saying incorrect username or incorrect password.	An error message is displayed on the screen.

Test case ID: TC02		Use case ID: **UC01**
Test case title: Login (Account Holder)		
Test case description: User has to enter account number and PIN		
Test steps	**Expected result**	**Actual result**
System prompts the user to enter account number and PIN. If correct details are entered.	Will show successful login message and opens User Menu.	Displayed login successful message and User Menu is opened.

- CREATE NEW ACCOUNT

Test case ID: TC03		Use case ID: **UC02**
Test case title: Create New Account		
Test case description: User has to fill Details		
Test steps	**Expected result**	**Actual result**
System asks to enter list of details. If incorrect details are entered	An error message saying incorrect format is displayed on the screen and user is forced to enter correct format.	Error message is displayed on the screen and user has to re-enter.

Test case ID: TC04		Use case ID: **UC02**
Test case title: Create New Account		
Test case description: User has to fill Details		
Test steps	**Expected result**	**Actual result**
System asks the user to fill the details. If all details are entered correctly	A unique account number and PIN should be generated and displayed on screen.	A unique account number and PIN is generated and displayed on screen

- VIEW ACCOUNT DETAILS

Test case ID: TC05		Use case ID: **UC03**
Test case title: View Account Details		
Test case description: User can view his account details		
Test steps	**Expected result**	**Actual result**
When the user selects this option, the system displays all the details of the user on the screen	The Account details of the user will be displayed on the screen.	The Account details of the user are displayed on the screen.

- UPDATE ACCOUNT DETAILS

Test case ID: TC06		Use case ID: **UC04**
Test case title: Update Account Details		
Test case description: User Details are displayed in the screen and user can select an option to edit.		
Test steps	**Expected result**	**Actual result**
The system prompts the user to enter an option to edit any field from the list of details. If invalid option is given.	Depending on input, should display an error message showing "particular field can't be edited" or "entered choice is invalid".	Error message is displayed on the screen showing the field can't be edited.

Test case ID: TC07		Use case ID: **UC04**
Test case title: Update Account Details		
Test case description: User Details are displayed in the screen and user can select an option to edit.		
Test steps	**Expected result**	**Actual result**
The system prompts the user to enter an option to edit any field from the list of details. If input is valid option.	Should take input from the user and update the details and should display message.	Input is taken from the user and the details are modified. Message is displayed on the screen.

- ACCOUNT BALANCE

Test case ID: TC08		Use case ID: **UC05**
Test case title: Account balance		
Test case description: User can view his/her account balance		
Test steps	**Expected result**	**Actual result**
User has to select this option from the Menu.	Account Balance should be displayed on the screen.	Account Balance is displayed on the screen.

Test case ID: TC09		Use case ID: **UC05**
Test case title: Account Balance		
Test case description: Account Balance after certain amount is deposited		
Test steps	**Expected result**	**Actual result**
User has to select this option from the Menu.	Updated account balance should be displayed when compared with previous test case.	Updated account balance is displayed when compared with previous test case.

- DEPOSIT

Test case ID: TC10		Use case ID: **UC06**
Test case title: Deposit		
Test case description: When user deposits money into his account after ATM limit is reached.		
Test steps	**Expected result**	**Actual result**
User has to select this option and enter the amount to deposit.	Should show an error message “ATM limit reached. Withdrawals can’t be processed.”	An error message “ATM limit reached. Withdrawals can’t be processed” Is displayed.

Test case ID: TC11		Use case ID: **UC06**
Test case title: Deposit		
Test case description: User deposits money into his account		
Test steps	**Expected result**	**Actual result**
User has to select this option and enter the amount to deposit.	Amount should be deposited and message is displayed on the screen and displays some options to perform.	Amount is deposited and message is displayed on the screen. Options are displayed on the screen.

- WITHDRAW

Test case ID: TC12		Use case ID: **UC07**
Test case title: Withdraw		
Test case description: If the account balance is less than requested withdrawal amount.		
Test steps	**Expected result**	**Actual result**
User has to select this option and enter the amount to withdraw.	Should display an error message “Insufficient account balance”.	Displayed an error message “Insufficient account balance”.

Test case ID: TC13		Use case ID: **UC07**
Test case title: Withdraw		
Test case description: If withdraw amount is more than amount in ATM		
Test steps	**Expected result**	**Actual result**
User has to select this option and enter the amount to withdraw.	An error message showing “ATM out of cash” should be displayed on the screen.	An error message showing “ATM out of cash” is displayed on the screen.

Test case ID: TC14		Use case ID: **UC07**
Test case title: Withdraw		
Test case description: If withdrawal amount is more than limit.		
Test steps	**Expected result**	**Actual result**
User has to select this option and enter the amount to withdraw.	An error message showing “Can’t withdraw more than 50,000” should be displayed on the screen.	An error message showing “Can’t withdraw more than 50,000” is displayed on the screen.

Test case ID: TC15		Use case ID: **UC07**
Test case title: Withdraw		
Test case description: If requested withdrawal amount is valid.		
Test steps	**Expected result**	**Actual result**
User has to select this option and enter the amount to withdraw.	Amount should be withdrawn and message is displayed on the screen and displays some options to perform.	Amount is withdrawn and message is displayed on the screen and displayed options to perform.

- TRANSACTION HISTORY

Test case ID: TC16		Use case ID: **UC08**
Test case title: Transaction History		
Test case description: User can view his transaction history (Before performing TC15)		
Test steps	**Expected result**	**Actual result**
User has to select this option from the Menu.	Transaction History should be displayed on the screen along with transaction type and time of transaction.	Transaction History is displayed on the screen along with transaction type and time of transaction.

Test case ID: TC17		Use case ID: **UC08**
Test case title: Transaction History		
Test case description: User can view his transaction history (After performing TC15)		
Test steps	**Expected result**	**Actual result**
User has to select this option from the Menu.	Updated Transaction History should be displayed on the screen along with transaction type and time of transaction.	Updated Transaction History is displayed on the screen along with transaction type and time of transaction.

- CHANGE PIN

Test case ID: TC18		Use case ID: **UC09**
Test case title: Change PIN		
Test case description: User can change his/her account PIN.		
Test steps	**Expected result**	**Actual result**
User has to select this option from the menu and verify account credentials. If credentials are entered wrong.	An error message should be displayed on the screen showing "Incorrect Account number or PIN".	Error message is displayed on the screen showing "Incorrect Account number or PIN".

Test case ID: TC19		Use case ID: **UC09**
Test case title: Change PIN		
Test case description: User can change his/her account PIN.		
Test steps	**Expected result**	**Actual result**
User has to select this option from the menu and verify account credentials. If credentials are entered correctly then the system asks for new PIN.	Should display an message that PIN is updated and takes to main menu.	Message showing "PIN is updated" is displayed and user menu is displayed.

- Close Account

Test case ID: TC20		Use case ID: **UC10**
Test case title: Close Account		
Test case description: User can close his bank account after account verification.		
Test steps	**Expected result**	**Actual result**
The system prompts the user to enter account number and PIN. User enters incorrect acc no or PIN.	An error message showing of incorrect account number or PIN should be displayed on the screen.	Error message of incorrect account number or PIN is displayed on the screen.

Test case ID: TC21		Use case ID: **UC10**
Test case title: Close Account		
Test case description: User can close his bank account after account verification.		
Test steps	**Expected result**	**Actual result**
The system prompts the user to enter account number and PIN. If details are entered correctly.	A message showing "Account is deleted" should be displayed on the screen and main menu should be displayed.	A message showing "Account is deleted" is displayed on the screen and main menu is displayed.

3.3.2. Employee Test Cases

- Login

Test case ID: TC22		Use case ID: **UC11**
Test case title: Employee Login		
Test case description: User has to enter username and password		
Test steps	**Expected result**	**Actual result**
The system prompts the user to enter username and password. If password entered is not same as the hard-coded one.	An error message should be displayed on the screen saying incorrect username or incorrect password.	An error message showing "Incorrect username or password" is displayed on the screen.

Test case ID: TC23		Use case ID: **UC11**
Test case title: Employee Login		
Test case description: User has to enter username and password		
Test steps	**Expected result**	**Actual result**
The system prompts the user to enter username and password. If correct credentials are entered.	Will show successful login message and opens Employee Menu.	Displayed successful login message and opened Employee Menu.

- View Account Holders list

Test case ID: TC24		Use case ID: **UC12**
Test case title: Account Holders List		
Test case description: Employee can view the list of Account holders in the bank.		
Test steps	**Expected result**	**Actual result**
The user has to select this option in the Main Menu.	Should display a list of account holders along with their phone numbers and account numbers.	Displayed a list of account holders along with their phone numbers and account numbers.

Test case ID: TC25		Use case ID: **UC12**
Test case title: Account Holders List		
Test case description: Employee can view the list of Account holders in the bank (Adding another account after TC24).		
Test steps	**Expected result**	**Actual result**
The user has to select this option in the Main Menu.	Should display an updated list of account holders along with their phone numbers and account numbers.	Displayed an updated list of account holders along with their phone numbers and account numbers.

- View Account Holder info

Test case ID: TC26		Use case ID: **UC13**
Test case title: View Account holder info		
Test case description: Employee can view the details of a specific account holder		
Test steps	**Expected result**	**Actual result**
The system prompts the user to enter the required account number. If invalid account number is entered.	An error message showing "Invalid account details" should be displayed on the screen.	An error message showing "Invalid account details" is displayed on the screen.

Test case ID: TC27		Use case ID: **UC13**
Test case title: View Account holder info		
Test case description: Employee can view the details of a specific account holder		
Test steps	**Expected result**	**Actual result**
The system prompts the user to enter the required account number. If account number is entered.	The account holder details should be displayed on the screen.	The details of the account holder are displayed on the screen.

- Edit account holder details

Test case ID: TC28		Use case ID: **UC14**
Test case title: Edit Account holder details		
Test case description: Employee can edit details of an account holder.		
Test steps	**Expected result**	**Actual result**
System takes input of account number and displays current details. Prompts to enter an option to edit. If invalid option (Ex: - account no) is given.	An error message saying "This field can't be edited" should be displayed on the screen.	An error message saying "This field can't be edited" is displayed on the screen.

Test case ID: TC29		Use case ID: **UC14**
Test case title: Edit Account holder details		
Test case description: Employee can edit details of an account holder.		
Test steps	**Expected result**	**Actual result**
System takes input of account number and displays current details. Prompts to enter an option to edit. If a valid option (some details even account holder can't edit) is given.	Should take input from the user and update the details and should display message.	Updated input is taken from the user and details are updated. Message is displayed on the screen.

- Delete account of an account holder

Test case ID: TC30		Use case ID: **UC15**
Test case title: Delete Account holder account		
Test case description: Employee can delete account of an account holder.		
Test steps	**Expected result**	**Actual result**
The system prompts the user to enter the required account number. If invalid account number is entered.	An error message showing "Invalid account details" should be displayed on the screen.	An error message showing "Invalid account details" is displayed on the screen.

Test case ID: TC31		Use case ID: **UC15**
Test case title: Delete Account holder account		
Test case description: Employee can delete account of an account holder.		
Test steps	**Expected result**	**Actual result**
The system prompts the user to enter the required account number. If account number is entered correctly.	A message showing "Account successfully deleted" should be displayed on the screen.	A message showing "Account successfully deleted" is displayed on the screen.

- Add money in ATM

Test case ID: TC32		Use case ID: **UC16**
Test case title: Add money in ATM		
Test case description: Employee can add money in ATM when ATM runs out of cash.		
Test steps	**Expected result**	**Actual result**
User has to select this option in the menu. Displays current ATM amount and asks amount to add. If a huge amount is entered.	An error message saying "ATM can't have more than 10 lakhs" should be displayed on the screen.	An error message saying "ATM can't have more than 10 lakhs" is displayed on the screen.

Test case ID: TC33		Use case ID: **UC16**
Test case title: Add money in ATM		
Test case description: Employee can add money in ATM when ATM runs out of cash.		
Test steps	**Expected result**	**Actual result**
User has to select this option in the menu. Displays current ATM amount and asks amount to add. If valid amount is entered.	Amount should be added into the ATM and should display a message on the screen.	Amount is added into the ATM and message is displayed on the screen.

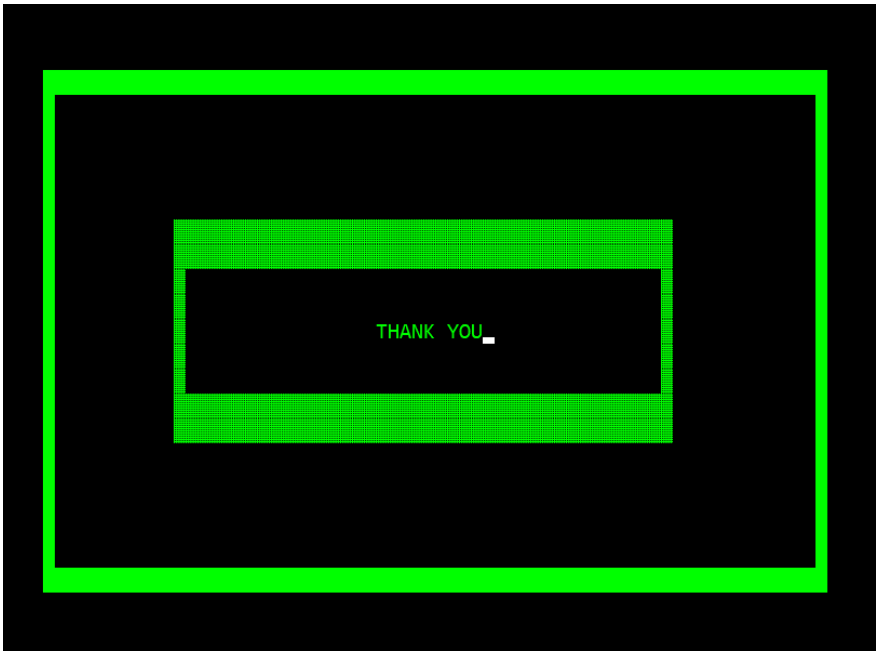
- Disable ATM Services

Test case ID: TC32		Use case ID: **UC16**
Test case title: Disable ATM services		
Test case description: Employee can temporarily enable/disable ATM services (deposits, withdrawals).		
Test steps	**Expected result**	**Actual result**
User has to verify login and confirm whether to disable ATM services.	A message should be displayed on the screen showing "ATM services are disabled for users"	A message is displayed on the screen showing "ATM services are disabled for users".

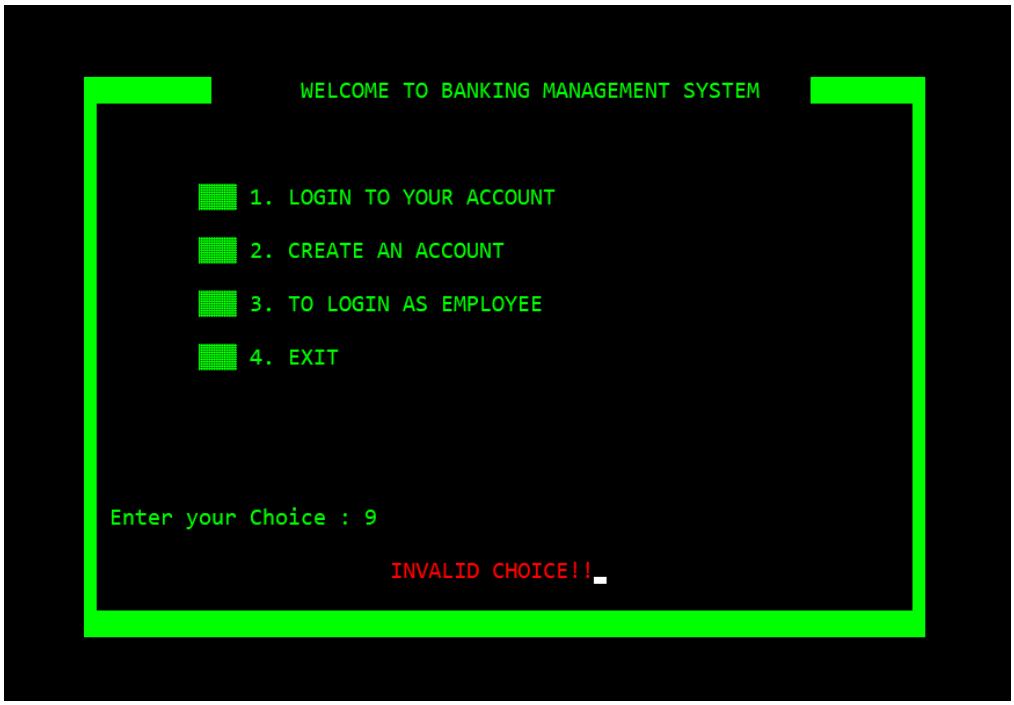
Test case ID: TC33		Use case ID: **UC16**
Test case title: Disable ATM services		
Test case description: Employee can temporarily enable/disable ATM services (deposits, withdrawals). When the costumer tries to use ATM features.		
Test steps	**Expected result**	**Actual result**
Employee has to disable ATM services and an account holder should login and try to deposit/withdraw money	A message should be displayed on the screen showing “ATM services are temporarily disabled”.	A message is displayed on the screen showing “ATM services are temporarily disabled”.

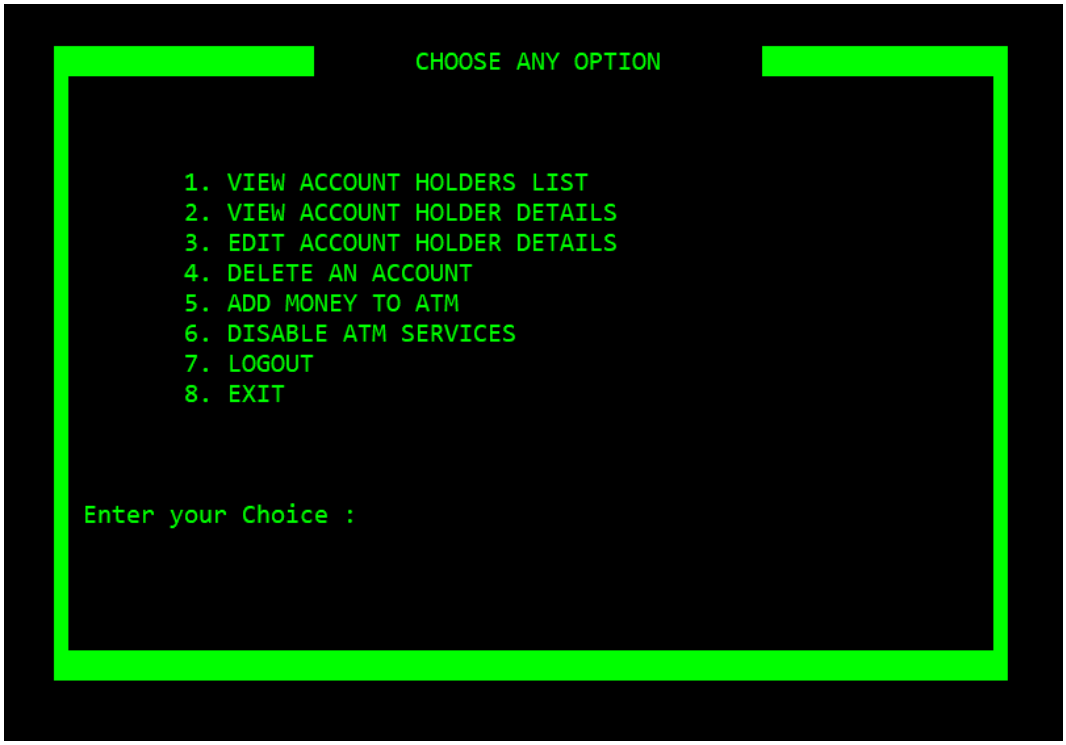
4. RESULTS

#1 - Welcome and Thank you pages



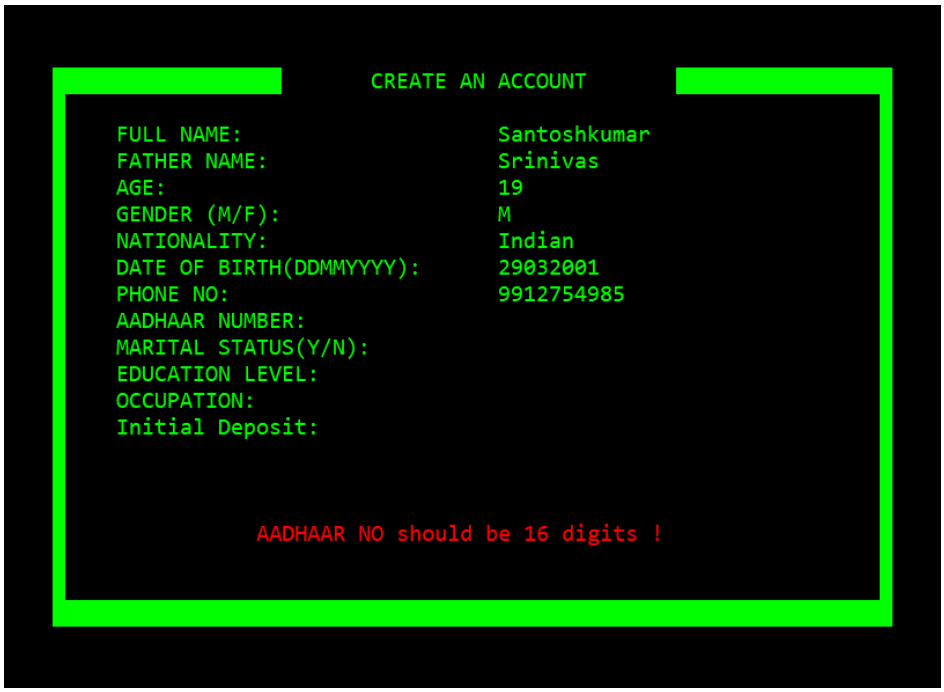
#2 - Main menu

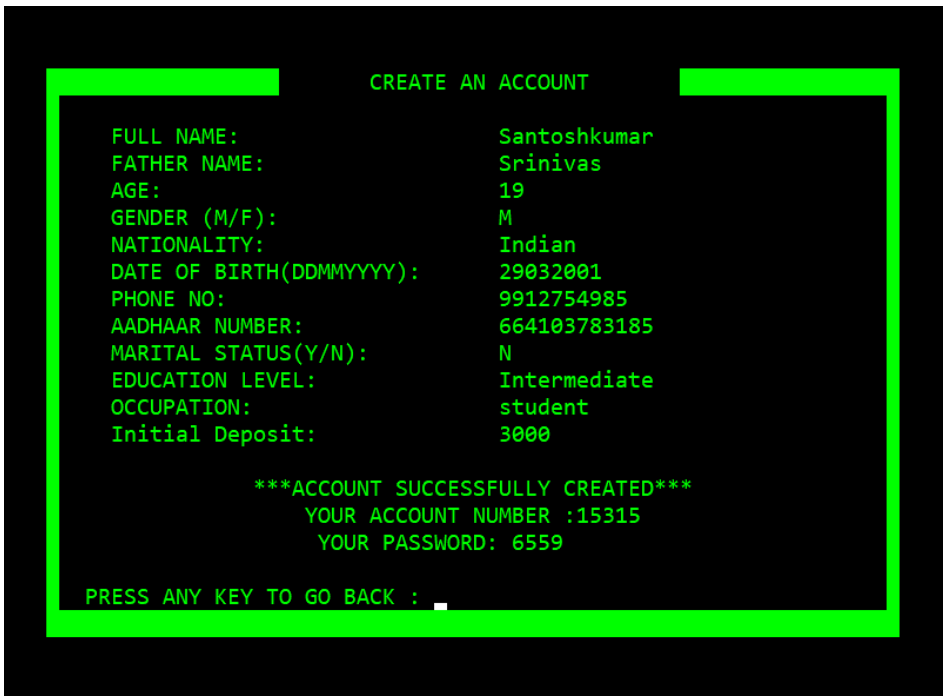

WELCOME TO BANKING MANAGEMENT SYSTEM
1. LOGIN TO YOUR ACCOUNT
2. CREATE AN ACCOUNT
3. TO LOGIN AS EMPLOYEE
4. EXIT
Enter your Choice : 9
INVALID CHOICE!!


CHOOSE ANY OPTION
1. VIEW ACCOUNT HOLDERS LIST
2. VIEW ACCOUNT HOLDER DETAILS
3. EDIT ACCOUNT HOLDER DETAILS
4. DELETE AN ACCOUNT
5. ADD MONEY TO ATM
6. DISABLE ATM SERVICES
7. LOGOUT
8. EXIT
Enter your Choice :

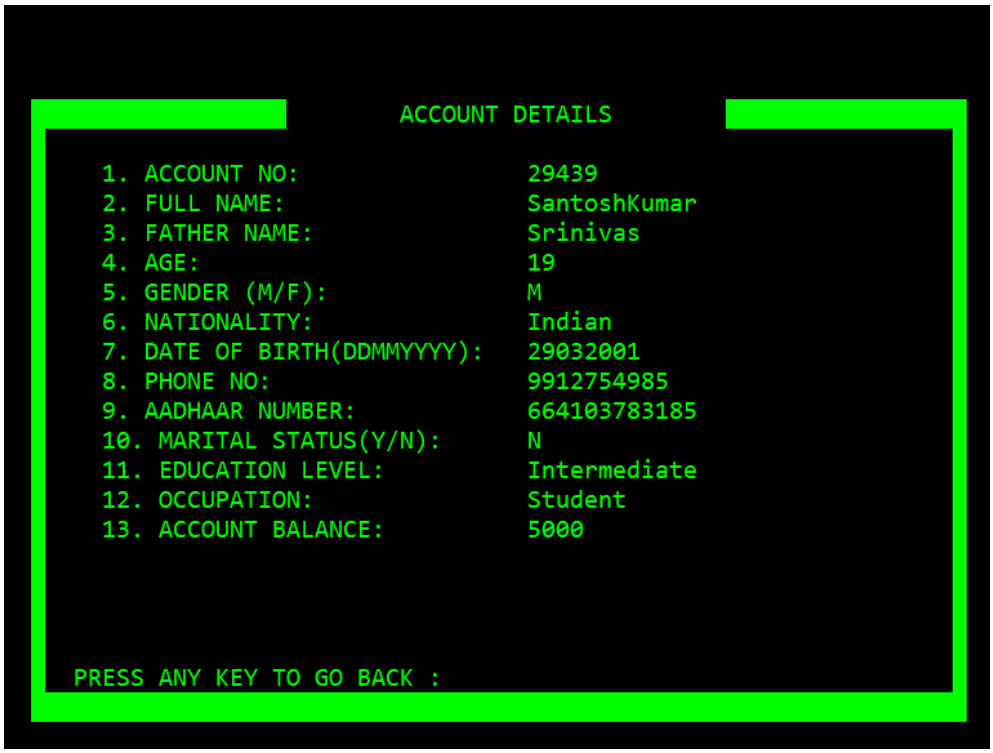
- **ACCOUNT HOLDER USE CASES**

#3 - Create new account

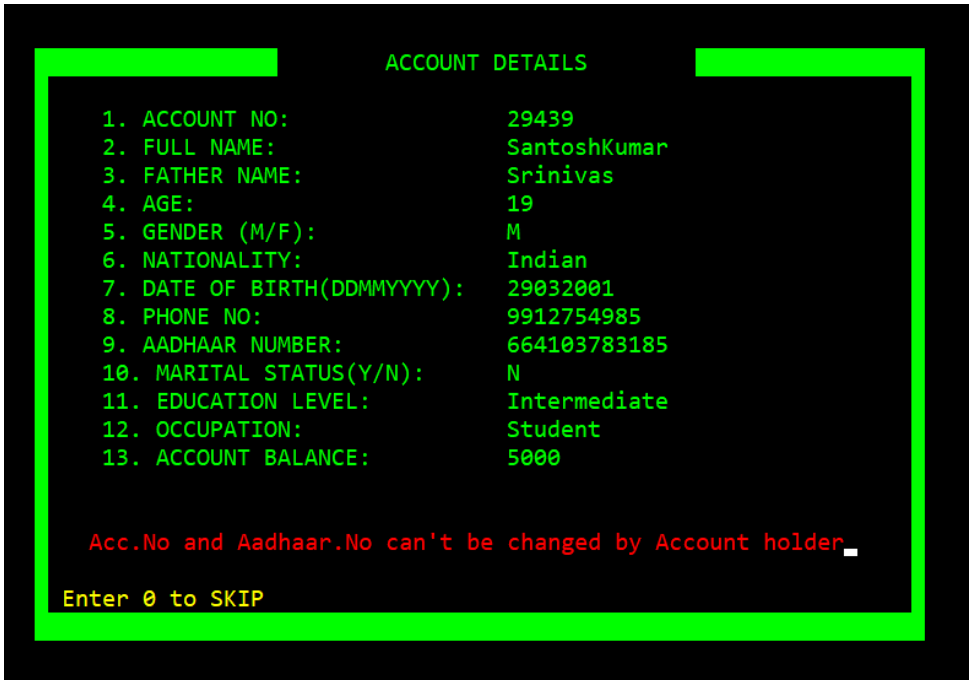


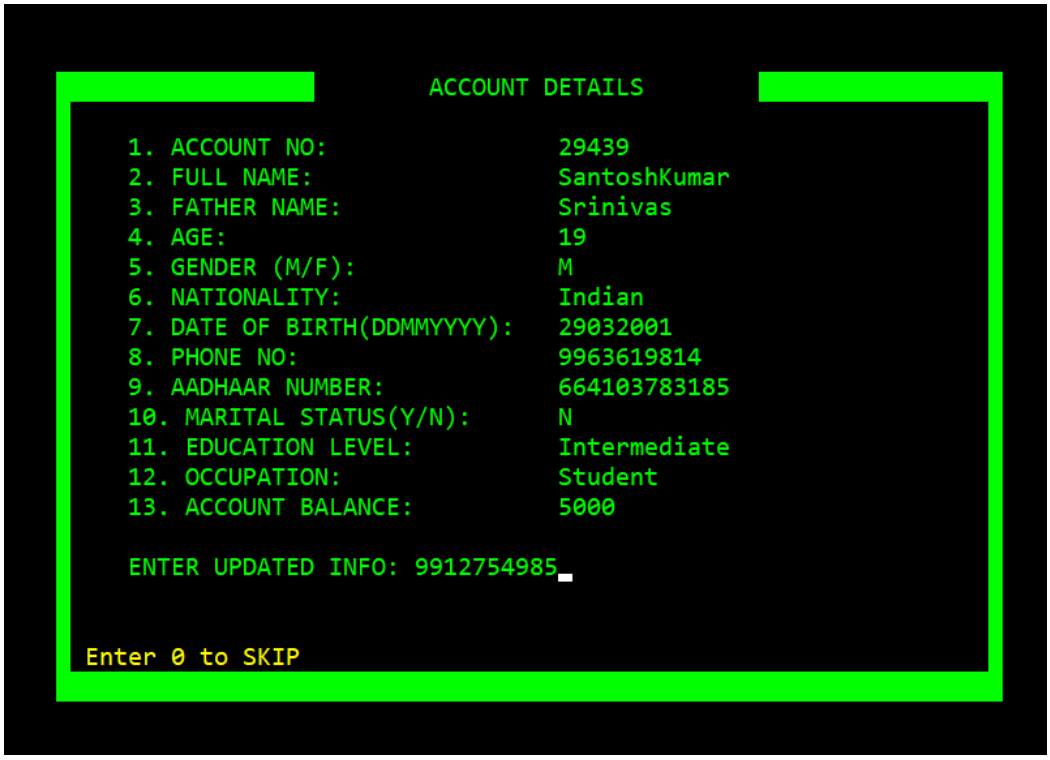


#4 - View Account Details



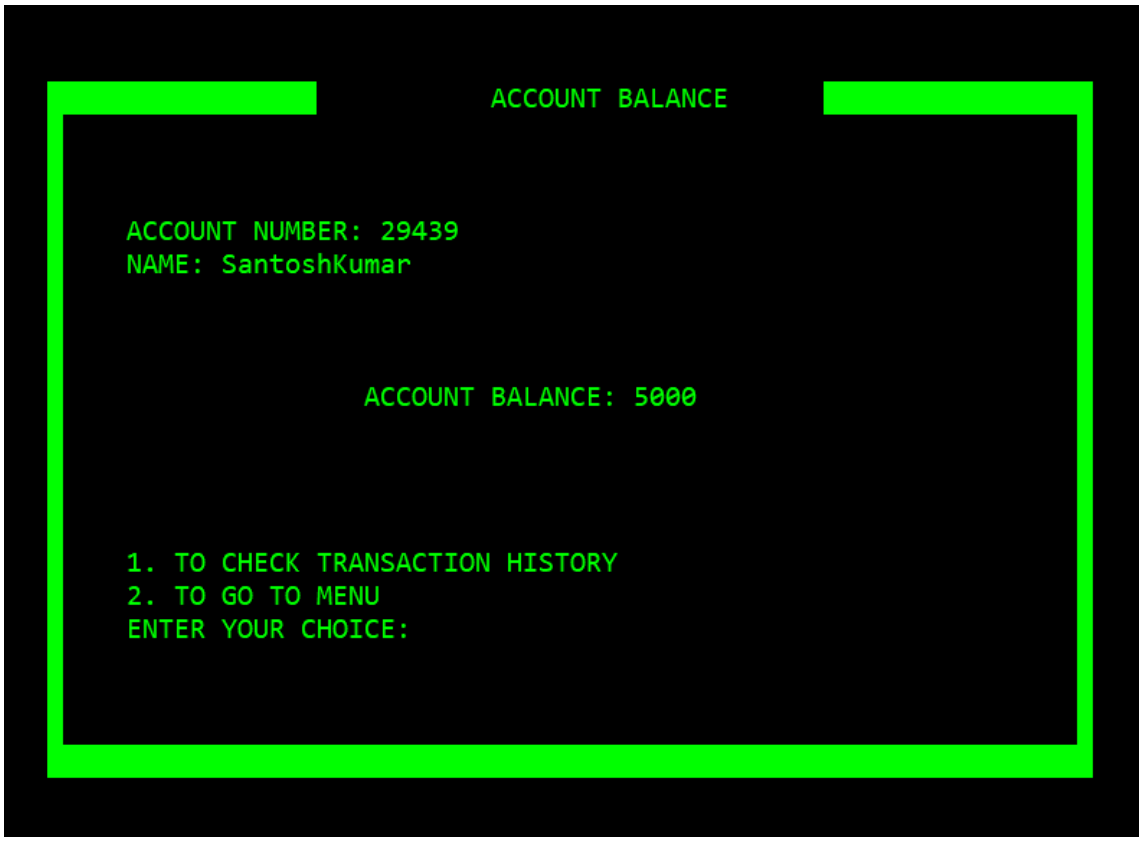
#5 - Update Account Details

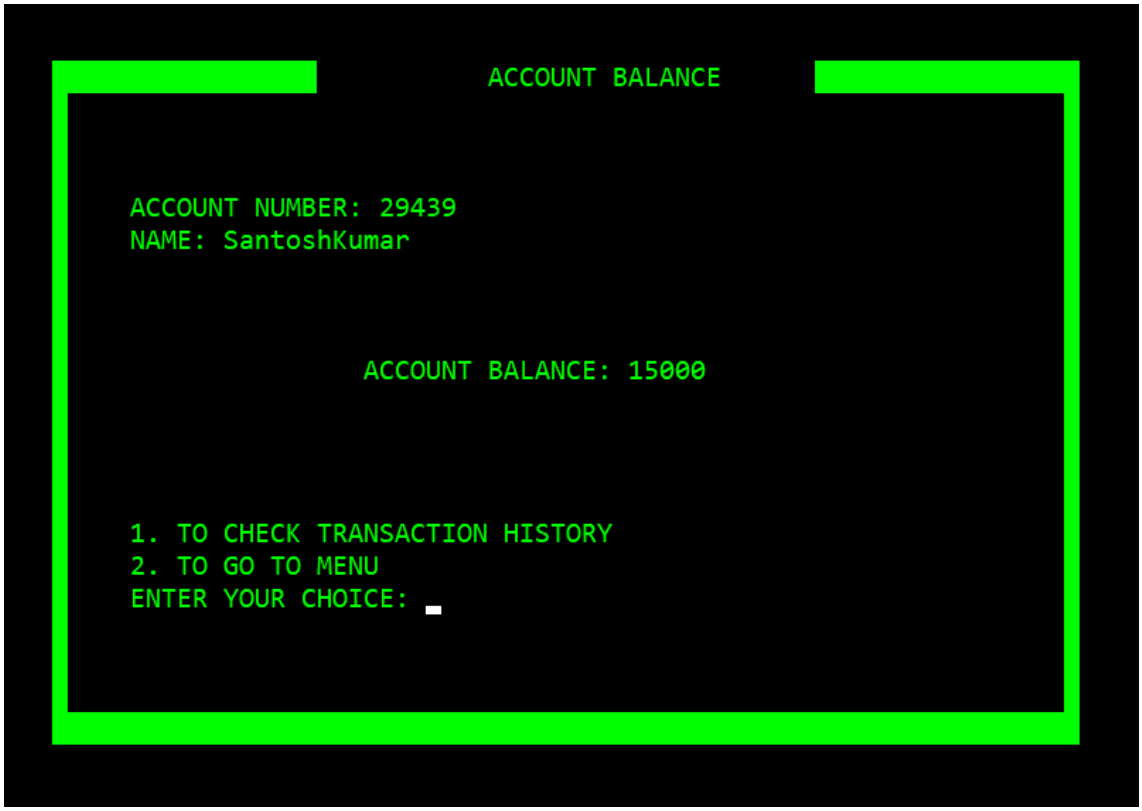



ACCOUNT DETAILS
1. ACCOUNT NO: 29439
2. FULL NAME: SantoshKumar
3. FATHER NAME: Srinivas
4. AGE: 19
5. GENDER (M/F): M
6. NATIONALITY: Indian
7. DATE OF BIRTH(DDMMYYYY): 29032001
8. PHONE NO: 9963619814
9. AADHAAR NUMBER: 664103783185
10. MARITAL STATUS(Y/N): N
11. EDUCATION LEVEL: Intermediate
12. OCCUPATION: Student
13. ACCOUNT BALANCE: 5000
ENTER UPDATED INFO: 9912754985
Enter 0 to SKIP

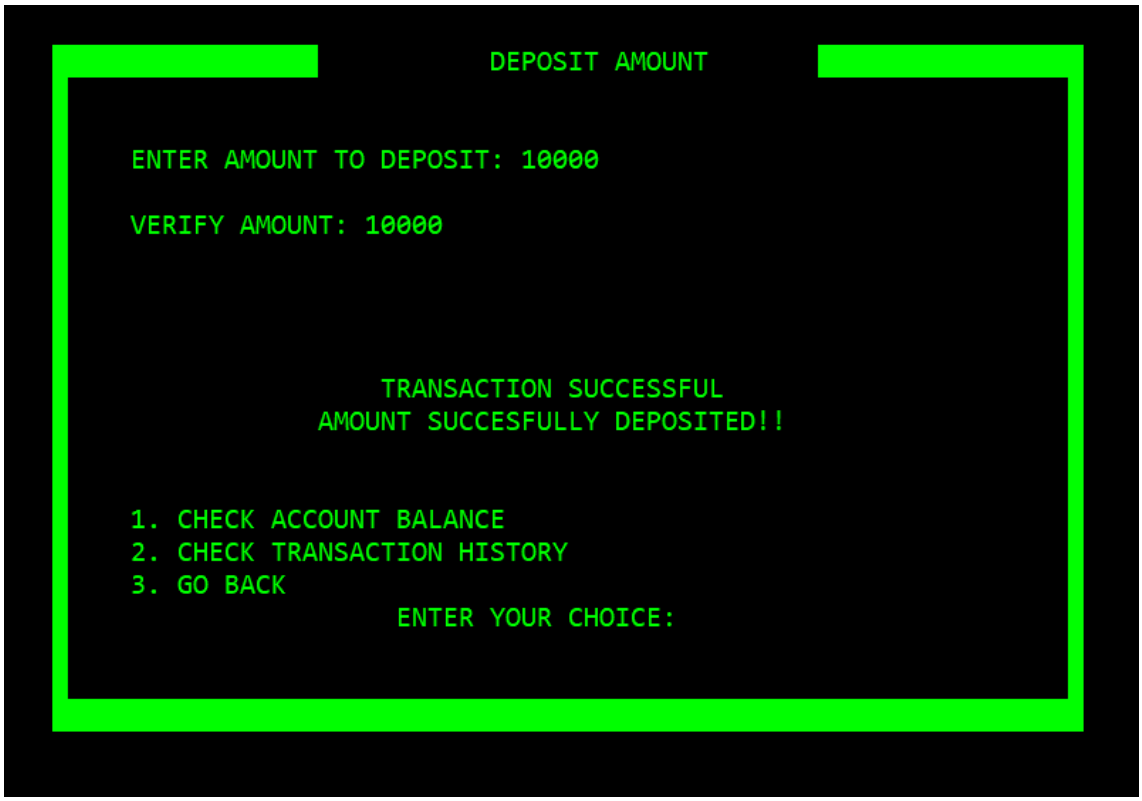

ACCOUNT DETAILS
1. ACCOUNT NO: 29439
2. FULL NAME: SantoshKumar
3. FATHER NAME: Srinivas
4. AGE: 19
5. GENDER (M/F): M
6. NATIONALITY: Indian
7. DATE OF BIRTH(DDMMYYYY): 29032001
8. PHONE NO: 9963619814
9. AADHAAR NUMBER: 664103783185
10. MARITAL STATUS(Y/N): N
11. EDUCATION LEVEL: Intermediate
12. OCCUPATION: Student
13. ACCOUNT BALANCE: 5000
CHANGES SAVED!!
Enter 0 to SKIP

#6 - View Account Balance



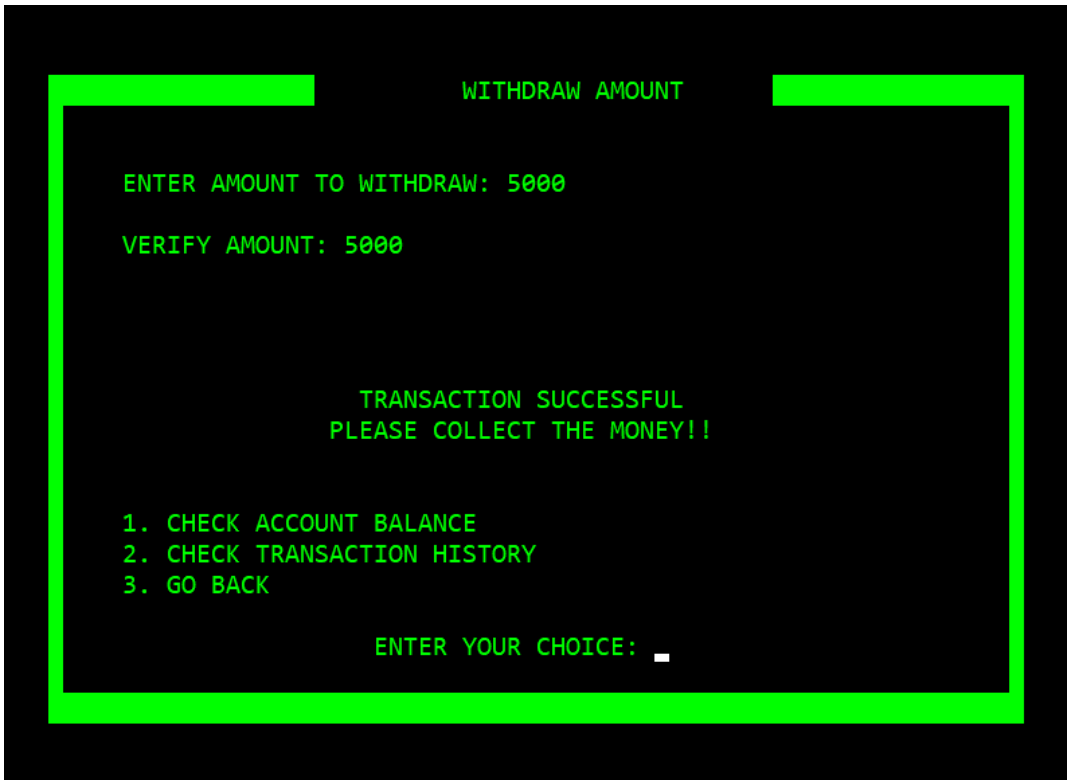


#7 - Deposit


DEPOSIT AMOUNT
ENTER AMOUNT TO DEPOSIT: 10000
VERIFY AMOUNT: 10000
TRANSACTION SUCCESSFUL
AMOUNT SUCCESFULLY DEPOSITED!!
1. CHECK ACCOUNT BALANCE
2. CHECK TRANSACTION HISTORY
3. GO BACK
ENTER YOUR CHOICE:

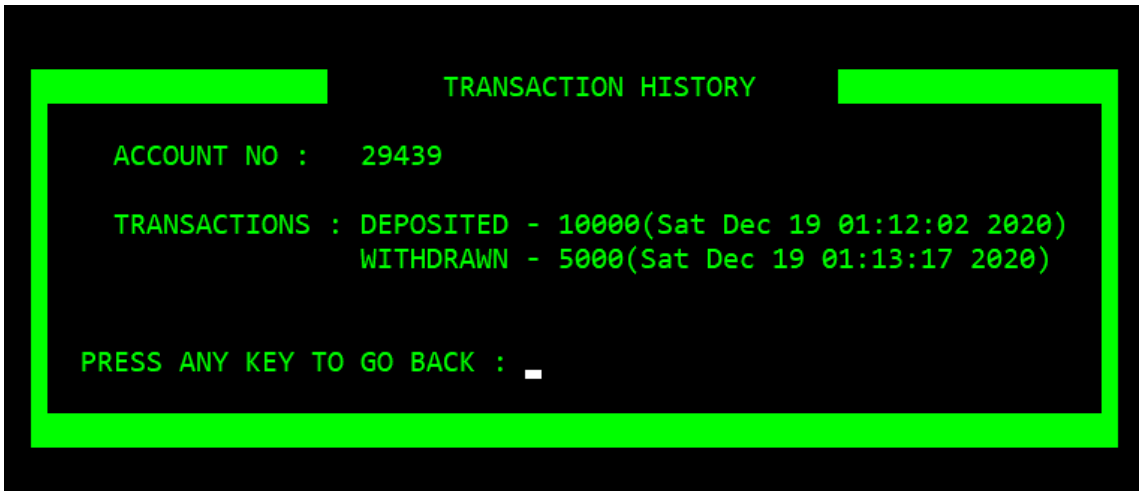

DEPOSIT AMOUNT
ENTER AMOUNT TO DEPOSIT: 2000
VERIFY AMOUNT: 20123
VERIFICATION FAILED!!!
RETURNING TO MENU

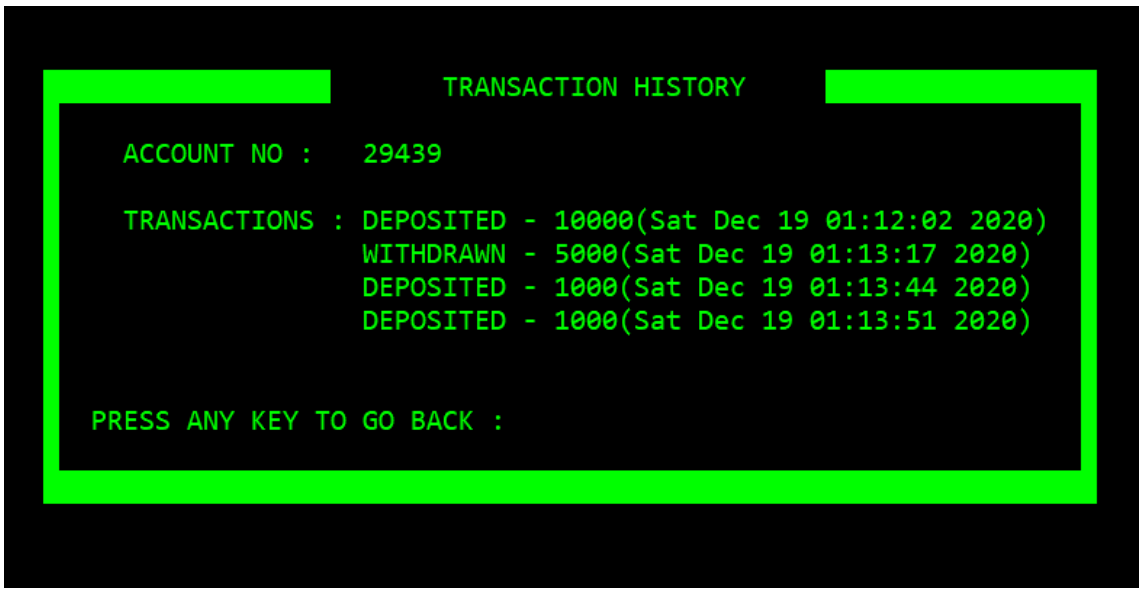
#8 - Withdraw


WITHDRAW AMOUNT
ENTER AMOUNT TO WITHDRAW: 5000
VERIFY AMOUNT: 5000
TRANSACTION SUCCESSFUL
PLEASE COLLECT THE MONEY!!
1. CHECK ACCOUNT BALANCE
2. CHECK TRANSACTION HISTORY
3. GO BACK
ENTER YOUR CHOICE:


WITHDRAW AMOUNT
ENTER AMOUNT TO WITHDRAW: 5000
VERIFY AMOUNT: 4000
VERIFICATION FAILED!!!
RETURNING TO MENU

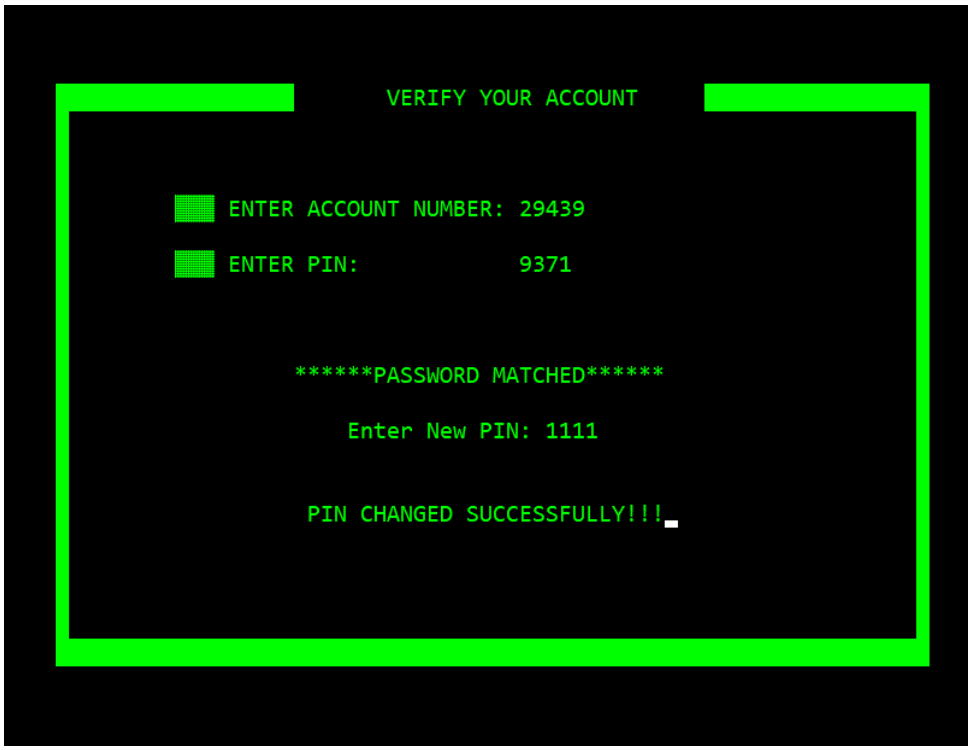
#7 - Transaction History



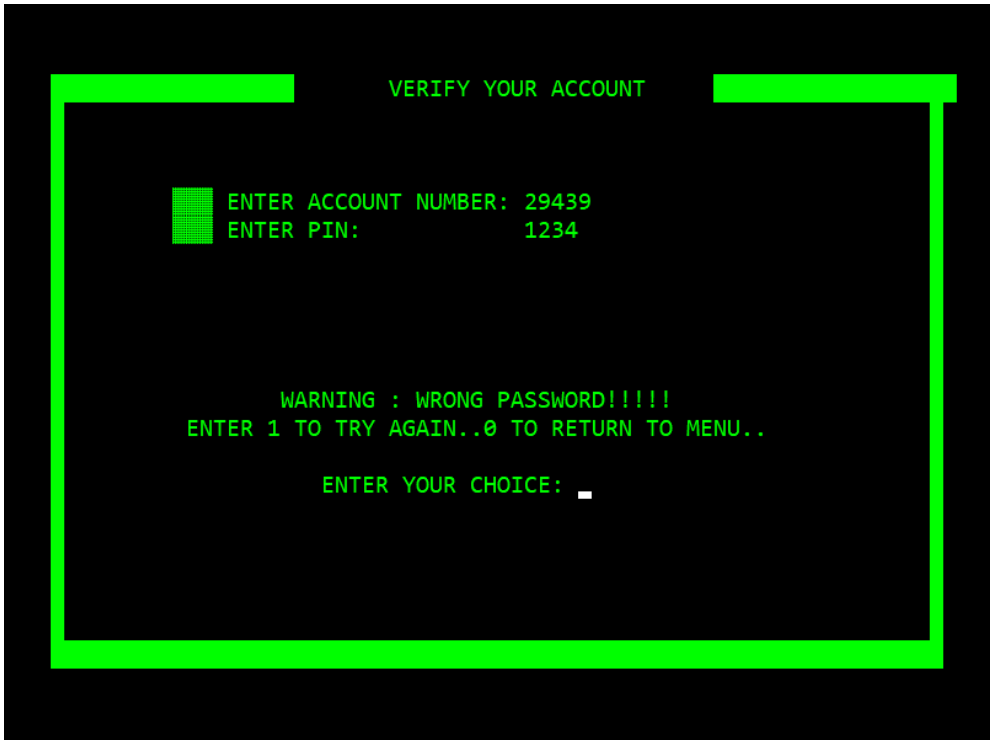


#8 - Change pin


VERIFY YOUR ACCOUNT
ENTER ACCOUNT NUMBER: 29439
ENTER PIN: 1111
WARNING : WRONG PASSWORD!!!!!
ENTER 1 TO TRY AGAIN..0 TO RETURN TO MENU..
ENTER YOUR CHOICE:

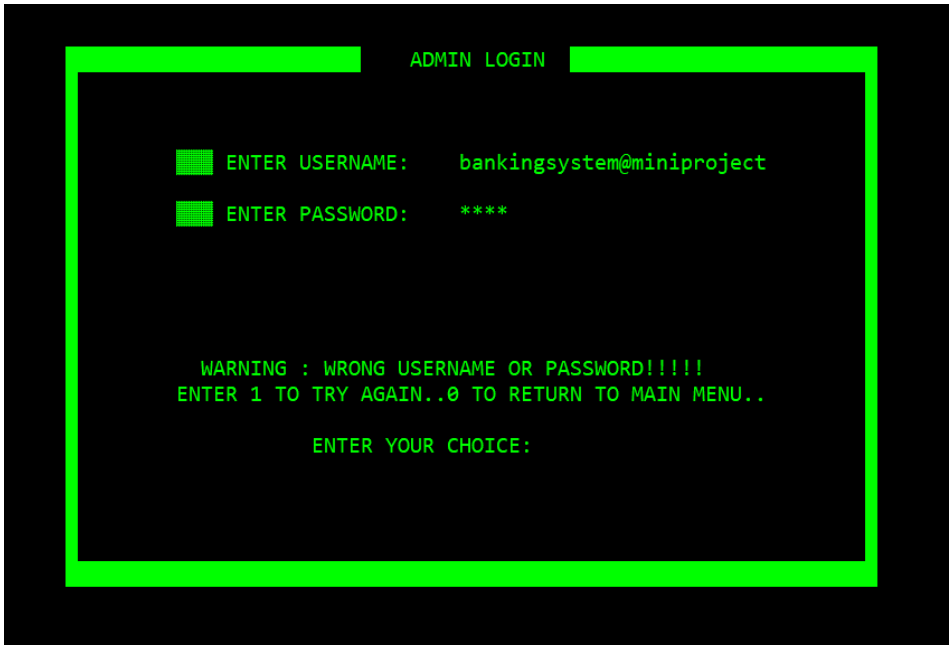

VERIFY YOUR ACCOUNT
ENTER ACCOUNT NUMBER: 29439
ENTER PIN: 9371
******PASSWORD MATCHED******
Enter New PIN: 1111
PIN CHANGED SUCCESSFULLY!!!

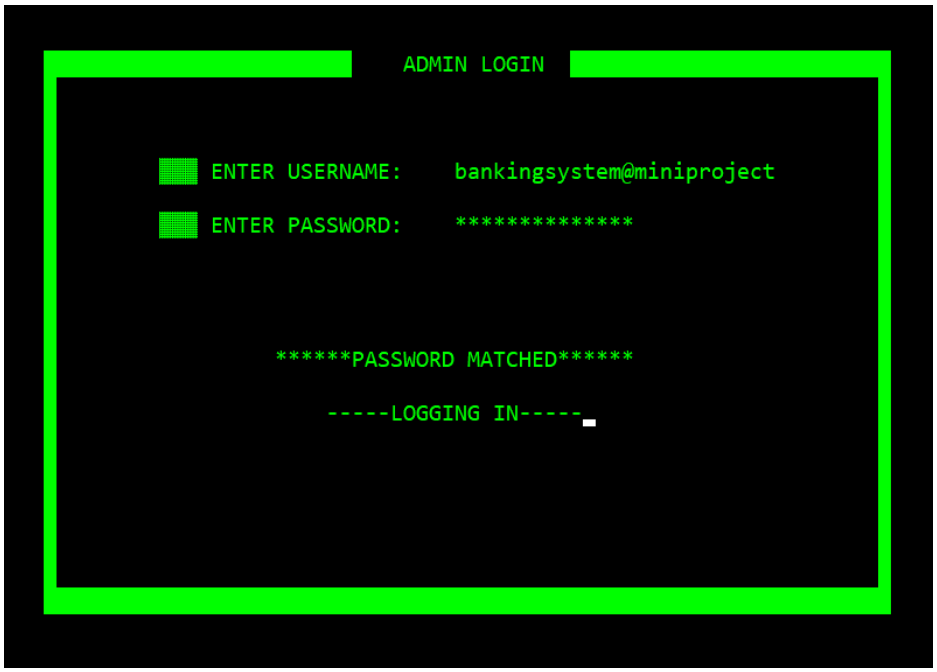
#9 - Close Bank Account



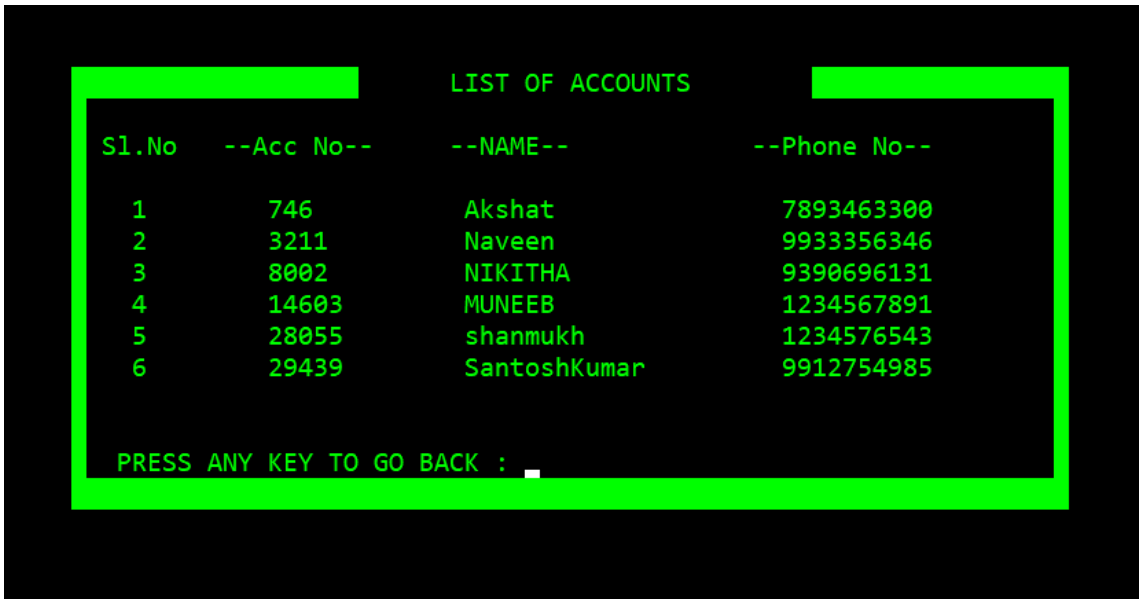
- **EMPLOYEE USE CASES**

#10 - Login

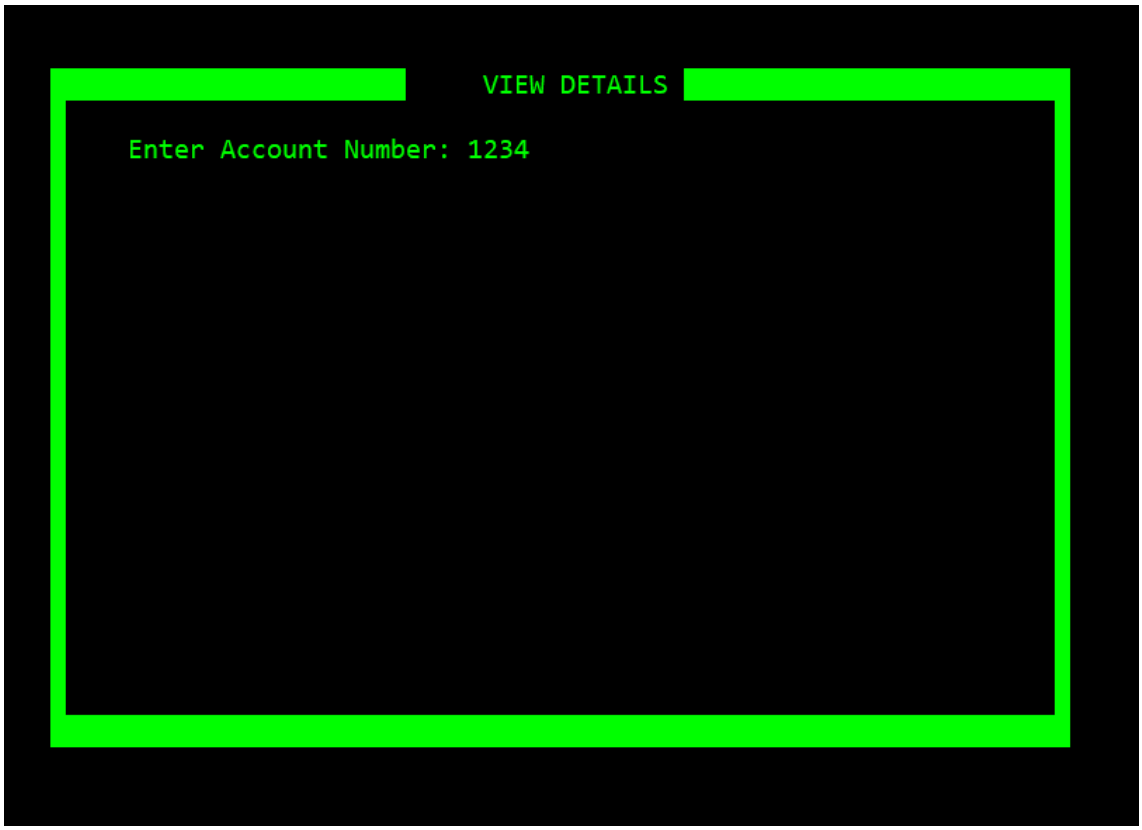


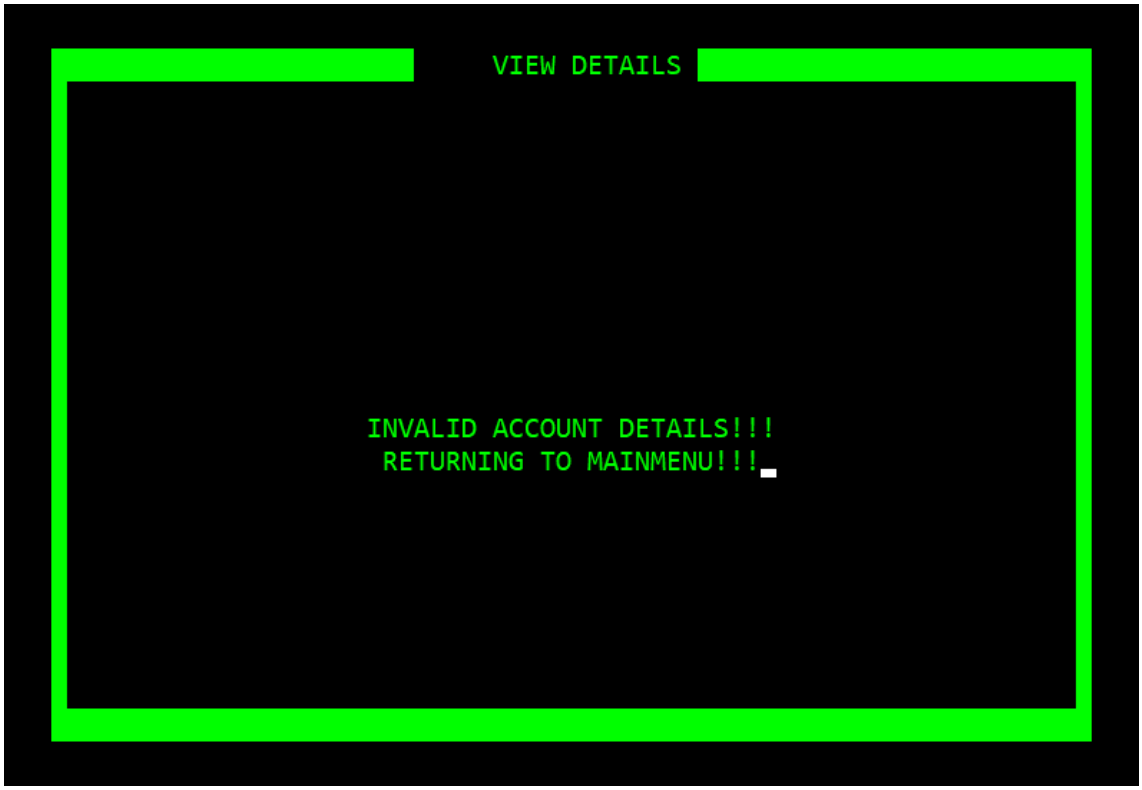


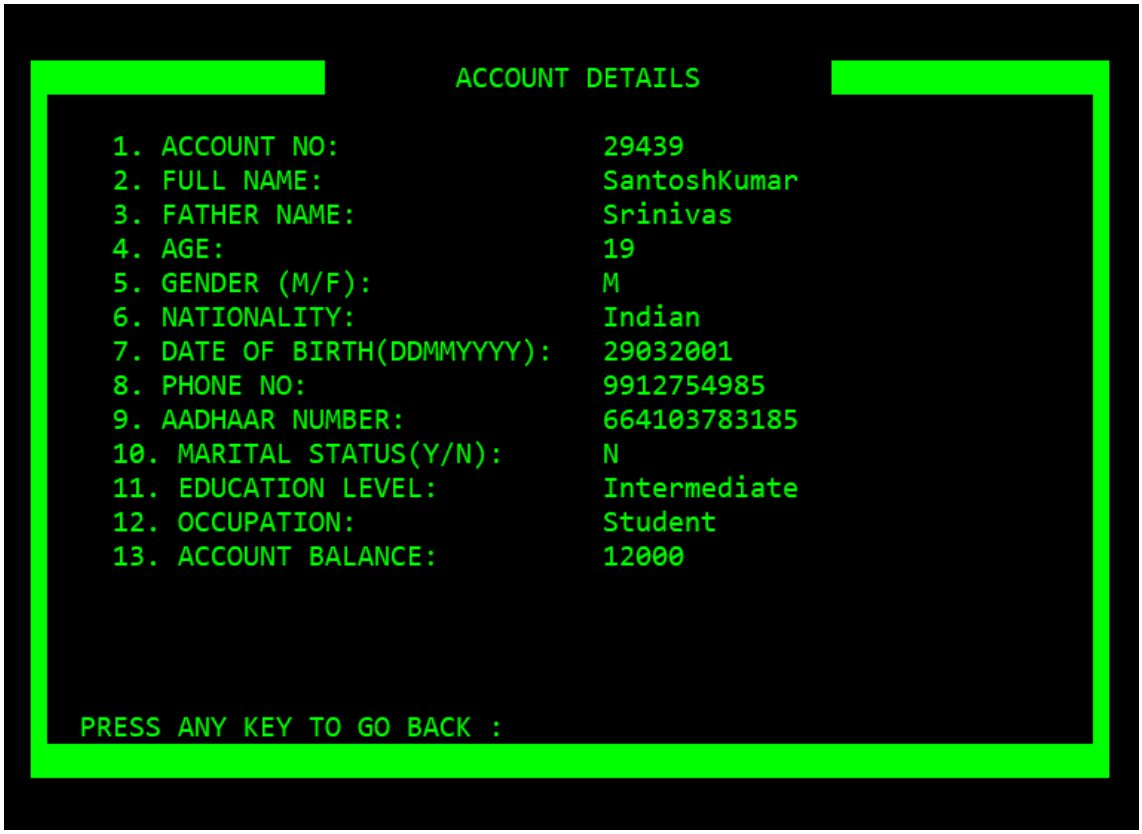
#11 - Account-Holder's List



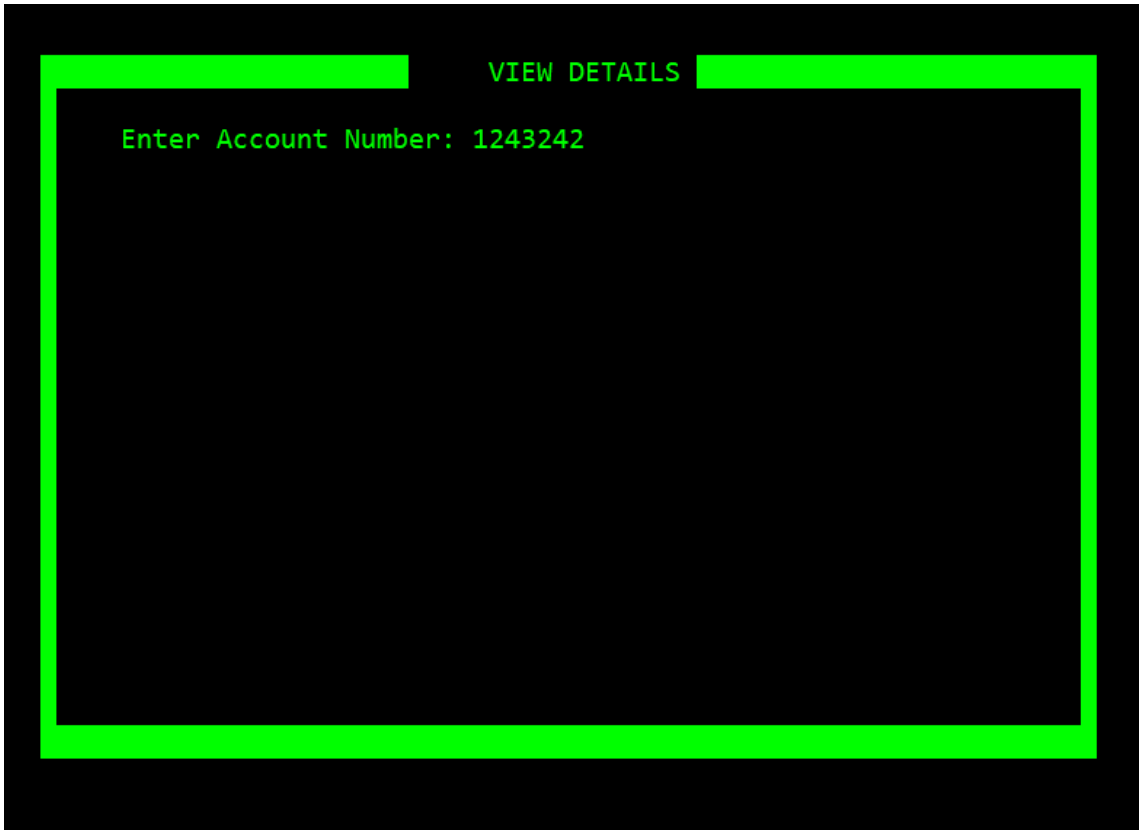
#12 - View Account-Holder Details

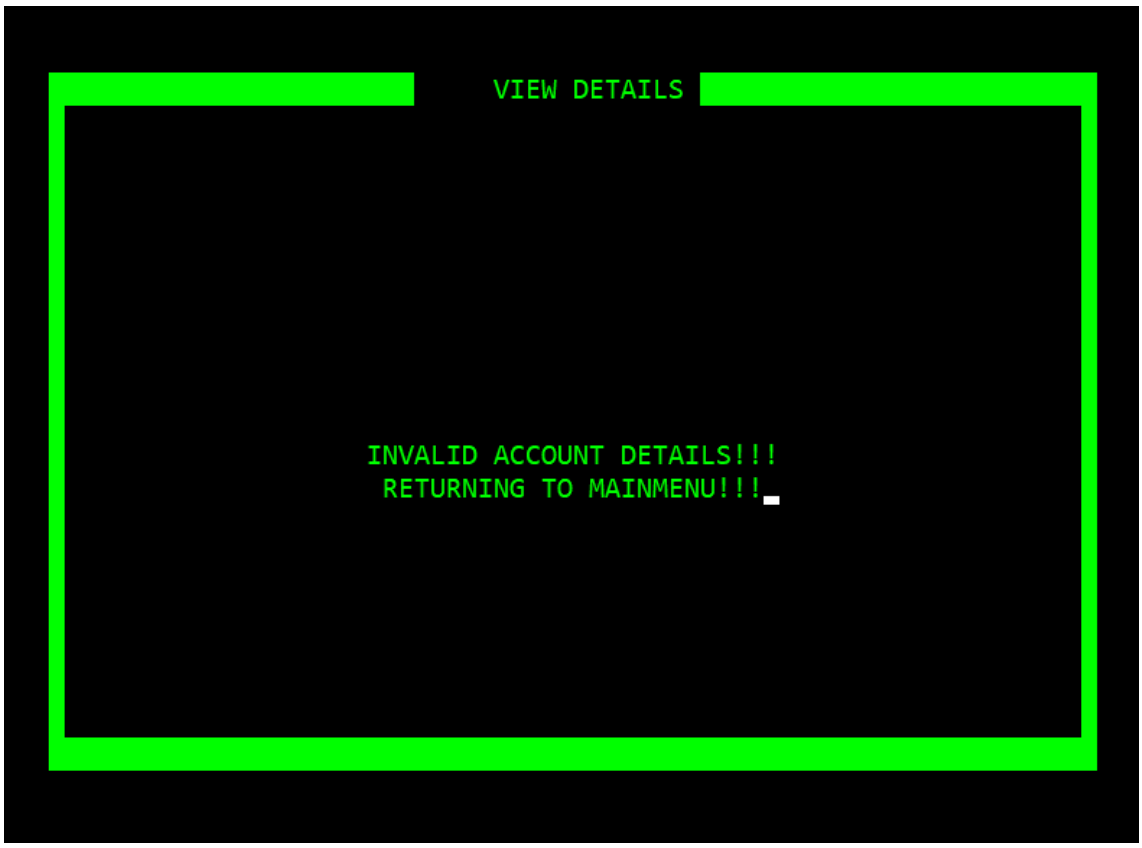


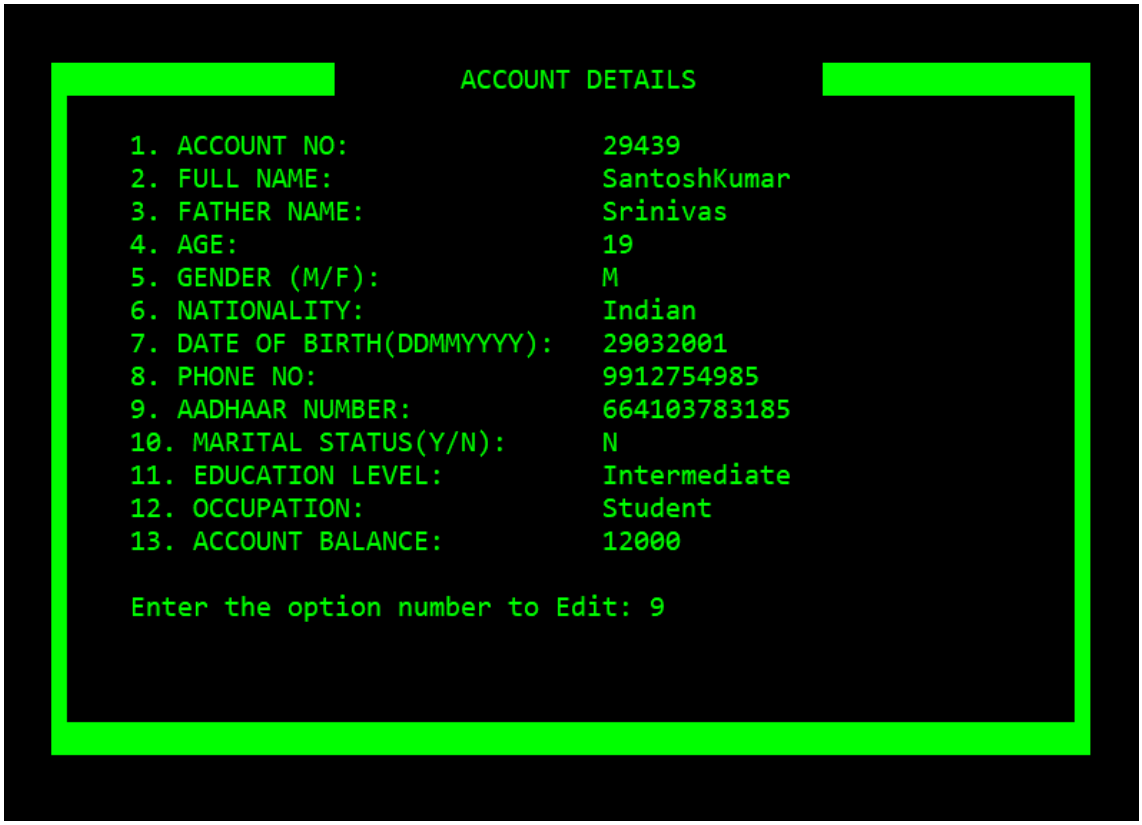

VIEW DETAILS
INVALID ACCOUNT DETAILS!!!
RETURNING TO MAINMENU!!!

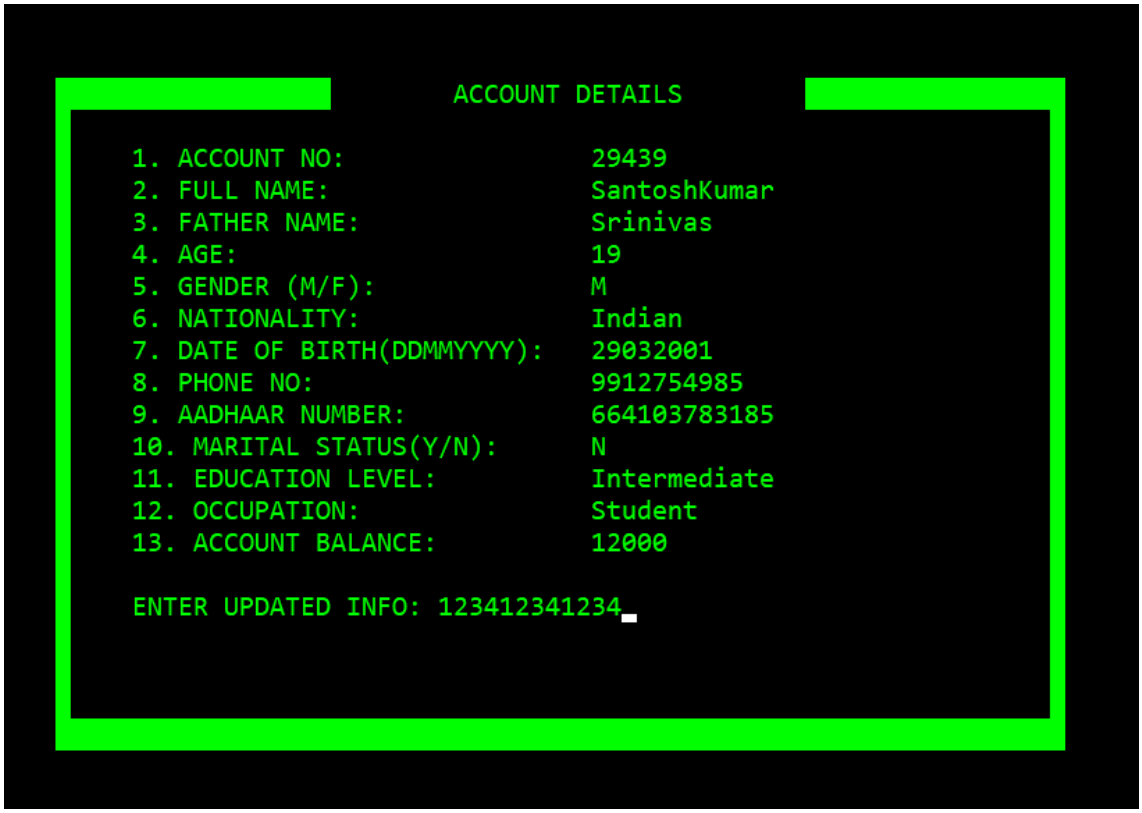

ACCOUNT DETAILS
1. ACCOUNT NO: 29439
2. FULL NAME: SantoshKumar
3. FATHER NAME: Srinivas
4. AGE: 19
5. GENDER (M/F): M
6. NATIONALITY: Indian
7. DATE OF BIRTH(DDMMYYYY): 29032001
8. PHONE NO: 9912754985
9. AADHAAR NUMBER: 664103783185
10. MARITAL STATUS(Y/N): N
11. EDUCATION LEVEL: Intermediate
12. OCCUPATION: Student
13. ACCOUNT BALANCE: 12000
PRESS ANY KEY TO GO BACK :

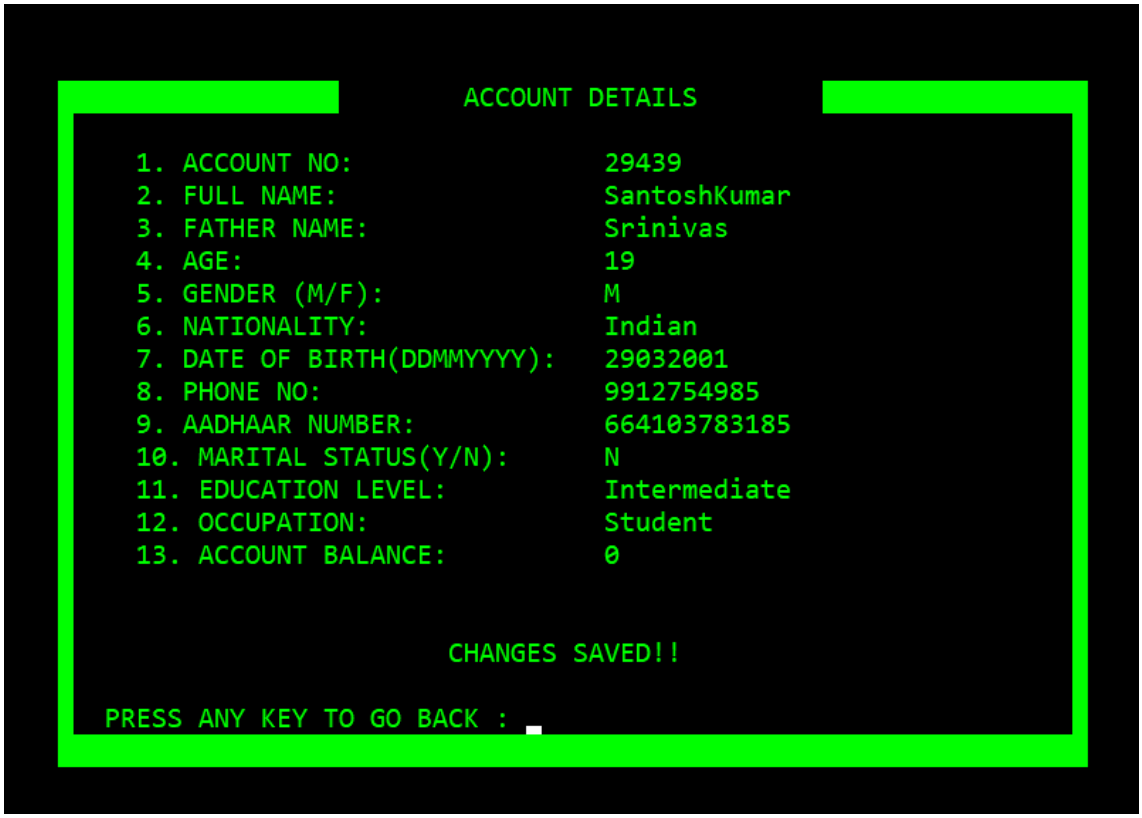
#13 - Edit Account Holder Details



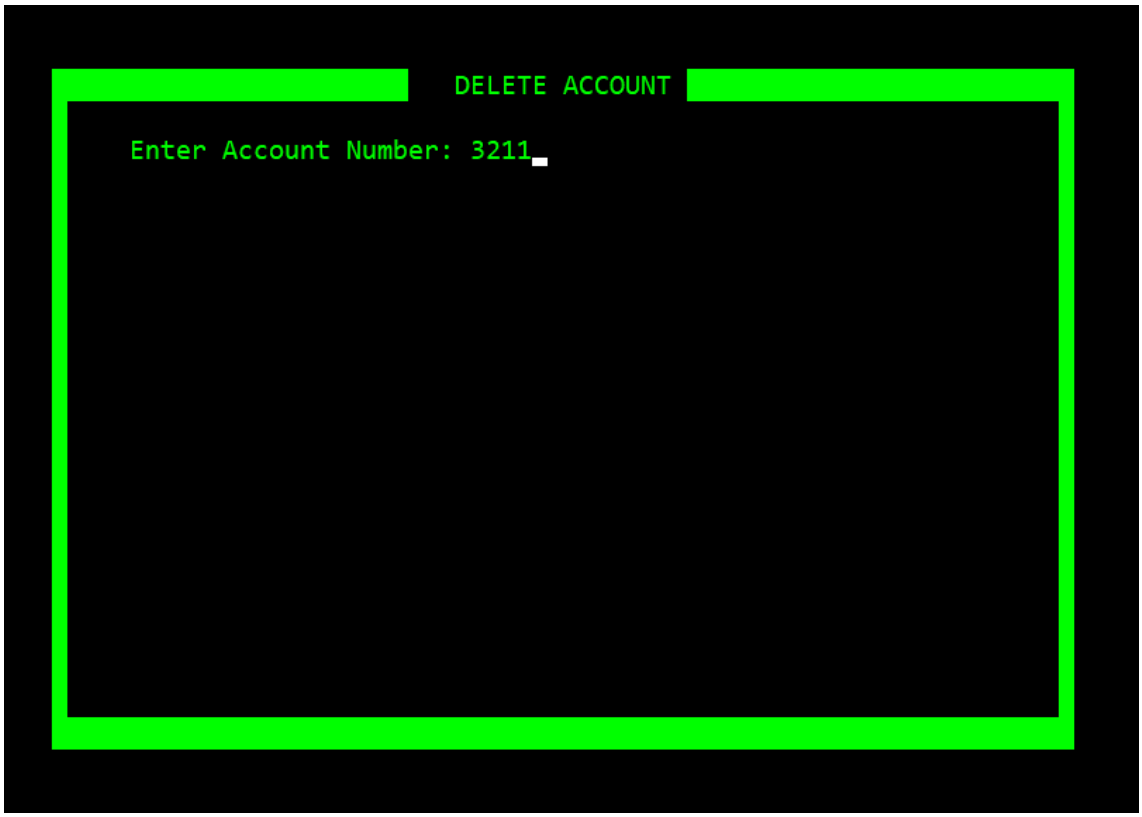


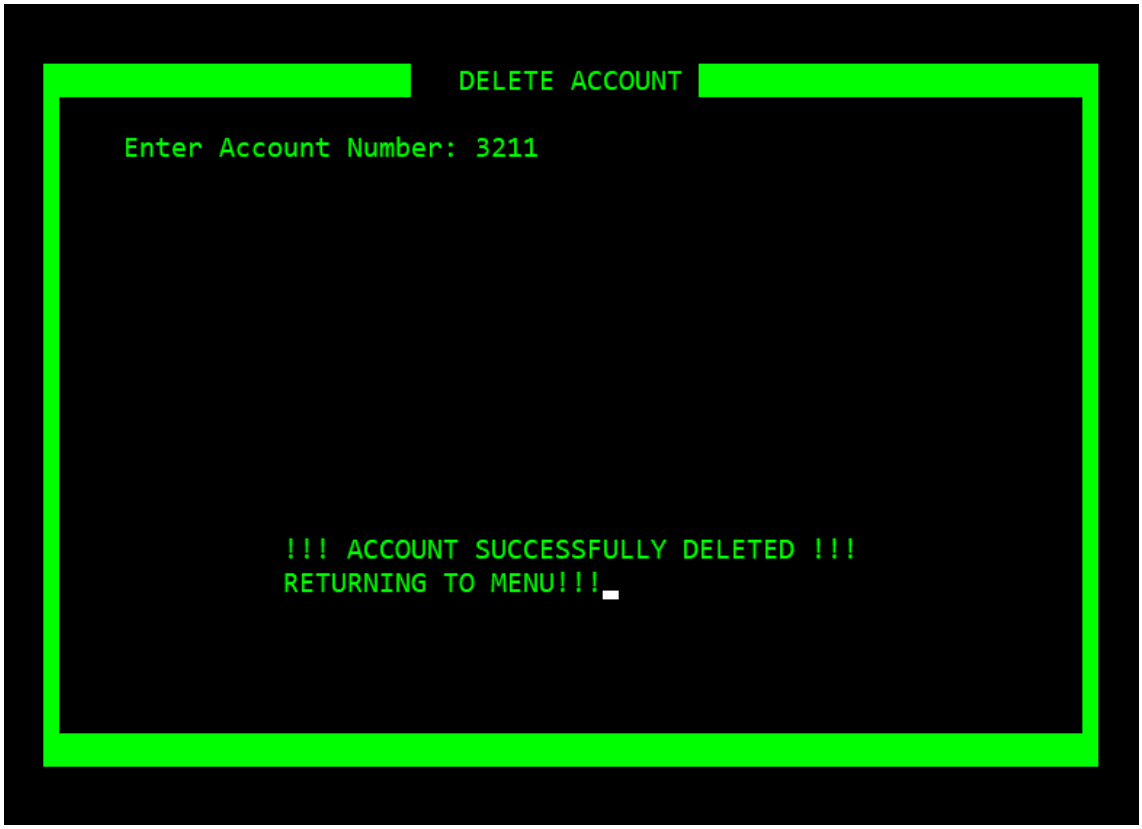

ACCOUNT DETAILS
1. ACCOUNT NO: 29439
2. FULL NAME: SantoshKumar
3. FATHER NAME: Srinivas
4. AGE: 19
5. GENDER (M/F): M
6. NATIONALITY: Indian
7. DATE OF BIRTH(DDMMYYYY): 29032001
8. PHONE NO: 9912754985
9. AADHAAR NUMBER: 664103783185
10. MARITAL STATUS(Y/N): N
11. EDUCATION LEVEL: Intermediate
12. OCCUPATION: Student
13. ACCOUNT BALANCE: 12000
Enter the option number to Edit: 9

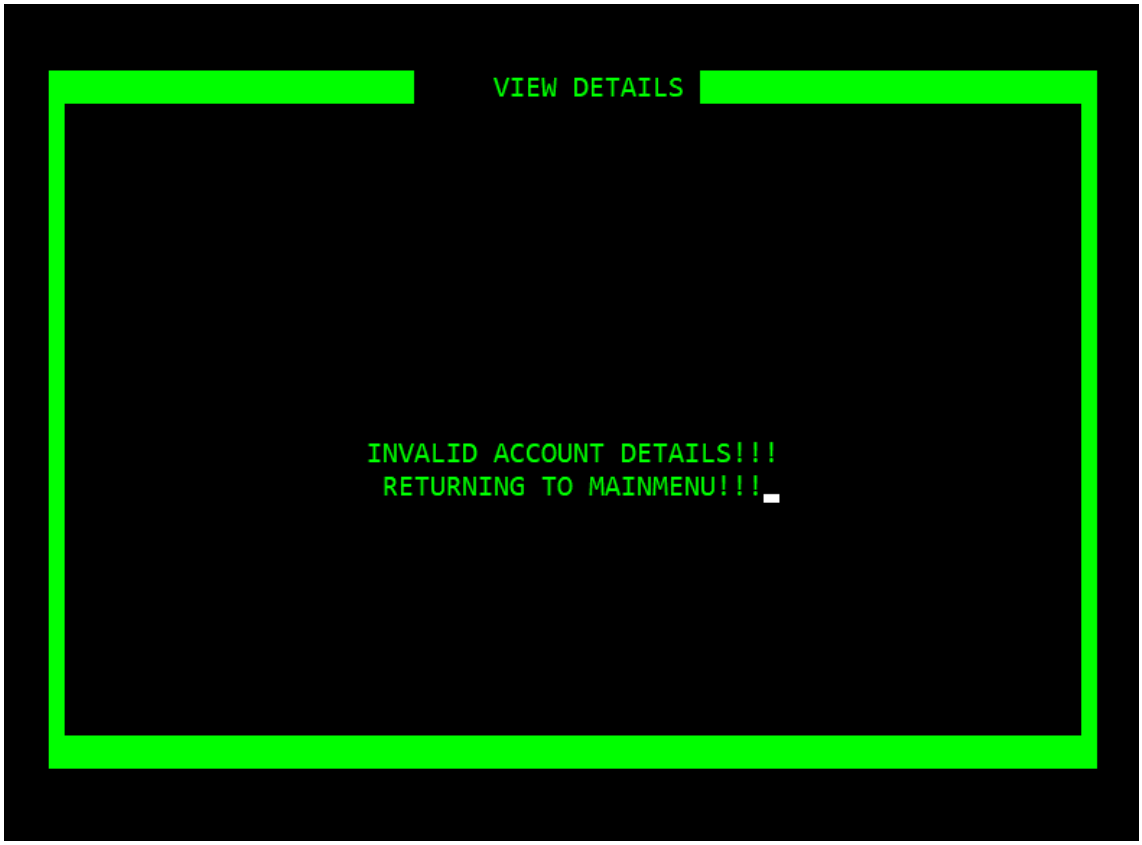

ACCOUNT DETAILS
1. ACCOUNT NO: 29439
2. FULL NAME: SantoshKumar
3. FATHER NAME: Srinivas
4. AGE: 19
5. GENDER (M/F): M
6. NATIONALITY: Indian
7. DATE OF BIRTH(DDMMYYYY): 29032001
8. PHONE NO: 9912754985
9. AADHAAR NUMBER: 664103783185
10. MARITAL STATUS(Y/N): N
11. EDUCATION LEVEL: Intermediate
12. OCCUPATION: Student
13. ACCOUNT BALANCE: 12000
ENTER UPDATED INFO: 123412341234



#14 - Delete Account of Account Holder

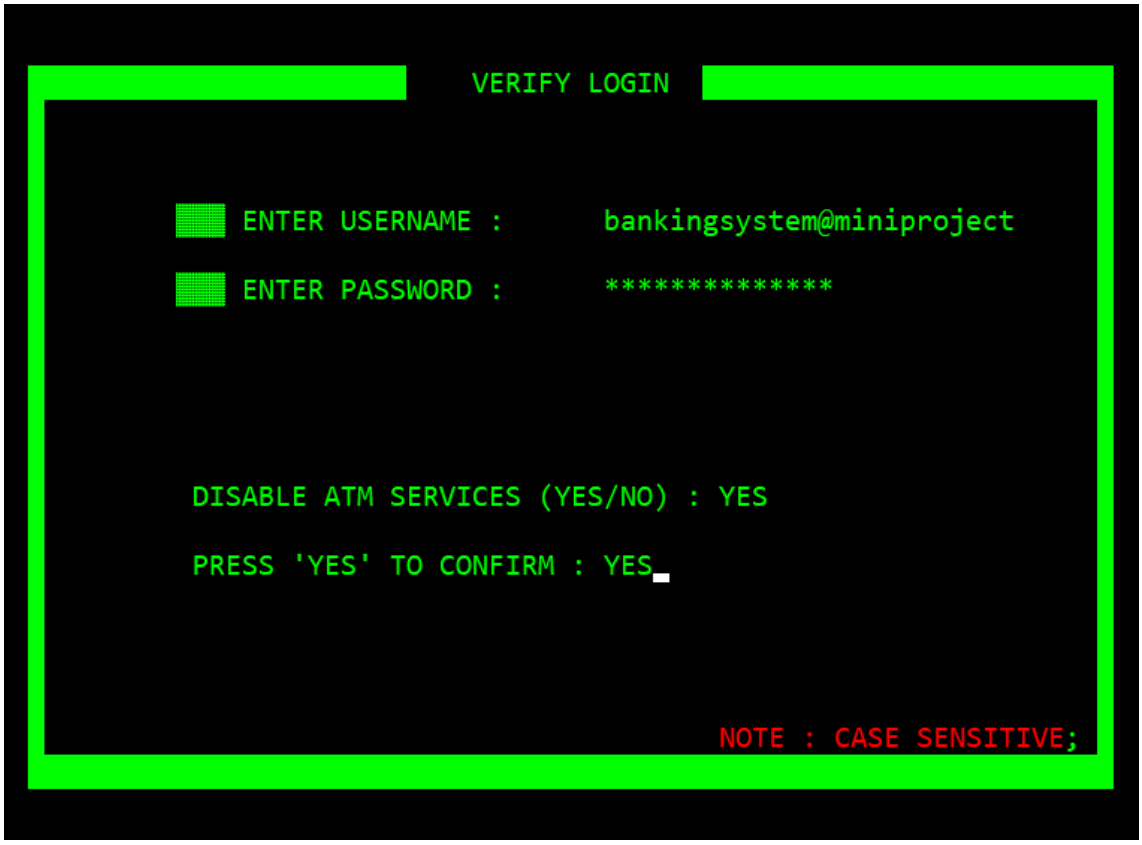


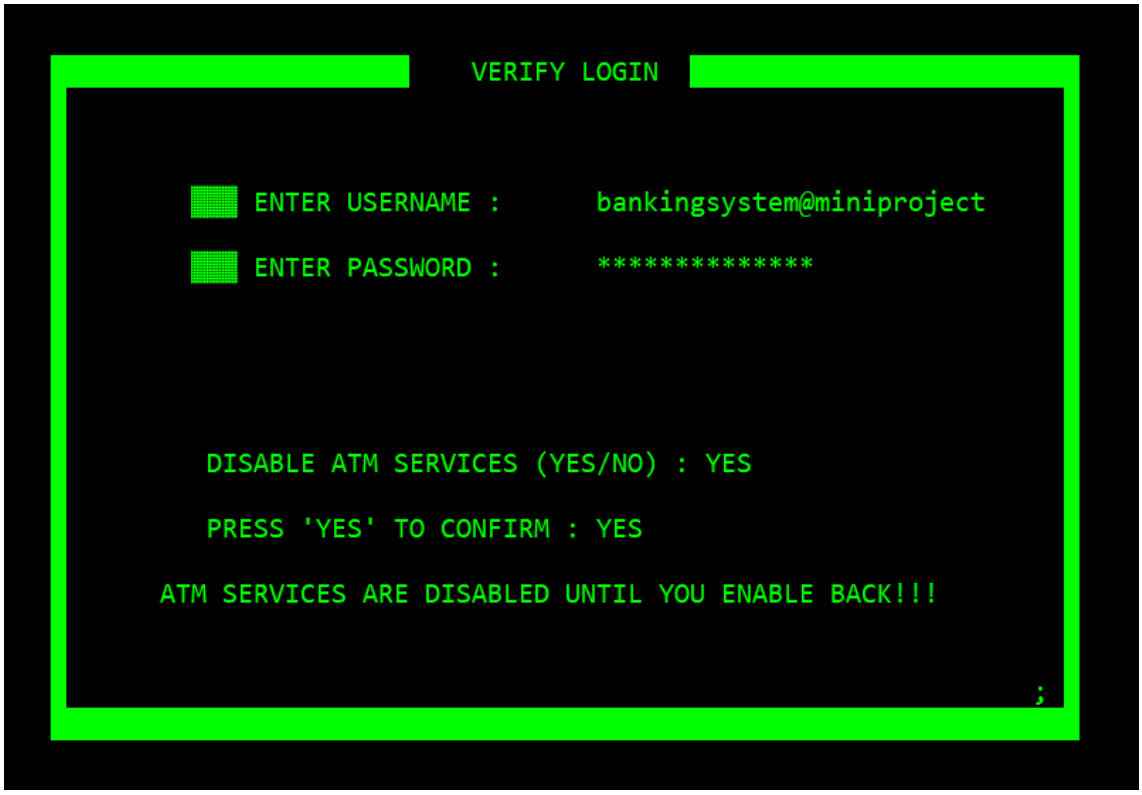

DELETE ACCOUNT
Enter Account Number: 3211
!!! ACCOUNT SUCCESSFULLY DELETED !!!
RETURNING TO MENU!!!

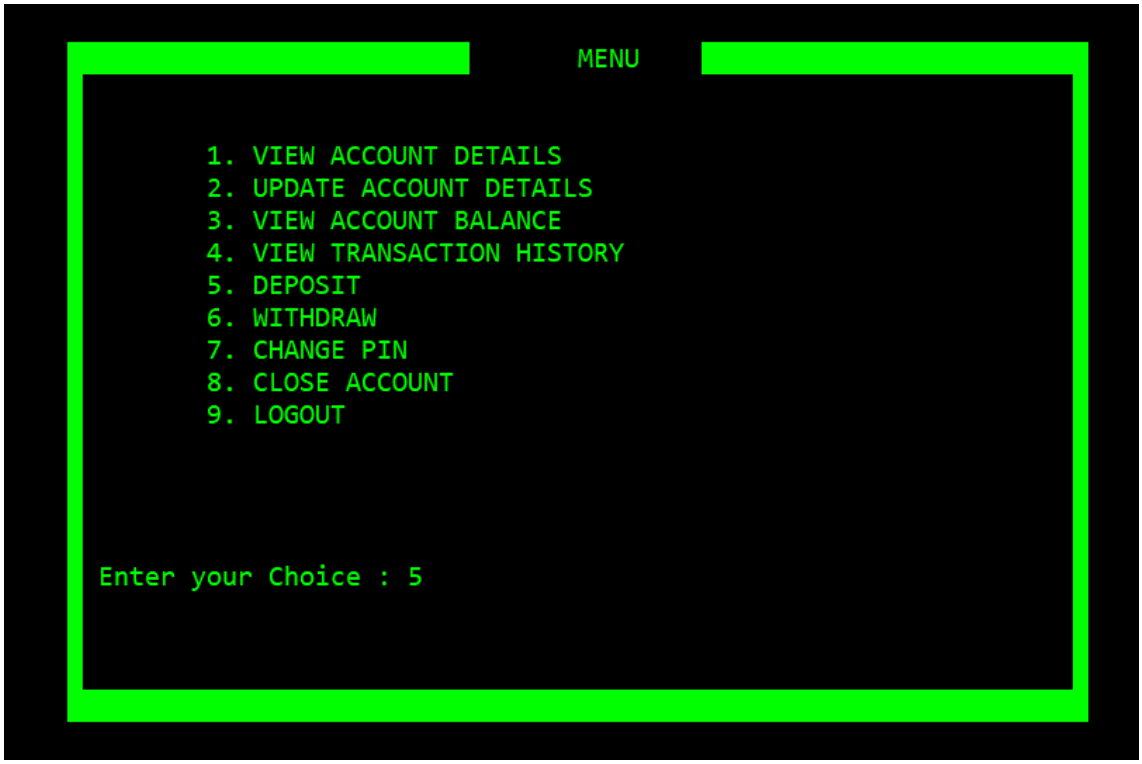

VIEW DETAILS
INVALID ACCOUNT DETAILS!!!
RETURNING TO MAINMENU!!!

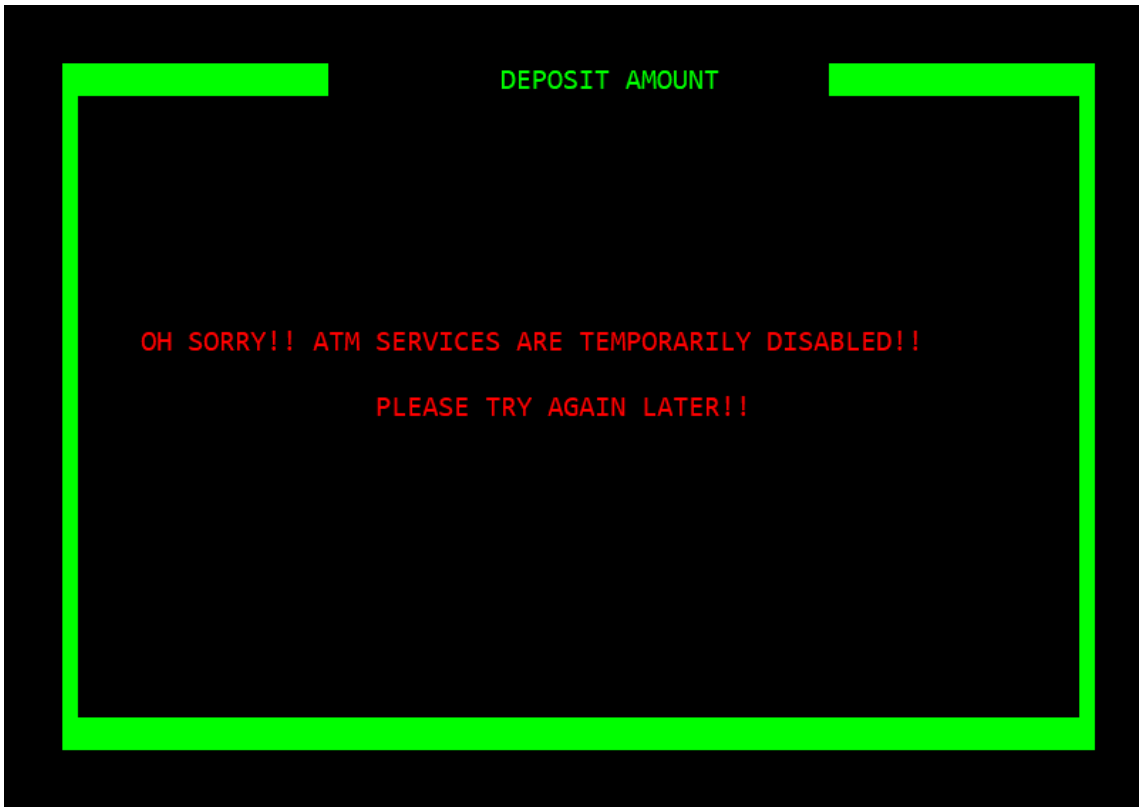
#15 - Add money to ATM

#16 - Disable ATM Services

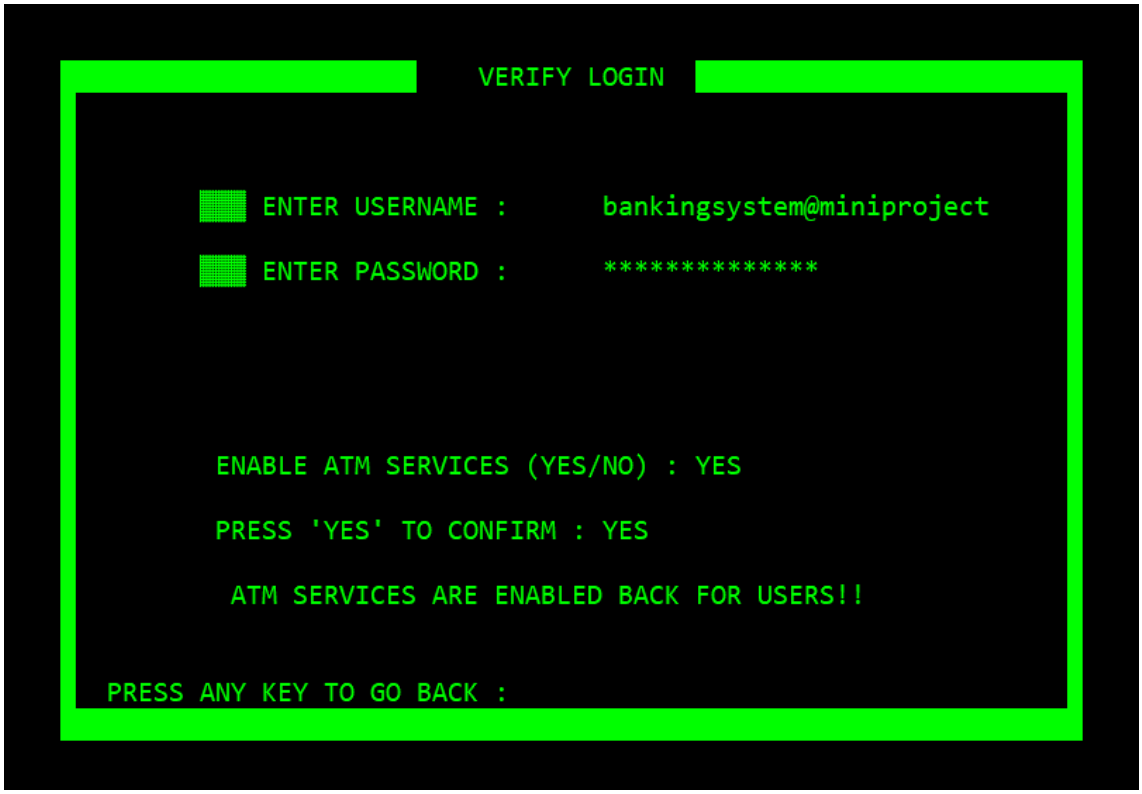

VERIFY LOGIN
ENTER USERNAME : bankingsystem@miniproject
ENTER PASSWORD : **********
DISABLE ATM SERVICES (YES/NO) : YES
PRESS 'YES' TO CONFIRM : YES
NOTE : CASE SENSITIVE;


VERIFY LOGIN
ENTER USERNAME : bankingsystem@miniproject
ENTER PASSWORD : **********
DISABLE ATM SERVICES (YES/NO) : YES
PRESS 'YES' TO CONFIRM : YES
ATM SERVICES ARE DISABLED UNTIL YOU ENABLE BACK!!!
;


MENU
1. VIEW ACCOUNT DETAILS
2. UPDATE ACCOUNT DETAILS
3. VIEW ACCOUNT BALANCE
4. VIEW TRANSACTION HISTORY
5. DEPOSIT
6. WITHDRAW
7. CHANGE PIN
8. CLOSE ACCOUNT
9. LOGOUT
Enter your Choice : 5


DEPOSIT AMOUNT
OH SORRY!! ATM SERVICES ARE TEMPORARILY DISABLED!!
PLEASE TRY AGAIN LATER!!

#17 - Enable ATM Services



5. ADDITIONAL KNOWLEDGE ACQUIRED

This project helped us to get a clear idea on fundamental concepts of C language. This project also helped us in improving our skills such as problem solving, critical thinking, time management and writing an efficient code which is more important.

In this project we learnt about gotoxy() function which enables us to change the cursor location to a desired location on the screen. We used ‘windows.h’ library to control the display colours for good look. We explored the ‘time.h’ and ‘conio.h’ libraries to implement the look-and-feel experience of a window application. We also learnt about some new functions in ‘string.h’ library like strtok() which is used for splitting a string by some delimiter. Also, this mini project helped us to know the importance of team work.

6. CONCLUSION AND FUTURE WORK

To conclude, this mini project of “Banking Management System” has helped us to get back with all the topics we have already learnt in C programming in the previous semester. We learnt to organize the code in a structured manner. It helped us to improve our skills in debugging.

Our future work includes adding more features in deposit processes like fixed deposits for a specific period of time, account to account transactions, creating different type of accounts like savings and current account. We would like to add a feature where costumers can raise a complaint or report a problem to the employee. We also want to implement the concepts of data structures in our code which will improve the performance of our application. We would also like to extend this project to a web based application.

7. REFERENCES

- **WEBSITE**: www.stackoverflow.com
  USED FOR: For Debugging

- **WEBSITE:** www.geeksforgeeks.com
  LINK: https://www.geeksforgeeks.org/strtok-strtok_r-functions-c-examples/
  USED FOR: strtok function

- **WEBSITE**: www.programmingsimplified.com
  LINK: https://www.programmingsimplified.com/c/dos.h/sleep
  USED FOR: sleep function

- **WEBSITE:** www.cprogramming.com
- **LINK:** https://cboard.cprogramming.com/c-programming/42482-setting-cursor-position-c.html
- **USED FOR:** gotoxy function