Amad Ahmed Ubaidullah Bisma Ijaz

352706 345348 351995

**DIGiBANK**

Object-Oriented Project

**Object Oriented Programming Using C++ & Windows form.**

**BSCS 10A**

**First look**

**Graphical user interface, website

Description automatically generated**

**Graphical user interface, website

Description automatically generated**

**Graphical user interface, text, application, Teams

Description automatically generated**

**Graphical user interface, text, application

Description automatically generated**

**Graphical user interface, application

Description automatically generated**

**Text

Description automatically generated**

**SOURCE CODE (only Console C++)**

/\*The following Program is of a bank management system

The system allows user to perform various actions with his money

deposited in his account ranging from applying for loans

,transferring payments, paying his bills to collecting his interest etc.\*/

#include<iostream>

#include<fstream>

#include<stdio.h>

#include<conio.h>

#include<windows.h>

#include <cstdio>

using namespace std;

class USER {

protected:

float User\_balance = 0.0;

string User\_Pin, UserFirst\_Name, User\_ID, User\_Password, User\_Fname, User\_Address, User\_Phone, AccountType, UserLast\_Name;

public:

void CheckBalance();

void WithdrawAmount();

void Transfer\_Option();

void Payment\_Option();

void AccountDetails();

void ApplyForLoan();

};

class ADMIN :public USER {

protected:

string AM\_Email, AM\_Password, AM\_Pin,

BI\_Email, BI\_Password, BI\_Pin,

UB\_Email, UB\_Password, UB\_Pin;

public:

ADMIN();

void NewUser();

void AlreadyUser();

void Deposit();

void SearchUserRecord();

void EditUserRecord();

void DeleteUserRecord();

void ShowAllRecords();

void PaymentAllRecords();

};

class BANK : public ADMIN {

public:

void Mainmenu();

void Admin();

void User();

void Security();

};

////////////////////////// MAINMENU ///////////////////////

void BANK::Mainmenu() {

int Ans;

do

{

system("cls");

// Main menu

cout << "\n\n\n\n\t\t\t\t\t \x1b\033[3;47;35mWELCOME TO DIGiBANK \033[0m\033[0m\n";

cout << "\n\x1b[33m\n\n\n\t 1. User\033[0m\t\t";

cout << "\x1b[33m\n\n\t 2. Admin\033[0m\t\t";

cout << "\x1b[33m\n\n\t 3. EXIT\033[0m\t\t";

cout << "\n\n\t Enter Your Choice as to who you represent :: ";

cin >> Ans;

// Decision

switch (Ans)

{

case 1: User(); break;

case 2: Security(); break;

case 3: exit(0);

default:cout << "\x1b[33m\n\n\n\n\t\t\t\t\t Invalid Value, Try Again....\n \033[0m\t\t";

cout << "\n\n\n\n\t\t\t\t\t ";

system("pause");

}

} while (1);

}

// For Admins

void BANK::Security() {

string Email, Password, Pin;

char PI, PASS;

system("cls");

cout << "\x1b[36m\n\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t| ADMIN LOGIN |\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

cout << "\n\n\n E-mail : ";

cin >> Email;

cout << "\n\n Password : ";

// Getting Password & Displaying "\*"

for (int i = 1; i <= 5; i++) {

PASS = \_getch();

Password += PASS;

cout << "\*";

}

cout << "\n\n Pin Code : ";

// Getting pin & Displaying "\*"

for (int i = 1; i <= 4; i++) {

PI = \_getch();

Pin += PI;

cout << "\*";

}

if (((Email == AM\_Email) || (Email == BI\_Email) || (Email == UB\_Email)) && ((Password == AM\_Password) || (Password == BI\_Password) || (Password == UB\_Password)) && ((Pin == AM\_Pin) || (Pin == BI\_Pin) || (Pin == UB\_Pin)))

Admin();

else {

cout << "\x1b[33m\n\n\n\t\t\t\t\t Your credentials are incorrect or have expired. Please try again or reset your password...\n \033[0m\t\t";

cout << "\n\n\n\n\t\t\t\t\t ";

system("pause");

}

}

// Menu of ADMIN

void BANK::Admin() {

do

{

system("cls");

int Ans;

cout << "\n\n\n\n\t\t\t\t\t\033[3;47;35m ADMIN MANAGEMENT \033[0m\t\t";

cout << "\x1b[36m\n\n\n\t 1. New User";

cout << "\n\n\t 2. Already User";

cout << "\n\n\t 3. Deposit Amount";

cout << "\n\n\t 4. Search User Records";

cout << "\n\n\t 5. Edit User Records";

cout << "\n\n\t 6. Delete User Records";

cout << "\n\n\t 7. Show All Records";

cout << "\n\n\t 8. Check Payments";

cout << "\n\n\t 9. GO BACK\033[0m\t\t";

cout << "\n\n Your Choice :: ";

cin >> Ans;

switch (Ans)

{

case 1:NewUser(); break;

case 2:AlreadyUser(); break;

case 3:Deposit(); break;

case 4:SearchUserRecord(); break;

case 5:EditUserRecord(); break;

case 6:DeleteUserRecord(); break;

case 7:ShowAllRecords(); break;

case 8:PaymentAllRecords(); break;

case 9:Mainmenu(); break;

default:cout << "\x1b[33m\n\n\n\n\t\t\t\t\t Invalid Value. Please try again ...\n \033[0m\t\t";

cout << "\n\n\t\t\t\t\t ";

system("pause"); break;

}

} while (1); // Infinte

}

// Menu of USER

void BANK::User() {

do

{

system("cls");

int Ans;

cout << "\n\n\n\n\t\t\t\t\t\033[4;47;35m USER ACCOUNT \033[0m\t\t\n";

//cout << "\n\n\n\n\t\t\t\t\t \x1b[36mUSER ACCOUNT";

cout << " \x1b[36m\n\n\t 1. Check Balance";

cout << "\n\n\t 2. WithDraw Amount";

cout << "\n\n\t 3. Transfer Options";

cout << "\n\n\t 4. Payment Options";

cout << "\n\n\t 5. Account Details";

cout << "\n\n\t 6. Apply for Loans";

cout << "\n\n\t 7. GO BACK\033[0m\t\t";

cout << "\n\n Your Choice :: ";

cin >> Ans;

switch (Ans)

{

case 1:CheckBalance(); break;

case 2:WithdrawAmount(); break;

case 3:Transfer\_Option(); break;

case 4:Payment\_Option(); break;

case 5:AccountDetails(); break;

case 6:ApplyForLoan(); break;

case 7:Mainmenu(); break;

default:cout << "\x1b[33m\n\n\n\n\t\t\t\t\t Invalid Value. Please try again ...\n \033[0m\t\t";

cout << "\n\n\t\t\t\t\t ";

system("pause"); break;

}

} while (1); // Infinte

}

////////////////////////// ADMIN ///////////////////////

ADMIN::ADMIN() {

AM\_Email = "amad@digibank.com";

AM\_Password = "Ammad";

AM\_Pin = "3527";

BI\_Email = "bisma@digibank.com";

BI\_Password = "Bisma";

BI\_Pin = "3519";

UB\_Email = "ubaid@digibank.com";

UB\_Password = "Ubaid";

UB\_Pin = "3453";

}

void ADMIN::NewUser() {

fstream file;

//Back: // If User ID Already Exit then again

system("cls");

// Temp variables to catch data from file

int ch, found = 0;

float bal;

string Na, La, Fna, Pa, Add, Ph, ID, pi, AT;

cout << "\x1b[36m\n\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t| ADD NEW USER |\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

cout << "\n\n\n User ID : ";

cin >> User\_ID;

cout << "\n\n First Name : ";

cin >> UserFirst\_Name;

cout << "\n\n Last Name : ";

cin >> UserLast\_Name;

cout << "\n\n Father Name : ";

cin >> User\_Fname;

cout << "\n\n Permanent Address : ";

cin >> User\_Address;

cout << "\n\n Pin Code : ";

cin >> User\_Pin;

cout << "\n\n Password : ";

cin >> User\_Password;

cout << "\n\n Phone No. : ";

cin >> User\_Phone;

cout << "\n\n Deposit Amount : ";

cin >> User\_balance;

abc:

cout << "\x1b[36m\n\n\t\tAccount Type\033[0m";

cout << "\n\n\t\x1b[35m1. Platinum\033[0m \x1b[33m2. Gold\033[0m 3. Silver\n\n Select One :: ";

cin >> ch;

switch (ch)

{

case 1: AccountType = "PLATINIUM"; break;

case 2: AccountType = "GOLD"; break;

case 3: AccountType = "SILVER"; break;

default:

cout << "\x1b[33m\n\n Invalid Choice ... \033[0m";

system("pause");

system("cls");

cout << "\x1b[36m\n\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t| ADD NEW USER |\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

cout << "\n\n User ID : " << User\_ID;

cout << "\n\n First Name : " << UserFirst\_Name;

cout << "\n\n Last Name : " << UserLast\_Name;

cout << "\n\n Father Name : " << User\_Fname;

cout << "\n\n Permanent Address : " << User\_Address;

cout << "\n\n Pin Code : " << User\_Pin;

cout << "\n\n Password : " << User\_Password;

cout << "\n\n Phone No. : " << User\_Phone;

cout << "\n\n Deposit Amount : " << User\_balance;

goto abc;

}

file.open("DATA.dat", ios::in); // Open file

if (!file) // If file does not exit

{

file.open("DATA.dat", ios::app | ios::out);

file << User\_ID << " " << UserFirst\_Name << " " << UserLast\_Name << " " << User\_Fname << " " << User\_Address << " " << User\_Pin << " " << User\_Password << " " << User\_Phone << " " << User\_balance << " " << AccountType << "\n";

file.close();

}

else

{

file >> ID >> Na >> La >> Fna >> Add >> pi >> Pa >> Ph >> bal >> AT;

while (!(file.eof()))

{

if (ID == User\_ID)

{

cout << "\x1b[33m\n\n\t\t\t\t\t User ID Already Exit...\n\n \033[0m";

cout << "\n\n\t\t\t\t\t ";

system("pause");

found++;

break;

//goto Back;

}

file >> ID >> Na >> La >> Fna >> Add >> pi >> Pa >> Ph >> bal >> AT;

}

file.close();

if (found == 0)

{

file.open("DATA.dat", ios::app | ios::out);

file << User\_ID << " " << UserFirst\_Name << " " << UserLast\_Name << " " << User\_Fname << " " << User\_Address << " " << User\_Pin << " " << User\_Password << " " << User\_Phone << " " << User\_balance << " " << AccountType << "\n";

file.close();

}

}

if (found == 0)

{

cout << "\n\n\t\t\t\t\t Account Created Successfully....\n\n ";

cout << "\n\n\t\t\t\t\t ";

system("pause");

}

}

void ADMIN::AlreadyUser() {

fstream file;

system("cls");

string id;

int Found = 0;

cout << "\x1b[36m\n\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t| ALREADY USER |\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

file.open("DATA.dat", ios::in);

// If file does not exit

if (!file) {

cout << "\x1b[33m\n\n\n\t\t\t\t\t File Does not Exit\n\n \033[0m";

cout << "\n\n\t\t\t\t\t ";

system("pause");

}

else

{

cout << "\n\n Enter User ID :: ";

cin >> id;

file >> User\_ID >> UserFirst\_Name >> UserLast\_Name >> User\_Fname >> User\_Address >> User\_Pin >> User\_Password >> User\_Phone >> User\_balance >> AccountType;

while (!file.eof())

{

if (id == User\_ID)

{

system("cls");

cout << "\x1b[36m\n\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t| ALREADY USER |\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

cout << "\n\n User ID :: " << User\_ID << "\n Pin code :: " << User\_Pin << "\n Password :: " << User\_Password;

Found++;

//break;

}

file >> User\_ID >> UserFirst\_Name >> UserLast\_Name >> User\_Fname >> User\_Address >> User\_Pin >> User\_Password >> User\_Phone >> User\_balance >> AccountType;

}

file.close();

if (Found == 0)

cout << "\x1b[33m\n\n\t\t\t\t\t Cannot find user record, Kindly check the ID or Record does not Exist.....\033[0m";

cout << "\n\n\t\t\t\t\t ";

system("pause");

}

}

void ADMIN::Deposit()

{

fstream file, updatedFile;

string id;

float depAmount;

int Found = 0;

system("cls");

cout << "\x1b[36m\n\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t| DEPOSIT AMOUNT |\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

file.open("DATA.dat", ios::in);

if (!file) {

cout << "\x1b[33m\n\n\n\t\t\t\t\t File Does not Exit\n\n \033[0m";

cout << "\n\n\t\t\t\t\t ";

system("pause");

}

else

{

cout << "\n\n Enter User ID :: ";

cin >> id;

updatedFile.open("updatedData.dat", ios::app | ios::out);

file >> User\_ID >> UserFirst\_Name >> UserLast\_Name >> User\_Fname >> User\_Address >> User\_Pin >> User\_Password >> User\_Phone >> User\_balance >> AccountType;

while (!file.eof())

{

if (id == User\_ID)

{

cout << "\n\n Amount for Deposit :: ";

cin >> depAmount;

User\_balance += depAmount;

updatedFile << User\_ID << " " << UserFirst\_Name << " " << UserLast\_Name << " " << User\_Fname << " " << User\_Address << " " << User\_Pin << " " << User\_Password << " " << User\_Phone << " " << User\_balance << " " << AccountType << "\n";

Found++;

cout << "\n\n\t\t\t\t\t Your Amount " << depAmount << " Successfully Deposit\n\n";

}

else

{

updatedFile << User\_ID << " " << UserFirst\_Name << " " << UserLast\_Name << " " << User\_Fname << " " << User\_Address << " " << User\_Pin << " " << User\_Password << " " << User\_Phone << " " << User\_balance << " " << AccountType << "\n";

}

file >> User\_ID >> UserFirst\_Name >> UserLast\_Name >> User\_Fname >> User\_Address >> User\_Pin >> User\_Password >> User\_Phone >> User\_balance >> AccountType;

}

file.close();

updatedFile.close();

if (remove("DATA.dat") == 0)

cout << ".";

if (rename("updatedData.dat", "DATA.dat") == 0)

cout << ".";

if (Found == 0)

cout << "\x1b[33m\n\n\t\t\t\t\t Cannot find user record, Kindly check the ID or Record does not Exist.....\033[0m";

cout << "\n\n\t\t\t\t\t ";

system("pause");

}

}

void ADMIN::SearchUserRecord()

{

fstream file;

system("cls");

string id;

int found = 0;

cout << "\x1b[36m\n\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t| SEARCHING RECORDS |\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

file.open("DATA.dat", ios::in);

if (!file) {

cout << "\x1b[33m\n\n\t\t\t\t\t File Does not Exit\n\n \033[0m";

cout << "\n\n\t\t\t\t\t ";

system("pause");

}

else

{

cout << "\n\n User ID :: ";

cin >> id;

file >> User\_ID >> UserFirst\_Name >> UserLast\_Name >> User\_Fname >> User\_Address >> User\_Pin >> User\_Password >> User\_Phone >> User\_balance >> AccountType;

while (!file.eof()) {

if (id == User\_ID) {

system("cls");

cout << "\x1b[36m\n\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t| SEARCHING RECORDS |\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

cout << "\n\n\n User ID :: " << User\_ID

<< "\n\n First Name :: " << UserFirst\_Name

<< "\n\n Last Name :: " << UserLast\_Name

<< "\n\n Father Name :: " << User\_Fname

<< "\n\n Permanent Address :: " << User\_Address

<< "\n\n Phone Number :: " << User\_Phone

<< "\n\n Balance :: " << User\_balance

<< "\n\n Account Type :: " << AccountType;

cout << "\n\n====================================================\n\n ";

found++;

}

file >> User\_ID >> UserFirst\_Name >> UserLast\_Name >> User\_Fname >> User\_Address >> User\_Pin >> User\_Password >> User\_Phone >> User\_balance >> AccountType;

}

file.close();

if (found == 0)

cout << "\x1b[33m\n\n\t\t\t\t\t Cannot find user record, Kindly check the ID or Record does not Exist.....\033[0m";

cout << "\n\n\t\t\t\t\t ";

system("pause");

}

}

void ADMIN::EditUserRecord()

{

int found = 0;

system("cls");

fstream file, updatedFile;

string id, N, La, Fn, Add, Pi, Pass, Ph, AT;

int ch;

cout << "\x1b[36m\n\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t| EDIT RECORDS |\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

file.open("DATA.dat", ios::in);

if (!file) {

cout << "\x1b[33m\n\n\n\t\t\t\t\t File Does not Exit\n\n \033[0m";

cout << "\n\n\t\t\t\t\t ";

system("pause");

}

else

{

cout << "\n\n Enter User ID :: ";

cin >> id;

updatedFile.open("updatedData.dat", ios::app | ios::out);

file >> User\_ID >> UserFirst\_Name >> UserLast\_Name >> User\_Fname >> User\_Address >> User\_Pin >> User\_Password >> User\_Phone >> User\_balance >> AccountType;

while (!file.eof()) {

if (id == User\_ID) {

system("cls");

cout << "\x1b[36m\n\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t| EDIT RECORDS |\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

cout << "\n\n\n First Name : ";

cin >> N;

cout << "\n\n Last Name : ";

cin >> La;

cout << "\n\n Father Name : ";

cin >> Fn;

cout << "\n\n Permanent Address : ";

cin >> Add;

cout << "\n\n Pin Code (4 digit) : ";

cin >> Pi;

cout << "\n\n Password : ";

cin >> Pass;

cout << "\n\n Phone No. : ";

cin >> Ph;

back:

cout << "\n\n\t\tAccount Type";

cout << "\x1b[35m\n\n\t1. Platinum\033[0m \x1b[33m2. Gold \033[0m3. Silver\n\n Select One :: ";

cin >> ch;

switch (ch)

{

case 1: AT = "PLATINIUM"; break;

case 2: AT = "GOLD"; break;

case 3: AT = "SILVER"; break;

default:

system("cls");

cout << "\x1b[36m\n\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t| EDIT RECORDS |\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

cout << "\n\n\n First Name : " << N;

cout << "\n\n Last Name : " << La;

cout << "\n\n Father Name : " << Fn;

cout << "\n\n Permanent Address : " << Add;

cout << "\n\n Pin Code : " << Pi;

cout << "\n\n Password : " << Pass;

cout << "\n\n Phone No. : " << Ph;

goto back;

}

updatedFile << User\_ID << " " << N << " " << La << " " << Fn << " " << Add << " " << Pi << " " << Pass << " " << Ph << " " << User\_balance << " " << AT << "\n";

cout << "\n\n\t\t\t\t\t Record updated Successfully\n\n";

found++;

}

else

{

updatedFile << User\_ID << " " << UserFirst\_Name << " " << UserLast\_Name << " " << User\_Fname << " " << User\_Address << " " << User\_Pin << " " << User\_Password << " " << User\_Phone << " " << User\_balance << " " << AccountType << "\n";

}

file >> User\_ID >> UserFirst\_Name >> UserLast\_Name >> User\_Fname >> User\_Address >> User\_Pin >> User\_Password >> User\_Phone >> User\_balance >> AccountType;

}

file.close();

updatedFile.close();

if (remove("DATA.dat") == 0)

cout << ".";

if (rename("updatedData.dat", "DATA.dat") == 0)

cout << ".";

if (found == 0)

cout << "\x1b[33m\n\n\t\t\t\t\t Cannot find user record, Kindly check the ID or Record does not Exist.....\033[0m";

cout << "\n\n\t\t\t\t\t ";

system("pause");

}

}

void ADMIN::DeleteUserRecord()

{

fstream file, updatedFile;

system("cls");

string id;

int found = 0;

cout << "\x1b[36m\n\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t| DELETE RECORDS |\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

file.open("DATA.dat", ios::in);

if (!file) {

cout << "\x1b[33m\n\n\n\t\t\t\t\t File Does not Exit\n\n \033[0m";

cout << "\n\n\t\t\t\t\t ";

system("pause");

}

else

{

cout << "\n\n Enter User ID :: ";

cin >> id;

updatedFile.open("updatedData.dat", ios::app | ios::out);

file >> User\_ID >> UserFirst\_Name >> UserLast\_Name >> User\_Fname >> User\_Address >> User\_Pin >> User\_Password >> User\_Phone >> User\_balance >> AccountType;

while (!file.eof()) {

if (id == User\_ID) {

cout << "\n\n\t\t\t\t\t Record Deleted Successfully\n\n";

found++;

}

else

{

updatedFile << User\_ID << " " << UserFirst\_Name << " " << UserLast\_Name << " " << User\_Fname << " " << User\_Address << " " << User\_Pin << " " << User\_Password << " " << User\_Phone << " " << User\_balance << " " << AccountType << "\n";

}

file >> User\_ID >> UserFirst\_Name >> UserLast\_Name >> User\_Fname >> User\_Address >> User\_Pin >> User\_Password >> User\_Phone >> User\_balance >> AccountType;

}

file.close();

updatedFile.close();

if (remove("DATA.dat") == 0)

cout << ".";

if (rename("updatedData.dat", "DATA.dat") == 0)

cout << ".";

if (found == 0)

cout << "\x1b[33m\n\n\t\t\t\t\t Cannot find user record, Kindly check the ID or Record does not Exist.....\033[0m";

cout << "\n\n\t\t\t\t\t ";

system("pause");

}

}

void ADMIN::ShowAllRecords()

{

fstream file;

system("cls");

cout << "\x1b[36m\n\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t| DISPLAYING ALL RECORDS |\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

string id;

int found = 0;

file.open("DATA.dat", ios::in);

if (!file) {

cout << "\x1b[33m\n\n\n\t\t\t\t\t File Does not Exit\n\n \033[0m";

cout << "\n\n\t\t\t\t\t ";

system("pause");

}

else

{

file >> User\_ID >> UserFirst\_Name >> UserLast\_Name >> User\_Fname >> User\_Address >> User\_Pin >> User\_Password >> User\_Phone >> User\_balance >> AccountType;

while (!file.eof()) {

cout << "\n\n\n User ID :: " << User\_ID

<< "\n\n First Name :: " << UserFirst\_Name

<< "\n\n Last Name :: " << UserLast\_Name

<< "\n\n Father Name :: " << User\_Fname

<< "\n\n Permanent Address :: " << User\_Address

<< "\n\n Phone Number :: " << User\_Phone

<< "\n\n Balance :: " << User\_balance

<< "\n\n=================================================";

file >> User\_ID >> UserFirst\_Name >> UserLast\_Name >> User\_Fname >> User\_Address >> User\_Pin >> User\_Password >> User\_Phone >> User\_balance >> AccountType;

found++;

}

file.close();

if (found == 0)

cout << "\x1b[33m\n\n\t\t\t\t\t Database is Empty.....\033[0m";

cout << "\n\n\t\t\t\t\t ";

system("pause");

}

}

void ADMIN::PaymentAllRecords()

{

fstream file;

system("cls");

cout << "\x1b[36m\n\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t| BILLING RECORDS |\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

float BillAmount;

string ID, BillName, Date;

int found = 0;

file.open("bill.dat", ios::in);

if (!file) {

cout << "\x1b[33m\n\n\n\t\t\t\t\t File Does not Exit\n\n \033[0m";

cout << "\n\n\t\t\t\t\t ";

system("pause");

}

else

{

file >> ID >> BillName >> BillAmount >> Date;

while (!file.eof()) {

cout << "\n\n\n User ID :: " << ID

<< "\n\n Bill Name :: " << BillName

<< "\n\n Bill Amount :: " << BillAmount

<< "\n\n Payment Date :: " << Date

<< "\n\n=================================================";

file >> ID >> BillName >> BillAmount >> Date;

found++;

}

file.close();

if (found == 0)

cout << "\x1b[33m\n\n\t\t\t\t\t Database is Empty.....\033[0m";

cout << "\n\n\t\t\t\t\t ";

system("pause");

}

}

////////////////////////// User ///////////////////////

void USER::WithdrawAmount()

{

fstream file, updatedFile;

string id, pin;

char ch;

float withAmount, c = 0;

int Found = 0;

system("cls");

cout << "\x1b[36m\n\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t| WITHDRAW AMOUNT |\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

file.open("DATA.dat", ios::in);

if (!file) {

cout << "\x1b[33m\n\n\n\t\t\t\t\t File Does not Exit\n\n \033[0m";

cout << "\n\n\t\t\t\t\t ";

system("pause");

}

else

{

cout << "\n\n\n Your ID :: ";

cin >> id;

cin.ignore(); //clear buffer before taking new Line

cout << "\n\n Pin :: ";

while (ch = \_getch())

{

if (ch != '\r')

pin += ch;

else

break;

cout << "\*";

}

updatedFile.open("updatedData.dat", ios::app | ios::out);

file >> User\_ID >> UserFirst\_Name >> UserLast\_Name >> User\_Fname >> User\_Address >> User\_Pin >> User\_Password >> User\_Phone >> User\_balance >> AccountType;

while (!file.eof())

{

if ((id == User\_ID) && (pin == User\_Pin))

{

cout << "\n\n Withdraw Amount :: ";

cin >> withAmount;

if (withAmount <= User\_balance)

{

User\_balance -= withAmount;

updatedFile << User\_ID << " " << UserFirst\_Name << " " << UserLast\_Name << " " << User\_Fname << " " << User\_Address << " " << User\_Pin << " " << User\_Password << " " << User\_Phone << " " << User\_balance << " " << AccountType << "\n";

cout << "\n\n\tYour Amount \x1b[32m" << withAmount << "Rs \033[0m" << " Successfully Withdrawn";

cout << "\n\n\tCurrent Balance is :: \x1b[32m" << User\_balance << "Rs.\033[0m" << "\n\n";

}

else

{

updatedFile << User\_ID << " " << UserFirst\_Name << " " << UserLast\_Name << " " << User\_Fname << " " << User\_Address << " " << User\_Pin << " " << User\_Password << " " << User\_Phone << " " << User\_balance << " " << AccountType << "\n";

cout << "\n\n\tYour are " << withAmount - User\_balance << " credits Less .... ";

}

Found++;

}

else

{

updatedFile << User\_ID << " " << UserFirst\_Name << " " << UserLast\_Name << " " << User\_Fname << " " << User\_Address << " " << User\_Pin << " " << User\_Password << " " << User\_Phone << " " << User\_balance << " " << AccountType << "\n";

}

file >> User\_ID >> UserFirst\_Name >> UserLast\_Name >> User\_Fname >> User\_Address >> User\_Pin >> User\_Password >> User\_Phone >> User\_balance >> AccountType;

}

file.close();

updatedFile.close();

if (remove("DATA.dat") == 0)

cout << ".";

if (rename("updatedData.dat", "DATA.dat") == 0)

cout << ".";

if (Found == 0)

cout << "\x1b[33m\n\n\t\t\t\t\t Cannot find user record, Kindly check the ID or Record does not Exist.....\033[0m";

cout << "\\n\n\t\t\t\t\t ";

system("pause");

}

}

void USER::Transfer\_Option()

{

fstream file, updatedFile;

system("cls");

string SenderID, ReciverID;

float TransactionAmount, found = 0;

cout << "\x1b[36m\n\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t| TRANSFER AMOUNT |\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

file.open("DATA.dat", ios::in);

if (!file) {

cout << "\x1b[33m\n\n\n\t\t\t\t\t File Does not Exit\n\n \033[0m";

cout << "\n\n\t\t\t\t\t ";

system("pause");

}

else

{

cout << "\n\n\n Sender User ID for Transaction :: ";

cin >> SenderID;

cout << "\n\n Receiver User ID for Transaction :: ";

cin >> ReciverID;

cout << "\n\n Enter Transaction Amount :: ";

cin >> TransactionAmount;

file >> User\_ID >> UserFirst\_Name >> UserLast\_Name >> User\_Fname >> User\_Address >> User\_Pin >> User\_Password >> User\_Phone >> User\_balance >> AccountType;

// Checking IDs exists or not

while (!file.eof())

{

if ((SenderID == User\_ID) && (TransactionAmount <= User\_balance))

found++;

else if (ReciverID == User\_ID)

found++;

file >> User\_ID >> UserFirst\_Name >> UserLast\_Name >> User\_Fname >> User\_Address >> User\_Pin >> User\_Password >> User\_Phone >> User\_balance >> AccountType;

}

file.close();

// IF both User Exists

if (found == 2)

{

file.open("DATA.dat", ios::in);

updatedFile.open("updatedData.dat", ios::app | ios::out);

file >> User\_ID >> UserFirst\_Name >> UserLast\_Name >> User\_Fname >> User\_Address >> User\_Pin >> User\_Password >> User\_Phone >> User\_balance >> AccountType;

while (!file.eof())

{

if (SenderID == User\_ID)

{

User\_balance -= TransactionAmount;

updatedFile << User\_ID << " " << UserFirst\_Name << " " << UserLast\_Name << " " << User\_Fname << " " << User\_Address << " " << User\_Pin << " " << User\_Password << " " << User\_Phone << " " << User\_balance << " " << AccountType << "\n";

}

else if (ReciverID == User\_ID)

{

User\_balance += TransactionAmount;

updatedFile << User\_ID << " " << UserFirst\_Name << " " << UserLast\_Name << " " << User\_Fname << " " << User\_Address << " " << User\_Pin << " " << User\_Password << " " << User\_Phone << " " << User\_balance << " " << AccountType << "\n";

}

else

{

updatedFile << User\_ID << " " << UserFirst\_Name << " " << UserLast\_Name << " " << User\_Fname << " " << User\_Address << " " << User\_Pin << " " << User\_Password << " " << User\_Phone << " " << User\_balance << " " << AccountType << "\n";

}

file >> User\_ID >> UserFirst\_Name >> UserLast\_Name >> User\_Fname >> User\_Address >> User\_Pin >> User\_Password >> User\_Phone >> User\_balance >> AccountType;

}

file.close();

updatedFile.close();

if (remove("DATA.dat") == 0)

cout << ".";

if (rename("updatedData.dat", "DATA.dat") == 0)

cout << ".";

cout << "\n\n\t\t\t\t\t Receiver Recevied Payment Successfully";

}

else

{

if (found == 1)

cout << "\n\n\t\t\t\t\t Transaction Amount is greater than User's Current Balance";

else

cout << "\x1b[33m\n\n\t\t\t\t\t Cannot find User records, Kindly check the IDs or Record does not Exist.....\033[0m";

}

cout << "\n\n\t\t\t\t\t ";

system("pause");

}

}

void USER::Payment\_Option()

{

system("cls");

fstream file, updatedFile;

string id, BillName, pin;

float BillAmount, found = 0;

int ch;

SYSTEMTIME time;

cout << "\x1b[36m\n\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t| BILLS PAYMENT OPTIONS |\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

file.open("DATA.dat", ios::in);

if (!file)

{

cout << "\x1b[33m\n\n\n\t\t\t\t\t File Does not Exit\n\n \033[0m";

cout << "\n\n\t\t\t\t\t ";

system("pause");

}

else

{

Back:

system("cls");

cout << "\x1b[36m\n\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t| BILLS PAYMENT OPTIONS |\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

cout << "\n\n Select One of the Following :: ";

cout << "\x1b[33m\n\n 1. Utility";

cout << "\n\n 2. TELCO";

cout << "\n\n 3. STUDENT FEE";

cout << "\n\n 4. CREDIT CARD";

cout << "\n\n 5. INTERNET BILLS";

cout << "\n\n 6. INTERNET SHOPPING";

cout << "\n\n 7. Govt. PAYMENTS";

cout << "\n\n 8. AIR FARE";

cout << "\n\n 9. INSURANCE";

cout << "\n\n 10. INVESTMENTS";

cout << "\n\n 11. elPO";

cout << "\n\n 12. ELECTRICITY";

cout << "\n\n 13. GAS BILLS\033[0m";

cout << "\n\n Select One :: ";

cin >> ch;

switch (ch)

{

case 1:BillName = "UTILITY"; break;

case 2:BillName = "TELCO"; break;

case 3:BillName = "STUDENT FEE"; break;

case 4:BillName = "CREDIT CARD"; break;

case 5:BillName = "INTERNET BILLS"; break;

case 6:BillName = "INTERNET SHOPPING"; break;

case 7:BillName = "GOVT PAYMENTS"; break;

case 8:BillName = "AIR FARE"; break;

case 9:BillName = "INSURANCE"; break;

case 10:BillName = "INVESTMENTS"; break;

case 11:BillName = "elPO"; break;

case 12:BillName = "ELECTRICITY"; break;

case 13:BillName = "GAS BILLS"; break;

default:

cout << "\n\n\t\t\t\t\t Invalid Choice ... ";

cout << "\n\n\t\t\t\t\t ";

system("pause");

goto Back;

}

system("cls");

cout << "\x1b[36m\n\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t| BILLS PAYMENT OPTIONS |\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

cout << "\n\n Your ID :: ";

cin >> id;

cin.ignore(); //clear buffer before taking new Line

cout << "\n\n Pin :: ";

while (ch = \_getch())

{

if (ch != '\r')

pin += ch;

else

break;

cout << "\*";

}

cout << "\n\n Bill Amount :: ";

cin >> BillAmount;

updatedFile.open("updatedData.dat", ios::app | ios::out);

file >> User\_ID >> UserFirst\_Name >> UserLast\_Name >> User\_Fname >> User\_Address >> User\_Pin >> User\_Password >> User\_Phone >> User\_balance >> AccountType;

while (!file.eof())

{

if ((id == User\_ID) && (BillAmount <= User\_balance) && (User\_Pin == pin))

{

User\_balance -= BillAmount;

updatedFile << User\_ID << " " << UserFirst\_Name << " " << UserLast\_Name << " " << User\_Fname << " " << User\_Address << " " << User\_Pin << " " << User\_Password << " " << User\_Phone << " " << User\_balance << " " << AccountType << "\n";

found++;

}

else

updatedFile << User\_ID << " " << UserFirst\_Name << " " << UserLast\_Name << " " << User\_Fname << " " << User\_Address << " " << User\_Pin << " " << User\_Password << " " << User\_Phone << " " << User\_balance << " " << AccountType << "\n";

file >> User\_ID >> UserFirst\_Name >> UserLast\_Name >> User\_Fname >> User\_Address >> User\_Pin >> User\_Password >> User\_Phone >> User\_balance >> AccountType;

}

file.close();

updatedFile.close();

if (remove("DATA.dat") == 0)

cout << ".";

if (rename("updatedData.dat", "DATA.dat") == 0)

cout << ".";

if (found == 1)

{

GetSystemTime(&time);

file.open("bill.dat", ios::app | ios::out);

file << id << " " << BillName << " " << BillAmount << " " << time.wDay << "/" << time.wMonth << "/" << time.wYear << "\n";

file.close();

cout << "\n\n\n\t\t\t\t\t " << BillName << " Bill Paid Successfully";

}

else

cout << "\x1b[33m\n\n\n\t\t\t\t\t User ID is Invalid or Amount Exceeds the Total Balance\033[0m";

cout << "\n\n\t\t\t\t\t ";

system("pause");

}

}

void USER::CheckBalance()

{

system("cls");

cout << "\x1b[36m\n\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t| USER LOGIN |\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

fstream file;

int found = 0;

string id, pin;

char ch, c = 0;

file.open("DATA.dat", ios::in);

if (!file)

{

cout << "\x1b[33m\n\n\n\t\t\t\t\t File Does not Exit\n\n \033[0m";

cout << "\n\n\t\t\t\t\t ";

system("pause");

}

else

{

cout << "\n\n Your ID :: ";

cin >> id;

cin.ignore(); //clear buffer before taking new Line

cout << "\n\n Pin :: ";

while (ch = \_getch())

{

if (ch != '\r')

pin += ch;

else

break;

cout << "\*";

}

file >> User\_ID >> UserFirst\_Name >> UserLast\_Name >> User\_Fname >> User\_Address >> User\_Pin >> User\_Password >> User\_Phone >> User\_balance >> AccountType;

while (!file.eof())

{

if ((id == User\_ID) && (pin == User\_Pin))

{

cout << "\n\n\t Your Current Balance is :: \x1b[32m" << User\_balance << "Rs.\033[0m";

cout << "\n\n ";

system("pause");

found++;

break;

}

file >> User\_ID >> UserFirst\_Name >> UserLast\_Name >> User\_Fname >> User\_Address >> User\_Pin >> User\_Password >> User\_Phone >> User\_balance >> AccountType;

}

file.close();

if (found == 0)

{

cout << "\x1b[33m\n\n\t\t\t\t\t Your credentials are incorrect or have expired. Please try again or reset your password...\n\n \033[0m";

cout << "\n\n\t\t\t\t\t ";

system("pause");

}

}

}

void USER::AccountDetails()

{

fstream file;

string id, pin;

char ch;

float c = 0;

int Found = 0;

system("cls");

cout << "\x1b[36m\n\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t| ACCOUNT DETAILS |\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

file.open("DATA.dat", ios::in);

if (!file) {

cout << "\x1b[33m\n\n\n\t\t\t\t\t File Does not Exit\n\n \033[0m";

cout << "\n\n\t\t\t\t\t ";

system("pause");

}

else

{

cout << "\n\n Your ID :: ";

cin >> id;

cin.ignore(); //clear buffer before taking new Line

cout << "\n\n Pin :: ";

while (ch = \_getch())

{

if (ch != '\r')

pin += ch;

else

break;

cout << "\*";

}

file >> User\_ID >> UserFirst\_Name >> UserLast\_Name >> User\_Fname >> User\_Address >> User\_Pin >> User\_Password >> User\_Phone >> User\_balance >> AccountType;

while (!file.eof())

{

if ((id == User\_ID) && (pin == User\_Pin))

{

system("cls");

cout << "\x1b[36m\n\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t| ACCOUNT DETAILS |\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

cout << "\n\n\n User ID :: " << User\_ID

<< "\n\n First Name :: " << UserFirst\_Name

<< "\n\n Last Name :: " << UserLast\_Name

<< "\n\n Password :: " << User\_Password

<< "\n\n Father Name :: " << User\_Fname

<< "\n\n Permanent Address :: " << User\_Address

<< "\n\n Phone Number :: " << User\_Phone

<< "\n\n Balance :: " << User\_balance

<< "\n\n Account Type :: " << AccountType

<< "\n\n=================================================";

Found++;

}

file >> User\_ID >> UserFirst\_Name >> UserLast\_Name >> User\_Fname >> User\_Address >> User\_Pin >> User\_Password >> User\_Phone >> User\_balance >> AccountType;

}

file.close();

if (Found == 0)

cout << "\x1b[33m\n\n\t\t\t\t\t Cannot find user record, Kindly check the ID or Record does not Exist.....\033[0m";

cout << "\n\n\t\t\t\t\t ";

system("pause");

}

}

void USER::ApplyForLoan()

{

fstream file;

string id, pin;

char ch;

float c = 0, Amount, MonthlyReturn = 0, Tenure, InterestRate = 0.08;

int Found = 0, choice;

system("cls");

cout << "\x1b[36m\n\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t| APPLY FOR LOANS |\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

file.open("DATA.dat", ios::in);

if (!file) {

cout << "\x1b[33m\n\n\n\t\t\t\t\t File Does not Exit\n\n \033[0m";

cout << "\n\n\t\t\t\t\t ";

system("pause");

}

else

{

cout << "\n\n Your ID :: ";

cin >> id;

cin.ignore(); //clear buffer before taking new Line

cout << "\n\n Pin :: ";

while (ch = \_getch())

{

if (ch != '\r')

pin += ch;

else

break;

cout << "\*";

}

file >> User\_ID >> UserFirst\_Name >> UserLast\_Name >> User\_Fname >> User\_Address >> User\_Pin >> User\_Password >> User\_Phone >> User\_balance >> AccountType;

while (!file.eof())

{

if ((id == User\_ID) && (pin == User\_Pin))

{

Back:

system("cls");

cout << "\x1b[36m\n\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t| APPLY FOR LOANS |\033[0m";

cout << "\x1b[36m\n\t\t\t\t\t|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\033[0m";

cout << "\n\n\n Select What Best Describes You\n";

cout << "\x1b[35m\n 1. Salaried Individual\033[0m";

cout << "\x1b[32m\n 2. Self-Employed\033[0m";

cout << "\n\n Select One :: ";

cin >> choice;

cout << "\n\n Amount For Loan :: ";

cin >> Amount;

cout << "\n Tenure of Loan (In Months) :: ";

cin >> Tenure;

switch (choice)

{

case 1:InterestRate = 0.06;

Amount += (Amount \* InterestRate);

MonthlyReturn = Amount / Tenure;

break;

case 2:Amount += (Amount \* InterestRate);

MonthlyReturn = Amount / Tenure;

break;

default:

cout << "\x1b[33m\n\n\t\t\t\t\t Invalid Choice ... \033[0m";

cout << "\n\n\t\t\t\t\t ";

system("pause");

goto Back;

}

cout << "\n\n\t\t\033[3;47;35m \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \033[0m";

cout << "\n\t\t\033[3;47;35m | | | \033[0m";

cout << "\n\t\t\033[3;47;35m | DESCRIPTION | INSTALLMENT AMOUNT | \033[0m";

cout << "\n\t\t\033[3;47;35m |\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_| \033[0m";

cout << "\n\t\t\033[3;47;35m | | | \033[0m";

cout << "\n\t\t\033[3;47;35m | ESTIMATED MONTHLY INSTALLMENT | " << MonthlyReturn << " | \033[0m";

cout << "\n\t\t\033[3;47;35m |\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_| ";

cout << "\n\t\t\033[3;47;35m \033[0m";

Found++;

break;

}

file >> User\_ID >> UserFirst\_Name >> UserLast\_Name >> User\_Fname >> User\_Address >> User\_Pin >> User\_Password >> User\_Phone >> User\_balance >> AccountType;

}

file.close();

if (Found == 0)

cout << "\x1b[33m\n\n\t\t\t\t\t Cannot find user record, Kindly check the ID or Record does not Exist.....\033[0m";

cout << "\n\n\t\t\t\t\t ";

system("pause");

}

}

int main()

{

for (int i = 0; i <= 10; i++)

{

cout << "\n\n\n\n\t\t\t\t\t \x1b\033[3;47;35mWELCOME TO DIGiBANK \033[0m\033[0m\n";

cout << "\n\n\n\n\n\n\t\t\t\t\t\t|===============|";

cout << "\n\t\t\t\t\t\t| LOADING. |";

cout << "\n\t\t\t\t\t\t|===============|";

system("cls");

cout << "\n\n\n\n\t\t\t\t\t \x1b\033[3;47;35mWELCOME TO DIGiBANK \033[0m\033[0m\n";

cout << "\n\n\n\n\n\n\t\t\t\t\t\t|===============|";

cout << "\n\t\t\t\t\t\t| LOADING.. |";

cout << "\n\t\t\t\t\t\t|===============|";

system("cls");

cout << "\n\n\n\n\t\t\t\t\t \x1b\033[3;47;35mWELCOME TO DIGiBANK \033[0m\033[0m\n";

cout << "\n\n\n\n\n\n\t\t\t\t\t\t|===============|";

cout << "\n\t\t\t\t\t\t| LOADING... |";

cout << "\n\t\t\t\t\t\t|===============|";

system("cls");

/\*cout << "\x1b[36m\n\n\n\n\n\n\t\t\t\t\t ---------------\n";

cout << "\t\t\t\t\t | LOADING. |\n";

cout << "\t\t\t\t\t ---------------\n";

system("cls");

cout << "\x1b[36m\n\n\n\n\n\n\t\t\t\t\t ----------------\n";

cout << "\t\t\t\t\t | LOADING.. |\n";

cout << "\t\t\t\t\t ---------------\n";

system("cls");

cout << "\x1b[36m\n\n\n\n\n\n\t\t\t\t\t -----------------\n";

cout << "\t\t\t\t\t | LOADING... |\n";

cout << "\t\t\t\t\t ---------------\n";

system("cls");\*/

}

//cout << "\n\n\n\n\n\n\t\t\t\t\t \x1b[35m WELCOME TO D.G.BANK\033[0m\n";

//cout << "\n\t\t\t\t\t\x1b[35mOUR MOTTO :: FAST AND SECURE\033[0m\n";

//cout << "\n\n\n\n\n\n\n\n";

cout << "\n\n\n\n\t\t\t\t\t \x1b\033[3;47;35mWELCOME TO DIGiBANK \033[0m\033[0m\n";

cout << "\n\n\n\n\n\n\t\t\t\t\t\t|==============|";

cout << "\n\t\t\t\t\t\t| Completed |";

cout << "\n\t\t\t\t\t\t|==============|";

cout << "\n\n\n\n\t\t\t\t\t ";

system("pause");

system("cls");

BANK ATM;

ATM.Mainmenu();

return 0;

}