

Bank MAnagement System

[Document subtitle]

**22\_NTU\_CS\_1256 Ayesha Aslam**

**June 12, 2024**

**Submitted to Dr Hamid**

#include <iostream>

#include <string>

using namespace std;

class Node {

private:

int Index;

float Cash;

string Phone;

string Email;

string Name;

string LastName;

int Id;

Node\* NextNode;

public:

Node(int accountId) {

Cash = 0;

Id = accountId;

NextNode = nullptr;

Index = 0;

}

int getVal() { return Index; }

float getCash() { return Cash; }

string getPhn() { return Phone; }

string getEmail() { return Email; }

string getName() { return Name; }

string getLname() { return LastName; }

int getId() { return Id; }

Node\* getNext() { return NextNode; }

void setVal(int v) { Index = v; }

void setCash(float c) { Cash = c; }

void setPhn(const string& p) { Phone = p; }

void setEmail(const string& e) { Email = e; }

void setName(const string& n) { Name = n; }

void setLname(const string& ln) { LastName = ln; }

void setId(int i) { Id = i; }

void setNext(Node\* n) { NextNode = n; }

};

class bank {

private:

Node\* First;

Node\* Last;

int length;

public:

bank() {

length = 0;

First = nullptr;

Last = nullptr;

}

void clearScreen();

void openAccount();

void displayDetails();

void deleteAccount();

void deposite();

void withdrawal();

void update();

void search();

void listing();

};

void bank::clearScreen() {

system("cls");

}

void bank::openAccount() {

clearScreen();

float ca;

string ph;

int ch1 = 1, ID;

string na;

string lna;

string em;

while (ch1 == 1) {

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

cout << "\n\t\t\t\t\*\tEnter name:\t\t\t";

cin >> na;

cout << "\t\t\t\t\*\tEnter last name:\t\t";

cin >> lna;

cout << "\t\t\t\t\*\tEnter phone number:\t\t";

cin >> ph;

cout << "\t\t\t\t\*\tEnter email id:\t\t\t";

cin >> em;

cout << "\t\t\t\t\*\tEnter account id:\t\t";

cin >> ID;

cout << "\t\t\t\t\*\tEnter expected cash (in Rupees):";

cin >> ca;

Node\* newNode = new Node(ID);

newNode->setVal(length + 1);

newNode->setCash(ca);

newNode->setName(na);

newNode->setLname(lna);

newNode->setPhn(ph);

newNode->setEmail(em);

if (First == nullptr) {

First = newNode;

Last = newNode;

}

else {

Last->setNext(newNode);

Last = newNode;

}

length++;

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

cout << "\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

cout << "\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\tAccount created successfully!!!\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*";

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

cout << "Press 1 to create another account or 2 to return to menu: ";

cin >> ch1;

if (ch1 == 1) {

clearScreen();

}

}

}

void bank::displayDetails() {

clearScreen();

Node\* p = First;

if (First == nullptr) {

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

cout << "\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

cout << "\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\tNo data found to display !!\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*";

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

}

else {

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

cout << "\n\t\t\t\*\*\*\*\*\*\*\*\*\* Bank Account Management System \*\*\*\*\*\*\*\*\*\*\n";

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

while (p != nullptr) {

cout << "\n\t\t\t\*\t\tAccount Holder\t\t\t" << p->getVal() << "\t\t\t\*\n\t\t\t\*\t\tLast Name:\t\t\t" << p->getLname() << "\t\t\t\*\n\t\t\t\*\t\tName:\t\t\t\t" << p->getName() << "\t\t\t\*\n\t\t\t\*\t\tPhone number:\t\t\t" << p->getPhn() << "\t\t\*\n\t\t\t\*\t\tAccount ID:\t\t\t" << p->getId() << "\t\t\t\*\n\t\t\t\*\t\tEmail ID:\t\t\t" << p->getEmail() << "\t\*\n\t\t\t\*\t\tCash in account: Rs.\t\t" << p->getCash() << "\t\t\t\*\n\t\t\t\*\t\tTotal Account Holders:\t\t" << length << "\t\t\t\*\n";

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

p = p->getNext();

}

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

}

cout << "Press any key to return to menu...";

cin.ignore();

cin.get();

}

void bank::deleteAccount() {

clearScreen();

int ID;

cout << "Enter the account ID to delete: ";

cin >> ID;

Node\* temp = First;

Node\* prev = nullptr;

if (temp != nullptr && temp->getId() == ID) {

First = temp->getNext();

delete temp;

length--;

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

cout << "\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

cout << "\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\tAccount deleted successfully!!!\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*";

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

return;

}

while (temp != nullptr && temp->getId() != ID) {

prev = temp;

temp = temp->getNext();

}

if (temp == nullptr) {

cout << "\nAccount ID not found!!!\n";

return;

}

prev->setNext(temp->getNext());

delete temp;

length--;

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

cout << "\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

cout << "\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\tAccount deleted successfully!!!\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*";

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

cout << "Press any key to return to menu...";

cin.ignore();

cin.get();

}

void bank::deposite() {

clearScreen();

int ID;

cout << "Enter account ID: ";

cin >> ID;

Node\* current = First;

bool found = false;

while (current != nullptr) {

if (current->getId() == ID) {

found = true;

break;

}

current = current->getNext();

}

if (found) {

float amt;

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

cout << "\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

cout << "\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\tEnter amount you want to deposit:\t\*";

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

cin >> amt;

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

current->setCash(current->getCash() + amt);

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

cout << "\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

cout << "\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\tCash deposited successfully!!!\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*";

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

}

else {

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

cout << "\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

cout << "\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\tAccount Id not found!!!\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*";

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

}

cout << "Press any key to return to menu...";

cin.ignore();

cin.get();

}

void bank::withdrawal() {

clearScreen();

int ID;

cout << "Enter account ID: ";

cin >> ID;

Node\* current = First;

bool found = false;

while (current != nullptr) {

if (current->getId() == ID) {

found = true;

break;

}

current = current->getNext();

}

if (found) {

float amt;

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

cout << "\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

cout << "\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\tEnter amount you want to withdraw:\t\*";

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

cin >> amt;

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

if (amt > current->getCash()) {

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

cout << "\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

cout << "\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\tInsufficient balance!!!\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*";

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

}

else {

current->setCash(current->getCash() - amt);

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

cout << "\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

cout << "\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\tCash withdrawn successfully!!!\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*";

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

}

}

else {

cout << "\nAccount ID not found!!!\n";

}

cout << "Press any key to return to menu...";

cin.ignore();

cin.get();

}

void bank::update() {

clearScreen();

int ID;

cout << "Enter account ID to update: ";

cin >> ID;

Node\* current = First;

bool found = false;

while (current != nullptr) {

if (current->getId() == ID) {

found = true;

break;

}

current = current->getNext();

}

if (found) {

float cash;

string phone, email, name, lastName;

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

cout << "\n\t\t\t\t\*\tEnter name:\t\t\t";

cin >> name;

cout << "\t\t\t\t\*\tEnter last name:\t\t";

cin >> lastName;

cout << "\t\t\t\t\*\tEnter phone number:\t\t";

cin >> phone;

cout << "\t\t\t\t\*\tEnter email id:\t\t\t";

cin >> email;

cout << "\t\t\t\t\*\tEnter expected cash (in Rupees):";

cin >> cash;

current->setName(name);

current->setLname(lastName);

current->setPhn(phone);

current->setEmail(email);

current->setCash(cash);

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

cout << "\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

cout << "\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\tAccount updated successfully!!!\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*";

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

}

else {

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

cout << "\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

cout << "\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\tAccount Id not found!!!\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*";

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

}

cout << "Press any key to return to menu...";

cin.ignore();

cin.get();

}

void bank::search() {

clearScreen();

int ID;

cout << "Enter account ID to search: ";

cin >> ID;

Node\* current = First;

bool found = false;

while (current != nullptr) {

if (current->getId() == ID) {

found = true;

break;

}

current = current->getNext();

}

if (found) {

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

cout << "\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

cout << "\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\tAccount Details:\t\t\*\n";

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

cout << "\t\t\t\*\t\tName: " << current->getName() << "\t\t\t\*\n";

cout << "\t\t\t\*\t\tLast Name: " << current->getLname() << "\t\t\*\n";

cout << "\t\t\t\*\t\tPhone: " << current->getPhn() << "\t\t\*\n";

cout << "\t\t\t\*\t\tEmail: " << current->getEmail() << "\t\*\n";

cout << "\t\t\t\*\t\tCash: Rs. " << current->getCash() << "\t\t\t\*\n";

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

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

}

else {

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

cout << "\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

cout << "\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\tAccount Id not found!!!\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*";

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

}

cout << "Press any key to return to menu...";

cin.ignore();

cin.get();

}

void bank::listing() {

clearScreen();

Node\* p = First;

if (First == nullptr) {

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

cout << "\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

cout << "\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\tNo data found to display !!\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*\n\t\t\t\*\t\t\t\t\t\t\*";

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

}

else {

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

cout << "\n\t\t\t\*\*\*\*\*\*\*\*\*\* Bank Account Management System \*\*\*\*\*\*\*\*\*\*\n";

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

while (p != nullptr) {

cout << "\n\t\t\t\*\t\tAccount Holder\t\t\t" << p->getVal() << "\t\t\t\*\n\t\t\t\*\t\tLast Name:\t\t\t" << p->getLname() << "\t\t\t\*\n\t\t\t\*\t\tName:\t\t\t\t" << p->getName() << "\t\t\t\*\n\t\t\t\*\t\tPhone number:\t\t\t" << p->getPhn() << "\t\t\*\n\t\t\t\*\t\tAccount ID:\t\t\t" << p->getId() << "\t\t\t\*\n\t\t\t\*\t\tEmail ID:\t\t\t" << p->getEmail() << "\t\*\n\t\t\t\*\t\tCash in account: Rs.\t\t" << p->getCash() << "\t\t\t\*\n\t\t\t\*\t\tTotal Account Holders:\t\t" << length << "\t\t\t\*\n";

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

p = p->getNext();

}

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

}

cout << "Press any key to return to menu...";

cin.ignore();

cin.get();

}

int main() {

bank myBank;

int choice;

while (true) {

myBank.clearScreen();

cout << "\n\t\t\t\*\*\*\*\*\*\*\*\*\* Bank Account Management System \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n\n";

cout << "\t\t\t\*\t\t1. Open a new account\t\t\t\t\*\n";

cout << "\t\t\t\*\t\t2. Display details of all account holders\t\*\n";

cout << "\t\t\t\*\t\t3. Delete an account\t\t\t\t\*\n";

cout << "\t\t\t\*\t\t4. Deposit\t\t\t\t\t\*\n";

cout << "\t\t\t\*\t\t5. Withdraw\t\t\t\t\t\*\n";

cout << "\t\t\t\*\t\t6. Update account details\t\t\t\*\n";

cout << "\t\t\t\*\t\t7. Search an account\t\t\t\t\*\n";

cout << "\t\t\t\*\t\t8. List all accounts\t\t\t\t\*\n";

cout << "\t\t\t\*\t\t9. Exit\t\t\t\t\t\t\*\n";

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

cout << "============>> Enter your choice: ";

cin >> choice;

switch (choice) {

case 1:

myBank.openAccount();

break;

case 2:

myBank.displayDetails();

break;

case 3:

myBank.deleteAccount();

break;

case 4:

myBank.deposite();

break;

case 5:

myBank.withdrawal();

break;

case 6:

myBank.update();

break;

case 7:

myBank.search();

break;

case 8:

myBank.listing();

break;

case 9:

cout << "Exiting...\n";

return 0;

default:

cout << "Invalid choice. Please try again.\n";

break;

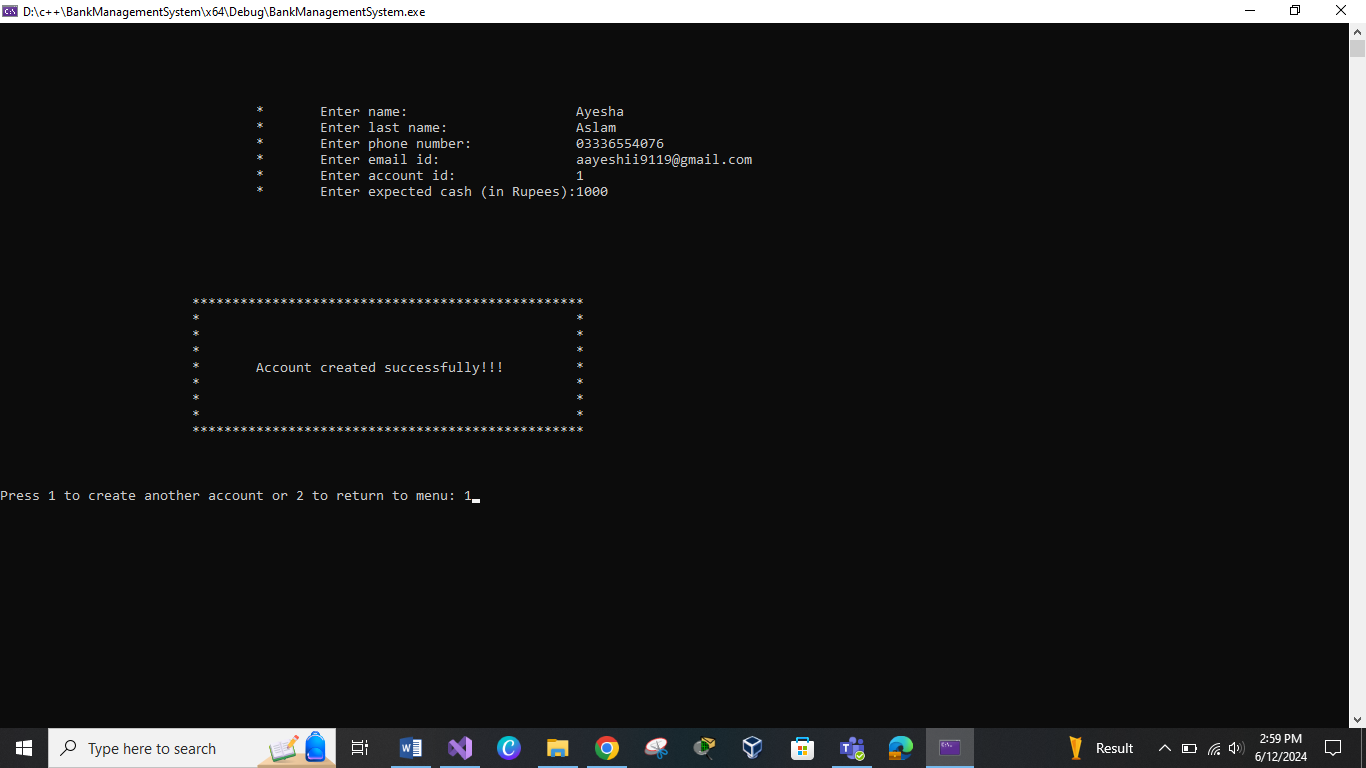
}

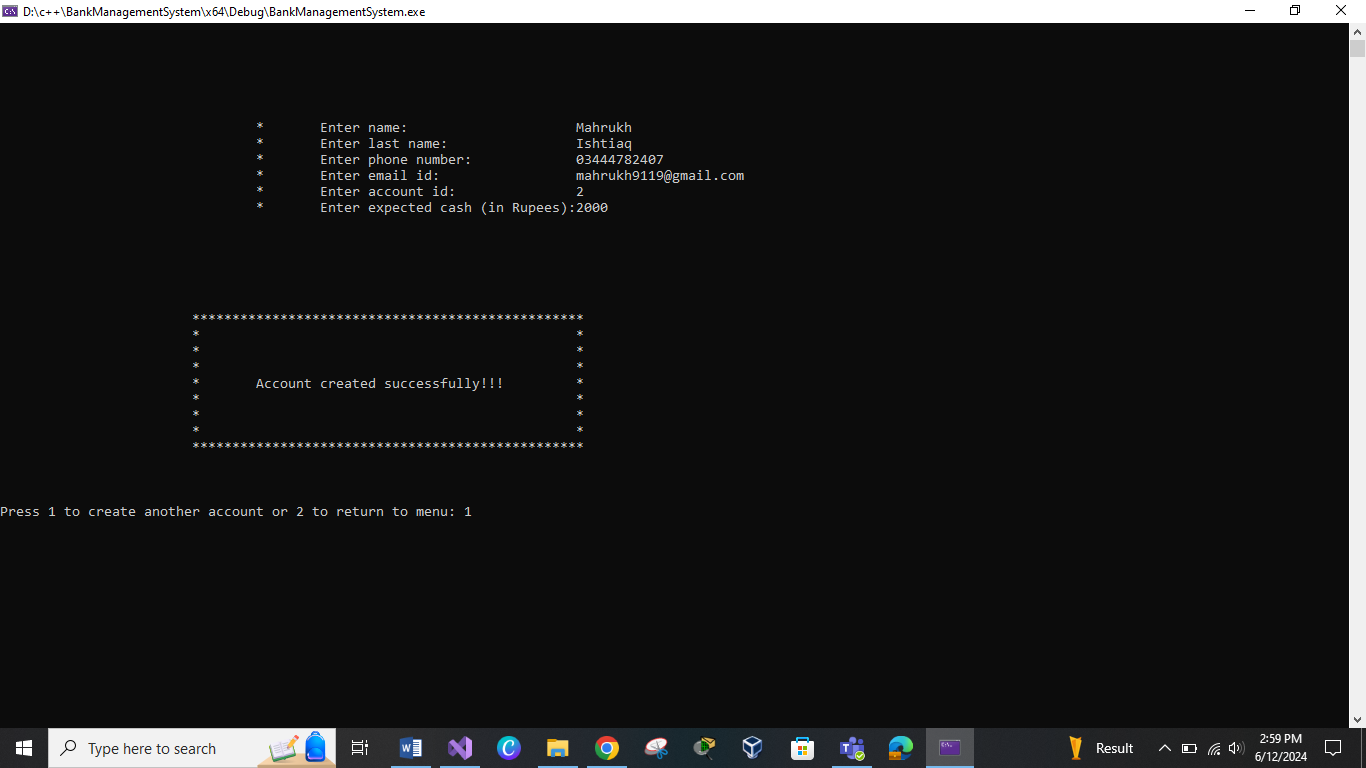
}

return 0;

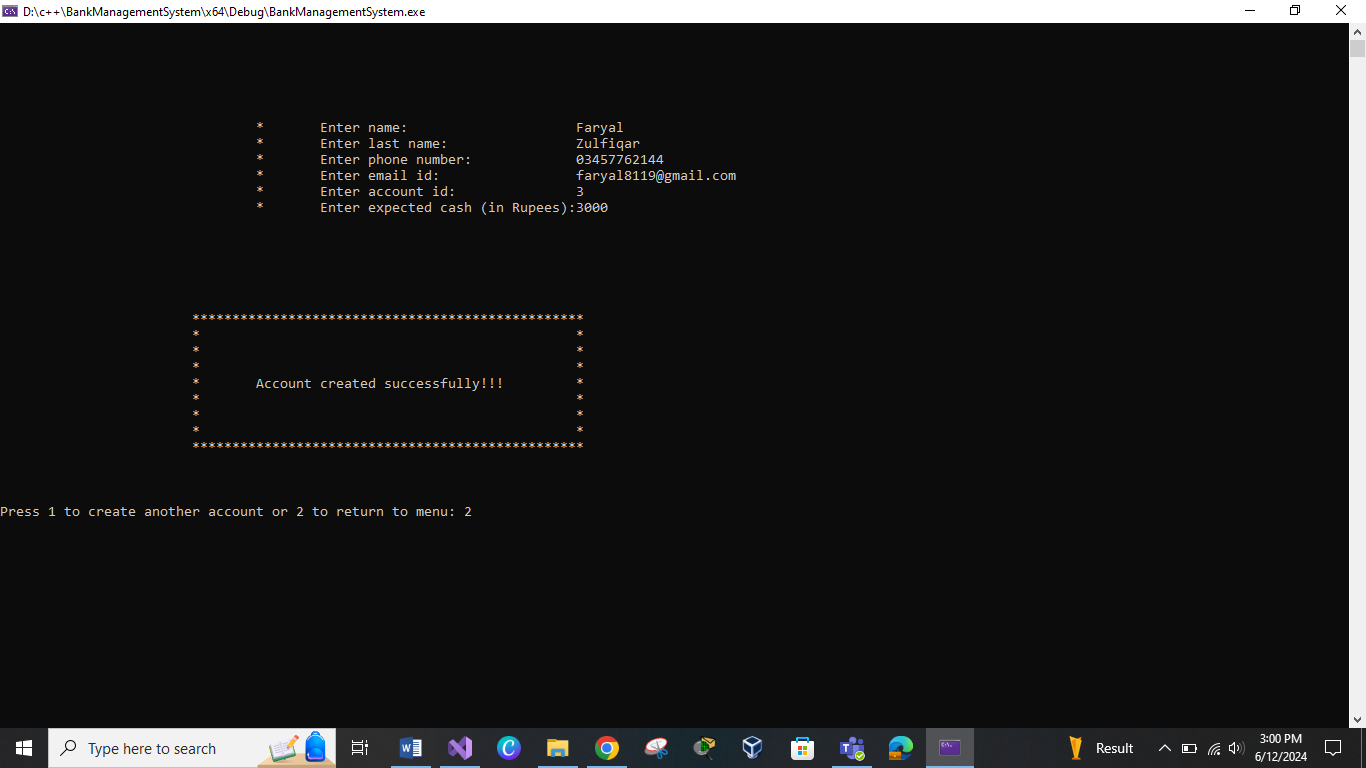
}

**OutPuts:  
Functionality Number 1:**  
To Open or create an account for customer the Open account will work as:

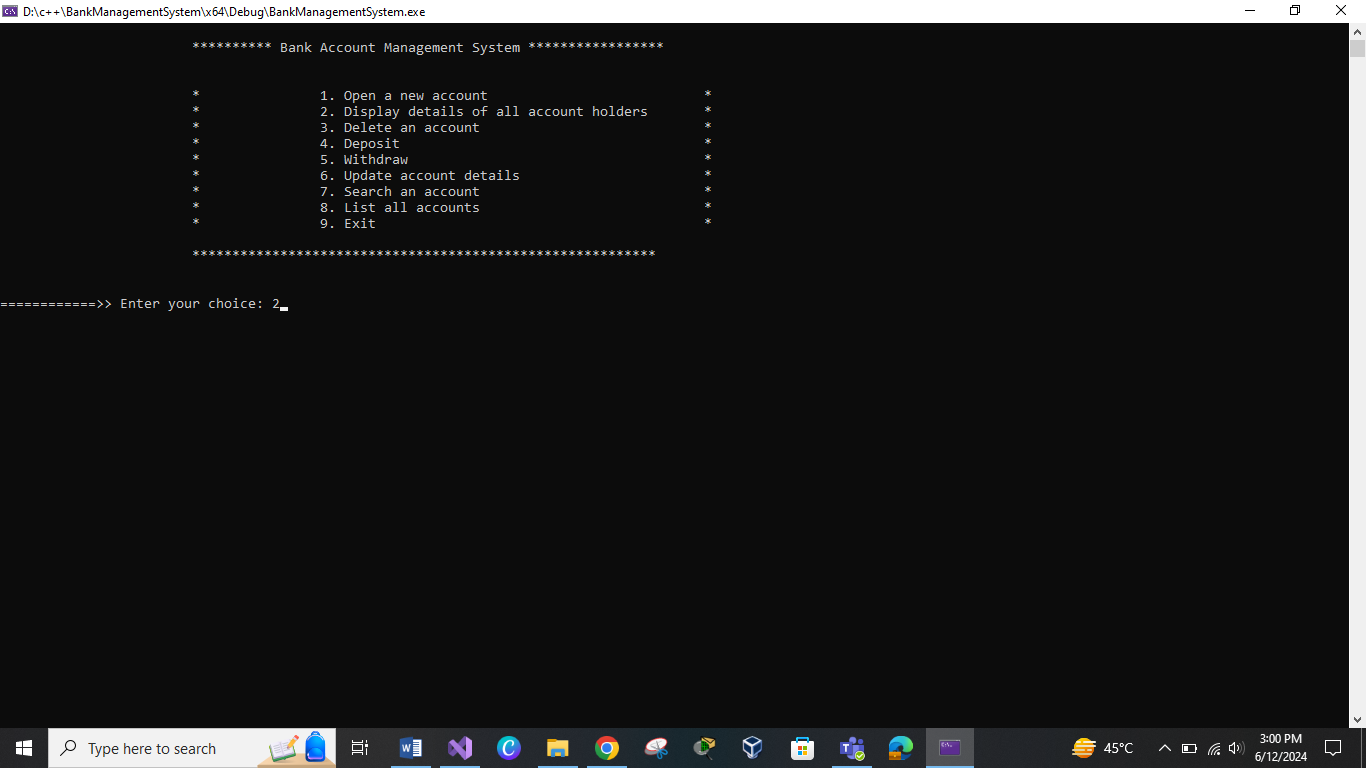


**Similarly to create an account for the 2nd person we will press 1:**  
****

**Similarly we can add third person**

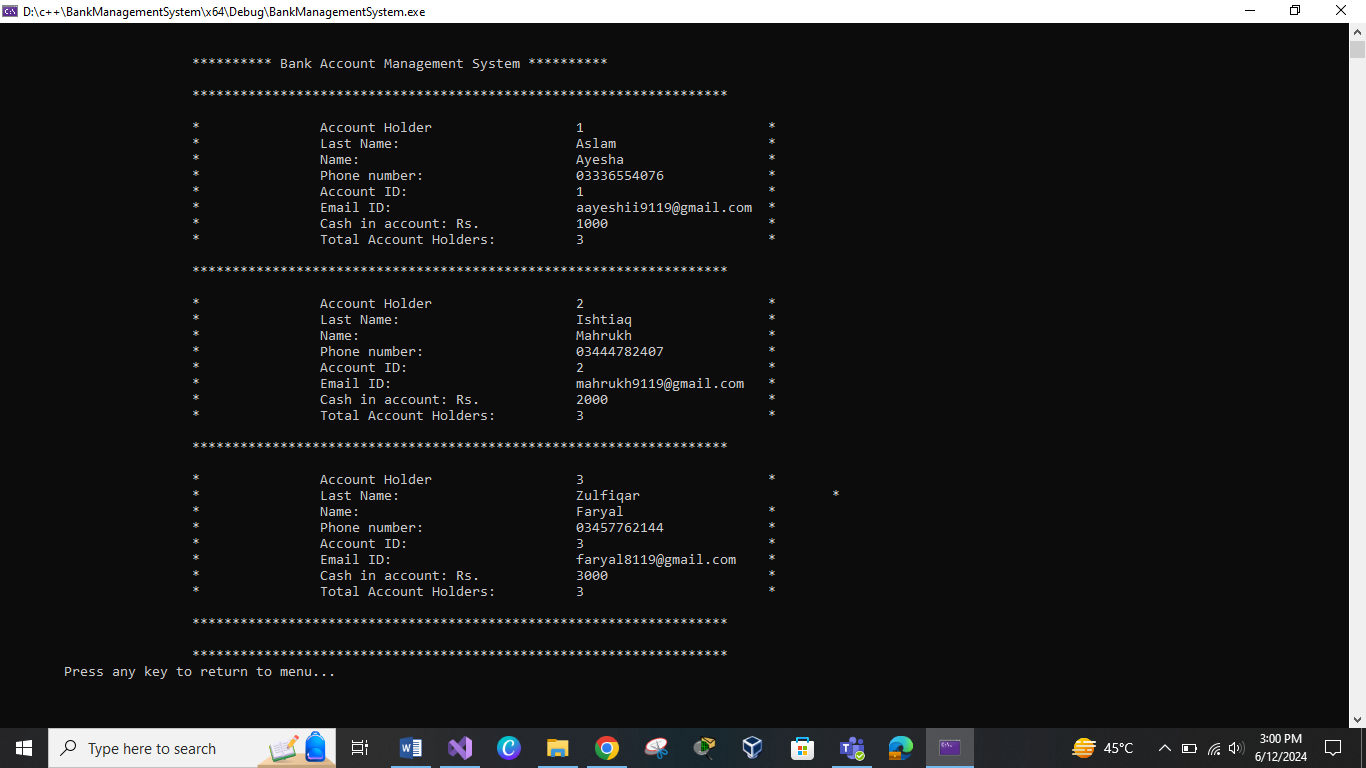


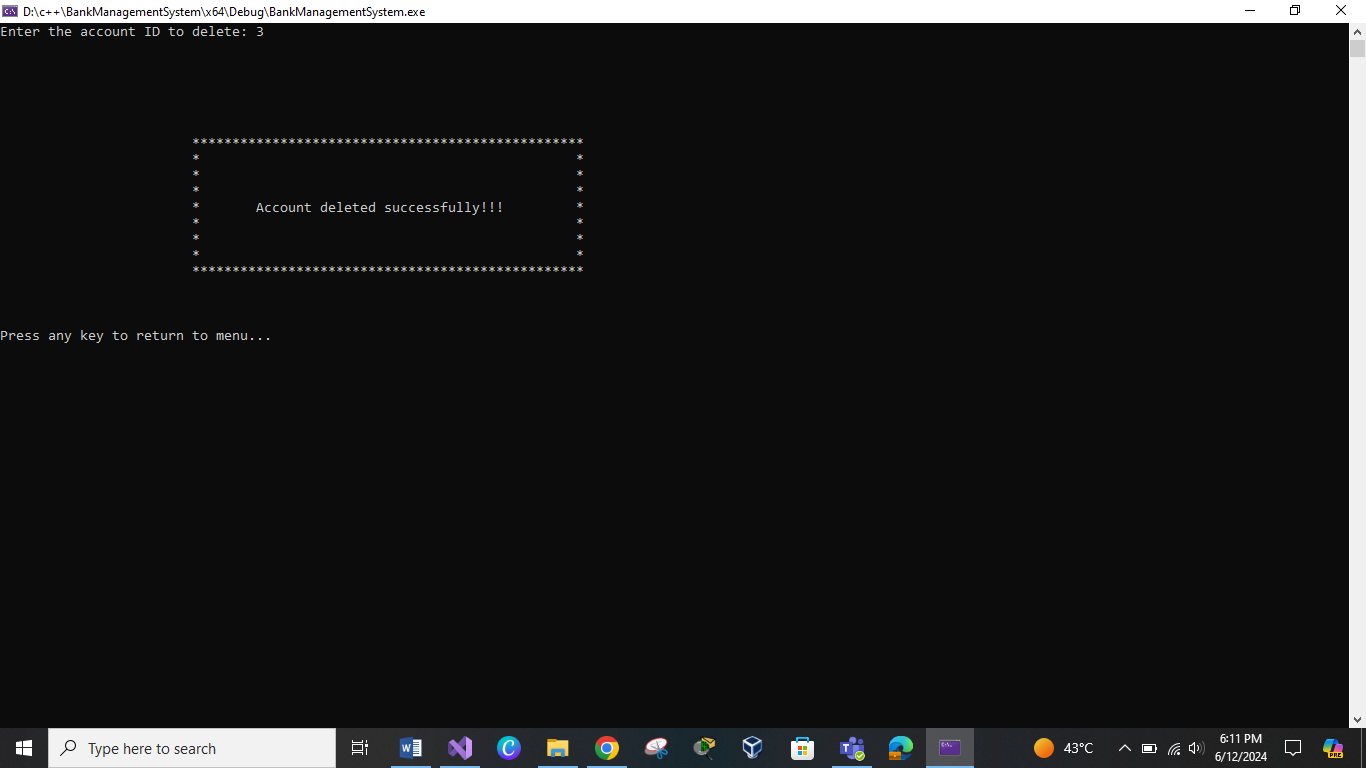
**Working:**  
The Opening of account uses linked list beacuase we don’t know how many accounts will be opened by customers and usig of aarays here will not be efficient.

When we will press 2 it will show a menue of operations which a user can perfom using this bank management system  
Like:  


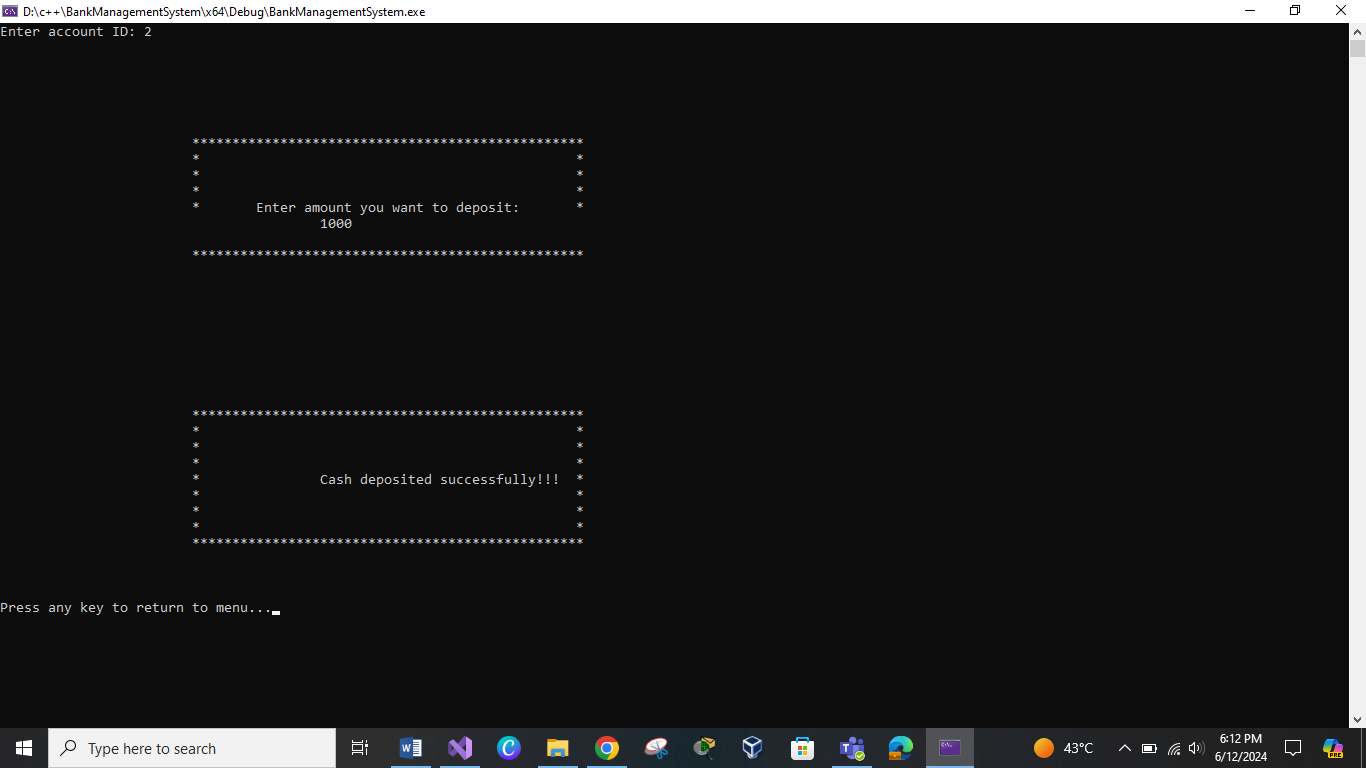
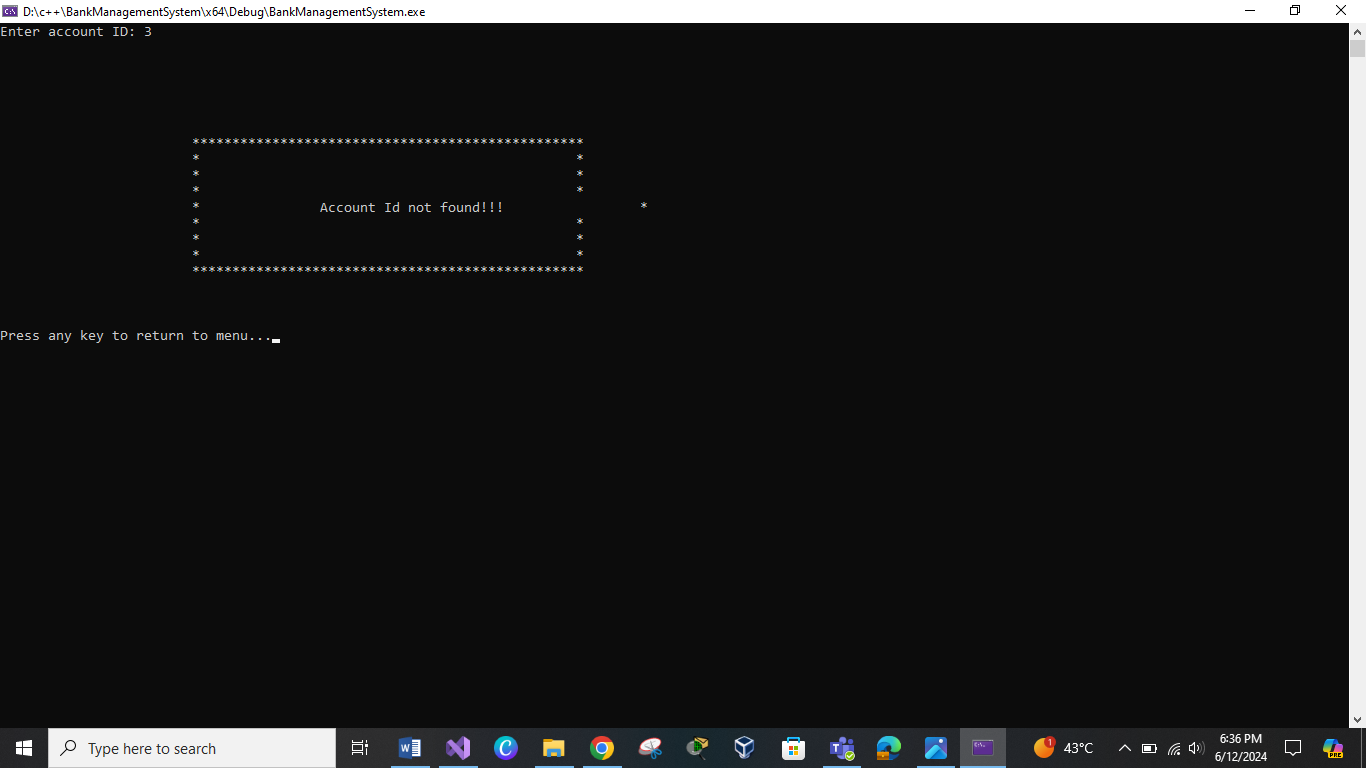
**Functionality number 2:**

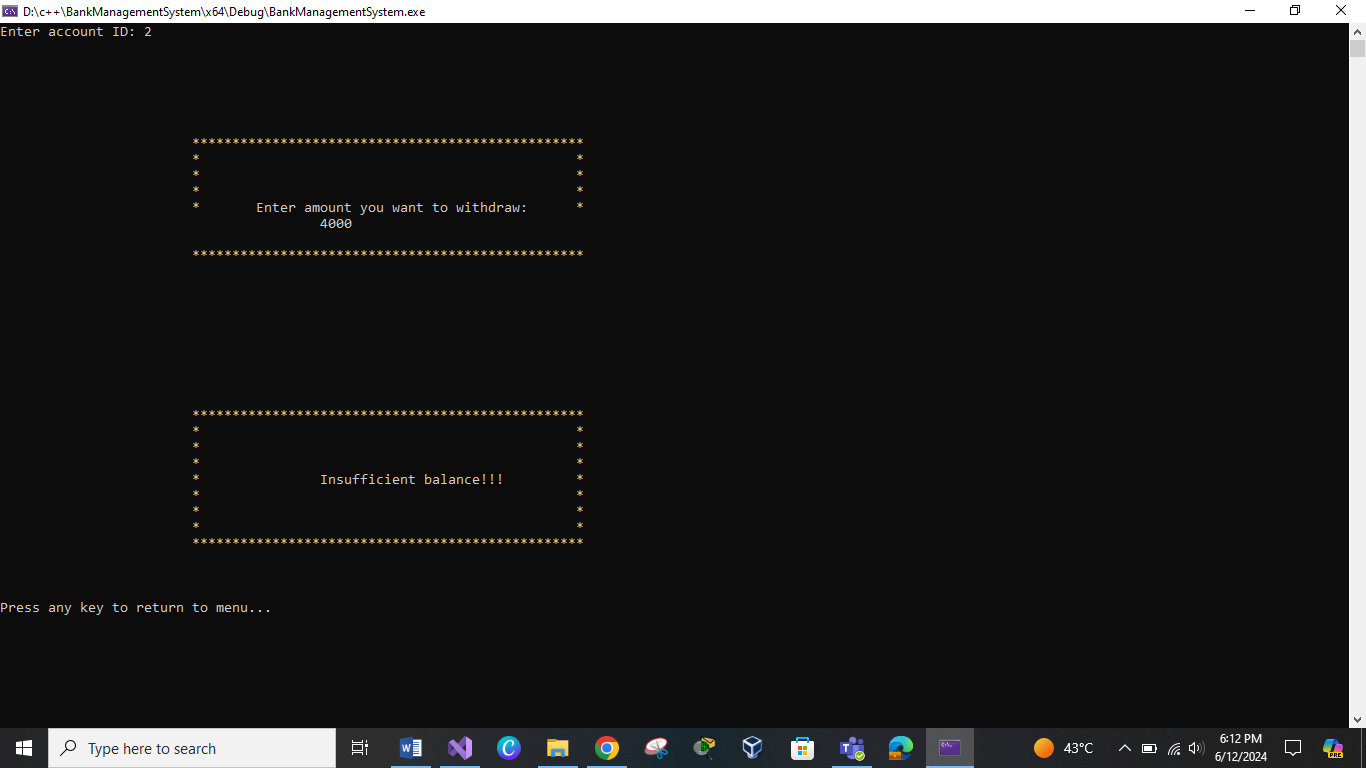
**Display Details:**

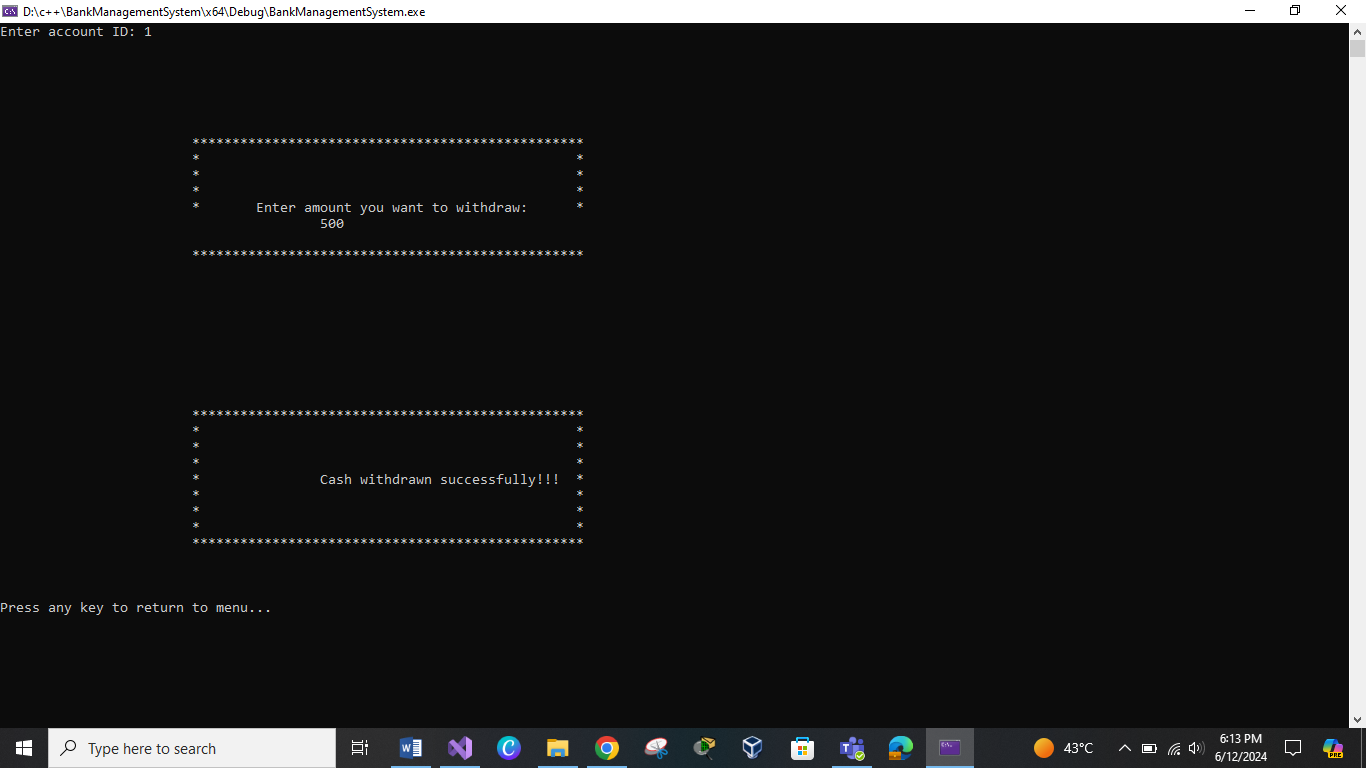
This Function displays the details of all accounts which are created in the bank   
Like:  


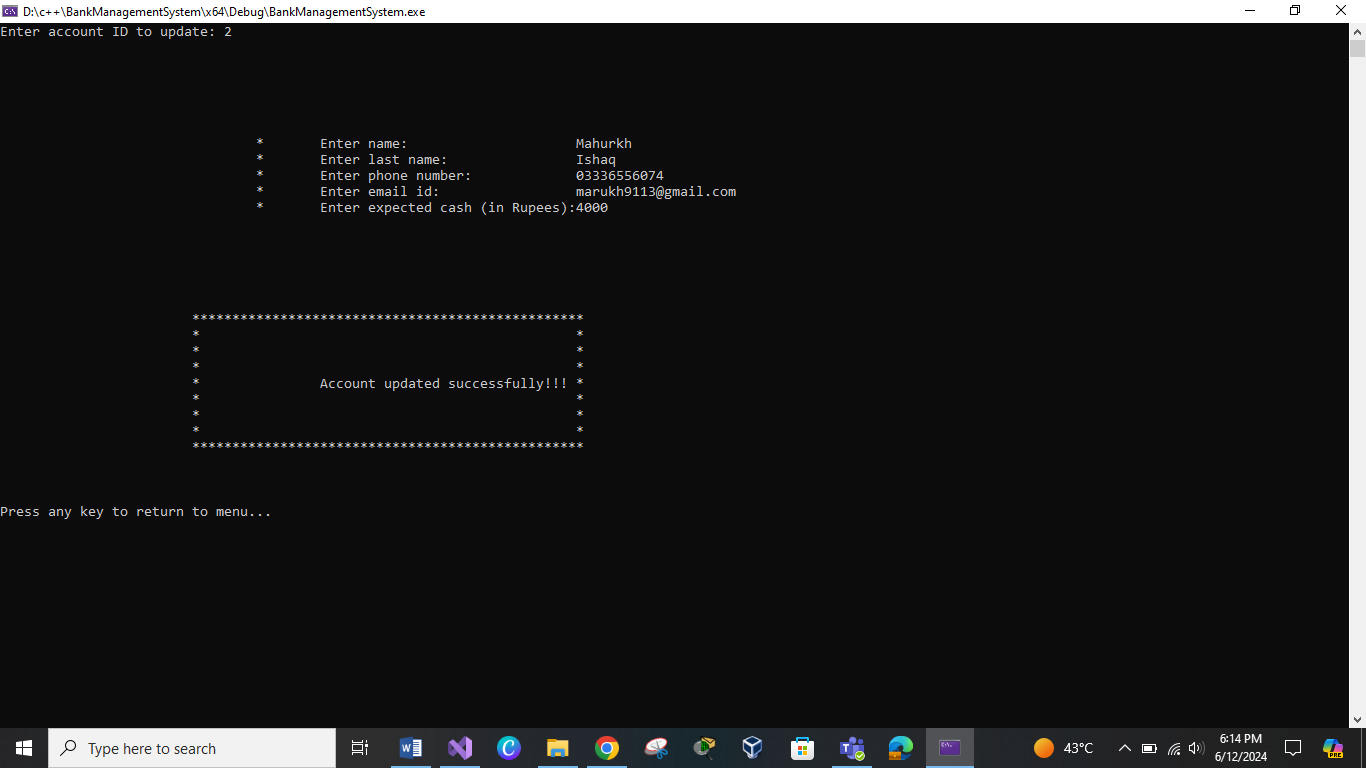
**Delete an account :**  
This functionality uses the linked list to make the deleteion efficient :  


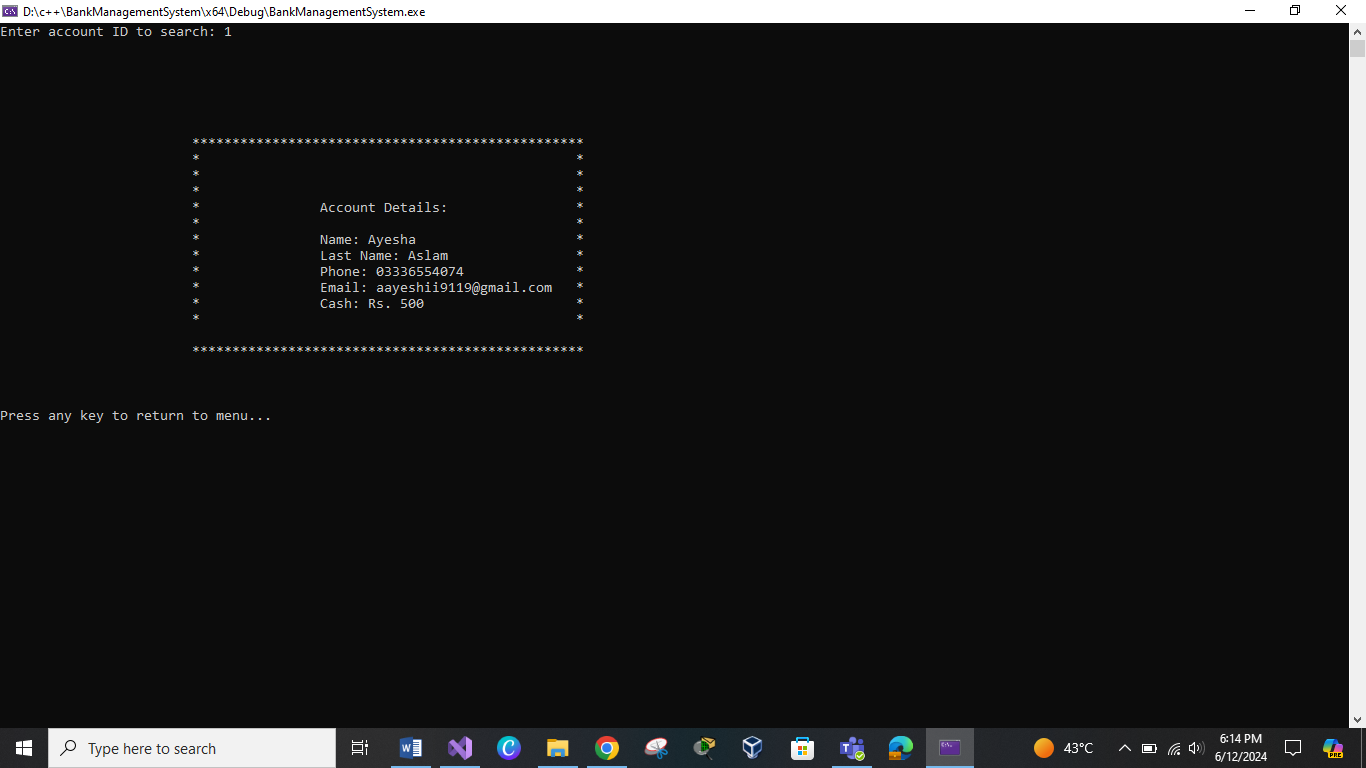
If the Account doent exits with the id then it will show the error message  
That this account id doenst exits:

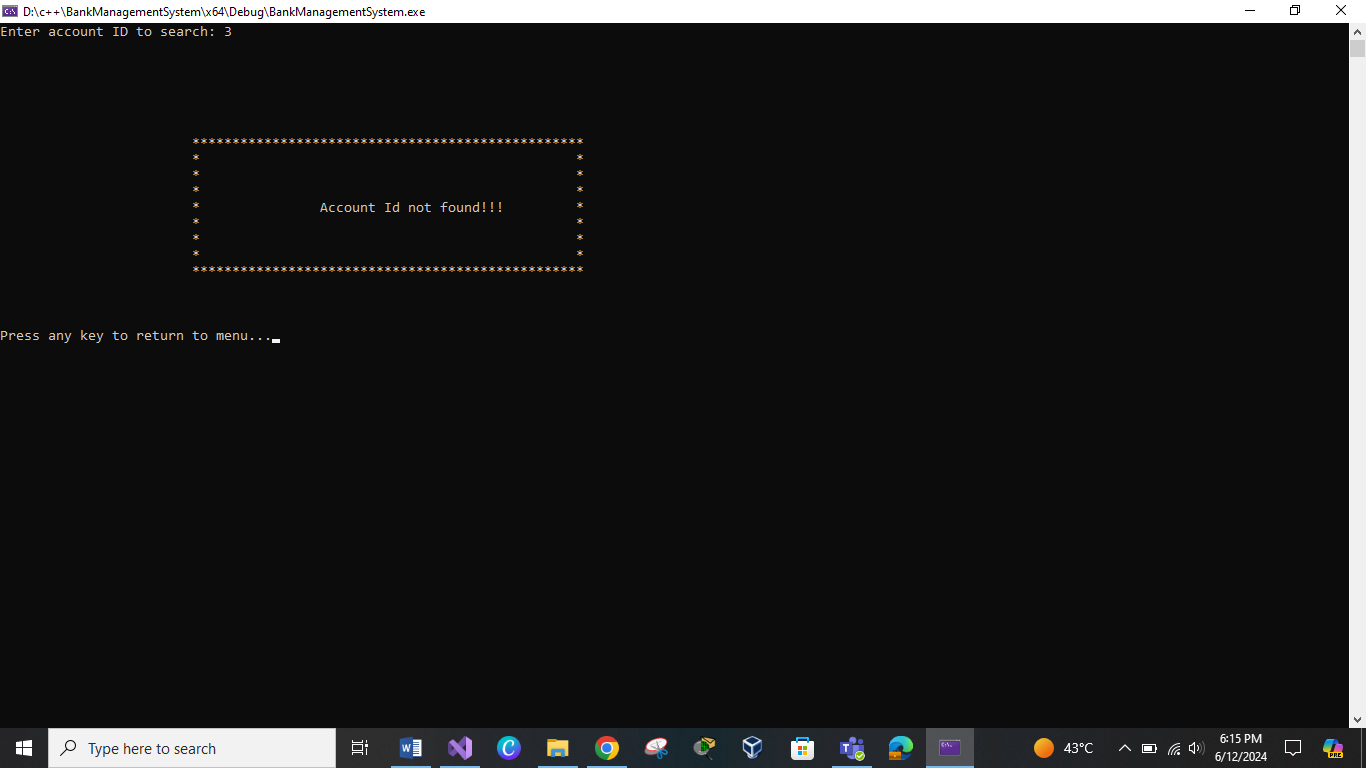
**Deposition:**  
If a customer wants to add some amount in its account then it suucessfully deposits in their accounts if their account exists.  
  
  
If there is no any account against the enterd pin code then it will not allow the customer to proceed next:  
**If id not exists**  


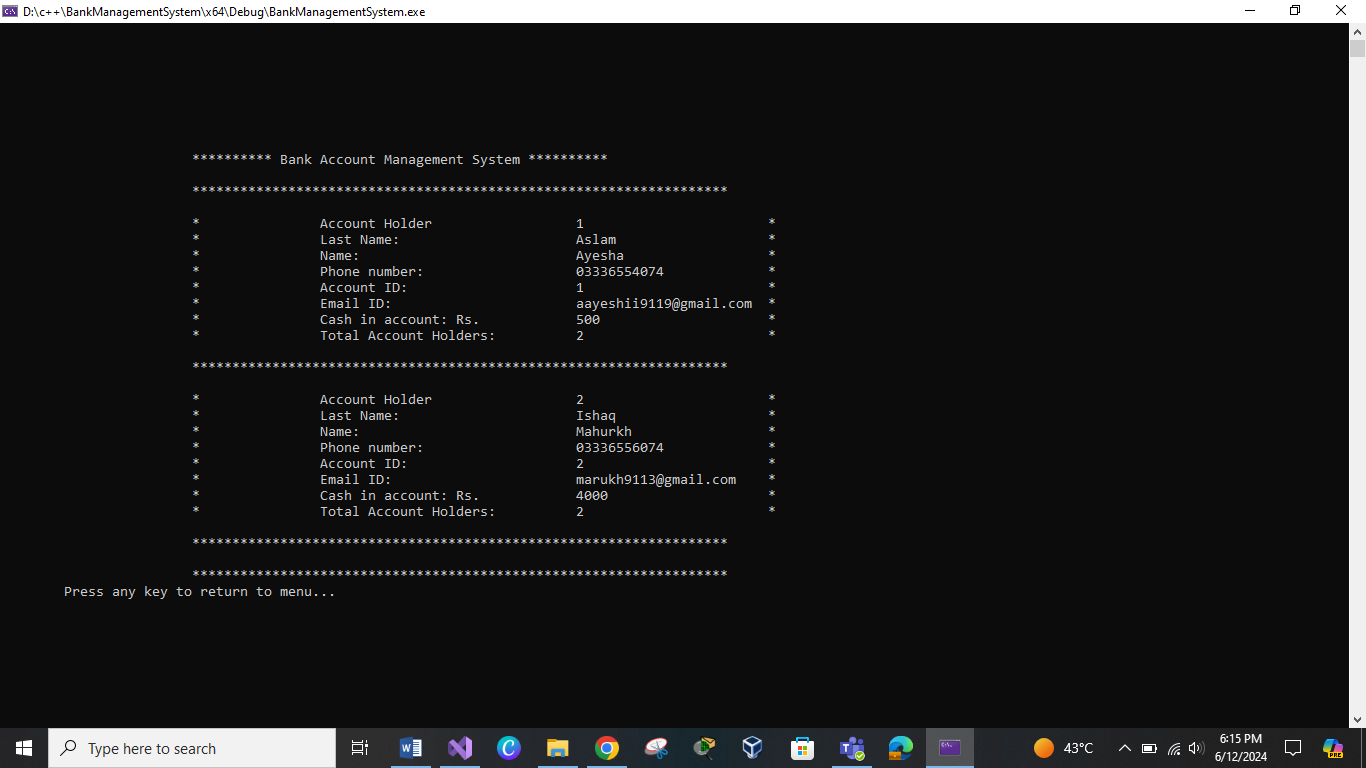
**WithDraw Functionality:**  
To with draw an amount the user has to enter its Id then it checks that id exists or not if id exists then it will check that there is sufficient amount in it :  
In case of insufficient Funds  


And to withdraw from id 3 persons account with suffiecient funds:  


**Updation Functionality()**  
The next one functionality is for updation:  


**Searching Functionality():**  
using linked list for its implementation is as:  


**Else case if id not matches:**Now here if we delete account id 3 and try to search it then it will show the error message:  


**List of accounts()**  
Here the list of all accounts after insertion deletion and updation is as :  


**Exiting Functionality();**  
To Exit this Program we have to press 9 then it will successfully exit :  
