**Semester Project :**

#include <iostream>

using namespace std;

class Mobile {

private:

string name;

double price;

Mobile\* next;

Mobile\* previous;

public:

Mobile(string name, double price)

{

this->name = name;

this->price = price;

this->next = NULL;

this->previous = NULL;

}

void setName(string name)

{

this->name = name;

}

string getName()

{

return name;

}

void setPrice(double price)

{

this->price = price;

}

double getPrice()

{

return price;

}

void setNext(Mobile\* next)

{

this->next = next;

}

Mobile\* getNext()

{

return next;

}

void setPrevious(Mobile\* previous)

{

this->previous = previous;

}

Mobile\* getPrevious()

{

return previous;

}

};

// Cart class to store items added to the cart and their quantity

class Cart {

private:

Mobile\* item; // Mobile item

int quantity; // Quantity of the item

Cart\* next; // Next item in the cart

public:

Cart(Mobile\* item, int quantity)

{

this->item = item;

this->quantity = quantity;

this->next = NULL;

}

Mobile\* getItem()

{

return item;

}

int getQuantity()

{

return quantity;

}

void setNext(Cart\* next)

{

this->next = next;

}

Cart\* getNext()

{

return next;

}

};

class Implementation {

private:

Mobile\* head; // Head of the linked list for mobile items

Cart\* cartHead; // Head of the linked list for cart items

double total; // Total price of the cart items

public:

Implementation()

{

this->head = NULL;

this->cartHead = NULL;

this->total = 0.0;

}

// Method to add a mobile to the list of mobiles

void add\_mobile()

{

string name;

double price;

cout << "Enter name of mobile: ";

cin >> name;

cout << "Enter price: ";

cin >> price;

// Check if mobile already exists

Mobile\* temp = head;

while (temp != NULL)

{

if (temp->getName() == name)

{

cout << "Mobile with this name already exists!\n";

return; // Don't add a duplicate

}

temp = temp->getNext();

}

// Add new mobile

Mobile\* new\_Mobile = new Mobile(name, price);

if (head == NULL)

{

head = new\_Mobile;

}

else

{

temp = head;

while (temp->getNext() != NULL)

{

temp = temp->getNext();

}

temp->setNext(new\_Mobile);

new\_Mobile->setPrevious(temp);

}

cout << "Mobile added successfully!\n";

}

// Method to remove a mobile from the list of mobiles

void remove\_mobile()

{

string name;

cout << "Enter the name of the mobile you want to remove: ";

cin >> name;

if (head == NULL)

{

cout << "No mobiles available.\n";

return;

}

Mobile\* temp = head;

while (temp != NULL && temp->getName() != name)

{

temp = temp->getNext();

}

if (temp == NULL)

{

cout << "Mobile not found.\n";

return;

}

if (temp == head)

{

head = temp->getNext();

if (head != NULL)

{

head->setPrevious(NULL);

}

delete temp;

return;

}

if (temp->getPrevious() != NULL)

{

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

}

if (temp->getNext() != NULL)

{

temp->getNext()->setPrevious(temp->getPrevious());

}

delete temp;

}

// Method to update the price of a mobile

void update\_price()

{

if (head == NULL)

{

cout << "No mobiles available.\n";

return;

}

string name;

double new\_price;

cout << "Enter the name of the mobile to update the price: ";

cin >> name;

Mobile\* temp = head;

while (temp != NULL && temp->getName() != name)

{

temp = temp->getNext();

}

if (temp == NULL)

{

cout << "Mobile not found.\n";

return;

}

cout << "Enter the new price: ";

cin >> new\_price;

temp->setPrice(new\_price);

cout << "Price updated successfully!\n";

}

// Method to search for a mobile

void search\_mobile()

{

if (head == NULL)

{

cout << "No mobiles available.\n";

return;

}

string name;

cout << "Enter the name of the mobile to search: ";

cin >> name;

Mobile\* temp = head;

while (temp != NULL)

{

if (temp->getName() == name)

{

cout << "\nMobile Found!\n";

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

cout << "Name: " << temp->getName() << "\n";

cout << "Price: $" << temp->getPrice() << "\n";

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

return;

}

temp = temp->getNext();

}

cout << "Mobile not found.\n";

}

// Method to view all available mobiles

void view\_all()

{

if (head == NULL)

{

cout << "No mobiles available.\n";

return;

}

Mobile\* temp = head;

cout << "\n--- Available Mobiles ---\n";

while (temp != NULL)

{

cout << "Name: " << temp->getName() << ", Price: $" << temp->getPrice() << "\n";

temp = temp->getNext();

}

}

// Method to add a mobile to the cart

void add\_to\_cart()

{

if (head == NULL)

{

cout << "No items available to add to the cart.\n";

return;

}

string name;

int quantity;

cout << "Enter the name of the item to add to cart: ";

cin >> name;

cout << "Enter the quantity: ";

cin >> quantity;

Mobile\* temp = head;

while (temp != NULL && temp->getName() != name)

{

temp = temp->getNext();

}

if (temp == NULL)

{

cout << "Item not found.\n";

return;

}

// Add the mobile to the cart

Cart\* newCartItem = new Cart(temp, quantity);

newCartItem->setNext(cartHead);

cartHead = newCartItem;

// Update total price

total += temp->getPrice() \* quantity;

cout << "Item added to the cart.\n";

}

// Method to view items in the cart

void view\_cart()

{

if (cartHead == NULL)

{

cout << "Your cart is empty.\n";

return;

}

Cart\* temp = cartHead;

cout << "\n--- Cart Items ---\n";

while (temp != NULL)

{

cout << "Name: " << temp->getItem()->getName()

<< ", Quantity: " << temp->getQuantity()

<< ", Total: $" << temp->getItem()->getPrice() \* temp->getQuantity() << "\n";

temp = temp->getNext();

}

cout << "Total Price: $" << total << "\n";

}

void remove\_from\_cart()

{

if (cartHead == NULL)

{

cout << "Your cart is empty.\n";

return;

}

string name;

cout << "Enter the name of the mobile to remove from cart: ";

cin >> name;

Cart\* temp = cartHead;

Cart\* prev = NULL;

// Traverse the cart to find the item

while (temp != NULL && temp->getItem()->getName() != name)

{

prev = temp;

temp = temp->getNext();

}

if (temp == NULL)

{

cout << "Item not found in cart.\n";

return;

}

// Update total price by removing the item's price

total -= temp->getItem()->getPrice() \* temp->getQuantity();

// Remove the item from the cart

if (prev == NULL)

{

cartHead = temp->getNext(); // Removing the first item in the cart

}

else

{

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

}

delete temp; // Delete the cart item

cout << "Item removed from cart.\n";

}

// Login method for admin

void LogIn()

{

string user = "admin";

string pass = "1234";

string username;

string password;

cout << "Enter username: ";

cin >> username;

cout << "Enter password: ";

cin >> password;

if (username == user && password == pass)

{

cout << "Login successful.\n";

Admin\_Menu();

} else

{

cout << "Incorrect username or password.\n";

}

}

// Admin menu with options

void Admin\_Menu() {

int choice;

do {

cout << " Admin Menu \n";

cout << " 1. Add mobile \n";

cout << " 2. Remove mobile \n";

cout << " 3. Search mobile \n";

cout << " 4. Update price \n";

cout << " 5. View all mobiles \n";

cout << " 6. Exit \n";

cout << "Enter your choice: ";

cin >> choice;

switch (choice)

{

case 1:

add\_mobile();

break;

case 2:

remove\_mobile();

break;

case 3:

search\_mobile();

break;

case 4:

update\_price();

break;

case 5:

view\_all();

break;

case 6:

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

break;

default:

cout << "Invalid choice.\n";

break;

}

} while (choice != 6);

}

// Customer menu with options

void Customer\_Menu() {

int choice;

do {

cout << " Customer Menu \n";

cout << " 1. View all mobiles \n";

cout << " 2. Search mobile \n";

cout << " 3. Add to cart \n";

cout << " 4. View cart \n";

cout << " 5. Remove from cart \n";

cout << " 6. Exit \n";

cout << "Enter your choice: ";

cin >> choice;

switch (choice)

{

case 1:

view\_all();

break;

case 2:

search\_mobile();

break;

case 3:

add\_to\_cart();

break;

case 4:

view\_cart();

break;

case 5:

remove\_from\_cart();

break;

case 6:

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

break;

default:

cout << "Invalid choice.\n";

break;

}

} while (choice != 5);

}

};

int main() {

Implementation imp;

int choice;

do {

cout << " Mobile Shop \n";

cout << " 1. Owner \n";

cout << " 2. Customer \n";

cout << " 3. Exit \n";

cout << "Enter your choice: ";

cin >> choice;

switch (choice)

{

case 1:

imp.LogIn();

break;

case 2:

imp.Customer\_Menu();

break;

case 3:

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

break;

default:

cout << "Invalid choice.\n";

break;

}

} while (choice != 3);

return 0;

}