#include <iostream>

#include <string>

#include <sstream>

#include <algorithm> // For transform

#include <iomanip> // for formatting the invoice

#include <fstream> // For file handling

#include <conio.h> // For getch()

using namespace std;

// Structure for product details

struct Product {

string name;

float price;

int quantity;

};

// Structure for purchase details

struct Purchase {

string customerName;

string itemsBought;

string quantitiesBought;

string pricesBought;

float totalAmount;

float discount;

};

// Function prototypes

void saveStockToFile(Product products[], int productCount);

void logCustomerPurchase(const Purchase &purchase);

// Function to mask password

string getPassword() {

string password;

char ch;

cout << "Enter your password: ";

while ((ch = getch()) != '\r') { // '\r' is the Enter key

if (ch == '\b') { // Handle Backspace

if (!password.empty()) {

cout << "\b \b"; // Erase character from console

password.pop\_back();

}

}

else {

password.push\_back(ch);

cout << '\*'; // Display '\*' for each character typed

}

}

cout << endl; // Move to the next line after Enter

return password;

}

// Main (Primary) function

int main() {

// Predefined prices and quantities of the products

Product products[] = {

{"Pen (Blue)", 25.5, 10000},

{"Pencil (lead)", 15, 11000},

{"Notebook", 100, 2000},

{"Book (FOP)", 450, 1000},

{"Book (ITC)", 950, 1000}

};

int productCount = 5;

bool exitFlag = false;

bool found = false;

string choice, choice1, itemName;

int buyQuantity;

string username, userType;

float totalAmount = 0;

char customerName[1000];

// To store the purchased items for the invoice

string itemsBought = "";

string quantitiesBought = "";

string pricesBought = "";

bool isLoggedIn = false; // Flag to track login status

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

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

cout << "|| WELCOME TO THE STATIONERY SYSTEM ||\n";

cout << "|| Managed by Alpha Squad ||\n";

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

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

cout << "|| FOP Lab Project - Semester 1 ||\n";

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

cout << "|| Your One-Stop Solution for Stationery! ||\n";

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

do {

cout << endl << "Login as:\n" << endl;

cout << "1. Staff\n";

cout << "2. Customer\n";

cout << "3. End Program\n\n";

cout << "Enter your choice: ";

cin >> userType;

if (userType == "1")

{

// Staff login section

while (!isLoggedIn) {

cout << "\nStaff Login\n";

cout << "Enter username: ";

cin >> username;

string password = getPassword(); // Masked password input

transform(username.begin(), username.end(), username.begin(), ::tolower);

if (username == "alpha1" && password == "1122Z") {

cout << "Login successful. \nWELCOME!!!, Manager Member1!\n";

isLoggedIn = true;

}

else if (username == "alpha2" && password == "1122K") {

cout << "Login successful. \nWELCOME!!!, Shop Owner Member2!\n";

isLoggedIn = true;

}

else if (username == "alpha3" && password == "1122A") {

cout << "Login successful. \nWELCOME!!!, Operational Head Member 3!\n";

isLoggedIn = true;

}

else if (username == "alpha4" && password == "1122F") {

cout << "Login successful. \nWELCOME!!!, Sales Director Member 4!\n";

isLoggedIn = true;

}

else if (username == "alpha5" && password == "1122AB") {

cout << "Login successful. \nWELCOME!!!, Tech Consultant Member 5!\n";

isLoggedIn = true;

}

else {

cout << "Invalid username or password. Access denied.\n";

}

}

while (isLoggedIn) {

cout << "\nStaff Menu:\n";

cout << "1. View Stock\n";

cout << "2. Update Stock\n";

cout << "3. Exit\n";

cout << "Enter your choice: ";

cin >> choice;

cout<<endl;

if (choice == "1")

{

// View stock

cout << "\nCurrent Stock:\n";

for (int i = 0; i < productCount; i++) // Use of FOR Loop

{

cout << products[i].name <<"\t\t"<< ": Price Rs " << products[i].price << "\t\t, Quantity: " << products[i].quantity << endl;

}

}

else if (choice == "2")

{

// Update stock

while (true)

{

cout << "\nEnter what you want to update\n";

cout << "1. for Quantity \n2. for Price \n3. for Exit\n";

cout << "\nChoice: ";

cin >> choice1;

if (choice1 == "1")

{

cout << "\nEnter the name of the item to update quantity:\n";

cout << "1. Pen \n";

cout << "2. Pencil \n";

cout << "3. Notebook \n";

cout << "4. FOP Book \n";

cout << "5. ITC Book \n";

cout << "6. Exit"<<endl;

do

{

// Update quantity

cout << "\nChoice: ";

cin >> itemName;

cout<<endl;

transform(itemName.begin(), itemName.end(), itemName.begin(), ::tolower);

if (itemName == "pen" || itemName == "1")

{

cout << "Current quantity of PEN: " << products[0].quantity << endl;

cout << "Enter new quantity: ";

cin >> buyQuantity;

products[0].quantity += buyQuantity;

cout << "Pen stock updated to " << products[0].quantity << " pens.\n";

}

else if (itemName == "pencil" || itemName == "2")

{

cout << "Current quantity of PENCIL: " << products[1].quantity << endl;

cout << "Enter new quantity: ";

cin >> buyQuantity;

products[1].quantity += buyQuantity;

cout << "Pencil stock updated to " << products[1].quantity << " pencils.\n";

}

else if (itemName == "notebook" || itemName == "3")

{

cout << "Current quantity of NOTEBOOK: " << products[2].quantity << endl;

cout << "Enter new quantity: ";

cin >> buyQuantity;

products[2].quantity += buyQuantity;

cout << "Notebook stock updated to " << products[2].quantity << " notebooks.\n";

}

else if (itemName == "book fop" || itemName == "4" || itemName == "fop")

{

cout << "Current quantity of BOOK (FOP): " << products[3].quantity << endl;

cout << "Enter new quantity: ";

cin >> buyQuantity;

products[3].quantity += buyQuantity;

cout << "Book (FOP) stock updated to " << products[3].quantity << " books.\n";

}

else if (itemName == "book itc" || itemName == "5" || itemName == "itc")

{

cout << "Current quantity of BOOK (ITC): " << products[4].quantity << endl;

cout << "Enter new quantity: ";

cin >> buyQuantity;

products[4].quantity += buyQuantity;

cout << "Book (ITC) stock updated to " << products[4].quantity << " books.\n";

}

else if (itemName == "exit" || itemName == "6")

{

break; // Exit the loop

}

else {

cout << "\nInvalid Selection\n";

}

} while (true); // Loop continues until 'exit' is chosen

}

else if (choice1 == "2")

{

// Update price

cout << "\nEnter the name of the item to update price:\n";

cout << "1. Pen\n";

cout << "2. Pencil\n";

cout << "3. Notebook\n";

cout << "4. FOP\n";

cout << "5. ITC\n";

cout << "6. STOP - Exit\n";

do {

cout << "\nChoice: ";

cin >> itemName;

// Convert itemName to lowercase

transform(itemName.begin(), itemName.end(), itemName.begin(), ::tolower);

if (itemName == "pen" || itemName == "1") {

cout << "Current price of PEN: " << products[0].price << endl;

cout << "Enter new Price: ";

cin >> products[0].price;

if (products[0].price < 0) {

cout << "\nInvalid price.\n";

}

else

{

cout << "Pen price updated to Rs " << products[0].price << ".\n\n";

}

}

else if (itemName == "pencil" || itemName == "2") {

cout << "Current price of PENCIL: " << products[1].price << endl;

cout << "Enter new Price: ";

cin >> products[1].price;

if (products[1].price < 0) {

cout << "\nInvalid price.\n";

}

else

{

cout << "Pencil price updated to Rs " << products[1].price << ".\n\n";

}

}

else if (itemName == "notebook" || itemName == "3") {

cout << "Current price of NOTEBOOK: " << products[2].price << endl;

cout << "Enter new Price: ";

cin >> products[2].price;

if (products[2].price < 0) {

cout << "\nInvalid price.\n";

}

else

{

cout << "Notebook price updated to Rs " << products[2].price << ".\n\n";

}

}

else if (itemName == "book fop" || itemName == "4" || itemName == "fop") {

cout << "Current price of BOOK (FOP): " << products[3].price << endl;

cout << "Enter new Price: ";

cin >> products[3].price;

if (products[3].price < 0) {

cout << "\nInvalid price.\n";

}

else

{

cout << "Book (FOP) price updated to Rs " << products[3].price << ".\n\n";

}

}

else if (itemName == "book itc" || itemName == "5" || itemName == "itc") {

cout << "Current price of BOOK (ITC): " << products[4].price << endl;

cout << "Enter new Price: ";

cin >> products[4].price;

if (products[4].price < 0) {

cout << "\nInvalid price.\n";

}

else

{

cout << "Book (ITC) price updated to Rs " << products[4].price << ".\n\n";

}

}

else if (itemName == "end" || itemName == "6" || itemName == "exit") {

exitFlag = true;

}

else

{

cout << "\nInvalid Selection";

}

}

while (!exitFlag);

}

else if (choice1 == "3") {

break; // Exit the update menu

}

}

}

else if (choice == "3") {

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

saveStockToFile(products, productCount); // Save stock data before exit

break;

}

}

}

else if (userType == "2") {

// Customer login section

cout << "\nRespected Customer, Welcome!\n";

cout << "Please enter your name: ";

cin >> customerName;

Purchase purchase; // Create a Purchase object

purchase.customerName = customerName;

purchase.totalAmount = 0;

purchase.discount = 0;

itemsBought = quantitiesBought = pricesBought = "";

// Buy item

cout << endl;

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

cout << "\* \*\*\* BUMPER OFFER ALERT! \*\*\* \*\n";

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

cout << "\* Spend Rs. 1000+ to avail a 5% Discount \*\n";

cout << "\* Spend Rs. 2500+ to avail a 10% Discount \*\n";

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

while (true) {

cout << "\nCustomer Menu:\n";

cout << "1. View Products\n";

cout << "2. Buy Products\n";

cout << "3. Exit\n";

cout << "Enter your choice: ";

cin >> choice;

if (choice == "1") {

// View stock for customer

cout << "\nAvailable Products:\n";

for (int i = 0; i < productCount; i++) {

cout << products[i].name <<"\t\t"<< ": Price Rs " << products[i].price << "\t\t, Quantity: " << products[i].quantity << endl;

}

}

else if (choice == "2")

{

cout << "\nEnter the name of the item to buy (or type 'exit' to stop):\n";

cout << "1. Pen\n";

cout << "2. Pencil\n";

cout << "3. Notebook\n";

cout << "4. FOP\n";

cout << "5. ITC\n";

cout << "6. STOP - Exit\n";

// Buy products

// Buy products

while (true) {

// Display the menu before getting the user input

cout << endl << "Choice : ";

cin >> itemName;

cout<<endl;

// Break the loop if the user types 'exit' or '6'

if (itemName == "exit" || itemName == "6")

break;

// Search for the product

bool found = false;

for (int i = 0; i < productCount; i++) {

if (itemName == products[i].name || (isdigit(itemName[0]) && stoi(itemName) - 1 == i)) {

found = true;

itemName = products[i].name;

cout << "Available quantity for " << itemName << " is " << products[i].quantity;

cout << endl << endl << "Enter quantity for " << itemName << " to buy: ";

cin >> buyQuantity;

if (buyQuantity > products[i].quantity) {

cout << "Sorry, not enough stock. Available stock: " << products[i].quantity << "\n";

}

else {

// Update product quantity and total amount

products[i].quantity -= buyQuantity;

float totalPrice = products[i].price \* buyQuantity;

totalAmount += totalPrice;

// Add purchase details to invoice

itemsBought += itemName + ", ";

quantitiesBought += to\_string(buyQuantity) + ", ";

pricesBought += to\_string(totalPrice) + ", ";

// Print purchase confirmation

cout << buyQuantity << " " << itemName << "(s) bought successfully.\n";

cout << "Total price for " << buyQuantity << " " << itemName << "(s): Rs " << totalPrice << "\n";

}

if (!found) {

cout << "Item not found. Please try again.\n";

}

break;

}

}

}

// Exit and print invoice

purchase.itemsBought = itemsBought;

purchase.quantitiesBought = quantitiesBought;

purchase.pricesBought = pricesBought;

purchase.totalAmount = totalAmount;

purchase.discount = 0.0; // Initialize discount to 0

// Apply discounts

if (totalAmount >= 2500) {

purchase.discount = totalAmount \* 0.10; // 10% discount

purchase.totalAmount -= purchase.discount;

cout << "\nCongratulations!!! You got 10% off\n";

}

else if (totalAmount >= 1000) {

purchase.discount = totalAmount \* 0.05; // 5% discount

purchase.totalAmount -= purchase.discount;

cout << "\nCongratulations!!! You got 5% off\n";

}

// Print the invoice

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

cout << " \*\*\* INVOICE \*\*\*\n";

cout << "Customer Name: " << purchase.customerName << endl;

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

// Format and print purchased items, quantities, and prices

stringstream itemsStream(purchase.itemsBought);

stringstream qtyStream(purchase.quantitiesBought);

stringstream priceStream(purchase.pricesBought);

string item, qty, price;

while (getline(itemsStream, item) &&

getline(qtyStream, qty) &&

getline(priceStream, price)) {

cout << "Item Name:\t" << item << "\n -> Quantity:\t" << qty<<"\t"

<< "\n -> Rs.\t" << fixed << setprecision(2)

<< totalAmount << endl;

}

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

cout << "Subtotal: Rs. " << fixed << setprecision(2) << purchase.totalAmount << endl;

// Show discount if any

cout << "Discount: -Rs. " << fixed << setprecision(2) << purchase.discount << endl;

// Total Payable

cout << "Total Payable: Rs. " << fixed << setprecision(2) << purchase.totalAmount << endl;

cout << "Thank you for your purchase, " << purchase.customerName << "!" << endl;

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

// Log the customer purchase to file

logCustomerPurchase(purchase); // Log the purchase to file

break;

}

else if (choice == "3")

{}

}

}else if (userType == "3") {

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

}

else

{

cout << "\nInvalid option. Please try again.\n";

}

} while (userType != "3");

return 0;

}

// Function to save stock to file

void saveStockToFile(Product products[], int productCount) {

ofstream stockFile("stock.txt");

if (stockFile.is\_open()) {

stockFile << "Item name \t\tPrice \t\tQuantity \n=============================================\n";

for (int i = 0; i < productCount; i++) {

stockFile << products[i].name << ",\t\t "

<< products[i].price << ",\t\t "

<< products[i].quantity << endl;

}

stockFile.close();

cout << "Stock data saved to stock.txt\n";

}

else

{

cout << "Unable to open stock file for saving.\n";

}

}

// Function to log customer purchase to file

void logCustomerPurchase(const Purchase &purchase) {

ofstream purchaseFile("purchases.txt", ios::app); // Open in append mode

if (purchaseFile.is\_open()) {

purchaseFile << endl;

purchaseFile << " \*\*\* INVOICE \*\*\* " << endl;

purchaseFile << "=============================================" << endl;

purchaseFile << "Customer Name: " << purchase.customerName << endl;

purchaseFile << "Items Bought: " << purchase.itemsBought << endl;

purchaseFile << "Quantities: " << purchase.quantitiesBought << endl;

purchaseFile << "Prices: " << purchase.pricesBought << endl;

purchaseFile << "Total Amount: Rs " << fixed << setprecision(2) << purchase.totalAmount << endl;

purchaseFile << "Discount Applied: Rs " << purchase.discount << endl;

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

purchaseFile.close();

cout << "Purchase data logged to purchases.txt\n";

}

else

{

cout << "Unable to open purchases file for logging.\n";

}

}