#include <iostream>

#include <vector>

#include <fstream>

#include <iomanip>

#include <string>

using namespace std;

struct Order {

string id;

string name;

string item;

int quantity;

float price;

string status;

};

void loadOrders(vector<Order>& orders);

void saveOrders(const vector<Order>& orders);

void placeOrder(vector<Order>& orders);

void displayOrders(const vector<Order>& orders);

void markServed(vector<Order>& orders);

void salesSummary(const vector<Order>& orders);

int main() {

vector<Order> orders;

loadOrders(orders);

int choice;

do {

cout << "\n\*\*\*\*\*\* Welcome to Restaurant Order Manager \*\*\*\*\*\*\n";

cout << "1. Place New Order\n2. Display All Orders\n3. Mark Order as Served\n";

cout << "4. View Sales Summary\n5. Exit\n→ ";

cin >> choice;

cin.ignore();

switch (choice) {

case 1: placeOrder(orders); break;

case 2: displayOrders(orders); break;

case 3: markServed(orders); break;

case 4: salesSummary(orders); break;

case 5:

saveOrders(orders);

cout << "Orders saved to orders.txt. Have a great day!\n";

break;

default:

cout << "Invalid option. Try again.\n";

}

} while (choice != 5);

return 0;

}

void loadOrders(vector<Order>& orders) {

ifstream file("orders.txt");

if (!file) return;

Order o;

char comma;

while (file >> o.id >> comma >> o.name >> comma >> o.item >> comma >> o.quantity >> comma >> o.price >> comma >> o.status) {

orders.push\_back(o);

}

file.close();

}

void saveOrders(const vector<Order>& orders) {

ofstream file("orders.txt");

for (const auto& o : orders) {

float total = o.price \* o.quantity;

file << o.id << "," << o.name << "," << o.item << ","

<< o.quantity << "," << o.price << "," << total << "," << o.status << "\n";

}

file.close();

}

void placeOrder(vector<Order>& orders) {

Order o;

cout << "Enter Order ID: ";

cin >> o.id;

cout << "Enter customer name: ";

cin.ignore();

getline(cin, o.name);

cout << "Enter menu item: ";

getline(cin, o.item);

cout << "Enter quantity: ";

cin >> o.quantity;

cout << "Enter price per item: ";

cin >> o.price;

o.status = "Pending";

orders.push\_back(o);

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

}

void displayOrders(const vector<Order>& orders) {

if (orders.empty()) {

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

return;

}

cout << "\n=========== CURRENT ORDERS ===========\n";

cout << left << setw(8) << "ID" << setw(15) << "Name" << setw(12) << "Item"

<< setw(6) << "Qty" << setw(8) << "Price" << setw(8) << "Total" << "Status\n";

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

for (const auto& o : orders) {

float total = o.quantity \* o.price;

cout << left << setw(8) << o.id << setw(15) << o.name << setw(12)

<< o.item << setw(6) << o.quantity << setw(8)

<< fixed << setprecision(2) << o.price << setw(8)

<< total << o.status << "\n";

}

}

void markServed(vector<Order>& orders) {

string id;

cout << "Enter Order ID to mark as served: ";

cin >> id;

for (auto& o : orders) {

if (o.id == id) {

if (o.status == "Served") {

cout << "Order " << id << " is already served.\n";

}

else {

o.status = "Served";

cout << "Order " << id << " marked as Served.\n";

}

return;

}

}

cout << "Order ID not found.\n";

}

void salesSummary(const vector<Order>& orders) {

int totalOrders = orders.size();

int served = 0;

float totalSales = 0;

for (const auto& o : orders) {

if (o.status == "Served") {

served++;

totalSales += o.quantity \* o.price;

}

}

cout << "\n========== SALES SUMMARY ==========\n";

cout << "Total Orders: " << totalOrders << "\n";

cout << "Served: " << served << "\n";

cout << "Pending: " << totalOrders - served << "\n";

cout << "Total Sales: $" << fixed << setprecision(2) << totalSales << "\n";

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

}