**Project code**

**#include <iostream>**

**#include <fstream>**

**#include <string>**

**#include <iomanip>**

**using namespace std;**

**const string MENU\_FILE = "menu.txt";**

**const string ORDER\_FILE = "orders.txt";**

**// Structure for Menu Item**

**struct MenuItem {**

**int id;**

**string name;**

**double price;**

**};**

**// Structure for Order Item (linked list node)**

**struct OrderNode {**

**MenuItem item;**

**int quantity;**

**OrderNode\* next;**

**};**

**// Display the menu from the file**

**void displayMenu() {**

**ifstream menuFile(MENU\_FILE);**

**if (!menuFile) {**

**cout << "Menu file not found.\n";**

**return;**

**}**

**cout << "\nAvailable Menu:\n";**

**cout << "ID\tName\t\tPrice\n";**

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

**int id;**

**string name;**

**double price;**

**while (menuFile >> id >> name >> price) {**

**cout << id << "\t" << name << "\t$" << fixed << setprecision(2) << price << "\n";**

**}**

**menuFile.close();**

**}**

**// Add a menu item to the linked list**

**void addOrder(OrderNode\*& head, int itemId, int quantity) {**

**ifstream menuFile(MENU\_FILE);**

**int id;**

**string name;**

**double price;**

**bool found = false;**

**while (menuFile >> id >> name >> price) {**

**if (id == itemId) {**

**OrderNode\* newOrder = new OrderNode;**

**newOrder->item.id = id;**

**newOrder->item.name = name;**

**newOrder->item.price = price;**

**newOrder->quantity = quantity;**

**newOrder->next = head;**

**head = newOrder;**

**cout << "Added " << quantity << " x " << name << " to the order.\n";**

**found = true;**

**break;**

**}**

**}**

**if (!found) {**

**cout << "Item ID " << itemId << " not found in the menu.\n";**

**}**

**menuFile.close();**

**}**

**// View the current order from the linked list**

**void viewOrder(OrderNode\* head) {**

**if (!head) {**

**cout << "No orders found.\n";**

**return;**

**}**

**cout << "\nCurrent Order:\n";**

**cout << "ID\tName\t\tQty\tPrice\n";**

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

**double total = 0;**

**OrderNode\* current = head;**

**while (current) {**

**double itemTotal = current->item.price \* current->quantity;**

**cout << current->item.id << "\t" << current->item.name << "\t"**

**<< current->quantity << "\t$" << fixed << setprecision(2) << itemTotal << "\n";**

**total += itemTotal;**

**current = current->next;**

**}**

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

**cout << "Total: $" << fixed << setprecision(2) << total << "\n";**

**}**

**// Generate the bill and write the order to a file**

**void generateBill(OrderNode\*& head) {**

**if (!head) {**

**cout << "No orders found.\n";**

**return;**

**}**

**ofstream orderFile(ORDER\_FILE, ios::app);**

**if (!orderFile) {**

**cout << "Error opening file.\n";**

**return;**

**}**

**cout << "\nFinal Bill:\n";**

**cout << "ID\tName\t\tQty\tPrice\n";**

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

**double total = 0;**

**OrderNode\* current = head;**

**while (current) {**

**double itemTotal = current->item.price \* current->quantity;**

**cout << current->item.id << "\t" << current->item.name << "\t"**

**<< current->quantity << "\t$" << fixed << setprecision(2) << itemTotal << "\n";**

**total += itemTotal;**

**orderFile << current->item.id << " " << current->item.name << " "**

**<< current->item.price << " " << current->quantity << "\n";**

**current = current->next;**

**}**

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

**cout << "Total: $" << fixed << setprecision(2) << total << "\n";**

**orderFile.close();**

**// Clear the order list**

**while (head) {**

**OrderNode\* temp = head;**

**head = head->next;**

**delete temp;**

**}**

**cout << "Order completed and cleared.\n";**

**}**

**// Preload the menu file (if not already present)**

**void preloadMenu() {**

**ofstream menuFile(MENU\_FILE, ios::app);**

**ifstream testMenu(MENU\_FILE);**

**if (testMenu.peek() == EOF) {**

**menuFile << "1 Burger 5.99\n";**

**menuFile << "2 Pizza 8.99\n";**

**menuFile << "3 Pasta 7.49\n";**

**menuFile << "4 Fries 2.99\n";**

**menuFile << "5 Soda 1.50\n";**

**cout << "Menu preloaded.\n";**

**}**

**menuFile.close();**

**testMenu.close();**

**}**

**// Main menu**

**void menu() {**

**cout << "\nRestaurant Management System\n";**

**cout << "1. Display Menu\n";**

**cout << "2. Add Order\n";**

**cout << "3. View Order\n";**

**cout << "4. Generate Bill and Exit\n";**

**cout << "Enter your choice: ";**

**}**

**int main() {**

**preloadMenu();**

**OrderNode\* orderHead = nullptr;**

**int choice;**

**while (true) {**

**menu();**

**cin >> choice;**

**if (choice == 1) {**

**displayMenu();**

**} else if (choice == 2) {**

**int itemId, quantity;**

**cout << "Enter Item ID to order: ";**

**cin >> itemId;**

**cout << "Enter Quantity: ";**

**cin >> quantity;**

**addOrder(orderHead, itemId, quantity);**

**} else if (choice == 3) {**

**viewOrder(orderHead);**

**} else if (choice == 4) {**

**generateBill(orderHead);**

**break;**

**} else {**

**cout << "Invalid choice. Try again.\n";**

**}**

**}**

**return 0;**

**}**

**ont know about preloaded menu write the menu file in this code**

**ChatGPT said:**

**ChatGPT**

**In the code I provided earlier, the "preloaded menu" feature is used to automatically populate the menu.txt file with default items when it is empty. If you want to manually create or view the menu.txt file, here is how the file should look:**

**menu.txt (Manual File)**

**Copy code**

**1 Burger 5.99**

**2 Pizza 8.99**

**3 Pasta 7.49**

**4 Fries 2.99**

**5 Soda 1.50**

**Explanation:**

* **The menu.txt file stores the restaurant's menu in a specific format: ID Name Price.**
  + **ID: A unique identifier for each item (e.g., 1 for Burger).**
  + **Name: The name of the item (e.g., Burger).**
  + **Price: The price of the item in dollars (e.g., 5.99).**

**How it Works in the Code:**

* **The preloadMenu() function checks if the menu.txt file is empty (or doesn't exist) and populates it with the default menu items. If the file is already populated, it doesn't overwrite it.**
* **The menu.txt file must be placed in the same directory where the C++ program is running.**

**Menu Preloading Function:**

**cpp**

**Copy code**

**void preloadMenu() {**

**ofstream menuFile(MENU\_FILE, ios::app); // Open in append mode to prevent overwriting**

**ifstream testMenu(MENU\_FILE);**

**if (testMenu.peek() == EOF) { // If the file is empty**

**menuFile << "1 Burger 5.99\n";**

**menuFile << "2 Pizza 8.99\n";**

**menuFile << "3 Pasta 7.49\n";**

**menuFile << "4 Fries 2.99\n";**

**menuFile << "5 Soda 1.50\n";**

**cout << "Menu preloaded.\n";**

**}**

**menuFile.close();**

**testMenu.close();**

**}**