 **Bahria University Islamabad**

**Department of Computer Sciences**

**Computer Engineering**

COMPUTER PROGRAMMING PROJECT.

* Online Food Ordering System.

SUBMITTED BY:

1. SARIB SAAD (01-134251-077)

2. AYAN SHAMSHAD (01-134242-030)

SUBMITTED TO:

MR.MOHSIN JAVED.

**CP PROJECT CODE:**

**#include <iostream>**

**#include <string>**

**#include <vector>**

**#include <windows.h>**

**#include <iomanip> // For setting precision**

**using namespace std;**

**// Structure for menu items**

**struct MenuItem {**

**string name;**

**float price;**

**};**

**// Function to display the menu**

**//added setprecision to limit the flaot decimal point to 2 setprecision(2)**

**void displayMenu(const vector<MenuItem>& menu) {**

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

**for (size\_t i = 0; i < menu.size(); ++i) {**

**cout << (i + 1) << ". " << menu[i].name << " ($" << fixed << setprecision(2) << menu[i].price << ")\n";**

**}**

**}**

**// Function to place an order**

**void placeOrder(const vector<MenuItem>& menu) {**

**int choice;**

**double total = 0.0;**

**vector<MenuItem> order;**

**while (true) {**

**cout << "Enter the item number to add to your order:\n";**

**cout << "Press 0 to finish order\n";**

**displayMenu(menu);**

**cin >> choice;**

**if (choice == 0) {**

**break;**

**}**

**else if (choice >= 1 && choice <= static\_cast<int>(menu.size())) {**

**order.push\_back(menu[choice - 1]);**

**total += menu[choice - 1].price;**

**cout << "Added " << menu[choice - 1].name << " to your order.\n";**

**Sleep(1000); // fake delay to show processing**

**}**

**else {**

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

**Sleep(1000); // fake delay to show processing**

**}**

**}**

**cout << "Processing order..." << endl;**

**Sleep(2000); // fake delay to show processing**

**system("cls");**

**if (order.empty())**

**{**

**cout << "No items in the order.\n";**

**return;**

**}**

**cout << "Your order:\n";**

**for (size\_t i = 0; i < order.size(); ++i) {**

**cout << i + 1 << ". " << order[i].name << " ($" << fixed << setprecision(2) << order[i].price << ")\n";**

**}**

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

**}**

**int main()**

**{**

**string fname, lname, name, address;**

**vector<MenuItem> foodMenu = {**

**{"Burger", 5.99},**

**{"Pizza", 8.49},**

**{"Salad", 3.95},**

**{"Pasta", 6.75},**

**{"Sushi", 12.99},**

**{"Taco", 4.25},**

**{"Sandwich", 5.50},**

**{"Steak", 15.75},**

**{"Soup", 3.25},**

**{"Fried Chicken", 7.50},**

**{"Ramen", 9.25},**

**{"Fish and Chips", 10.99},**

**{"Burrito", 6.95},**

**{"Shawarma", 8.75},**

**{"Curry", 11.50},**

**{"Omelette", 5.25},**

**{"Smoothie Bowl", 4.50},**

**{"Waffle", 6.25},**

**{"Samosa", 2.99},**

**{"Pho", 8.25}**

**};**

**system("cls");**

**vector<MenuItem> drinkMenu = {**

**{"Mojito", 4.50},**

**{"Piña Colada", 5.25},**

**{"Moscow Mule", 4.75},**

**{"Bellini", 3.99},**

**{"Sangria", 4.25},**

**{"Paloma", 3.50},**

**{"Mimosa", 2.95},**

**{"Dark 'n' Stormy", 4.25},**

**{"Aperol Spritz", 5.50},**

**{"Whiskey Sour", 4.25},**

**{"Negroni", 5.75},**

**{"Old Fashioned", 6.25},**

**{"Sidecar", 5.99},**

**{"Caipirinha", 4.50},**

**{"Gin Rickey", 3.75},**

**{"Tequila Sunrise", 3.25},**

**{"Blue Lagoon", 3.50},**

**{"Hurricane", 5.25},**

**{"Lemon Drop Martini", 4.75},**

**{"Irish Coffee", 3.99},**

**{"Hot Toddy", 4.25},**

**{"Pisco Sour", 3.50},**

**{"French 75", 5.75}**

**};**

**system("cls");**

**vector<MenuItem> coffeeMenu = {**

**{"Espresso", 3.0},**

**{"Cappuccino", 4.5},**

**{"Latte", 4.0},**

**{"Mocha", 4.25},**

**{"Lemon Drop Martini", 4.75},**

**{"Irish Coffee", 3.99},**

**{"Hot Toddy", 4.25},**

**{"Pisco Sour", 3.50},**

**{"French 75", 5.75},**

**{"Boulevardier", 4.25},**

**{"Gimlet", 3.75},**

**{"Manhattan", 5.99},**

**{"Sazerac", 4.50},**

**{"Corpse Reviver", 4.25}**

**};**

**system("cls");**

**vector<MenuItem> SweetsMenu = {**

**{"Chocolate Cake", 4.99},**

**{"Vanilla Cupcake", 2.50},**

**{"Strawberry Cheesecake", 5.75},**

**{"Brownie", 3.25},**

**{"Macaron", 1.99},**

**{"Lemon Tart", 3.50},**

**{"Pecan Pie", 6.25},**

**{"Red Velvet Cookie", 2.75},**

**{"Churros", 3.95},**

**{"Caramel Flan", 4.50},**

**{"Fruit Salad", 3.25},**

**{"Chocolate Truffle", 4.75},**

**{"Rice Pudding", 2.99},**

**{"Blueberry Muffin", 2.25},**

**{"Coconut Macaroon", 1.75},**

**{"Pumpkin Pie", 5.50},**

**{"Banana Split", 4.25},**

**{"Tiramisu", 6.99},**

**{"Cotton Candy", 3.50},**

**{"Gingerbread Cookie", 2.49} };**

**system("cls");**

**vector<MenuItem> icecreamsMenu = {**

**{"Vanilla", 3.50},**

**{"Chocolate", 3.75},**

**{"Strawberry", 4.25},**

**{"Mint Chocolate Chip", 4.50},**

**{"Cookies and Cream", 4.99},**

**{"Rocky Road", 5.25},**

**{"Butter Pecan", 4.75},**

**{"Coffee", 3.95},**

**{"Pistachio", 4.50},**

**{"Salted Caramel", 4.25},**

**{"Neapolitan", 5.50},**

**{"Cookie Dough", 4.25},**

**{"Raspberry Ripple", 3.99},**

**{"Matcha Green Tea", 4.75},**

**{"Toasted Marshmallow", 4.50},**

**{"Cherry Garcia", 5.75},**

**{"Maple Walnut", 4.25},**

**{"Bubblegum", 3.50},**

**{"Tiramisu", 5.99},**

**{"Red Velvet", 4.25} };**

**system("cls");**

**vector<MenuItem> fishMenu = {**

**{"Salmon",4.3},**

**{"Tuna",8},**

**{"Cod",4.2},**

**{"Trout",2.3},**

**{"Halibut",3},**

**{"Sardines",5.5},**

**{"Mahi Mahi",6},**

**{"Barramundi",7.1},**

**{"Snapper",2.2},**

**{"Grouper",1.2},**

**{"Haddock",2.2},**

**{"Mackerel",3.8},**

**{"Catfish",5.5},**

**{"Sole",5.6},**

**{"Flounder",6.6},**

**{"Perch",7.7},**

**{"Rainbow Trout",10},**

**{"Anchovies",4.5},**

**{"Swordfish",3.3},**

**{"Bluefish",2.2}**

**};**

**system("cls");**

**system("color 1f");**

**cout << setw(54) << "\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";**

**cout << setw(54) << "\t\t\t\t|Subject : Computer Programming |\n";**

**cout << setw(54) << "\t\t\t\t|Semester : 1st |\n";**

**cout << setw(54) << "\t\t\t\t|To : Mr Mohsin Javed |\n";**

**cout << setw(54) << "\t\t\t\t|From : Roll No 077 & 030 |\n";**

**cout << setw(54) << "\t\t\t\t|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\n";**

**cout << setw(54) << "\t\t\t\t|Welcome to our project. |\n";**

**cout << setw(54) << "\t\t\t\t|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\n";**

**Sleep(4000);**

**system("cls");**

**system("color 1f");**

**cout << "\n\t\n\t\n\t\n\t\n\t\n\t\n\t\n\t\n\t\n\t\n\t\n\t\n\t\t\t\t\t LOADING........." << endl;**

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

**cout << setw(57) << "\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;**

**cout << setw(57) << "\t\t\t\t| Developed By Sarib Saad & Ayan Shamshad |" << endl;**

**cout << setw(57) << "\t\t\t\t|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;**

**Sleep(2000);**

**system("cls");**

**system("color 1f");**

**cout << setw(54) << "\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;**

**cout << setw(57) << "\t\t\t| |" << endl;**

**cout << setw(57) << "\t\t\t| A SSSSSSSS |" << endl;**

**cout << setw(57) << "\t\t\t| A A S |" << endl;**

**cout << setw(57) << "\t\t\t| A A S |" << endl;**

**cout << setw(57) << "\t\t\t| A A SSSSSSSS |" << endl;**

**cout << setw(57) << "\t\t\t| AAAAAAAAA S |" << endl;**

**cout << setw(57) << "\t\t\t| A A S |" << endl;**

**cout << setw(57) << "\t\t\t| A A S |" << endl;**

**cout << setw(57) << "\t\t\t| A A SSSSSSSS |" << endl;**

**cout << setw(57) << "\t\t\t| RESTAURANT |" << endl;**

**cout << setw(57) << "\t\t\t|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;**

**cout << "\n\n\t\t\tWelcome to our online food delivery service!" << endl;**

**cout << "\t\t\tHere are some important rules to follow:" << endl;**

**cout << "\n\n1. Ensure that the food has proper packaging to maintain freshness." << endl;**

**cout << "2. Keep track of temperature control during delivery." << endl;**

**cout << "3. Avoid physical and chemical contamination." << endl;**

**cout << "4. Be aware of allergens and label them appropriately." << endl;**

**cout << "5. include any conditions or deadlines for requesting refunds." << endl;**

**cout << "6. Send order confirmation and delivery updates via SMS or email(saribsaad3252@GMAIL.COM)" << endl;**

**Sleep(4000);**

**system("cls");**

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

**cin >> fname;**

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

**cin >> lname;**

**name = fname + " " + lname;**

**cout << "\n\n\t\t\tEnter your Address: ";**

**cin >> address;**

**system("color 6f");**

**cout << "\n\n\n\n\n\t\t\t\t\t\tHello, " << name << endl;**

**cout << "\n\t\t\t\t\t Your Delviery Address: " << address << endl;**

**Sleep(3000);**

**system("cls");**

**int menuChoice;**

**while (true) {**

**system("color 3f");**

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

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

**cout << "\n\n\t\t\t1. Food";**

**cout << "\n\n\t\t\t2. Drinks";**

**cout << "\n\n\t\t\t3. Coffee";**

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

**cout << "\n\n\t\t\t5. Ice creams";**

**cout << "\n\n\t\t\t6. Fish";**

**cout << "\n\n\t\t\t7. Exit";**

**cout << "\n\n\n\n\t\t\tPlease choose a menu to order from: ";**

**cin >> menuChoice;**

**system("cls");**

**switch (menuChoice) {**

**case 1:**

**placeOrder(foodMenu);**

**break;**

**case 2:**

**placeOrder(drinkMenu);**

**break;**

**case 3:**

**placeOrder(coffeeMenu);**

**break;**

**case 4:**

**placeOrder(SweetsMenu);**

**break;**

**case 5:**

**placeOrder(icecreamsMenu);**

**break;**

**case 6:**

**placeOrder(fishMenu);**

**break;**

**case 7:**

**cout << "\n\n\t\t\tThank you for visiting AS Restaurants.";**

**return 0;**

**default:**

**cout << "\n\n\t\t\tInvalid choice. Please try again.";**

**Sleep(2000);**

**system("cls");**

**break;**

**}**

**}**

**}**

**CP PROJECT REPORT:**

**Report on Online Food Delivery System**

**Introduction**

This C++ program simulates an online food delivery system and includes a basic calculator for performing various mathematical operations. The program is designed to allow users to place orders from different categories of menus (food, drinks, coffee, sweets, ice creams, and fish) and calculate the total bill.

**Structure and Functionality**

**1. MenuItem Structure**

The MenuItem structure is defined to store information about each menu item, including its name and price:

struct MenuItem {

string name;

float price;

};

**2. Display Menu Function**

The displayMenu function displays the list of menu items:

void displayMenu(const vector<MenuItem>& menu) {

cout << "Menu:\n";

for (size\_t i = 0; i < menu.size(); ++i) {

cout << (i + 1) << ". " << menu[i].name << " ($" << fixed << setprecision(2) << menu[i].price << ")\n";

}

}

This function ensures that the prices are displayed with two decimal places using fixed and setprecision.

**3. Place Order Function**

The placeOrder function allows users to place an order from a selected menu:

void placeOrder(const vector<MenuItem>& menu) {

int choice;

double total = 0.0;

vector<MenuItem> order;

while (true) {

cout << "Enter the item number to add to your order:\n";

cout << "Press 0 to finish order\n";

displayMenu(menu);

cin >> choice;

if (choice == 0) {

break;

} else if (choice >= 1 && choice <= static\_cast<int>(menu.size())) {

order.push\_back(menu[choice - 1]);

total += menu[choice - 1].price;

cout << "Added " << menu[choice - 1].name << " to your order.\n";

Sleep(1000); // Simulating processing delay

} else {

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

Sleep(1000); // Simulating processing delay

}

}

cout << "Processing order..." << endl;

Sleep(2000); // Simulating processing delay

system("cls");

if (order.empty()) {

cout << "No items in the order.\n";

return;

}

cout << "Your order:\n";

for (size\_t i = 0; i < order.size(); ++i) {

cout << i + 1 << ". " << order[i].name << " ($" << fixed << setprecision(2) << order[i].price << ")\n";

}

cout << "Total: $" << fixed << setprecision(2) << total << endl;

}

This function allows users to add items to their order, displays the total cost, and handles invalid choices.

**Menu Initialization and User Interface**

**1. Menu Initialization**

The program initializes multiple vectors containing MenuItem structures for different categories of menus:

vector<MenuItem> foodMenu = {{"Burger", 5.99}, {"Pizza", 8.49}, ...};

vector<MenuItem> drinkMenu = {{"Mojito", 4.50}, {"Piña Colada", 5.25}, ...};

vector<MenuItem> coffeeMenu = {{"Espresso", 3.0}, {"Cappuccino", 4.5}, ...};

vector<MenuItem> SweetsMenu = {{"Chocolate Cake", 4.99}, {"Vanilla Cupcake", 2.50}, ...};

vector<MenuItem> icecreamsMenu = {{"Vanilla", 3.50}, {"Chocolate", 3.75}, ...};

vector<MenuItem> fishMenu = {{"Salmon", 4.3}, {"Tuna", 8}, ...};

Each vector contains a list of items available in that category.

**2. User Interface**

The user interface is created using a combination of console output, input, and system commands to clear the screen and change text color:

system("cls");

system("color 1f");

cout << setw(54) << "\t\t\t\t|Subject : computer programming|\n";

// More UI code...

This enhances the user experience by providing a visually appealing interface and smooth transitions.

**Conclusion**

This C++ program successfully combines an online food ordering system. The program demonstrates the use of structures, vectors, input/output handling, and various mathematical functions. The user interface is designed to be interactive and user-friendly, providing a smooth experience for placing orders .

Overall, this program serves as a practical example of combining multiple functionalities in a single application, showcasing the versatility of C++ programming in real-world scenarios.

Top of Form

Bottom of Form