A Project Report  
on  
**Restaurant Menu Calculator**



**By**

**Sharjeel Yasin** - BCY243067  
**Salman Haider** - BCY243058

A Project Report submitted to the  
**DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING**  
in partial fulfillment of the requirements for the degree of  
**BACHELORS IN CYBER SECURITY**

**Faculty of Engineering**  
**Capital University of Science & Technology, Islamabad**  
**January, 2025**

Copyright © 2025 by CUST Student

All rights reserved. Reproduction in whole or in part in any form requires the prior written permission of **SHARJEEL YASIN and SALMAN HAIDER** or designated representative.

**DECLARATION**

We declare that this report is our original work, except where otherwise acknowledged. This work has not been submitted for another degree or diploma at any university or institution.

Sharjeel Yasin  
Reg. No. BCY243067

Salman Haider  
Reg. No. BCY243058

January 2025

**Certificate of Approval**

It is certified that the project titled "Restaurant Menu Calculator" carried out by Sharjeel Yasin (BCY243067) and Salman Haider (BCY243058), under the supervision of SIR WAQAS, is fully adequate in scope and quality as a semester project for the degree of BS Cyber Security.

Supervisor: --------------------------------------

WAQAS AYUB KHAN

Lecturer

Department of Electrical and Computer Engineering

Faculty of Engineering

Capital University of Science & Technology,

HoD: --------------------------------------

Dr. Noor Mohammad Khan

Professor

Department of Electrical and Computer Engineering

Faculty of Engineering

Capital University of Science & Technology, Islamabad

**Acknowledgment**

We would like to extend our gratitude to “SIR WAQA AYUB KHAN” for their invaluable guidance throughout the development of this project. We also thank our group members and peers for their support and encouragement.

**Abstract**

The “Restaurant Menu Calculator” project is designed to provide a seamless and user-friendly interface for restaurant management. The system allows users to view an extensive menu, place orders, and calculate the total cost efficiently. Using C++, this project ensures simplicity, accuracy, and professional-grade features for practical use. Key functionalities include real-time menu display, order summary generation, and error handling for smooth user experience.

**Contents**

1. **Introduction**
   * Overview
   * Objectives
2. **Literature Review**
   * Related Work
   * Limitations of Existing Solutions
3. **Design and Implementation**
   * System Features
   * Functional Flow
4. **Results and Evaluation**
5. **Conclusion and Future Work**

**Chapter 1:**

**Introduction**

* 1. **Overview**

The **Restaurant Menu Calculator** aims to simplify restaurant operations by digitizing menu management and billing processes. It offers a professional and user-friendly interface for displaying a wide variety of dishes, capturing customer orders, and calculating total costs with precision.

**1.2 Objectives**

* To streamline order-taking and billing processes.
* To provide a professional, extensive menu interface.
* To ensure accuracy and efficiency in cost calculations.
* To incorporate error handling for improved user experience.

**Chapter 2:**

**Literature Review**

**2.1 Related Work**

Many restaurant management systems rely on complex hardware and software. This project adopts a lightweight yet efficient approach using **C++**, offering simplicity and cost-effectiveness.

**2.2 Limitations of Existing Solutions**

* High costs for advanced systems.
* Steep learning curves for beginners.
* Lack of personalization options for smaller businesses.

**Chapter 3:**

**Design and Implementation**

**3.1 System Features**

* **Menu Display**: Showcases 15 professionally curated dishes with pricing.
* **Order Input**: Accepts and validates customer orders and quantities.
* **Cost Calculation**: Automatically computes the total cost based on the order.

**3.2 Functional Flow**

The project uses a modular approach with functions for menu display, order capture, and total calculation. The use of **maps** and **vectors** ensures data organization and manipulation efficiency.

**Chapter 4:**

**Results and Evaluation**

The system was tested with multiple scenarios, including valid and invalid inputs, large orders, and edge cases. The results demonstrated:

* **Accuracy**: All computations matched expected outcomes.
* **User Experience**: The interface was intuitive and easy to navigate.

**Chapter 5:**

**Conclusion and Future Work**

**5.1 Conclusion**

This project successfully implements a **Restaurant Menu Calculator** that meets professional standards while maintaining simplicity. It serves as a reliable tool for restaurant management.

**5.2 Future Work**

* Integration with databases for dynamic menu updates.
* Development of a graphical user interface (GUI) for better interaction.
* Addition of features like tax calculation and discounts.

# Appendices

* Appendix A: Full C++ Code Listing

#include <iostream>

#include <string>

#include <vector>

#include <map>

#include <iomanip> // For formatting

using namespace std;

// Structure to represent a menu item

struct MenuItem {

string name;

double price;

};

// Function to display the menu

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

cout << "\n===== Welcome, Hope you were doing well.. =====\n";

cout << "Please have a look at our exquisite menu:\n\n";

cout << left << setw(5) << "No." << setw(30) << "Dish" << "Price ($)" << endl;

cout << string(40, '-') << endl;

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

cout << setw(5) << i + 1 << setw(30) << menu[i].name

<< fixed << setprecision(2) << menu[i].price << endl;

}

cout << endl;

}

// Function to get user input for order

map<int, int> getOrder() {

map<int, int> order;

int choice;

do {

cout << "Enter the item number to order (or 0 to finish): ";

cin >> choice;

if (choice != 0) {

if (cin.fail() || choice < 1 || choice > 15) {

cout << "Invalid choice. Please select a valid item number.\n";

cin.clear();

cin.ignore(1000, '\n');

continue;

}

cout << "Enter quantity: ";

int quantity;

cin >> quantity;

if (cin.fail() || quantity <= 0) {

cout << "Invalid quantity. Please enter a positive number.\n";

cin.clear();

cin.ignore(1000, '\n');

continue;

}

order[choice] += quantity; // Accumulate quantities if the item is selected again

}

} while (choice != 0);

return order;

}

// Function to calculate the total cost

double calculateTotal(const vector<MenuItem>& menu, const map<int, int>& order) {

double total = 0;

for (const auto& item : order) {

total += menu[item.first - 1].price \* item.second;

}

return total;

}

int main() {

// Define the menu items

vector<MenuItem> menu = {

{"Grilled Salmon", 12.50},

{"Steak with Mushroom Sauce", 15.00},

{"Caesar Salad", 8.00},

{"Margherita Pizza", 10.00},

{"Cheeseburger Deluxe", 9.00},

{"Pasta Alfredo", 11.50},

{"Chicken Tikka", 7.50},

{"Lamb Chops", 14.00},

{"Seafood Paella", 16.00},

{"Vegetable Stir Fry", 6.50},

{"Beef Tacos", 7.00},

{"Chicken Wings (Spicy)", 8.50},

{"Shrimp Tempura", 10.50},

{"Chocolate Lava Cake", 6.00},

{"Ice Cream Sundae", 5.50}

};

// Display the menu

displayMenu(menu);

// Get the order from the user

map<int, int> order = getOrder();

// Check if the order is empty

if (order.empty()) {

cout << "\nNo items ordered. Thank you for visiting Gourmet Bistro!\n";

return 0;

}

// Calculate the total cost

double total = calculateTotal(menu, order);

// Print the order summary

cout << "\n========== Order Summary ==========\n";

cout << left << setw(30) << "Dish" << setw(10) << "Quantity"

<< "Price ($)" << endl;

cout << string(50, '-') << endl;

for (const auto& item : order) {

cout << setw(30) << menu[item.first - 1].name

<< setw(10) << item.second

<< fixed << setprecision(2)

<< menu[item.first - 1].price \* item.second << endl;

}

cout << string(50, '-') << endl;

cout << setw(40) << "Total" << "$" << fixed << setprecision(2) << total << endl;

cout << "\nThank you for dining with us. Enjoy your meal!\n";

return 0;

}

* Appendix B: Additional Screenshots

