

FOOD Ordering System.

A Data Structure Project Report.



UNIVERSITY OF LAYYAH.

A Group Effort By:

**ALeena Khan (UL-BSITM-23-59), Arisha Khan (UL-BSITM-23-58),
Aiman Akbar (UL-BSITM-23-15), Hannia Fatima (UL-BSITM-23-44),
Saba Shehzad (UL-BSITM-23-33).**

Course Instructor: Faisal Hafeez

CLASS: BS IT (Morning) 3rd

Table of Contents

- 1. Acknowledgment..... 3
- 2. Summary..... 3
- 3. Introduction 3
- 4. Objectives 3
 - 4.1. Create a functional food ordering system:..... 3
 - 4.2. Implement basic customer service features: 3
 - 4.3. Demonstrate the use of C++ concepts:..... 4
 - 4.4. Provide a basic user experience: 4
- 5. Tools and Technologies 4
- 6. Methodology 4
 - 6.1. Problem Definition:..... 4
 - 6.2. Design:..... 4
 - 6.3. User Interface: 5
 - 6.4. Implementation: 5
 - 6.5. Testing: 5
 - 6.6. Documentation:..... 5
- 7. Implementation 5
- 8. Working and Features..... 6
- 9. Challenges..... 8
- 10. Conclusion 9
- 11. References..... 9

1. Acknowledgment

We extend our deepest gratitude to our instructor, Mr. Faisal Hafeez, for his invaluable guidance in selecting an appropriate project and programming language. His advice to focus on a project involving data structures and GUI development inspired us to undertake the Numbers Puzzle Game. We also thank our group members for their dedication and collaboration, which made this project a success.

2. Summary

The project demonstrates a basic implementation of a food ordering system using fundamental C++ concepts like arrays, functions, and user input/output. It provides a simple framework for handling orders, customer inquiries, and basic order tracking.

3. Introduction

This C++ project simulates a basic food ordering system for a restaurant named 'Layyah Food'. The system allows users to place orders from a menu with various food categories, including pizzas, burgers, sandwiches, rolls, and biryanis. Customers can also inquire about their order status or contact customer care for general inquiries. The system incorporates key features such as user interaction through a menu-driven interface, order processing with price calculations, and basic payment options.

4. Objectives

The primary objectives of the "Layyah Food" project are:

4.1. Create a functional food ordering system:

- **Allow users to place orders for various food items (pizzas, burgers, etc.).**
- **Provide a menu-driven interface for easy navigation.**
- **Enable users to select items, specify quantities, and calculate the total bill.**

4.2. Implement basic customer service features:

- **Offer a "Customer Assistant" option for inquiries.**
- **Allow users to inquire about their order status using an order number.**
- **Provide a general inquiry option for customer support.**

4.3. Demonstrate the use of C++ concepts:

- **Utilize arrays to store food item names and prices.**
- **Implement functions for modularity and code reusability (e.g., displayMainMenu(), takeOrder(), customerCare()).**
- **Handle user input and output effectively.**

4.4. Provide a basic user experience:

- **Present information in a clear and user-friendly manner.**
- **Guide users through the ordering process with appropriate prompts and messages.**

In essence, the project aims to create a simple yet functional food ordering system while showcasing fundamental C++ programming concepts.

5. Tools and Technologies

The tools and technologies used in this project include:

1. **Programming Language: C++**
2. **Standard Input/Output Library (iostream)**
3. **String Library (string)**
4. **Arrays**

This project demonstrates a basic application of C++ programming concepts to create a simple interactive food ordering system.

6. Methodology

This C++ project follows a structured approach to develop a simple food ordering system:

6.1. Problem Definition:

- **Clearly define the scope and functionalities of the system:**
- **Allow users to place orders from a menu.**
- **Calculate and display the total bill.**
- **Provide customer care options (order inquiry, general inquiry).**
- **Implement basic payment options (cash on delivery, online).**

6.2. Design:

1. Data Structures:

- **Use arrays to store food item names and their corresponding prices for each category.**

2. Functions:

- **Divide the program into well-defined functions for better organization and reusability:**
- **displayMainMenu():** Display the main menu options to the user.
- **displaySubMenu():** Display the available items within a specific category.
- **takeOrder():** Handle order placement, including item selection, quantity input, and order summary.
- **customerCare():** Handle customer inquiries (order status, general inquiries).
- **paymentMethod():** Present payment options to the user and handle payment processing.

6.3. User Interface:

- **Design a user-friendly interface with clear prompts and messages to guide users through the ordering process.**

6.4. Implementation:

- **Write the C++ code, implementing the defined functions and data structures.**
- **Handle user input and output effectively using cin and cout.**
- **Incorporate conditional statements (if-else) and loops (do-while) to control program flow and handle user choices.**

6.5. Testing:

- **Thoroughly test the program with various input scenarios to ensure it functions correctly and handles errors gracefully.**
- **Test all menu options, order placement, customer care, and payment options.**
- **Check for unexpected behavior or potential issues.**

6.6. Documentation:

- **Add comments within the code to explain its functionality.**
- **Create a brief documentation or a README file to describe the project's purpose, features, and how to use it.**

This methodology provides a structured approach to develop the "Layyah Food" ordering system using C++. By following these steps, you can create a functional and well-organized program.

7. Implementation

The important implementations in the project are following below:

1. Use of Functions:

- The code effectively utilizes functions (displayMainMenu(), displaySubMenu(), takeOrder(), customerCare(), paymentMethod()) to break down the program into smaller, reusable modules.

This improves code organization, readability, and maintainability. Functions make the code easier to understand, debug, and modify.

2. Data Structures:

The use of arrays (pizzas, pizzaPrices, etc.) to store food item names and their corresponding prices demonstrates the use of a basic data structure to organize and manage information efficiently

User Input and Output:

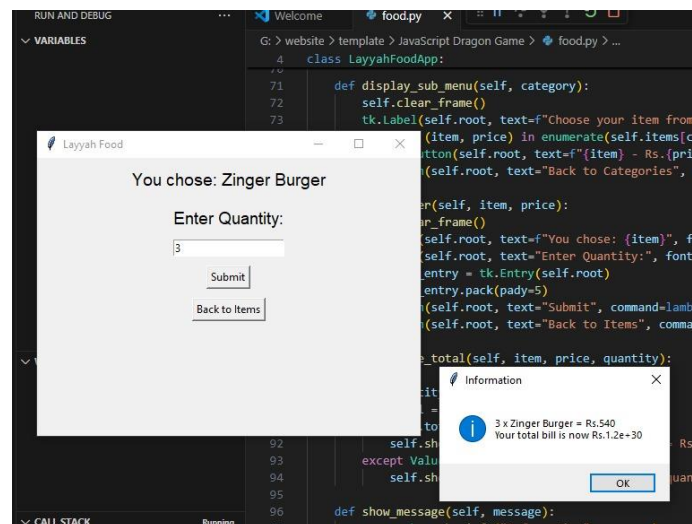
The code effectively handles user input using cin and provides informative output to the user using cout. This includes:

- Displaying menus and sub-menus.
- Prompting the user for input (choices, quantities, etc.).
- Providing feedback on user actions (order confirmation, payment options).



4. Control Flow:

- The use of switch statements and do-while loops to handle different user choices and allow for multiple orders within a single session demonstrates effective control flow.
- Conditional statements (if-else) are used to validate user input and handle different scenarios.



5. Basic Object-Oriented Concepts (though not fully implemented):

- While not a full-fledged object-oriented approach, the use of functions can be seen as a step towards modularity and encapsulation, which are key concepts in object-oriented programming.

6. User Interaction:

The code provides a basic interactive user interface with a menu-driven system, making it easier for users to navigate and interact with the program.

8. Working and Features

A. Order Inquiry

Users can browse the menu to view available food items.

Each item will have a description, price, and an option to view additional details (e.g., ingredients, dietary restrictions).

Users can search for specific items or filter by categories (e.g., appetizers, main courses, desserts).



B. Customer Assistance

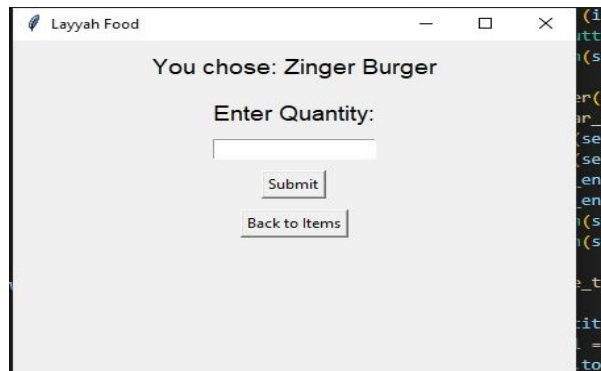
- A built-in chatbot or customer service feature will be available to answer common queries.
- Users can ask about the status of their orders, delivery times, and any issues they may encounter.
- FAQs will be included to help users with common questions, enhancing their experience.

C. Placing Orders

Users can select items from the menu and specify the quantity for each.

After selecting items, the system will display a summary of the order, including total cost.

Users will have the option to modify their order before finalizing it.



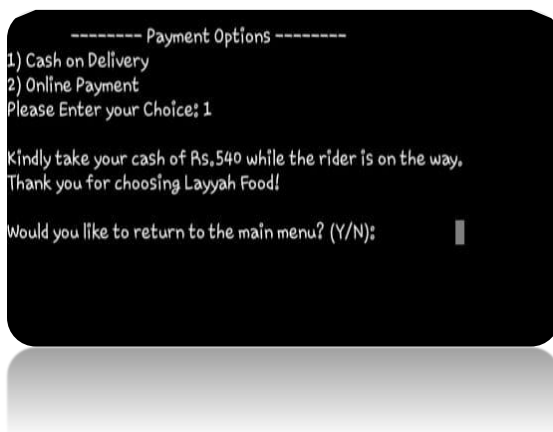
D. Menu

- The menu will be dynamic and can be updated by the restaurant management.
- Each food item will include:
 - Name
 - Description
 - Price

Additional features could include daily specials, combo offers, and seasonal items.

E. Payment Options

- Users will have two primary payment methods:
 - a. Online Payment: Secure payment gateway integration for credit/debit cards and digital wallets.
 - b. Cash on Delivery: Users can opt to pay in cash upon receiving their order.



9. Challenges

1. Data Management:

- Challenge: Difficulty in updating menu data (items, prices) without code changes.

- Solution: Use external data files or databases for flexible menu management.
2. **Order Tracking:**
- Challenge: Limited order tracking capabilities.
 - Solution: Implement a simple database to store order details for better tracking.
3. **Payment Integration:**
- Challenge: Basic payment simulation; integrating with real payment gateways requires further development.
 - Solution: Explore secure payment gateway libraries or APIs for advanced payment integration. This version focuses on the core challenges and their concise solutions.

10. Conclusion

This C++ project successfully implements a basic food ordering system. It demonstrates key programming concepts such as arrays, functions, and user input/output. Users can navigate through a menu, select food items, and calculate their total bill. The system also includes basic customer care features like order inquiry and general inquiries. While this project provides a simplified model, it effectively showcases how these core programming concepts can be applied to create an interactive application with practical functionalities.

11. References

- Sources YT tutorials
- AI
- website