

A
PROJECT REPORT
ON
“RESTAURANT ORDER MANAGEMENT SYSTEM”

SUBMITTED BY:

Mr. Jay Rajendra Ushir (2124UCEM1021)

SUBJECT:

Programming in Problem

Solving using C++

Under the guidance of

Miss. Ishwari Tirse



Department of Computer Science and Engineering

Sanjivani Rural Education Society's

SANJIVANI UNIVERSITY

KOPARGAON – 423603, DIST : AHMEDNAGAR

2024-2025

INDEX

| SR. NO | CONTENT | PAGE NO. |
|-------------------|---------------------|-----------------|
| 1. | INTRODUCTION | 3 |
| 2. | CODE | 4 |
| 3. | OUTPUT | 8 |
| 4. | CONCLUSION | 10 |

INTRODUCTION

The Restaurant Order Management System is designed to help streamline the ordering process in a restaurant, making it easier to manage customer orders, kitchen processing, and billing. In a traditional setup, orders are taken manually and often get delayed due to lack of coordination between the waitstaff and kitchen staff. This project introduces an efficient way to manage orders, reduce human error, and improve customer satisfaction.

This system allows users to:

View the menu with prices

Place orders by selecting items and quantities

Track the order status (e.g., pending, completed)

Generate and print the bill

CODE

```
#include <iostream>
#include <vector>
#include <iomanip>
using namespace std;

// MenuItem class to represent each item in the menu
class MenuItem {
public:
    string name;
    double price;

    MenuItem(string itemName, double itemPrice) : name(itemName),
price(itemPrice) {}
};

// OrderItem class to represent each ordered item
class OrderItem {
public:
    MenuItem item;
    int quantity;

    OrderItem(MenuItem mItem, int qty) : item(mItem), quantity(qty) {}

    double getTotalPrice() const {
        return quantity * item.price;
    }
};

// Order class to manage the list of ordered items
class Order {
public:
    vector<OrderItem> items;

    void addItem(MenuItem menuItem, int quantity) {
        items.push_back(OrderItem(menuItem, quantity));
    }

    void printOrderSummary() const {
        cout << fixed << setprecision(2);
        double total = 0;
        cout << "\nOrder Summary:\n";
        cout << "-----\n";
    }
};
```

```

        for (const auto& orderItem : items) {
            double itemTotal = orderItem.getTotalPrice();
            cout << orderItem.item.name << " x" << orderItem.quantity
                << " - $" << itemTotal << endl;
            total += itemTotal;
        }
        cout << "-----\n";
        cout << "Total Bill: $" << total << endl;
    }
};

// Function to display the menu
void displayMenu(const vector<MenuItem>& menu) {
    cout << "Menu:\n";
    cout << "-----\n";
    for (int i = 0; i < menu.size(); i++) {
        cout << i + 1 << ". " << menu[i].name << " - $" << menu[i].price << endl;
    }
    cout << "-----\n";
}

int main() {
    // Sample menu
    vector<MenuItem> menu = {
        MenuItem("Pizza", 10.99),
        MenuItem("Burger", 6.49),
        MenuItem("Pasta", 8.99),
        MenuItem("Salad", 4.99)
    };

    Order order;
    int choice, quantity;

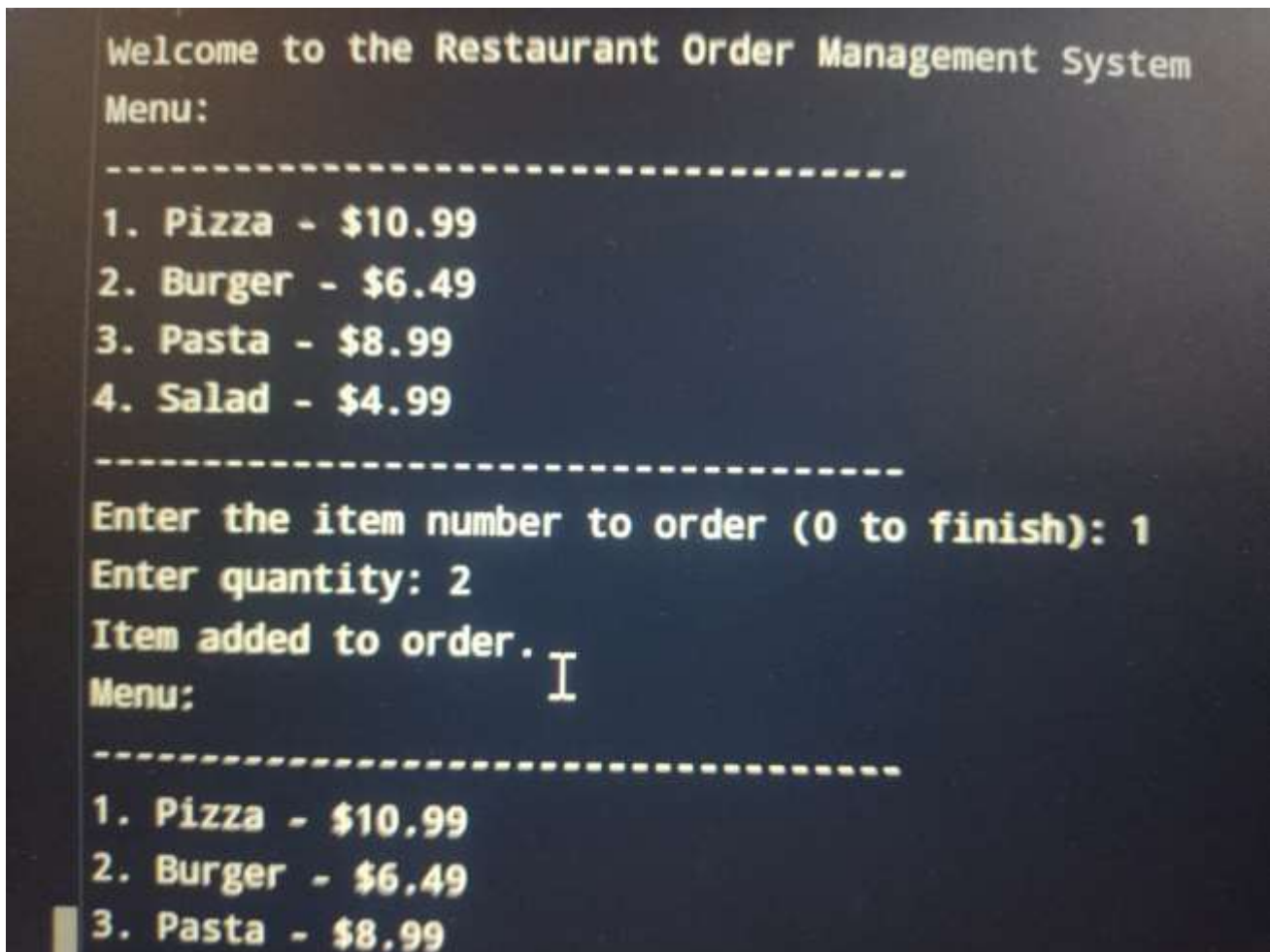
    cout << "Welcome to the Restaurant Order Management System\n";
    while (true) {
        displayMenu(menu);
        cout << "Enter the item number to order (0 to finish): ";
        cin >> choice;
        if (choice == 0) break;
        if (choice < 1 || choice > menu.size()) {
            cout << "Invalid choice. Please try again.\n";
            continue;
        }

        cout << "Enter quantity: ";
        cin >> quantity;
    }
}

```

```
        order.addItem(menu[choice - 1], quantity);  
        cout << "Item added to order.\n";  
    }  
  
    order.printOrderSummary();  
    cout << "Thank you for your order!\n";  
    return 0;  
}
```

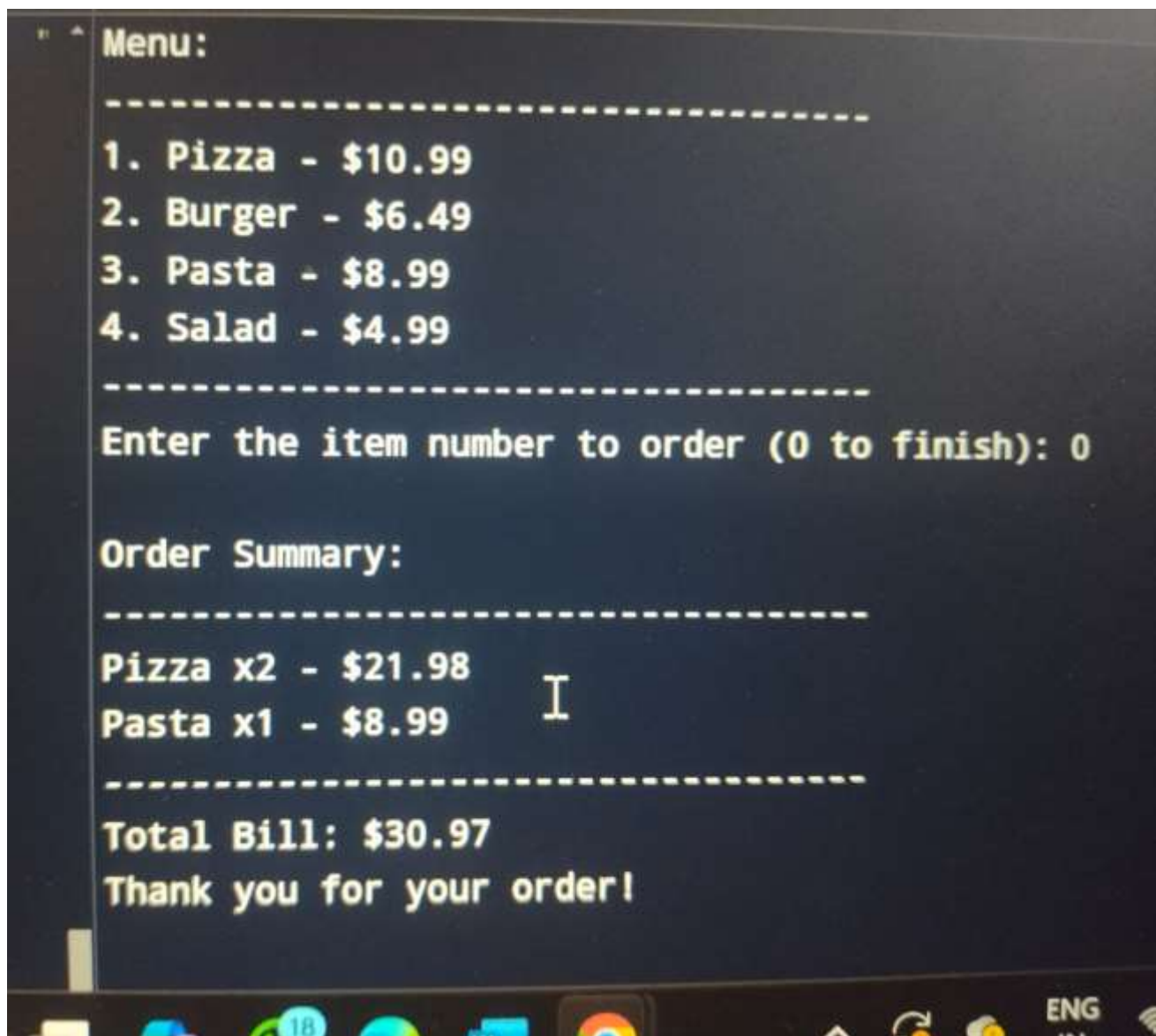
OUTPUT



```
Welcome to the Restaurant Order Management System  
Menu:  
-----  
1. Pizza - $10.99  
2. Burger - $6.49  
3. Pasta - $8.99  
4. Salad - $4.99  
-----  
Enter the item number to order (0 to finish): 1  
Enter quantity: 2  
Item added to order.  
Menu:  
-----  
1. Pizza - $10.99  
2. Burger - $6.49  
3. Pasta - $8.99
```

```
inish): " 2. Burger - $6.49
          3. Pasta - $8.99
          4. Salad - $4.99
          -----
          Enter the item number to order (0 to finish): 3
          Enter quantity: 1
          Item added to order.
          Menu:
          -----
          1. Pizza - $10.99
          2. Burger - $6.49
          3. Pasta - $8.99
          4. Salad - $4.99
          -----
          Enter the item number to order (0 to finish): 0

          Order Summary:
```



CONCLUSION

programming and OOP principles in a real-world scenario. By breaking down the functionality into classes like MenuItem, OrderItem, and Order, the project is modular, maintainable, and extendable. This system enhances order accuracy, reduces processing time, and improves the dining experience for both customers and restaurant staff. Future improvements could include adding user authentication, saving order history, and integrating the system with external billing or inventory software.