****

**Project Report**

**Course Code: CSE 124**

**Course Title: Data Structures Lab**

**Project Title: Food order Management in a restaurant.**

**Submitted To**

**Md. Nahid Hasan**

**Lecturer**

**Department of Computer Science And Technology**

**Daffodil International University**

**Submitted By**

**Name: Md Naimur Rahman**

**ID: 0242310005101275**

**Section: 64\_Q**

**Department of Computer Science And Technology**

**Daffodil International University**

**Table of Contents**

1. **Introduction……………………………………………………..…… 2**
2. **Uses……………………………………………………….……….………… 2**
3. **2. Methodology.……………………………………….…….. 2**
4. **Code Implementation………………………….……..……… 3**
5. **Results ……………………………………………………..……… 6**
6. **Challenges Faced and Probable Solutions: …………………… 8**
7. **Future Scopes ……………………………………………………………….. 9**
8. **Conclusion …………………………………………….……………………… 9**
9. **References ……………………………………………………………………. 10**

**Introduction:**

The Restaurant Order Management System is designed to streamline the process of managing food orders in a restaurant. The system keeps track of incoming orders, serves them in a first-come-first-serve basis, and provides information on the total number of orders received, served, and waiting.

**Uses:**

The Restaurant Order Management System can be used in various real-world scenarios to enhance the efficiency and organization of food order processing in a restaurant. Here are some potential uses:

* **Small to Medium-sized Restaurants:**

The system is suitable for small to medium-sized restaurants that want to automate and streamline their order management process.

* **Quick Service Restaurants (QSRs):**

Fast-food restaurants or QSRs with high customer turnover can benefit from the system to manage a large number of orders efficiently.

* **Cafes and Coffee Shops:**

Cafes and coffee shops can use the system to manage both food and beverage orders, helping staff prioritize and serve customers in a timely manner.

* **Food Trucks:**

Mobile food vendors, such as food trucks, can utilize the system to keep track of incoming orders and maintain an organized workflow, especially during peak hours.

* **Training Environments:**

The system can be used in training environments for restaurant staff to simulate order processing scenarios and practice serving customers.

**Methodology:**

The project is structured around the following key elements:

* **Data Structure:** Utilization of structures in C to represent a contact with attributes such as food number, size, and quantity.
* **Menu-Driven Interface:** Implemented a menu-driven approach to allow users to interact with the system seamlessly.
* **Modular Programming:** Divided the project into functions for specific tasks, enhancing code readability and maintainability.

**Code Implementation:**

#include<stdio.h>

#include<stdlib.h>

struct food\_order

{

    int food\_number;

    char size;

    int quantity;

    struct food\_order \*link;

};

typedef struct food\_order node;

node \*front=NULL,\*end=NULL;

int total=0,served=0;

int isempty()

{

    if(front==NULL && end==NULL)

    {

        return 1;

    }

    else

    {

        return 0;

    }

}

void place\_order(int fd\_number,char s,int qnty)

{

    node \*new1=malloc(sizeof(node));

    if(isempty())

    {

        front=new1;

        end=new1;

        end->food\_number=fd\_number;

        end->size=s;

        end->quantity=qnty;

        end->link=NULL;

        total++;

    }

    else

    {

        new1->food\_number=fd\_number;

        new1->size=s;

        new1->quantity=qnty;

        end->link=new1;

        end=new1;

        end->link=NULL;

        total++;

    }

}

void serve\_order()

{

    if(!isempty())

    {

        if(front==end)

        {

            printf("\nFood Number: %d\n",front->food\_number);

            printf("Size: %c\n",front->size);

            printf("Quantity: %d\n",front->quantity);

            printf("Order has Served.\n");

            front=NULL;

            end=NULL;

            served++;

        }

        else

        {

            printf("\nFood Number: %d\n",front->food\_number);

            printf("Size: %c\n",front->size);

            printf("Quantity: %d\n",front->quantity);

            printf("Order has Served.\n");

            front=front->link;

            served++;

        }

    }

    else

    {

        printf("\nPlease Order first.\n");

    }

}

void display()

{

    printf("\nTotal Order Received: %d\n",total);

    printf("Total Order Served: %d\n",served);

    printf("Order Waiting: %d\n",total-served);

}

int main()

{

    printf("\*-\*-Welcome our Restaurant-\*-\*\n");

    while(1)

    {

        printf("\n  \*\*\*Food Order\*\*\*\n");

        printf("\*-------------------\n");

        printf("    1. Place Order\n");

        printf("    2. Serve Order\n");

        printf("    3. Display Order\n");

        printf("    4. Exit\n");

        printf("-------------------\*\n");

        int choice,x;

        printf("Enter your Choice: ");

        scanf("%d",&choice);

        if(choice==4)

        {

            printf("\nThank you. Will come again.\n");

            break;

        }

        else if(choice==1)

        {

            int food\_number;

            char size;

            int quantity;

            printf("\nEnter food number: ");

            scanf("%d",&food\_number);

            printf("Enter Size(S/M/L): ");

            scanf(" %c",&size);

            printf("Enter the quantity: ");

            scanf("%d",&quantity);

            place\_order(food\_number,size,quantity);

        }

        else if(choice==2)

        {

            serve\_order();

        }

        else if(choice==3)

        {

            display();

        }

        else

        {

            printf("\nInvalid choice. Please enter a valid option.\n");

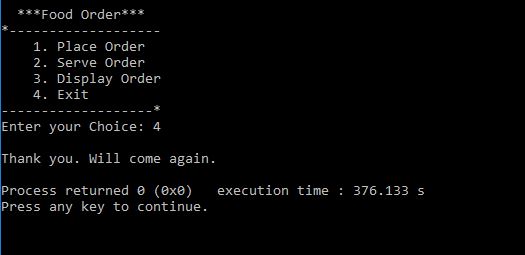
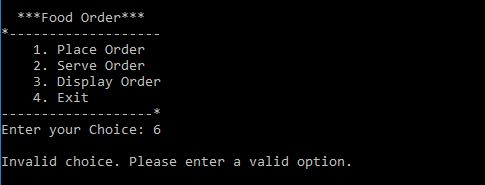
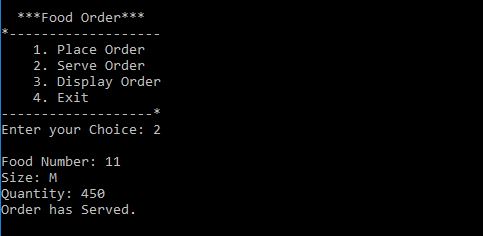
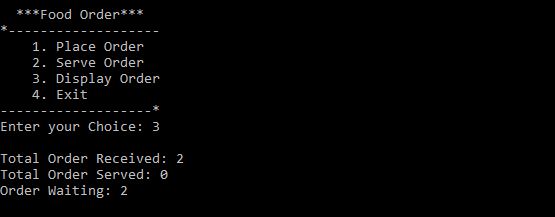
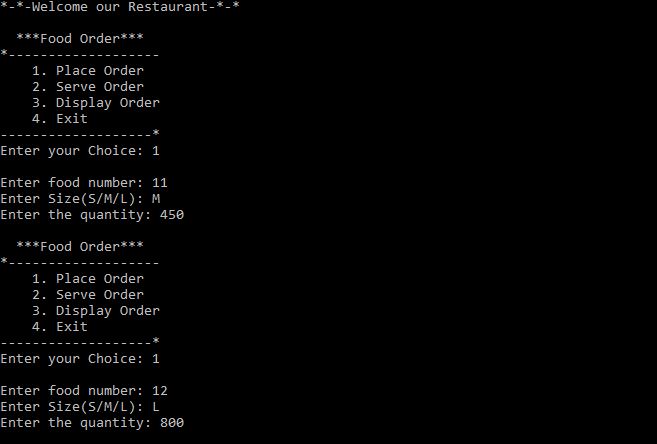
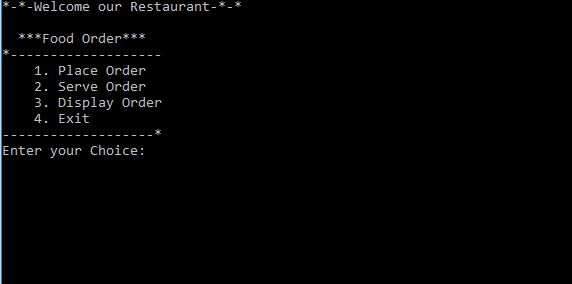
        }

    }

    return 0;

}

**Results:**



**Challenges Faced and Probable Solutions:**

**1.Order Complexity:**

**Challenge:** Managing complex orders with multiple items, customizations, and special requests.

**Solution:** Implement a flexible order structure that can handle various components, quantities, and customer preferences. Use a well-designed user interface to facilitate easy order entry.

**2. Real-Time Updates:**

**Challenge:** Providing real-time updates on order status and availability of menu items.

**Solution:** Implement a robust notification system to update customers and staff on order status changes. Utilize technologies like WebSocket for real-time communication.

**3. Scalability:**

**Challenge:** Handling a growing number of orders and users as the restaurant expands.

**Solution:** Design the system with scalability in mind, considering database optimization, load balancing, and cloud-based solutions. Regularly monitor system performance and scale resources as needed.

**4. Data Security:**

**Challenge:** Protecting sensitive customer and order data from security threats.

**Solution:** Implement secure coding practices, encrypt sensitive data, and use authentication and authorization mechanisms. Regularly update security protocols to address emerging threats.

**5. Feedback Handling:**

**Challenge:** Effectively managing and responding to user feedback.

**Solution:** Set up a feedback system for users to report issues or suggest improvements. Establish a feedback loop to regularly assess and address user concerns. Prioritize and implement enhancements based on user needs.

**Future Scopes:**

The current implementation provides a basic framework for order management. Future enhancements could include:

* **Billing System:** Integrate a billing system to calculate and display the total cost of each order.
* **Menu Management:** Extend the system to include a menu with prices and options for customers to choose from.
* **User Authentication:** Implement user authentication for staff members to access and manage the system securely.
* **Database Integration:** Store order data in a database for persistence and historical tracking.

**Conclusion:**

The Restaurant Order Management System offers a simple and effective solution for handling food orders in a restaurant. It can be customized and extended based on specific requirements and serves as a foundation for future enhancements. The project promotes efficiency in order processing, providing valuable insights into the restaurant's operational status.

**References:**

**Websites:**

<https://www.geeksforgeeks.org/>

https://www.programiz.com/dsa/linked-list/