

Unified Food Ordering & Delivery System

1. Task Description

The project has been split between two students as follows:

Module	Student A	Student B
Core CLI & Navigation	CLI loops, input parsing	Utility functions, color macros
Menu Parsing	File I/O and parsing	API design
Cart & Order Logic	Cart class & operations	Data structures definitions
File Persistence	CSV read/write	Format definitions
Customization Flow	Pizza customization prompts	Validation logic
History Display	Pretty-printing records	CSV utilities
Build & Documentation	Makefile	README & code comments

2. Data Structures & Class Relationships

2.1 Class Definitions

```
class CartItem {
public:
    std::string itemName;
    double unitPrice;
    int quantity;
    std::string customizations;
};

class Cart {
public:
    std::vector<CartItem> items;
    void addItem(const CartItem& item);
    double getTotal() const;
};

class DeliveryInfo {
public:
    std::string name;
    std::string address;
    std::string phone;
};

class OrderRecord {
public:
    int orderID;
    DeliveryInfo customer;
    Cart cart;
    double totalPrice;
    std::string dateTime;
};
```

2.2 Class Relationships

- CartItem → Cart: A Cart contains a collection (std::vector) of CartItem objects.
- Cart → OrderRecord: An OrderRecord aggregates a Cart instance representing the items in that order.

- DeliveryInfo → OrderRecord: An OrderRecord contains a DeliveryInfo object for customer details.
- OrderRecord → IO Module: OrderRecord objects are serialized to and from CSV via the IO module.
- Menu Module → CartItem: Menu items loaded from text files are used to instantiate CartItem objects.

3. File Structure

- Menu files (*.txt): One line per item: \$ (e.g., Pizza Diavola\$12.50)
- Order history (orders_history.csv):
 - Header: orderID,customerName,address,phone,dateTime,totalPrice,items
 - Records example: 1,"Alice","123 Main St","071234567","16/05/2025 14:22",45.50,"Pizza Diavola x1; Coke x2"

4. Interacting with Executables

Below are the available commands and their parameters:

Command	Description
./foodapp	Runs in interactive mode (no parameters)
./foodapp --order <RestaurantName> --item <ItemName> -p <price> -a <address> -c <phone> -t <time> -s <items>	Places an order off-online with an optional customization string
./foodapp --history	Displays past orders from the CSV history file
./foodapp --help	Shows usage information and available flags