ECOMMERCE SYTEM PROJECT

CODE :

```cpp
#include <iostream>

#include <vector>

#include <string>

using namespace std;

class Product {

protected:

    int id;

    string name;

    double price;

public:

    Product(int id, string name, double price) : id(id), name(name), price(price) {}

    virtual void displayDetails() const = 0;

    int getId() const { return id; }

    string getName() const { return name; }

    double getPrice() const { return price; }

    virtual ~Product() {}

};

class Electronics : public Product {

    int warranty;

public:

    Electronics(int id, string name, double price, int warranty) : Product(id, name, price),
warranty(warranty) {}

    void displayDetails() const override {

        cout << "Electronics [ID: " << id << ", Name: " << name << ", Price: PKR " << price << ", Warranty: "
<< warranty << " months]" << endl;

    }

};

class Clothing : public Product {

    string size;
```

```cpp
    string material;
public:
    Clothing(int id, string name, double price, string size, string material) : Product(id, name, price), size(size), material(material) {}
    void displayDetails() const override {
        cout << "Clothing [ID: " << id << ", Name: " << name << ", Price: PKR " << price << ", Size: " << size << ", Material: " << material << "]" << endl;
    }
};
class Food : public Product {
    string expiryDate;
public:
    Food(int id, string name, double price, string expiryDate) : Product(id, name, price), expiryDate(expiryDate) {}
    void displayDetails() const override {
        cout << "Food [ID: " << id << ", Name: " << name << ", Price: PKR " << price << ", Expiry Date: " << expiryDate << "]" << endl;
    }
};
class Stationery : public Product {
    string brand;
public:
    Stationery(int id, string name, double price, string brand) : Product(id, name, price), brand(brand) {}
    void displayDetails() const override {
        cout << "Stationery [ID: " << id << ", Name: " << name << ", Price: PKR " << price << ", Brand: " << brand << "]" << endl;
    }
};
class Books : public Product {
    string author;
public:
    Books(int id, string name, double price, string author) : Product(id, name, price), author(author) {}
```

```cpp
    void displayDetails() const override {

        cout << "Books [ID: " << id << ", Name: " << name << ", Price: PKR " << price << ", Author: " <<
author << "]" << endl;

    }

};

class GirlsMakeup : public Product {

    string brand;

public:

    GirlsMakeup(int id, string name, double price, string brand) : Product(id, name, price), brand(brand)
{}

    void displayDetails() const override {

        cout << "Girls' Makeup [ID: " << id << ", Name: " << name << ", Price: PKR " << price << ", Brand: "
<< brand << "]" << endl;

    }

};

class Cart {

    vector<Product*> products;

public:

    void addProduct(Product* product) {

        products.push_back(product);

    }


    void deleteProduct(int productId) {

        for (auto it = products.begin(); it != products.end(); ++it) {

            if ((*it)->getId() == productId) {

                delete *it;

                products.erase(it);

                cout << "Product with ID " << productId << " has been removed from the cart." << endl;

                return;

            }

        }

        cout << "Product with ID " << productId << " not found in the cart." << endl;
```

```cpp
    }

    void updateProduct(int productId, int newId, string newName, double newPrice) {
        for (auto& product : products) {
            if (product->getId() == productId) {
                Product* updatedProduct = createProduct(newId, newName, newPrice);
                delete product;
                product = updatedProduct;
                cout << "Product with ID " << productId << " has been updated." << endl;
                return;
            }
        }
        cout << "Product with ID " << productId << " not found in the cart." << endl;
    }

    void showCart() const {
        if (products.empty()) {
            cout << "Cart is empty." << endl;
            return;
        }
        cout << "Cart Contents:" << endl;
        for (const auto& product : products) {
            product->displayDetails();
        }
    }

    Product* createProduct(int id, string name, double price) const {
        if (name == "Refrigerator" || name == "Air Conditioner" || name == "Generator" || name == "Food
Processor" || name == "Mobile Phone" || name == "LED TV") {
            return new Electronics(id, name, price, 0);
        } else if (name == "Shirt" || name == "Jeans" || name == "Jacket" || name == "T-shirt" || name ==
"Kurta" || name == "Shalwar Kameez") {
            return new Clothing(id, name, price, "", "");
```

```cpp
        } else if (name == "Apple" || name == "Bread" || name == "Milk" || name == "Eggs" || name ==
"Biryani" || name == "Naan") {

            return new Food(id, name, price, "");

        } else if (name == "Notebook" || name == "Pen" || name == "Pencil" || name == "Eraser" || name ==
"Marker" || name == "Highlighter") {

            return new Stationery(id, name, price, "");

        } else if (name == "Novel" || name == "Textbook" || name == "Magazine" || name == "Comic" || name
== "Biography" || name == "Poetry Book") {

            return new Books(id, name, price, "");

        } else if (name == "Lipstick" || name == "Foundation" || name == "Mascara" || name == "Blush" ||
name == "Eyeliner" || name == "Nail Polish") {

            return new GirlsMakeup(id, name, price, "");

        } else {

            return nullptr;

        }

    }

    void printReceipt(const string& customerName, int customerId) const {

        double total = 0;

        cout << "\nReceipt\n";

        cout << "Customer Name: " << customerName << "\n";

        cout << "Customer ID: " << customerId << "\n";

        cout << "----------------------------------\n";

        for (const auto& product : products) {

            product->displayDetails();

            total += product->getPrice();

        }

        cout << "----------------------------------\n";

        cout << "Total: PKR " << total << "\n";

        cout << "Thank you for shopping with us!\n";

    }

    ~Cart() {
```

```cpp
        for (auto& product : products) {

            delete product;

        }

    }

};

int displayMenu() {

    int choice;

    cout << "\nMenu:\n";

    cout << "1. View and Add Electronics\n";

    cout << "2. View and Add Clothing\n";

    cout << "3. View and Add Food\n";

    cout << "4. View and Add Stationery\n";

    cout << "5. View and Add Books\n";

    cout << "6. View and Add Girls' Makeup\n";

    cout << "7. Show Cart\n";

    cout << "8. Delete Product from Cart\n";

    cout << "9. Update Product in Cart\n";

    cout << "10. Checkout\n";

    cout << "Enter your choice: ";

    cin >> choice;

    while (cin.fail() || choice < 1 || choice > 10) {

        cin.clear();

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

        cout << "Invalid input. Please enter a number between 1 and 10: ";

        cin >> choice;

    }

    return choice;

}

void displayAndAddProducts(const vector<Product*>& products, Cart& cart) {

    int choice;

    cout << "Available Products:\n";
```

```cpp
        for (const auto& product : products) {

            product->displayDetails();

        }

        cout << "Enter the ID of the product to add to the cart (0 to cancel): ";

        cin >> choice;

        while (cin.fail() || choice < 0) {

            cin.clear();

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

            cout << "Invalid input. Please enter a valid product ID or 0 to cancel: ";

            cin >> choice;

        }

        if (choice != 0) {

            for (const auto& product : products) {

                if (product->getId() == choice) {

                    cart.addProduct(cart.createProduct(product->getId(), product->getName(), product->getPrice()));

                    cout << product->getName() << " added to the cart." << endl;

                    return;

                }

            }

            cout << "Product with ID " << choice << " not found." << endl;

        }

}

void deleteProductFromCart(Cart& cart) {

    int productId;

    cout << "Enter the ID of the product to delete from the cart: ";

    cin >> productId;

    while (cin.fail() || productId < 0) {

        cin.clear();

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

        cout << "Invalid input. Please enter a valid product ID: ";
```

```cpp
        cin >> productId;
    }
    cart.deleteProduct(productId);
}
void updateProductInCart(Cart& cart) {
    int productId, newId;
    string newName;
    double newPrice;
    cout << "Enter the ID of the product to update in the cart: ";
    cin >> productId;
    while (cin.fail() || productId < 0) {
        cin.clear();
        cin.ignore(1000, '\n');
        cout << "Invalid input. Please enter a valid product ID: ";
        cin >> productId;
    }
    cout << "Enter the new product ID: ";
    cin >> newId;
    while (cin.fail() || newId < 0) {
        cin.clear();
        cin.ignore(1000, '\n');
        cout << "Invalid input. Please enter a valid product ID: ";
        cin >> newId;
    }
    cout << "Enter the new product name: ";
    cin.ignore();
    getline(cin, newName);
    while (newName.empty()) {
        cout << "Invalid input. Please enter a valid product name: ";
        getline(cin, newName);
    }
```

```cpp
        cout << "Enter the new product price: ";

        cin >> newPrice;

        while (cin.fail() || newPrice < 0) {

            cin.clear();

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

            cout << "Invalid input. Please enter a valid product price: ";

            cin >> newPrice;

        }

        cart.updateProduct(productId, newId, newName, newPrice);

}

int main() {

    vector<Product*> electronics = {

        new Electronics(1, "Refrigerator", 50000, 12),

        new Electronics(2, "Air Conditioner", 60000, 24),

        new Electronics(3, "Generator", 70000, 36),

        new Electronics(4, "Food Processor", 10000, 12),

        new Electronics(5, "Mobile Phone", 40000, 12),

        new Electronics(6, "LED TV", 80000, 24)

    };

    vector<Product*> clothing = {

        new Clothing(7, "Shirt", 2000, "M", "Cotton"),

        new Clothing(8, "Jeans", 3000, "L", "Denim"),

        new Clothing(9, "Jacket", 4000, "XL", "Leather"),

        new Clothing(10, "T-shirt", 1500, "S", "Polyester"),

        new Clothing(11, "Kurta", 2500, "M", "Cotton"),

        new Clothing(12, "Shalwar Kameez", 3500, "L", "Cotton")

    };

    vector<Product*> food = {

        new Food(13, "Apple", 100, "2023-12-31"),

        new Food(14, "Bread", 50, "2023-11-01"),

        new Food(15, "Milk", 80, "2023-10-15"),
```

```cpp
        new Food(16, "Eggs", 120, "2023-09-20"),

        new Food(17, "Biryani", 200, "2023-08-30"),

        new Food(18, "Naan", 30, "2023-07-10")

    };

    vector<Product*> stationery = {

        new Stationery(19, "Notebook", 100, "Dawn"),

        new Stationery(20, "Pen", 50, "Parker"),

        new Stationery(21, "Pencil", 30, "Faber-Castell"),

        new Stationery(22, "Eraser", 20, "Staedtler"),

        new Stationery(23, "Marker", 80, "Sharpie"),

        new Stationery(24, "Highlighter", 60, "Stabilo")

    };

    vector<Product*> books = {

        new Books(25, "Novel", 500, "Jane Austen"),

        new Books(26, "Textbook", 800, "Richard Feynman"),

        new Books(27, "Magazine", 200, "National Geographic"),

        new Books(28, "Comic", 150, "Stan Lee"),

        new Books(29, "Biography", 700, "Steve Jobs"),

        new Books(30, "Poetry Book", 400, "Rumi")

    };

    vector<Product*> girlsMakeup = {

        new GirlsMakeup(31, "Lipstick", 600, "Mac"),

        new GirlsMakeup(32, "Foundation", 1200, "Maybelline"),

        new GirlsMakeup(33, "Mascara", 800, "L'Oreal"),

        new GirlsMakeup(34, "Blush", 700, "NARS"),

        new GirlsMakeup(35, "Eyeliner", 500, "Sephora"),

        new GirlsMakeup(36, "Nail Polish", 300, "OPI")

    };

    Cart cart;

    string customerName;

    int customerId;
```

```cpp
cout << "Enter customer name: ";

getline(cin, customerName);

while (customerName.empty()) {

    cout << "Invalid input. Please enter a valid customer name: ";

    getline(cin, customerName);

}

cout << "Enter customer ID: ";

cin >> customerId;

while (cin.fail() || customerId < 0) {

    cin.clear();

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

    cout << "Invalid input. Please enter a valid customer ID: ";

    cin >> customerId;

}

while (true) {

    int choice = displayMenu();

    switch (choice) {

    case 1:

        displayAndAddProducts(electronics, cart);

        break;

    case 2:

        displayAndAddProducts(clothing, cart);

        break;

    case 3:

        displayAndAddProducts(food, cart);

        break;

    case 4:

        displayAndAddProducts(stationery, cart);

        break;

    case 5:
```

```cpp
                displayAndAddProducts(books, cart);

                break;

        case 6:

                displayAndAddProducts(girlsMakeup, cart);

                break;

        case 7:

                cart.showCart();

                break;

        case 8:

                deleteProductFromCart(cart);

                break;

        case 9:

                updateProductInCart(cart);

                break;

        case 10:

                cart.printReceipt(customerName, customerId);

                return 0;

        default:

                cout << "Invalid choice." << endl;

        }

    }

    for (auto& product : electronics) delete product;

    for (auto& product : clothing) delete product;

    for (auto& product : food) delete product;

    for (auto& product : stationery) delete product;

    for (auto& product : books) delete product;

    for (auto& product : girlsMakeup) delete product;

    return 0;

}
```

OUTPUTS:

```
Enter customer name: MUSKAN
Enter customer ID: 41

Menu:
1. View and Add Electronics
2. View and Add Clothing
3. View and Add Food
4. View and Add Stationery
5. View and Add Books
6. View and Add Girls' Makeup
7. Show Cart
8. Delete Product from Cart
9. Update Product in Cart
10. Checkout
Enter your choice: █
```

TRY 1ST OPTION :

```
Enter your choice: 1
Available Products:
Electronics [ID: 1, Name: Refrigerator, Price: PKR 50000, Warranty: 12 months]
Electronics [ID: 2, Name: Air Conditioner, Price: PKR 60000, Warranty: 24 months]
Electronics [ID: 3, Name: Generator, Price: PKR 70000, Warranty: 36 months]
Electronics [ID: 4, Name: Food Processor, Price: PKR 10000, Warranty: 12 months]
Electronics [ID: 5, Name: Mobile Phone, Price: PKR 40000, Warranty: 12 months]
Electronics [ID: 6, Name: LED TV, Price: PKR 80000, Warranty: 24 months]
Enter the ID of the product to add to the cart (0 to cancel): 4
Food Processor added to the cart.
```

TRY 2ND OPTION:

```
Enter your choice: 2
Available Products:
Clothing [ID: 7, Name: Shirt, Price: PKR 2000, Size: M, Material: Cotton]
Clothing [ID: 8, Name: Jeans, Price: PKR 3000, Size: L, Material: Denim]
Clothing [ID: 9, Name: Jacket, Price: PKR 4000, Size: XL, Material: Leather]
Clothing [ID: 10, Name: T-shirt, Price: PKR 1500, Size: S, Material: Polyester]
Clothing [ID: 11, Name: Kurta, Price: PKR 2500, Size: M, Material: Cotton]
Clothing [ID: 12, Name: Shalwar Kameez, Price: PKR 3500, Size: L, Material: Cotton]
Enter the ID of the product to add to the cart (0 to cancel): 8
Jeans added to the cart.
```

TRY 3rd OPTION:

```
Enter your choice: 3
Available Products:
Food [ID: 13, Name: Apple, Price: PKR 100, Expiry Date: 2023-12-31]
Food [ID: 14, Name: Bread, Price: PKR 50, Expiry Date: 2023-11-01]
Food [ID: 15, Name: Milk, Price: PKR 80, Expiry Date: 2023-10-15]
Food [ID: 16, Name: Eggs, Price: PKR 120, Expiry Date: 2023-09-20]
Food [ID: 17, Name: Biryani, Price: PKR 200, Expiry Date: 2023-08-30]
Food [ID: 18, Name: Naan, Price: PKR 30, Expiry Date: 2023-07-10]
Enter the ID of the product to add to the cart (0 to cancel): 17
Biryani added to the cart.
```

TRY 4th OPTION:

```
Enter your choice: 4
Available Products:
Stationery [ID: 19, Name: Notebook, Price: PKR 100, Brand: Dawn]
Stationery [ID: 20, Name: Pen, Price: PKR 50, Brand: Parker]
Stationery [ID: 21, Name: Pencil, Price: PKR 30, Brand: Faber-Castell]
Stationery [ID: 22, Name: Eraser, Price: PKR 20, Brand: Staedtler]
Stationery [ID: 23, Name: Marker, Price: PKR 80, Brand: Sharpie]
Stationery [ID: 24, Name: Highlighter, Price: PKR 60, Brand: Stabilo]
Enter the ID of the product to add to the cart (0 to cancel): 22
Eraser added to the cart.
```

TRY 5th OPTION:

```
Enter your choice: 5
Available Products:
Books [ID: 25, Name: Novel, Price: PKR 500, Author: Jane Austen]
Books [ID: 26, Name: Textbook, Price: PKR 800, Author: Richard Feynman]
Books [ID: 27, Name: Magazine, Price: PKR 200, Author: National Geographic]
Books [ID: 28, Name: Comic, Price: PKR 150, Author: Stan Lee]
Books [ID: 29, Name: Biography, Price: PKR 700, Author: Steve Jobs]
Books [ID: 30, Name: Poetry Book, Price: PKR 400, Author: Rumi]
Enter the ID of the product to add to the cart (0 to cancel): 30
Poetry Book added to the cart.
```
TRY 6th OPTION:

```
Enter your choice: 6
Available Products:
Girls' Makeup [ID: 31, Name: Lipstick, Price: PKR 600, Brand: Mac]
Girls' Makeup [ID: 32, Name: Foundation, Price: PKR 1200, Brand: Maybelline]
Girls' Makeup [ID: 33, Name: Mascara, Price: PKR 800, Brand: L'Oreal]
Girls' Makeup [ID: 34, Name: Blush, Price: PKR 700, Brand: NARS]
Girls' Makeup [ID: 35, Name: Eyeliner, Price: PKR 500, Brand: Sephora]
Girls' Makeup [ID: 36, Name: Nail Polish, Price: PKR 300, Brand: OPI]
Enter the ID of the product to add to the cart (0 to cancel): 31
Lipstick added to the cart.
```

TRY 7th OPTION:

```
Enter your choice: 7
Cart Contents:
Electronics [ID: 4, Name: Food Processor, Price: PKR 10000, Warranty: 0 months]
Clothing [ID: 8, Name: Jeans, Price: PKR 3000, Size: , Material: ]
Food [ID: 17, Name: Biryani, Price: PKR 200, Expiry Date: ]
Stationery [ID: 22, Name: Eraser, Price: PKR 20, Brand: ]
Books [ID: 30, Name: Poetry Book, Price: PKR 400, Author: ]
Girls' Makeup [ID: 31, Name: Lipstick, Price: PKR 600, Brand: ]
```

TRY 8th OPTION:

```
Enter your choice: 8
Enter the ID of the product to delete from the cart: 22
Product with ID 22 has been removed from the cart.
```

TRY 9th OPTION:

```
Enter your choice: 9
Enter the ID of the product to update in the cart: 30
Enter the new product ID: 15
Enter the new product name: Milk
Enter the new product price: 80
Product with ID 30 has been updated.
```

TRY 10th OPTION :

```
Receipt
Customer Name: MUSKAN
Customer ID: 41
-----------------------------------
Electronics [ID: 4, Name: Food Processor, Price: PKR 10000, Warranty: 0 months]
Clothing [ID: 8, Name: Jeans, Price: PKR 3000, Size: , Material: ]
Food [ID: 17, Name: Biryani, Price: PKR 200, Expiry Date: ]
Food [ID: 15, Name: Milk, Price: PKR 80, Expiry Date: ]
Girls' Makeup [ID: 31, Name: Lipstick, Price: PKR 600, Brand: ]
-----------------------------------
Total: PKR 13880
Thank you for shopping with us!
```