# E-Commerce Project Book

## Introduction

This project book is a comprehensive guide to help you design and implement an eCommerce system using C++. It covers core concepts, components, and step-by-step implementation required to build the project, providing theoretical and practical insights. As a beginner in C++, this project will serve as a valuable learning opportunity to understand object-oriented programming, file handling, and user input management.

## 1. **Understanding the Requirements**

An eCommerce system involves users interacting with products and orders. In this project, we will define two main roles: **Admin** and **Customer**.

### User Roles:

1. **Admin**:
   - Add, edit, or remove products from the catalog.
   - View all orders placed by customers.
2. **Customer**:
   - Register and log in.
   - Browse products and add them to the cart.
   - Place orders and checkout.

### Core Functionalities:

- **User Authentication**: Login system for both admin and customers.

- **Product Management**: CRUD (Create, Read, Update, Delete) operations for products.

- **Order Processing**: Managing the cart and placing orders.

- **Persistent Data**: Store data in files for later use.

### Expanded Features:

- **Search and Filter**: Customers can search for products by name or category.

- **Data Encryption**: Passwords and sensitive data are encrypted for security.

- **Analytics**: Admin can view reports such as top-selling products.

## 2. **Project Components**

### Core Classes:

#### a) `User`

A base class that defines shared attributes and functions for users.

#### b) `Admin`

Derived from `User`, this class handles admin-specific tasks.

#### c) `Customer`

Derived from `User`, this class handles customer-specific tasks such as browsing products, managing a cart, and placing orders.

#### d) `Product`

Represents items in the catalog.

#### e) `Order`

Manages customer orders.

## 3. **Step-by-Step Implementation**

1. Set Up the Environment

2. Implement Core Classes

3. Add Product Management

4. Add Customer Cart and Orders

5. Build the Login System

6. Combine Components

7. Error Handling and Testing

## 4. **Sample Code Walkthrough**

Example for adding products:

```cpp
#include <iostream>
#include <vector>
#include <string>

class Product {
    std::string name;
    double price;
    int stock;

public:
    Product(std::string pname, double pprice, int pstock)
        : name(pname), price(pprice), stock(pstock) {}

    void displayProduct() const {
        std::cout << "Name: " << name << ", Price: $" << price
                  << ", Stock: " << stock << std::endl;
    }
};

int main() {
    std::vector<Product> catalog;

    catalog.push_back(Product("Laptop", 1200.50, 10));
    catalog.push_back(Product("Phone", 800.99, 20));

    std::cout << "Product Catalog:
";
    for (const auto &product : catalog) {
        product.displayProduct();
    }

    return 0;
}
```

```

---

## 5. **Enhancements**

- Add a search feature for products.

- Implement a graphical interface using libraries like Qt or SDL.

- Add encryption for user passwords.

- Create an analytics dashboard for admins to view sales reports and stock trends.

- Introduce multi-language support for better usability.

---

## 6. **Case Studies and Practical Insights**

- **Case Study 1: Small Online Store**: Analyze how a simple eCommerce store can scale using modular and reusable code.

- **Case Study 2: File Handling Best Practices**: Explore ways to structure file input/output efficiently in C++.

- **Case Study 3: Customer Feedback Integration**: Learn how to gather and process customer feedback through the system.

---

## 7. **Conclusion**

This project introduces you to several essential programming concepts in C++, including object-oriented design, file handling, and program structure. By completing this project, you will gain a deeper understanding of how real-world applications are built and the importance of clean,
```

maintainable code.

## 8. **Appendix**

### A. Additional Resources

- **Books**: "The C++ Programming Language" by Bjarne Stroustrup.

- **Online Courses**: "C++ for Beginners" on Udemy.

- **Documentation**: Visit cplusplus.com for extensive reference material.

### B. Glossary

- **Encapsulation**: Bundling data and methods that operate on the data within a class.

- **Polymorphism**: The ability to present the same interface for different data types.

- **Inheritance**: Mechanism by which a class derives from another class, inheriting its properties and methods.