

Online Shopping System & AI Integration

1. Project Summary

This project encompasses the software architecture of a fundamental online shopping cart system and the Artificial Intelligence (AI) solutions that can be integrated into this architecture. The system is built upon user management, product cataloging, order processing, and payment security.

2. User Needs & AI Solutions

To enhance system efficiency and user experience, the following four core needs and corresponding AI strategies have been identified:

A. Personalized Shopping Experience

- **User Need:** Enabling customers to quickly find new products that may interest them based on their past shopping preferences.
- **AI Solution:** Recommendation Engine.
- **Related Classes:** Order, Product.

B. Smart Inventory Management

- **User Need:** Allowing administrators (Admin) to perform stock optimization by accessing estimated demand data before product stocks are depleted.
- **AI Solution:** Demand Forecasting.
- **Related Classes:** Product, Order.

C. Dynamic Pricing

- **User Need:** Increasing profitability by automatically updating product prices based on market conditions and stock status.
- **AI Solution:** Dynamic Pricing Algorithm.
- **Related Classes:** Product.

D. Payment Security

- **User Need:** Instant detection and prevention of potential fraudulent activities during the payment process.
- **AI Solution:** Anomaly Detection (Fraud Detection).
- **Related Classes:** Payment, Order.

3. Developed AI Prompts

Professional AI prompts (commands) prepared in accordance with the system design are as follows:

1. For the Recommendation System:

"You are an e-commerce data analyst. Using the historical order data from the Order class and product categories from the Product class, design an algorithmic logic to create a personalized 'You may also like' list for the user."

2. For Inventory Forecasting:

"For the Admin user, analyze the stockQuantity in the Product class and the frequency of orders in the Order class. Prepare a forecast report identifying products whose stock levels are expected to drop to critical levels within the next 30 days."

3. For Security and Payment:

"Write an anomaly detection rule for the payment system: If a Customer initiates a transaction via the Payment class that exceeds their normal spending limit by 500%, mark the transaction as 'Suspicious' and submit it for Admin approval."

4. Technical Architecture Notes

- **Class Diagram:** Inheritance from the User class to Customer and Admin classes has been implemented in accordance with Object-Oriented Programming (OOP) principles.
- **Use Case Diagram:** The system boundaries, external payment systems, and actor hierarchy (Customer < Visitor) have been modeled according to industry standards.