E-commerce Recommendation System (With and Without Machine Learning)

# 1. Project Overview

Project Name: E-commerce Recommendation System  
Objective: To build a personalized product recommendation system that enhances user experience on an e-commerce platform by suggesting products based on user preferences, browsing behavior, and past purchases.  
Technology Stack: Python, Flask/Django, HTML/CSS, SQL (PostgreSQL/SQLite), Pandas, APIs (for web scraping, optional), and Machine Learning Libraries (for ML-based version).

# 2. Requirements

## 2.1 Functional Requirements

Functional requirements define the system’s behavior and functionalities, both from the perspective of the user and the backend.

1. 1. User Management

- Users must be able to register, log in, and log out.  
- Users should have profiles storing details like username, email, and history of viewed and purchased products.  
- Implement user roles (Admin and Customer):  
 - Admin can manage products and view user statistics.  
 - Customer can browse, search for, and purchase products.

1. 2. Product Catalog

- Admins should be able to add, update, and delete products in the catalog.  
- Products should include attributes like Name, Category, Description, Price, and Product Image.  
- Customers should be able to:  
 - View all products or filter by categories.  
 - View product details, including related products and recommendations.

1. 3. Recommendation Engine

- ML-based Version:  
 - Provide personalized product recommendations based on user purchase history, ratings, or browsing behavior using collaborative filtering or content-based filtering.  
 - Show “People who bought this also bought...” or “Recommended for you” based on similar users or product features.  
 - Implement product similarity using item-item or user-user collaborative filtering.  
- Non-ML Version:  
 - Rule-based recommendations, e.g., products from the same category, top-selling products, or recently viewed products.  
 - Offer recommendations based on predefined criteria (e.g., price range, category, popularity).

1. 4. Search and Filtering

- Customers should be able to search for products by name, description, or category.  
- Filtering options should allow users to sort products by price, rating, or popularity.

1. 5. Cart and Checkout System

- Users should be able to add products to a cart and review their selections before checking out.  
- Implement basic payment functionality (mocked for development).

1. 6. Order History and Recommendations

- After a successful purchase, users should see recommendations for other products.  
- Store order history for each user, which can later be used to provide recommendations.

## 2.2 Non-Functional Requirements

Non-functional requirements cover system performance, security, and user experience.

1. 1. Performance

- The recommendation engine should respond with product suggestions within 2 seconds.  
- Ensure the system can handle multiple users and concurrent requests efficiently, especially for the recommendation engine.

1. 2. Scalability

- The application should support a growing product catalog and increasing user activity.  
- The recommendation system (especially the ML-based one) should be able to scale with additional user and product data.

1. 3. Security

- Implement user authentication and authorization.  
- Protect sensitive user data (e.g., passwords should be hashed).  
- Secure all communications using HTTPS and SSL.

1. 4. User Experience

- The application should have a simple, intuitive UI.  
- Responsive design should ensure a smooth experience across devices (desktop, tablet, mobile).

# 3. Scope of Work

## 3.1 Phase 1: Basic E-commerce Functionality

In this phase, the goal is to implement core e-commerce functionality without focusing on the recommendation engine.

1. 1. User Registration and Authentication
2. 2. Product Management
3. 3. Cart and Checkout System
4. 4. Order History

## 3.2 Phase 2: Recommendation Engine (With and Without ML)

### Phase 2A: Non-ML Recommendation Engine

1. 1. Rule-based Recommendations
2. 2. Testing and Optimization

### Phase 2B: ML-based Recommendation Engine

1. 1. Data Collection and Preprocessing
2. 2. Model Selection and Training
3. 3. Integration into the Application
4. 4. Continuous Model Improvement

## 3.3 Phase 3: Testing and Optimization

1. 1. Unit Testing
2. 2. Performance Testing
3. 3. Security Testing
4. 4. User Testing

# 4. Deliverables

1. 1. Phase 1 Deliverables:
2. 2. Phase 2 Deliverables:
3. 3. Phase 3 Deliverables:

# 5. Timeline

A rough timeline for project phases:  
- Phase 1 (Core E-commerce Functionality): 4 weeks  
- Phase 2A (Non-ML Recommendation System): 2 weeks  
- Phase 2B (ML-Based Recommendation System): 4-6 weeks  
- Phase 3 (Testing and Optimization): 2 weeks

# 6. Risks and Mitigations

1. 1. Data Privacy Concerns
2. 2. Scalability Issues
3. 3. Model Accuracy (For ML)