1. Overview of the System:

This system is designed for an e-commerce platform with functionalities such as user management, product listing, order processing, reviews, and payments. It includes both customers and admins, who interact with different aspects of the platform.

2. Functional Requirements:

User Management:

- Users can register, log in, and manage their profiles.
- Admins have a special role with additional permissions (e.g., managing products, viewing orders).
- Users can place orders, view their order history, and leave reviews for products.

Product Management:

- Admins can manage product categories, add, update, and delete products.
- Products are listed under categories, and each product can have images associated with it.
- Each product has a price and stock count, with updates to stock based on order quantities.

Order Management:

- Users can add products to a cart and checkout.
- Orders have a status (complete, failed, processing, refunded, pending) and contain order items linked to products.
- Users can apply coupons during checkout for discounts.
- Payment integration for processing transactions and updating order statuses.

Reviews:

- Users can rate products and leave comments.
- Admins can view reviews to monitor product feedback.

Coupon System:

- Admins can create and manage coupons with discount codes.
- Coupons are linked to orders and can provide a discount on the total price.

Payment System:

- Payments are processed for orders and have a status of "complete" or "failed."
- Each payment has a unique transaction ID and is linked to an order.

---

3. Database Schema and Table Structure:

1. Table: users

Stores information related to the users of the platform (both customers and admins).

Column Name	Data Type	Constraints	Description
id	bigint	PK, auto-increment	Unique identifier for each user
name	varchar(255)	NOT NULL	Name of the user
email	varchar(255)	UNIQUE, NOT NULL	Email address, unique for each user
password	varchar(255)	NOT NULL	Hashed password for authentication
gender	varchar(10)	NULL	Gender of the user
age	int	NULL	Age of the user
user_role	enum('customer', 'admin')	NOT NULL	Role of the user, either "customer" or "admin"
address	varchar(255)	NULL	Address of the user
phone	varchar(20)	NULL	Phone number of the user
image	varchar(255)	NULL	Profile image URL
created_at	timestamp	NOT NULL	Timestamp when the user was created
updated_at	timestamp	NOT NULL	Timestamp when the user was last updated
deleted_at	timestamp	NULL	Soft delete timestamp

---

2. Table: admins

Stores information for admin users who manage the platform.

Column Name	Data Type	Constraints	Description
id	bigint	PK, auto-increment	Unique identifier for the admin
name	varchar(255)	NOT NULL	Name of the admin
email	varchar(255)	UNIQUE, NOT NULL	Email address, unique for each admin
password	varchar(255)	NOT NULL	Hashed password for authentication
created_at	timestamp	NOT NULL	Timestamp when the admin was created
updated_at	timestamp	NOT NULL	Timestamp when the admin was last updated
deleted_at	timestamp	NULL	Soft delete timestamp

---

3. Table: categories

Stores categories of products for easier browsing.

Column Name	Data Type	Constraints	Description
id	bigint	PK, auto-increment	Unique identifier for each category
name	varchar(255)	NOT NULL	Name of the category
image	varchar(255)	NULL	Image associated with the category

created_at	timestamp	NOT NULL	Timestamp when the category was created
updated_at	timestamp	NOT NULL	Timestamp when the category was last updated
deleted_at	timestamp	NULL	Soft delete timestamp

---

4. Table: products

Stores information about the products being sold.

Column Name	Data Type	Constraints	Description
id	bigint	PK, auto-increment	Unique identifier for each product
name	varchar(255)	NOT NULL	Name of the product
description	text	NULL	Detailed description of the product
price	decimal(10, 2)	NOT NULL	Price of the product
stock	int	NOT NULL	Available stock quantity
category_id	bigint	FK, NOT NULL	References categories.id
created_at	timestamp	NOT NULL	Timestamp when the product was created
updated_at	timestamp	NOT NULL	Timestamp when the product was last updated
deleted_at	timestamp	NULL	Soft delete timestamp

---

5. Table: orders

Contains all the orders placed by users.

Column Name	Data Type	Constraints	Description
id	bigint	PK, auto-increment	Unique identifier for each order
user_id	bigint	FK, NOT NULL	References users.id
coupon_id	bigint	FK, NULL	References coupons.id (optional)
total_price	decimal(10,2)	NOT NULL	Total order price
status	enum('complete', 'failed', 'processing', 'refunded', 'pending')	NOT NULL	Order status
shipping_location	varchar(255)	NULL	Shipping address for the order
created_at	timestamp	NOT NULL	Timestamp when the order was created
updated_at	timestamp	NOT NULL	Timestamp when the order was last updated
deleted_at	timestamp	NULL	Soft delete timestamp

---

6. Table: order_items

Contains details of each product in an order.

Column Name	Data Type	Constraints	Description
id	bigint	PK, auto-increment	Unique identifier for each order item
order_id	bigint	FK, NOT NULL	References orders.id

product_id	bigint	FK, NOT NULL	References products.id
quantity	int	NOT NULL	Quantity of the product ordered
price	decimal(10, 2)	NOT NULL	Price of the product at the time of the order
deleted_at	timestamp	NULL	Soft delete timestamp

---

7. Table: reviews

Stores product reviews left by users.

Column Name	Data Type	Constraints	Description
id	bigint	PK, auto-increment	Unique identifier for each review
user_id	bigint	FK, NOT NULL	References users.id
product_id	bigint	FK, NOT NULL	References products.id
rating	int	NOT NULL	Rating (1 to 5 stars)
comment	text	NULL	Optional comment about the product
created_at	timestamp	NOT NULL	Timestamp when the review was created
deleted_at	timestamp	NULL	Soft delete timestamp

---

8. Table: images

Stores images associated with products.

Column Name	Data Type	Constraints	Description
id	bigint	PK, auto-increment	Unique identifier for each image

product_id	bigint	FK, NOT NULL	References products.id
url	varchar(255)	NOT NULL	URL to the image
alt_text	varchar(255)	NULL	Alternative text for the image
deleted_at	timestamp	NULL	Soft delete timestamp

---

9. Table: cart

Stores the cart of each user before they checkout.

Column Name	Data Type	Constraints	Description
id	bigint	PK, auto-increment	Unique identifier for each cart
user_id	bigint	FK, NOT NULL	References users.id
created_at	timestamp	NOT NULL	Timestamp when the cart was created
updated_at	timestamp	NOT NULL	Timestamp when the cart was last updated
deleted_at	timestamp	NULL	Soft delete timestamp

---

10. Table: cart_items

Contains items that are in the user's cart before checkout.

Column Name	Data Type	Constraints	Description
id	bigint	PK, auto-increment	Unique identifier for each cart item

cart_id	bigint	FK, NOT NULL	References cart.id
product_id	bigint	FK, NOT NULL	References products.id
quantity	int	NOT NULL	Quantity of the product in the cart
deleted_at	timestamp	NULL	Soft delete timestamp

---

11. Table: coupons

Stores information about discount coupons.

Column Name	Data Type	Constraints	Description
id	int	PK, auto-increment	Unique identifier for the coupon
code	varchar(255)	UNIQUE, NOT NULL	Coupon code for users to apply
discount	decimal(10, 2)	NOT NULL	Discount amount
expiry_date	datetime	NOT NULL	Expiry date of the coupon
created_at	timestamp	NOT NULL	Timestamp when the coupon was created
deleted_at	timestamp	NULL	Soft delete timestamp

---

12. Table: payments

Stores payment details for orders.

Column Name	Data Type	Constraints	Description
id	bigint	PK, auto-increment	Unique identifier for each payment
order_id	bigint	FK, NOT NULL	References orders.id

amount	decimal(10,2)	NOT NULL	Amount paid for the order
payment_method	varchar(50)	NOT NULL	Method of payment (credit card, PayPal, etc.)
status	enum('complete', 'failed')	NOT NULL	Payment status (complete, failed)
transaction_id	varchar(255)	UNIQUE, NOT NULL	Unique identifier for the transaction
processed_at	timestamp	NULL	Timestamp when the payment was processed
created_at	timestamp	NOT NULL	Timestamp when the payment was created
updated_at	timestamp	NOT NULL	Timestamp when the payment was last updated
deleted_at	timestamp	NULL	Soft delete timestamp

---

4. Entity Relationship Diagram (ERD)

- **Users** are related to **orders**, **reviews**, and **carts**.
- **Orders** contain **order items**, each of which refers to a **product**.
- **Products** belong to a **category** and have associated **images** and **reviews**.
- **Payments** are linked to **orders**, while **coupons** can be applied to **orders** to get discounts.

Software Design and Functional Requirements for the E-Commerce System

This document outlines the software design, architecture, and functional requirements for an e-commerce system that manages users, products, orders, payments, reviews, coupons, and more.

1. Software Design Overview

System Architecture:

The e-commerce platform follows a **client-server architecture**, where users interact with the system through a web-based interface (client), and the server (back-end) handles data management, processing, and business logic. The system will be built using a combination of technologies such as:

- **Frontend:** HTML, CSS, JavaScript .
- **Backend:** LARAVEL 10 with MVC AND WAMP
- **User Authentication:** Breeze Laravel.
- **Database:**MySQL.
- **Payment Gateway:** Stripe, PayPal
- **File Storage:** DB and Storage in laravel project

The system will have **two main roles**:

1. **Customers** – Users who can browse products, manage their cart, place orders, and write reviews.
2. **Admins** – Users with administrative privileges to manage products, users, and order statuses.

---

2. Functional Requirements

1. User Management:

- **Registration and Login:** Users (both customers and admins) can register for an account with an email and password. Admins have a separate admin panel for managing platform content. Authentication will be done via hashed passwords.
  Functional Requirements:
  - A user must be able to register, log in, and log out of the system.
  - Passwords must be stored securely using hashing algorithms.
  - Admins must have different permissions from regular customers.
- **Profile Management:** Users can update their profile information such as name, email, phone number, and address.
  Functional Requirements:
  - Users can update their personal information (e.g., address, phone number).
  - Admins can view and manage user accounts.

2. Product Management (Admin):

Admins can add, update, delete, and manage products within the system.

Functional Requirements:

- Admins must be able to add new products with a name, description, price, stock count, and category.
- Admins must be able to update product information, including price and stock.
- Admins can delete products (soft delete with a timestamp).
- **Category Management:** Admins can add, edit, or delete product categories.
  Functional Requirements:
    - Admins must be able to create categories for organizing products.
    - Admins can associate products with specific categories.

3. Cart Management:

Customers can add products to their cart, view their cart, update quantities, and proceed to checkout.

Functional Requirements:

- Users can add, update, and remove products from the cart.
- The system should calculate the total price for the items in the cart.
- Cart items are stored even after logging out (persisted in the database).

4. Order Management:

- **Order Creation:** Once a user proceeds to checkout, an order is created. The order includes user details, product items, shipping address, and order status.
  Functional Requirements:
    - Users must be able to place an order with items from their cart.
    - The system should handle different order statuses such as "pending," "processing," "completed," "failed," and "refunded."
- **Order History:** Users can view past orders, including details about products, status, and total price.
  Functional Requirements:
    - Users can view a history of all past orders.
    - Admins can view and manage all user orders.

5. Payment Integration:

- **Payment Processing:** Once an order is placed, the user proceeds to make a payment using one of the available payment methods (e.g., credit card, PayPal).
  Functional Requirements:
    - The system should integrate with payment providers (e.g., Stripe, PayPal) for processing payments.
    - Payments should be linked to the orders, and payment statuses should be updated (complete, failed).
    - Payments should be securely processed with a unique transaction ID.

6. Reviews and Ratings:

- Users can leave a review and rate products they have purchased.
  Functional Requirements:
    - Users can leave a review with a rating between 1 and 5 stars.
    - Reviews should be linked to both the product and the user who wrote it.
    - Admins can view and manage user reviews.

7. Coupon System:

- Admins can create discount coupons that users can apply during checkout.
  Functional Requirements:
    - Coupons must have a unique code and an expiry date.
    - Admins can create and manage coupons that apply a discount to the total order amount.
    - Coupons can be linked to orders at checkout to apply discounts.

8. Image Management:

- Products can have multiple images associated with them, including a main image and alternative images for better product display.
  Functional Requirements:
    - Admins can upload and manage images for each product.
    - Images should have alt text for accessibility.
    - Images should be stored securely and linked to the correct product.

---

3. System Flow

Customer Workflow:

1. **Registration/Login:**

    - The customer registers or logs in with their credentials (email and password).
2. **Browse Products:**
    - The customer browses the product catalog by category or search.
3. **Add Products to Cart:**
    - The customer adds products to their cart and updates quantities as needed.
4. **Checkout:**
    - The customer reviews the cart, applies any available coupons, and proceeds to checkout.
5. **Payment:**
    - The customer selects a payment method and completes the payment process.
6. **Order Confirmation:**
    - After successful payment, the customer receives an order confirmation with tracking information.
7. **Write Reviews:**
    - After receiving products, the customer can leave reviews for purchased products.

Admin Workflow:

1. **Login:**
    - The admin logs in with administrative credentials.
2. **Manage Users:**
    - The admin can view, update, or delete user accounts.
3. **Product Management:**
    - The admin can add, update, or delete products and manage their categories.
4. **Order Management:**
    - The admin can view and manage all orders, update their statuses, and process refunds.
5. **Manage Coupons:**
    - The admin can create and manage discount coupons for customers.
6. **Review Management:**
    - The admin can monitor, approve, or delete user reviews.

---

4. Data Flow Diagram

- **User Interaction Layer:**
    - Customers interact with the system via the web interface for product browsing, order placement, and payments.
- **Business Logic Layer (Backend):**
    - Handles user requests, manages cart and orders, and processes payments.
- **Database Layer:**
    - All data regarding users, orders, products, payments, and reviews are stored in the relational database.
- **Payment Gateway:**
    - Integrates with third-party services (e.g., Stripe) for secure payment processing.

---

5. Non-Functional Requirements

1. **Scalability:**
    - The system should be able to handle high traffic and large numbers of users and products.
2. **Performance:**
    - Response time for user actions (e.g., adding to cart, placing orders) should be under 2 seconds.
3. **Security:**
    - User data must be securely stored (passwords hashed, sensitive data encrypted).
    - Payment transactions must be securely processed using industry-standard encryption.
4. **Usability:**
    - The platform should have an intuitive user interface for both customers and admins.
    - The system must be mobile-friendly and responsive.
5. **Availability:**
    - The platform should be available 99.9% of the time, with a disaster recovery plan in place for critical systems.