DIG5127 Database and Web Application Development

Milestone 2 - ERM proposal

Student Name: Nayan Subedi

Student ID: 23140735

Date: 31/01/2024

Wordcount: 960 Words

Page count: 8 Pages

Contents

1. Introduction

This proposal presents the design and features of AromaAlchemy, an online platform dedicated to the sale and distribution of premium perfumes. Our aim is to offer a seamless and user-friendly shopping experience for fragrance enthusiasts, providing a wide range of scents and personalized customer service.

2. Application Overview

AromaAlchemy will function as a digital marketplace for perfume enthusiasts, offering detailed product descriptions, customer reviews, and secure payment options. The platform will serve customers, ensuring a diverse array of products to suit various preferences and occasions.

3. Users

AromaAlchemy is designed to accommodate the following user categories:

- Customers: Those looking to purchase perfumes for personal use.
- Administrators: Responsible for managing product listings, orders, and customer service.

4. Functionality

The core functionalities of AromaAlchemy encompass:

- User Account Management: Streamlined processes for account creation and user profile management.
- Product Discovery: Intuitive browsing and search features based on categories, brands, or scent preferences.
- Shopping Cart Management: Users can add products to their cart, review selections, and manage items effectively.
- Order Placement and Tracking: Seamless processes for placing orders with an integrated tracking system for order status.
- Secure Payment Processing: Implementation of robust security measures for safe and secure payment transactions.
- Product Reviews: Users can contribute and view product reviews, fostering a community-driven approach.
- Inventory Management: Administrators can efficiently manage product stock, ensuring accurate listings.

5. Entity Relationship Model (ERM)

The Entity Relationship Model (ERM) is a graphical illustration of how different entities in a database relate to each other. It's a tool that aids in the organization and comprehension of the interconnections between various data points. In the case of AromaAlchemy, the ERM is instrumental in creating the database structure that will store and manage data about users, products, orders, and other facets of the online platform.

Let's dive into the primary elements of the AromaAlchemy ERM:

A. **Users Entity**:
- Attributes: UserID, FullName, Email, Password, Phone, Address.

This entity holds data about the platform's users, including their unique ID, complete name, contact information, and residential address.

B. **Products Entity:**
- Attributes: ProductID, CategoryID, Name, Brand, Images, Description, Price, StockQuantity.

This entity provides the information about the perfumes sold on the platform, such as their unique ID, category, name, brand, images, description, price, and the quantity available in stock.

C. **Orders Entity:**
- Attributes: OrderID, UserID, OrderDate, Quantity, TotalAmount, PaymentStatus.

This entity records data about orders made by users, including the order ID, user ID, order date, quantity of products ordered, total amount, and the status of the payment.

D. **Cart Entity:**
- Attributes: CartID, UserID, ProductID, Quantity.

The cart entity is responsible for managing the items users have placed in their shopping carts. It includes a unique cart ID, user ID, product ID, and the quantity of each product in the cart.

A. Category Entity:

- Attributes: CategoryID, Name.

This entity classifies products into various categories. It includes a unique category ID and the name of each category.

B. Reviews Entity:

- Attributes: ReviewID, UserID, ProductID, Rating, Date, Comment.

The reviews entity holds data about user-generated product reviews, including the review ID, user ID, product ID, rating, review date, and user comments.

C. Payments Entity:

- Attributes: PaymentID, OrderID, PaymentDate, PaymentAmount, PaymentMethod.

This entity maintains records of payment details related to orders, with attributes such as payment ID, order ID, payment date, payment amount, and the selected payment method.

Entities	Attributes	Data Types	Constraints
A. Users	UserID	Integer	Primary Key, Not Null, Auto Increment
	FullName	Varchar(50)	Not Null
	Email	Varchar(50)	Not Null
	Password	Varchar(255)	Not Null
	Phone	Varchar(20)	Not Null
	Address	Varchar(255)	Not Null
B. Orders	OrderID	Integer	Primary Key, Not Null, Auto Increment
	UserID	Integer	Foreign Key, Not Null
	OrderDate	Timestamp	Default Current Timestamp, Not Null
	Quantity	Integer	Not Null
	TotalAmount	Decimal(10,2)	Not Null
	PaymentStatus	Varchar(20)	Not Null
C. Cart	CartID	Integer	Primary Key, Not Null, Auto Increment

	UserID	Integer	Foreign Key, Not Null
	ProductID	Integer	Foreign Key, Not Null
	Quantity	Integer	Not Null
	Price	Decimal(10,2)	Not Null
D. Review	ReviewID	Integer	Primary Key, Not Null, Auto Increment
	UserID	Integer	Foreign Key, Not Null
	ProductID	Integer	Foreign Key, Not Null
	Rating	Integer	Not Null
	Date	Timestamp	Default Current Timestamp, Not Null
	Comment	Text	
E. Product	ProductID	Integer	Primary Key, Not Null, Auto Increment
	CategoryID	Integer	Foreign Key, Not Null
	Name	Varchar(50)	Not Null
	Brand	Varchar(50)	Not Null
	Images	Varchar(255)	Not Null
	Description	Text	
	Price	Decimal(10,2)	Not Null
	StockQuantity	Integer	Not Null
F. Category	CategoryID	Integer	Primary Key, Not Null, Auto Increment
	Name	Varchar(50)	Not Null
G. Payment	PaymentID	Integer	Primary Key, Not Null, Auto Increment
	OrderID	Integer	Foreign Key, Not Null
	PaymentDate	Timestamp	Default Current Timestamp, Not Null
	PaymentAmount	Decimal(10,2)	Not Null
	PaymentMethod	Varchar(50)	Not Null

6. Relationships

i.   Users to Orders (One to Many) : Users can place many orders.

ii.   Users to Cart(One Mandotary) : Users must use cart to place order

iii.   Users to Payment (One to Many) : Users can pay for many orders.

iv.   Users to Review (0 to many optional): Users may or maynot write review.

v.   Orders to Product (1 mandotary to Many Optional) : Orders must contain one product.

vi.   Orders to Payment (1 to 1) : One order can only have one payment.

vii.   Category to Products (One to Many) : One Category can contains many products.

viii.   Cart to Products (1 mandotary to Many Optional) : Cart must contain atleast one product to place orders.

ix.   Product to Review (0 to many optional): Products may or maynot contain review.

7. **Entity Relationship Model (ERM) Diagram**

payment

PK	ID(INT,NOT NULL,AUTO_INCREMENT)
FK	User (id) (INT,NOT NULL)
FK	Order(id) (INT,NOT NULL)
	PaymentDate (TIMESTAMP DEFAULT CURRENT TIMESTAMP, NOT NULL)
	PaymentAmount(Decimal(10,2), NOT NULL)
	PaymentMethod (VARCHAR(50),NOT NULL)

users

PK	ID(INT,NOT NULL,AUTO_INCREMENT)
	FullName(VARCHAR(50), NOT NULL)
	Email(VARCHAR(50), NOT NULL)
	Password(VARCHAR(255), NOT NULL)
	Phone (VARCHAR(20), NOT NULL)
	Address (VARCHAR(255), NOT NULL)

orders

PK	ID(INT,NOT NULL,AUTO_INCREMENT)
FK	Product(id) (INT,NOT NULL)
FK	User(id) (INT,NOT NULL)
	OrderDate (TIMESTAMP DEFAULT CURRENT TIMESTAMP)
	Quantity (INT, NOT NULL)
	TotalAmount (Decimal(10,2), NOT NULL)
	PaymentStatus (VARCHAR(20) , NOT NULL)

cart

PK	ID(INT,NOT NULL,AUTO_INCREMENT)
FK	User(id) (INT,NOT NULL)
FK	Product(id) (INT,NOT NULL)
	Quantity(INT, NOT NULL)
	Price(Decimal(10,2), NOT NULL)

review

PK	ID(INT,NOT NULL,AUTO_INCREMENT)
FK	User(id) (INT,NOT NULL)
FK	Product(id) (INT,NOT NULL)
	Rating(INT(5), NOT NULL)
	Date (DATE, NOT NULL)
	Comment (TEXT)

category

PK	ID(INT,NOT NULL,AUTO_INCREMENT)
	Name(VARCHAR(50), NOT NULL)

product

PK	ID(INT,NOT NULL,AUTO_INCREMENT)
FK	Category(id) (INT,NOT NULL)
	Name(VARCHAR(50), NOT NULL)
	Brand(VARCHAR(50), NOT NULL)
	Images (VARCHAR(255), NOT NULL)
	Description (TEXT)
	price (DECIMAL(10, 2) NOT NULL)
	StockQuantity(INT(11), NOT NULL)