**DATABASE SCHEMA AND EXPLANATION**

Basic outline of a database schema for an e-commerce platform focused on online book sales. This schema will cover the essential entities and relationships needed to manage products, users, orders, and other necessary functionalities. Here's a simplified version:

**Users/Customer Table:**

UserID (Primary Key)

Username

Password (hashed)

Email

Full Name

Address

Phone Number

Other relevant information

**Books Table:**

BookID (Primary Key)

Title

Author

ISBN

Description

Price

Quantity Available

Category/Genre

Publisher

Publication Date

Other relevant information

**Orders Table:**

OrderID (Primary Key)

UserID (Foreign Key referencing Users table)

OrderDate

TotalAmount

OrderStatus (e.g., pending, processing, shipped, delivered, cancelled)

**OrderItems Table:**

OrderItemID (Primary Key)

OrderID (Foreign Key referencing Orders table)

BookID (Foreign Key referencing Books table)

Quantity

Price

**Payment Table:**

PaymentID (Primary Key)

OrderID (Foreign Key referencing Orders table)

PaymentMethod

Amount

PaymentDate

**Reviews/Ratings Table:**

ReviewID (Primary Key)

BookID (Foreign Key referencing Books table)

UserID (Foreign Key referencing Users table)

Rating (1 to 5 stars)

ReviewText

ReviewDate

**Cart Table:**

CartID (Primary Key)

UserID (Foreign Key referencing Users table)

BookID (Foreign Key referencing Books table)

Quantity

DateAdded

This schema covers the basics. However, depending on the complexity of your e-commerce platform, you might need additional tables for features like discounts, promotions, shipping information, etc.

**Explanation:**

**Users/Customer Table:** Stores information about registered users/customers.

**Books Table:** Contains details about the books available for sale.

**Orders Table:** Records details of each order placed by customers.

**OrderItems Table:** Represents individual items within each order along with their quantities and prices.

**Payment Table:** Stores payment information related to orders.

**Reviews/Ratings Table:** Allows users to review and rate books.

**Cart Table:** Stores temporary information about items added to the shopping cart before checkout.

This schema enables you to manage the core functionalities of an online book sales platform, including user registration, product management, order processing, payments, and customer feedback.

**DATABASE CODE**

**DATABASE MY SQL**

-- Create the Users table

CREATE TABLE Users (

user\_id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR (50) NOT NULL,

email VARCHAR (100) NOT NULL,

password VARCHAR (100) NOT NULL,

address VARCHAR (255),

phone\_number VARCHAR (20),

registration\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

-- Create the Books table

CREATE TABLE Books (

book\_id INT AUTO\_INCREMENT PRIMARY KEY,

title VARCHAR (255) NOT NULL,

author VARCHAR (100) NOT NULL,

ISBN VARCHAR (20) NOT NULL,

price DECIMAL (10, 2) NOT NULL,

quantity\_in\_stock INT NOT NULL,

description TEXT,

genre\_id INT,

FOREIGN KEY (genre\_id) REFERENCES Genres(genre\_id)

);

-- Create the Genres table

CREATE TABLE Genres (

genre\_id INT AUTO\_INCREMENT PRIMARY KEY,

genre\_name VARCHAR (50) NOT NULL

);

-- Create the Orders table

CREATE TABLE Orders (

order\_id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT,

order\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

total\_amount DECIMAL (10, 2) NOT NULL,

status ENUM ('pending', 'shipped', 'delivered') DEFAULT 'pending',

FOREIGN KEY (user\_id) REFERENCES Users(user\_id)

);

-- Create the Order\_Details table

CREATE TABLE Order\_Details (

order\_detail\_id INT AUTO\_INCREMENT PRIMARY KEY,

order\_id INT,

book\_id INT,

quantity\_ordered INT NOT NULL,

price\_at\_purchase DECIMAL (10, 2) NOT NULL,

FOREIGN KEY (order\_id) REFERENCES Orders(order\_id),

FOREIGN KEY (book\_id) REFERENCES Books(book\_id),

}

}

**FRONTEND DEVELOPMENT**

**HTML**

<!-- Example HTML structure for book listing page -->

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Bookstore</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<header>

<h1>Bookstore</h1>

<!-- Navigation links -->

</header>

<main>

<section id="book-list">

<!-- Display books here -->

</section>

</main>

<footer>

<p>&copy; 2024 Bookstore</p>

</footer>

<script src="scripts.js"></script>

</body>

</html>

**CSS**

/\* Example CSS styles \*/

body {

font-family: Arial, sans-serif;

}

header, footer {

background-color: #333;

color: #fff;

padding: 10px;

text-align: center;

}

main {

padding: 20px;

}

#book-list {

display: grid;

grid-template-columns: repeat(auto-fill, minmax(250px, 1fr));

grid-gap: 20px;

}

.book-card {

border: 1px solid #ccc;

padding: 10px;

border-radius: 5px;

}

.book-card img {

max-width: 100%;

}

.book-title {

font-weight: bold;

}

.book-author {

font-style: italic;

}

.book-price {

color: green;

}

**BACKEND DEVELOPMENT**

// Example JavaScript code for fetching and displaying books

const bookList = document.getElementById('book-list');

async function fetchBooks() {

try {

const response = await fetch('/books');

const books = await response.json();

displayBooks(books);

} catch (error) {

console.error('Error fetching books:', error);

}

}

function displayBooks(books) {

bookList.innerHTML = '';

books.forEach(book => {

const bookCard = document.createElement('div');

bookCard.classList.add('book-card');

const img = document.createElement('img');

img.src = book.imageUrl;

img.alt = book.title;

const title = document.createElement('h2');

title.textContent = book.title;

const author = document.createElement('p');

author.textContent = by ${book.author};

const price = document.createElement('p');

price.classList.add('book-price');

price.textContent = $${book.price};

bookCard.appendChild(img);

bookCard.appendChild(title);

bookCard.appendChild(author);

bookCard.appendChild(price);

bookList.appendChild(bookCard);

});

}

fetchBooks();