Online Bookstore System Report

Database Systems (Fall 2025)

Begad Mohamed 8584
Hazem Barakat 8621
Ahmed Saied 8574
Adham Zakaria 8648
Faculty of Engineering, Alexandria University

December 26, 2025

Contents

1 Introduction

This project involves the analysis, design, and implementation of a simplified online bookstore system. The system supports two distinct user roles: Administrators and Customers, managing books, publishers, stock levels, orders, sales transactions, and shopping carts.

The system ensures data integrity through database constraints and triggers. The implementation utilizes a modern technology stack:

- **Frontend:** React with TypeScript
- **Backend:** Node.js
- **Database:** MySQL (Relational Schema)

2 Database Design

2.1 Entity Relationship Diagram (ERD)

The database structure was designed to handle users, books, publishers, and orders. Below is the ERD representing the system's logical structure.

The diagram utilizes **Crow's Foot notation** to visualize the relationships and cardinality constraints (such as One-to-Many and Many-to-Many) between the entities.

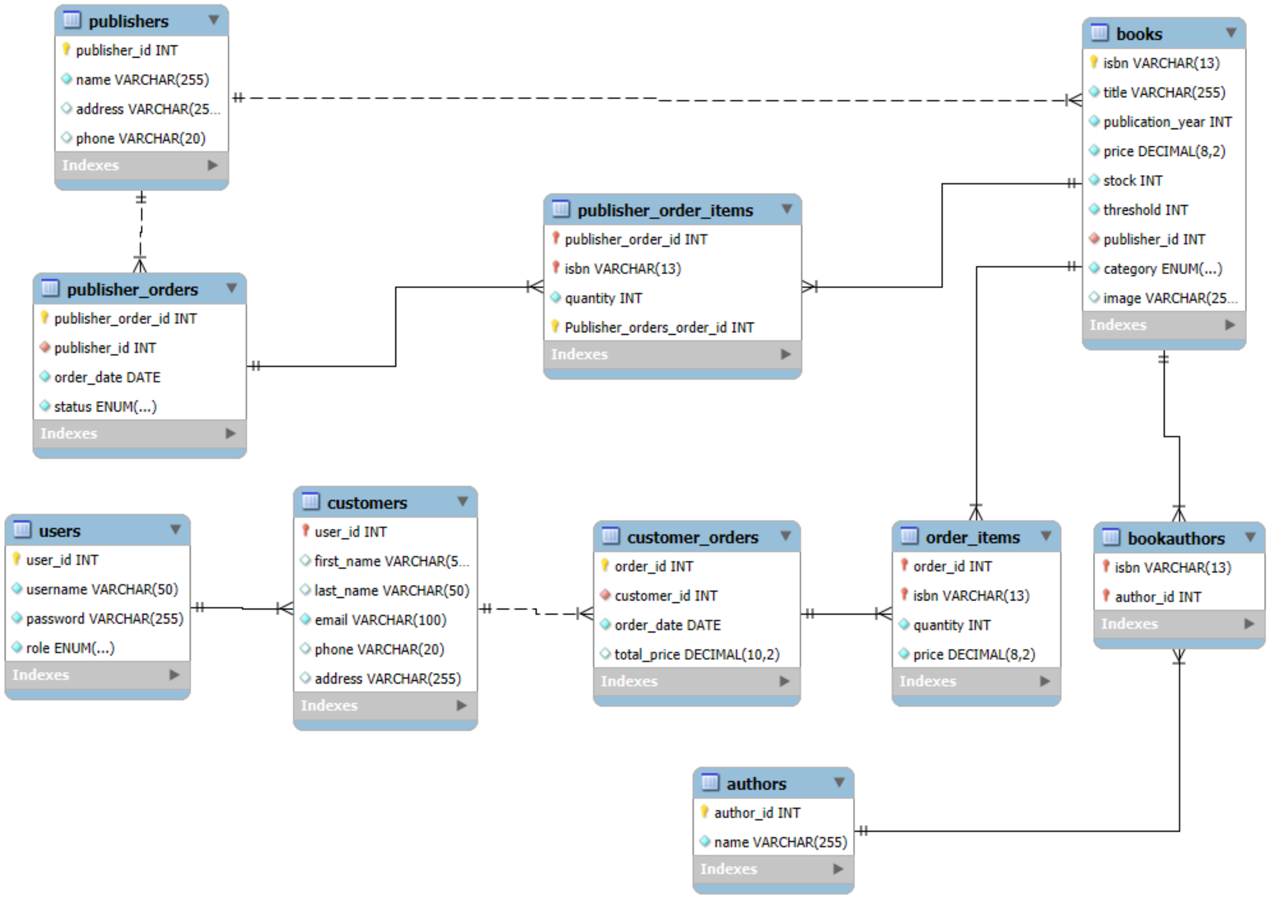

Figure 1: Bookstore System Entity Relationship Diagram

2.2 Relational Schema

Based on the implemented SQL database, the relational schema is defined as follows (Primary Keys are underlined):

Users (user_id, username, password, role).

Customers (user_id, first_name, last_name, email, phone, address).

Publishers (publisher_id, name, address, phone).

Books (ISBN, title, publication_year, price, stock, threshold, category, image, publisher_id).

Authors (author_id, name).

Book_Authors (ISBN, author_id).

Customer_Orders (order_id, customer_id, order_date, total_price).

Order_Items (order_id, ISBN, quantity, price).

Publisher_Orders (publisher_order_id, publisher_id, order_date, status).

Publisher_Order_Items (publisher_order_id, ISBN, quantity).

3 Implemented Features

3.1 Core System Features

- **Search Functionality:** Users can search for books by ISBN, title, category, author, or publisher.
- **Authentication:** Secure login and signup processes for both customers and administrators.

3.2 Administrator Features

- **Book Management:** Admins can add new books with threshold values and modify existing book details.
- **Stock Triggers:**
  - Updates are prevented if they result in negative stock.
  - Orders are automatically placed to publishers when stock drops below the threshold.
- **Order Confirmation:** Admins confirm publisher orders, which automatically updates the book stock.
- **System Reports:** The system generates analytics for Total Sales (monthly/daily), Top 5 Customers, and Top 10 Selling Books.

3.3 Customer Features

- **Profile Management:** Customers can edit personal information and passwords.
- **Shopping Cart:** Users can add items, view totals, and remove items.
- **Checkout:** Secure mock transaction handling requiring credit card input, which deducts purchased quantities from the store stock.
- **Order History:** Customers can view detailed logs of their past orders.

4 User Interface Logic

The frontend is built using React and TypeScript to ensure a responsive user experience. The logic for key screens is described below:

1. **Login/Signup Screen:** Handles user input for credentials. Upon success, a session token is stored. New users provide full details (address, phone) during signup.
2. **Book Store & Search:** Displays a grid of books. Filter components allow sorting by Category or Author. Clicking a book adds it to the local cart state via the Backend API.
3. **Cart & Checkout:** Displays selected items and calculates the total price. The ”Checkout” button triggers a modal for credit card entry. Valid input triggers the `/checkout` API endpoint to update the database.
4. **Admin Dashboard:** Contains forms for ”Add Book” and ”Modify Book.” It utilizes charting libraries to visualize the Sales Reports returned by the reporting endpoints.

5 Team Roles and Contributions

The project was divided among four members, covering Frontend, Backend, and Database logic.

Member	**Role**	**Primary Responsibilities**
Hazem Barakat	Frontend Developer	React UI, Integration, State Management, Authentication Pages, Customer Features UI.
Ahmed Saied	Backend A	Security, Customer API, Shopping Cart Logic, Checkout Transaction Handling.
Adham Zakaria	Backend B	Admin Dashboard APIs, Inventory Management, System Reports (Analytics).
Begad Mohamed	Database Specialist	ERD Design, Relational Schema, Triggers (Negative Stock, Auto-Order), Data Population.

Table 1: Team Task Breakdown