****

**CS-2005**

**Database Systems Project Report**

**Shelfly: Online Bookstore Management System**

**Group Members:**

* Arsalan Ahmed (23K-0536)[Group Leader]
* Muhammad Yousuf (23K-0690)

**1. Introduction**

The Shelfly Online Bookstore Management System is a web-based platform designed to digitalize bookstore operations. The main aim is to provide a reliable system where customers can browse books, place orders, and track their purchases, while administrators can efficiently manage inventory, orders, and payments. The motivation behind this project stems from the need to reduce manual errors in bookstore management and improve customer satisfaction through a user-friendly digital interface.

**2. Background**

Before starting development, research was conducted on existing e-commerce platforms to understand common features and user expectations. Systems like Amazon and local online stores LibertyBooks were analyzed to identify essential functionalities such as shopping carts, discount systems, and order tracking. Based on this research, the decision was made to build a full-stack bookstore management system using Django for the backend (due to its built-in admin panel and ORM) and HTML/CSS for a dynamic frontend experience. MySQL was selected as the database for its reliability and compatibility with Django.

**3. Project Specification**

**Technology Stack:**

* **Frontend:** React.js (for dynamic UI components)
* **Backend:** Django (Python web framework)
* **Database:** MySQL (relational database)
* **Development Tools:** VS Code, Git/GitHub for version control

**Core Entities:**

1. **Customer** - Stores user account information
2. **Book** - Contains book details (title, author, price, stock)
3. **Order** - Manages order information
4. **OrderItem** - Junction table linking orders and books
5. **Payment** - Tracks payment transactions
6. **Cart/CartItem** - Shopping cart functionality
7. **Coupon** - Discount code management
8. **Delivery** - Delivery information (BCNF normalization)
9. **OrderCancellation** - Cancellation records (BCNF normalization)

**Database Normalization:** The database is normalized to BCNF (Boyce-Codd Normal Form) to eliminate redundancy. For example, delivery information is separated into its own table with a one-to-one relationship with Order, ensuring that delivery details don't create partial dependencies.

**ERD Diagram:**

A diagram of a network

AI-generated content may be incorrect.

**4. Problem Analysis**

Traditional bookstores face several challenges:

* Manual inventory tracking leading to stock discrepancies
* Difficulty in managing customer orders and payment records
* Lack of transparency in order status for customers
* Time-consuming manual discount calculations

The Shelfly system addresses these problems by:

* Automating stock updates when orders are placed or cancelled
* Providing a centralized database for all transactions
* Allowing customers to track orders in real-time
* Implementing automatic discount calculations (first-time buyer discount, order value discounts, coupon codes)

**5. Solution Design**

**Project Architecture:**

The system follows the MVT (Model-View-Template) pattern through Django's architecture:

* **Models** define database structure
* **Views** handle business logic
* **Templates** render the user interface

**Key Features Implemented:**

**For Customers:**

* User registration and authentication
* Browse books with search functionality
* Shopping cart with quantity management
* Apply discount coupons
* Multiple payment methods (Cash on Delivery, Credit/Debit Card)
* Order history and cancellation
* Profile management

**For Administrators:**

* Complete CRUD operations on books
* Order management and status updates
* Payment tracking
* Coupon creation and management
* Customer information access
* Contact message handling

**Discount System:** The system implements three types of discounts that can be applied simultaneously:

1. **Coupon Discount** - Codes with fixed or percentage discounts
2. **Order Value Discount** - Automatic discounts based on cart value (5% for Rs.1000+, 10% for Rs.2000+, 15% for Rs.5000+)
3. **First-Time Buyer Discount** - 15% discount for new customers

**Shipping Calculation:**

* Free shipping for orders above Rs. 5000
* Rs. 50 base fee for orders below Rs. 5000
* Additional Rs. 10 per book if more than 5 books

**Card Payment Validation:** The system includes real card validation using the Luhn algorithm to verify card numbers, expiry date checks, and CVV validation. Card types (Visa, Mastercard, etc.) are automatically detected.

**6. Implementation & Testing**

The project was developed in phases:

**Phase 1 - Database Setup:** Created models for all entities with proper relationships. Applied migrations to create MySQL tables.

**Phase 2 - Authentication System:** Implemented user registration, login, and logout functionality. Created a custom decorator to restrict certain views to customers only (preventing admin users from accessing customer features).

**Phase 3 - Core Features:** Developed book listing, cart management, and checkout process. Implemented the discount calculation logic with proper order of operations.

**Phase 4 - Payment Integration:** Added Cash on Delivery and Card payment options. Implemented card validation functions and payment success/failure pages.

**Phase 5 - Admin Panel:** Customized Django admin to provide detailed views for managing books, orders, payments, and coupons. Added inline editing for related models.

**Testing Approach:**

* Tested user registration with duplicate usernames/emails
* Verified cart calculations with multiple discount combinations
* Tested card validation with valid and invalid card numbers
* Checked stock updates on order placement and cancellation
* Verified coupon usage limits and expiry validation
* Tested order cancellation and stock restoration

**7. Project Breakdown Structure**

**Week 1-2:**

* Project planning and database design
* ERD creation and normalization
* Django project setup

**Week 3-4:**

* Model implementation
* Admin panel customization
* Authentication system

**Week 5-6:**

* Book listing and detail pages
* Cart functionality
* Search implementation

**Week 7-8:**

* Checkout process
* Payment methods
* Order management

**Week 9-10:**

* Discount system implementation
* Card payment validation
* Order cancellation feature

**Week 11-12:**

* Testing and bug fixes
* UI improvements
* Documentation

**8. Results**

The system successfully handles all planned functionalities:

* Users can register and manage their accounts
* Book browsing works with search filters
* Cart correctly calculates subtotals, discounts, and shipping
* Orders are created with proper stock deduction
* Payment records are maintained accurately
* Administrators can manage all aspects through the admin panel
* Order cancellation properly restores stock
* Multiple discounts can be applied simultaneously

**Key Achievements:**

* Discount amounts are stored at order creation time, preventing calculation issues later
* BCNF normalization ensures data integrity
* Proper stock management with automatic updates
* Secure password handling using Django's built-in authentication
* Validation prevents invalid orders (insufficient stock, expired coupons)

A screenshot of a book

AI-generated content may be incorrect.

A screenshot of a login form

AI-generated content may be incorrect.

A screenshot of a login form

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a video game

AI-generated content may be incorrect.

A screenshot of a phone

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a shopping cart

AI-generated content may be incorrect.

A screenshot of a chat

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a credit card payment

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a website

AI-generated content may be incorrect.A screenshot of a phone

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a contact us

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a video game

AI-generated content may be incorrect.

**9. Conclusion**

The Shelfly Online Bookstore Management System successfully achieves its objectives of digitalizing bookstore operations. The system provides a complete solution for managing books, customers, orders, and payments while maintaining data integrity through proper normalization.

The implementation demonstrates how a well-structured database design can prevent redundancy and ensure consistency. The discount system showcases business logic implementation with multiple promotional strategies working together. Card payment validation adds security without requiring external payment gateways.

**Challenges Faced:**

* Implementing multiple simultaneous discounts required careful calculation order
* Stock management during order cancellation needed proper transaction handling
* BCNF normalization for Delivery and OrderCancellation required restructuring from initial design

**Future Enhancements:**

* Integration with actual payment gateways (Stripe, PayPal)
* Email notifications for order updates
* Book recommendations based on purchase history
* Advanced reporting and analytics for administrators
* Wishlist functionality
* Customer reviews and ratings

The project demonstrates practical application of database concepts, web development principles, and business logic implementation in a real-world scenario.