

**Project Title:**

**Book Store WEB APP**

**Course:**

**Software Engineering**

**Submitted By:**

* **Muhammad Ibrahim** (Team Lead)
* **M Abdullah Bin Abdul Aziz**
* **Wajhi ur Rehman**

**Instructor:**

**Adnan Jelani**

**Submission Date:**

**8th December 2024**

Table of Contents

[1. Problem Description 1](#_Toc184585039)

[2. Stakeholders/Users 1](#_Toc184585040)

[3. Feasibility Study 1](#_Toc184585041)

[3.1 Purpose 1](#_Toc184585042)

[3.2 Advantages 1](#_Toc184585043)

[3.3 Disadvantages 1](#_Toc184585044)

[4. Proposed Solution 1](#_Toc184585045)

[4.1 Front-end Design Sketch 2](#_Toc184585046)

[5. Software Requirement Specification Document 4](#_Toc184585047)

[5.1 Functional Requirements 4](#_Toc184585048)

[1. User Functional Requirements: 4](#_Toc184585049)

[2. Admin Functional Requirements: 4](#_Toc184585050)

[5.2 Non-Functional Requirements 5](#_Toc184585051)

[1. Performance Requirements 5](#_Toc184585052)

[2. Scalability Requirements 5](#_Toc184585053)

[3. Usability Requirements 5](#_Toc184585054)

[4. Security Requirements 5](#_Toc184585055)

[5. Reliability Requirements 5](#_Toc184585056)

[6. Maintainability Requirements 6](#_Toc184585057)

[7. Compatibility Requirements 6](#_Toc184585058)

[8. Accessibility Requirements 6](#_Toc184585059)

[9. Ethical Requirements 6](#_Toc184585060)

[5.3 Use Case Diagrams/User Stories 6](#_Toc184585061)

[5.4 Assumptions/Constraints 11](#_Toc184585062)

[6. Project Plan 11](#_Toc184585063)

[6.1 Timeline and Milestones 11](#_Toc184585064)

[6.2 Team Responsibilities 11](#_Toc184585065)

[6.3 Technology Stack and Tools 11](#_Toc184585066)

# 1. Problem Description

The primary problem this application aims to solve is the difficulty of accessing books conveniently. Users often struggle to find a wide range of books with detailed descriptions, reviews, and easy purchasing options. Traditional bookstores are limited by physical space, availability of books and geographical constraints, and existing online solutions may lack user-friendly interfaces or specific features such as author information and personalized recommendations.

# 2. Stakeholders/Users

* **End Users:** Individuals looking to buy books online conveniently.
* **Admin:** Responsible for managing the book inventory, orders, and user accounts.

# 3. Feasibility Study

## 3.1 Purpose

The application is needed to provide an easy-to-use platform where users can browse, purchase, and review books. It aims to bridge the gap between users and bookstores by providing detailed book information and a smooth purchasing process.

## 3.2 Advantages

* Wide range of books available online.
* Enhanced user experience with detailed information and recommendations.
* Convenient for users with limited access to physical bookstores.

## 3.3 Disadvantages

* Initial development costs and time.
* Dependency on a reliable internet connection for access.

# 4. Proposed Solution

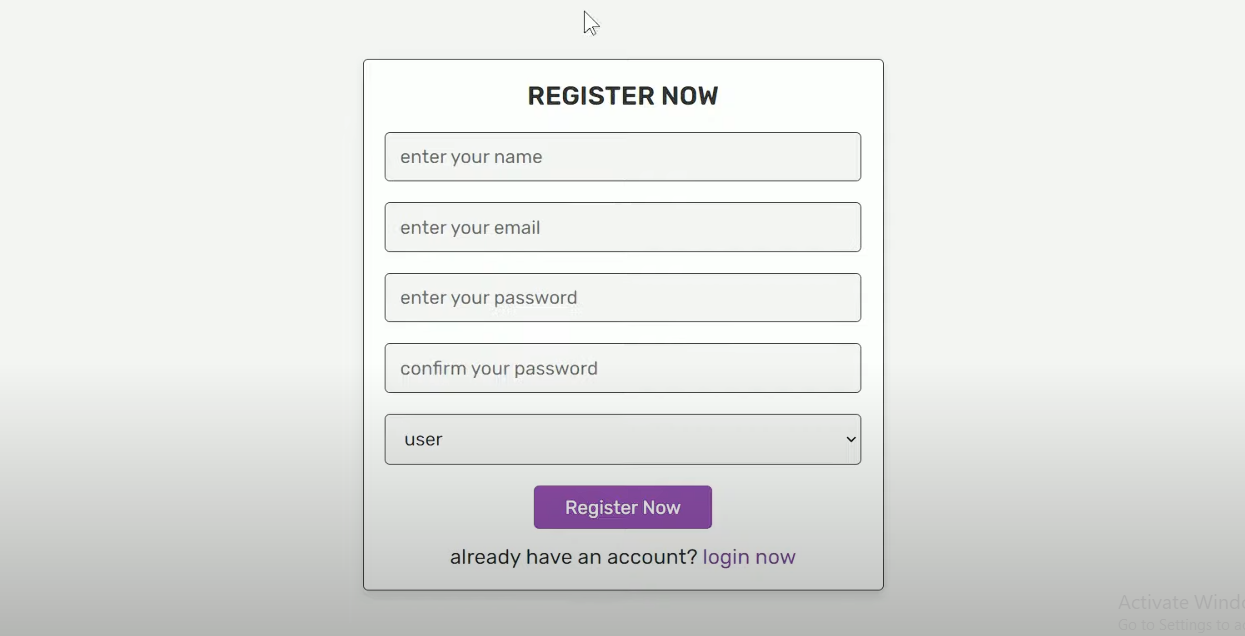
The proposed solution is an online bookstore web application with the following features:

* Users can browse books by category, search books by name, and view details such as summaries, author bios, and reviews.
* A user-friendly shopping cart and checkout system.
* Admin functionalities for managing books and orders.
* Secure user authentication and payment processes.

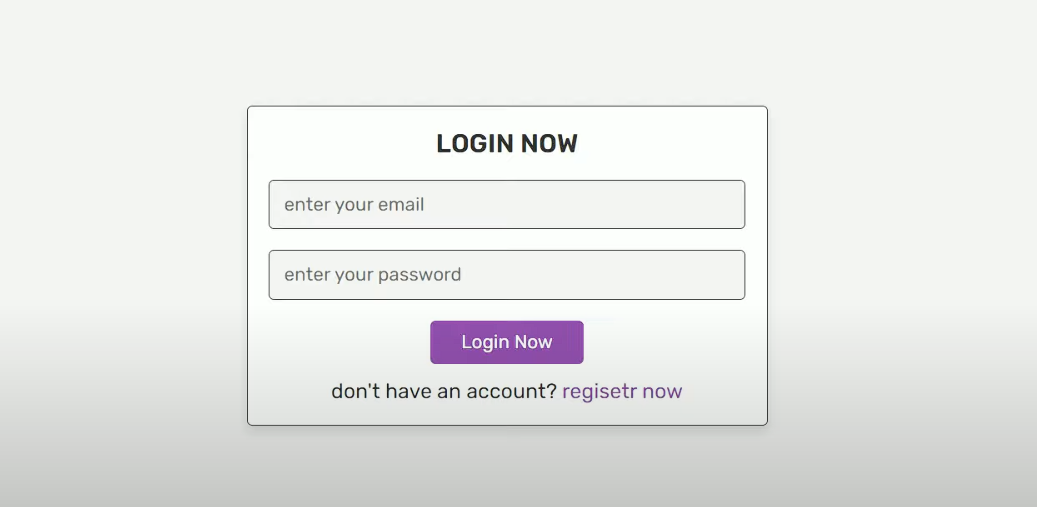
## 4.1 Front-end Design Sketch

The project will include following pages:

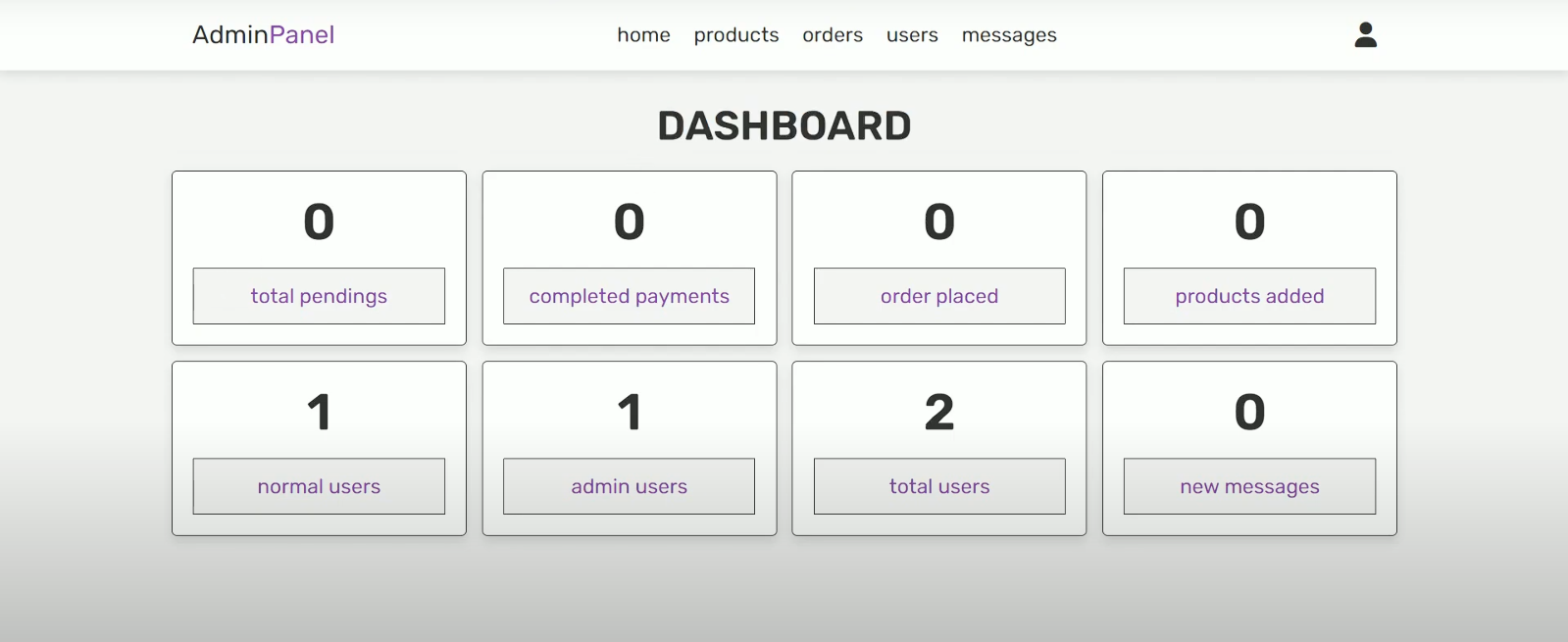
**Register Form:**



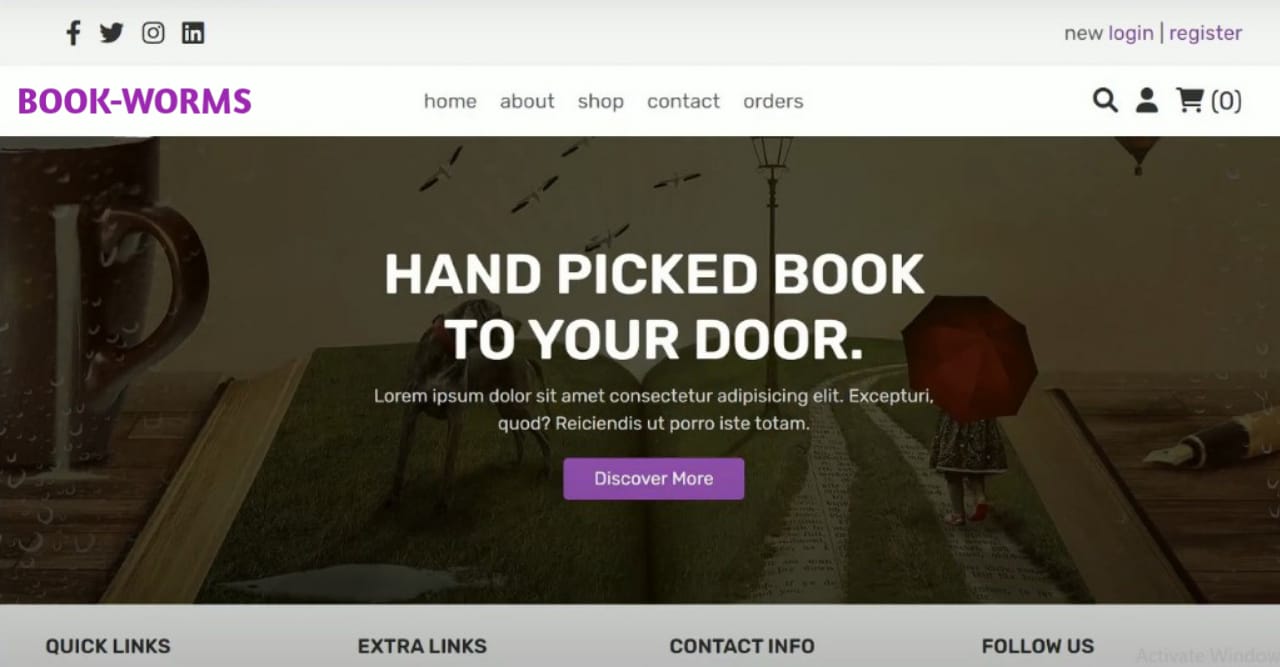
**Login Page:**



**Admin Panel Page:**



**User Home Page:**



# 5. Software Requirement Specification Document

## 5.1 Functional Requirements

### User Functional Requirements:

* **Browse Books:**
* Users should be able to view a list of books categorized by genres, authors, or price ranges.
* **Search Books:**
* Users should be able to search for books by title, author, ISBN, or keywords.
* **View Book Details:**
* Users should see detailed information about each book, including a summary, description, author information, and reviews.
* **User Account Management:**
* Users should be able to register, log in, and log out securely.
* Users should have the ability to update their profile information and passwords.
* **Shopping Cart:**
* Users should be able to add, view, update, and remove books in their cart.
* Display total price and book quantities in the cart.
* **Wishlist:**
* Users should have the option to save books to a Wishlist for future purchases.
* **Checkout and Order Placement:**
* Users should be able to place orders by providing necessary details (e.g., shipping address, payment method).
* Generate an order summary before finalizing the purchase.
* **Order History:**
* Users should be able to view past orders with details like order date, status, and items purchased.
* **Leave Reviews and Ratings:**
* Users should have the ability to rate books and leave reviews visible to other users.
* **Notifications:**
* Provide notifications for successful registration, order placement, and promotional offers.

### Admin Functional Requirements:

* **Book Management:**
* Admin should be able to add, edit, and delete books in the inventory.
* Assign categories and tags to books for better organization.
* **Order Management:**
* Admin should be able to view all placed orders with their statuses (e.g., pending, shipped, delivered).
* Update the status of orders.
* **User Management:**
* Admin should be able to view, update, and delete user accounts if necessary.
* **Analytics and Reports:**
* Provide admins with reports on book sales, user activity, and revenue trends.
* **Promotional Management:**
* Admin should be able to create and manage discount codes or promotional offers.

## 5.2 Non-Functional Requirements

### 1. Performance Requirements

* The system should respond to user activity, for example, pages loading, adding items to cart within few seconds for normal operation.
* It should support multiple users at the same time without considerable performance degradation.
* Database queries should be executed efficiently enough to avoid any kind of problem in operation.

### 2. Scalability Requirements

* The system will be designed to support the growth factor, including new features, books, categories, and user accounts.
* It will support scaling horizontally (adding servers) or vertically (upgrading servers) as traffic increases.

### 3. Usability Requirements

* The application will provide an user-friendly interface.
* Users will be able to navigate the website easily on both desktop and mobile devices.
* All forms and processes (e.g., checkout, registration) will be simple, clear, and validated with user-friendly error messages.

### 4. Security Requirements

* All sensitive information (passwords, personal information) will be encrypted (hashed passwords).
* The system will prevent unauthorized access via user authentication and session management.
* The system will protect against common web threats such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF).

### 5. Reliability Requirements

* The system will have maximum uptime with high availability for users.
* Daily database backups will be made to avoid data loss.
* In the event of a crash, the system will recover in few minutes with redundant servers or backup plans.

### 6. Maintainability Requirements

* The codebase will be based on clean coding practices with comments for readability and easier debugging.
* The system will facilitate easy updates to features, UI, and backend without major disruptions.

### 7. Compatibility Requirements

* The website will be compatible with modern web browsers (e.g., Chrome, Firefox, Safari, Edge).
* It will work smoothly on other operating systems such as Windows, macOS, Android, iOS.
* Responsive design is guaranteed for multiple screen sizes (desktop, tablet, mobile).

### 8. Accessibility Requirements

* The application will be accessible to everyone including those with disabilities under web accessibility guidelines (e.g. WCAG 2.1)
* Features like high contrast mode, readable font and navigation using the keyboard should be there.

### 9. Ethical Requirements

* The application will not collect or make use of user data without consent.

## 5.3 Use Case Diagrams/User Stories

1. **Admin Use Cases:**

* **Book Management:** Add, Edit and Delete book entries. Upload book cover images. Assign categories and tags.
* **Order Management:** View all orders with statuses (pending, shipped, delivered). Update order status.
* **User Management:** View and manage user accounts. Block or delete accounts if necessary.
* **Reports and Analysis:** Track sales trends such as most purchased books, revenue reports.
* **Promotion Management:** Highlight featured or newly added books. Email for new arrivals, or email for discounts.

1. **Admin User Stories:**

Following are few user stories;

* + **Book Management**:

As an admin, I want to add, edit, and delete book entries and upload cover images so that I can keep the inventory up-to-date and visually appealing for users.

* + **Order Management**:

As an admin, I want to view and update order statuses (e.g., pending, shipped, delivered) so that I can efficiently manage customer orders and ensure timely delivery.

* + **User Management**:

As an admin, I want to manage user accounts by viewing, blocking, or deleting them if necessary to maintain a secure and user-friendly platform.

* + **Reports and Analysis**:

As an admin, I want to track sales trends and generate revenue reports so that I can make data-driven decisions to improve the store's performance.

* + **Promotion Management**:

As an admin, I want to highlight featured books and send promotional emails about discounts or new arrivals so that I can engage customers and increase sales.



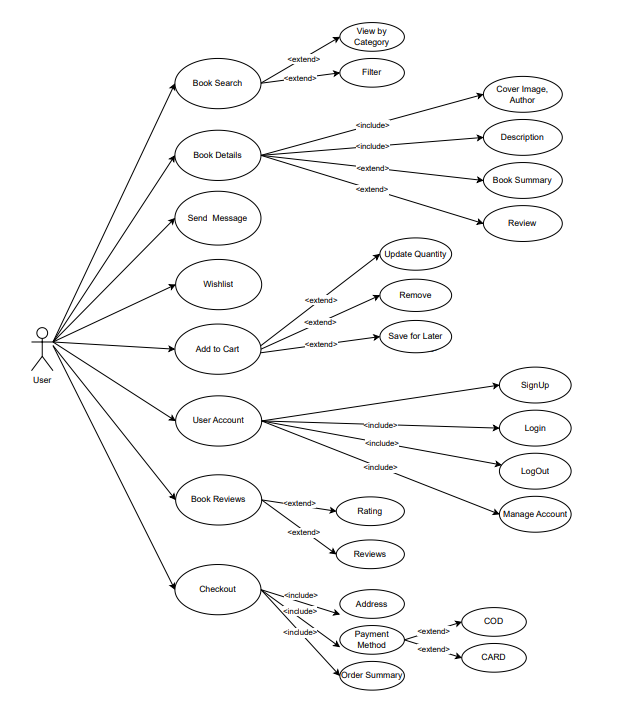
**Admin Use Case Diagram**

1. **User Use Cases:**

* **Book Browsing and Searching** (View books by category (title, author, price range, etc.). Filters (price, rating, publication year).)
* **Book Details** (Display book summary, detailed description, and cover image. Author bio and background. User reviews and ratings.)
* **Shopping Cart** (Add to cart, update quantity in cart, remove items from cart. Save for later option.)
* **Checkout Process** (Address input form and payment method selection (cash on delivery or online payment simulation). Order summary and confirmation.)
* **User Account** (Sign up, log in, and logout functionality. Manage profile (update personal information, password). View order history.)
* **Wishlist** (Add books to a Wishlist for future reference.)
* **Book Reviews** (Users can leave ratings and reviews for books.)
* **Send Message:** User can send message to contact admin.

1. **User User Stories:**

* **Book Browsing and Searching**: As a user, I want to browse books by categories and use filters like price, rating, and publication year so that I can quickly find books of interest.
* **Book Details:** As a user, I want to view detailed book information, including summaries, author bios, and user reviews, so that I can make informed purchasing decisions.
* **Shopping Cart and Wishlist**: As a user, I want to add books to my cart or Wishlist, update quantities, and save items for later so that I can organize my purchases conveniently.
* **Checkout Process**: As a user, I want to provide my address and select a payment method to confirm my order, so that I can complete my purchase efficiently.
* **User Account and Interaction**: As a user, I want to sign up, log in, manage my profile, view my order history, and contact the admin so that I can personalize my experience and resolve issues effectively.



**User Use Case Diagram**

## 5.4 Assumptions/Constraints

* The application assumes users have an internet connection.
* Payment integration will be simulated without real transactions.

# 6. Project Plan

## 6.1 Timeline and Milestones

|  |  |  |
| --- | --- | --- |
| **Phases** | **Tasks** | **Weekly Duration** |
| Phase 1 | **UI/UX Design** | 1-2 |
| Phase 2 | **Frontend Implementation** | 3-4 |
| Phase 3 | **Backend Setup** | 5 |
| Phase 4 | **Incremental Integration** | 6-7 |
| Phase 5 | **Testing and Deployment** | 8 |

The project is divided into weekly tasks as under:

* **Weeks 1-2:** UI/UX Design. Focus on designing a responsive and user-friendly interface to enhance the user experience.
* **Weeks 3-4:** Frontend Implementation. Development of the application's frontend using HTML and CSS, ensuring a clean and intuitive design.
* **Week 5:** Backend Setup. Configuring the backend using PHP and integrating it with a MySQL database for data management.
* **Weeks 6-7:** Incremental Integration. Gradual integration of frontend and backend functionalities, testing features incrementally for compatibility.
* **Week 8:** Testing and Deployment. Conducting comprehensive testing to ensure functionality and deploying the completed application.

## 6.2 Team Responsibilities

* Muhammad Ibrahim: Frontend development (HTML/CSS design).
* M Abdullah Bin Abdul Aziz: Backend development (PHP and database integration).
* Wajhi Ur Rehman: Testing and documentation.

## 6.3 Technology Stack and Tools

* Frontend: HTML, CSS
* Backend: PHP
* Database: MySQL
* Tools: VS Code, XAMPP, JIRA (for Project Management), Canva (for design mockups)