

# Integration Training Assignment

## Objective:

Your task is to design and implement a secure, API-driven online bookstore platform using WSO2 products. The goal is to gain practical understanding of WSO2 product capabilities.

This assignment is designed to simulate an integration project scenario, where you will have to make design decisions, configure components, implement flows, and validate end-to-end functionality.

## Time duration:

7 working days

- To design the solution - 1.5 working days
- To implement the solution - 5.5 working days

## High-Level Use Case:

In this project, the 'books.com' company is planning to re-design and deploy an online bookstore system. The goal is to create a seamless experience for users by developing a frontend application that enables interactions for both customers and staff.

The below table lists the users and tasks they can carry out.

User role	Task	Description
Customer	Browse Books	Customers can browse the available books and view detailed information.
	Make Purchases	Customers can place purchase orders (order status pending).
Staff	Manage Inventory	Staff can add, update, or remove books from the system's inventory.
	Process Orders	Staff can view the purchase order and process the order to complete the purchase (order status complete).
Admin	All the staff activities	
	Validate user registrations	Validate user registrations and approve or reject them.

The system will provide different levels of access to its features. Some functionalities, such as viewing books, will be available to all users without any restrictions. On the other hand, certain actions such as making purchases and processing orders will require users to be logged in and authenticated.

The system will also integrate sending email notifications for purchase confirmations to the customers and a daily purchase summary to the admin user. It is expected to make use of WSO2 IS server as a key manager and make use of its functionalities where possible. And please refer to the Appendix section for the database design.

The system will be designed to handle increasing loads, with a focus on ensuring scalability and performance by leveraging best practices for system integration. To ensure smooth operations, the system will also feature monitoring and logging mechanisms for debugging, tracking API usage, and ensuring the overall health of the system.

You can also add any improvements to the design mentioned above when developing the solution.

## Documentation & Deliverables

- Architecture diagram
- Sample curl/Postman requests with expected outputs
- Troubleshooting notes for issues observed

# Appendix

## Database Design

```
CREATE TABLE books (  
    book_id INT PRIMARY KEY AUTO_INCREMENT,  
    title VARCHAR(255) NOT NULL,  
    price DECIMAL(10, 2) NOT NULL,  
    stock_quantity INT NOT NULL  
);  
  
CREATE TABLE orders (  
    order_id INT PRIMARY KEY AUTO_INCREMENT,  
    user_email VARCHAR(255) NOT NULL,  
    order_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    payment_status ENUM('pending', 'completed') NOT NULL,  
    total_amount DECIMAL(10, 2) NOT NULL  
);  
  
CREATE TABLE order_items (  
    order_item_id INT PRIMARY KEY AUTO_INCREMENT,  
    order_id INT NOT NULL,  
    book_id INT NOT NULL,  
    quantity INT NOT NULL,  
    FOREIGN KEY (order_id) REFERENCES orders(order_id),  
    FOREIGN KEY (book_id) REFERENCES books(book_id)  
);  
  
CREATE TABLE payments (  
    payment_id INT PRIMARY KEY AUTO_INCREMENT,  
    order_id INT NOT NULL,  
    payment_amount DECIMAL(10, 2) NOT NULL,  
    payment_status ENUM('pending', 'completed') NOT NULL,  
    FOREIGN KEY (order_id) REFERENCES orders(order_id)  
);
```