**Intelligent Book Management System Documentation**

**1. Introduction**

The Intelligent Book Management System is a RESTful API designed to manage books and reviews efficiently. It leverages a local Large Language Model (LLM) through Ollama to provide intelligent features like book summaries and recommendations. This system is built using Flask, a lightweight Python web framework, and PostgreSQL as its database. Docker and Docker Compose are employed to simplify development and deployment, ensuring a consistent environment across different systems.

**2. Features**

* **Book Management:**
  + Create, read, update, and delete book records.
  + Retrieve detailed information about individual books.
  + List all books available in the system.
* **Review Management:**
  + Add, read, update, and delete reviews for books.
  + Retrieve all reviews associated with a specific book.
  + User authentication is required to post or modify reviews.
* **LLM Integration:**
  + Generate summaries for books using a local LLM through Ollama.
  + Provide book recommendations based on the content of a given book.
* **User Authentication:**
  + Secure user registration and login using JSON Web Tokens (JWT).
  + Protected API endpoints require authentication.
* **API Documentation:**
  + Interactive API documentation using Swagger UI.
  + Comprehensive API documentation in the README.md file.
* **Containerization:**
  + Docker and Docker Compose for easy setup and deployment.
  + Consistent development and production environments.

**3. Technologies Used**

* **Flask:** Python web framework for building the API.
* **Flask-SQLAlchemy:** SQLAlchemy ORM for database interactions.
* **Flask-Migrate:** Database migration tool for managing schema changes.
* **Flask-JWT-Extended:** JWT library for authentication.
* **python-dotenv:** Loads environment variables from .env files.
* **psycopg2:** PostgreSQL adapter for Python.
* **marshmallow:** Object serialization and deserialization.
* **langchain:** Library for LLM integrations.
* **ollama:** Python library to interact with Ollama.
* **pytest:** Testing framework.
* **Docker:** Containerization platform.
* **Docker Compose:** Tool for defining and running multi-container Docker applications.
* **Swagger UI:** Interactive API documentation tool.
* **PostgreSQL:** Relational database management system.

**4. API Endpoints**

* **/auth/register (POST):** Register a new user.
* **/auth/login (POST):** Authenticate and obtain a JWT.
* **/api/books (GET):** Get all books.
* **/api/books/{book\_id} (GET):** Get a book by ID.
* **/api/books (POST):** Add a new book.
* **/api/books/{book\_id} (PUT):** Update a book.
* **/api/books/{book\_id} (DELETE):** Delete a book.
* **/api/books/{book\_id}/reviews (GET):** Get reviews for a book.
* **/api/books/{book\_id}/reviews (POST):** Add a review to a book.
* **/api/reviews/{review\_id} (GET):** Get a review by ID.
* **/api/reviews/{review\_id} (PUT):** Update a review.
* **/api/reviews/{review\_id} (DELETE):** Delete a review.
* **/api/books/{book\_id}/summary (POST):** Generate a book summary.
* **/api/books/{book\_id}/recommendations (GET):** Get book recommendations.

(Add details for each endpoint, including request/response examples and error codes.)

**5. Testing**

Run unit tests using pytest:  
  
 Bash  
pytest

* Tests are located in the tests/ directory.

**7. Deployment (Production)**

* Use a production-ready WSGI server
* Implement a reverse proxy.
* Configure logging and monitoring.
* Secure environment variables