**CODEBASE 1: Book Management System Backend using FastAPI:**

Link: [ACE\_Python\_codebases / codebase1 · GitLab](https://gitrepo.valuelabs.com/ace_python_codebases/codebase1) (Branch: Master)

### Basic Overview of Repo

This is a FastAPI-based backend system that supports typical CRUD operations on books and authors. It uses PostgreSQL as the database and SQLAlchemy for ORM. Authentication and role management are also included.

**1. Assignment 1: Full-Text Search on Books and Authors**

**Objective of Assignment**

Add full-text search capability for books and authors using PostgreSQL's native search.

### Detailed Assignment

* Implement a /search?q={query} endpoint.
* Use PostgreSQL’s tsvector or ilike query fallback to search:
  + Book title
  + Book description
  + Author name
* Optional: Add weights to fields (e.g., title > description).
* Return ranked results in response.

### End Goal Which Will Be Evaluated

* Effectiveness of search results.
* Use of appropriate DB-level optimization.
* Schema updates and query logic.
* Performance under large datasets.

**2. Assignment 2: Role-Based Access Control (RBAC) Enhancement**

**Objective of Assignment**

Introduce middleware-based role validation to restrict access to certain endpoints based on roles (admin, editor, reader).

### Detailed Assignment

* Enhance the existing User model to support multiple roles.
* Create a decorator or middleware to check role permissions.
* Apply role restrictions to endpoints like:
* Only admin can delete books or authors.
* Only editor and above can update book details.
* reader role can only read data.
* Add a secure way to assign/update roles via an endpoint.

### End Goal Which Will Be Evaluated

* Proper segregation of access.
* Secure implementation.
* Reusability of permission logic.
* Clean and testable architecture.

**3. Assignment 3: Implement Book Recommendation Engine**

**Objective of Assignment**

Enhance the system by integrating a book recommendation feature based on book genres and author similarity.

### Detailed Assignment

* Create a new /recommendations/{user\_id} endpoint.
* When a request is made to this endpoint:
* Retrieve the books read by the user (use a simple reading\_history table that you must add).
* Based on genre and author similarity, return 5 recommended books that the user hasn’t read yet.

### End Goal Which Will Be Evaluated

* Correctness and relevance of recommendations.
* Schema design and extensibility.
* Code structure and reuse (services/models separation).
* Error handling and edge case coverage.