HiggsBosonHealth

Take-Home Assignment: Online Books Library API

Background:

You are tasked with building a RESTful API for an online books library system. The library offers subscription plans for its users, allowing them to borrow books and magazines based on their plan. The subscription plans are as follows:

- **Silver Plan**: Allows users to borrow up to 2 books but no magazines.
- **Gold Plan**: Allows users to borrow up to 3 books and 1 magazine.
- **Platinum Plan**: Allows users to borrow up to 4 books and 2 magazines.

Apart from the subscription plan the library also follows following two rules:

- Books of Crime genre are only available for users with 18+ age.
- User can perform only 10 transactions in a month.

Requirements:

You need to design and implement the following two APIs:

Order API:

- This API should allow users to order a book or magazine.
- The API should accept the title (or any other unique identifier) of the book/ magazine as input.
- The API should enforce the borrowing limits based on the user's subscription plan.
- If the user is eligible to borrow the requested item, the API should process the order and return a success message.
- If the user is not eligible (e.g., exceeded the borrowing limit or requested an unavailable item), the API should return an appropriate error message.

Return API:

- This API should allow users to return one or more borrowed books or magazines.
- The API should accept a list of titles (or any other unique identifiers) of the items to be returned.
- The API should update the availability status of the returned items and return a success message.

Evaluation Criteria
The primary objective of the assignment is to understand how do you model the problem and enforce the system constraints.

You can choose to implement the APIs using Ruby (e.g., Ruby on Rails), Python (e.g., Django or Flask), or Java (e.g., Spring Boot).

You do not need to implement authentication for this assignment.

Your submission will be evaluated based on the following criteria:

- Code quality, structure, and adherence to best practices
- Implementation of required features and functionality
- Proper modeling of entities and relationships
- Enforcement of borrowing limits and constraints
- Test coverage and quality of test cases
- API documentation quality and completeness
- Overall project organization and presentation

Submission
Please submit your assignment as a ZIP file. Include the `.git` folder as well, we would like to see how you've arrived at the solution.

Include a README file with detailed instructions on how to set up and run the application locally, as well as any assumptions or design decisions made.

Feel free to ask any clarifying questions if you need further information or have any concerns regarding this take-home assignment.