Тестовое Задание: Разработка API для Управления Библиотекой

Описание проекта:
Вам необходимо разработать RESTful API для управления библиотечным каталогом. Система должна позволять управлять информацией о книгах, авторах, читателях и выдачей книг.

Требования к функционалу:

1. **Авторизация и Аутентификация:**
   - Реализовать регистрацию и аутентификацию пользователей с использованием JWT токенов.
   - Разделить пользователей на роли: администратор и читатель.
   - Администратор может управлять всеми ресурсами, читатель — только просмотр и взаимодействие с книгами.

2. **Управление книгами:**
   - CRUD операции для книг.
   - Каждая книга должна иметь следующие поля:
     - ID
     - Название
     - Описание
     - Дата публикации
     - Автор(ы) (связь с таблицей авторов)
     - Жанр(ы)
     - Количество доступных экземпляров

3. **Управление авторами:**
   - CRUD операции для авторов.
   - Поля автора:
     - ID
     - Имя
     - Биография
     - Дата рождения

4. **Управление читателями:**
   - Администратор может просматривать список читателей.
   - Читатели могут обновлять свою информацию.

5. **Выдача и возврат книг:**
   - Возможность выдачи книги читателю.
   - Ограничить количество одновременно выдаваемых книг на одного читателя до 5.
   - Фиксировать дату выдачи и предполагаемую дату возврата.
   - Обрабатывать возврат книг и обновлять количество доступных экземпляров.

6. **Дополнительные требования:**
   - Реализовать пагинацию и фильтрацию для списков книг и авторов.

- Валидация входящих данных с использованием Pydantic.
- Обработка ошибок с соответствующими HTTP статусами.
- Логирование основных событий (например, выдача и возврат книг).
- Использовать Alembic для управления миграциями базы данных.
- Написать юнит-тесты для основных эндпоинтов.

Технические требования

- Язык программирования: Python 3.8+
- Фреймворк: FastAPI
- База данных: PostgreSQL
- ORM: SQLAlchemy
- Аутентификация: JWT
- Управление миграциями: Alembic
- Тестирование: Pytest

Дополнительные условия

- Репозиторий должен быть инициализирован с использованием Git и содержать осмысленные коммиты.
- Код должен быть оформлен по стандартам PEP8.
- Подготовить README с инструкциями по запуску проекта, описанием архитектуры и любыми другими важными деталями.
- Использовать Docker для контейнеризации приложения и базы данных (необязательно, но приветствуется).

Критерии оценки

1. Функциональность: Все ли требуемые функции реализованы корректно.
2. Кодовая база: Чистота и читаемость кода, использование лучших практик.
3. Структура проекта: Логичная и удобная структура проекта.
4. Документация: Наличие и качество документации, описывающей процесс установки и использования.
5. Тестирование: Покрытие кода тестами, их качество и эффективность.
6. Использование инструментов: Корректное использование FastAPI, SQLAlchemy, Alembic и других технологий.
7. Решение нестандартных задач: Способность кандидата решать задачи, выходящие за рамки базового функционала.

Примечание: Это задание направлено на оценку вашего умения проектировать, реализовывать и тестировать API, а также следовать лучшим практикам разработки. Пожалуйста, постарайтесь учесть все требования и продемонстрировать свою способность создавать надежные и масштабируемые решения.

Удачи!