TASK
Please provide an API implementation for the backend of a simple book store. No frontend required.

MODELS
- Book: isbn, title, authors, language, length, price, genre
- Author: first name, last name, age, nationality
- Order: books, price, total price of the order, shipping data, order date, buyer (user)
- User: first name, last name, username, password, role (buyer/admin)

ENDPOINTS
- Books CRUD + batch delete (read-only for buyers)
- Authors CRUD + batch delete (read-only for buyers)
- Orders CRUD + batch delete
- Listing of all books filtered by author, title and genre
- Order reports:
    - Most popular books in provided time range
    - Most popular author in genre
- Users CRUD (accessible only by admins)
- Auth endpoint (all users are able to log in, buyers are able to create an account)

DOD
- Authentication via a session token returned from the auth endpoint
- Tokens should have a configurable expiry date, after which they become invalid
- Session tokens are stored along with information on which IP/user-agent requested it
- Database setup handled via migrations
- Graceful error handling
- The solution shared as a Git repository, e.g. via Github, or archived and sent via email

PREFERABLE TECHSTACK
- Flask
- PostgreSQL and SQLAlchemy
- Docker and Docker Compose

What will we be looking into
- Quality of your codebase and tests
- API and codebase setup documentation
- Flexibility of ORM mapping
- Thoughtfulness of the database design
- Configuration and architecture
- Your approach toward security
- Tidiness of your Git workflow