Technical Assignment, Node.js/NestJS Backend developer

Project: Advanced Bookstore API

Objective: Develop a RESTful API for a bookstore application. The API will allow users to perform CRUD operations on books, authors, and genres. Additionally, users should be able to search for books by title, author, or genre.

Requirements

Project Setup:

- Use Node.js with the NestJS framework.
- Use TypeScript as the programming language.
- Utilize a relational database (PostgreSQL or MySQL).
- Use TypeORM or Prisma for database interactions.
- Implement authentication and authorization using JWT.

Entities:

- **Book:**
  - `id`: UUID
  - `title`: string
  - `description`: string
  - `publicationDate`: Date
  - `authorId`: UUID (foreign key to Author)
  - `genreId`: UUID (foreign key to Genre)
- **Author:**
  - `id`: UUID
  - `name`: string
  - `biography`: string
  - `birthDate`: Date
- **Genre:**
  - `id`: UUID
  - `name`: string

Endpoints:

- **Books:**
  - `GET /books`: Retrieve a list of all books.
  - `GET /books/:id`: Retrieve a single book by ID.
  - `POST /books`: Create a new book.
  - `PUT /books/:id`: Update an existing book by ID.
  - `DELETE /books/:id`: Delete a book by ID.
  - `GET /books/search`: Search for books by title, author, or genre.
- **Authors:**
  - `GET /authors`: Retrieve a list of all authors.
  - `GET /authors/:id`: Retrieve a single author by ID.
  - `POST /authors`: Create a new author.
  - `PUT /authors/:id`: Update an existing author by ID.
  - `DELETE /authors/:id`: Delete an author by ID.
- **Genres:**
  - `GET /genres`: Retrieve a list of all genres.
  - `GET /genres/:id`: Retrieve a single genre by ID.
  - `POST /genres`: Create a new genre.
  - `PUT /genres/:id`: Update an existing genre by ID.
  - `DELETE /genres/:id`: Delete a genre by ID

Authentication & Authorization:

- Implement user registration and login endpoints.
- Protect book creation, update, and delete endpoints with JWT authentication.
- Only authenticated users can perform CRUD operations.

Validation & Error Handling:

- Validate request payloads using DTOs and class-validator.
- Implement global exception handling and proper error messages.

Testing:

- Write unit tests for service and controller layers.
- Write integration tests for the endpoints.

Plus Section (Optional for Additional Points):

1. **Caching:**
    - Cache the responses for book list and individual book details..
    - Implement cache invalidation when a book is created, updated, or deleted.
2. **Queuing:**
    - Use and handle asynchronous tasks, such as sending notifications when a new book is added.
    - Implement a worker that processes these queued tasks.
3. **Design Patterns:**
    - Apply and use design patterns. For example the Observer pattern for handling events (e.g., new book added).

Deliverables:

- A GitHub repository containing the Laravel project.
- A README file with instructions on how to set up and test the API.

Evaluation Criteria:

1. **Code Quality**
    - Adherence to coding standards and best practices.
    - Proper use of TypeScript features.
    - Clean and readable code with appropriate comments.
2. **Architecture & Design**
    - Proper use of NestJS modules, controllers, and services.
    - Clear separation of concerns and modularity.
    - Use of dependency injection.
3. **Functionality**
    - All required features and endpoints are implemented correctly.
    - Proper handling of authentication and authorization.
    - Effective validation and error handling.
4. **Database Design & Efficient database queries**
5. **Coverage and quality of unit and integration tests**
6. **Documentation**

By evaluating these criteria, we aim to find a candidate who not only possesses technical skills but also follows best practices in software development, can write maintainable and scalable code, and understands advanced programming concepts and design patterns

We encourage you to put all your effort and approach this assignment as a reflection of your professional capabilities.

We wish you the best of luck with the assignment

looking forward to your submission.