Interview Test: Building a REST API for Articles (.NET)

Objective:

This test aims to assess your understanding of key concepts in building a REST API for articles using .NET. It will evaluate your knowledge of:

- **.NET API framework:** Creating controllers, routing, and handling HTTP requests.
- **Data models:** Defining models representing articles and handling data access (simulated data or mock repository can be used).
- **RESTful principles:** Designing resource-oriented endpoints following REST conventions.
- **Serialization:** Converting data objects to JSON for API responses.

Scenario:

Develop a basic REST API for managing articles using .NET Web API. Any minimum two of the following functionalities are required:

1. **GET /api/articles:** Retrieves a list of all articles.
2. **GET /api/articles/{id}:** Retrieves a single article by its ID.
3. **POST /api/articles:** Creates a new article.
4. **PUT /api/articles/{id}:** Updates an existing article by its ID.
5. **DELETE /api/articles/{id}:** Deletes an article by its ID.

Technical Requirements:

- Use .NET 8
- Define a model class representing an article with properties like `id`, `title`, `content`, and `publishedDate`.
- Implement appropriate HTTP verbs (GET, POST, PUT, DELETE) for endpoints.
- Handle error scenarios, returning appropriate HTTP status codes and error messages.
- Use JSON serialization for request and response payloads.

Bonus points:

- Implement basic authentication or authorization mechanisms.
- Include pagination for retrieving large lists of articles.
- Add validation logic for article data on creation and update.
- Utilize dependency injection for data access layer.

Deliverables:

- Gitlab / Github source code for the completed API project.

- A brief document outlining any assumptions and design decisions made.

Evaluation Criteria:

- Functionality and correctness of the implemented endpoints.
- Adherence to RESTful principles and best practices.
- Code quality, readability, and maintainability.
- Documentation and clarity of design choices.

Additional Notes:

- You are free to choose any specific version of .NET and related libraries for this project.
- No specific database connection is required. Simulating data or using a mock repository is acceptable.
- Feel free to ask clarifying questions