Take-Home Assignment
Backend Engineer (C#, .NET)
2024

H!

Congratulations on advancing in our hiring process! We're thrilled to have you on this journey with us.

As our Recruitment Specialist mentioned, this stage involves a take-home assignment designed to give us insight into your skills and work approach. It's also an opportunity for you to get a feel for the challenges our team tackles daily.

Included below, you'll find detailed instructions outlining the task requirements. Please review them carefully, as a clear understanding of the expectations is key to completing the assignment successfully.

Key Details:

- **Deadline**: The assignment must be completed and submitted within **5 days** of receiving it.
- **Submission**: Please send your completed work to our Recruitment Specialist via email. If applicable, you can use a GitHub link—just ensure all files are accessible and functional.

If you have any questions or need clarification, don't hesitate to reach out to our Recruitment Specialist.

Good luck - we can't wait to see your work! 🤞

Best regards,
Recruitment team
4Create Software

Task: Build REST API for Uploading and Processing JSON Metadata Files

Objective: Develop a robust and scalable .NET 8 RESTful API that allows users to upload small files containing JSON metadata related to clinical trials. Upon receiving a file, the service should:

- Validate the JSON metadata against a provided JSON schema.
- Normalize and transform the data according to specific business rules.
- Store the processed data into a SQL database of your choice.
- Provide endpoints to retrieve the processed data.
- Include logging, error handling, and tests to ensure code quality.
- Containerize the application using Docker to ensure consistency across environments.

Functional Requirements:

1. **File Upload Constraints:**

- File Size Limitation: Restrict the size of uploaded files to prevent excessively large uploads.
- File Type Validation: Accept only files with a .json extension.

2. **JSON Validation:**

- Schema Validation: Validate the uploaded JSON file against the provided JSON schema.
- Schema Storage: Store the JSON schema as an embedded resource or in a database of your choice.

3. **Data Conformance and Storage:**

- Database Schema Alignment: Ensure the data conforms to the database schema before storage.
- Data Mapping: Map the JSON data to corresponding classes and tables using an Object-Relational Mapping (ORM) tool like Entity Framework (EF) Code First.

4. **Data Retrieval Endpoints:**

- Get Specific Record by ID: Implement an endpoint to retrieve a specific record using its unique identifier.
- Filtering Support: Allow filtering of records based on query parameters (e.g., status).

5. **Testing:**

- Unit and Integration Tests: Write unit or integration tests for critical components to ensure reliability and maintainability.

6. **API Documentation:**

- Swagger/ OpenAPI: Utilize Swagger/ OpenAPI to generate interactive API documentation.
- README File: If necessary, provide a README file with setup instructions and usage guidelines.

7. **Architectural Design:**

- Clean Architecture: Utilize Clean Architecture principles to separate concerns, promote scalability, and facilitate maintainability.
- Code Quality: Write clean, maintainable, and testable code following industry best practices.

8. **Containerization:**

- Docker: Containerize the application using Docker to ensure consistency across different environments and ease of deployment.

JSON Schema Example:

```
{
	"$schema": "http://json-schema.org/draft-07/schema#",
	"title": "ClinicalTrialMetadata",
	"type": "object",
	"properties": {
		"trialId": {
			"type": "string"
		},
		"title": {
			"type": "string"
		},
		"startDate": {
			"type": "string",
			"format": "date"
		},
		"endDate": {
			"type": "string",
			"format": "date"
		},
		"participants": {
			"type": "integer",
			"minimum": 1
		},
		"status": {
			"type": "string",
			"enum": [
				"Not Started",
```

```
                        "Ongoing",
                        "Completed"
                    ]
                }
        },
        "required": [
                "trialId",
                "title",
                "startDate",
                "status"
        ],
        "additionalProperties": false
}
```

Business Rules for Data Transformation:

1. **Duration Calculation:**

- Calculate the duration of the trial in days and store it as an additional field.

2. **Default endDate:**

- Set the endDate to one month from the startDate if it is not provided and the status is "Ongoing".