

Тестовое задание – Library Web-application

Вступительная задача для стажировки по направлению .NET

CONTENTS

1.	ОПИСАНИЕ	3
1.1.	ФУНКЦИОНАЛ WEB API	3
1.2.	ИНФОРМАЦИЯ О КНИГЕ	3
1.3.	ИНФОРМАЦИЯ ОБ АВТОРЕ	4
1.4.	РАЗРАБОТКА КЛИЕНТСКОЙ ЧАСТИ ПРИЛОЖЕНИЯ	4
1.5.	ТРЕБОВАНИЯ К WEB API	4
1.6.	НЕОБХОДИМЫЕ К ИСПОЛЬЗОВАНИЮ ТЕХНОЛОГИИ	5
1.7.	АРХИТЕКТУРА.....	5

1. Описание

Разработка web-приложения для имитации библиотеки (создание, изменение, удаление, получение), выполняется на .Net Core, с использованием EF Core. Для разработки клиентской части приложения следует использовать Angular или React.

Должна прилагаться инструкция по запуску проекта.

1.1. Функционал Web API

Работа с книгами:

1. Получение списка всех книг;
2. Получение определенной книги по её Id;
3. Получение книги по её ISBN;
4. Добавление новой книги;
5. Изменение информации о существующей книге;
6. Удаление книги;
7. Выдача книг на руки пользователю;
8. Возможность добавления изображения к книге и его хранение;
9. *Отправка уведомления об истечении срока выдачи книги (будет плюсом).

Работа с авторами:

1. Получение списка всех авторов;
2. Получение автора по его Id;
3. Добавление нового автора;
4. Изменение информации о существующем авторе;
5. Удаление автора;
6. Получение всех книг по автору.

1.2. Информация о книге

1. ISBN;
2. Название;
3. Жанр;
4. Описание;
5. Автор;
6. Время, когда книгу взяли;

7. Время, когда книгу надо вернуть.

1.3. Информация об авторе

1. Имя;
2. Фамилия;
3. Дата рождения;
4. Страна.

1.4. Разработка клиентской части приложения

1. Должна быть реализована страница регистрации/аутентификации;
2. Реализовать страницу отображения списка книг. Если все книги с данным автором и названием были взяты из библиотеки, отображать, что книги нет в наличии;
3. Реализовать страницу, отображающую информацию о конкретной книге. Если книга в наличии, пользователь может ее взять;
4. Для админа реализовать страницу добавления/редактирования книги;
5. Для админа на странице с информацией о книге должны быть кнопки "Редактировать", "Удалить". При нажатии на кнопку редактирования происходит редирект на страницу редактирования/добавления книги. При нажатии на кнопку удаления открывается модальное окно с подтверждением об удалении книги;
6. У каждого пользователя должна быть страница, на которой он сможет просматривать книги, которые взял из библиотеки;
7. Реализовать пагинацию, поиск по названию книги, фильтрацию по жанру/автору.

1.5. Требования к Web API

1. Реализация policy-based авторизации с использованием refresh и jwt access токенов;
2. Внедрение паттерна репозиторий и Unit of Work;
3. Разработка middleware для глобальной обработки исключений;
4. Реализация пагинации;

5. Обеспечение покрытия unit-тестами всех use cases одной модели и двух методов ее репозитория (один на получение и один на сохранение данных). Модель выбирается на усмотрение кандидата (для всех моделей должны быть написаны use cases). Разрешено использовать InMemoryDatabase;
6. *Внедрение кеширования изображений (будет плюсом).

1.6. Необходимые к использованию технологии

1. .Net 5.0+;
2. Entity Framework Core;
3. MS SQL / PostgreSQL or any other;
4. AutoMapper / Mapster or any other;
5. FluentValidation;
6. Authentication via jwt access & refresh token (ex.: IdentityServer4);
7. Swagger;
8. EF Fluent API;
9. Angular/React;
- 10.xUnit/nUnit.

1.7. Архитектура

Приложение должно быть разработано на чистой архитектуре.

Перед выполнением задачи- сообщите HR сколько потребуется времени на всю реализацию (в часах)!!!