Тестовое задание c#.

Вам нужно создать ASP.NET Core Web API, который будет предоставлять CRUD (Create, Read, Update, Delete) операции для управления списком пользователей. У каждого пользователя должны быть следующие атрибуты: Id, Имя (Name), Возраст (Age), Email и связанная сущность "Роль" (Role) со значениями (User, Admin, Support, SuperAdmin), где у одного юзера может несколько ролей.

**Требования:**

-Создайте новый проект ASP.NET Core Web API с использованием языка C# и .NET Core.

-Создайте модель данных для пользователя (User) с атрибутами Id, Name, Age, Email и связанной сущностью "Роль" (Role).

-Создайте контроллер (UserController) с методами для выполнения следующих операций:

-Получение списка всех пользователей (ОБЯЗАТЕЛЬНО реализовать пагинацию, сортировку и фильтрацию по каждому атрибуту в модели User и в модели Role).

-Получение пользователя по Id и всех его ролей.

-Добавление новой роли пользователю.

-Создание нового пользователя.

-Обновление информации о пользователе по Id.

-Удаление пользователя по Id.

-Добавьте в контроллер бизнес-логику для валидации данных:

-Проверка наличия обязательных полей (Имя, Возраст, Email).

-Проверка уникальности Email.

-Проверка возраста на положительное число.

-Используйте Entity Framework Core (или любой другой ORM по вашему выбору) для доступа к данным и сохранения пользователей и ролей в базе данных.

-Создайте миграцию использую ORM для создания необходимой таблицы в базе данных.

-Добавьте обработку ошибок и возврат соответствующих статусных кодов HTTP (например, 404 при отсутствии пользователя).

-Документируйте ваш API с использованием инструментов Swagger или подобных.

**Дополнительные задания (по желанию):**

-Добавьте аутентификацию и авторизацию к вашему API с использованием JWT-токенов.

-Настройте логирование действий в API (например, с использованием Serilog).

Инструкции по предоставлению решения:

-Создайте проект и разработайте API, следуя вышеуказанным требованиям.

-Загрузите ваш проект в репозиторий на GitHub или другую платформу управления версиями.

-Предоставьте ссылку на ваш репозиторий.

Оценка:

-Ваше решение будет оцениваться на основе следующих критериев:

-Качество кода и его чистота.

-Соблюдение требований и функциональности API.

-Обработка ошибок и возвращаемые статусные коды.

-Эффективное использование Entity Framework Core (или другой ORM).

-Документация API.

-Работа с связанными данными.

Дополнительные задания (по желанию).