Веб-приложение: Angular + ASP.NET (Web Api) + PostgreSQL (DBeaver)

Учебный предмет в колледже, для которого пишу проект: Техника коммуникации и основы командообразования.

На этом предмете мы проходим всё с этим связанное, например:

* как вести себя на публике, готовиться к публичным выступлениям, не волноваться на них;
* как находить общий язык в команде и всё в таком духе;
* как правильно составлять резюме (можно в приложение встроить конструктор для этого, но это если время останется);
* прикольные игрульки для игры в команде (но это уже потом, если останется время, а его не останется)

В чем суть платформы: т.к. предмет один и программа у всех uhegg одинаковая, препод будет создавать не «новый курс», а «новую группу». И когда он будет выкладывать лекции/практические занятия – он сможет с помощью «чекбоксов» проставить, какие группы должны видеть этот материал, а какие нет (соответственно, после публикации материала тоже можно это проставить). Это потому, что что-то все же может отличаться у разных групп (допустим, препод решил выложить таблицу успеваемости в группе, и хочет открыть её только для одной конкретной группы)

У нас есть три «роли»: администратор, преподаватель, обучающийся.

Администратор: человек, который имеет доступ ко всей базе данных. И, по идее, он может создавать/удалять преподам аккаунты (либо же проводить верификацию после того, как препод попробовал зарегистрироваться самостоятельно). По факту, функционал у него тот же, что у преподов, только он вместо того, чтобы видеть действия одного преподавателя, может видеть действия всех: он видит материалы, опубликованные ВСЕМИ преподами; он может просматривать успеваемость, статистику по ВСЕМ группам у ВСЕХ преподов и т.д. Функционал администратора будет делаться уже исходя из того, что есть в функционале препода.

Преподаватель: создаёт новые группы, публикует для них материалы: лекции (может прикрепить любой файл, либо же написать его самостоятельно через встроенный редактор) и практические занятия (прикрепить файл/тест/интерактив какой-то). Преподаватель сможет видеть (и менять, об этом позже) статистику, сколько в какой группе занятий пройдено, сколько осталось, какой средний балл у группы, у всех ли сданы задания и т.д.

Обучающийся: вводит код группы (препод даст его лично группе), чтобы студенты могли присоединиться к ней. Сдаёт задания, комментирует их (общие комментарии и комментарии лично преподавателю).

И это всё это должно быть реализовано гибко: препод может удалить студента из группы; он может студенту, у которого задание числится невыполненным, поставить галочку «Задание выполнено» (например, студент сдал задание преподу лично, на платформу не загрузил, или они как-то договорились); всячески менять статистику. В общем, везде, где потенциально хоть на 1% возможно что-то отредактировать – эта функция должна быть.

Теперь картина приложения:

**Users**

ID (?)

Почта/номер телефона

Хэш пароля

Фамилия

Имя

Отчество

Это преподаватель (да/нет)

Дата рождения

Препод: + группы, у которых преподает

Студент: + реализовать:

хранение всех ответов, прикрепленных студентом файлов к разным заданиям (или если студент нажал «Отметить как выполненное» вместо прикрепления файла)

какие задания выполнил, а какие нет

информация об оценках студента

средний балл

**Группа**

Студенты группы (айдишник)

Занятия, доступные группе (?)

Средний балл группы

**Задание/занятие**

Дата публикации

Срок сдачи

Заголовок

Описание

Путь к прикрепленному вложению (файлы/тест (гугл формы)

Названия вложения

::Прикрепленные лекции (написанные внутри редактора) – если успею реализовать редактор

Раздел занятия

Каким группам занятие доступно, а каким нет. Каким отдельным пользователям доступно (?)

? Студенты, которые выполнили/не выполнили задание

**Препод**

+ плюшки обычного юзера (ФИО, ДР, почта/телефон, хэш пароля)

Стаж

Группы, у которых ведёт предмет

**Запланированные мероприятия**

Дата и время начала мероприятия

Окончание мероприятия

Преподаватель

Локация

Приглашённые студенты (можно выбрать по группам, либо кого-то лично)

Запланированные мероприятия сможет создавать как студент, так и преподаватель. Должна быть гибкость.