MVC: Фотоальбом с валидацией изображений.

Приложение для размещения фотографий в публичном доступе.

Список страниц к реализации (некоторые из них могут быть рализованы в виде диалогов):

- Страница входа в систему
- Страница регистрации нового пользователя
- Страница со списком последних публикаций
- Сраницы пользователя:
  - Личный кабинет
  - Страница для создания альбома
  - Страница администрирования своих альбомов
- Страница просмотра публичных альбомов, фотографий
- Обеспечить валидацию вводимых данных. Необходимые поля для регистрации взять из личного кабинета, раздел "персональная информация".

Личный кабинет пользователя должен содержать следующую персональную информацию:

- Имя
- Фамилия
- Псевдоним
- E-mail
- Адресс

Работа с альбомами пользователя:

- Предусмотреть функцию добавления/удаления изображений в альбом,
- перехода на страницу просмотра изображения в увеличенном размере.

PS: Вместо реализации собственной системы загрузки картинок, можно оперировать ссылками на изображения из других источников сети или локальной машины.

Особенности реализации:

Перед публикацией фотография должна пройти процесс валидации. Например, пройти проверку на предмет наличия запрещенного контента (нецензурные жесты,

обнаженные фигуры и т.д.).

Для валидации контента изображения *можно* воспользоваться следующим сервисом: https://algorithmia.com/algorithms/sfw/NudityDetection (http://geektimes.ru/post/252442/). Серверный код должен будет вызвать API данного сервиса для проверки изобажения в этому случае.

Посколько валидация изображений может реализовывать различные алгоритмы проверки в метод валидации можно добавить Thread.Sleep, который будет эмулировать задержку в работе алгоритмов. А учитывая что время валидации может быть более одной секунды, можно воспользоваться модификатором async.

PS: Чтобы не терять время при разработке и в целях тестирования, можно реализовать фейковый-сервис проверки, который будет засыпать на некоторое время и возвращать рандомный результат проверки. В конце же можно будет заменить реализацию данного сервиса на рабочую версию сервиса проверки (например на сервис, который будет использовать стороннее API упомянутое выше).

Страница со списком последних публикаций

- Реализовать возможность перехода к просмотру публикации в увеличенном варианте или к альбому пользователя.
- Реализовать автоматическую подгруздку n-ного (на свой выбор) числа картинок по достижению нижнего конца страницы.

Дополнительные задания

1. Реализовать множественность альбомов
2. Реализовать загрузку изображений в систему
3. Реализовать сортировку публикаций по сфере деятельности их авторов (предусмотреть в этом случае наличие такого поля у пользователя) или другим критериям.

Примечание

Для загрузки списка картинок можно реализовать и использовать на сервере OData Controller

Этапы разработки

1. Реализовать БД использую подход DB-frist. В результате должны быть SQL скрипты, которые создадут всю стркутуру базы данных. Также должен быть скрипт, который заполнит базу тестовыми данными.

2. По готовности БД выдам 4 Sql запроса на реализацию.
3. Создать структуру приложения (проекты, модели, интерфейсы репозиториев/ сервисов)
4. Подключить EF
5. Реализация серверной логики
6. Реализация клиентской логики с использованием Angular, Knockout (выбор по желанию)

PS: пункты 5,6 могут выполнятся параллельно