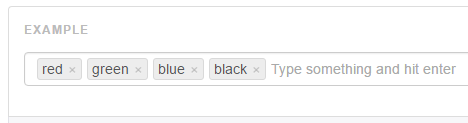
Организационные задачи:

* 1. Запустить и настроить Outlook. Всегда запускать его, когда находишься в офисе.
  2. Установить Skype и добавить в контакты e.romanko. Скайп так же держать открытым.
  3. Установить Visual Studio 2015 Community Edition
  4. Установить MSSQL Server (2014 или 2016)
  5. Установить любой Git-клиент

Задание: написать сайт-аналог amdm.ru, используя ASP.NET MVC + EF

1. Создать репозиторий на Github и дать ссылку Егору.
2. Установить VS2015 и MSSQL
3. Спроектировать структуру Solution, принимая во внимание, что возможно планируется так же создание Windows-приложения с тем же функционалом.
4. Создать иерархию классов
5. Сгенерить базу, используя EF CodeFirst подход.
6. Написать функционал по парсингу списка исполнителей из топа, используя HtmlAgilityPack
7. Подключить и использовать Twitter Bootstrap для верстки, клиент-серверные взаимодействия реализовывать посредством стандартных ASP.NET MVC приемов (View, PartialView, Хелперы Html.\* и Ajax.\*)
8. При невозможности реализовать какую-нибудь функциональность на клиенте посредством существующего функционала ASP.NET MVC – подключить jQuery и использовать его для этих целей.
9. На главную страницу сайта, вывести список всех исполнителей с именами, количеством песен и просмотров
10. Создать страницу, где будет отображаться информация об исполнителе: имя, биография, список песен (имя, количество просмотром и т.д.). Добавить ссылки на эту страницу в таблицу из пункта 9.
11. Создать страницу, где будет отображаться информация о песне: название, текст с аккордами, картинки используемых аккордов. Добавить ссылки на эту страницу в таблицу из пункта 10.
12. Добавить возможность редактировать песню: текст с аккордами и добавлять/удалять элементы в список используемых аккордов. Изменение списка аккордов реализовать таким образом:



Пример библиотеки: <http://sliptree.github.io/bootstrap-tokenfield/> (можно использовать другую).

1. Добавить серверную сортировку в таблицу артистов из пункта 9 (по всем столбцам).
2. Добавить серверную сортировку в таблицу песен из пункта 10 (по всем столбцам).
3. Добавить постраничный вывод в таблицу артистов из пункта 9
4. Добавить постраничный вывод в таблицу песен из пункта 10
5. На страницу из пункта 11 добавить кнопки «вперед» и «назад» (или > и < и оформить как слайдер галереи). Эти кнопки должны открывать следующую или предыдущую песню данного исполнителя (возможность «листать» песни).
6. Реализовать пункт 17 используя Ajax (без перезагрузки страницы)
7. Дополнить пункт 18 и изменять URL в браузере при переключении песни.
8. Добавить кэширование запросов, чтобы снизить нагрузку на базу. Например использовать InMemory Cache.
9. Добавить инвалидацию кэша, например, чтобы после редактирования песни тянулись новые данные.
10. На все страницы добавить Breadcrumbs для навигации.
11. Добавить в Solution консольное приложение.
12. Консольное приложение из пункта 23 должно получать список всех исполнителей и обновлять список их песен.
13. Используя библиотеку NLog, результаты обновления должны быть записаны в лог файл в виде:

Артист 1: 15 песен обновлены

Артист 2: 33 песни обновлены

1. Настроить запуск этого приложения каждые 10 минут, используя Windows Task Scheduler.
2. В MVC проект добавить SignalR.
3. Консольное приложение из пункта 23 должно уведомлять пользователя о том, что оно закончилось подсчет. Для этого, используя SignalR, сайт должен отображать Bootstrap Alert с временем завершения подсчета (время передается из консольного приложения):

