# Уеб календар – курсов проект по „Уеб приложения с ASP.NET MVC” – вариант #3

Да се проектира и реализира уеб базиран календар за срещи. Календарът трябва да поддържа списък с контакти и списък с въведени срещи. Календарът трябва да предупреждава потребителя за наближаващи срещи и да му дава информация за лицата, с които ще се срещне, и трябва да се предоставя възможност за търсене на срещи по определени критерии.

## Потребители

1. Системата допуска посетителите единствено до страницата за регистрация на нов потребител. Всички регистрирани потребители влизат в системата със своя username и парола.
2. Системата трябва да поддържа регистрация на нов потребител, достъпна за всички посетители. За всеки потребител в системата се съхранява следната информация:

* username (съдържа само букви и цифри, уникално за системата)
* парола
* e-mail
* име
* фамилия

1. Всеки потребител разполага с личен календар и със списък от контакти, които не могат да бъдат разглеждани и редактирани от други потребители.
2. Потребителят може да управлява:

* контактите си
* срещите си

## Контакти

1. За всеки контакт се съхранява следната информация:

* име (въвежда се задължително)
* фамилия
* e-mail
* дата на раждане
* телефон
* адрес
* забележка (допълнителна информация)

1. За контактите се поддържат следните операции:

* Добавяне на контакт – въвежда се и се съхранява информация за контакт.
* Редактиране на контакт – позволява промяна на информацията за даден контакт.
* Изтриване на контакт – изтрива наличната информация за даден контакт. Изтриването е позволено само ако контактът не участва в списъка от контактите за някоя среща.
* При добавяне на контакт с въведена дата на раждане автоматично да се добавя среща в специалната категория "Рождени дни" с дата предишния ден и час "14:00" и описание "Да поздравя <името на контакта>!".

## Срещи

1. За всяка среща се поддържат следните данни:

* дата и час на срещата (въвежда се задължително)
* описание (въвежда се задължително)
* място на срещата
* списък от контактите, които ще участват в срещата
* категория – избира се от списъка с категории за потребителя

1. За срещите се поддържат следните операции:

* Добавяне на среща – въвежда се и се съхранява информацията за нова среща.
* Редактиране – позволява промяна на информацията за дадена среща.
* Изтриване – изтрива дадена среща от календара.

## Категории

1. За всяка среща може да се задава категория, към която принадлежи (примерно категория „важни срещи”).
2. За категориите се поддържат следните операции:

* Добавяне на категория – дефинира нова категория.
* Редактиране на категория – позволява промяна на наименованието на дадена категория.
* Изтриване на категория – изтрива вече дефинирана категория. Изтриването е позволено само ако няма срещи, чиято категория съвпада с изтриваната.

## Търсене на срещи и предупреждение за наближаващи срещи

1. Трябва да се предостави възможност за търсене на срещи по съвкупност от следните критерии (произволна комбинация от тях):

* Начална дата – ако е зададена се търсят срещи, чиято дата на провеждане е след зададената начална дата.
* Крайна дата – ако е зададена се търсят срещи, чиято дата на провеждане е преди зададената крайна дата.
* Категория на срещата – извеждат се само срещи, попадащи в зададената категория.
* Име на контакт, участващ в срещата.
* Фамилия на контакт, участващ в срещата.

1. При търсенето не е задължително да бъдат въведени всички критерии. При търсенето по име или фамилия на контакт, не е задължително те да са изписани напълно, т.е. може да се изпише само част от името или фамилията. Например, ако при търсене не сме сигурни за името на контакта, можем да изпише само няколко букви. Ако напишем като име „**имир**”, ще се изведат срещи, в които участват примерно Бран**имир**, Стан**имир**, както и Крас**имир**а.
2. При влизане в системата, трябва да се извежда списък с информация за срещите, които ще се състоят в същия ден.

За всички таблици, извеждани при търсенето, да се имплементира възможност за сортиране (по всяка от колоните) и страниране (извеждане на страници от по 15 реда с възможност за навигация между страниците).

## Допълнителни изисквания

* Потребителският интерфейс на системата може да е на български или на английски език (по ваш избор).
* Използване на **.NET Framework 4.0 или 4.5** и езика **C#**.
* Използване на **ASP.NET MVC 4**.
* Съхранение на данните в релационна база данни **Microsoft SQL Server 2008 или 2012**.
* Използване на **ADO.NET Entity Framework** за достъп до базата данни.
* Използване на **Microsoft Visual Studio 2010 или 2012**.
* За управлението на потребителите, регистрация, login и logout е разрешено да се използва **ASP.NET Membership**, но не е задължително изискване.
* Потребителското име на всеки трябва да бъде уникална поредица от латински букви и цифри с дължина минимум 3 символа и да започва с буква.
* Паролата трябва да бъде с дължина поне 3 символа и е задължителна за всички регистрирани потребители.
* Email адресът е задължителен и трябва да се валидира минимум по следния формат: **<знак/знаци>@<знак/знаци>.<знак/знаци>**.

## Бонуси

Допълнителни точки ще се дават за:

* Качествен програмен код – според всички препоръки от курса High-Quality Programming Code (<http://codecourse.telerik.com>).
* Наличие на unit тестове.
* Реализирана добра ползваемост – удобен, лесен за употреба и интуитивен потребителски интерфейс.
* Коректност и правилна работа при необичайни ситуации: например при въвеждане на невалидни данни (празни полета, прекалено дълги полета, невалиден формат).
* Липса на пробиви в сигурността: SQL injection, cross-site scripting, unauthorized directory traversal и други.
* Добър и удобен външен вид (уеб дизайн) – можете да ползвате ваш собствен дизайн или да адаптирате дизайн със свободен лиценз от сайтове като: [http://www.oswd.org](http://www.oswd.org/) и <http://www.openwebdesign.org>.
* Реализирането на функционалност, която не се изисква в условията на проектите не носи непременно допълнителни точки. По-важно е да се реализира коректно изискваната функционалност.

## Предаване на проектите

Проектите трябва да съдържат следните активи, пакетирани в ZIP / RAR архив:

* База данни – релационна схема на базата данни в SQL Server с примерни тестови данни в таблиците. Базата данни да се предостави като SQL скрипт, който създава релационната схема и вмъква примерни данни в таблиците. Примерните данни трябва да са смислени, не "dsafdshga", "dsagfdsa" и "test1".
* Работещо ASP.NET MVC уеб приложение, реализиращо в пълнота и според изискванията цялата функционалност на настоящия проект.
* Пълен сорс код на проекта (C# класове, модели, изгледи, контролери, страници, картинки, скриптове и други ресурси).
* По желание проектът може да се качи в AppHarbor, Windows Azure, UhuruCloud или на друг обществен облак за .NET приложения с цел по-лесна демонстрация по време на защитата.

Проектите се предават онлайн от адрес: <http://telerikacademy.com/Courses/Courses/Details/15>.

Краен срок за предаване на проектите: **10 февруари 2013 г.**

## Защита на проектите

Защитата на проектите ще се извърши в края на курса. На защитата всеки студент ще трябва да представи проекта си, да покаже каква част от изискванията са реализирани и работят и да разкаже как е реализирал функционалността, след което ще му бъде дадена задача – да реализира допълнителна функция към проекта, на място в изпитната зала.

Силно препоръчително е всеки да си носи лаптоп с подготвена работна среда, на който да работи по време на защитата.

За да спестите време по време на защитата моля подгответе достатъчно примерни данни в базата данни на вашата система, с които да покажете всичките й функции в действие. Например, за да покажете страниране на таблиците, е необходимо да имате подготвени достатъчно данни в съответната таблица.

Датата за защита на проектите ще бъде обявена по-късно.

## Дискусионна група

За въпроси и коментари може да използвате дискусионната група на курса:

* <http://forums.academy.telerik.com/asp-net>