**Задача 1.**

Направете MVC приложение което ще има няколко екрана.

Единия екран ще покаже списък със студенти.

Друг екран ще покаже страница с данните за един студент.

Трети екран ще дава възможност да редактираме данните за 1 студент.

В детайли:

Направете MVC приложение в което да има следните фунционалности:

1) само един Controller с име HomeController

2) Index action и Index view в коeто да се покажат списък от студенти. Използвайте razor и html <table>.

За един студент показвайте следната информация - име и факултетен номер.

View-то да е strong type view. (т.е. направете class Student в папка Models и в началото на view-to дефинирайте модел от тип Student[] )

Елементите на масива ги задайте ръчно в action-a.

3) ViewStudent action метод и view със същото име което е strong typed view с модел от тип Student.

Показвайте само 2 реда с името и факултетния номер на студента.

Името и факултетния номер да са hard-coded стойности в controller-a.

action метода трябва да има входящ параметър - името на студента. И ако има подадена стойност от GET request-a, да се показва информация за точно това име, в противен случай - hard-coded стойността която сте написали.

4) EditStudent view което е strong typed view с модел от тип Student - форма с 2 текстови полета - име и факултетен номер, като факултетния номер е read-only поле.

Направете и 2 action метода за него - EditStudent() и EditStudentPost, като втория е само POST метод и приема стойността на името от формата.

Втория метод, трябва да покаже EditStudent view-то като изпише надпис "The info was updated for the user ...." и вместо точките, да пише новото име на студента.

5) Всички страници трябва да имат еднакъв header и footer.

Във footer-a да пише само текущата година, а в header-a да пише само общо заглавие на сайта с <h1> таг.

6) (доб. 18.03.2017) Да се създаде Login view, в което ще има форма за логване - полета username и password. При Submit на формата към сървъра няма да се прави никаква проверка, само ще се добавя в сесията username (Session["username"] = username;). И добавете в десния горен ъгъл на всички страници от сайта да се изписва името което е добавено в сесията (в \_Layout-а, добавете @Session["username"])

За справка може да гледате проекта BaseSample1.

**Задача 2.** (срок за предаване - 26.04.2017г)

Целта на задачата е да свикнете да създавате сами проекти в MVC приложение и да ги реферирате.

**Кратко описание:**

Направете MVC приложение, което ще има само една страница - Home\Index, в която ще се показва само едно число - броя на записите в таблица Users от базата.

Трябва да имате няколко проекта - един за MVC приложението, един проект за структурата на базата, един проект DataAccess за EntityFramework и един проект Repository за Repository функционалността

**Детайлно описание и стъпки:**

* Направете празно MVC приложение
* Направете база данни в която ще имате само една таблица - Users с няколко колони по ваш избор(не е важно какви ще са)
* Направете проект Database от тип “sql server project” и го синхронизирайте с базата (чрез Schema compare...)
* Направете проект от тип “class library” с име DataAccess в който ще добавите item от тип “ADO.NET Entity Data Model” и там извлечете таблицата Users от вашата база за да се генерира клас User.cs
* Направете проект от тип “class library” с име Repository, в който добавете един клас с име UserRepository и само един метод - GetAll() който ще връща List<User>, т.е. Списък с всички User-и от базата използвайки генерирания DbContext от предната точка
* В Home/Index action-a, извикайте метода GetAll() на UserRepository за да намерите броя на user-ите в базата.
* Така намерената бройка за потребители, трябва да я изведете на екрана Home/Index - или чрез ViewBag или създайте ViewModel клас

**Задача 3: (по желание) - още не сме стигнали до нея**

Направете промяна в проекта FormFieldValidation, така че, да не се допуска повече от една регистрация от един IP адрес.

Т.е. добавете server-side валидация, която проверява от кое IP е request-a, и ако вече това IP е използвано за регистрация - покажете съобщение за грешка.

За целта, най-лесно е да добавяте в сесията списък с IP-тата които вече са използвани за регистрация - List<string> и всеки път проверявайте дали IP-то е в списъка.

-------------------------------------------------------------------------------------------

**Това не го четете:**

**Задача 2.**

В проекта DatabaseRestaurants попълнете класа CategoryRepository по аналогия на класа CityRepository, като разпишете трите метода, които са празни в момента - GetByID(), GetAll() и Create(). Метода Create() го напишете като използвате stored procedure, както е метода DeleteByID() в класа RestaurantRepository. Тествайте новите методи в конзолата.

След това, обновете кода на проекта Restaurants.DB за да се добави новата stored procedure, която сте написали за метода Create(). Това става с десен бутон върху проекта Restaurants.DB -> “Schema Compare...”