**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)**

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.

**Задача 4: (по желание)**

Направете промяна в проекта EntityFrameworkSample\_Completed, така че, когато търсим ресторанти с търсачката и някой от намерените ресторанти няма файла с картинката, то да се показва картинка по подразбиране imageNotFound.jpg (която вие ще изберете от Интернет и сложите в проекта). Същата картинка да се показва и при Edit на ресторант, ако файла с картинката липсва от папката с картинки.

**Изисквания за курсов проект:**

**(допълнени на 26.05.2017. Допълненията са със *italic* шрифт)**

*Не е нужно да направите всичко, а само това което можете. Колкото повече от точките направите - по висока оценка ще имате. Задължителни са функционалността за регистрация, логване и показване/редактиране/триене на един от главните ви обекти в базата. Както и може да правите повече неща от това което се изисква.*

1. Направете MVC сайт, по ваш избор.
2. Визуалното оформление на сайта (html и css) няма да е важно при оценяване, а ще се гледа основно неговата функционалност (C# кода).
3. Хубаво е първо да направите структурата на базата данни, както и на лист хартия или на компютър да скицирате всеки екран от сайта ви какво точно ще съдържа (не е важно да е красиво нарисувано). Всеки от вас трябва да ги покаже в час. Това е с цел да си спестите евентуални проблеми в бъдеще.
4. Проекта трябва да съдържа следните функционалности:

* Регистрация на потребител
* Логване към системата
* Два вида потребители - администратор и обикновен потребител
* Всеки solution трябва да включва следните проекти - Database, DataAccess (с Entity Framework), Repositories и MVC проект.
* Базата данни трябва да съдържа поне 3 таблици
* Администратора трябва да може да вижда в табличен вид поне два от основните обекти в сайта ви (например ресторанти, хотели, продукти, категории и т.н.). Също така да може да ги редактира и изтрива.
* *Хубаво е ако можете да забраните на потребителите които не са админситратори достъп до страниците които са за администриране*
* Трябва да има валидация за всички полета където има нужда (например при регистрация, в админ частта при edit на обект, и т.н.)
* Трябва да има поне една custom валидация (client+server side), която да е нещо типично за вашия проект
* Клиента трябва да може да търси по един или няколко критерия в сайта и да се извежда резултата от това търсене. Не е нужно да изглежда както сме учили в час. Може да е както вие искате да изглежда.
* При избиране на обект(натискане на мишка) от резултата при търсене, да се отваря страница с повече информация за обекта
* При търсене, резултатите да могат да се странират на страници и сортират по избран критерий

*5. В главната папка на проекта сложете файл readme.txt или readme.docs в който опишете накратко каква е целта на сайта ви; какви функционалности има той. Целта на това описание е като ви проверяваме проектите да знаем какво точно сте направили от заданието или ако сте направили допълнителни неща - да не ги пропуснем.  
(Пример за описание на функционалност - “Има екран за показване на всички ресторанти, който е достъпен само от администратора. В него можем да създаваме/редактираме или изтриваме ресторант. ”.*

*6. Готовите задачи пращайте като архив на email-и* [*trtrifonov@sbnd.net*](mailto:trtrifonov@sbnd.net) *и* [*venelinmonev@gmail.com*](mailto:venelinmonev@gmail.com) *като първо изтрийте “bin”, “obj” и “packages” папките за да стане по малък архива*

Примерни сайтове:

* Сайта който правихте 1 семестър по фронт-енд
* Сайт за автобусното разписание на автогара Запад(сега няма сайт)
* Сайт за продажба на автомобили (mobile.bg)
* Техномаркет
* Технополис
* БДЖ
* Новинарски сайтове
* Сайт за продажба на лекарства