**Уеб програмиране с .Net**

**Полезни линкове:**

* Вход в Pluralsight: <https://www.pluralsight.com>

User: [si2017@abv.bg](mailto:si2017@abv.bg)

Password: si12345678

В секция Bookmarks сме добавили полезните видеа (засега е едно)

<https://app.pluralsight.com/library/bookmarks>

Но може да търсите и други.

**Задача за домашно:**

* <http://tinyurl.com/hlc9th6>

==================================================================

**I тема** (упражнения 1-3)

1. Клиент - сървър приложение.

* Видове клиент- сървър приложения според протокола:

- web server (apache, iis) – web browser

- ftp server(ftpd) – ftp client (filezilla)

- Email server (gmail) – email client (Outlook)

- Sql Server – Sql Server Management Studio

- BitTorrent Tracker – torrent client (uTorrent)

- DNS server (bind) – DNS client (resolver)

- DHCP server (wireless router firmware) - DHCP client(mobile phone/ Android DHCP client/ )

* Заявки и отговори

2. Протоколът HTTP. Обработка на заявки.

* Requests (методи GET, POST, PUT, DELETE, OPTION, TRACE, HEAD)
* Response
* Структура на една http заявка и на един отговор от сървъра (пример F12)
* Категории за кодовете на състоянието - Status code: <https://www.tutorialspoint.com/http/http_responses.htm>

<https://http.cat/>

* Разлики между HTTP1.1 и HTTP2.0

-> google: akamai http 2 demo

-> SPDY, QUIC

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

***Пример:*** разглеждане на http заявки

В браузъра с F12 отваряме Developer Tools, оттам секцията Network.

Разглеждаме отваряне на страница (GET).

Логване или попълване на form-а (POST).

Избираме една от заявките и преглеждаме Request headers и Response headers.

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

3. Многослойни архитектури (кратко резюме)

* Трислойна архитектура (The 3-tier Architecture)

4. История на Web преди MVC

* ASP, Web Forms
* Малко за ASP.NET

5. Обяснение на MVC архитектурата

* Картинка
* Или статия

6. История на ASP.NET MVC

* Основни характеристики на : MVC 1, MVC 2, **MVC3**, MVC4, MVC5

7. Създаване на нов проект

* Обяснение за възможните проекти, които могат да се създават ( MVC, WebApi, Empty, Web Forms, Single Page Application)
* Обяснение за Change Authentication
* Стартиране и какво е IIS Express и принудително спиране
* Стартиране без да има създаден Controller
* Стартиране без да има създадено View
* Проекта да има само едно View с текст
* Закоментиране на Layout в \_ViewStart.cshtml и обяснение
* пример като не се намира контролер
* пример като не се намира action
* пример "ShowString"
* пример "ShowCurrentDate"
* пример "GetUrlParameter" (пример като не се намира parameter)
* пример "ShowSubmitForm" - "SaveSubmitForm(string username)"
* пример "ViewPhonebookStrongType"
* Пример за EditUser и [POST]EditUser
* създаване на index
* създаване на partial view - “\_TopMenu.cshtml” (покажи диаграма 7.)
* Обяснение на @Html и @Html.ActionLink, @Html.TextBoxFor, @Html.TextBox
* Обяснение на \_Layout.cshtml
* Обяснение на \_ViewStart.cshtml
* създаване на @section - Footer
* Добавяне на css и javascript files в \_Layout.cshtml
* Изчистване на кеша на браузъра:

*F5 or Control + R = Reload the current page*

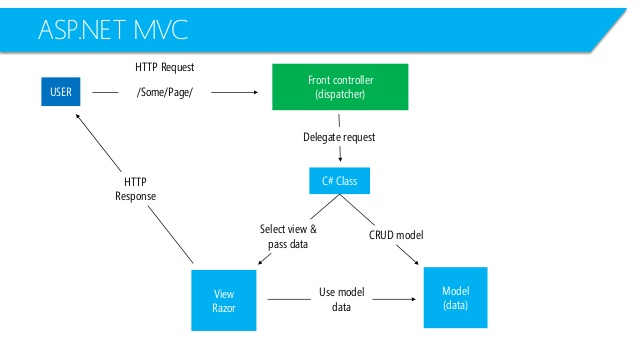
*Control + Shift + R or Control + F5 or Shift + F5 = Reload your current page, ignoring cached content*

8. Затваряме проекта и отваряме примера “А”.

9. За упражнение: създаване на MVC проект и html форма с 2-3 полета в нея

Полезни линкове:

1. <http://ucha.se/watch/859/Vavedenie-v-web-tehnologiite-Uebsaytove-i-Uebprilozheniya>
2. <https://www.youtube.com/watch?v=0fk7f1yyhL8>
3. MVC diagram <http://www.dotnettricks.com/learn/mvc/understanding-model-view-controller-in-aspnet-mvc>
4. <http://image.prntscr.com/image/0d7798417ad04c66852c83e25bec8819.png>



1. Razor basics - <https://weblogs.asp.net/scottgu/introducing-razor>
2. Razor syntax (не гледайте трета колона) <http://haacked.com/archive/2011/01/06/razor-syntax-quick-reference.aspx/>
3. Layout - <http://discusscode.blogspot.bg/2011/12/aspnet-mvc-3-razor-section-defined-but.html>

============================================================

**II тема** (упражнение 4)

1. Обяснения на проекта BaseSample2

Проекта показва списък от потребители в Index и можем да добавяме нов потребител, редактираме потребител и трием потребител. Данните ги запазваме в сесията.

* Създаваме празен проект
* Създаване на ViewModel class User и обяснение на List<User>
* Обяснение на Session

*SessionID= @HttpContext.Current.Session.SessionID*

* Създаване на SessionObjects class (singleton pattern за сесия) в нова директория Helpers.
* Създаване Index, ViewUser и EditUser
* Показване че не работи ако няма hidden поле при EditUser
* Задаваме Home страница да е \User\Index а не Home\Index(routing)
* Добавяме \_TopMenu

2.Полезни линкове

Session - <http://stackoverflow.com/questions/6665619/is-session-stored-in-client-side-or-server-side>

Session - <http://rajeeshcv.com/2008/06/12/How-Web-Servers-Identify-Your-Session/>

==================================================================

**III тема**

Пример за извличане на данни и запис на данни с ADO.NET и Repository pattern.

Пример за SQL Server Database project.

Базата данни е за ресторанти - 3 таблици: Restaurants, Categories и Cities.

* Създаваме база данни Restaurants и 3 таблици в нея.

Или наготово направени или с sql заявки за създаване или създаване с дизайнера.

* Създаване на празен SQL Server Database project
* Извличане на структурата на базата със Schema compare
* Създаване на Post deployment script за инициализация на базата с данни
* Обяснение за Project properties -> Target Platform за настройка на версията на SQL server-a който ще използваме
* Пример с “Publish...” в нова база данни
* Пример с добавяне на колона в таблица и Schema compare
* Създаваме проект DataAccess и 2 папки - Entities и Repositories
* Създаване на 3 class-a в Entities - Restaurant, Category и City със структура подобна на таблиците в базата
* Обяснение на идеята да се ползва Repository - еднотипна CRUD фунционалност
* Осъзнаване на необходимостта за базов клас (всяко repository има ConnectionString property)- BaseRepository
* Създаване на ConnectionString с Views -> Sql server explorer
* Създаваме console application за демонстрация на долните методи:
* Създаване на CityRepository и 2 метода - GetByID и GetAll
* Извикване на тия 2 метода в Console Application и отпечатването на данните
* SqlConnection, SqlCommand, SqlParameter, SqlDataReader
* CommandType видове - Text, StoredProcedure, TableDirect
* Създаване на RestaurantRepository - GetByID, GetAll, GetByCityID, Insert

За домашно: Направете DeleteByID(int restaurantID) метод в RestaurantRepository

И по желание - който иска да се опита да вместо Console application да създаде MVC приложение което да използва DataAccess и Repository проектите

2.Полезни линкове:

Важно е тази отметка да е махната в SQL Management Studio:

