## Изпитна тема № 16: Интернет програмиране

ASP.NET MVC е платформа, създадена от Microsoft, която служи за изработване на уеб приложения с шаблон за дизайн Модел-Изглед-Контролер (на английски: Model-View-Controller, MVC). Платформата използва C#, HTML, CSS, JavaScript и бази данни и е съвременно средство за уеб приложения, което обаче не замества изцяло уеб формите. Платформата включва нови тенденции в разработката на уеб приложения, притежава много добър контрол върху HTML и дава възможност за създаване на всякакви приложения. ASP.NET MVC може да бъде много лесно тествана и допълвана, защото е изградена от отделни модули, изцяло независими един от друг. Чрез платформата се създават цялостни приложения, които се стартират, а не единични скриптове (като при PHP например). Уеб сървърът по замисъл е направен да доставя уеб страници на уеб браузър. Цялата комуникация е възможна чрез обмен на „HTTP requests“ („HTTP заявки“), отправени от браузъра и „HTTP responses“ („HTTP отговори“) от страна на сървъра, което осигурява комуникацията съгласно приет стандарт. HTML формите позволяват на потребителя да попълва данни и да ги изпраща до сървъра. CSS определя стила на HTML елементите: шрифтове, цветове, полета, размери, позициониране и т.н. CSS се декларира в следния формат: свойство : стойност. Вграденият CSS дефинира правила за форматиране на определен HTML елемент.

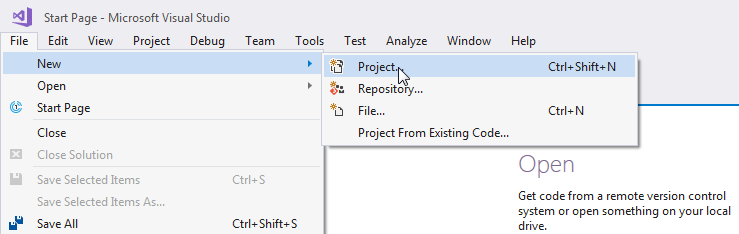
DataContext е основният клас, чрез който се осъществява връзка към база данни, извличане на обекти от нея и изпращате на промени обратно към нея. Използването на DataContext се базира на ADO.NET SqlConnection. Всъщност DataContext се инициализира с връзка или низ за връзка, който доставяте като константа вътре в класа. Целта на DataContext е да преведе заявките за обекти в SQL заявки, които трябва да бъдат направени спрямо базата данни, и след това да сглоби обекти от резултатите. DataContext дава възможност за интегрирана в езика заявка (LINQ) чрез прилагане на същия модел на оператора като операторите на стандартни заявки, като Where and Select.

Понеже обекта на нашето изследване са не Web Application сървърите сами по себе си, а как те се използват за създаване на динамични страници и свързването на тези страници с бази данни, ще разгледаме и основните програмни езици за създаването на Web сайтове. След създаването на SAPI, Microsoft и други независими разработчици произведоха приложения, които се свързват към Web сървъра чрез него. Това е начинът- чрез разработката на Microsoft - ISAPI, Active Server Pages, ASP да се свързват към IIS, както и на други техники на Microsoft за динамично свързване от сървърната страна. Преди ASP най-често използваното средство беше Internet Database Connector (IDC). Вследствие успеха на ASP, Microsoft разработи Windows Scripting Host (WSH) - технологията, която дава възможност за изпълнение на скриптове в Windows 95, 98 и NT 4. WSH е независим от езика и поддържа JScript. Както и при ASP, аналогичен е случая при Cold Fusion страниците на Allaire, които се свързват към Web сървъра си посредством стандартния сървърен API, но освен това поддържат и CGI интерфейса. Продължителят на ASP от страна на Microsoft се появи съвсем наскоро и беше наречен ASP+ или ASP.Net. Ако сравним ASP и ASP+, ще забележим че има два основни разлики във вторите сървърни страници: ASP+ поддържа кодово написани и компилируеми езици като например Visual Basic, C++, Perl, и поддържа сървърни контроли, които могат да разделят кода от съдържанието, позволявайки по този начин добавяне в страниците. И въпреки че ASP+ не е обратно съвместима с ASP технологията, е възможно да се развива заедно с нея. ASP е съкращение, означаващо Active Server Pages. Това е една отворена среда за разработване на динамични и мощни Web-базирани приложения, в които се използват HTML, scripts и ActiveX компоненти. Може би сте запознати с някой скриптов език (VBScript или JScript). За разлика от тях ASP е сървърско скриптово обкръжение (server-side scripting environment). Нека поясним това по-подробно. При обикновените скриптове програмата е част от страницата. Тя пътува от сървъра през Интернет до браузъра, който я изпълнява (интерпретира). По този начин потребителя получава изходния код на програмата. При ASP нещата стоят по друг начин. Програмата е пак част от страницата, но се изпълнява на сървъра и до потребителя достига само крайният резултат. Това дава някои предимства, по-важните от които са:

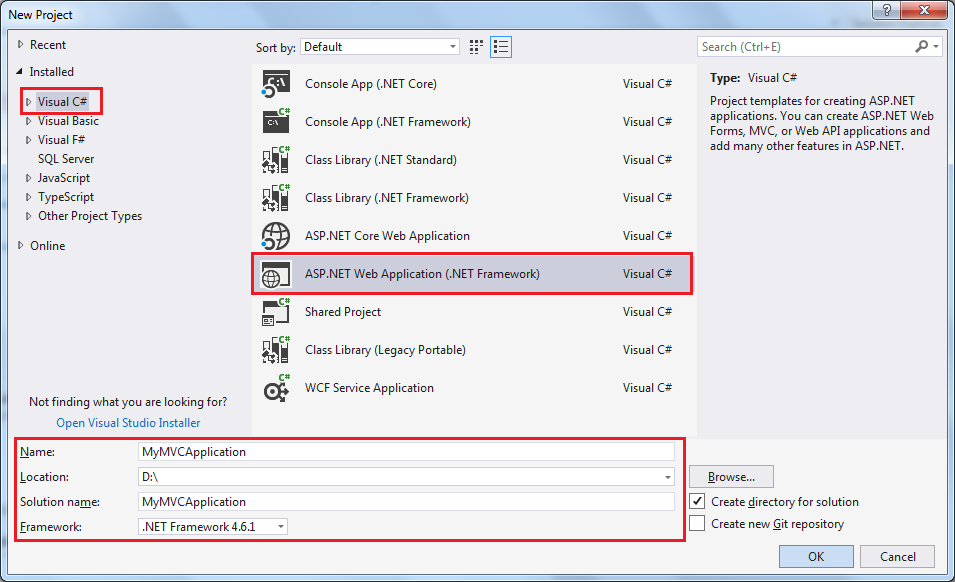
1. Крайният потребител не вижда изходния код на програмата,
2. Скриптът може да използва всички (на практика) ресурси на сървъра.

Основното предимство на ASP обаче е неговата разширяемост. Написаната вече програма може да комуникира с определен клас ActiveX контроли, предварително инсталирани на сървъра. Такива ActiveX контроли могат лесно да бъдат създадени например чрез Delphi 5, като по този начин можете на практика да правите всичко. На всеки сървър има инсталирани няколко стандартни ActiveX контроли.

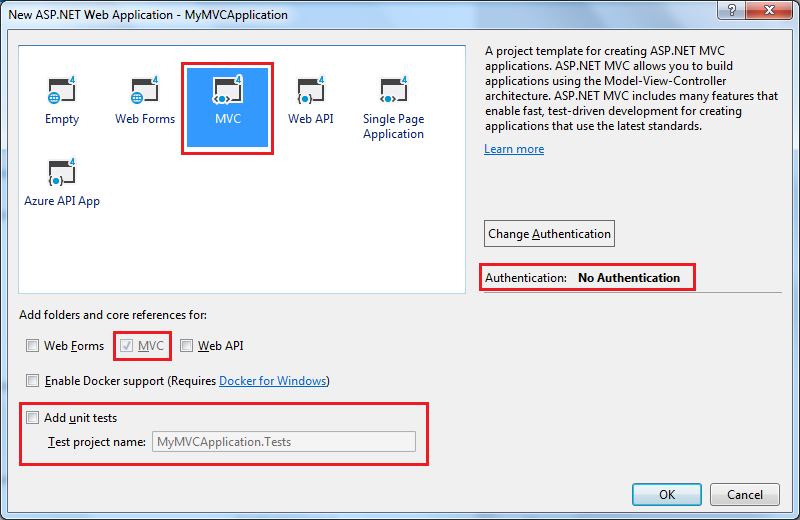
В примера тук ще използваме ASP.NET MVC v5.2 и Community изданието на Visual Studio 2017 и .NET Framework 4.6, за да създадем първото си приложение MVC. Изтеглете последната версия на Visual Studio от visualstudio.microsoft.com/downloads . Отворете Visual Studio 2017 и изберете меню File -> New -> Project, както е показано по-долу:

[](https://www.tutorialsteacher.com/Content/images/mvc/create-mvc-project.png)

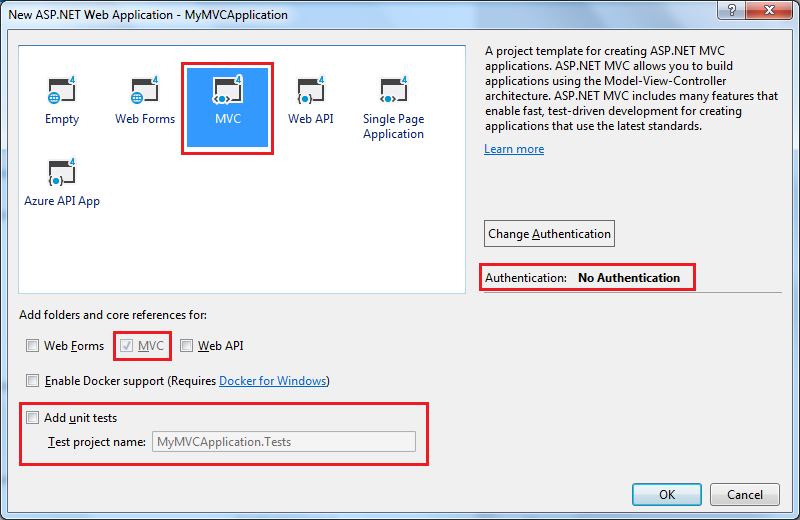
От диалоговия прозорец New Project, както е показано по-долу, изберете Visual C # и изберете Web в левия прозорец и след това изберете ASP.NET Web Application (.NET Framework) в средния прозорец. Въведете желаното име на вашия проект. Също така можете да промените местоположението на приложението MVC, като кликнете върху бутона Browse... Накрая щракнете върху OK.

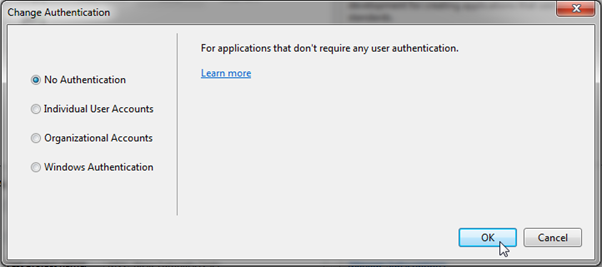
[](https://www.tutorialsteacher.com/Content/images/mvc/MVC-template.png)

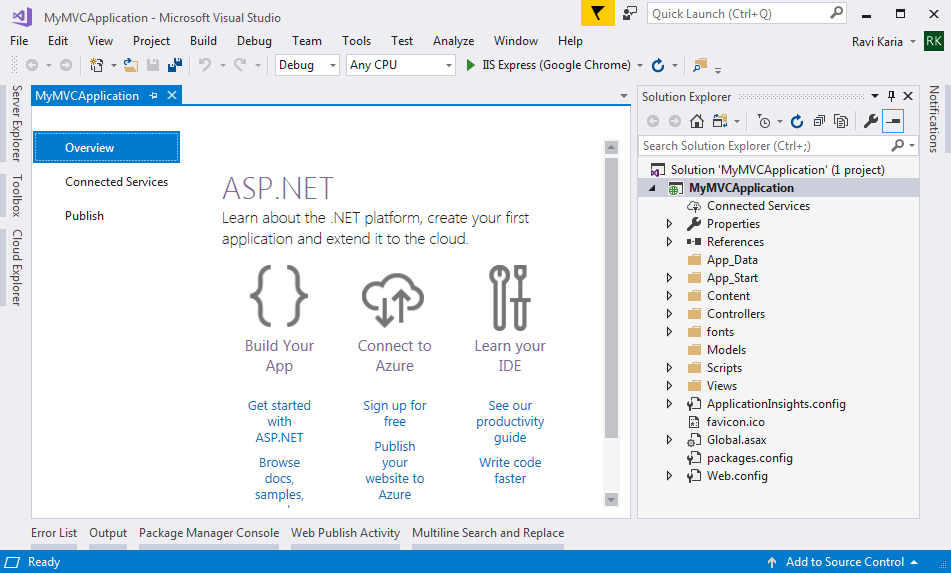
От диалоговия прозорец New ASP.NET Web Application изберете MVC (ако вече не е избран), както е показано по-долу.

[](https://www.tutorialsteacher.com/Content/images/mvc/Create-MVC-Project2.png)

Изберете MVC Project Template.

[](https://www.tutorialsteacher.com/Content/images/mvc/Create-MVC-Project2.png) Тук пазим автентификацията по подразбиране за нашето приложение, която е No Authentication, но по желание може да се промени.

[](https://www.tutorialsteacher.com/Content/images/mvc/first-mvc-app4.png)Кликнете върху OK, за да продължите. Изчакайте известно време, докато Visual Studio създаде прост MVC проект, използвайки шаблона по подразбиране, както е показано по-долу.

[](https://www.tutorialsteacher.com/Content/images/mvc/MVC-application.png)

Понятията автентикация и оторизация често се използват като синоними, но всъщност те има различен смисъл. Трябва да се прави разграничение между автентификацията, която представлява установяване на самоличността на даден потребител, и оторизацията – тя дефинира действията, които може да извършва даден потребител.

Сигурността на информацията е най-важното нещо в Интернет, но тя не може да се осигури, ако има пропуски във фундаменталните принципи. За да се обезпечи сигурността на информацията е необходимо да се избегне неоторизирания достъп до електронните данни, в критичните от гледна точка на бизнеса системи в дадена компания. Резултатът от неоторизиран достъп може да бъде промяна, заместване, разпространение или нарушаване на дадена информация. Конфиденциалността е необходима за да се контролира получателя на дадена информация и за да се избегне нейното разпространение. Целостта трябва да гарантира, че информацията и програмите се променят само по определен и оторизиран начин, че данните са действителни и не са променени или изтрити по време на предаване. Интернет предлага голямо разнообразие от възможни атаки. Въпреки че по-голямата част от тях са малко вероятни трябва да прецените колко опасни могат да бъдат те за вашата компания. Заплахите могат да се разделят в четири основни категории: загуба целостта на данните, загуба на конфиденциалност на данните, загуба на услуги и загуба на контрол. Има много начини, по които може да се атакува една система. Един от тези начини е например, да се наблюдава комуникацията между двама партньори. По подразбиране, комуникацията през Интернет е абсолютно незащитена и информацията се предава като прозрачен текст. Чрез наблюдение на комуникацията може да се извлече конфиденциална информация и пароли. Ако например, осъществите достъп до вашия mail сървър, името на потребителя и паролата се изпращат до сървъра и всеки в Интернет може да ги прихване. Ако даден хакер успее да прихване това, той може да промени информацията преди тя да е стигнала до предназначеното местоположение. В случай, че получателя няма специален софтуер, той няма да забележи промените направени от трето лице. Една друга възможност за неприятности е софтуера или хардуера да бъдат откраднати. Софтуер, като например бази данни може да съдържа конфиденциални данни и пароли, а хардуера може да даде възможност на хакера да разбере как е изградена вътрешната мрежа. Освен това, при достъп на хакери до хардуера те могат да разберат фабрично кодираната в отделните хардуерни модули информация, като например код на смарт карта, посредством който може да се разбере по какъв начин се използва тази карта. Освен прихващане на мрежови комуникации, могат да се прихващат и електромагнитни изходни сигнали от устройства, като например монитори. Съществуват устройства, които могат да копират съдържание, показано на екрана на един монитор върху друг монитор, който се намира на разстояние стотици метри. "Троянските коне" са друг метод за нелегитимно проникване в дадена система. Обикновено „троянският кон" е скрит в безобидно изглеждащ софтуер, който го активира при стартирането си. „Троянският кон" работи на заден план и събира информация за системата и нейните потребители. След това, тази информация се изпраща на хакера, който може да влезе в системата и да я контролира отдалечено. Друг известен тип атака е „маскирането" (известно още като „лъжливо представяне" ). Представяйки се за друг потребител, хакерът може да влезе в системата. Повечето атаки се изпълняват с лъжлив IP адрес. Много системи позволяват достъп до ресурсите само на ограничен набор IP адреси. Хакерът показва IP адрес точно от този набор, при което получава автоматичен достъп до определени ресурси.