Script ASP.NET MVC les

**Aansturing**

**1. Inleiding:**

- Studenten welkom heten  
 - Uitleggen waar we het over gaan hebben en hoe de les gaat verlopen

**2. Introductie/uitleg ASP.NET**

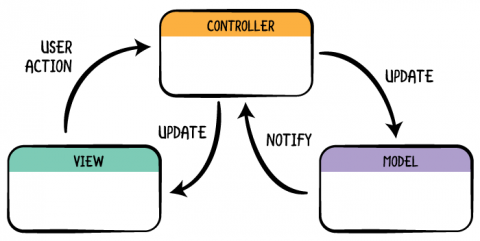
ASP.NET is een server side web applicatie framework voor web development. ASP staat voor Active Server Pages en hiermee kun je dynamische pagina’s produceren. Het is gemaakt door Microsoft om programmeurs de mogelijkheid te geven om dynamische websites, web applicaties en web services te maken. Het was gereleased in Januari 2002 met versie 1.0 van het .NET Framework en is de opvolger van Mircosoft’s Active Server Pages(ASP) technologie. ASP.NET is gebouwd op het Common Language Runtime (CLR), hierdoor kunnen programmeurs ASP.NET code gebruiken die ondersteund wordt door de .NET language.

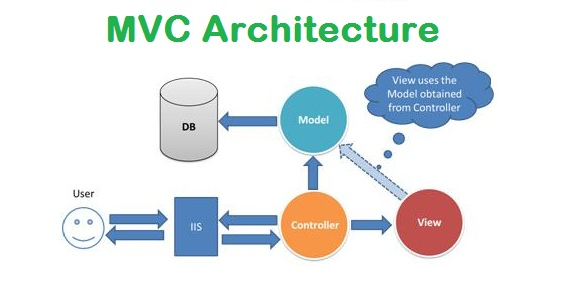
**3. Uitleg MVC**

MVC staat voor Model View Controller. Het is een ontwerppatroon(designpattern) waarmee complexe applicaties leesbaarder worden en dat het veranderen van de user interface (View) niet direct invloed heeft op de code die het aanstuurt en de data die ervoor wordt gebruikt. Ditzelfde geld voor de Controller en het Model. Elk van deze componenten zou kunnen worden vervangen door een ander zolang alle naamgevingen en benodigde methodes maar aanwezig zijn. De functies van het Model de View en de Controller worden later in de presentatie behandeld met voorbeelden in ASP.NET MVC.

De verbindingen zijn:  
- Controller -> View  
- View -> Controller  
- Controller -> Model  
- Model -> Controller

Getoond in de afbeelding hieronder:





**4. Uitleg verschil tussen ASP.NET en ASP.NET MVC**

Het grootste verschil tussen ASP.NET en ASP.NET MVC is dat bij ASP.NET achter elke pagina een code behind file aanwezig is. Bij ASP.NET MVC zijn er controllers die zorgen voor requests van meerdere pagina’s. Dit betekend dat bij ASP.NET die view strikt verbonden is met de code behind file waaraan de view gekoppeld is. Bij ASP.NET MVC is alles apart van elkaar. Dit betekend dat je gemakkelijk views, controllers en models uit kan wisselen tussen elkaar zolang alle naamgeving en methodes maar aanwezig zijn.

**Zelfstanding trainen**

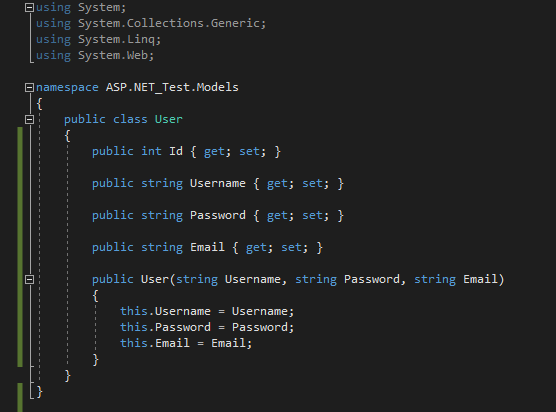
**Makkelijke oefeningen:**

**5. Model**

Definieert de representatie van de informatie waarmee de applicatie werkt. Aan ruwe gegevens wordt betekenis gegeven door relaties tussen de data en logica toe te voegen. De daadwerkelijke data wordt opgeslagen in een opslagmedium zoals een database. De applicatie zal de gegevens die gebruikt worden in het model ophalen en wegschrijven van en naar de dataopslag.

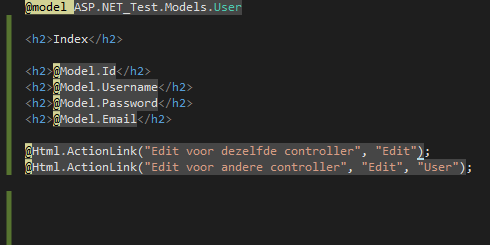
Een model kun je dus eigenlijk een beetje vergelijken met een tabel in een database. De properties die je er aan toe wijst zijn de kolommen de data die je hierin zet kun je dan opslaan in de database. In ASP.NET wordt de model geschreven in C# of VB.

Voorbeeld in ASP.NET met een model(User):



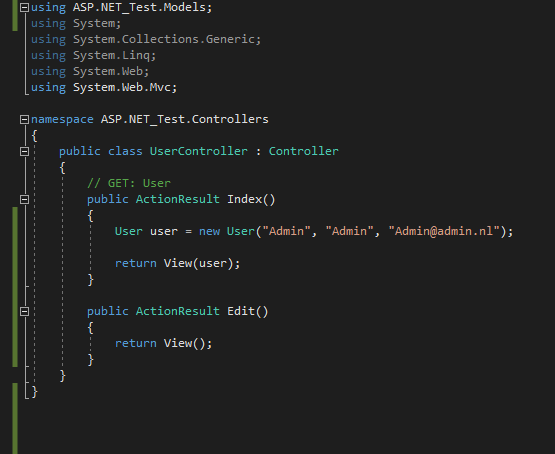
**6. View**

De view is de user interface van de web applicatie. Alle data die de gebruiker zal zien wordt in de view weergeven. De view doet geen verwerkingingen zoals berekeningen en controles van de gegevens die getoond worden. Een view wordt in ASP.NET gemaakt door middel van HTML, CSS en Javascript. In de view kan door middel van één of meerdere models en de Razor engine in ASP.NET data aan de gebruiker worden getoond die bijvoorbeeld uit de database komt. Wij komen later in de les terug op de Razor engine en hoe deze werkt. Je kan in ASP.NET MVC ook partial views inladen in je main views. Later komen wij tijdens de les terug op wat deze partial views inhouden en hoe je ze kunt gebruiken.

Voorbeeld in ASP.NET met een view(Index, User):  


**7. Controller**

De controller verwerkt en reageert op events. Deze zijn meestal het gevolg van handelingen die de gebruiker uitvoert, zoals bijvoorbeeld een klik op een knop in de view. Aan een controller kunnen één of meerdere view worden gekoppeld. Een knop in de view kan gekoppeld worden aan verschillende methodes in de view liggend aan wat je wilt doen. Een controller kan dan verschillende actionresults teruggeven waardoor bijvoorbeeld kan worden genavigeerd naar een andere view die gekoppeld is aan dezelfde controller of een andere controller. Met deze actionresults kunnen models worden meegestuurd naar de view zodat de view de data hierin kan weergeven.

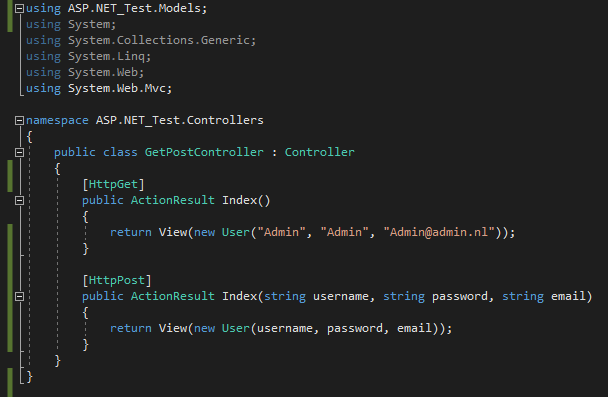
Voorbeeld in ASP.NET met een controller(User):  


**8. Get en Post**

Twee soorten HTTP requests zijn Get en Post. Get is voor als je data wilt opvragen en Post is voor als je data wilt versturen. Je kan in een controller afdwingen dat een bepaalde methode alleen get requests of alleen post requests verwerkt.

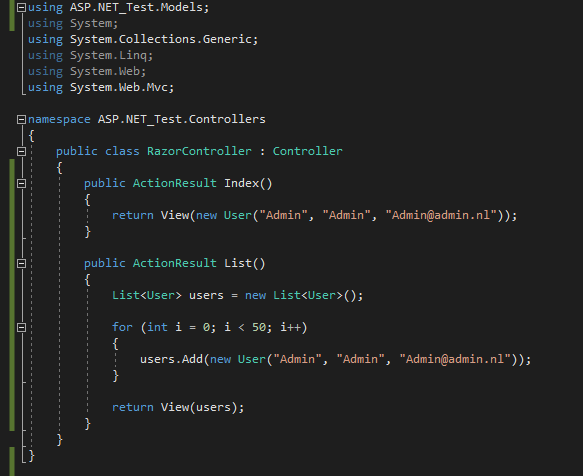
Voorbeeld in ASP.NET met Get en Post:

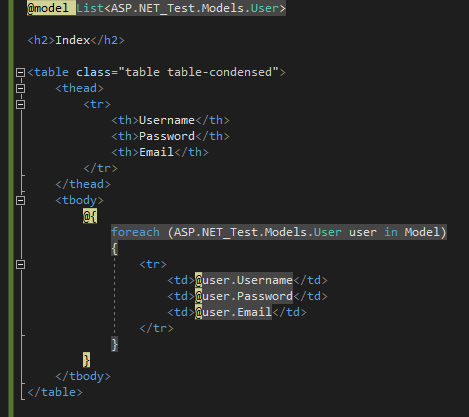
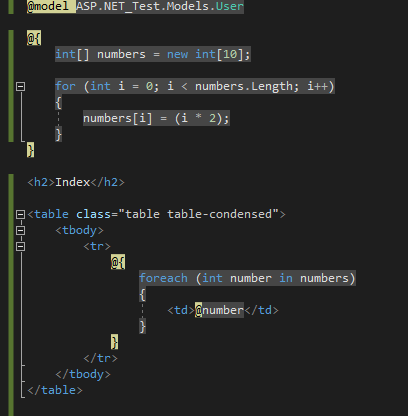




**9. Razor**

De Razor engine zorgt er als het ware voor dat je C# of Visual Basic kan gebruiken in je View om daar bijvoorbeeld een Model aan te roepen en dus bijvoorbeeld een tabel te vormen waar de data van het Model in moet komen te staan. Je kan met Razor ook variabelen aanmaken in de view zolang de @{} tags eromheen staan. Al kun je functies met Razor gemakkelijk in de view uitvoeren is het over het algemeen niet aangeraden om complexere code in de view uit te voeren, deze complexere logica kun je beter aan de controller overlaten. Als je dit wel zou doen ga je eigenlijk tegen het concept van MVC in.

Voorbeeld in ASP.NET met Razor:  




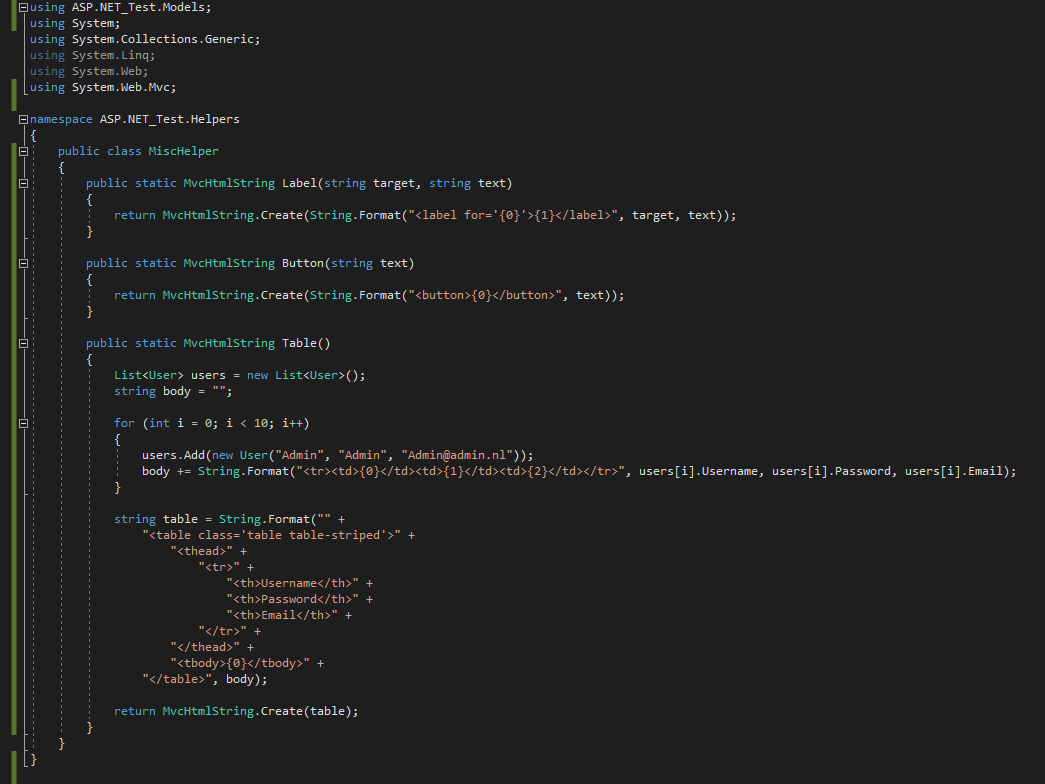
**10. Html helpers**

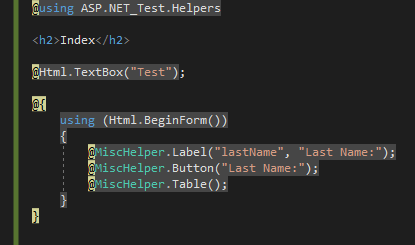
Html helpers helpen je bij het maken van de view door het hoeveelheid typen te verminderen. Een Html helper geeft een string terug waarin je Html kan verwerken. Hierdoor zou je een knop, label, tabel en allerlei andere dingen kunnen maken. Er zitten in ASP.NET MVC standaard een aantal Html helpers die je kan gebruiken. Maar je kan ook je eigen custom Html Helper schrijven.

Lijst van een aantal Html helpers die standaard in ASP.NET MVC zitten:

Html.ActionLink() Html.BeginForm() Html.CheckBox() Html.DropDownList()  
Html.EndForm() Html.Hidden() Html.ListBox() Html.Password()  
Html.RadioButton() Html.TextArea() Html.TextBox()

Voorbeeld in ASP.NET met Html Helper:



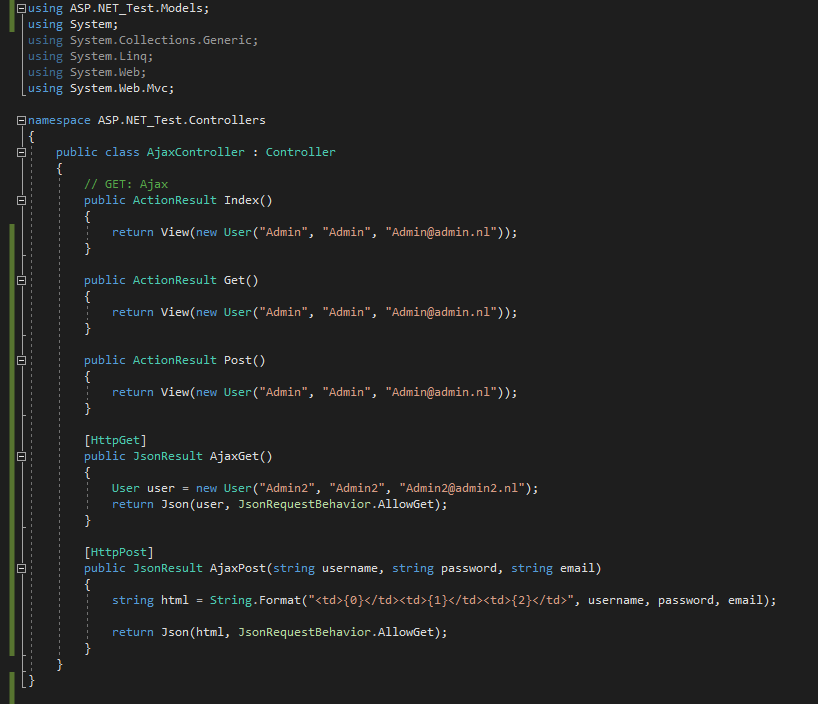


**Moeilijkere oefeningen:**

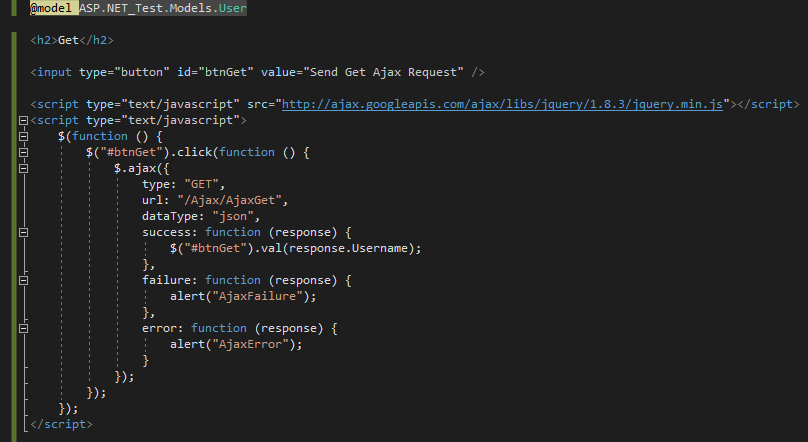
**11. jQuery Ajax(Get en Post)**

Met jQuery Ajax kun je calls maken via Javascript om mogelijk data uit de database te halen of een andere bepaalde functie uit te voeren zonder dat je de pagina hoeft te verversen. Hierdoor kun je bijvoorbeeld live feeds maken van bijvoorbeeld aandelen of wat dan ook. Net zoals een Http request heeft een Ajax call ook verschillende types. Twee daarvan zijn Get en Post. Deze zijn te vergelijk met die van de Http request die we een stukje terug in de presentatie hebben behandeld. Voor een Ajax call kun je meerdere soorten datatypes ontvangen zoals tekst, html, json enz.

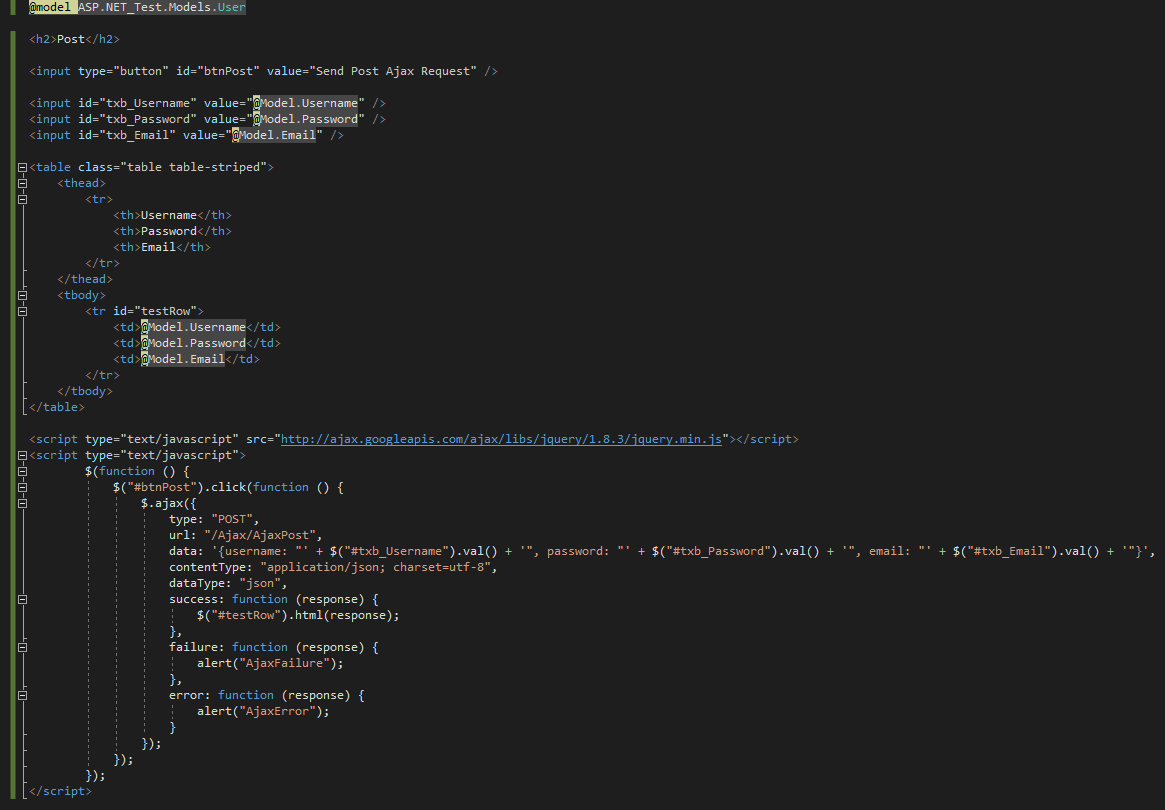
Voorbeeld in ASP.NET met jQeury Ajax(Get en Post):



Get:



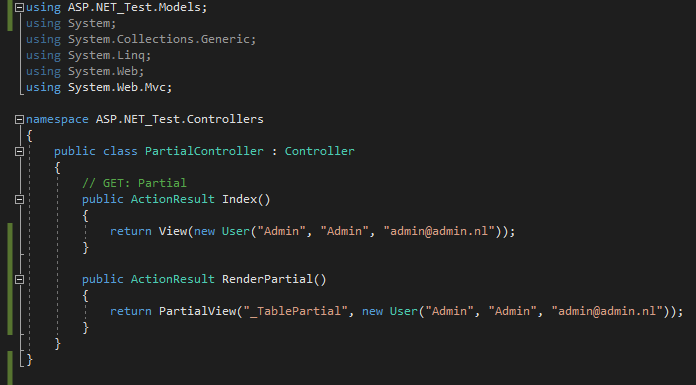
Post:



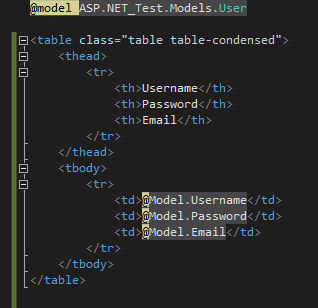
**12. Partial loading met en zonder Ajax**

Een partial is in principe een stuk Html met mogelijk in ASP.NET ook Razor die je kan laden in een andere pagina. Stel je hebt een Index view en je wilt er een tabel in laden. Hiervoor zou je een aparte partial view kunnen maken waarin je de tabel vormgeeft. Later kun je dan in de index view dit partial view inladen. Door gebruik te maken van partial views worden je views overzichtelijker naar mate je meer functionaliteit aan een bepaalde pagina toevoegd. Zou je dit zonder partial views uitwerken dan wordt het stuk html code al snel veel en dus onoverzichtelijker. Je kan een partial view inladen door mate van een html helper of jQuery Ajax.

Voorbeeld in ASP.NET met Partials(met en zonder Ajax):



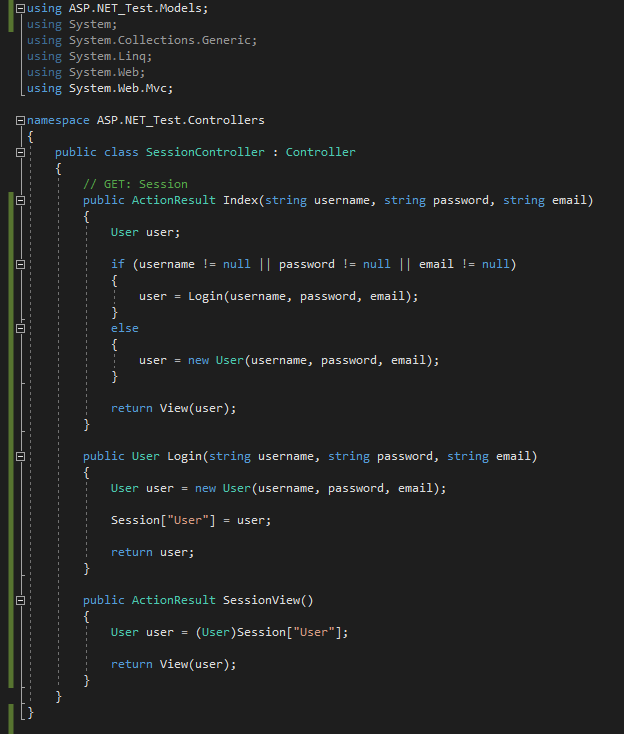


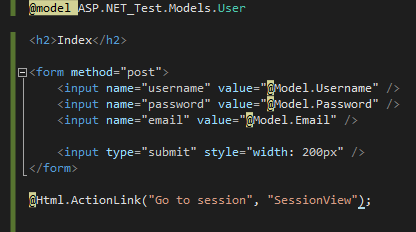


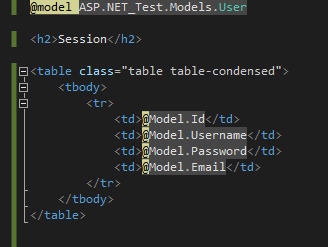
**13. Sessions**

ASP.NET session state laat je variablen opslaan en opvragen voor een gebruiker terwijl hij of zijn de ASP.NET applicatie navigeert. HTTP is een staatloos protocol. Dit betekent dat de Web server elk HTTP request behandeld als een onafhankelijk request. De server heeft hierbij geen weten van waardes van variabelen die zijn gebruikt in een vorig request. ASP.NET session state indentificeerd requests van dezelfde brower in een gelimiteerde tijd als een session, en biedt een mogelijkheid om waardes van variablen te gebruiken in de tijd dat de session actief is. ASP.NET session state in geactiveerd voor elke ASP.NET applicatie.

In een session kun je alle soorten .net-framework datatypes gebruiker. Deze variabelen worden opgeslagen in een SessionStateItemCollection en kan aangeroepen worden door de HttpContext.Session propertie.

Voorbeeld in ASP.NET met Sessions:  






**Terugkoppeling**

Feedback geven over hoe wij de les vonden gaan en vragen aan de studenten hoe zij de les ervaarden.

**Bronnen:**

<https://en.wikipedia.org/wiki/ASP.NET>   
<https://nl.wikipedia.org/wiki/Model-view-controller-model>   
<https://www.codeproject.com/Articles/668182/Difference-betweeen-ASP-NET-WebForms-and-ASP-NET-M>   
<https://www.quora.com/What-is-the-difference-between-ASP-NET-MVC-and-ASP-NET>   
<https://docs.microsoft.com/en-us/aspnet/web-pages/overview/getting-started/introducing-razor-syntax-c>   
<https://docs.microsoft.com/en-us/aspnet/mvc/overview/older-versions-1/views/creating-custom-html-helpers-cs>   
<https://www.aspsnippets.com/Articles/ASPNet-MVC-jQuery-AJAX-and-JSON-Example.aspx>   
<https://docs.microsoft.com/en-us/aspnet/core/mvc/views/partial>   
<https://msdn.microsoft.com/en-us/library/ms178581.aspx>