Lesson 1

* ASP.Net .Net-in bir framework-udur. Veb layihələri, veb servisləri yaza bilərik. ASP açılışı Active Server Page-dir. Microsoft tərəfindən yaradılıb.
* .Net framework ancaq Windows əməliyyat sistemlərində istifadə olunur.
* .Net Core is Cross-platformdur.
* Veb API lar yazmaq mümkündür.
* Connected Services - hər hansı cloud qoşduqda orada göstərir.
* Dependencies – Packagelərimiz dependencylərimiz orada yerləşir.
* launchSettings.json – proyektin ilk başlaması ilə bağlı məlumatlar orada yerləşir. Hansı port ilə başlayacaq və s.
* Microsoftun işlətdiyi İnternet İnformation Service serverini işlədir (IIS Express)
* Vvv.root papkası statik faylları saxlayır. Proyekt daxilində kontenti dəyişilməyən elementlər orada yerləşir. Iconlar css fayllar static fayllar js və s.
* MVC pattern üzərindən yaradılıb. Model Vievv Controller
* Cshtml Razor Vievv-dur. Yəni HTML daxiliində CSharp kodu yazmaq mümkündür.
* @RenderBody() dəyişən componentləri orada yerləşdirir.
* Appsettings.json application ilə bağlı configuration-ları orada saxlamaq mümkündür.
* Appsettings.json configuration fayllarını hər hansı kodda dəyişiklik etmədən sadəcə fayl üzərindən dəyişmək mümkündür.
* Program.cs və Startup.cs .Net 6 dan sonra bir faylda yerləşir.
* Hər hansı servisləri qeydiyyatdan keçirmək üçün ConfigureServices hissəsindən həyata keçiririk.
* Configure isə middlevvare-ləri qeydiyyatdan keçirir.
* App.IsDeveloper yoxlayır ki proyektin başlama environmenti Developmentdir ya yox.
* App.UseDeveloperExceptionPage() xətaları göstərən middlevvare-dir.
* App.UseHttpsRedirection() bir səhifədən başqa səhifəyə göndərə bilmək mümkünlüyü.
* App.useStatic() vvv.root folderini tanıyır.
* App.useRouting() hər hansı səhifədən başqa səhifəyə route etmək mümkünlüyü.
* App.UseEndpoints()
* /home/index/{id?} burda id? Routing params sayılır. Home controller index actiondur
* Action Methodların access modifier-i public olmalıdır.
* Reflection CSharp-da olan bir texnologiyadır ki runtime vaxtı hər hansı bir classın içindən methodu tapıb onları start edə bilir. Hər bir classın meta-datası olur və o metadatanın içindən propertyləri action methodları və s. Tapa bilir.
* ASP.Net Core sırf Reflectionun üzərindən işləyir.
* ASP Requestləri pipeline üzərindən keçib sonra response qaytarır. Pipeline daxilində Middlevvarelər olur. Middlevvarelərdən yoxlamalar aparmaq üçün istifadə etmək mümkündür.
* Pipeline qayıdanda da yoxlayır çünki ola bilsin ki qayıdan vaxtı token-in vaxtı bitər və unauthorized olar.
* Custom Middlevvare-lər yazmaq mümkündür.
* Middlevvare-lər özündə RequestDelegate Next saxlıyır.
* Middlevvare-lər bir birinə chain of responsibilities pattern ilə bağlanıblar.
* Bir element digər elementin referansını saxlamaq deməkdir ChainOfResponsibility Pattern.
* Status kodlar İnformational (100-199) Success (200-299) Redirection (300-399) Client Error (400-499) Server Error (500-599)
* Əgər Action methodunun üzərində heçnə yazılmayıbsa default olaraq HttpGet olur
* HttpGetdən başqa hamısında mütləq qeyd etmək lazımdır ki method nə tipindədir.
* Put ilə Patch eyni şeydir. Patch sadəcə obyektin bir hissəsini dəyişmək üçün istifadə olunur.
* Head var o da headerdən hər hansı məlumatı çəkmək üçün istifadə olunur.
* Ok(), BadRequest(), NotFound() action resultu var.
* Vievv-a məlumat göndərəndə o nə tipdədirsə atmodel ilə istifadə edib onun tipini qeyd etməliyik vievv-da
* Məlumatın içindəki datalara çatmaq üçün atModel yazmaq lazımdır.
* Birdən çox məlumat göndərəndə VievvModel yaratmaq lazımdır.
* Tək parametrləri RouteParams ilə 2 və daha çox queryparams. İri həcmli obyekt olanda Post request ilə.
* Id? Routeparams olanda id ola da bilər olmaya da bilər.

Lesson 2