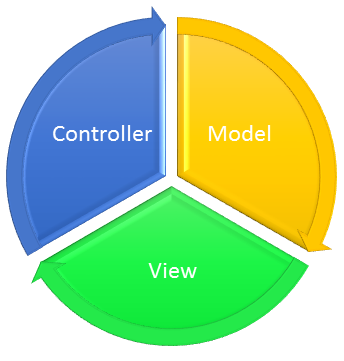
1. **GİRİŞ**

ASP.NET MVC, MVC pattern’ini ASP.NET’e eklemek için Microsoft’un geliştirdiği framework’tür.

MVC, uygulama geliştirmede (özellikle web uygulaması geliştirmede) önemli yere sahip mimari desenlerden biridir. Günümüzde MVC denince akla Microsoft’un geliştirdiği ASP.NET MVC Framework gelmektedir, oysa 1979 yılından beri (Microsoft 1975 yılında kurulmuştur) yazılım dünyasında yer almaktadır. MVC, Model , View , Controller kelimelerinin baş harflerinden oluşur ve her kelime MVC’nin farklı bir katmanını ifade eder.



* 1. **Model**

MVC dünyasında model uygulama verisinin veya durumunun saklandığı yerdir, genellikle veritabanı veya xml/json dosyası formatındadır.  
Model, veri katmanını (database, xml, json dosyası, vb.) uygulamadan izole eder, böylece diğer katmanlarda veri katmanının neresi olduğunun bilinmesine gerek kalmaz.  
Model katmanı sıklıkla Entity Framework, Nhibernate, LLBLGen, vb. gibi araçlar kullanılarak oluşturulur.

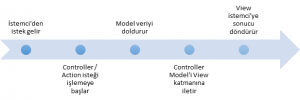
* 1. **View**

View, istemcinin gördüğü arayüzü içeren katmandır, genellikle Model katmanındaki verinin kullanılması ile oluşturulur. View katmanının Model ve Controller katmanlarından ayrılması ile arayüz değişikliklerinin uygulamanın diğer katmanlarını değiştirmeye gerek kalmadan yapılabilmesi sağlanmıştır.

View katmanında HTML5 ve CSS3 gibi son versiyon teknolojiler kullanmak mümkündür. HTML5 ve CSS3 ile masaüstü ve mobil tarayıcılarda çalışabilen uygulamalar geliştirmek çok kolaylaşmıştır. *Hatta Windows Store uygulamaları geliştirmek için HTML5 ve CSS3 teknolojilerinden yararlanılabilir.*

* 1. **Controller**

Controller, istemciden gelen isteği işlemek, Model ve View katmanları arasında köprü olmak gibi görevleri yerine getirir. Controller içerisinde bir veya daha fazla Action olabilir, genellikle her Action bir web sayfası üretmek için kullanılır.

[](http://www.borakasmer.com/wp-content/uploads/2014/07/MVC_02.png)

*MVC Life Cycle*

MVC’nin diğer bir önemli yapıtaşı Routing mekanizmasıdır.

* 1. **Routing**

Routing, istemci’nin uygulamaya yaptığı isteği uygun Controller ve Action’a yönlendiren yapıdır. İstemci, isteği uygulamanın belli bir adresine gönderir, routing mekanizması sayesinde ilgili adres için en uygun Controller ve içerisindeki Action tespit edilir ve çalıştırılır.

1. **Projenin Amacı**

Projede amaç, HTML,CSS,ASP.NET,JAVA, gibi programlama dillerinden birini, MSSQL, MYSQL, TSQL veritabanı yönetim sistemlerinden herhangi biriyle birlikte kullarak staj yönetim sistemi geliştirmektir.

1. **Projenin Geliştirme Yaklaşımı**

Projede kullanılmak üzere iterative proje geliştirme yaklaşımı kullanılmıştır. Bu model, proje geliştirme safhasında projenin erken bölümlerinde hataların tespit edilip düzenlenmesine olanak sağlayan bir model olup, bu sayede proje geliştirme sürecinin daha etkin kılınmasına olanak sağlanmıştır. Bu yöntemde, proje geliştirme safhaları birçok parçaya bölünmüş olup, süreçlerin kontrolü küçük parçalar halinde yapılmıştır. Bu yöntem sayesinde bölünen parçalar tamamlandığında test işlemleri de yapılmış olur.

1. **Proje Ekip ve Görevleri**

* Mustafa KILINÇ -> Gereksinim analizi, rapor taslağı, sorguların yazılması, MVC classlarının implementastonu.
* Zeynep KURT -> Veri tabanı tasarımı, ER diyagramının oluşturulması.
* Serkan -> Oluşturulan tabloların Entitiy Framework Code First’e import dahil edilmesi.

1. **Proje Hakkında Bilgiler**

**Kullanılan Dil :** ASP .NET CORE MVC

**Kullanılan Malzemeler :** Kişisel bilgisayarlar, Visual Studio 2015

**Desteklenen Sistemler :** Windows, MacOS

**Güvenilirlik :** Herhangi bie yedekleme yapılmadığı için saldırılara karşı güvenli.

**Kullanılabilirlik :** Kullanıcı Ana sayfada bulunan stajları inceleyebilir, öğrenci, komisyon, staj ve mülakat sekmelerinden ilgili bilgileri elde edebilir.

1. **WEB GELİŞTİRMEYE GENEL BAKIŞ**
   1. **Web Nedir?**

İngilizcede Worl Wide Web sözcüklerinin baş harflerinden oluşan bir kısaltmadır. Dünyayı saran Web olarak da açıklayabiliriz. Kısaca Web sözcüğü ile ifade edilir.

* 1. **Web Geliştirme Nedir?**

Web sayfaları, özel bir programlama dili ile yazılır. Bu dile HTML (Hyper Text Markup Language) denir. Bu dil yalnız yazıları değil, grafik, resim, ses iletimi de yapabilmektedir. Yazı, resim, ve grafiklere hareket katarak, kısaca video diye adlandırılan dinamik web sayfaları geliştirilebilmektedir. Bunların hazırlanabilmesi için birçok yazılım kullanılır. Bu yazılımların her biri ayrı bir uzmanlık ister. Ancak kullanıcı bunları bilmek zorunda değildir. Web sayfaları hazırlayanlar, kullanıcının önüne etkileşimli, kullanıcı dostu menüler koyar. Fare yardımıyla, kullanıcı, web servisinin izin verdiği her işi yapabilir.

* 1. **Web Back-End Development**

Back-End development, bir web sitesinin arka planında yapılan işlerdir. Bu bağlamda bir web sitesinin veri tabanı mimarisini oluşturan, web sitesindeki kullanıcı yetkilendirmelerini yapabilen, web sitesinin işlevselliği ile ilgili kodları yazmaktır.

* 1. **Front-End Development**

Front-End development, bir web sitesinin ön planında olan işleri yapmaktır. Web sitesinin görsel işleri, nerede hangi buton olacak, renkler nasıl olacak gibi konuları içeren kısımdır. Bu anlamda HTML, CSS, JAVASCRIPT ve diğer web sitesinin ön planında kullanılan araçlar, web sitesinin görsel ara yüzünü oluşturur. Aynı zamanda Back-End’te yapılan işler Front-End’e bağlanır.

1. **Projenin Anlatımı**
   1. **Entity Sınıfları**

Projeye başlarken öncelikle kullanacak olduğumuz entity sınıfları eklenmiştir. MVC projelerinde veritabanı işlemlerinin yapılması için klasik olarak kullanılan Microsoft SQL Server veri tabanı oluşturma işlemini kullanmamaktayız. Bu yöntem yerine, Entity Framwork yapısını kullanarak veri tabanı yapısı, MVC projesi içerisinde bulunan Model klasörünün içerisinde oluşturulmuştur. Bunun için öncelikle MVC projemize Code First için Entity Framework yapısı dahil edilmiştir.

* 1. **Context Sınıfları**

Context sınıflarının amacı, Entity Framwork Code First’ün kullanılmasını ve uygulama içerisindeki veri tabanı yapısının kullanılabilmesini sağlamaktadır. Model klasörünün içerisine kullanılacak olan tablo yapıları eklenip veri tabanı yapısı kullanılabilmektedir.

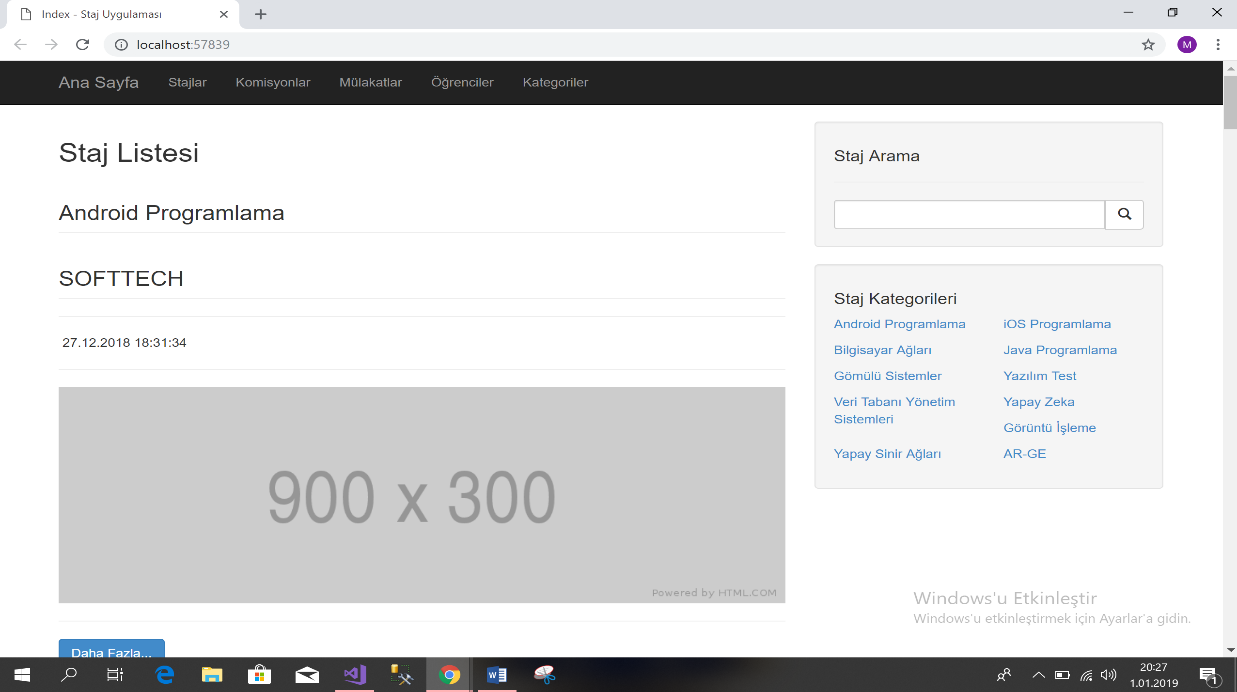
* 1. **Initializer Sınıfı**

Initializer sınıfının oluşturulmasındaki amaç, veri tabanı şeması içerisindeki yapılarda herhangi bir değişiklik yapılması halinde context sınıfının bağlı olduğu veri tabanında da bir değişiklik yapılması gerekmesidir. Değişiklik söz konusu olması halinde iki seçenek vardır. Birincisi, veri tabanını silip yeniden oluşturmak, ikinci seçenek ise bir initializer sınıfı oluşturup, herhangi bir değişiklik olduğunda otomatik algılanıp son değişikliklerin veri tabanı şemasına uydurulmasıdır.

* 1. **Controller**

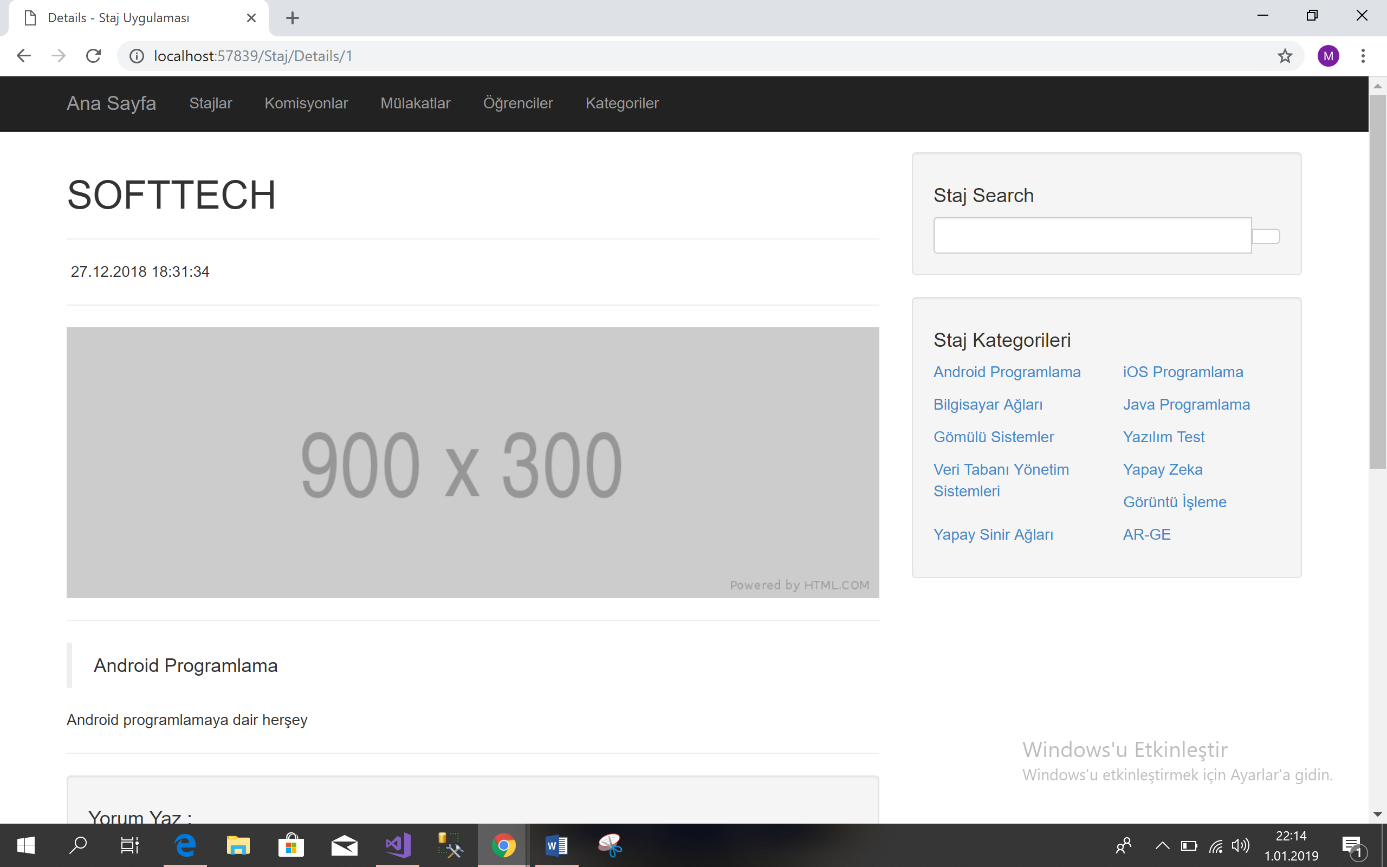
Controller, istemciden gelen isteği işlemek, Model ve View katmanları arasında köprü olmak gibi görevleri yerine getirir.

* 1. **Home/Index.cshtml Sayfası**



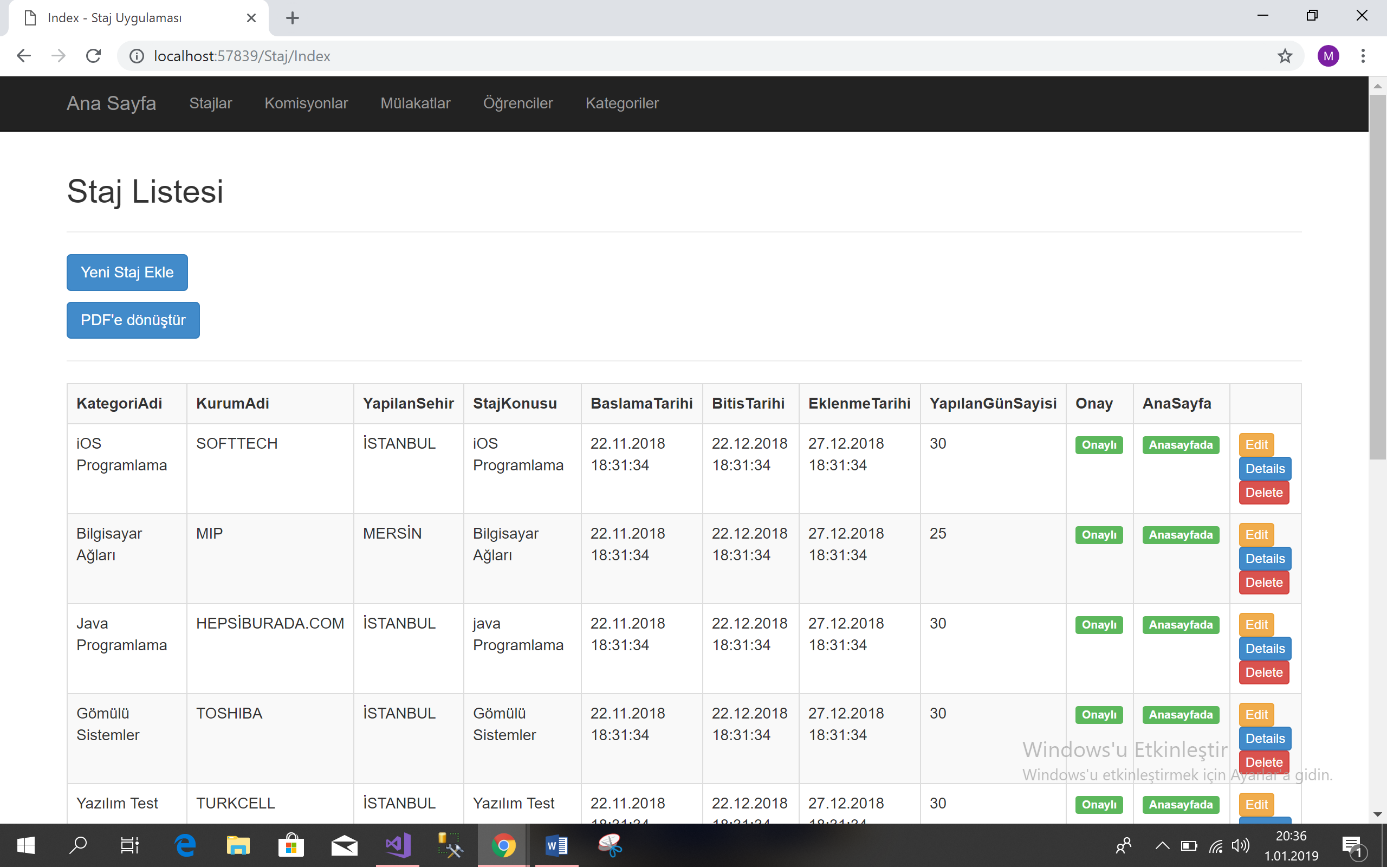
**Şekil 7.5.a Home/Index.cshtml Sayfası**

Şekil 7.5.a, Home/Index.cshtml sayfasıdır. Senaryoya göre bu sayfa ana sayfadır. Ana sayfa içerisinde yapılmış olan stajlara ait kategori bilgisi, kurum adı bilgisi ve stajın sisteme yüklenme tarihi bilgisi bulunmaktadır. Ayrıca yapılan staj ile ilgili resim alanı koyulmuştur. Bu sayede ana sayfa görselliğinin ön plana çıkarılması amaçlanmıştır. Ana sayfada, yapılan staj kategorilerine, kurum isimlerine ve staj yapılan şehirlere göre arama yapılabilmektedir. Ayrıca staj kategorileri kısmından ilgili kategoriye ait yapılmış olan staja ulaşılabilmektedir. Yapılmış olan staj altında bulunan “Daha fazla” butonu ile ilgili staja ait içeriğe ulaşılabilmektedir.



**Şekil 7.5.b Staj/Details.cshtml sayfası**

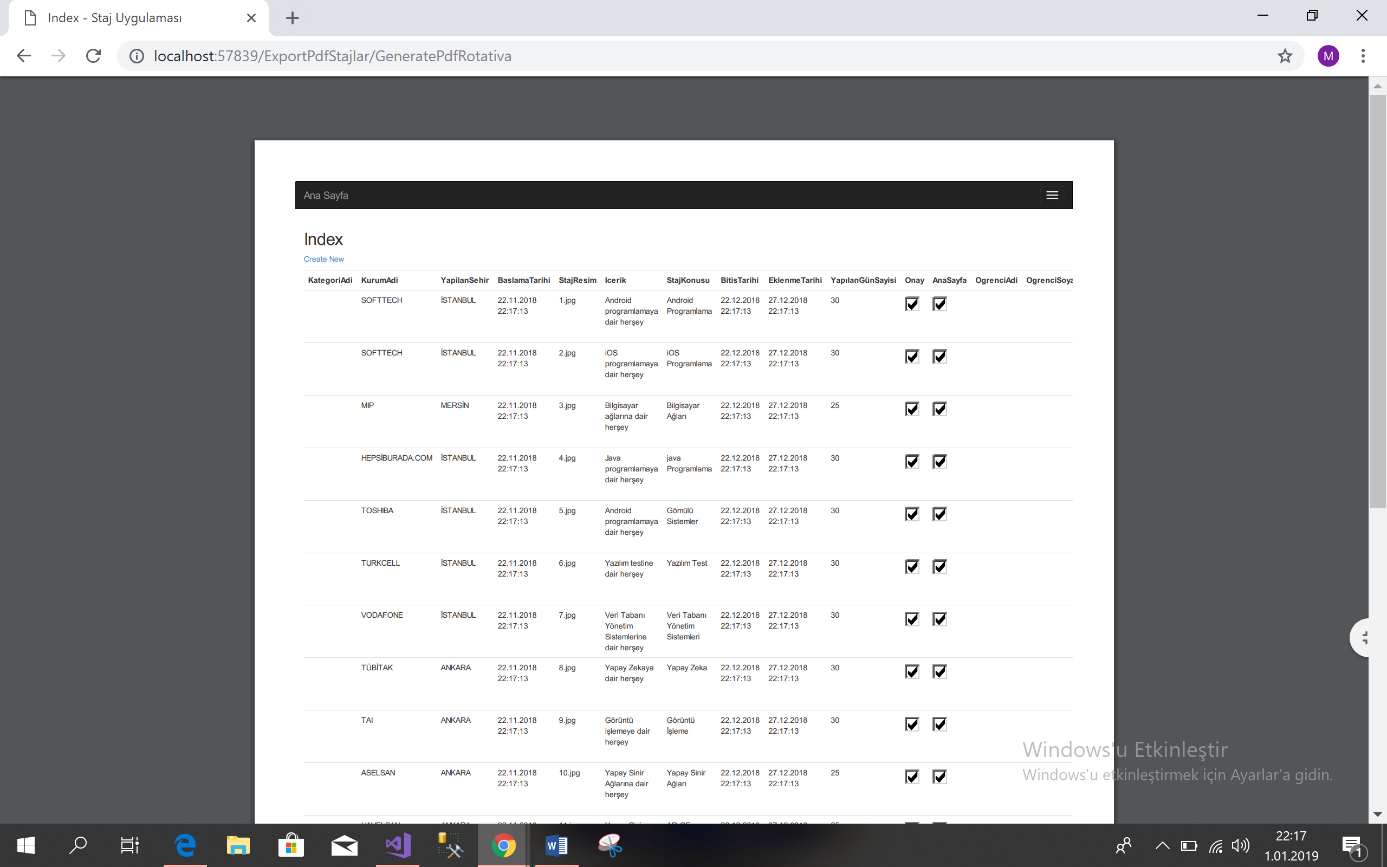
* 1. **Staj Sayfası**



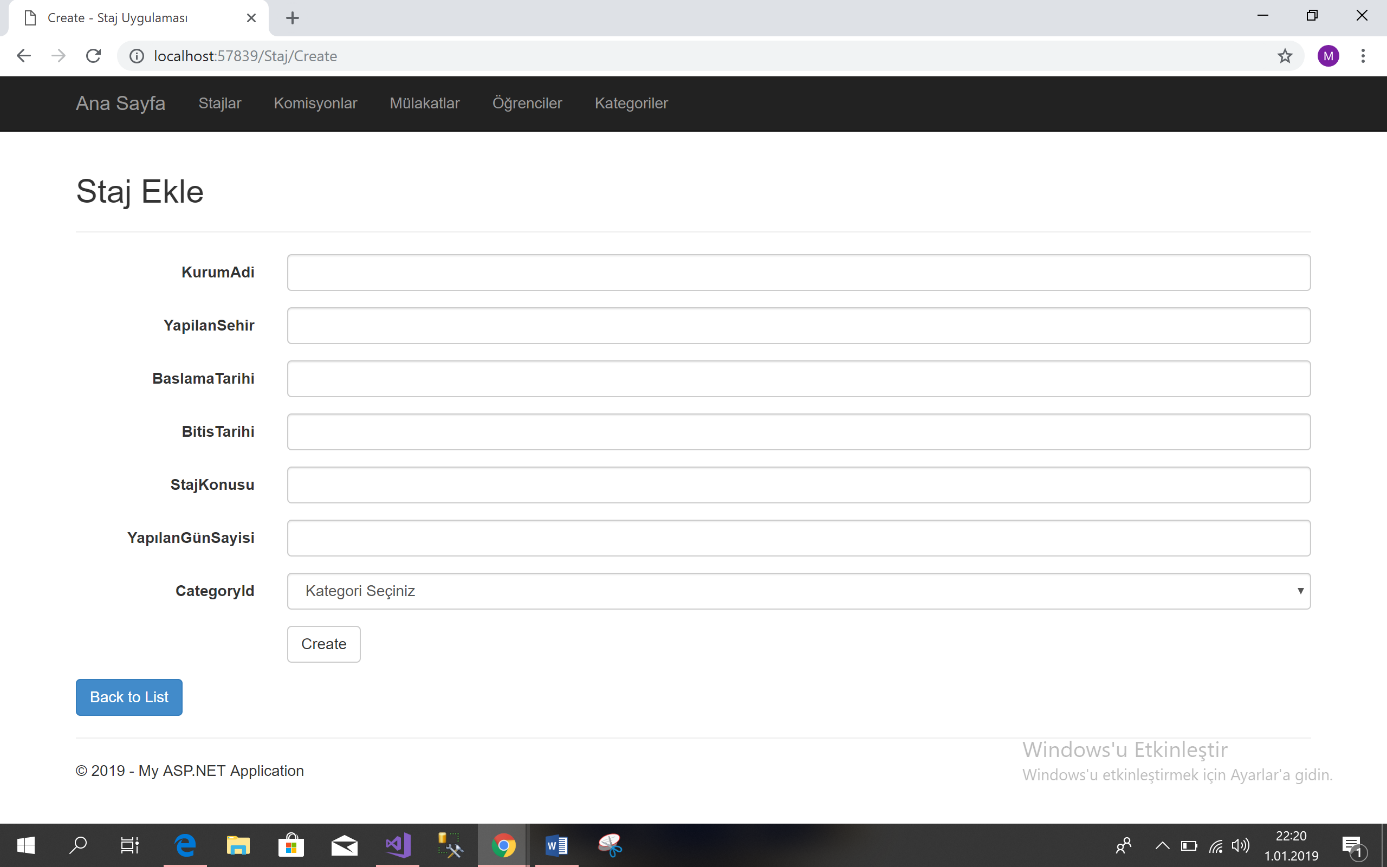
**Şekil 7.6.a Staj/Index Sayfası**

Staj/Index sayfasında yapılmış olan stajların liste şeklinde gösterilmesi amaçlanmıştır. Bu listede, yapılan staja ait kategori, kurum adı, stajın yapılmış olduğu şehir, ilgili stajın konusu, staja başlama tarihi, stajın bitiş tarihi, stajın sisteme eklenme tarihi, yapılmış olan gün sayısı bilgileri ve ayrıca, stajın onay bilgisi ve anasayfada olup olmadığı bilgileri de bu listede görülmektedir.

Yan tarafta bulunan Edit butonu ile ilgili staj bilgisi güncellenebilmekte, Details butonu ile ilgili staja ait detaylar görülebilmekte ve Delete butonu ile ilgili staj listeden kaldırılabilmektedir. Sağ üstte bulunan “Yeni Staj Ekle” botunu ile listeye yeni bir staj eklenebilmekte ve “PDF’e Dönüştür ” butonu ile staj listesinin pdf dökümanı elde edilebilmektedir.

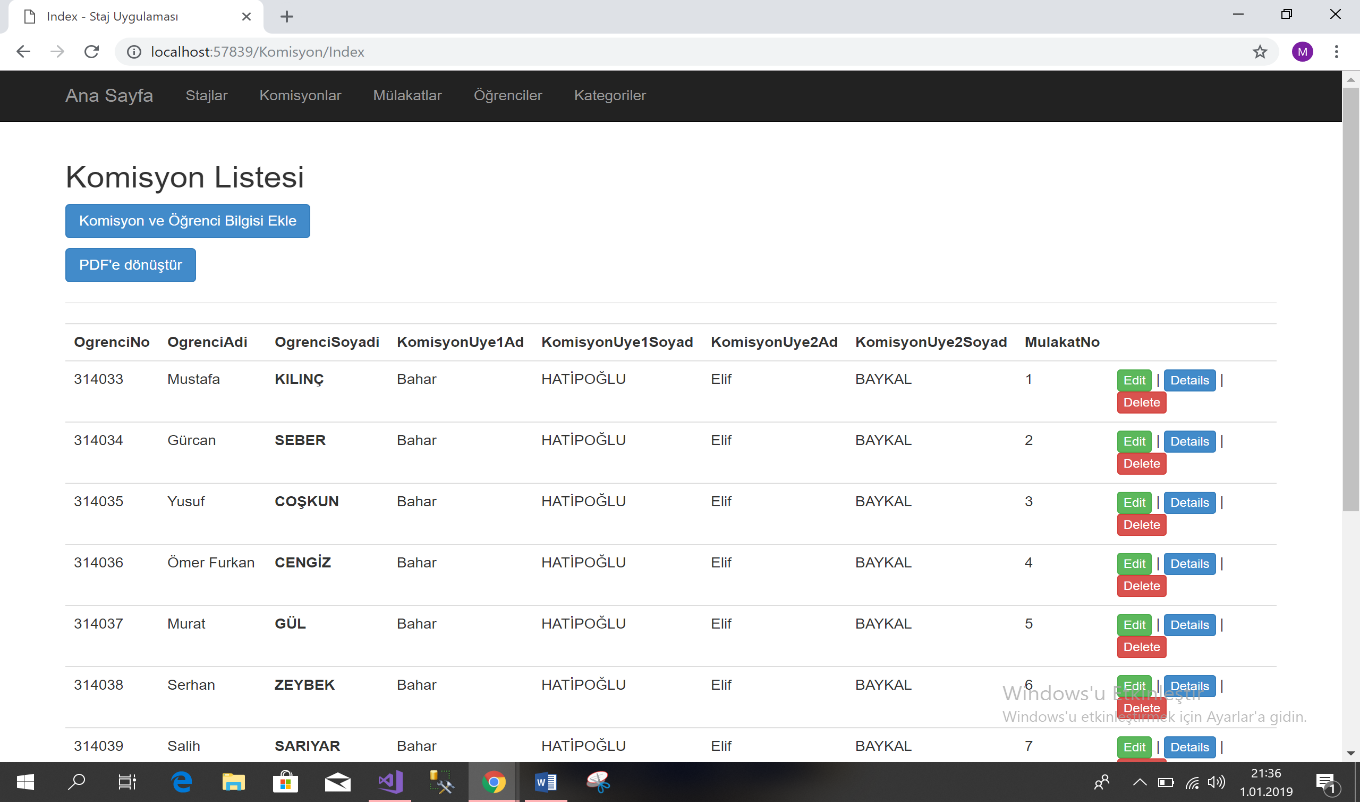


**Şekil 7.6.b Staj sayfasının pdf dökümantasyonu**



**Şekil 7.6.c Staj ekleme sayfası**

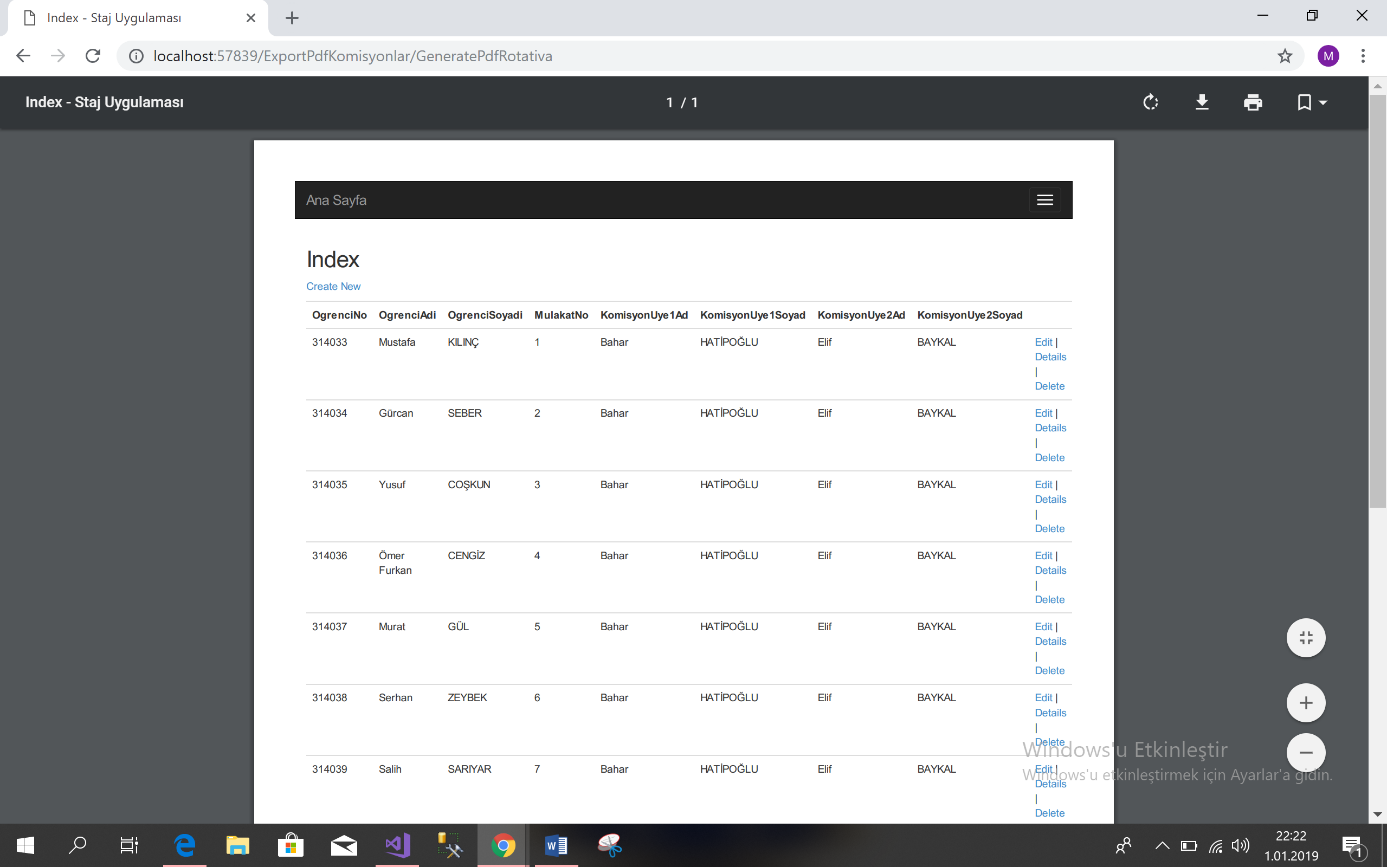
* 1. **Komisyon Sayfası**



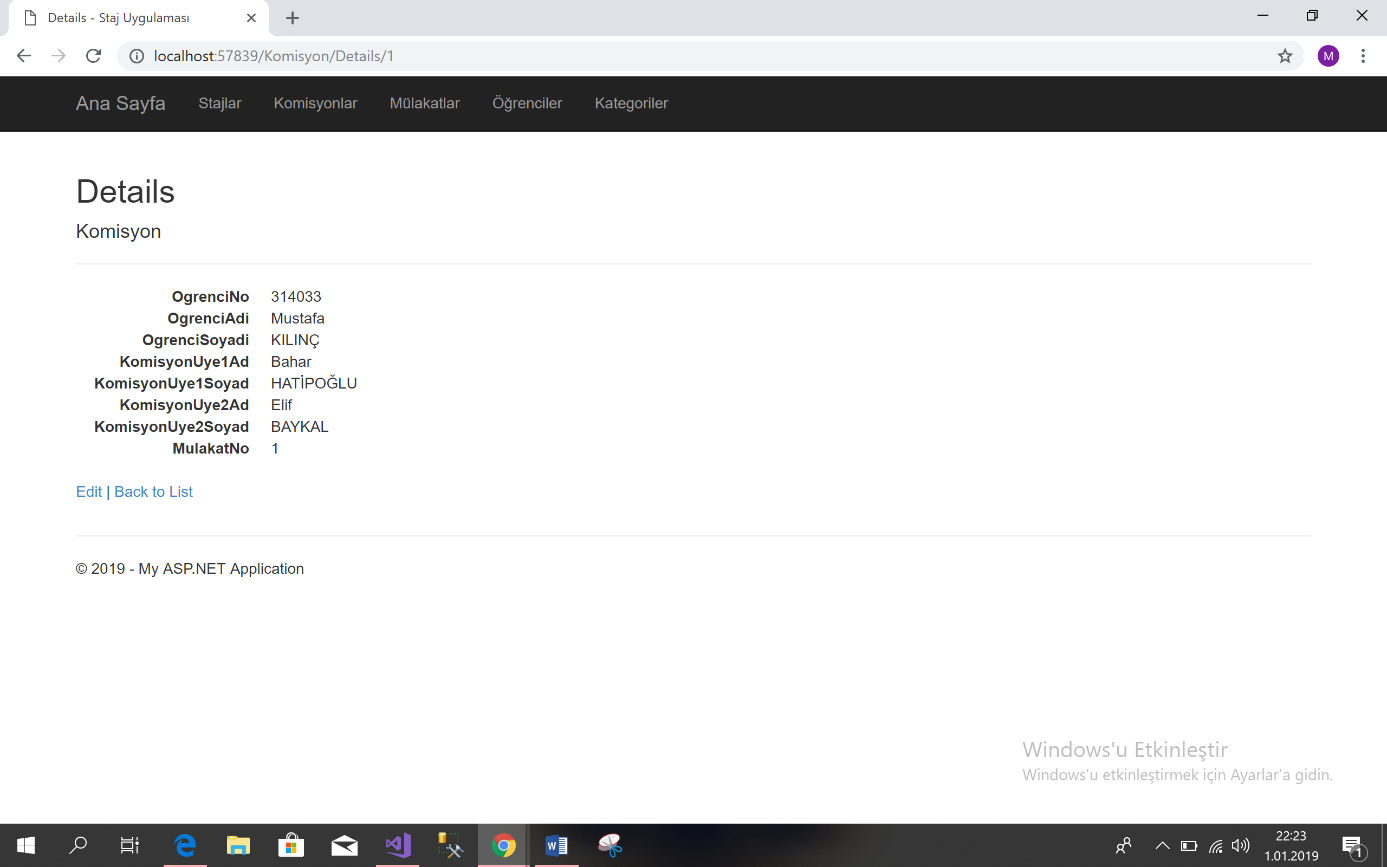
**Şekil 7.7.a Komisyon/Index Sayfası**

Komisyon/Index sayfasında öğrencilerin stajlarına ait komisyon bilgilerinin liste şeklinde gösterilmesi amaçlanmıştır. Bu listede, öğrencilere ait öğrenci numarası öğrenci adı ve soyadı bilgileri ile birlikte komisyon üyelerine ait ad ve soyad bilgileri bulunmaktadır.

Yan tarafta bulunan Edit butonu ile ilgili komisyon bilgisi güncellenebilmekte, Details butonu ile ilgili komisyona ait detaylar görülebilmekte ve Delete butonu ile ilgili komisyon listeden kaldırılabilmektedir. Sağ üstte bulunan “Komisyon ve öğrenci Ekle” botunu ile listeye yeni bir staj eklenebilmekte ve “PDF’e Dönüştür ” butonu ile komisyon listesinin pdf dökümanı elde edilebilmektedir.

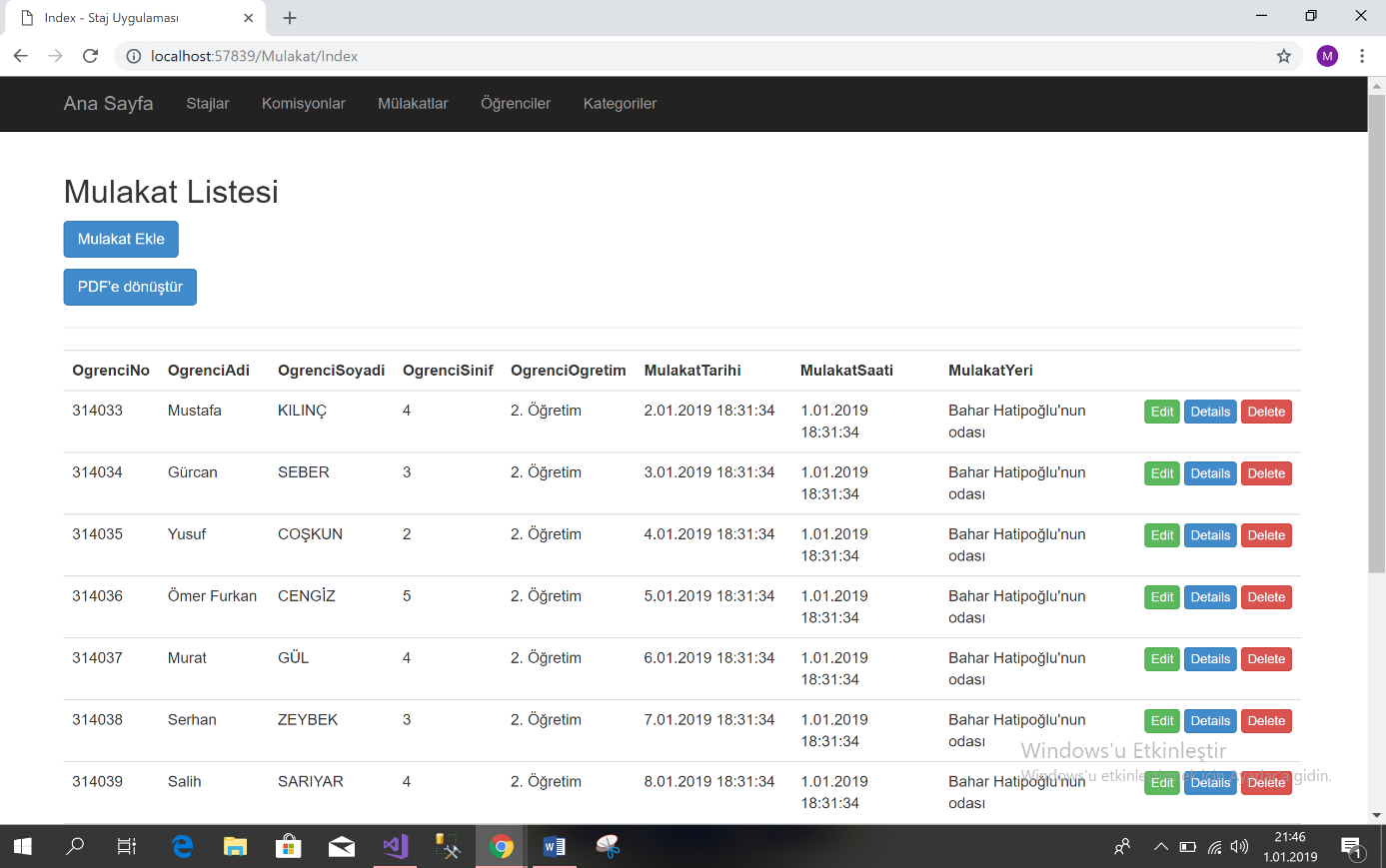


**Şekil 7.7.b Komisyon Listesini pdf dökümantasyonu**



**Şekil 7.7.c Komisyon sayfasındaki ilgili komisyonun detayları**

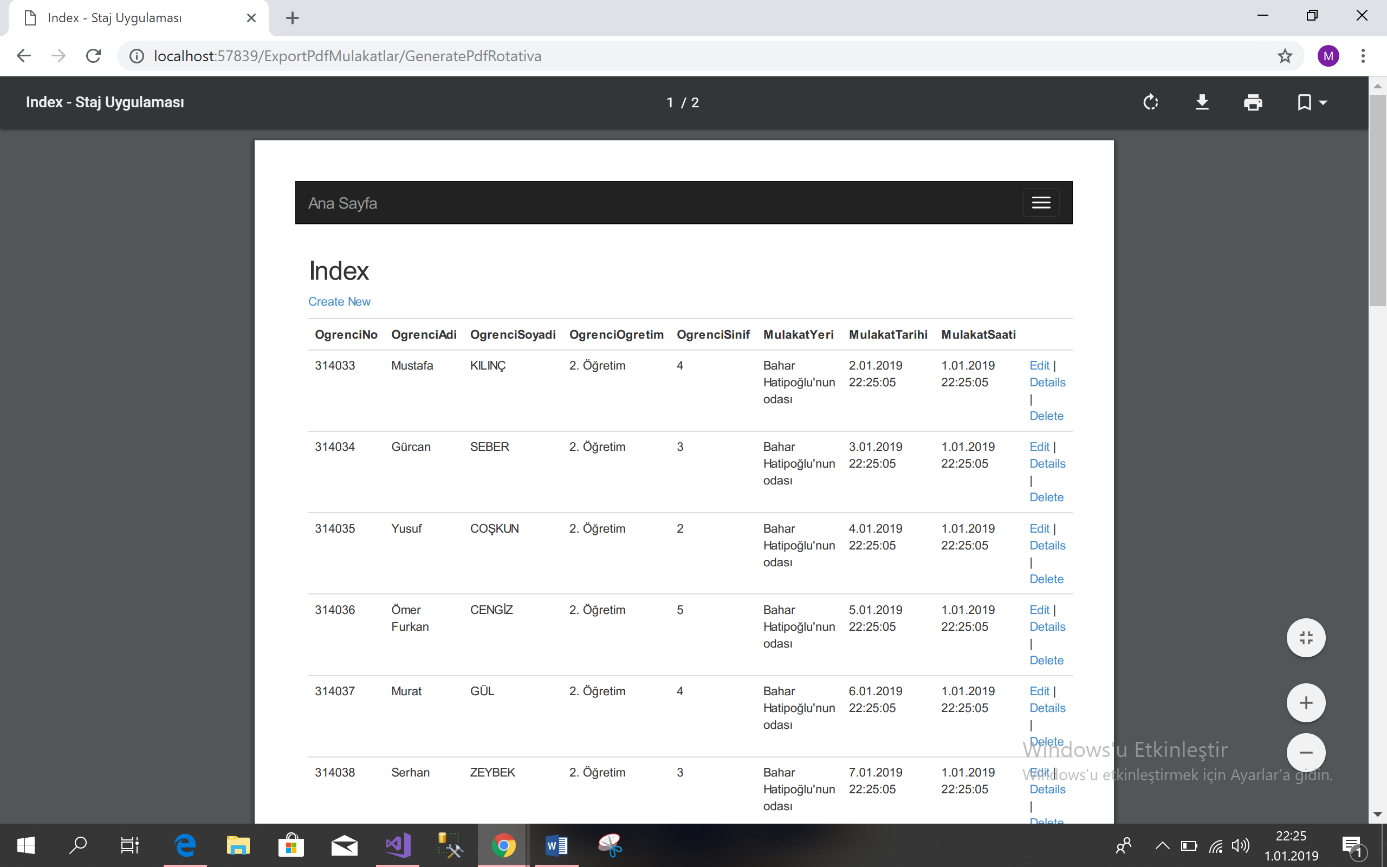
* 1. **Mülakat Sayfası**



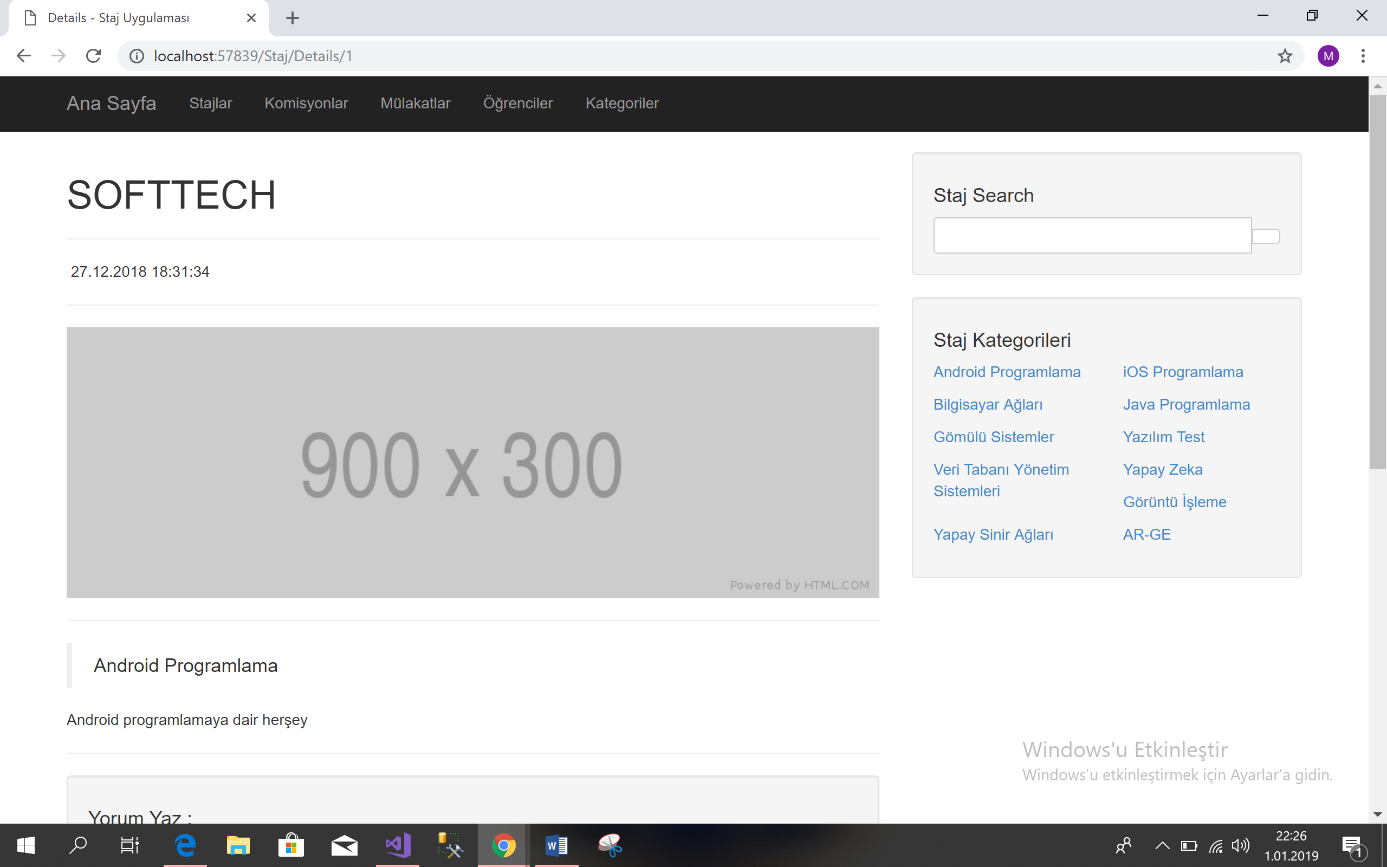
**Şekil 7.8.a Mulakat/Index Sayfası**

Mulakat/Index sayfasında öğrencilerin mülakatlarına ait mülakat bilgilerinin liste şeklinde gösterilmesi amaçlanmıştır. Bu listede, öğrencilere ait öğrenci numarası öğrenci adı ve soyadı bilgileri ile birlikte mülakata ait mülakat tarihi, mülakat saative mülakat yeri bilgileri bulunmaktadır.

Yan tarafta bulunan Edit butonu ile ilgili mülakat bilgisi güncellenebilmekte, Details butonu ile ilgili mülakata ait detaylar görülebilmekte ve Delete butonu ile ilgili mülakat listeden kaldırılabilmektedir. Sağ üstte bulunan “Mülakat Ekle” botunu ile listeye yeni bir mülakat eklenebilmekte ve “PDF’e Dönüştür ” butonu ile mülakat listesinin pdf dökümanı elde edilebilmektedir.

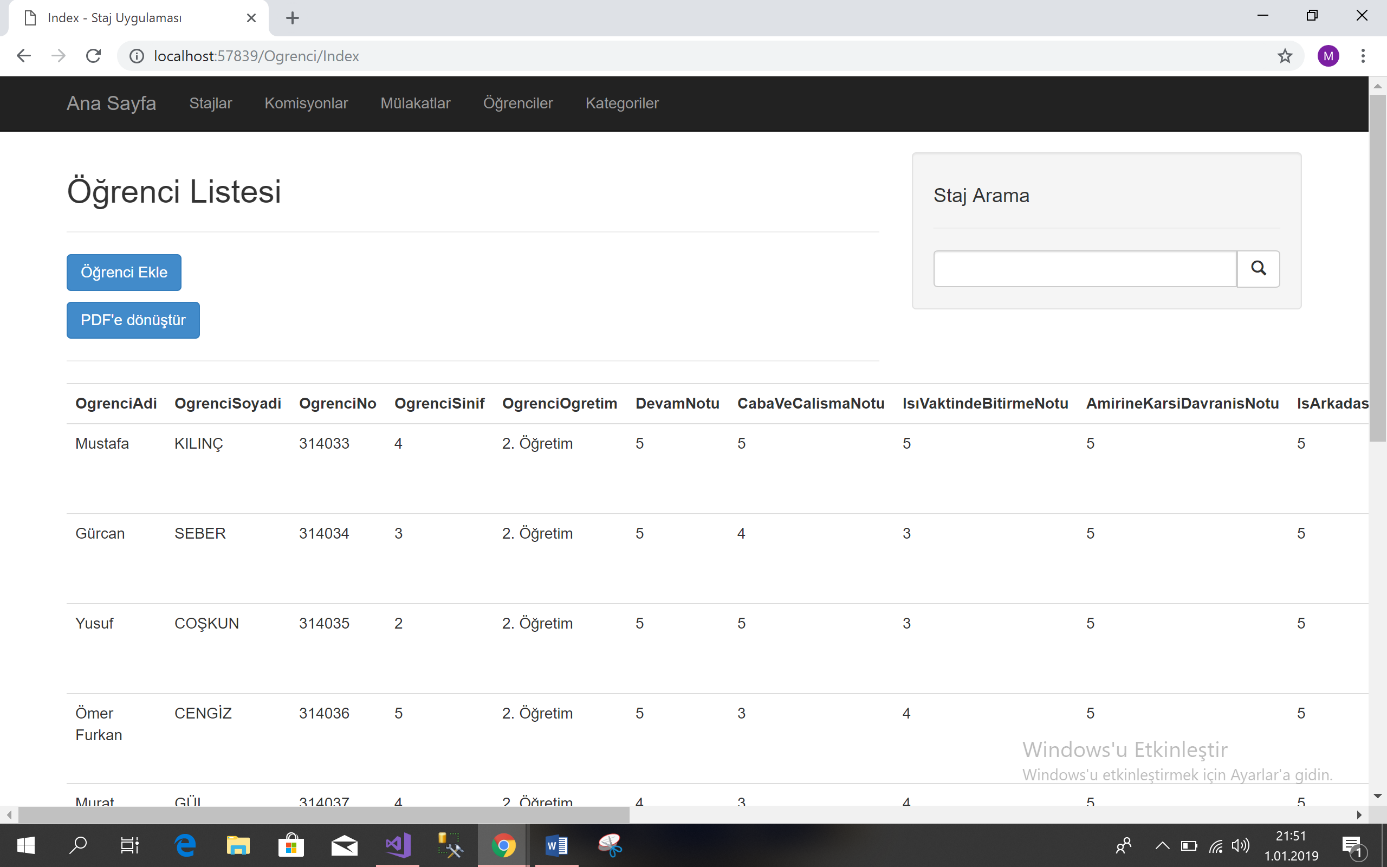


**Şekil 7.8.b Mulakat listesinin pdf dökümantasyonu**

****

**Şekil 7.7.c İlgili mülakata ait detaylar**

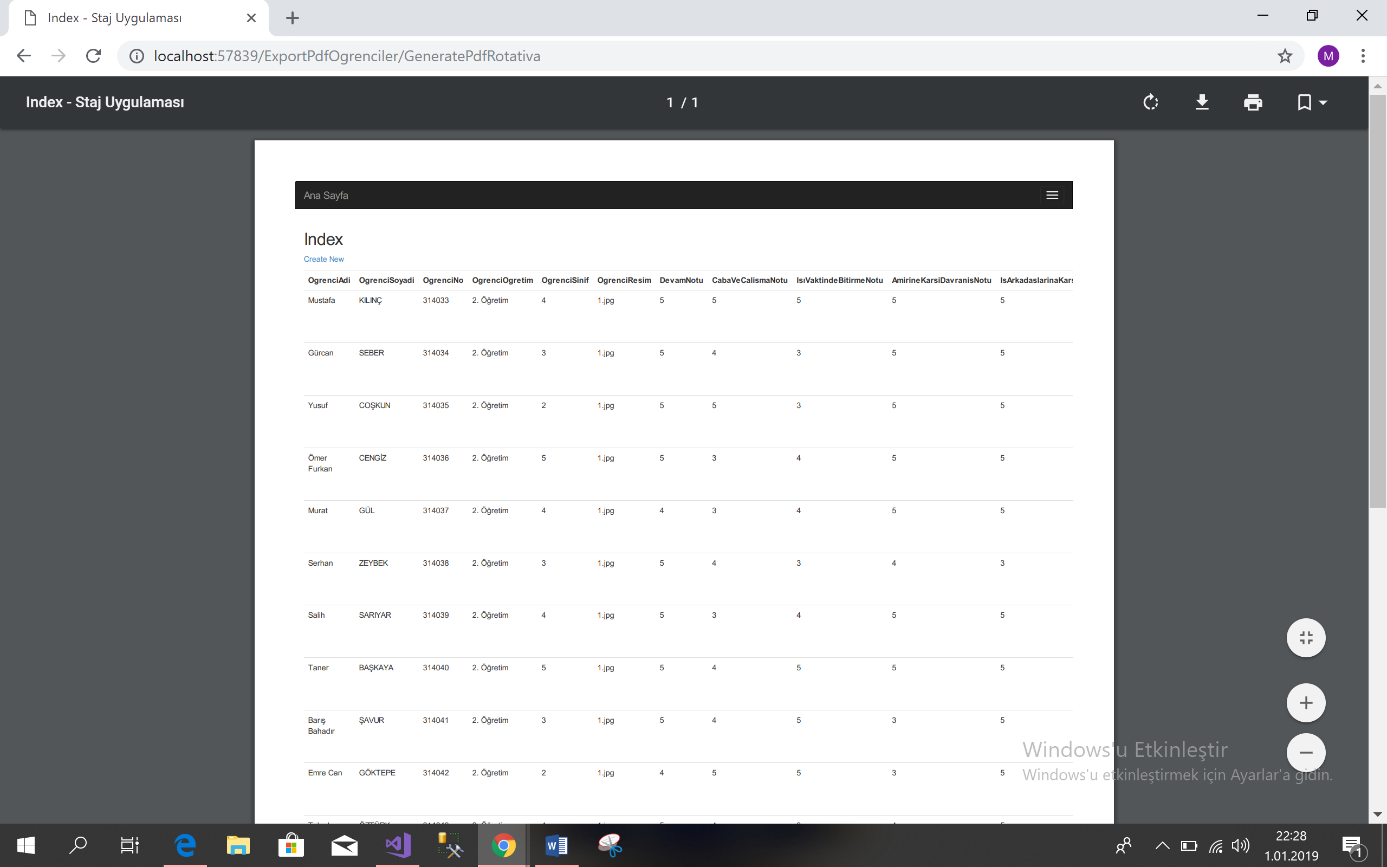
* 1. **Öğrenci Sayfası**



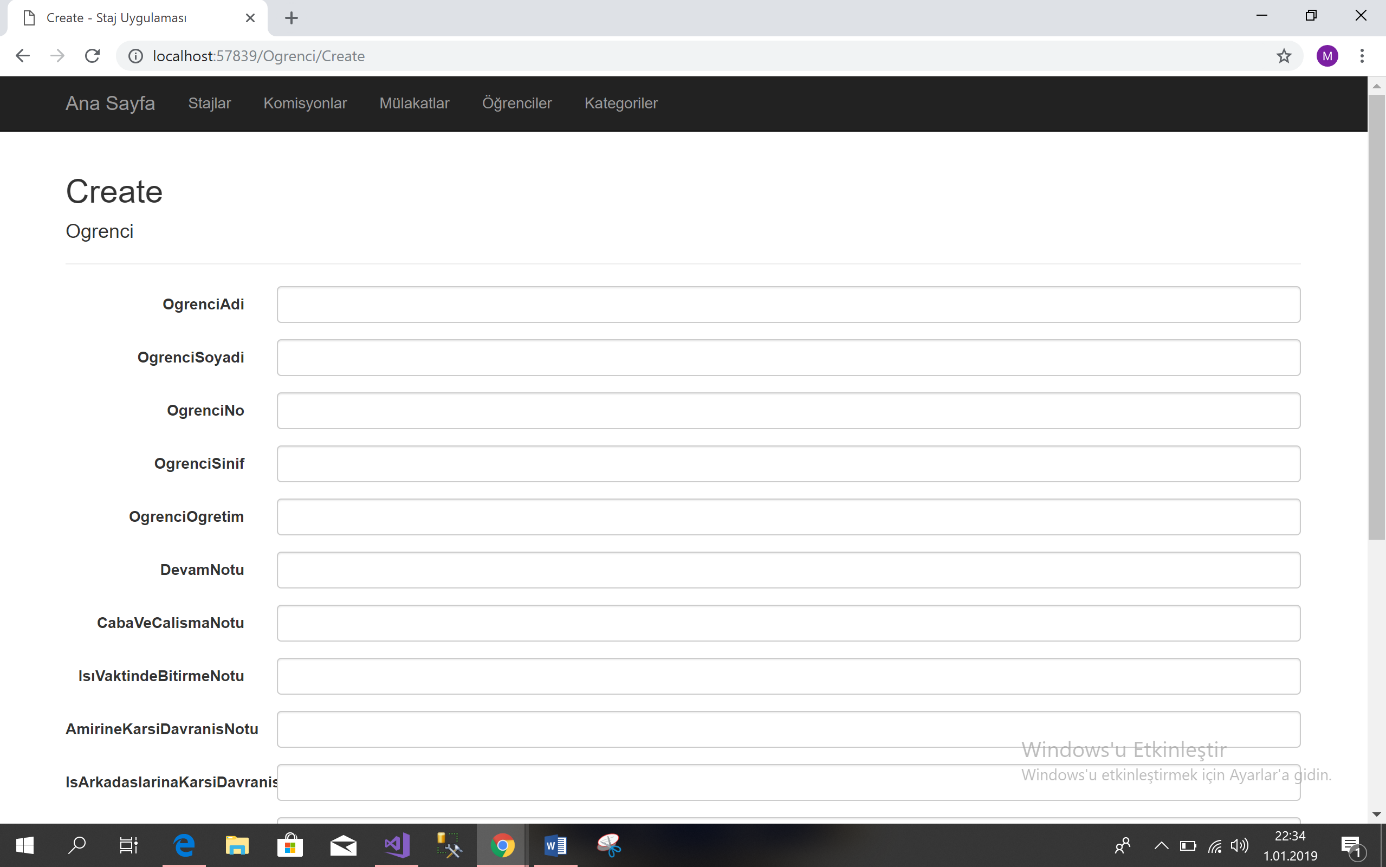
**Şekil 7.9.a Öğrenci/Index Sayfası**

Öğrenci/Index sayfasında öğrencilere ait bilgilerinin liste şeklinde gösterilmesi amaçlanmıştır. Bu listede, öğrencilere ait öğrenci numarası, öğrenci adı ve soyadı, numarası, sınıf bilgisi, öğretim bilgisi bilgileri ile birlikte, Devam notu, çaba ve çalışma notu, işi vaktinde bitirme notu, amirine karşı davranış notu, iş arkadaşlarına karşı davranış notu, proje notu, düzen notu, sunum notu, içerik notu ve mülakat notu, hesaplanan not katsayısı, staj yapılan gün sayısı, kabul edilen gün sayısı, kalan gün sayısı, stajın tamamlanma durumu bilgileri bulunmaktadır.

Yan tarafta bulunan Edit butonu ile ilgili öğrenci bilgisi güncellenebilmekte, Details butonu ile ilgili öğrenciye ait detaylar görülebilmekte ve Delete butonu ile ilgili öğrenci listeden kaldırılabilmektedir. Sağ üstte bulunan “Öğrenci Ekle” botunu ile listeye yeni bir öğrenci eklenebilmekte ve “PDF’e Dönüştür ” butonu ile öğrenci listesinin pdf dökümanı elde edilebilmektedir.



**Şekil 7.9.b Öğrenci listesinin pdf dökümantasyonu**

****

**Şekil 7.9.c Öğrenci ekleme sayfası**