

J.S. ISMAGULOVA¹, G.N. KAZBEKOVA², S. ÖZER³

¹Hoca Ahmet Yesevi Türk-Kazak Üniversitesi Yüksek Lisans Öğrencisi (Kazakistan, Türkistan), e-mail: selim.ozер@ayu.edu.kz

²PhD, Hoca Ahmet Yesevi Türk-Kazak Üniversitesi (Kazakistan, Türkistan), e-mail: zhuldyz.ismagulova@ayu.edu.kz

³PhD Hoca Ahmet Yesevi Türk-Kazak Üniversitesi (Kazakistan, Türkistan), e-mail: gulnur.kazbekova@ayu.edu.kz

ASP.NET MVC MİMARİSİYLE ÖĞRENCİ İŞLERİ BİRİMİ WEB UYGULAMASI GELİŞTİRME

Özet. Günümüzde internet teknolojisinde yaşanan gelişmeler birçok alanda kurumları yenilemeye itmiştir. Bunlardan biri de eğitim kurumlarıdır. Özellikle üniversitelerin öğrenci işleri daire başkanlığı birimleri için, gelişen teknolojiyle birlikte, etkili bir web uygulamasına sahip olmak önemli hale gelmiştir. Gelişime önem veren tüm üniversiteler hem üniversitenin tanıtımını yapacak hem de ilgili öğrenci işlerinin gerçekleştirilebileceği, tasarım ve yazılım açısından güçlü ve etkileşimli bir öğrenci işleri web uygulamasına sahip olmalıdır. Web uygulamaları ile öğrenci işlerindeki hata oranının ciddi oranda azaldığı, öğrencilere erişim ve hizmet bakımından birim performansının da ciddi oranda arttığı saptanmıştır.

Bu çalışmada Öğrenci İşleri Daire Başkanlığı web uygulaması ASP.Net MVC temeliyle geliştirilmiştir. ASP(Active Server Page – Aktif Sunucu Sayfası).Net MVC(Model View Controller – Model Görünüm Kontrol) teknolojisi web uygulamalarının daha güvenli olmasına, daha rahat tasarlanmasına ve kodlanmasına olanak sağlar. Model, görünüm ve kontrol sistemi, ön taraf (Frontend) ve arka taraf (Backend) yapılarının düzenlenmesi sürecinde kolaylık sağlamıştır. Veri tabanı olarak Microsoft SQL kullanılmış ve veri tabanı yöntemi olarak Code First yaklaşımı tercih edilmiştir. Veri tabanı ile iletişim için arka taraf yapısında Dil ile Bütünleşik Sorgu (Language Integrated Query) bileşeni kullanılmıştır.

Anahtar kelimeler: Öğrenci işleri, Web otomasyonu, ASP.Net, MVC, Entity çerçevesi, Identity çerçevesi, Visual Studio.

Ж.С. Исмагулова¹, Г.Н. Казбекова², С. Öзер³

¹Қожа Ахмет Ясауи атындағы Халықаралық қазақ-түрік университетінің магистранты (Қазақстан, Түркістан қ.), e-mail: selim.ozер@ayu.edu.kz

²техника ғылымдарының кандидаты, Қожа Ахмет Ясауи атындағы Халықаралық қазақ-түрік университеті (Қазақстан, Түркістан қ.), e-mail: zhuldyz.ismagulova@ayu.edu.kz

³техника ғылымдарының кандидаты, Қожа Ахмет Ясауи атындағы Халықаралық қазақ-түрік университеті (Қазақстан, Түркістан қ.), e-mail: gulnur.kazbekova@ayu.edu.kz

ASP.NET MVC АРХИТЕКТУРАНЫ ҚОЛДАНА ОТЫРЫП "СТУДЕНТТІК БӨЛІМ" ВЕБ-ҚОСЫМШАНЫ ӘЗІРЛЕУ

Аңдатпа. Бүгінгі таңда Интернет-технологияның қарыштап дамуы көптеген ұйымдарды әр түрлі сала өзгерістерге жол ашты. Осындай бағыттардың бірі білім беру мекемелері. Әсіресе жоғарғы оқу орындарында білім алушыларға арналған «Студенттік бөлім» интерактивті веб-қосымшасы болуы өте маңызды. Бұл мақалада қарастырылып отырған

«Студенттік бөлім» веб-қосымшасының негізгі мақсаты білім алушылар контингентінің статистикалық есебі мен мониторингі, оның қозғалысын бақылау, жеке істерін жүргізу және білім алушылар бойынша ақпаратты уақтылы ұсыну болып табылады. Веб-қосымша құжаттарды дайындауда қателіктерге жол бермеуге, сонымен қатар білім алушыларға қол жетімділікті қамтамасыз етуге және бөлімнің қызмет көрсету сапасын арттыруға үлкен септігін тигізеді.

Бұл жұмыста студенттік істердің веб-бөлімінің қосымшасы келесі негізде жасалды ASP.Net MVC. ASP технологиясы (Active Server page - белсенді Сервер беті). NET MVC (Model-View-Controller) веб-қосымшаларды жобалауға және бағдарламалауға қауіпсіз, ыңғайлы етеді. Модель, презентация және бақылау жүйесі Front-end және back-end құрылымын ұйымдастыруда ыңғайлылықты қамтамасыз етеді. Microsoft SQL дерекқор ретінде пайдаланылды және Code First әдісі дерекқор әдісі ретінде таңдалды. Back-end құрылымы мәліметтер базасымен өзара әрекеттесу үшін Language Integrated Query компоненті қолданылды.

Кілт сөздер: веб-қосымша, ASP.Net, MVC, Entity Framework, Identity Framework, Visual Studio.

Ж.С. Исмагулова¹, Г.Н. Казбекова², С. Озер³

¹магистрант, *Международного казахско-турецкого университета имени Ходжи Ахмеда Ясави (Казахстан, г. Туркестан), e-mail: selim.ozер@ayu.edu.kz*

²кандидат технических наук, *Международный казахско-турецкий университет имени Ходжи Ахмеда Ясави (Казахстан, г. Туркестан), e-mail: zhuldyz.ismagulova@ayu.edu.kz*

³кандидат технических наук, *Международный казахско-турецкий университет имени Ходжи Ахмеда Ясави (Казахстан, г. Туркестан), e-mail: gulnur.kazbekova@ayu.edu.kz*

Разработка веб-приложения для студенческого отдела с использованием архитектуры ASP.NET MVC.

Аннотация. Сегодня стремительное развитие Интернет-технологий открыло многим организациям путь к переменам в различных сферах. Одним из таких направлений являются образовательные учреждения. Особенно важно, чтобы в вузах было интерактивное веб-приложение «Студенческий отдел» для обучающихся. Основной целью веб-приложения «Студенческий отдел», рассматриваемого в данной статье, является статистический учет и мониторинг контингента обучающихся, контроль его движения, ведение личных дел и своевременное представление информации по обучающимся. Веб-приложение будет способствовать недопущению ошибок при подготовке документов, а также обеспечению доступа обучающихся и повышению качества обслуживания отдела. Приложение «Студенческий отдел» было разработано на основе ASP.Net MVC. Технология ASP (Active Server Page - Активная серверная страница) .Net MVC (Model View Controller - Модель-Представление-Контроллер) обеспечивает более безопасную, более удобную для проектирования и программирования веб-приложений. Система модели, представления и контроля обеспечила удобство при организации структуры front-end и back-end. В качестве базы данных использовалась Microsoft SQL, а в качестве метода базы данных выбран подход Code First. В составе структуры back-end использован компонент Language Integrated Query для взаимодействия с базой данных.

Ключевые слова: веб-приложение, ASP.Net, MVC, Entity Framework, Identity Framework, Visual Studio.

J.S. Ismagulova¹, G.N. Kazbekova², S. Ozer³

¹Graduate student of *Khoja Akhmet Yassawi International Kazakh-Turkish University(Kazakhstan, Turkistan), e-mail: selim.ozер@ayu.edu.kz*

²*Candidate of Technical Sciences, Khoja Akhmet Yassawi International Kazakh-Turkish University
(Kazakhstan, Turkistan), e-mail: zhuldyz.ismagulova@ayu.edu.kz*

³*Candidate of Technical Sciences, Khoja Akhmet Yassawi International Kazakh-Turkish University
(Kazakhstan, Turkistan), e-mail: gulnur.kazbekova@ayu.edu.kz*

Developing Web Application For The Student Affairs Department Using The ASP.Net MVC Architecture.

Abstract. The developments in internet technology have led many institutions to renew themselves in various fields, and one of them is educational institutions. Especially for the student affairs departments of universities, having an effective web application has become important with the developing technology. All universities that prioritize development should have a strong and interactive student affairs web application that will not only promote the university but also facilitate the implementation of related student affairs from a design and software perspective. It has been found that the error rate in student affairs significantly decreases with web applications, and the unit performance in terms of access and service to students also significantly increases.

In this study, the Student Affairs Department web application was developed based on ASP.Net MVC (Model View Controller) technology. ASP.Net MVC technology enables web applications to be more secure, easier to design, and code. The model-view-controller system has provided convenience in the organization of both the front-end and back-end structures. Microsoft SQL was used as the database, and the Code First approach was preferred as the database method. The Language Integrated Query component was used in the back-end structure for communication with the database.

Overall, this design allows for easy updates, optimization, and dynamization of the code, as well as removing repetitions from the code.

Key words: Student affairs, web automation, ASP.Net, MVC, Entity Framework, Identity Framework, Visual Studio, multilingual.

Giriş

İnternetin hayatımıza girmesi ve teknolojinin çok hızlı gelişmesi diğer sektörleri etkilediği gibi eğitim sektörünü de etkilemiştir. Web uygulamalarının güvenilir ve hızlı bir şekilde birçok işlemin gerçekleştirilmesine olanak sağlaması, artık her kurumun bir web sitesi sahibi olmasını zorunlu hale getirmiştir. Bu durum üniversitelerde de aynı etkiyi oluşturmuştur. Özellikle öğrencilerin doğrudan ve sürekli temas halinde oldukları Öğrenci İşleri Birimi için web sayfası, öğrencilere hızlı ulaşım ve doğru bilgilendirme açısından çok önemlidir[1].

Öğrenci İşleri Web Uygulamasında öğrencilere her türlü bilgilendirme ve duyuruların tek yerden ulaştırılması, öğrenci kayıtlarının gerçekleştirilmesi, kayıt yenilemeleri, ders seçimleri, not takibi, askerlik işlemleri, sık kullanılan evrakların elektronik olarak erişimi, genel konularda dilekçelerin hazır taslakları, yurt başvuruları, genel istatistik bilgileri gibi daha sayılabilecek birçok hizmet hızlı ve kolay şekilde sunulabilmektedir[2].

Bu tür web uygulamalarının hazırlanmasında kullanılabilen çok çeşitli yazılım dilleri ve teknolojiler bulunmaktadır. Kullanım amacına göre bu diller ve teknolojilerin tercihi değişebilir. Bu çalışmada kolaylık, dinamiklik, esneklik, rahat tasarlama ve kodlama açısından ASP.Net MVC temelini kullanmayı tercih edildi.

ASP.Net Microsoft tarafından geliştirilen bir çerçevedir (Framework). MVC ise ilk olarak 1979 yılında tanıtılmış bir desendir. Yazılımcılara HTML, CSS, JavaScript, LINQ, C# gibi birden fazla teknolojinin bir sayfa içerisinde rahat bir şekilde kullanım kolaylığı ve diğer teknolojilerle entegrasyon kolaylığı sağladığı için yaygın olarak tercih edilir[3].

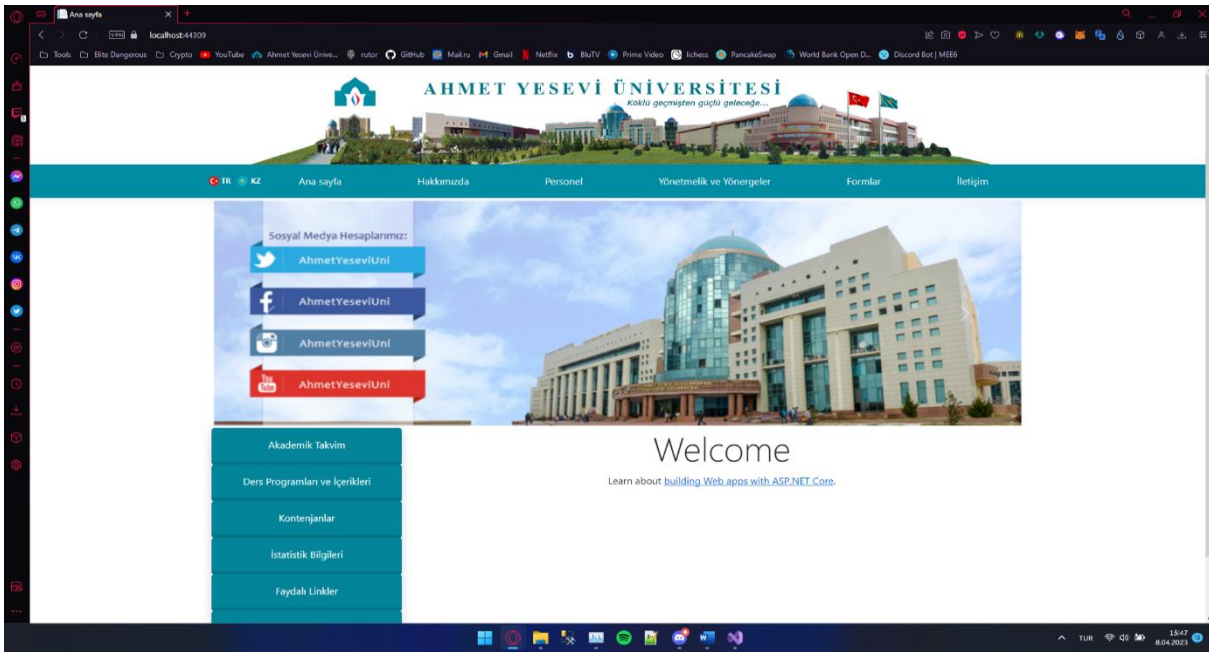
Planlama

Planlama bölümü, yazılım geliştirme sürecinin karmaşıklığı ve tekrarlanma problemleri içermesinden ötürü önemli bir alandır. Bu bölümde, uygulamaya ait daha soyut ve belirsiz fikirler yer alır.

Öncelikle hazırlanacak olan web uygulamasının gereksinimleri belirlenir. Bu noktada Sitede çoklu dil seçeneği olacak mı? Yönetim Paneli oluşturulacak mı? Farklı elektronik cihazlarda görüntülemeye uygun, duyarlı web tasarımı (Responsive) yapılacak mı? Gibi sorulara verilen cevaba göre planlama yapılır. Bu çalışmada çoklu dil desteği, yönetim paneli ve duyarlı web tasarımı yapılacaktır

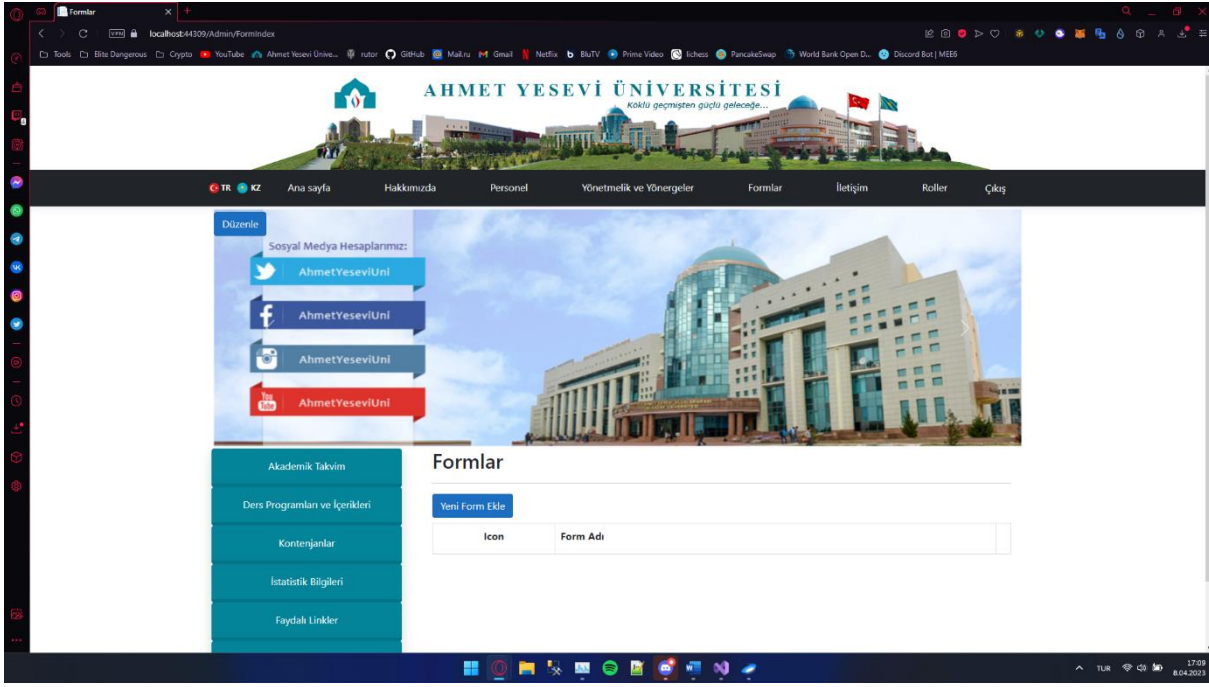
Sayfa Tasarımı ve Görünüm

Sayfa tasarımında önce ana renkler belirlenir. Bu ana renkler genelde kurumun logosundaki renkler baz alınarak belirlenir ve sayfa tasarımının genel hatları bu renkler üzerinden ilerler. İkinci aşama olarak sayfanın katmanları (Layout) belirlenir.



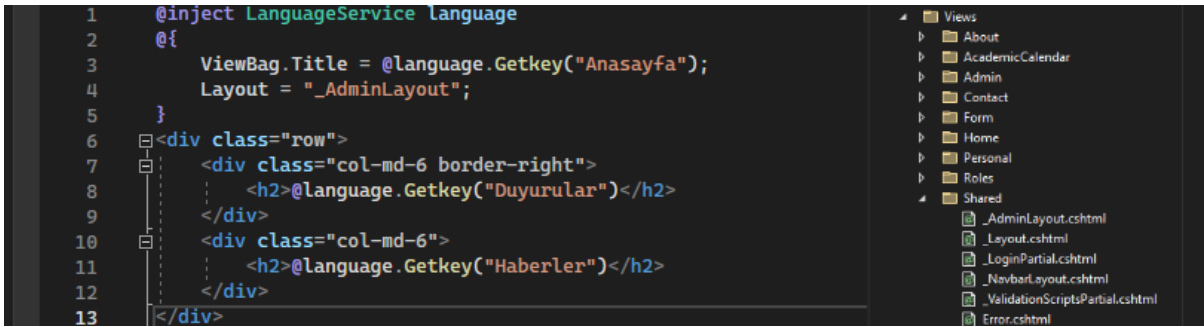
Resim-1 – Web sayfasının ana katmanı

Uygulama içerisinde bütün sayfalarda görünür olmasını istediğimiz kısımlar, ana katman içerisinde tasarlayarak belirtilir. Genel kullanımda, uygulama içi gezintiyi kolaylaştırmak için ana katmanda sadece yönlendirme menüsü (Navbar) tasarlanır ve bütün sayfalarda kullanılabilir olur. Tercihe göre Resim-1'deki gibi yan menüler de (Sidebar) kullanılabilir.



Resim-2 – Yönetim Paneli ana katmanı

Yönetim Paneli modülü kullanılacak olan uygulamalarda Yönetim Paneli için ayrı bir katman oluşturulur. Bu katman da bir ana katman görevi görür ve yöneticiler için oluşturulan sayfalar arası yönlendirmelerde karışıklığı azaltır. Resim-2’de görüldüğü gibi genel olarak kullanıcı tarafından tasarının benzeri olur. Katman benzerliğinden kaynaklanan karışıklıkları önlemek adına ana renklerde değişiklik yapılabilir. Yönetim Panelinin geliştirilmesindeki amaç, uygulamayı yönetecek olan birim veya kişinin herhangi bir yazılım bilgisine sahip olmadan sayfalar üzerinde değişiklik yapabilmesini sağlamaktır. İçeriklerin yanına eklenen butonlar aracılığıyla sayfalarda düzenleme ve değişiklik yapar.



Resim-3 – Katman tanımlama örneği

Ana katman olarak hazırlanacak olan sayfa tasarımı için Resim-3’de gösterilen View -> Shared klasörü içerisinde .cshtml uzantılı dosya oluşturulur. Dosya içerisinde HTML, CSS, JavaScript ile tasarım geliştirilir. Hazırlanan bu ana katmanın görünmesini istediğimiz sayfaların, sayfa tasarımı esnasında Resim-3’de gösterildiği üzere “Layout” komutuyla sayfanın hangi katmana bağlı olacağı belirtilir[4].

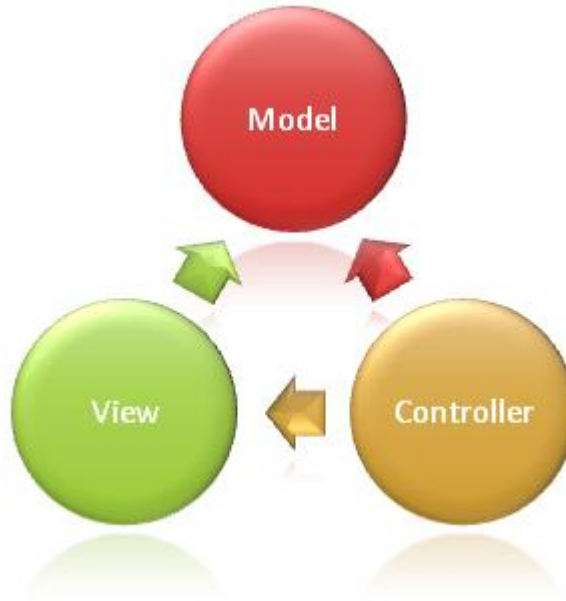
Model

Bir web sayfasında yer alacak olan içeriklerin belirli bir düzen içerisinde arka taraftan sayfanın görüntüleneceği Görünüm bileşenine taşınması için veya Görünüm bileşeninde yer alan

girdilerin işlenmek üzere arka tarafa taşınabilmesi için Model bileşeni kullanılır. Örnek verecek olursak bir bilgi doldurma formu içeren sayfa açılırken, form üzerinde daha önce girilmiş olan girdiler, Model bileşeni ile sayfaya aktarılıp görüntülenir. Bu girdiler üzerinde değişiklik yapıldığında veya yeni girdiler gerçekleştirildiğinde, girdiler sayfa üzerinden Model aracılığıyla arka tarafa taşınarak işlenir. Genel olarak Model, arka taraf ile ön taraf arasında veri alışverişini sağlayan bileşendir[5].

Kontrol

Kontrol bileşeni, uygulamada arka taraf işlemlerinin gerçekleştiği alandır. Bu bileşende Model bileşeni ile getirilen girdiler, veri tabanından getirilen veriler ve Görünüm bileşeninden gönderilen veriler bir araya gelir. Ortaya gelen bu veriler üzerinde gerekli kontroller, işlemler gerçekleştirilir ve veri tabanına kaydedilmesi için ilgili bileşene gönderilir[6].



Resim-4 – MVC deseni

Genel olarak MVC deseni bileşenleri Resim-4’de gösterildiği gibi ilişki içerisinde. Görünüm bileşenine Kontrol bileşeni aracılığıyla Model üzerinden veriler gönderilir. Görünüm üzerindeki veriler Model ile Kontrol bileşenine gönderilir. Kontrol bileşeninde veriler düzenlenir, kontrol edilir ve Görünüm, Model veya veri tabanına kaydedilmesi için Entity çerçevesine gönderilir.

Entity

Entity çerçevesi (Framework) bir Nesne-İlişkisel Eşleme (ORM – Object Relational Mapping) aracıdır. İlişkisel veritabanı ile nesneye yönelik programlama arasında köprü görevi görür. Uygulama içerisindeki nesnelere veri tabanına bağlayan ve veri alışverişini yapan bir çerçevedir. Veri tabanı üzerinde gerçekleştirilen ekleme, silme, güncelleme vb. işlemlerin hepsi, Dil ile Bütünleşik Sorgu (LINQ – Language Integrated Query) aracı ile daha sade ve düzenli şekilde Entity üzerinde gerçekleştirilir.

```
1 namespace Oğrenciİsleri.Entity
2 {
3     public class Personal
4     {
5         4 references
6         public int Id { get; set; }
7         5 references
8         public string Image { get; set; }
9         5 references
10        public string Name { get; set; }
11        5 references
12        public string Task { get; set; }
13        5 references
14        public string Phone { get; set; }
15        5 references
16        public string EMail { get; set; }
17        5 references
18        public string Type { get; set; }
19        5 references
20        public int Sort { get; set; }
21        5 references
22        public string Culture { get; set; }
23    }
24 }
```

Resim-5 – Personel bilgileri için oluşturulmuş bir Entity örneği

Microsoft Visual Studio üzerinde NuGet Paketleri Yönetim aracından Entity paketi kurulduktan sonra Çözümleme Ekranından proje içerisine Entity klasörü açılır ve bu klasör içerisine veri tabanımızın içerisinde oluşturulacak olan tabloların her biri için ayrı ayrı sınıflar açılır. Bu sınıfların içerisine Resim-5’de gösterildiği gibi tabloların içerisinde tutulacak olan veri türleri belirtilir. Entity içerisinde bir Veri Bağlamı (Data Context) sınıfı oluşturulur ve burada veri tabanı adresimiz tanımlanır. Proje çalıştırıldığında Migrasyon Sistemi, Entity içerisindeki sınıflara göre veri tabanını otomatik olarak düzenler.

ASP.Net Identity

ASP.Net Identity bir üyelik sistemi çerçevesidir. Microsoft tarafından Membership çerçevesinin yerine geliştirilen, yönetilebilirlik açısından daha geniş kapsamlı ve özelleştirilebilir niteliktedir. NuGet Paketleri Yönetim aracı üzerinden kolayca kurulabilir. Üyelik sisteminin bulunduğu her türlü siteler için kullanıma uygundur. Öğrenci İşleri Web Uygulamasında genel bir üyelik sistemi olmasa da Yönetim Panelinin güvenliğini sağlamak adına bir yönetici giriş kısmı oluşturulur.

ASP.Net Identity çerçevesini kurduğunuz zaman neredeyse her şey hazır demektir. Kurulum sonrasında tek yapmanız gereken, Identity içerisindeki Veri Bağlamı sınıfına veri tabanı adresinizi belirtmenizdir. Kullanıcı giriş ve çıkış, şifre değiştirme, yeni hesap oluşturma, şifre hatırlatma, yetkilendirme gibi daha birçok üyelik sistemi özellikleri, sayfa tasarımlarıyla beraber hazır olarak gelir. Elbette bunlar üzerinde isteğe bağlı olarak düzenlemeler ve değişiklikler yapılabilir[7].

```
[Authorize(Roles = "Administrator")]  
public class AdministrationController : Controller  
{  
    public IActionResult Index() =>  
        Content("Administrator");  
}
```

Resim-6 – Yetkilendirme çalışması

Üyelik sistemi hazırlarken dikkat edilmesi gereken en önemli konulardan birisi de rol sistemidir. Özellikle Yönetim Paneli kullanılan uygulamalarda rol sistemi hatalı hazırlırsa, kullanıcıların yönetim paneline erişimi gibi büyük bir güvenlik açığı oluşabilir. Bunu önlemek için yönetim panelinin kontrol bileşenine Resim-6’da örneklendirildiği gibi yetkilendirme verilerek sadece yönetici rolüne sahip kullanıcıların erişmesi sağlanır. Bu durumda yönetici sayfasına giriş talebi geldiği zaman Kontrol bileşeni giriş yapan kullanıcının rolünü kontrol ederek erişim verecektir.

Çoklu Dil

Web uygulamalarında çoklu dil sistemi günümüzde kurumlar için olmazsa olmaz bir eklentidir. Eğer uygulamamız sadece yurt içi hizmet göstermeyecekse mutlaka bir çoklu dil

```
services.AddSingleton<LanguageService>(C);  
services.AddLocalization(options => options.ResourcesPath = "Resources");  
  
services.AddMvc()  
    .AddViewLocalization(LanguageViewLocationExpanderFormat.Suffix)  
    .AddDataAnnotationsLocalization(options =>  
    {  
        options.DataAnnotationLocalizerProvider = (type, factory) =>  
        {  
            var assemblyName = new AssemblyName(typeof(ShareResource).GetTypeInfo().Assembly.FullName);  
            return factory.Create("ShareResource", assemblyName.Name);  
        };  
    });  
  
services.Configure<RequestLocalizationOptions>(options =>  
{  
    var supportedCultures = new List<CultureInfo>  
    {  
        new CultureInfo("tr-TR"),  
        new CultureInfo("kk-KZ")  
    };  
  
    options.DefaultRequestCulture = new RequestCulture(culture: "tr-TR", uiCulture: "tr-TR");  
    options.SupportedCultures = supportedCultures;  
    options.SupportedUICultures = supportedCultures;  
  
    options.RequestCultureProviders.Insert(0, new QueryStringRequestCultureProvider());  
});
```

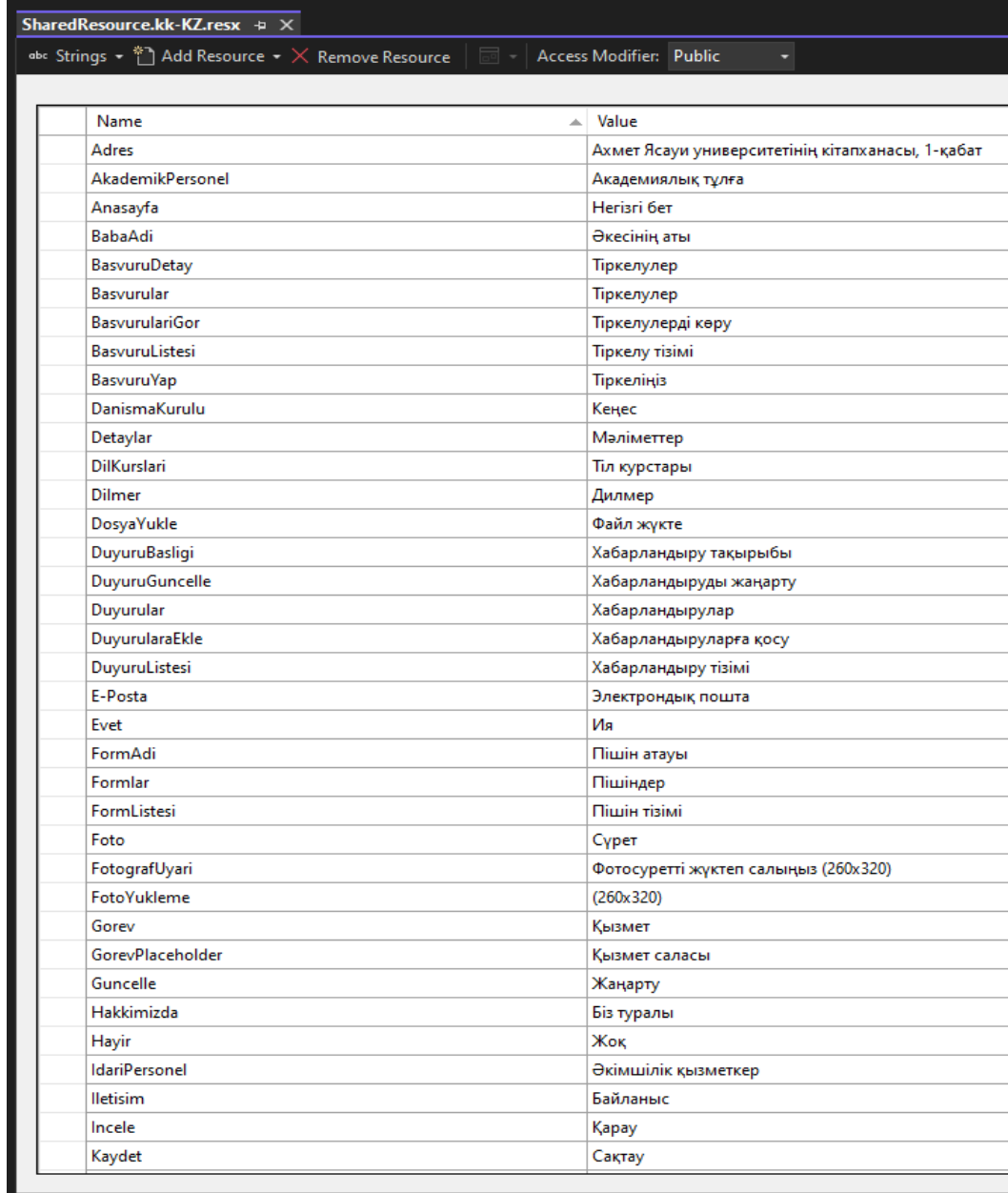
Resim-7 – Çoklu dil sistemi Startup.cs dosyası

sistemi oluşturmamız gerekmektedir. Bu çalışmada, tasarıma en uygun çoklu dil sistemi olan Küreselleşme ve Yerelleşme (Globalization and Localization) sistemi ele alındı.

Microsoft Visual Studio üzerinde ASP.Net MVC projesi oluşturduğunuz zaman Küreselleşme ve Yerelleşme Sisteminin kütüphaneleri, proje içerisinde zaten mevcut olacaktır. Ekstra bir paket kurulumuna gerek yoktur. Öncelikle Model bileşeni içerisinde bir dil servisi oluşturulur. Bu dil servisinde çevirilerin olduğu kaynak dosya belirtilir. Startup.cs dosyası içerisinde ilgili servisler tanımlanır. Öncelikle Model içerisinde oluşturulan dil servisi tanımlanır. Çeviri dosyalarının yer alacağı klasör yolu tanımlanır. Son olarak genel ayarlamaların yapılacağı servis, Resim-7’de gösterildiği gibi oluşturulur. Bu servisin içerisinde, uygulamada kullanılacak diller ve uygulamanın ilk açılışı için varsayılan dil belirtilir[8].

Çevirilerin yapılacağı dosyaları hazırlamak için, servislerde belirtilen dosya yolunda .resx uzantılı kaynak dosyaları oluşturulur. Kaynak dosyasının içerisinde Resim-8’de görüldüğü gibi “Name” ve “Value” adında iki sütun olacaktır. Name sütununa anahtar kelimeler yazılır. Burada

sadece İngilizce karakterler kullanılır ve tek kelime haline boşluksuz yazılır. Value sütununa ise çevirisi yazılır. Bir başka dil için de aynı şekilde kaynak dosyası oluşturulur. Bu sefer “Value” sütununa o dilde çeviriler yazılır. Anahtar kelimeler tüm dillerde aynı olmalıdır. Görünüm bileşeninde sayfa tasarımı yapılırken, Resim-3’de görüldüğü gibi “@language.GetKey(“Keyword”)” kodu ile çeviri sağlanmış olur. Bu sistemde sayfa o anda hangi dilde görüntüleniyorsa sistem o dilin kaynak dosyasına giderek anahtar kelimenin karşısındaki çeviriyi ekrana gösterir. Bu da yeni dil eklemek gerektiği durumlarda işlemi çok kolaylaştırır. Projeye üçüncü bir dil eklemek istendiği takdirde, sadece kaynak dosyasından bir tane daha oluşturulması ve anahtar kelimelerin karşısına çevirilerinin yazılması yeterli olacaktır.



Name	Value
Adres	Ахмет Ясауи университетінің кітапханасы, 1-қабат
AkademikPersonel	Академиялық тұлға
Anasayfa	Негізгі бет
BabaAdi	Әкесінің аты
BasvuruDetay	Тіркелулер
Basvurular	Тіркелулер
BasvurulariGor	Тіркелулерді көру
BasvuruListesi	Тіркелу тізімі
BasvuruYap	Тіркеліңіз
DanismaKurulu	Кеңес
Detaylar	Мәліметтер
DilKurslari	Тіл курстары
Dilmer	Дилмер
DosyaYukle	Файл жүкте
DuyuruBasligi	Хабарландыру тақырыбы
DuyuruGuncelle	Хабарландыруды жаңарту
Duyurular	Хабарландырулар
DuyurularaEkle	Хабарландыруларға қосу
DuyuruListesi	Хабарландыру тізімі
E-Posta	Электрондық пошта
Evet	Ия
FormAdi	Пішін атауы
Formlar	Пішіндер
FormListesi	Пішін тізімі
Foto	Сурет
FotografUyari	Фотосуретті жүктеп салыңыз (260x320)
FotoYukleme	(260x320)
Gorev	Қызмет
GorevPlaceholder	Қызмет саласы
Guncelle	Жаңарту
Hakkimizda	Біз туралы
Hayir	Жоқ
IdariPersonel	Әкімшілік қызметкер
Iletisim	Байланыс
Incele	Қарау
Kaydet	Сақтау

Resim-8 – Çoklu dil kaynak dosyası

Sonuç

Bu çalışmada ASP.Net MVC temelli bir web uygulamasının geliştirilmesini, projenin kurulum ve programlanma süreçlerini incelendi. Projenin Microsoft Visual Studio üzerinde kurulumu yapıldı. Bir web uygulamasında kullanıcılar tarafından ilk değerlendirilen kısım olan

sayfa tasarımlarının nasıl hazırlanacağı gösterildi. Ana katman ve sayfalar ayrımı, bunların nasıl tanımlanacağı belirtildi. Görünüm bileşeninin tamamlanmasının ardından arka taraf programlamasının yapılacağı Kontrol ve Model bileşenleri incelendi. Model bileşeninde, Görünüm sayfasında işlenecek olan veriler için sınıflar oluşturuldu. Kontrol bileşeninin kodların düzenlendiği, gerekli mantıksal işlemlerin ve düzenlemelerin yapıldığı yer olduğu kavrandı. Kontrol bileşeni ile Görünüm bileşenine Model aracılığıyla veri gönderimini ve veri tabanına verileri işlemenin, Entity çerçevesini ve LINQ sorgularını kullanarak Kontrol bileşeninden yönlendirmeye nasıl yapılacağı incelendi. Entity çerçevesinin veri tabanı işlemlerinin LINQ sorgularıyla çok daha kullanışlı ve kolay olarak nasıl gerçekleştirildiği incelendi. Yönetim paneli kullanımı için üyelik ve rol sistemi kullanımının önemi kavrandı. Üyelik ve rol sistemi için Microsoft tarafından geliştirilen Identity çerçevesi tanımlandı. Identity çerçevesinin kurulum ve kullanım yöntemi incelendi. Proje, bütün işlemlerin sonunda artık yayına ve kullanıma hazır hale geldi. Bu alanda çalışmalar yapacak olanlar, kendilerine göre tasarlayarak düzenlemeler yapabilirler.

Standart web programlama sistemlerinde (HTML, CSS, JavaScript) ön taraf sayfa tasarımları için bir proje, arka taraf programlaması için bir proje oluşturulur. Ön taraf ve arka taraf projeleri birbirine entegre edilerek çalışması sağlanır. Bu sistem dışarıdan bakıldığında gayet kolay ve daha az karmaşık görünse de aslında öyle değildir. ASP.Net, MVC, Entity Framework, Identity Framework, Microsoft SQL, HTML, CSS, JavaScript, C#, LINQ gibi birden fazla teknoloji kullanılarak oluşturulan bu web uygulamasında, bu kadar farklı teknoloji ile çalışma yapmak zor görünse de sistem kurulduktan sonra tasarım ve kodlama sürecinde, çoklu dil sistemi kurulumunda, arka taraf ve ön taraf iletişimde programcılara büyük kolaylıklar sağladığı gösterilmiştir.

KAYNAKLAR

1. Akçay M., Kasım Ö., Taşdelen Z. ASP.NET Ve MVC Temelli Esnek (Responsive) Web Uygulaması // ESTUDAM Bilişim Dergisi. - 2021. - Cilt 2, Sayı 1. – S. 34-41.
2. Ergin İ., Akseki B. Lisanüstü Eğitimde Kullanılan Öğrenci Bilgi Sistemi // Eğitim ve Öğretim Araştırmaları Dergisi. – 2012. – Cilt 1 Sayı 2. – S. 364-380.
3. Uçar E., Altunsöğüt Ö. ASP.Net Teknolojisini Kullanarak Bir Satın Alma Portalı Uygulaması Geliştirilmesi // Trakya Üniversitesi Fen Bilimleri Dergisi. -2009. - 10(2). – S. 119-126.
4. Views in ASP.NET Core MVC (2022). Web: <https://learn.microsoft.com/en-us/aspnet/core/mvc/views/overview?view=aspnetcore-7.0>
5. Part 4, add a model to an ASP.NET Core MVC app (2023). Web: <https://learn.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/adding-model?view=aspnetcore-7.0&tabs=visual-studio>
6. ASP.NET MVC Controller Overview (C#) (2022). Web: <https://learn.microsoft.com/en-us/aspnet/mvc/overview/older-versions-1/controllers-and-routing/aspnet-mvc-controllers-overview-cs>
7. Role-based authorization in ASP.NET Core (2022). Web: <https://learn.microsoft.com/en-us/aspnet/core/security/authorization/roles?view=aspnetcore-7.0>
8. Globalization and localization in ASP.NET Core (2023). Web: <https://learn.microsoft.com/en-us/aspnet/core/fundamentals/localization?view=aspnetcore-7.0>

REFERENCES

1. Akçay M., Kasım Ö., Taşdelen Z. ASP.NET Ve MVC Temelli Esnek (Responsive) Web Uygulaması // Journal of ESTUDAM Information. - 2021. - Volume 2, Issue 1. – P. 34-41.
2. Ergin I., Akseki B. Student Information System Used In Graduate Education // Journal of Research in Education and Teaching. – 2012. – Volume 1 Issue 2. – P. 364-380.

3. Ucar E., Altunsogut O. Development of an E-Purchase Portal Application Using ASP.NET Technology // Trakya University Journal of Science. -2009. - 10(2). – P. 119-126.
4. Views in ASP.NET Core MVC (2022). Web: <https://learn.microsoft.com/en-us/aspnet/core/mvc/views/overview?view=aspnetcore-7.0>
5. Part 4, add a model to an ASP.NET Core MVC app (2023). Web: <https://learn.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/adding-model?view=aspnetcore-7.0&tabs=visual-studio>
6. ASP.NET MVC Controller Overview (C#) (2022). Web: <https://learn.microsoft.com/en-us/aspnet/mvc/overview/older-versions-1/controllers-and-routing/aspnet-mvc-controllers-overview-cs>
7. Role-based authorization in ASP.NET Core (2022). Web: <https://learn.microsoft.com/en-us/aspnet/core/security/authorization/roles?view=aspnetcore-7.0>
8. Globalization and localization in ASP.NET Core (2023). Web: <https://learn.microsoft.com/en-us/aspnet/core/fundamentals/localization?view=aspnetcore-7.0>