

**T.C.**

**DÜZCE ÜNİVERSİTESİ**

**TEKNOLOJİ FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**E-Ticaret Web Site Projesi**

**UMUT DEMİRAY**

**152114016**

**Bitirme Tezi**

**Proje Danışmanı**

Dr. Öğr. Üyesi Serdar BİROĞUL

**DÜZCE, 2020**

T.C.

DÜZCE ÜNİVERSİTESİ

**TEKNOLOJİ FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**E-Ticaret Sitesi Projesi**

Umut Demiray tarafından hazırlanan bitirme tezi çalışması aşağıdaki jüri tarafından Düzce Üniversitesi Teknoloji Fakültesi Bilgisayar Mühendisliği Bölümü’nde **LİSANS BİTİRME TEZİ** olarak kabul edilmiştir.

**Tez Danışmanı**

Dr. Öğr. Üyesi Serdar BİROĞUL

Düzce Üniversitesi

**Jüri Üyeleri**

Dr. Öğr. Üyesi Serdar BİROĞUL

Düzce Üniversitesi \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Düzce Üniversitesi \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Düzce Üniversitesi \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tez Savunma Tarihi:

**BEYAN**

Bu tez çalışmasının kendi çalışmam olduğunu, tezin planlanmasından yazımına kadar bütün aşamalarda etik dışı davranışımın olmadığını, bu tezdeki bütün bilgileri akademik ve etik kurallar içinde elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara kaynak gösterdiğimi ve bu kaynakları da kaynaklar listesine aldığımı, yine bu tezin çalışılması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını beyan ederim.

25 Mayıs 2019

Umut Demiray

**TEŞEKKÜR**

Lisans öğrenimimde ve bu tezin hazırlanmasında gösterdiği her türlü destek ve yardımdan dolayı çok değerli hocam Dr. Öğr. Üyesi Serdar BİROĞUL’a en içten dileklerimle teşekkür ederim.

Gerek lisans ders sürecinde ve gerekse de tez çalışmam boyunca benden desteğini esirgemeyen, çalışmalarımda beni sabırla destekleyen annem Mutlu Demiray, kız kardeşim Duygu Demiray, babam Şaban Demiray, erkek kardeşim Ömer Demiray, kız arkadaşım Kübra Nur Cin, yakın arkadaşlarım Ahmet Sercan Niyazi Üzel, Hakan Ergovan, Döne Deniz ve Özgür Özbek’e sonsuz teşekkürlerimi sunarım.

**24 Mayıs 2020 Umut DEMİRAY**

**ÖZET  
E-Ticaret Web Site Projesi**

Umut DEMİRAY

Düzce Üniversitesi

Teknoloji Fakültesi

Bilgisayar Mühendisliği Bölümü

Bitirme Tezi

Danışman: Dr. Öğr. Üyesi Serdar BİROĞUL

Mayıs 2020, 45 sayfa

**C# programla dilinde önemli bir yer edinmiş, .Net Core web projelerinde kulanılan bir yapıdır. Bu tez projesinde bu teknoloji üzerine geliştirme yapılmıştır. İç yüzünde, birçok design patern ön planda olsa da bunlardan yapımıza en uygun olanı olan Repository Design Partern koda adapte edilmiştir. Orm teknolojisi olarak Entityframeworkten yararlanılmıştır. Web arayüz programlama açısından uygun olan Mvc yapısı projede kullanılarak bir e-ticaret sistemi geliştirilmiştir.**

**Anahtar Sözcükler** :C#, .Net Core, Entityframework, uygulama, geliştirme

**ABSTRACT  
E-Commerce Web Site Project**

Umut DEMİRAY

Düzce University

Technology Faculty, Department of Computer Engineering

Undergraduate Graduation Thesis

Supervisor: Dr. Öğr. Ü. Serdar BİROĞUL

Mayıs 2020, 45 sayfa

**It is a structure which has gained an important place in the language with C # program and used in .Net Core web projects. In this thesis project, development has been made on this technology. On the inside, Repository Design Partern, which is the most suitable for our construction, has been adapted to the code, although many design patterns are in the foreground. Entityframework was used as Orm technology. An e-commerce system has been developed by using the Mvc structure, which is suitable for web interface programming, in the project.**

**Keywords:** C#, .Net Core, Entityframework, application, development

**İÇİNDEKİLER**

ÖZET ………………………………………………………………...5

ABSTRACT…………………………………………………………..6

ŞEKİL LİSTESİ……………………………………………………...8

1.GİRİŞ .………………………………………………………………9

2.MATERYAL ve YÖNTEM ………………………….……………10

2.1.MS Visual Studio ………………………………………………10

2.2.SQL Server Data Tools (SSDT) ……………………………….10

2.3.Repository Design Pattern …………………………………….11

2.4.Entity Framework ……………………………………………..11

2.5.Entity Framework LINQ ……………………………………...14

2.6.ASP.Net Migration …………………………………………….14

2.7.ASP.Net Identity ……………………………………………….15

2.8.ASP.Net MVC ………………………………………………….16

2.9.SendGrid ……………………………………………………….17

2.10.iyzico …………………………………………………………..17

3.UYGULAMA ve TEST ……………………………………………18

4.SONUÇ ……………………………………………………………..42

KAYNAKLAR ……………………………………………………….43

ÖZGEÇMİŞ ………………………………………………………….46

ŞEKİL LİSTESİ

**Sayfa No**

Şekil 1 Ana Sayfa ….……………………………………………………………………...18

Şekil 2 Ürün Sayfası ……………………………………………………………………....19

Şekil 3 Tüm Ürünler Kategorisi ……………………………………………………….….20

Şekil 4 IPhone Telefon Kategorisi .…………………………………………………….….20

Şekil 5 Apple Aksesuar Kategorisi ….………………………………………………….....21

Şekil 6 IPad Kategorisi …………………………………………………………………....21

Şekil 7 Apple Watch Kategorisi …………………………………………………………..22

Şekil 8 Mac Bilgisayar Kategorisi .……………………………………………….……….22

Şekil 9 Ürün Detay Sayfası ………………………………………………………………..23

Şekil 10 Kullanıcı Giriş Sayfası ……….…………………………………………………..23

Şekil 11 Üye Ol Sayfası ..…………...……………………………………………………..24

Şekil 12 E-Mail Onay İsteği ………………..……………………………………………..25

Şekil 13 Kullanıcı Bulunamadı Hatası ……..……………………………………………..25

Şekil 14 Kullanıcı Onay Maili ...…………………………………………………………..25

Şekil 15 Kullanıcı Onaylandı ……………………………………………………………...26

Şekil 16 Kullanıcı Ana Sayfa ……………………………………………………………...26

Şekil 17 Kullanıcı Alışveriş Sepeti Boş …………………………………………………...27

Şekil 18 Kullanıcı Siparişleri Boş …………………………..……………………………..27

Şekil 19 Kullanıcı Alışveriş Sepeti Dolu ………………………………………………….28  
Şekil 20 Kullanıcı Ödeme Sayfası ………….……………………………………………..29

Şekil 21 Kullanıcı Ödeme Sayfası Gerekli Veriler ………………………………………..30

Şekil 22 Kullanıcı Siparişleri Dolu ………………………………………………………..31

Şekil 23 Şifremi Unuttum Sayfası ….……………………………………………………..31

Şekil 24 Şifremi Unuttum İsteği Alındı ….………………………………………………..32

Şekil 25 Şifremi Unuttum Maili …………………………………………………………..32

Şekil 26 Şifre Sıfırlama Sayfası ….………………………………………………………..32

Şekil 27 Admin Ana Sayfa ………………………………………………………………..33

Şekil 28 Admin Kategori İşlemleri Sayfası …...…………………………………………..34

Şekil 29 Admin Kategori Ekle Sayfası …………..………………………………………..34

Şekil 30 Admin Kategori Eklendi .………………………………………………………...35

Şekil 31 Admin Kategori Güncelle Sayfası …..…………………………………………...35

Şekil 32 Ürüne Kategori Eklenmesi ..……………………………………………………...36

Şekil 33 Admin Ürüne Kategori Eklendi …..……………………………………………...36

Şekil 34 Admin Kategori Menü Kontrol …………………………………………………..37

Şekil 35 Admin Kategori Sil ……………………………………………………………...38

Şekil 36 Admin Ürün İşlemleri ……….…………………………………………………...38

Şekil 37 Admin Ürün Ekle ………………………………………………………………...39

Şekil 38 Admin Ürün Eklendi ...…………………………………………………………...39

Şekil 39 Admin Ürün Güncelle ….………………………………………………………...40

Şekil 40 Admin Ürün Güncellendi ………………………………………………………...41

# **GİRİŞ**

Bu projede C#, Class Library yapıları ile Repository Design Patern yapısı üzerine katmanlı bir yapı kurularak ve Web Application kullanılarak ara yüz kısmında MVC yapısı üzerine kurulu bir web sitesi geliştirilmiştir. Bu proje herhangi bir ürünü satan bir iş yerinin, e ticaret sitesi olabilecek yapıda değiştirilebilecek yapıda yazılmıştır. Fakat bir senaryo üzerinde geliştirilmesi gerektiği için bir mini Apple marketinin sitesi olarak tasarlanmıştır.

Web site giren işinin üye olmadan ürünleri görüp bakabileceği bir şekilde tasarlanmıştır. Üye olup, e posta onayını gerçekleştiren kullanıcılar sepet ve siparişlerim kısımlarına ulaşabilmekteler ve ürün sipariş edebilmektedirler. Kullanıcılar şifrelerini unuttularsa giriş kısmından şifremi unuttum diyerek e postalarına gelen link ile şifrelerini değiştirebilmektedirler.

Web projesinin kontrol paneli için yönetici girişi yapıp ürün ve kategori ekleme işlemleri de yapılmaktadır. İstenilen ürün gerekli alanlar doldurulduktan sonra eklenebilmektedir. Daha sonradan ürünün silinmesi güncellenmesi ya da bir kategoriye atanma işlemi yapılabilmektedir. İstenilen kategori oluşturulup ya da silinip, içine ürün ataması yapılıp buradan güncellenebilmekte ve kategorilerden ürün siline bilinmektedir.

# **MATERYAL VE YÖNTEM**

## **2.1** **MS Visual Studio**

Visual Studio, birçok programlama dilini kullanarak program, uygulama ya da web sitesi yapabileceğiniz bir IDE yani entegre geliştirme ortamıdır. Microsoft Windows için bilgisayar programları, web siteleri, web uygulamaları, web hizmetleri ve mobil uygulamalar geliştirmek için kullanılır.  
 Visual Studio programı içerisinde yalnızca Microsoft Windows tarafından desteklenen yerel kodlar kullanılmaktadır. Visual Studio, Windows API, Windows Forms, Windows Presentation Foundation, Windows Store ve Microsoft Silverlight gibi Microsoft yazılım geliştirme platformlarını kullanır. Hem yerel kod hem de yönetilen kod üretebilir.

Visual Studio, IntelliSense'i (kod tamamlama bileşeni) ve kod yeniden düzenleme işlemini destekleyen bir kod düzenleyici içerir. Entegre hata ayıklayıcı, hem kaynak düzeyinde hata ayıklayıcı hem de makine düzeyinde hata ayıklayıcı olarak çalışır. Diğer yerleşik araçlar arasında bir kod profili oluşturucu, GUI uygulamaları oluşturmak için form tasarımcısı, web tasarımcısı, sınıf tasarımcısı ve veritabanı şeması tasarımcısı bulunur. Neredeyse her düzeyde işlevselliği artıran eklentileri kabul eder.

## **2.2 SQL Server Data Tools (SSDT)**

SQL Server Data Tools, SQL Server 2012 ile birlikte gelen yeni bir tooldur. SSDT sayesinde yazılım geliştiriciler SQL Server Management Studio kullanma ihtiyacı olmadan, database oluşturma, tablo, view, procedure gibi nesleleri oluşturma imkanına sahip olmaktadır.

SQL Server Data Tools, Local DB özelliği ile geliştirme yapılan makinede **SQL Server** kurulumu yapılmamış olsa da sanki SQL Server servislerinin kurulumu yapılmış gibi veritabanı oluşturmaya ve geliştirmeye imkân sağlar. **Local DB** özelliği sayesinde yazılım geliştiriciler makinalarına **SQL Server** kurulumu yapmak zorunda kalmazlar.

## **2.3 Repository Design Patern**

Repository Design Pattern, veritabanı sorumluluğunu üstlenen sınıfı tasarlarken bir standart üzerine oturtmayı hedefleyen ve Entity Framework gibi ORM(Object Relational Mapping) araçlarıyla kombine edilerek sorgusal anlamda az sayıda operatif metotla yüksek seviyede veri erişim imkanı sağlayan bir strateji üzerine kurulu tasarım desenidir.

Yazılım uygulamaları genellikle yoğun bir şekilde veritabanı işlemleri gerçekleştiren ekosisteme sahip yapılar olduğundan dolayı, ilgili uygulamanın her bir noktasında gerekli veritabanı işlemlerini tekrar tekrar yazmak yerine bu işlemleri tekrar kullanılabilirlik prensibi çerçevesinde daha pratik bir şekilde tek seferde yapmamızı sağlayan bir yapılanma geliştirmemiz gerekecektir. İşte bu yapılanma Repository sınıfı olacaktır.

Repository sınıfı, içerisinde generic yapılanmalarla geliştirilen temel operasyonel veritabanı metotlarını barındıran bir sınıftır. Yukarıdaki diyagramdan da görüldüğü üzere sorgu generate etme ve genellikle ORM araçlarıyla kombine edilerek veri eşleştirme sorumluluğunu üstlenmektedir. Bu yeteneklerini ana business sorumluluğunu yüklenen sınıflara kalıtım yoluyla aktarmakta ve bu şekilde kazandırmaktadır.

Böylece Repository’den türeyen her bir sınıfın instance’i üzerinden de veritabanı işlemlerini yürütebilirsiniz.

## **2.4 Entity Framework**

Entity Framework ORM(Object Relational Mapping) araçlarından biridir. ORM nedir dersek: İlişkisel veritabanı ile nesneye yönelik programlama(OOP) arasında bir köprü görevi gören araçtır. Bu köprü, ilişkisel veri tabanındaki bilgilerimizi yönetmek için nesne modellerimizi kullandığımız bir yapıdır. Kısaca veri tabanına bizim nesnelerimizi bağlayan ve bizim için veri alışverişini yapan **Microsoft**tarafından geliştirilmiş bir framework’tür.

Entity Framework(EF) Ado.Net altyapısını kullanmaktadır. İçerisinde UnitOfWork design pattern’nini barındırmaktadır. Nedir bu Ado.Net nedir Bu UnitOfWork dersek Ado.Net kısaca Microsoft tarafından yazılmış veri erişim teknolojisidir. Günümüzde hala kullanılmaktadır ancak yavaş yavaş yerini ORM teknolojilerine bırakmaktadır çünkü veritabanı işlemleri için Ado.Net kullanırsanız birçok işi kendiniz yapmanız gerekmektedir.

Örnek vermek gerekirse ekleme, silme, güncelleme ve diğer sorgular için string formatında sorgu cümleciklerini oluşturmanız gerekecektir buda büyük bir projede ayrı bir yük getirmektedir. UnitOfWork ise yapılan her değişikliğin anlık olarak veri tabanına yansıması yerine, bu işlemlerin toplu halde tek bir kanal üzerinden gerçekleşmesini sağlar.

Burada çok önemli bir kavram olan context sınıfımı karşımıza çıkmaktadır. Context sınıfı genel anlamda veri tabanı işlemlerinin halledildiği sınıftır. Context sınıfı, dbcontext sınıfından türemiştir ve Entitiy Framework için çok önemli bir class’dır.

Context Sınıfı İşlevleri:

* Veritabanı bağlantılarını  yönetir.
* Model ve ilişkileri ayarlar.
* Veritabanı sorgulama.
* Değişikleri yönetir.
* Caching
* Transaction yönetimi
* Object Materialization (Veri tabanından aldığı verileri entity nesnelerine dönüştürür).

Entity Frameworke geri dönecek olursak bu frameworkü projemize dahil etmek için yapmamız gereken komut ekranında proje dizinimize “Install-Package EntityFramework” yazmaktır ya da projemize nuget paket yöneticisinden EntityFramework paketini aratıp projemize kurmamızdır.

Entity Framework ile 4 farklı yöntem kullanarak proje geliştirebilirsiniz. Bu yöntemler;

* Model First (Yeni Veritabanı Oluşturma Visual Studio İle)
* Database First (Var Olan Veri tabanını Kullanma)
* Code First (Yeni Veritabanı Kod Yazarak)
* Code First(Var Olan Veri tabanını Kullanma)

Biz projemizde Code First yaklaşımını kullandık. Bundan dolayı buna da değinmek istiyorum.

**Code First:** Bu yöntem classlarımızı Visual Studio ortamında oluşturmaya başlayarak gerçekleştirdiğimiz bir yöntemdir. Veri tabanımız bu classlardan türetilmektedir. Burada Mapping işlemleri yazılımcı tarafından classlar oluşturulurken **Attribute**’lar sayesinde yapılabilmektedir. Bu arada Mapping işlemi kısaca tablolarımızdaki kısıtlarımızı belirlediğimiz olaydır.

Son olarak Entity Framework’ün avantaj ve dezavantajlarına değinmek istiyorum. Kullanma nedenimizin daha iyi anlaşılması açısından.

Avantajları:

* OOP olarak kod geliştirmemize olanak tanır.
* Hiçbir SQL bilgisi olmayan bir kimse veritabanı işlemlerini **EF**ile gerçekleştirebilir.
* Herhangi bir veri tabanına bağımlılık yoktur. Oracle, MS SQL ile kullanılabilir.
* Code First sayesinde projenizi veri tabanınızı taşıma gereği duymadan istediğiniz yerde oluşturabilirsiniz. Buda projelerinize büyük bir esneklik sağlamaktadır.
* Yazılım geliştirme zamanını azaltır.
* Yazılım geliştirme maliyetini azaltır.

Dezavantajları:

* En büyük sorunumuz performans. Ado.Net gibi hızlı bir performansı yoktur. Tabi bu yavaş olduğu anlamına gelmez.
* Veri tabanından veri alış-verişi yapılacağı zaman kontrol bizde değil Entity Frameworktedir. Yani arka planda veritabanı işlemleri için kendisi sorgular oluşturmaktadır. Basit bir veri işlemi için karmaşık bir sorgu gönderebilmektedir. Bu Sorguları SQL Server Profiler ile görüntüleyebilmektesiniz.
* Schema’da herhangi bir değişiklik yapıldığı zaman EF çalışmayacaktır. Sizin bu Schema’yı solution’da güncellemeniz gerekmektedir.

## **2.5 Entity Framework LINQ**

Program ve veri arasındaki ilişkiyi başka bir boyuta taşıyan devrimsel bir programlama metodolojisidir. LINQ, bir programlama ara yüzü sunar. C# diline getirdiği eklentilerle SQL benzeri tek bir söz dizimi ile farklı tiplerdeki verilerin sorgulanmasını sağlar. C# ile sorgu yazmak, tip güvenli çalışma, ifadelerin otomatik olarak tamamlanması ve IntelliSense gibi özelliklerle geliştiricinin üretkenliğini artırmayı sağlar.

ADO.NET Entity Framework tarafından oluşturulan varlıkları sorgulamak. Bu sayılan LINQ uygulamaları aynı söz dizimini kullanır, ancak farklı veri kümelerini hedeflemektedir.

**Peki,** LINQ neden kullanmalıyız. LINQ kodlarımız içerisinde, farklı dil kullanmadan, çeşitli ortamlardaki verileri sorgulayıp, filtrelemek için kullanılır. LINQ öncesinde daha çok kod yazarak yapılan işlemler artık daha az eforla daha kısa zamanda yapılabiliyor. Bu durum da üretkenliği arttırıyor.

Çoğu SQL komutlarına benzer ya da aynı olan C# anahtar kelimelerinden oluşan ifadeler ile .NET koleksiyonlarına LINQ sorguları yazılır. Bu anahtar kelimeler, LINQ Standart Sorgu Operatörleri olarak anılır(LINQ Standart Query Operators).

## **2.6 ASP.Net Migration**

Her projede olduğu gibi Code First yaklaşımıyla inşa ettiğimiz projemizde de güncellenmesi gerekecek noktalar kaçınılmaz olacaktır. Gerek kodlarda, gerekse de programın akışını sağlayan algoritmada düzenlenme olacaksa da bu yazımızı asıl ilgilendiren veri tabanının güncellenmesi gereken durumlardır. Biliyoruz ki, Code First deseni uygulanan bir projede veri tabanına direkt olarak müdahale etmek oldukça sakıncalıdır. Haliyle yapacağımız ekleme, çıkarma veyahut güncelleme durumlarını tabloları temsil eden classlar üzerinde gerçekleştirmeli ve generate etmeliyiz.

Gerçekleştirmemiz gereken işlem çok kolay komut satırında projemizin repository dizinine gelip öncelikle dotnet ef migrations add {isim}(isim: migration’a verilmek istenen isim) komutu çalıştırarak migrations klasöründe bir migration oluşmasını sağlarız. Bu migration ile database’mizi güncellemek için dotnet ef database update yazmamız yeterli olucaktır. Komut satırı kullanmak istemeyenler için bu işi kolaylaştıracak olan paket nuget paket yöneticisinde bulunur buradan indirerek otomatikleştirebiliriz migration işlemini.

## **2.7 ASP.Net Identity**

Asp.NET Core Identity; üyelerin, login(giriş), out(çıkış), yetkilendirme, token, şifre hatırlatma vs. tüm işlemleri hızlı bir şekilde gerçekleştirmemizi sağlayan ve bunların dışında önceki nesillere nazaran herhangi bir kısıtlaması olmaksızın uygulamalarımızı destekleyen çağdaş bir üyelik sistemidir. Çağdaştır çünkü günnümüzdeki üçüncü kaynaklardan sağlanan modern oturum süreçlerini(Facebook Login, Google Login, Twitter Login vs.) desteklemekte ve tüm inşayı hızlı bir şekilde gerçekleştirmektedir.

Asp.NET Core Identity kütüphanesini diğerlerinden ayıran en büyük özelliği esnek olmasıdır. Yani sağladığı özelliklerden tarafımızca beğenilmeyen yahut daha iyisi düşünülebilen sistemler custom olarak geliştirilebilmekte ve çok rahat bir şekilde uygulamaya identity üzerinden entegre edilebilmektedirler. İşte bu özelliği de büyük fark yaratmakta ve bizleri gelişmenin ileriki noktalarında olası uyumsuzluk şüphelerinden kurtarmaktadır. Keza bu özellik, önceki Membership üyelik sisteminden Identity’i ayıran en büyük farkı yaratmaktadır.

Asp.NET Core Identity’i bu projede temel sevide kullandığımız için giriş seviyesindeki belli temel kavramların bu kısmını ele almakta fayda var. Her şeyden önce üyelik sistemi(Asp.NET Core Identity) deyince şu iki kavramın izahını yapabilmek gerekmektedir;

* **Authentication:**  
  Kullanıcının sistem tarafında tanımlanan kişi olup olmadığının doğrulanmasıdır. Bir başka deyişle kimlik doğrulamasıdır.
* **Authorization:**  
  Sistemde doğrulanan kullanıcının hangi sınırlara sahip olduğunun belirlenmesidir. Buna da kimlik yetkilendirmesi diyebiliriz.

Peki, bir web uygulamasında Asp.NET Identity kütüphanesinin kullanılabilmesi için yapılması gereken ayarlar nelerdir. Bir Asp.NET Core uygulamasında Identity üyelik sisteminin kullanılabilmesi için “Microsoft.AspNetCore.Identity” kütüphanesinin ilgili projeye entegre edilmiş olması yeterlidir. Bunun için bizlerin ekstradan bir işlem yapmasına gerek yoktur. Çünkü ilgili kütüphane varsayılan olarak Core projelerinde dahili bir şekilde gelmektedir.

## **2.8 ASP.Net MVC**

MVC aslında design patterndir. Design pattern yazılım geliştirme sürecinde kodların yeniden kullanılabilirliğini sağlama, ortaya çıkan yazılımsal sorunları çözme ve bu çözümlerin standartlar haline getirip yeniden kullanılabilir şekilde kodlanması sonucu oluşan sisteme denir. İşte MVC’de oluşturduğumuz yazılımları belirli standartlar haline getirip yazılımlarımızda karşılaştığımız sorunlara kolaylıkla çözüm getirebilmemizi sağlıyor. Model, View ve Controller kelimelerinin baş harflerinden oluşan MVC ilk olarak 1979 yılında Trygve Reenskaug tarafından ortaya çıkarılmıştır. Daha sonra Microsoft tarafından geliştirilerek Asp.Net platformuna entegre edilmiştir.

MVC Model, View ve Controller olmak üzere üç bileşenden oluşmaktadır. Ve bu üç bileşen birbirinden tamamen bağımsız bir şekilde çalışmaktadır. Özellikle büyük çaplı projelerde proje yönetimini kolaylaştırdığı için sıklıkla kullanılmaktadır.

Şimdi bu katmalara bir göz atalım;

**Model:**

Veri işlemleri bu katmanda gerçekleşir. Veri tabanına erişim, sınıflar (class), veritabanı ilişkileri gibi veri ile ilgili bütün işlemler bu katmanda gerçekleşir.

**View:**

Bu katman son kullanıcının gördüğü katmandır. Bu katmanda Html, css ve Javascript vb. programlama dilleri ile görsel geliştirmeler yapılır.

**Controller:**

Model ve View arasındaki iletişimi sağlayan katmandır. Aynı zamanda kullanıcıdan gelen ilk isteği karşılayan katmandır. Controller katmanında fonksiyonlar, metodlar bu katmandan çağrılarak kullanılır.

Sonuç olarak MVC yapısı işlerimizi kolaylaştırıp daha basit ve kolay bir şekilde kodlamamıza yardımcı olmaktadır.

**2.9 SendGrid**

SendGrid bir e-mail sistemidir. Geliştiricilere ve şirketlere projelerinde kullanmaları için ücretli ya da ücretsiz olarak destek sağlayan girişimcilik projesi olarak başlayan start-up şuan dünyada birçok projede kullanılmakla beraber birçok da benzeri olan bir sistemdir.

Projemize Visual Studio’nun paket yöneticisi olan nuget paket yöneticisinden indirip kurduktan sonra siteden bize sağlanan api-key ile projemize adapte edip kullanılabilmektedir.

**2.10 iyzico**

iyzico; kendi değişleri ile e-ticaret şirketleri, bireysel kullanıcılar ve pazar yerleri için sanal pos ve güvenli ödeme çözümleri sunan finans teknolojileri şirketidir.

En can alıcı özelliği token ödeme sistemi. Hayatımızı epey kolaylaştıran bir sistem. Kart bilgilerinizin bir güvenlik anahtarı ile birlikte ödeme sağlayıcı kuruluşta muhafaza edilmesi ve satıcı kuruluşun siz kart bilgilerinizi tekrar girme ihtiyacı duymadan ya da düzenli ödeme gerektiren şeylerde aracı firmaya sizden ödeme tahsil ederek iletebilmesini sağlar. Bu token dediğim şeyde aslında bu güvenlik anahtarıdır.

En büyük artısı, son kullanıcı kart bilgilerinin güvende olduğunu bilir. Ufak bir token sayesinde kullanıcı ödeme yap butonuna tıklayarak uğraşmadan ödeme yapar hatta belki siz otomatik bile çekebilirsiniz…

Projeye dahil edilmesi ise nuget paket yöneticisinden şirketin kendi ürettiği asp.net uyumlu olan iyzico paketini ya da asp.net core için geliştiricilerin oluşturduğu iyziPay Core paketini indirerek sitedeki örneklerinden birebir alarak iyzico’nun sizin için ürettiği api key ve security keyi değiştirerek kullanabilirsiniz.

# 

# **UYGULAMA ve TEST**

**Uygulamanın Çalıştırılması İçin;**

Öncelikle cmd’yi açıp dotnet --info yazıp .Net Core SDK versiyonunun 3.1.2xx veya daha üstü bir versiyonda olduğunu kontrol ediyoruz. Eğer versiyon daha düşük ise proje çalışmaz. Bu gibi bir sorunla karşılaşırsak yapmamız gerekenler şunlardır;

* <https://download.visualstudio.microsoft.com/download/pr/73718445-e2bd-40b7-b698-e8a9ac65f4e5/0816570f697c4e8f1b53ecfb33eaed7f/dotnet-sdk-3.1.300-win-x64.exe>
* <https://download.visualstudio.microsoft.com/download/pr/b0ae36b7-15a2-4ed5-8055-74946bee46cf/b56a70de4576bb3c3af8e15af19b4d70/aspnetcore-runtime-3.1.4-win-x64.zip>

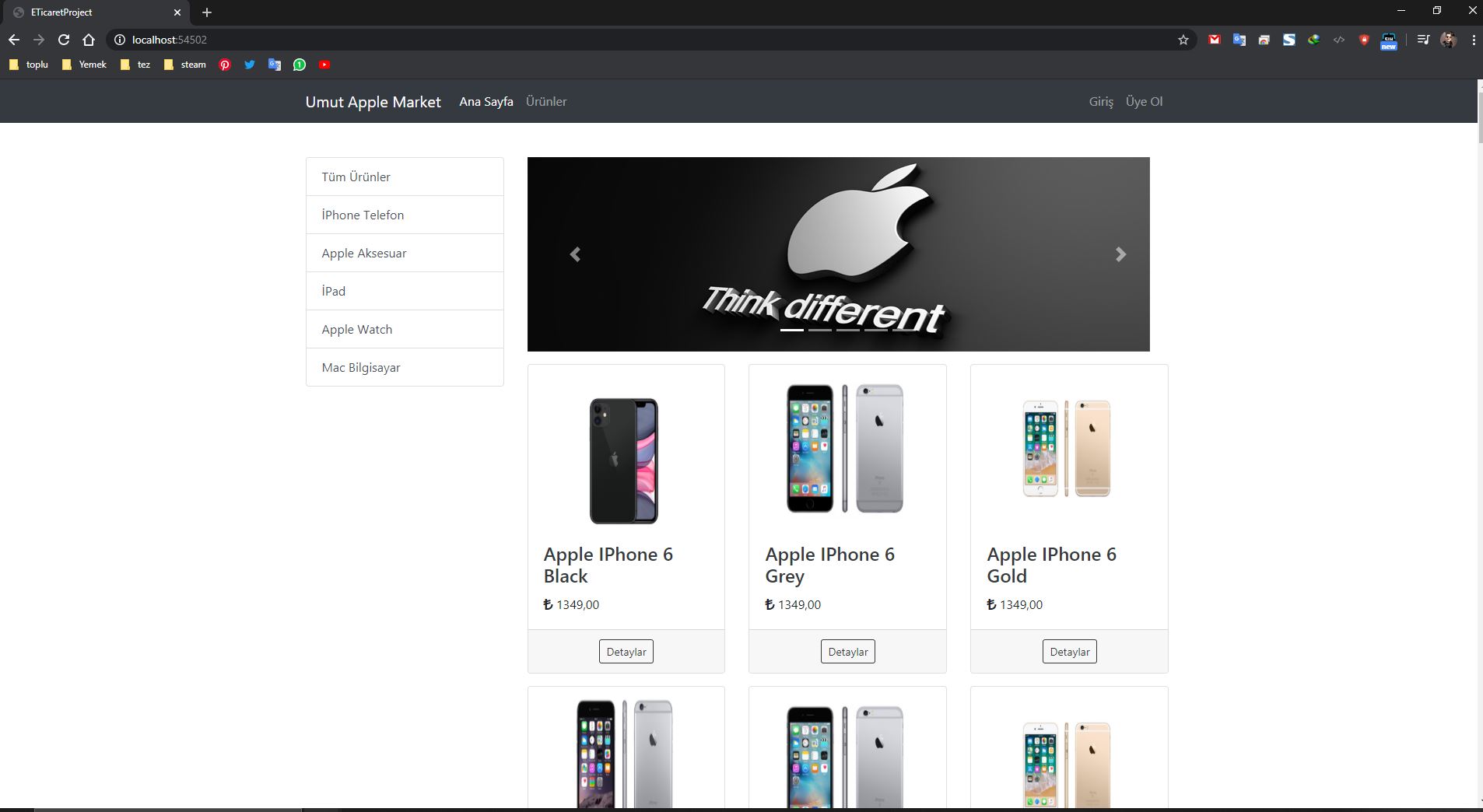
Bu paketleri indirip yükledikten sonra tekrardan dotnet --info komutunu çalıştırıp versiyonun 3.1.2xx olduğunu görelim.

Daha sonra Visual Studio’yu açıp veri tabanını import etmeliyiz. Bunu yapmazsak kodu çalıştıramayız. Bunun için Visual Studio’nun menüsünden View(Görünüm) sekmesinden SQL Server Object Explorer(SQL Server Nesne Gezgini) açmalıyız. Açılan sekmede SQL Server altında (localdb)\MSSQLLocalDB altında Databases’e gelip sağ tıkladıktan sonra açılan pencereden Publish Data-tier Application’u seçiyoruz. Açılan pencerede File on disk kısmında Browse… seçip proje klasörünün altında database.dacpac seçiyoruz. Yine açılan pencerede “Database name:” kısmına eticaret yazıyoruz. İşlemimizi onaylamak için Publish’e tıklıyoruz ve veri tabanımız ve verilerimiz SQL Server Object Explorer altında oluşuyor.

Şimdi projemize geçebiliriz.

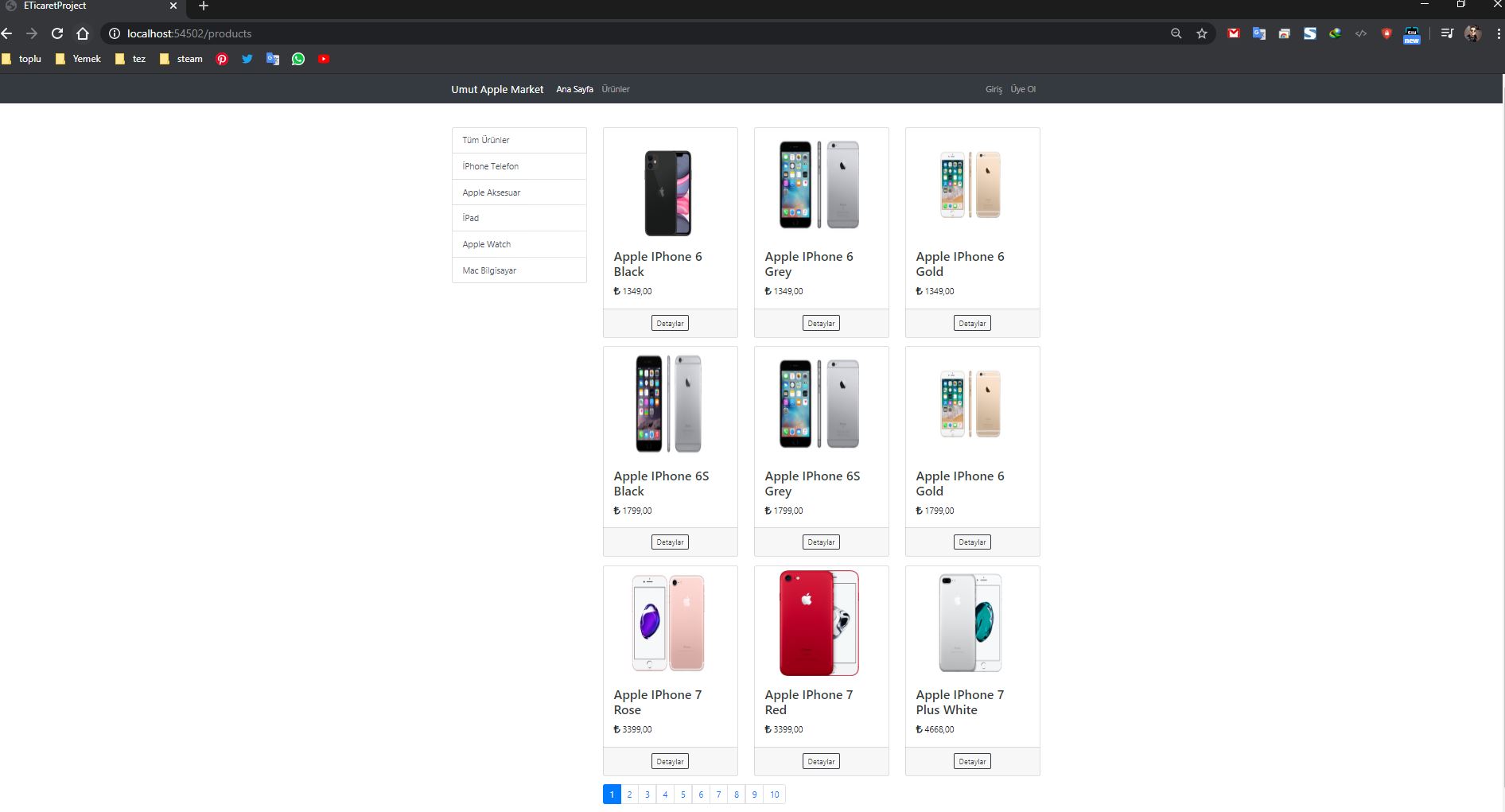
Uygulamayı ilk defa çalıştırdığımızda bizi öncelikle Ana Sayfamız karşılar. Şuan hiçbir şekilde giriş işlemi yapılmadığını farz edersek sayfa bizi ziyaretçi (login olmamış) olarak görüyor. Ana sayfada uygulamadaki tüm ürünler ve üstte bir slider bölümü bulunmaktadır. Buradan navbar’dan ya da menüden diğer sayfalara geçiş yapabiliriz. Gidebileceğimiz sayfalara bakarsak;

* Navbar’dan ürünler linkini seçerek ya da menüden herhangi bir kategoriyi seçerek ürünler kısmına ulaşabiliriz.
* Aşağıdaki ürünlerin görüntülendiği kartlarda ürünün altındaki Detaylar butonundan ürün detay sayfasına geçiş yapabiliriz.
* Navbar’dan Üye ol linkinden üyelik sayfasına geçiş yapabiliriz.
* Navbar’dan Giriş linkinden giriş yap sayfasına geçiş yapabiliriz.



Şekil 1 Ana Sayfa

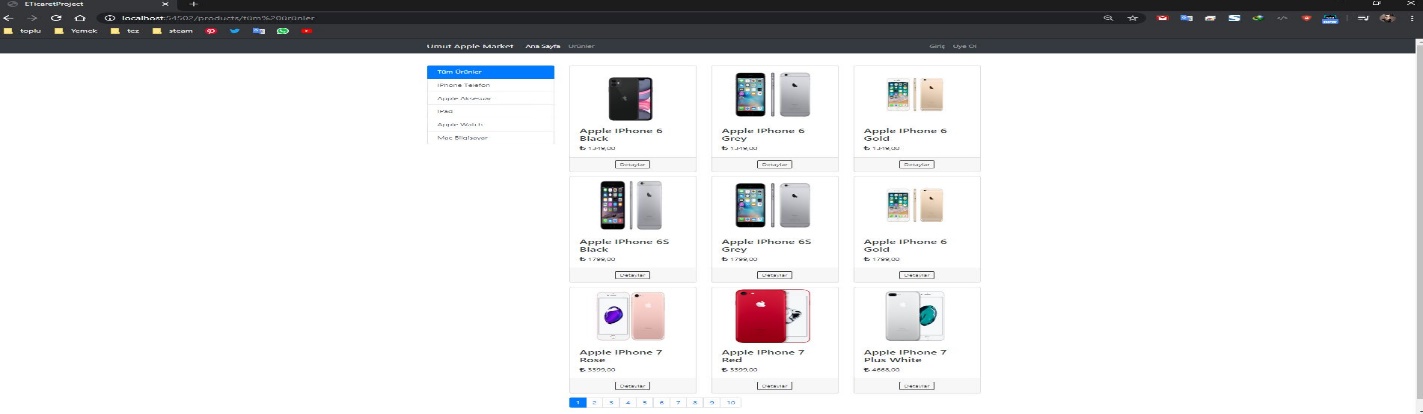
Şimdi üstteki navbardan ürünleri linkini seçtiğimizde karşımıza gelecek olan sayfa şu şekildedir.



Şekil 2 Ürünler Sayfası

Ürünlerin tümünün görüldüğü sliderın kaybolduğu fakat sayfanın dokuzlu şekilde ürünlerin görüleceği şekilde bir pageable bir düzende geldiğini göreceğiz. Menümüzdeki tüm ürünler kategorisi de aynı işlemi sorgusu daha karmaşık bir şekilde gerçekleştirmektedir. Fakat getAllProduct metodumuz burada daha performanslı olduğu için tüm ürünler linkine tıklamadığımız sürece linq sorgusu çalıştırmadan tüm ürünleri daha basit bir sql sorgusu ile sayfamıza yerleştirdik. Bu sitede az ürünümüz bulunduğu için performans farkı gözle görülür şekilde olmaz. Fakat büyük bir sitede bunun yararını göreceğiz. Tek farkı urlde değişiklik olması.

Şimdi ise kategorileri gezelim. Burada projemiz hangi kategori seçildiyse o kategoriye ait tüm ürünleri karşımıza getirmektedir. İlk önce tüm ürünler sayfamızı görelim ve navbardaki linkten tek farkının url ve arkada çalışan sorgu olduğunu onaylayalım.

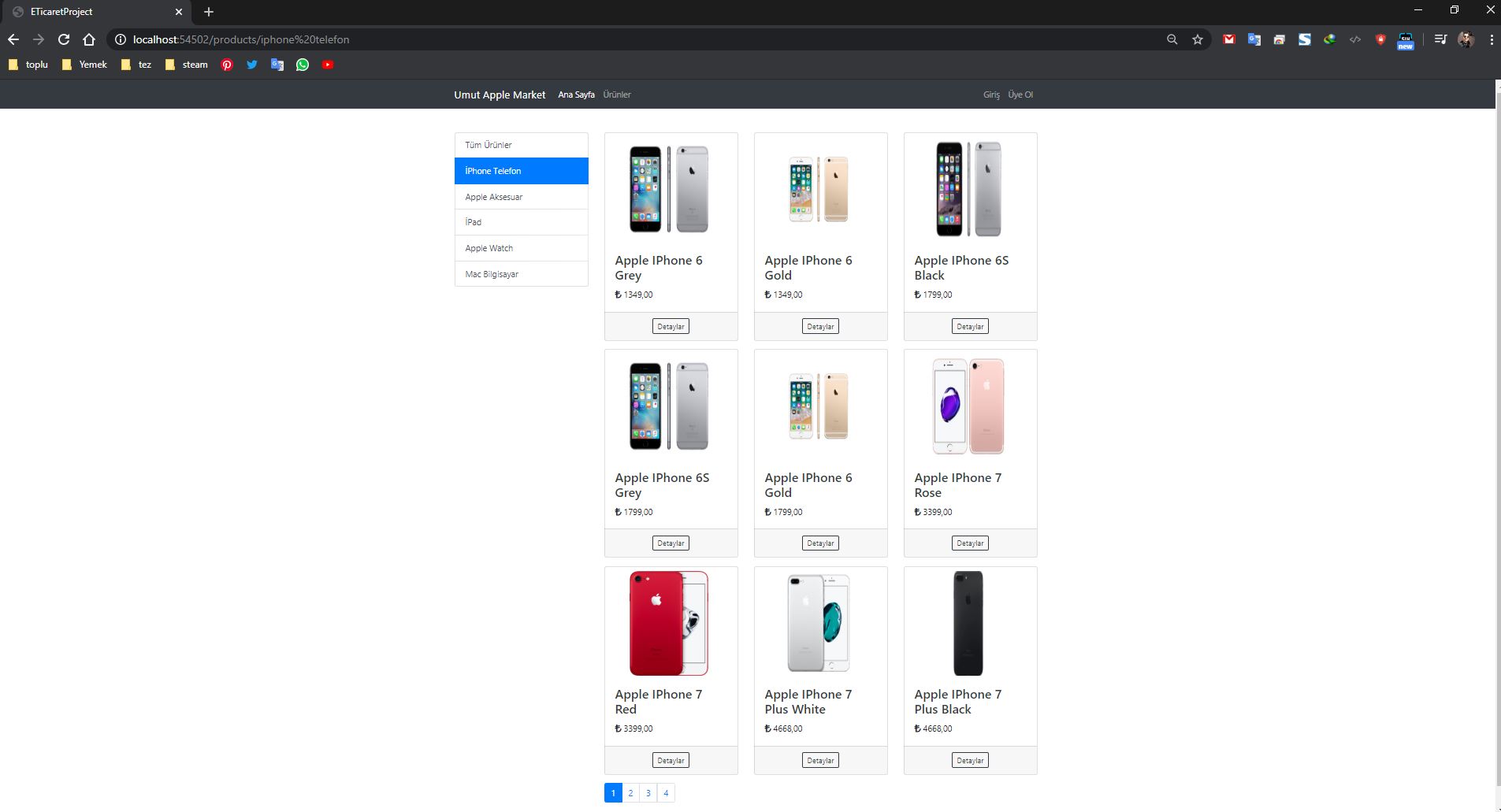


Şekil 3 Tüm Ürünler Kategorisi

Resimlerde de görüldüğü gibi pageable sayfalarda değişiklik yok ürünler aynı sırada her hangi bir değişiklik yok tek değişen url ve arkada çalışan sorgu.

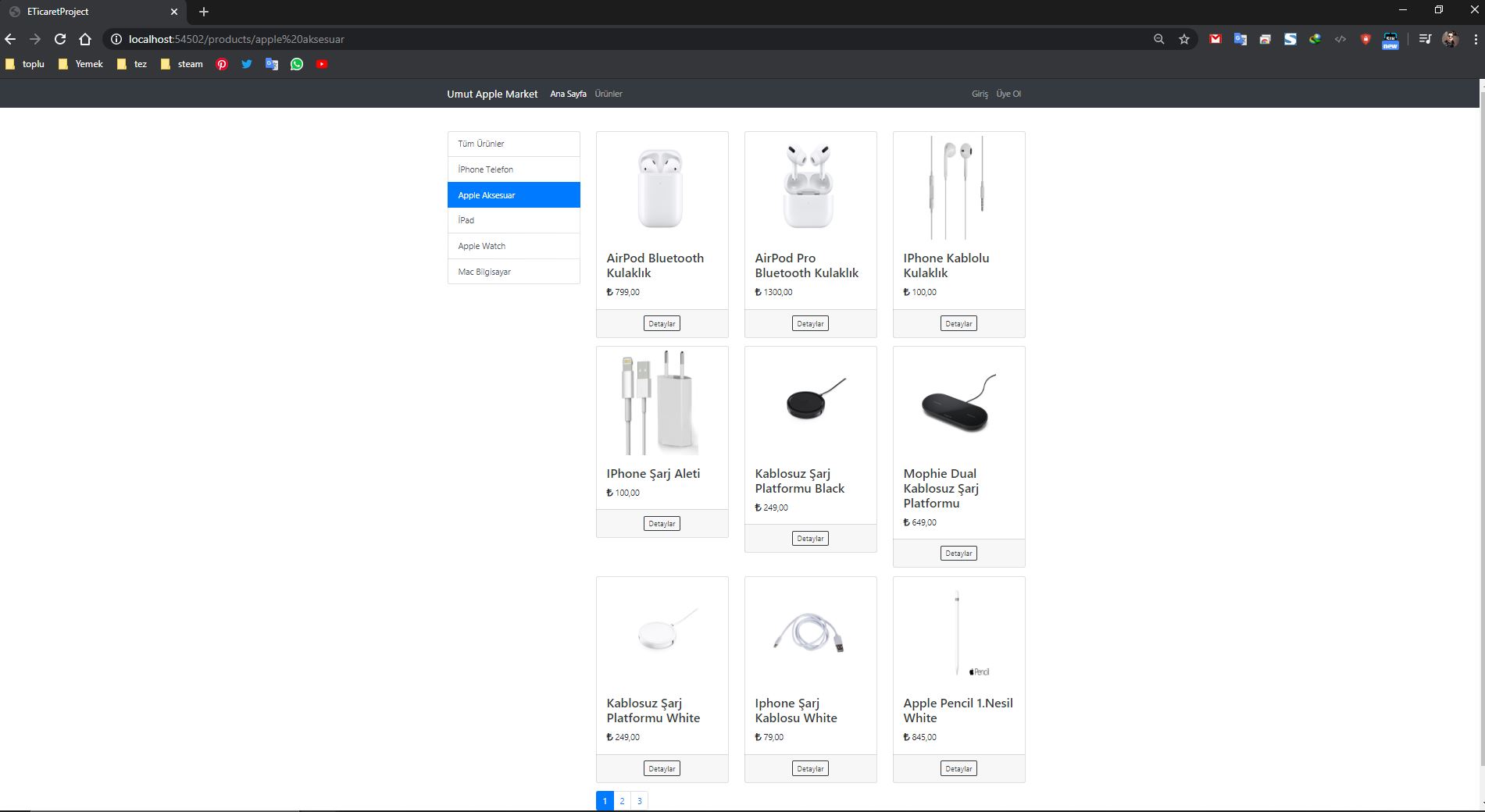
Şimdi diğer kategorilere de bakalım.

İphone Telefon kategorimizdeki ürünler.



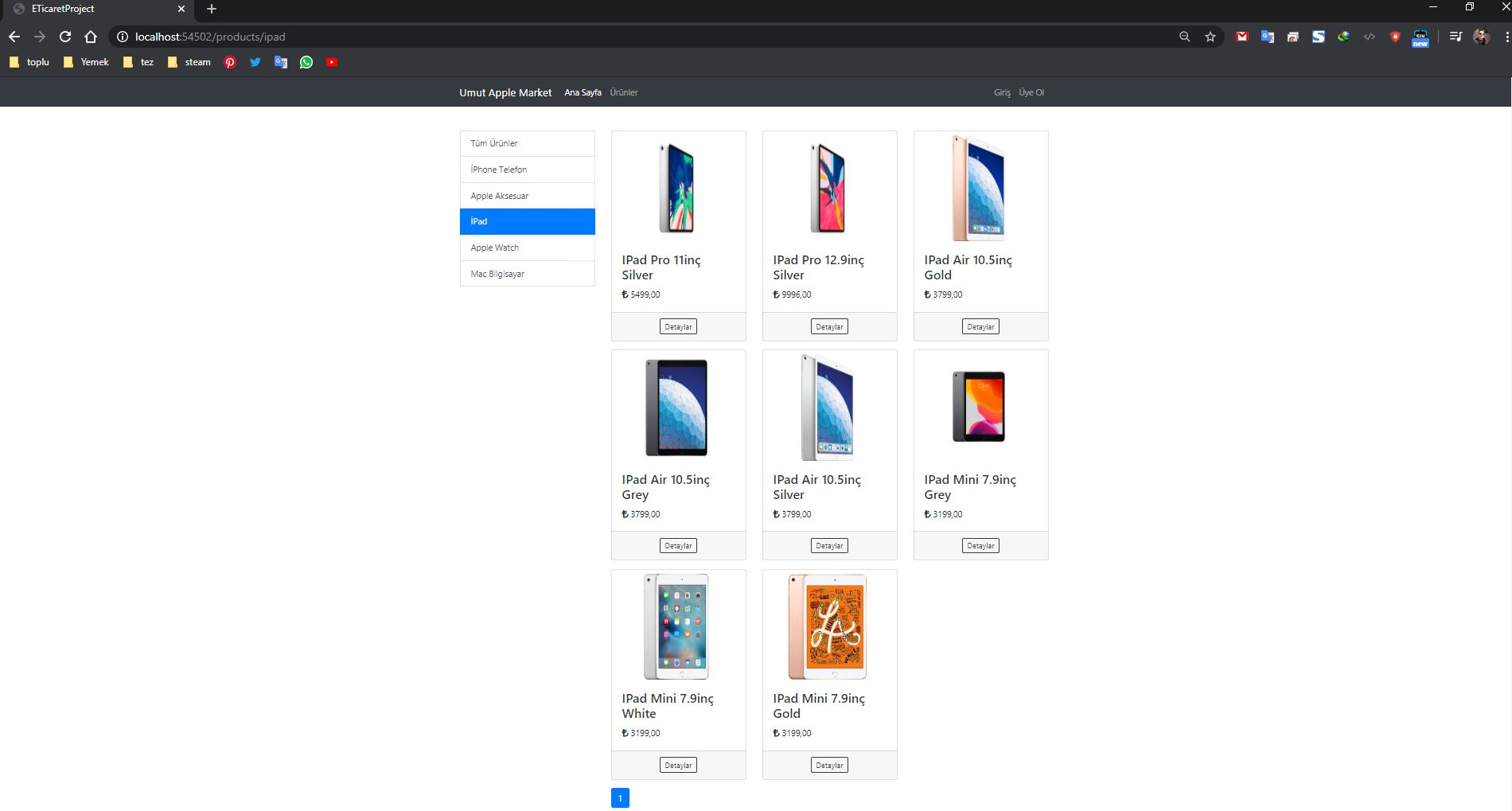
Şekil 4 IPhone Telefon Kategorisi

Apple Aksesuar kategorimizdeki ürünler.



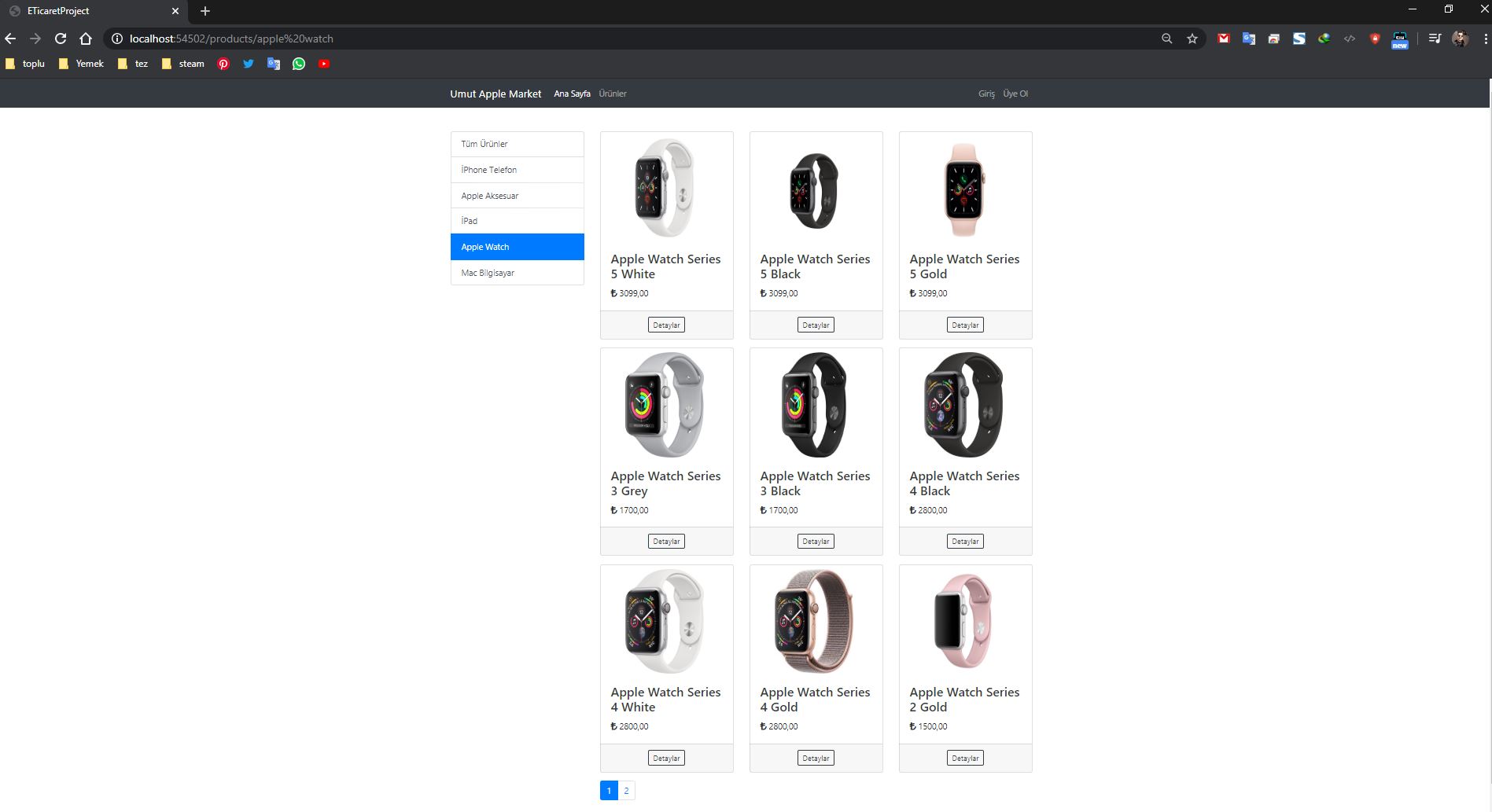
Şekil 5 Apple Aksesuar Kategorisi

İPad kategorimizdeki ürünler. Görüldüğü üzere bu kategorimizde az ürün olduğu için pageable özelliğimiz aşağıda bize tek sayfa açmış. İkinci bir sayfa geçiş yok fakat sayfa sınırımız dokuz olduğu için eğer iki ürün eklersek ikinci sayfa otomatikman alta buton olarak gelecektir.



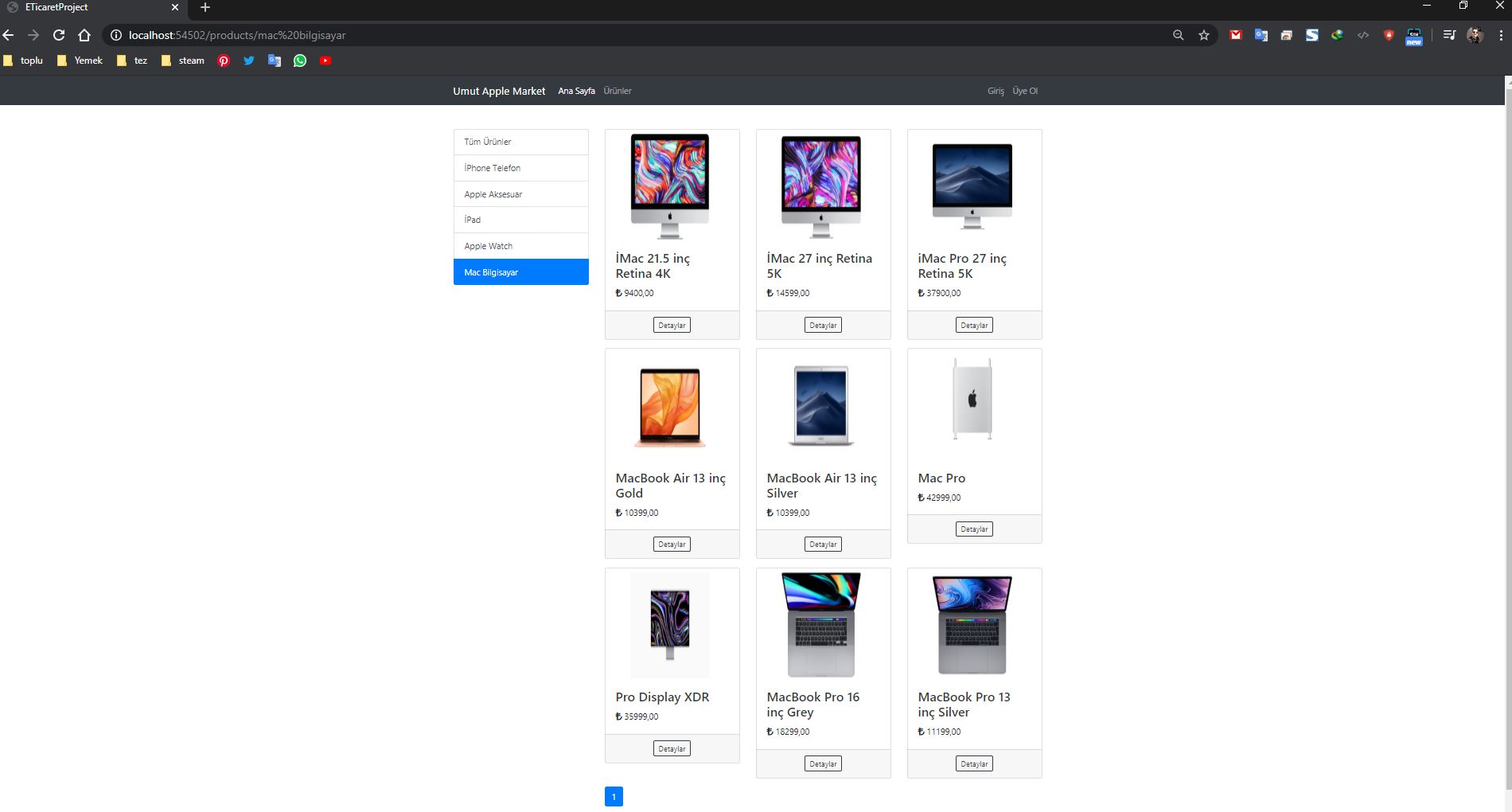
Şekil 6 IPad Kategorisi

Apple Watch kategorimizdeki ürünler.



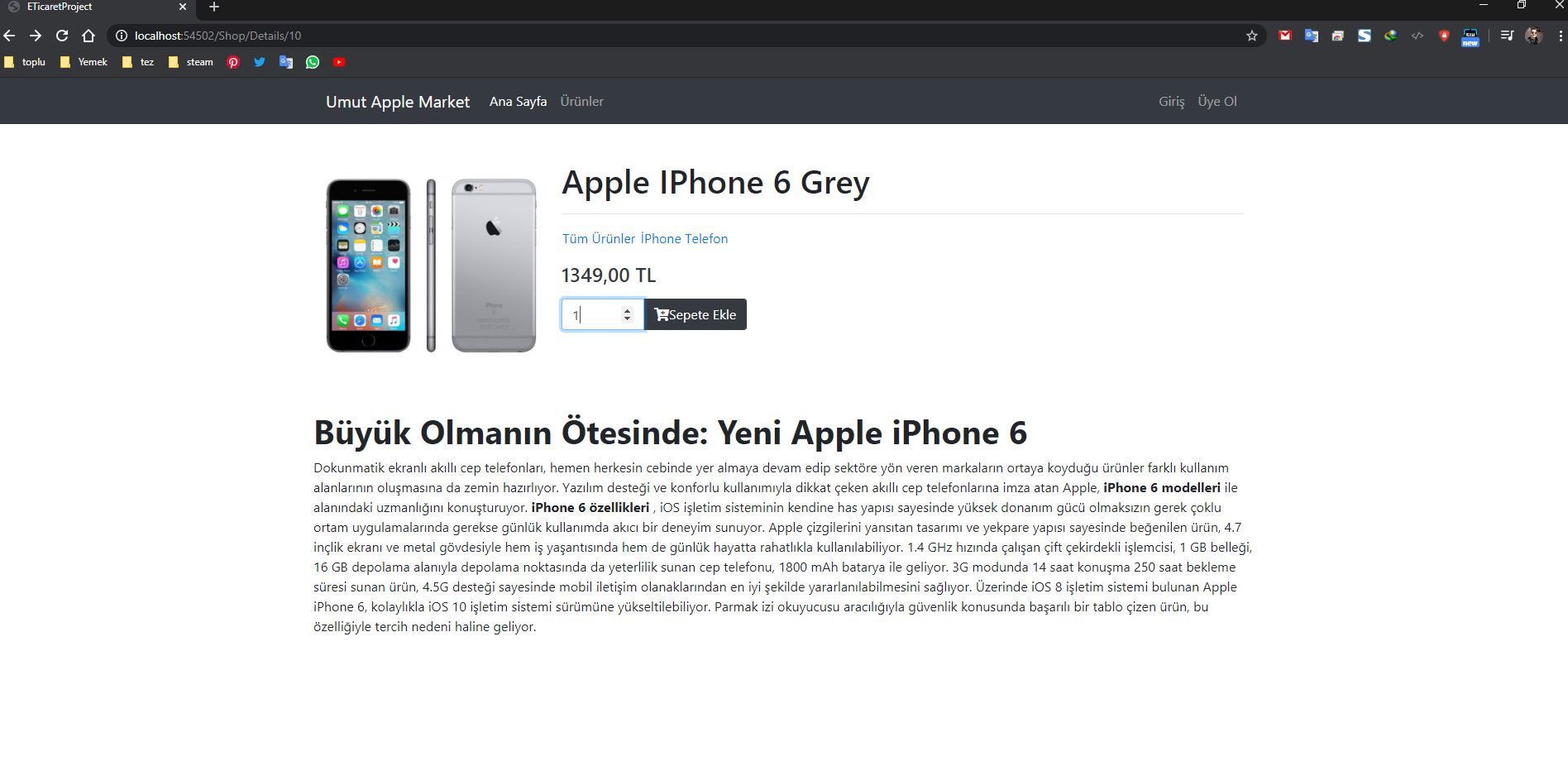
Şekil 7 Apple Watch Kategorisi

Mac Bilgisayar kategorimizdeki ürünler.



Şekil 8 Mac Bilgisayar Kategorisi

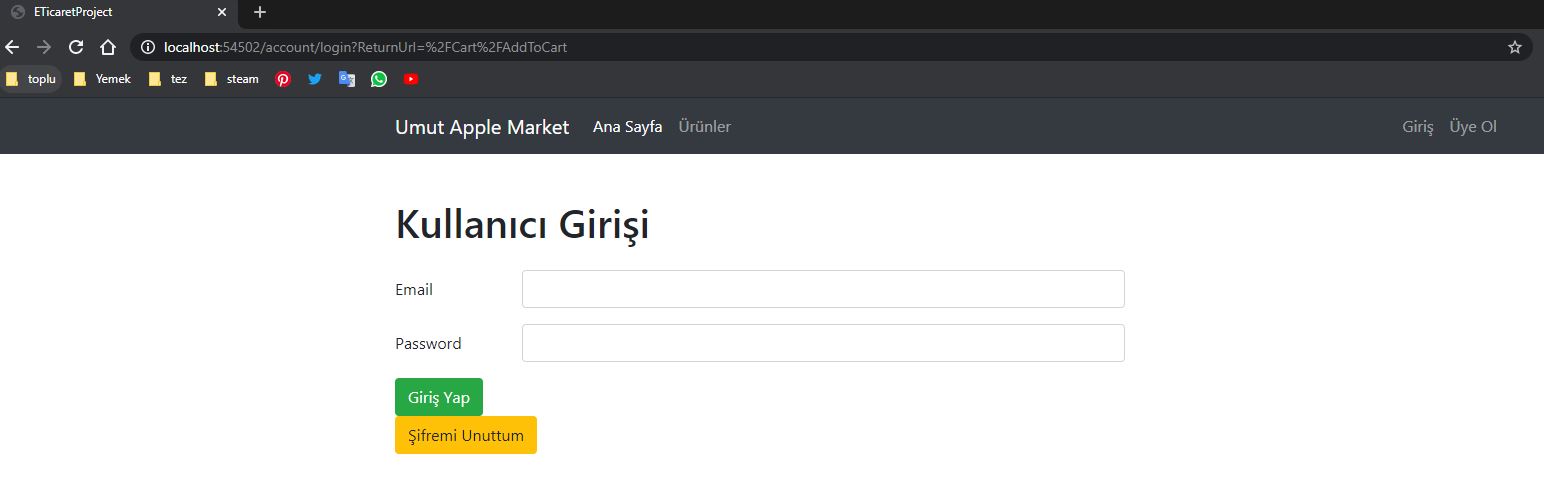
Kategorileri de gördük ve çalıştığını test ettik. Şimdi herhangi bir sayfada bir ürünün detaylar butonuna basıp ürün detayını görelim.



Şekil 9 Ürün Detay Sayfası

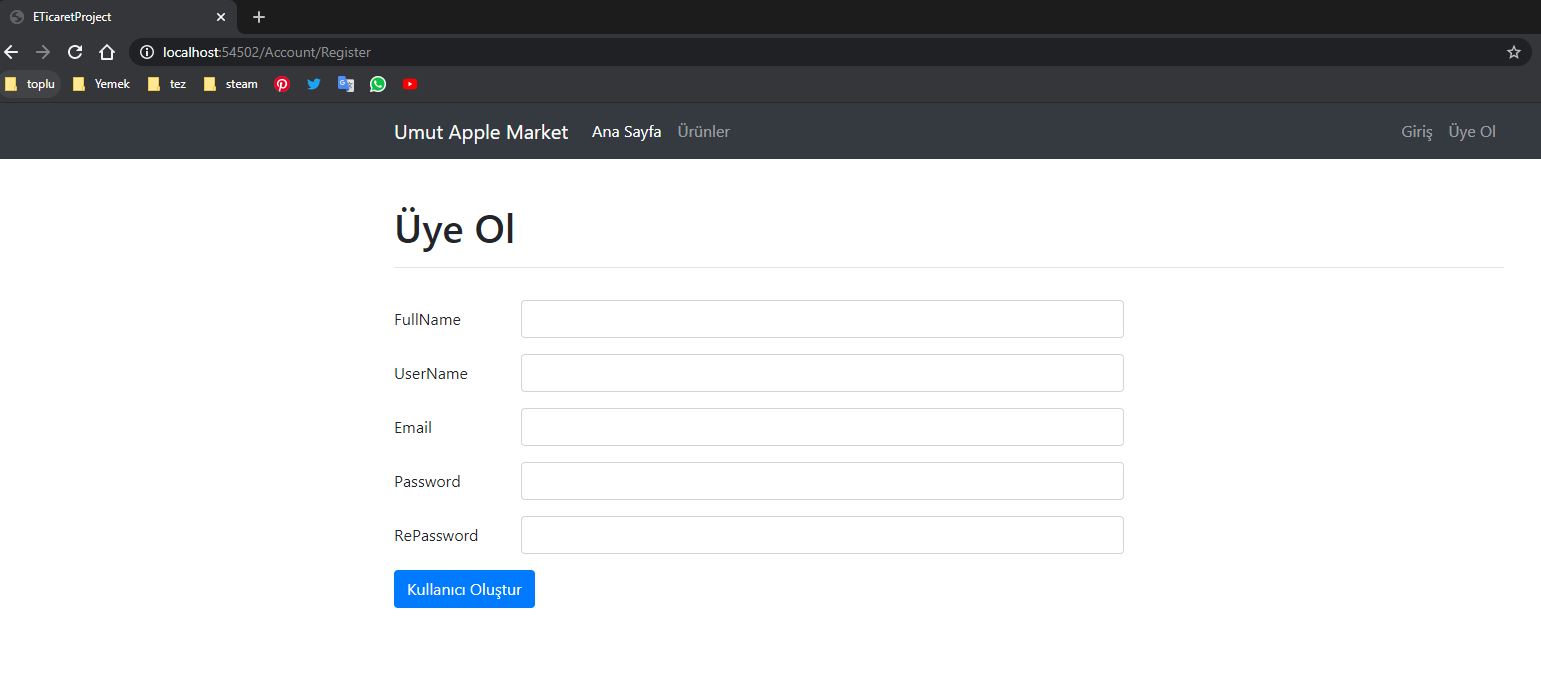
Evet, detaylar sayfamıza tıkladığımızda ürünümüzün isminin, kategorilerinin, sepete ekle kısmının ve açıklamasının olduğu bir sayfa önümüze açılmaktadır. Bu sayfada ürünün kategorilerine tıkladığımızda o kategorilere dönebilmekteyiz.

Sepete ekle kısmında üründen kaç adet almak istediğimizi seçtikten sonra tıkladığımızda bizi eğer üye girişi yaptıysak sepet sayfamıza eğer üye girişi yapmadıysak bizi login sayfamıza götürmektedir. Ürün kategorilerini incelemiştik şimdi sepete ekle kısmına tıklayıp bizi login ekranına götürmesini sağlayalım.



Şekil 10 Kullanıcı Giriş Sayfası

Sepete ekle kısmından login ekranına ulaştık. Şimdi buradan giriş yapabilmemiz mümkün değil öncelikli olarak üye ol sayfamızdan bir üyelik oluşturup e-mail onayımızı yapalım. Bunun için navbardan son link olan Üye Ol kısmını tıklayalım.



Şekil 11 Üye Ol Sayfası

Üye ol kısmına ulaştığımızda bizi inputların olduğu bir sayfa karşılar. Burada üye olabilmek için belirli kıstaslar bulunur. Bunlar;

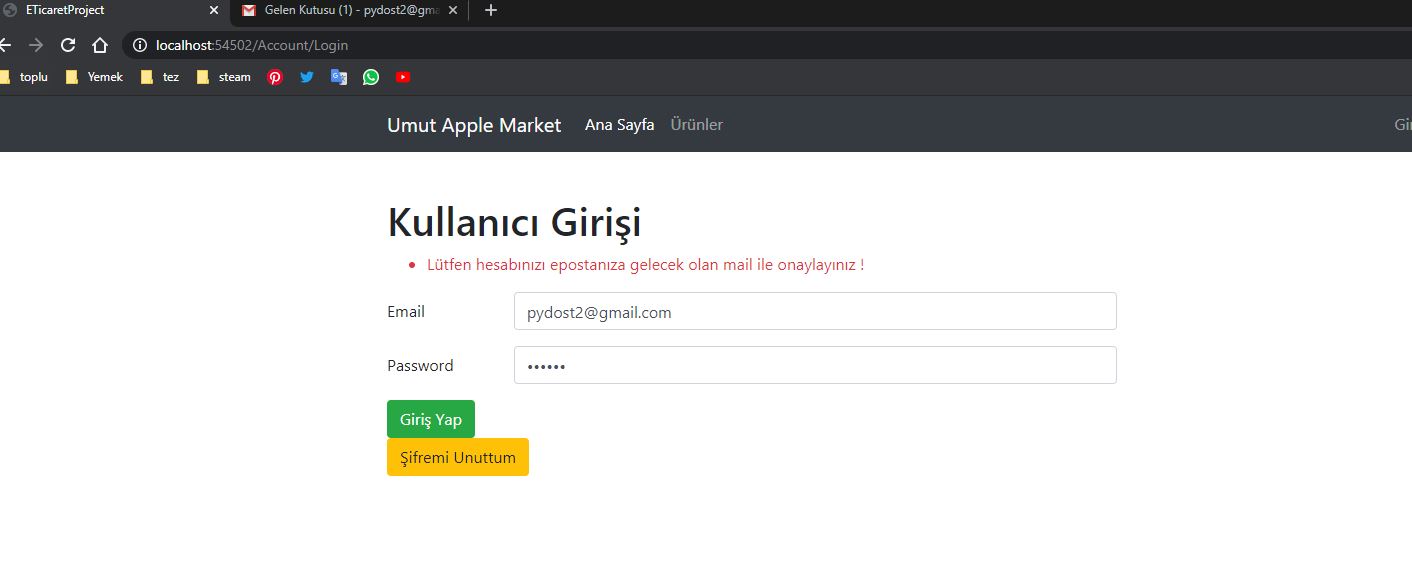
* FullName kısmının doldurulması gerekmektedir.
* UserName kısmının doldurulması gerekmektedir.
* Email kısmının doldurulması ve e-mail formatına uygun olması gerekmektedir.
* Password kısmının doldurulması ve minimum altı karakterden oluşması gerekmektedir.
* RePassword kısmının Password alanı ile aynı olması gerekmektedir.

şeklinde belirlenmiştir.

Bu şekilde yapmazsak doğrulama hataları oluşur ve sayfa güncellenmeden bize otomatik olarak belirli hatalar döndürülür. Şimdi bir üye oluşturalım ve e-mail onayı olmadan sayfamıza giriş yapıp yapamadığımızı görelim.

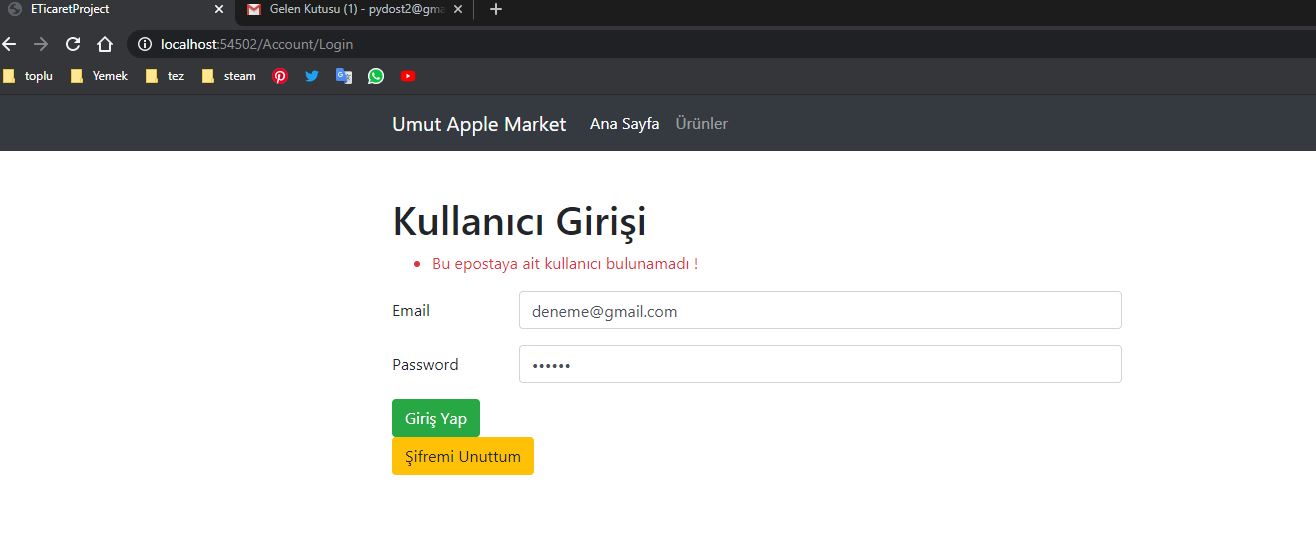
* FullName: Umut Demiray
* UserName: UmutD
* Email: pydost2@gmail.com
* Password: deneme
* RePassword: deneme

Yukarıdaki bilgileri yazarak bir kullanıcı oluşturuyorum. Üyelik oluşturulduktan sonra bizi login sayfasına otomatik yönlendirme yapıyor sistemimiz. Şimdi burada onay olmadan girişi ve farklı bir eposta ile girişi deneyelim ve hata aldığımızı görelim.



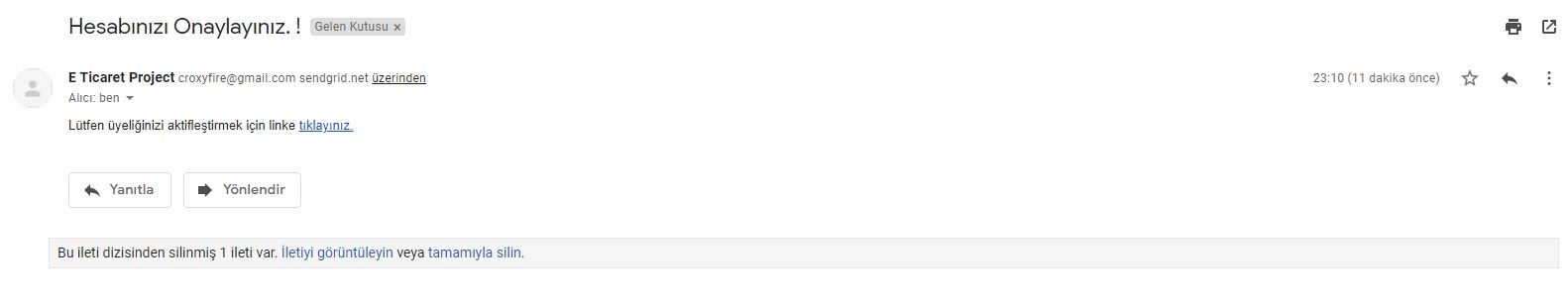
Şekil 12 E-Mail Onay İsteği

Herhangi bir onaylama işlemi yapmazsak bu hata ile karşılaşırız.



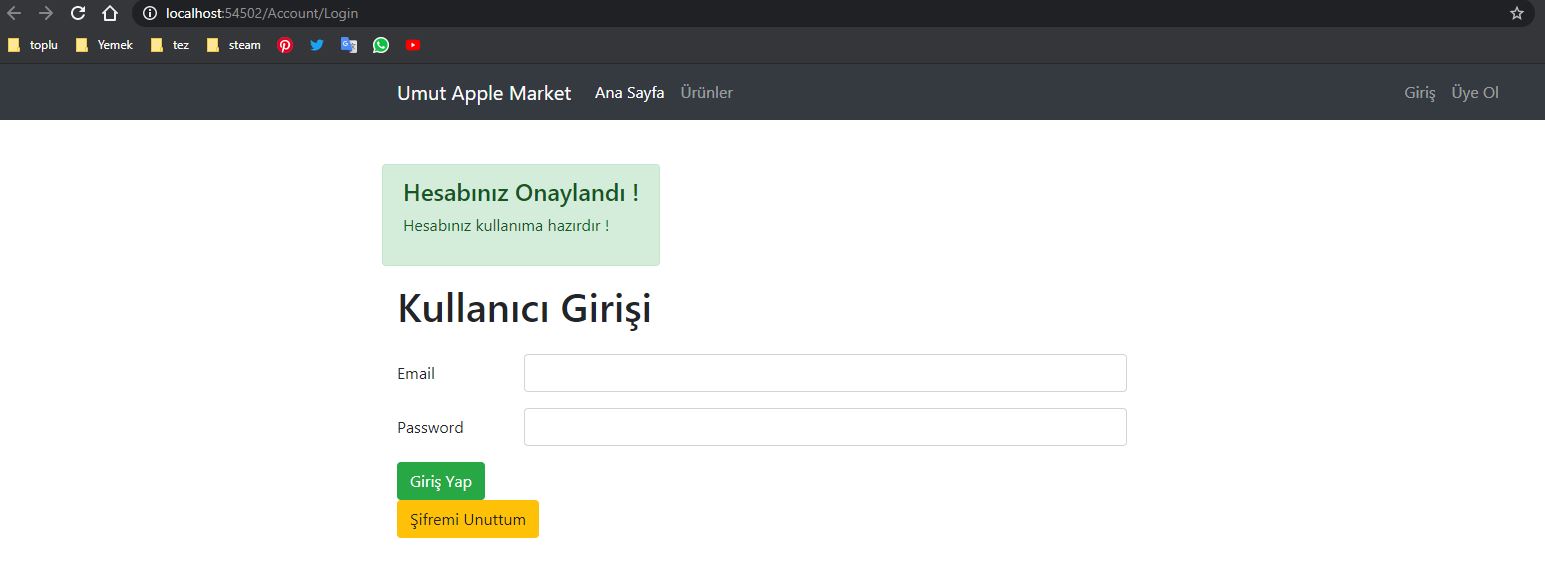
Şekil 13 Kullanıcı Bulunamadı Hatası

Üye olmadığımız bir eposta ile giriş denersek bu hata ile karşılaşırız. Şimdi epostamızı gidip linkimizi görelim ve bu linke tıklayıp giriş yapalım.



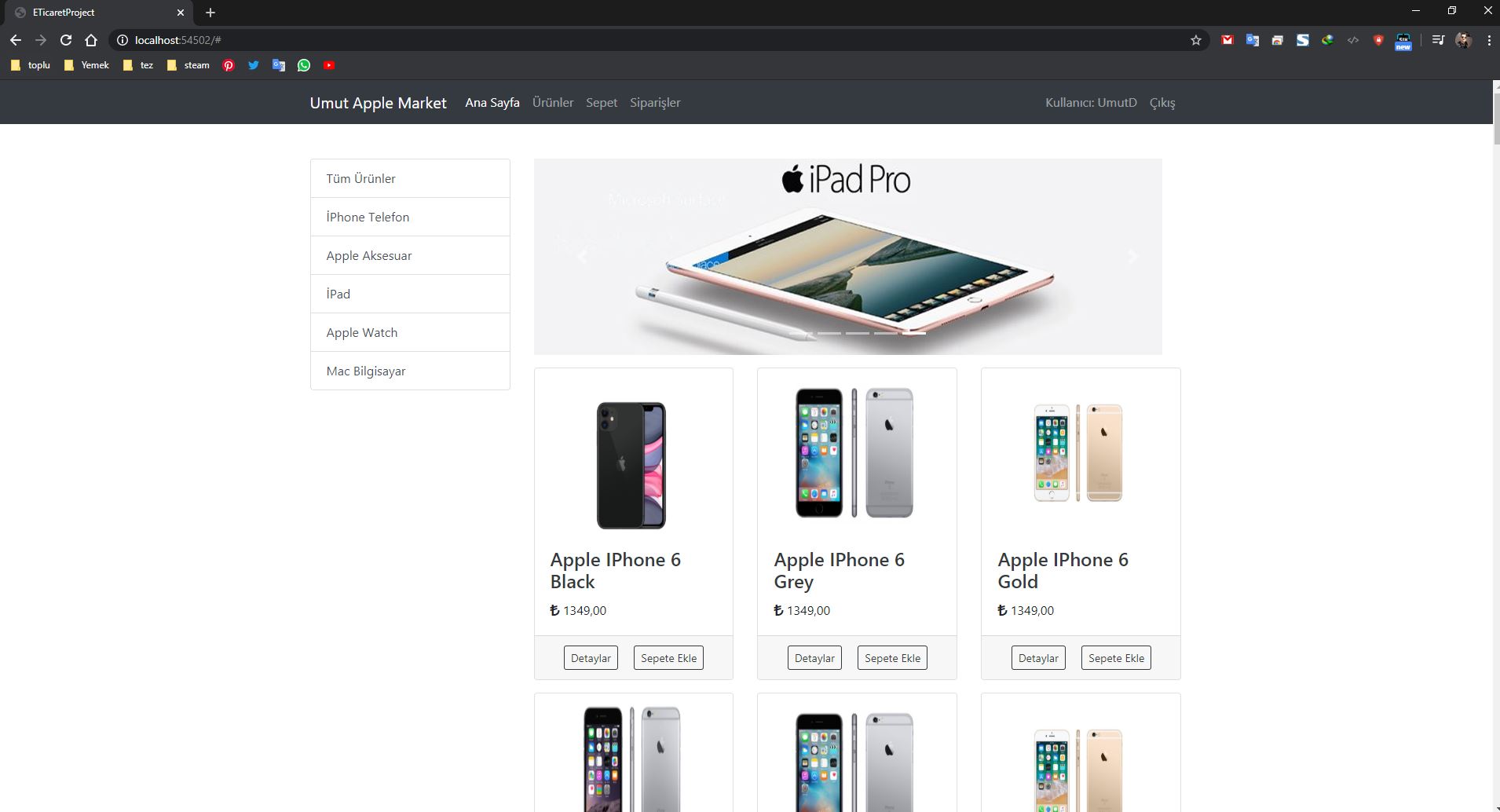
Şekil 14 Kullanıcı Onay Maili

E-mailimize girdiğimiz de böyle bir mail geldiğini göreceğiz ve tıklayınız linkine tıkladığımızda bize yeni sayfada web sitemizin login kısmını açar ve onaylandığına dair bir mesaj verir.



Şekil 15 Kullanıcı Mail Onaylandı

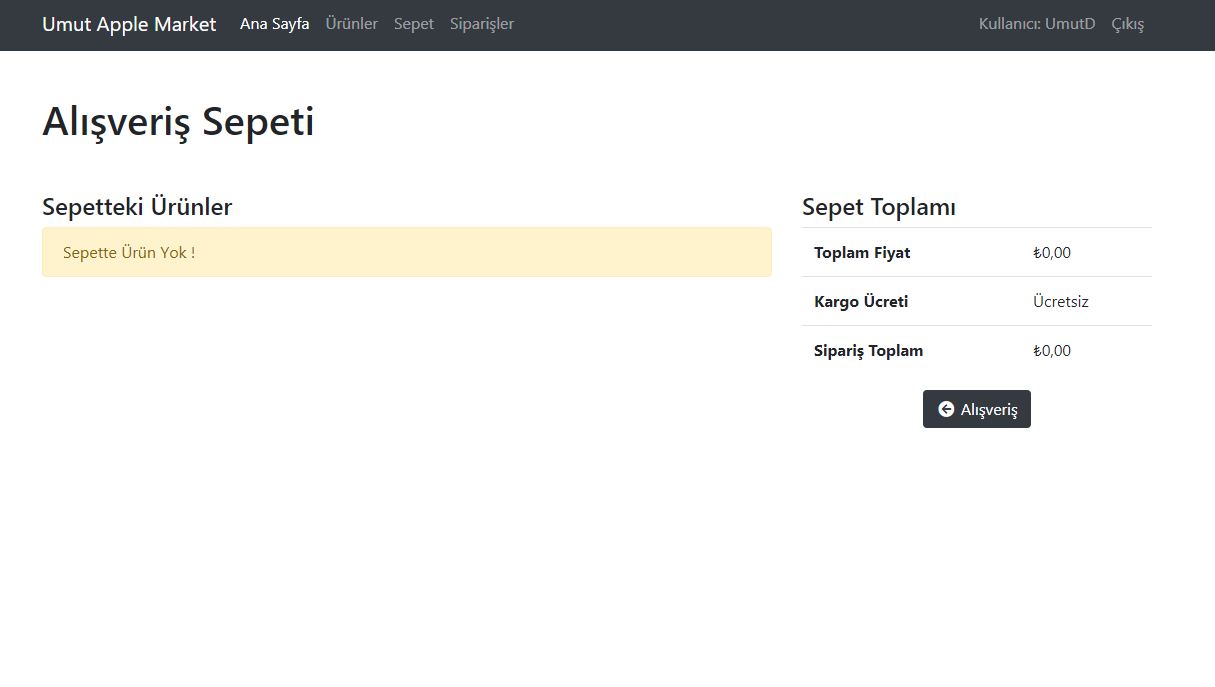
Onay aldığımıza dair mesajı gördüğümüze göre şimdi giriş yapabiliriz. Email ve password bilgilerimizi girip giriş yapıyoruz.



Şekil 16 Kullanıcı Ana Sayfa

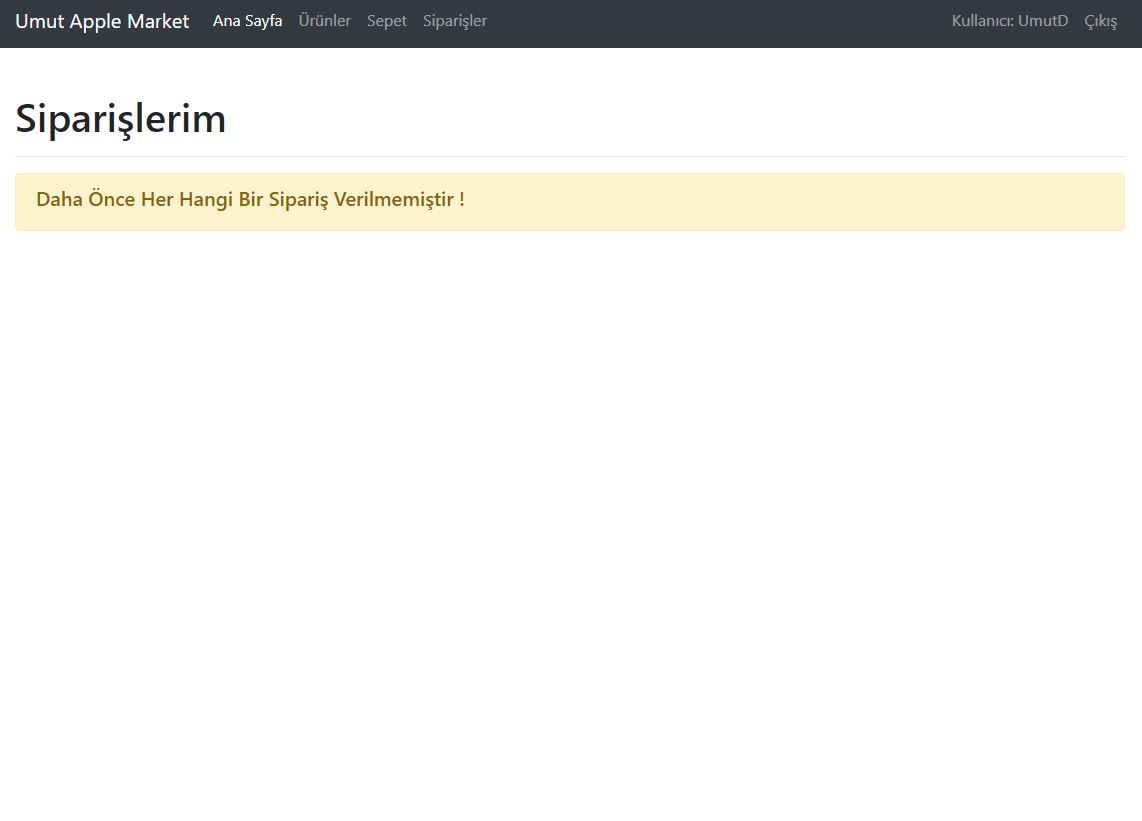
Giriş yaptığımızda navbar kısmında 2 yeni alan bize açılmış olur. Bunlar üstte de göründüğü gibi sepet ve siparişler linkleridir. Ayrıca artık ana sayfada ürünlerin detaylar kısmı yanında sepete ekle butonu da bulunmaktadır. Ekstra olarak kullanıcı adımız ve çıkış kısmımız oluşturulmuş, üye ol ve giriş yap kısımları da navbar’dan çıkarılmıştır.

Öncelikli olarak yeni gelen sayfalarımıza bir göz atalım. Sepet kısmına girdiğimizde bizi aşağıdaki resimdeki görüntü karşılar.



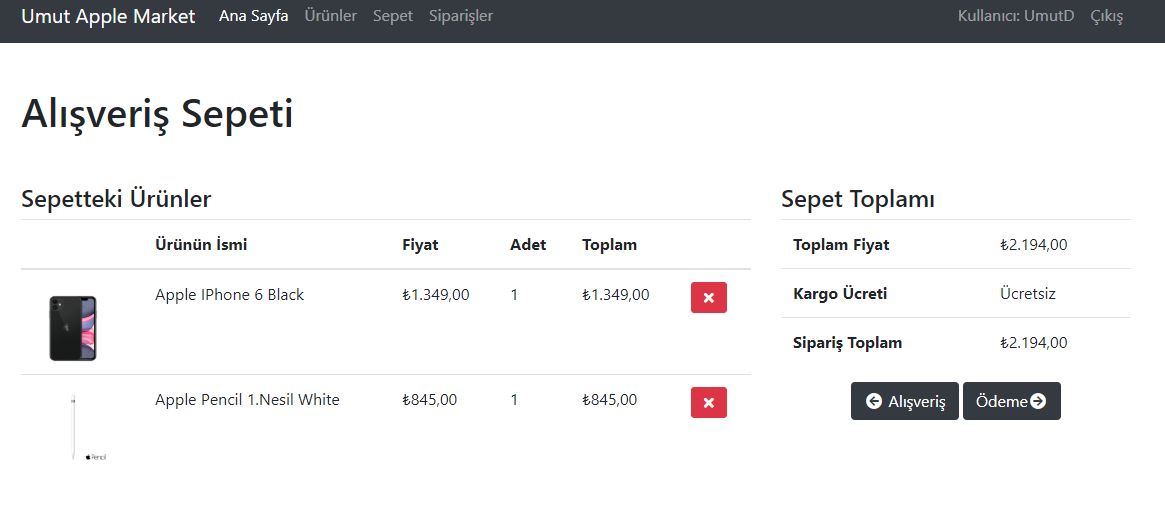
Şekil 17 Kullanıcı Alışveriş Sepeti Boş

Şuan bir ürün eklenmediğine dair bir bilgi kutucuğu ve alışverişe dönebilmemiz için sağ tarafta bir buton bulunmaktadır. Sipariş kısmını da inceledikten sonra ürün ekleyip burayı tekrardan kontrol edeceğiz.



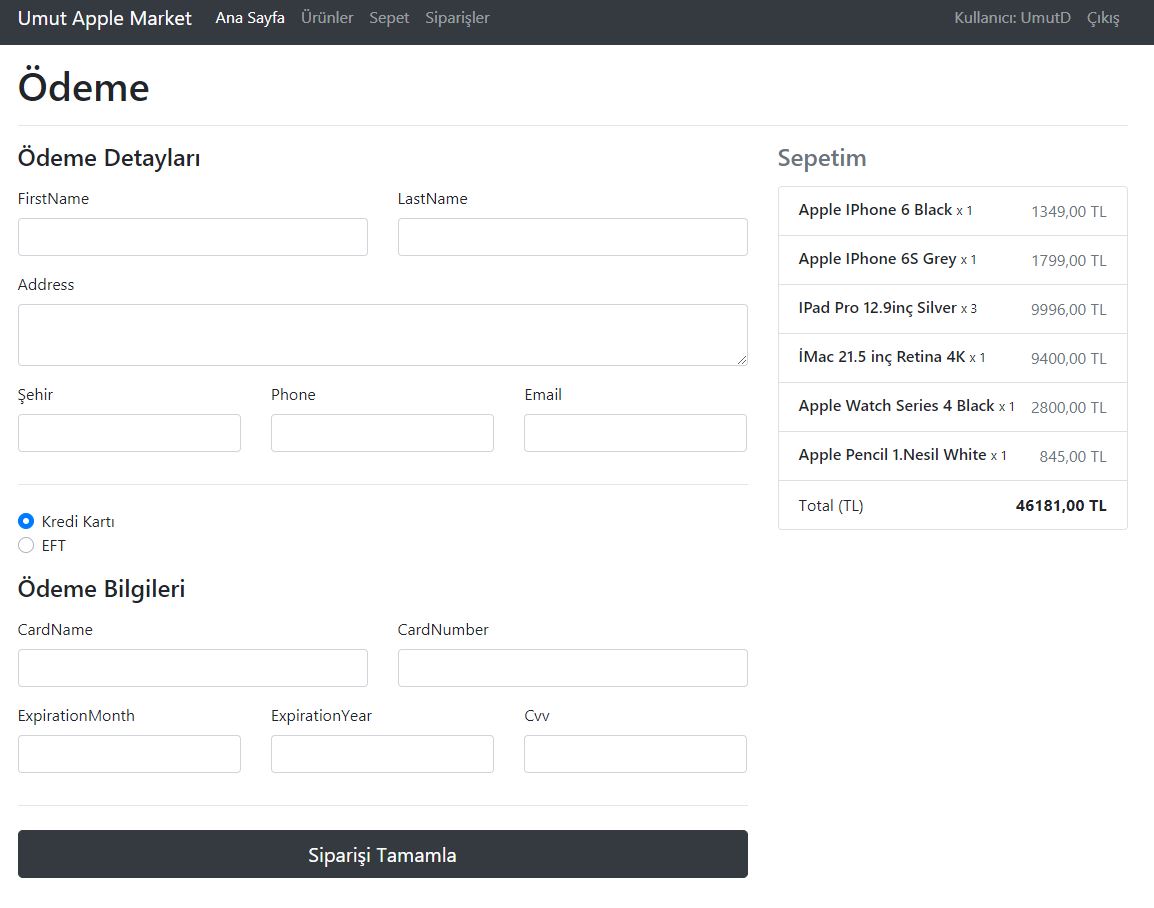
Şekil 18 Kullanıcı Siparişleri Boş

Evet, siparişler kısmında da herhangi bir sipariş göremiyoruz. Sadece bir bilgi kutusu ile bizim daha önce bir sipariş vermediğimizi söylemektedir. Şimdi sayfalara gidip bir miktar ürünü sepete ekle diyerek ekleyip sepet sayfamıza geri dönelim. Siparişimizi verdikten sonra buraya tekrar döneceğiz. Karşımıza gelen sayfa aşağıdaki gibidir.



Şekil 19 Kullanıcı Alışveriş Sepeti Dolu

Evet, sepetimize bir miktar ürün ekledik. Burada önemli olan nokta belli bir fiyatın üstünde sipariş vermememiz. Çünkü ödeme apimiz deneme amaçlı olduğu için yüksek miktarlarda ödemeyi kabul etmiyor. Ürün eklediğimizde sağ tarafta sepet toplamının güncellendiğini ve ödeme butonunun geldiğini görmüş durumdayız. Ürünleri kırmızı çarpı butonu ile sepetimizden çıkarabiliriz burada ya da alışveriş butonu ile alışverişe devam edebiliriz. Ödeme butonuna basıp ödeme kısmına geçiş yapalım. Karşımıza gelen sayfa aşağıdaki gibidir.



Şekil 20 Kullanıcı Ödeme Sayfası

Burada sipariş vermek için gerekli alanlar karşımıza gelmektedir. Üst kısımda siparişi alanın bilgilerinin alındığı bir kısım bulunmaktadır. Orta kısımda siparişin nasıl ödeneceği ilgili radio butonlar bulunmakta. Şimdilik sadece görsel amaçlı oluşturuldu. EFT ödemesini ödeme apimiz kabul etmediği için bu seçenek devre dışı bırakılmış durumda. Alt kısımda ise kredi kartı için gerekli alanlar bulunmakta.

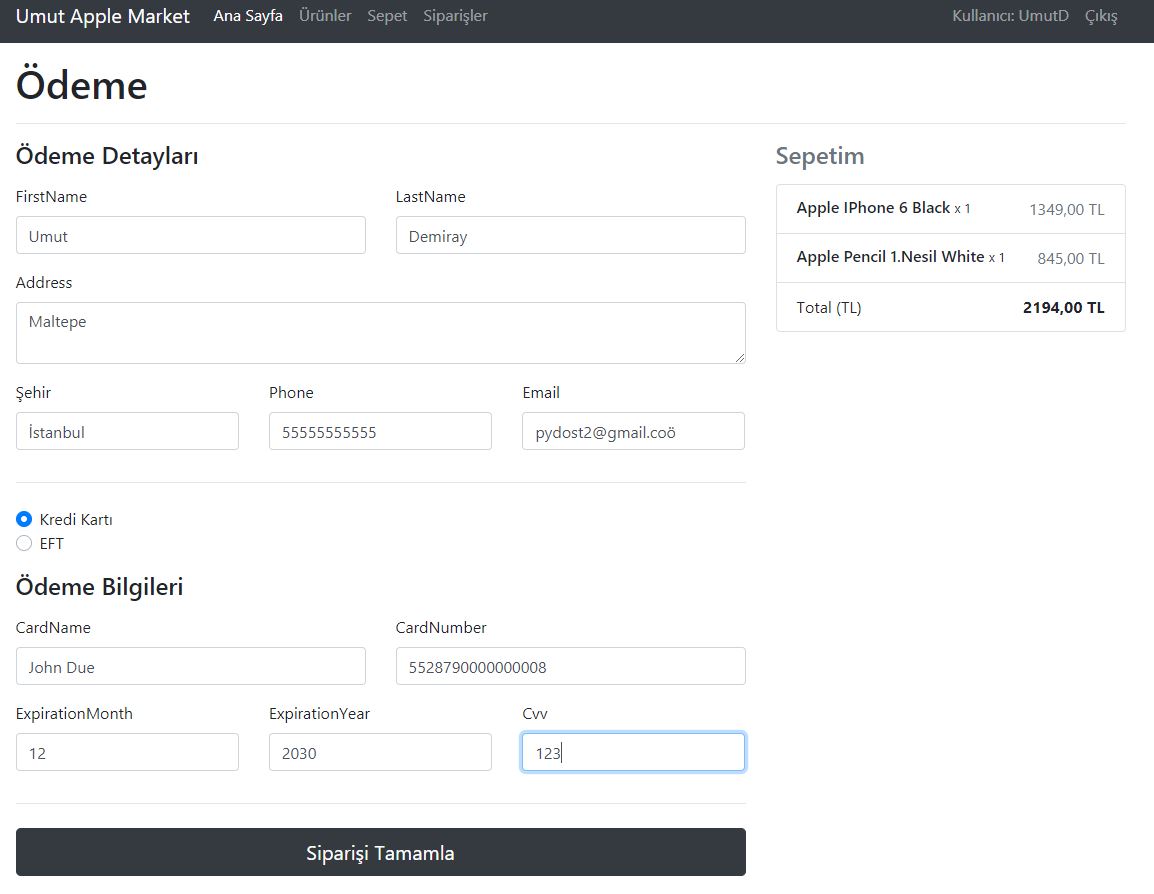
Buraya girilmesi gereken değerler gerçek kart bilgileri olmalıdır. İyzico’nun apisi bunu kontrol etmektedir eğer gerçek bir kart değilse işlemi yapmaktadır. Bundan dolayı bize verdikleri bot kart verisi ile ödeme tamamlanmalıdır. Sol kısımda ise ürünlerimizin göründüğü bir alan bulunmaktadır. Şimdi gerekli verilerimizi girelim. Test verisi için kredi kartı kısmına girilmesi gereken değerler şunlardır;

* CardName: John Doe
* CardNumber: 5528790000000008
* ExpiretionMonth: 12
* ExpiretionYear:2030
* CVC: 123

Burada eksik veri girilmemesi için bazı doğrulamalar yapılmıştır. Bunlar;

* Hiçbir alan boş olamaz
* Telefon bölümüne on ya da on bir haneli giriş yapılabilir ve sadece sayı girilebilir.
* Email bölümünde e-mail formatında bir mail adresi yazılmalıdır.
* CardNumber bölümünde on altı haneli giriş yapılabilir ve sadece sayı girilebilir.
* ExpiretionMonth bölümünde iki haneli giriş yapılabilir ve sadece sayı girilebilir.
* ExpiretionYear bölümünde iki haneli giriş yapılabilir ve sadece sayı girilebilir.
* Cvc bölümünde üç haneli giriş yapılabilir ve sadece sayı girilebilir.

Bu şekilde düzgün bir giriş sağlanırsa ve sipariş tamamla butonuna tıkladığımızda siparişimiz işleme alınır.



Şekil 21 Kullanıcı Ödeme Sayfası Gerekli Veriler

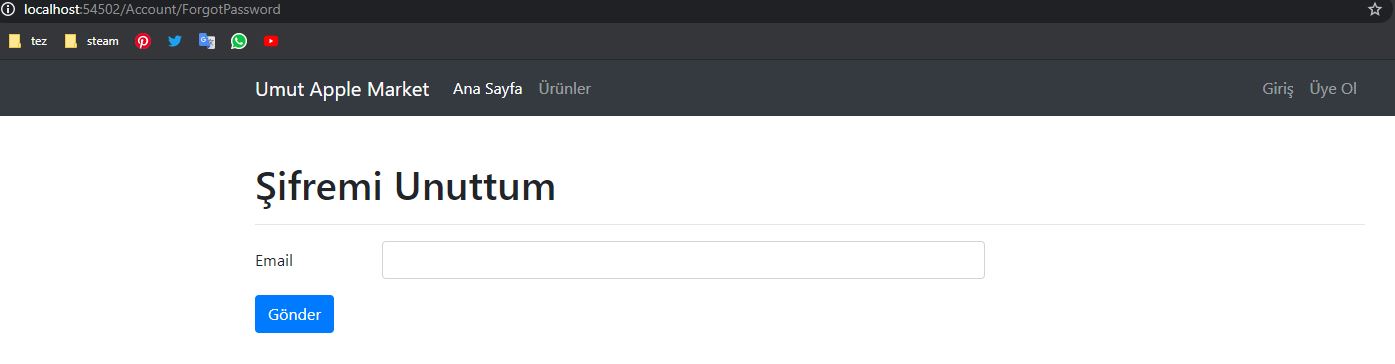
Evet, verilerimiz girdik. Şimdi işlemi tamamlayalım.



Şekil 22 Kullanıcı Siparişleri Dolu

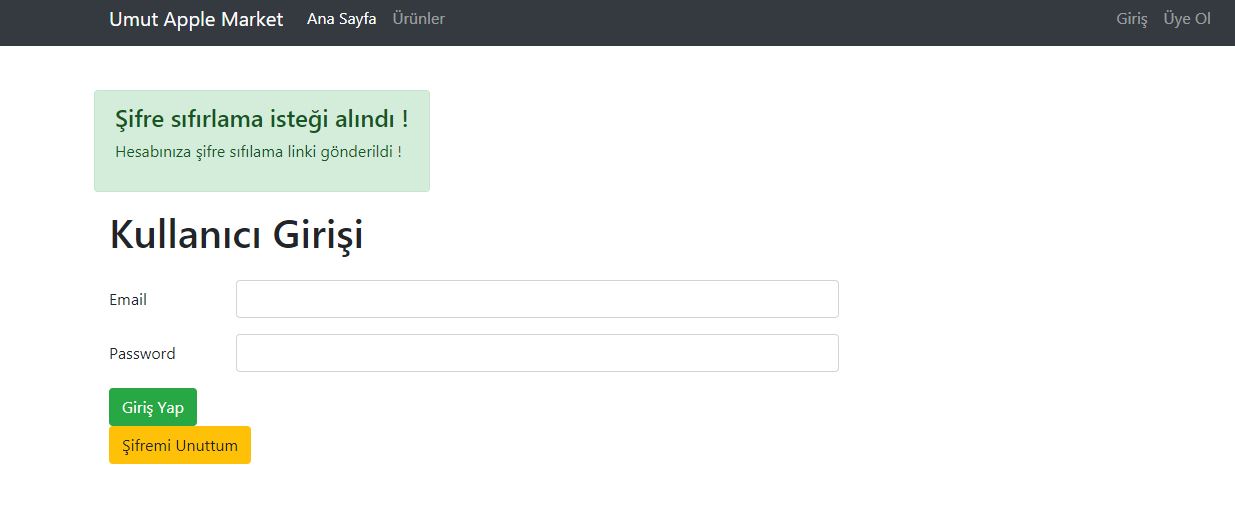
Siparişimiz tamamlandığında bizi sipariş sayfasına otomatik yönlendirme yapmaktadır. Siparişimiz gerçekleştiğine dair bize bir bilgi kutucuğu açtı ve buraya fatura bilgimizi getirdi ve eğer sepete dönecek olursak sepetimizin de tamamen boşaltılmış olduğunu göreceğiz. Eğer siparişimizde bir hata varsa örneğin kart bilgileri yanlışsa ya da miktarı karşılayacak kadar kartta para yoksa siparişiniz işleme alınamamıştır mesajı ile bizi sepet sayfasına yönlendirecekti ve işlemi tekrar yapmamızı isteyecekti web sitemiz.

Şimdi çıkış yapıp şifremi unuttum kısmını deneyelim ve çalıştığından emin olalım. Navbar’dan çıkışa bastığımızda bizi ana sayfaya yönlendirip çıkış yapıldı bilgi kutucuğu çıkartır. Daha sonra giriş kısmına girip şifremi unuttum butonuna tıklıyoruz.



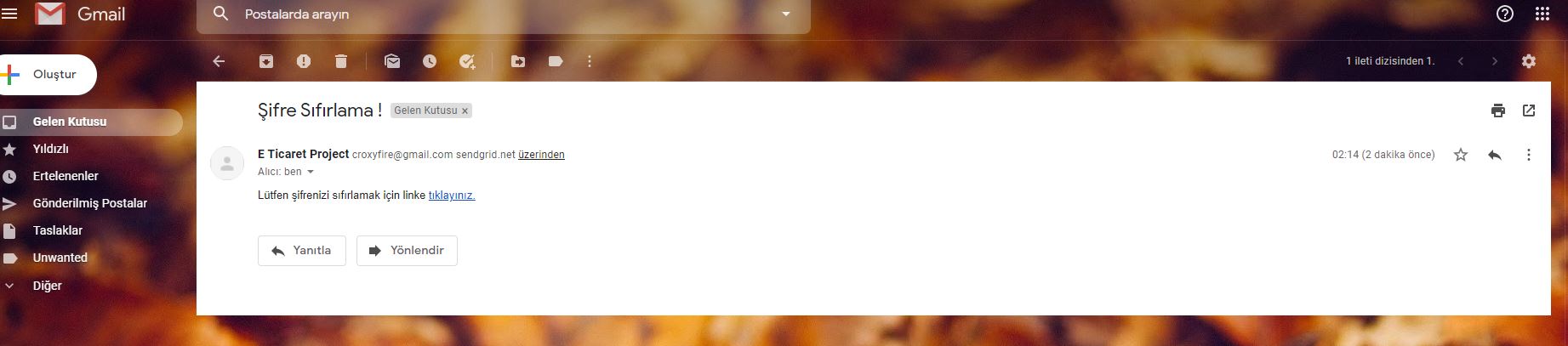
Şekil 23 Şifremi Unuttum Sayfası

Önümüze gelen sayfada bir input bulunmakta ve e-mail istemekte eğer e-mailimizi doğru girersek bizi login sayfasına yönlendirip şifre sıfırlama isteği alınmıştır şeklinde bir bilgi kutusu gösterip e-mailimize bir şifre sıfırlama linki gönderilir.



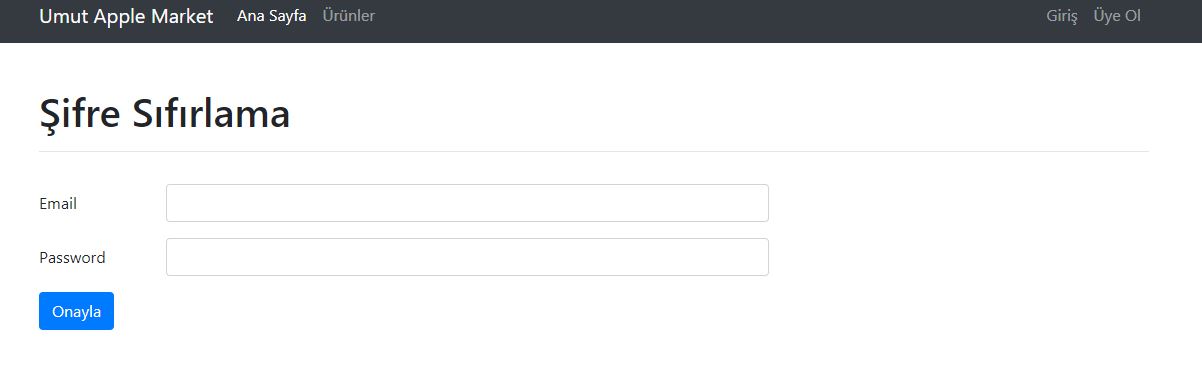
Şekil 24 Şifremi Unuttum İsteği Alındı

E-mailimize girip şifre sıfırlama linkine tıklamamız gerekmektedir. Öncelikte e-mailimizi kontrol edelim.



Şekil 25 Şifremi Unuttum Maili

E-mailimizde gelen postayı açıp linke tıklıyoruz ve bizi sitede şifre sıfırlama sayfasına yönlendiriyor. Burada e-mail ve şifre bilgisi istenmekte bu şifre bizim yeni şifremiz olacak.



Şekil 26 Şifre Sıfırlama Sayfası

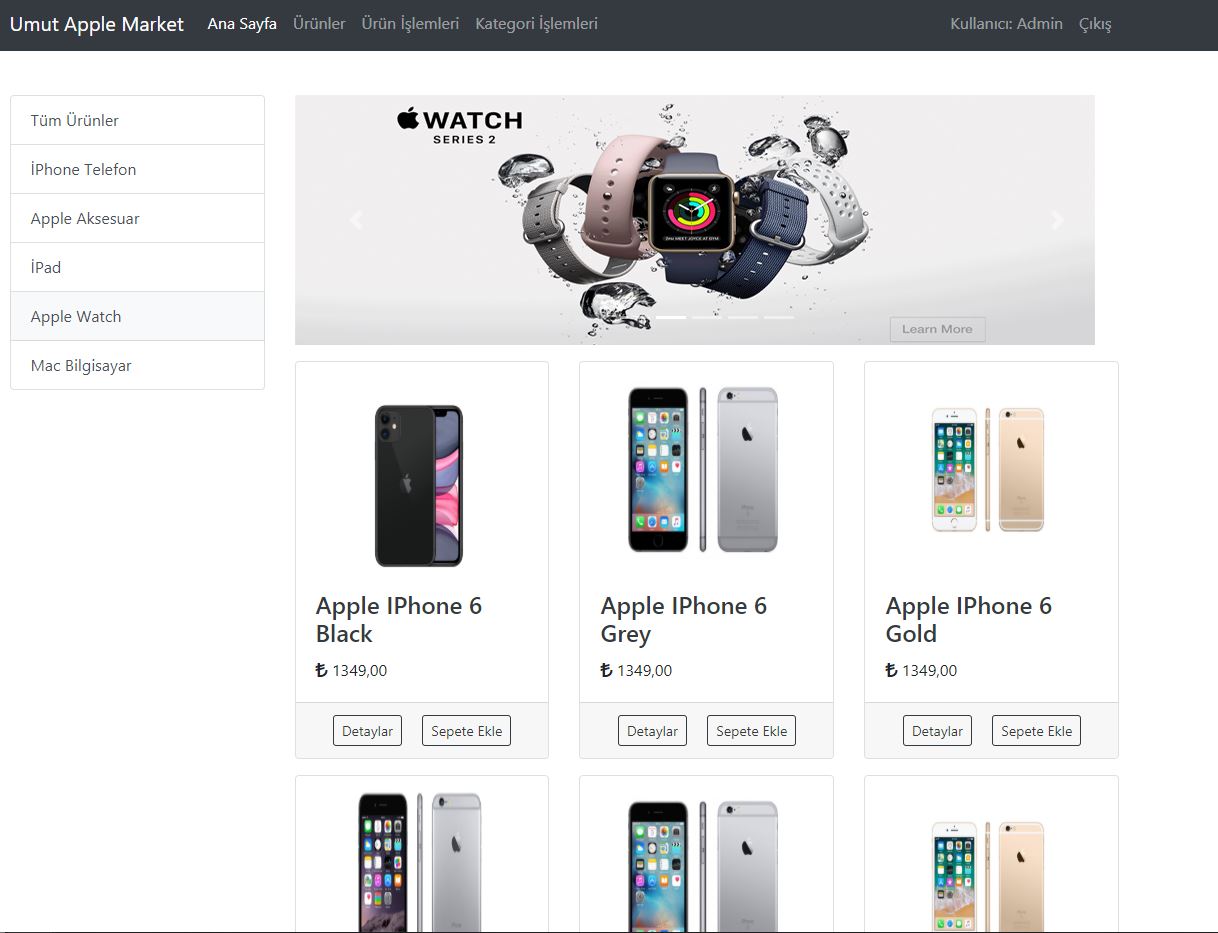
İstenen değerleri düzgünce girdiğimizde bizi giriş sayfasına tekrar yönlendirmekte ve yeni şifremizle giriş yapabilmekteyiz.

Evet, buradan sonra üye ile yapabileceğimiz başka işlem kalmadığı için admin rolüne sahip olan kullanıcıya giriş yapıp ürün ekleme kategori ekleme işlemlerini yapabiliriz. Web sitede tek admin bulunmakta ve bu veri static olarak uygulama her tekrar çalıştırıldığında kontrol edilip eğer yoksa veri tabanına tekrardan eklenmektedir. Admin kullanıcımızı kaybetmemek için böyle bir yola başvurduk.

Admin kullanıcımız ile giriş yapmak için giriş sayfasına geçip aşağıdaki bilgiler ile giriş yapmalıyız.

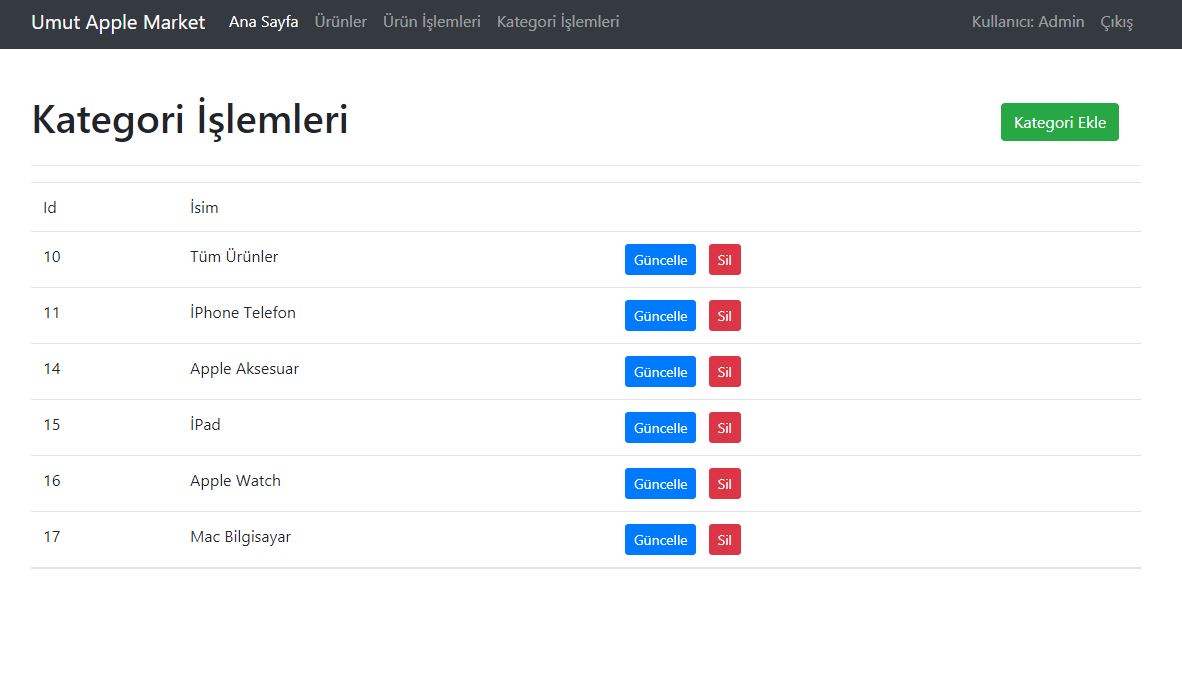
* Email: admin@admin
* Password: admin@admin

Bunlarla giriş yaptığımızda önümüze gelecek olan sayfa şu şekildedir.



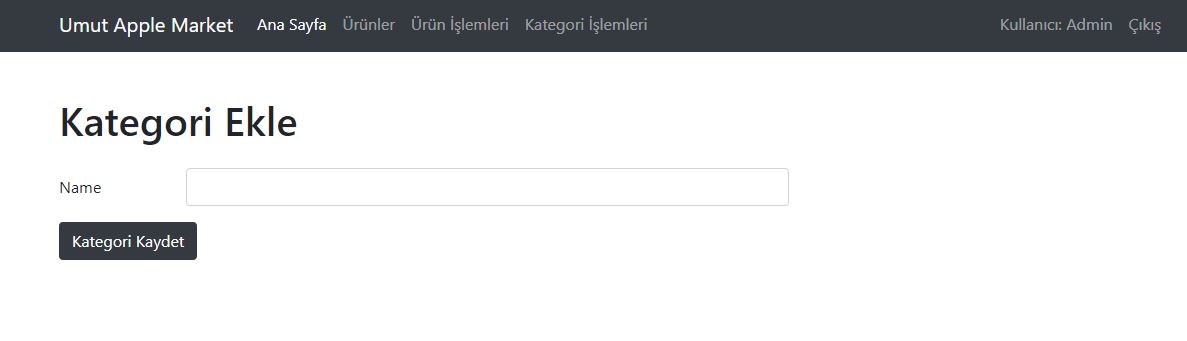
Şekil 27 Admin Ana Sayfa

Admin kullanıcıda üyeden farklı olarak sepet ve siparişlerim kısmı bulunmamaktadır. Bunun yerine adminin görevi olana ürün işlemleri ve kategori işlemleri bulunmaktadır. Öncelikli olarak kategori işlemlerine bakalım. Bunun için navbardan kategori işlemleri linkine tıklıyoruz.



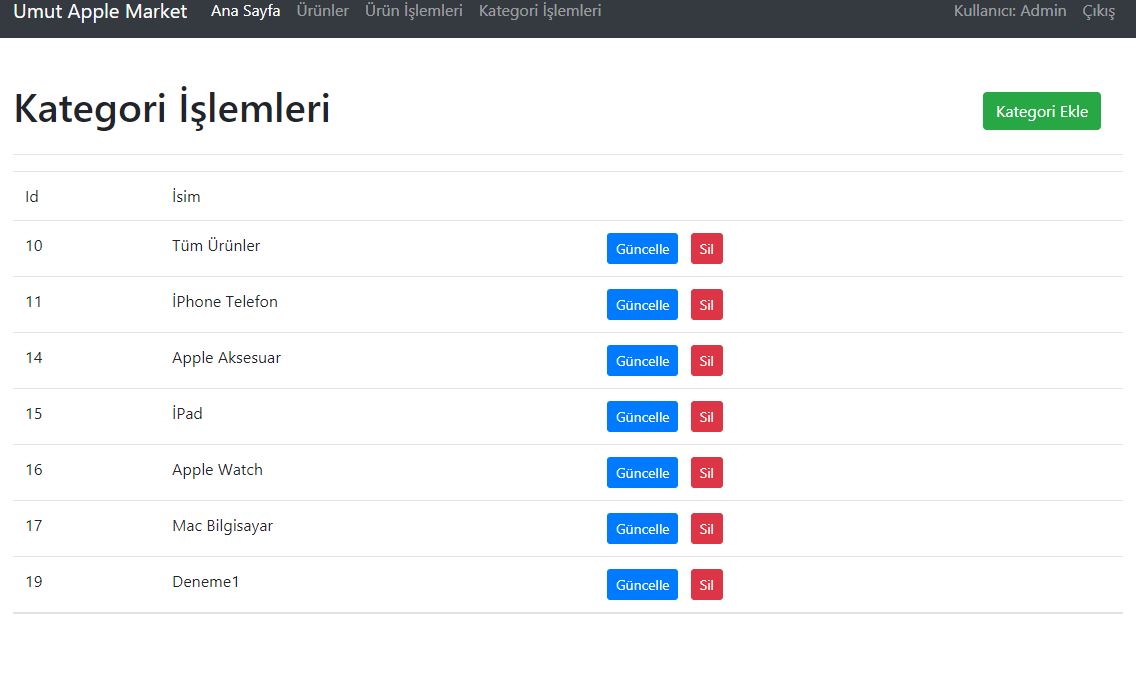
Şekil 28 Admin Kategori İşlemleri Sayfası

Karşımıza sitenin sahip olduğu kategorilerin olduğu, bu kategorileri silip güncelleye bileceğimiz iki butonla beraber yeni kategori ekleyebileceğimiz bir buton çıkıyor. Biz öncelikle yeni bir kategori ekleme işlemi yapalım. Bunun için ilk önce üstteki buton olan kategori ekleye tıklıyoruz.



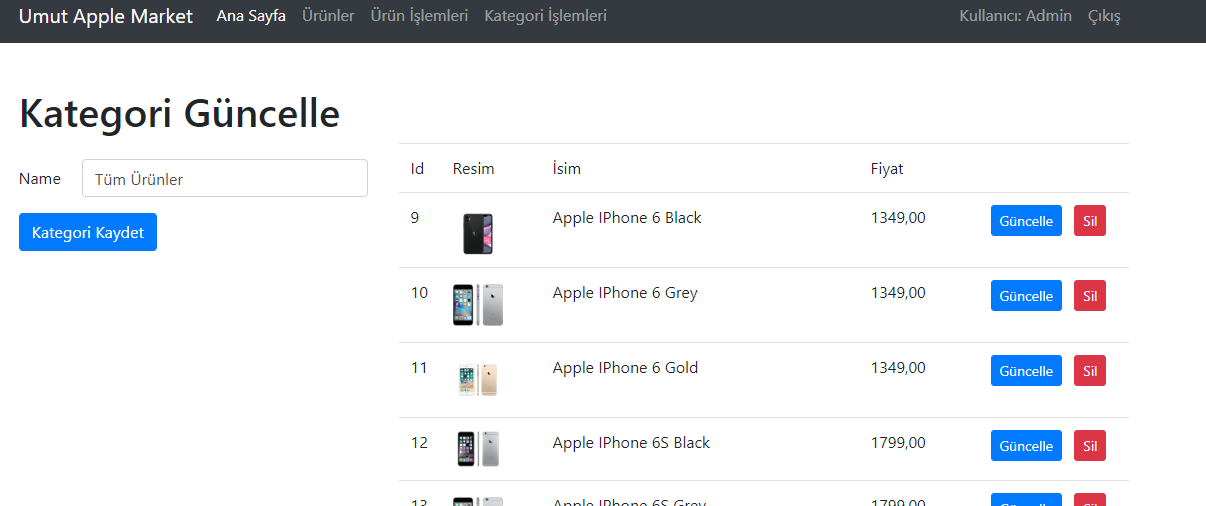
Şekil 29 Admin Kategori Ekle Sayfası

Karşımıza üstteki gibi bir sayfa açılıyor. Burada bizden yeni eklenecek olan kategoriyi istemektedir. Buraya Deneme1 yazıp kategoriyi kaydet diyelim.



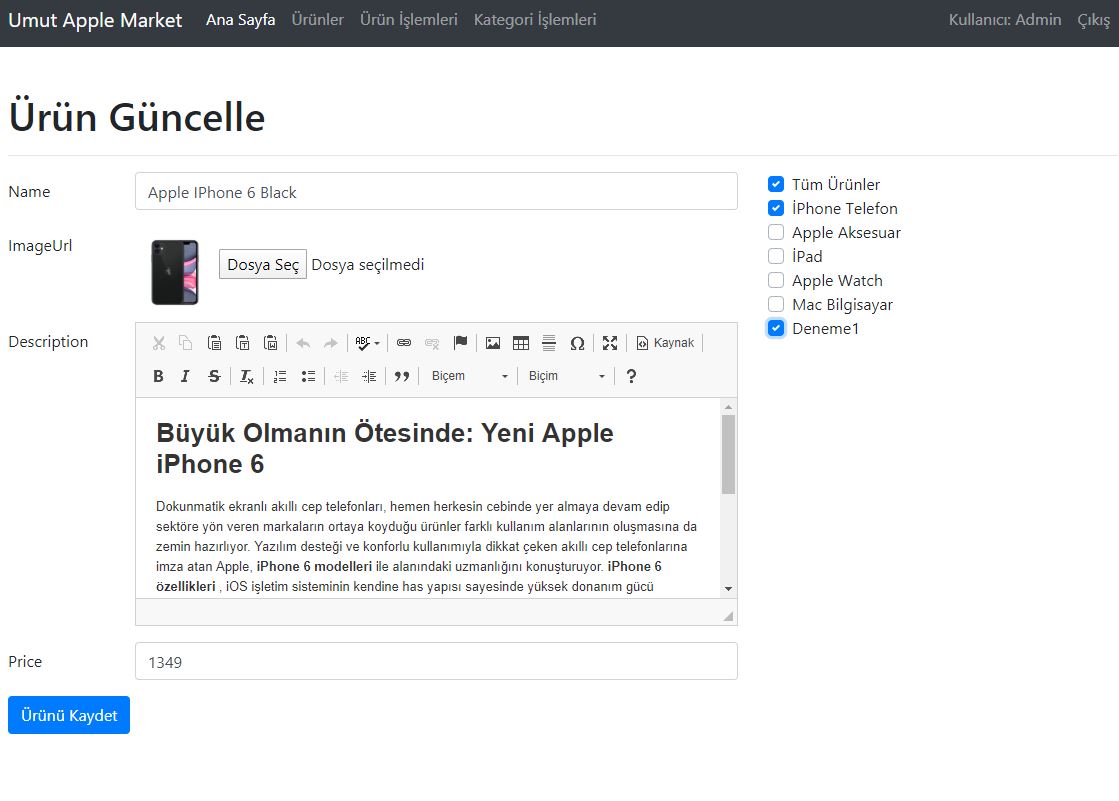
Şekil 30 Admin Kategori Eklendi

Evet, kategorimizin geldiğini görmekteyiz. Şimdi tüm ürünler kategorisinin düzenle kısmına girelim ve buradaki ürünlerden bir kaçını Deneme1 Kategorisine ekleyelim. Bunun için tüm kategoriler kategorisine güncelle diyelim ve aşağıdaki sayfaya ulaşalım.



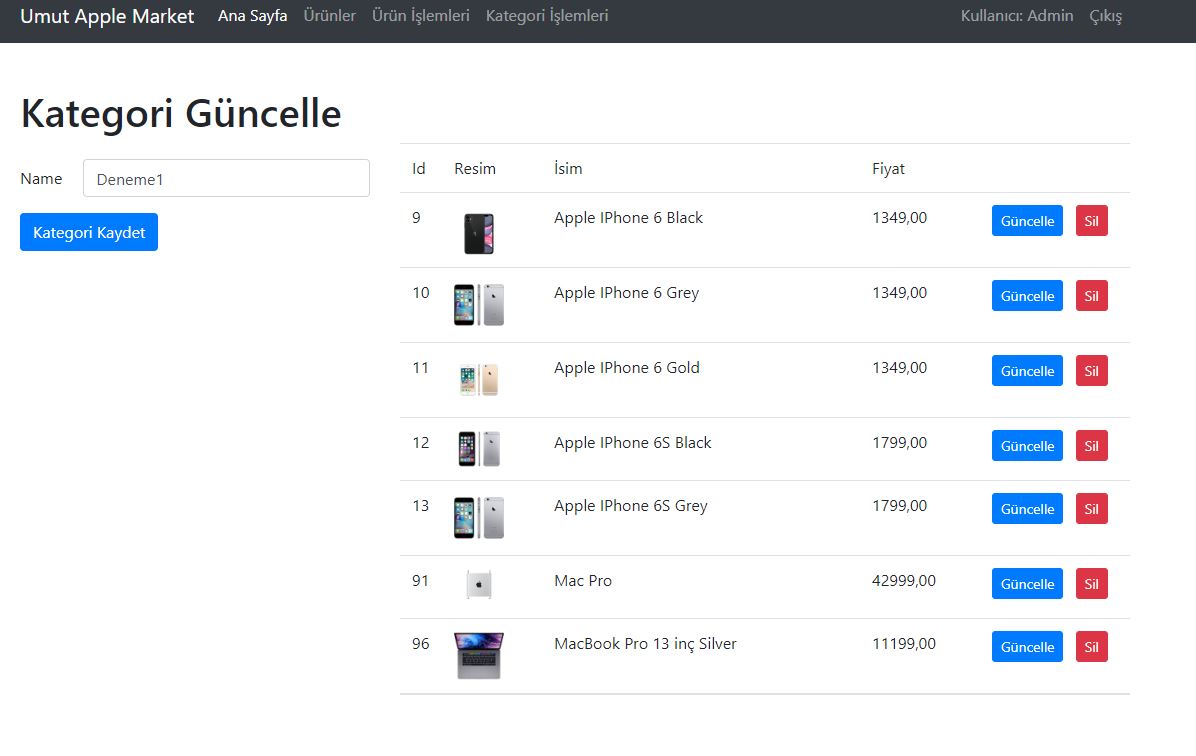
Şekil 31 Admin Kategori Güncelle Sayfası

Karşımıza gelen ekran kategori güncelle ekranıdır. Bu ekranı daha sonra Deneme1 için inceleyeceğiz. Şimdi birkaç ürüne tıklayıp kategoriye ürün nasıl atanmaktadır görelim. 9 Id’li ürün olan Apple IPhone 6 Black için bu işlemi görelim.



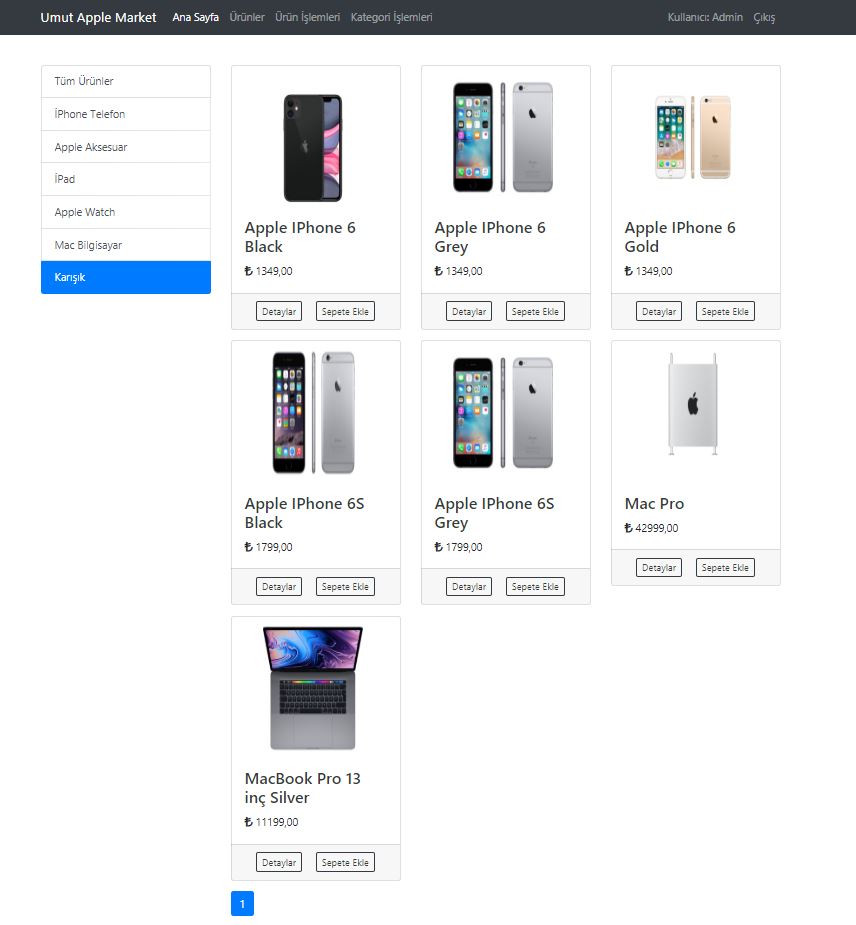
Şekil 32 Ürüne Kategori Eklemesi

Güncelleye bastığımızda karşımıza ürünü ve kategorisini güncelleyebileceğimiz bir sayfa açılmaktadır. Sağ tarafta gördüklerimiz kategorilerimiz. Biz burada Deeme1i işaretleyip ürünü kaydet diyeceğiz ve bu işlemi birkaç ürün için yapacağız. Daha sonra Deneme1 için silme ürün güncelleme ve ürünler sayfasında ürünlerin geldiğini tek tek kontrol edeceğiz. Ürünlerimizi ekledikten sonra Deneme1 sayfasına girip nasıl göründüğünü test edelim.



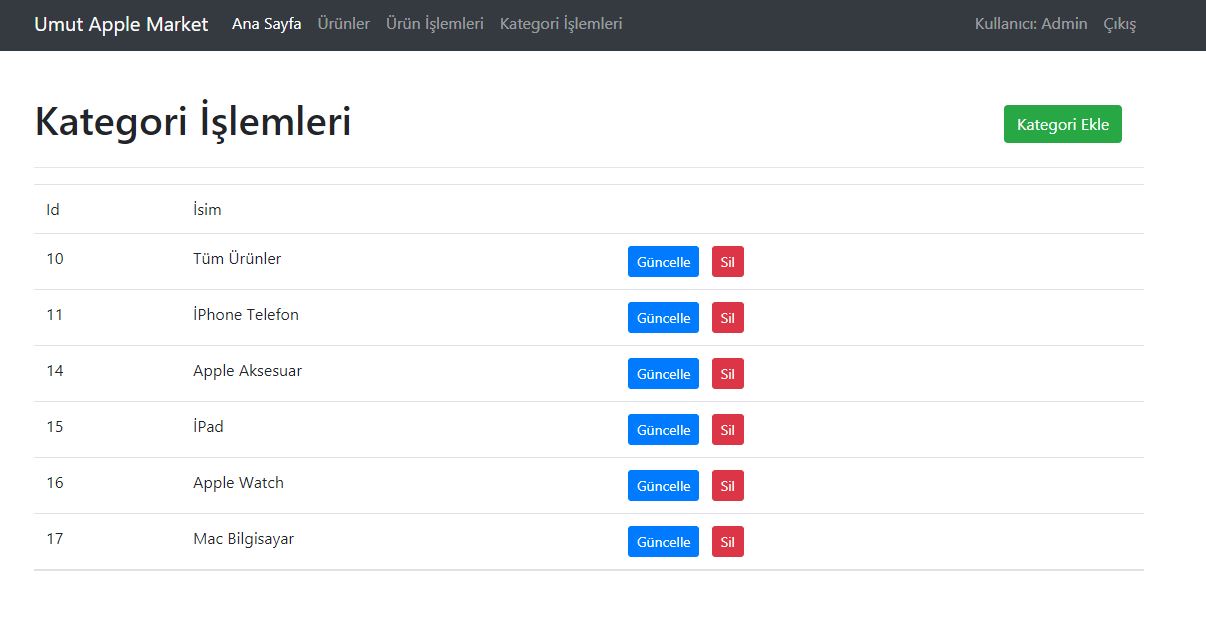
Şekil 33 Admin Ürüne Kategori Eklendi

Evet, eklediğimiz ürünler görünüyor şimdi kategorimizin ismini Karışık olarak güncelleyip ürün sayfasında görüntüleyelim. Bunun için ilk önce sol üstteki Name alanından kategorimizin ismini Karışık olarak güncelleyeceğiz. Daha sonra navbardan ürünler sayfasında sağ taraftaki menüden karışığa geleceğiz. Şimdi işlemi gerçekleştirip ürünlerimizi görelim.



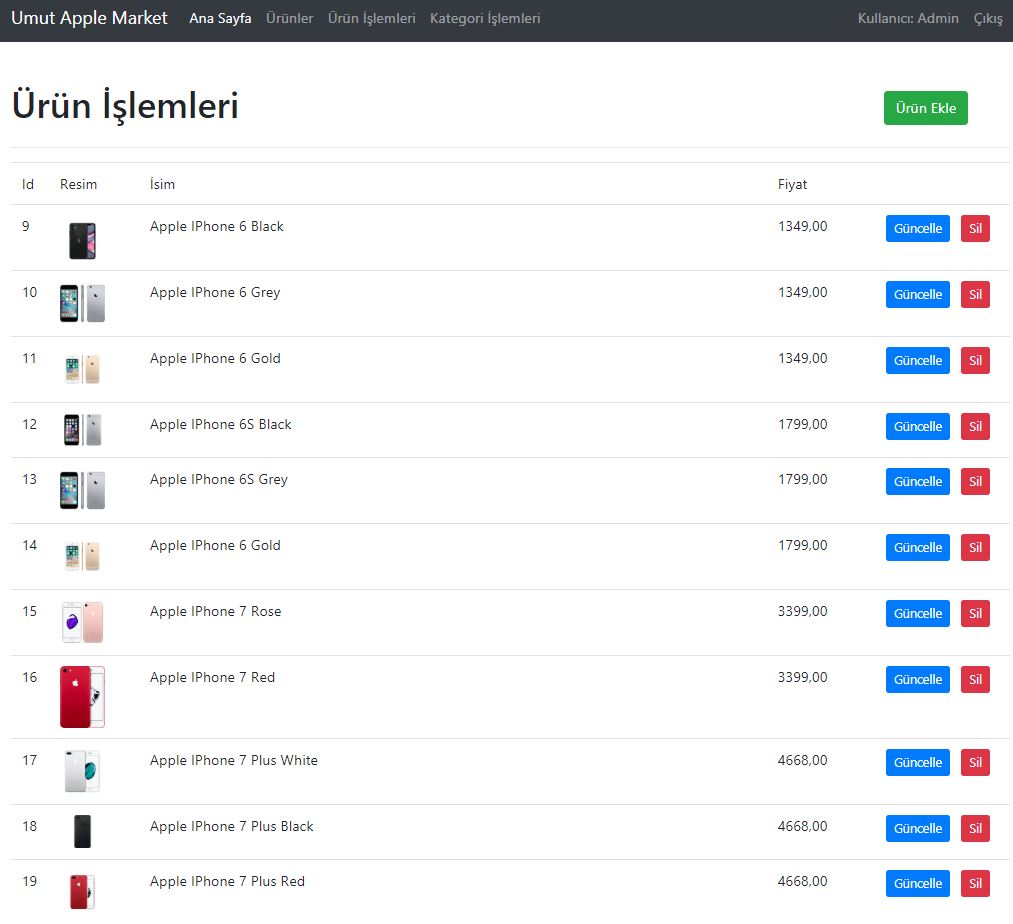
Şekil 34 Admin Kategori Menü Kontrol

Kategorimizi güncelledik, ekledik, ürün ekledik. Şimdi ise kategoriyi tamamen silme işlemini deneyelim ve Karışık kategorisini tamamen silelim. Bunun için kategori işlemleri linkine tıkladıktan sonra açılan sayfada Karışık kategorisinin sil butonuna tıklıyoruz.



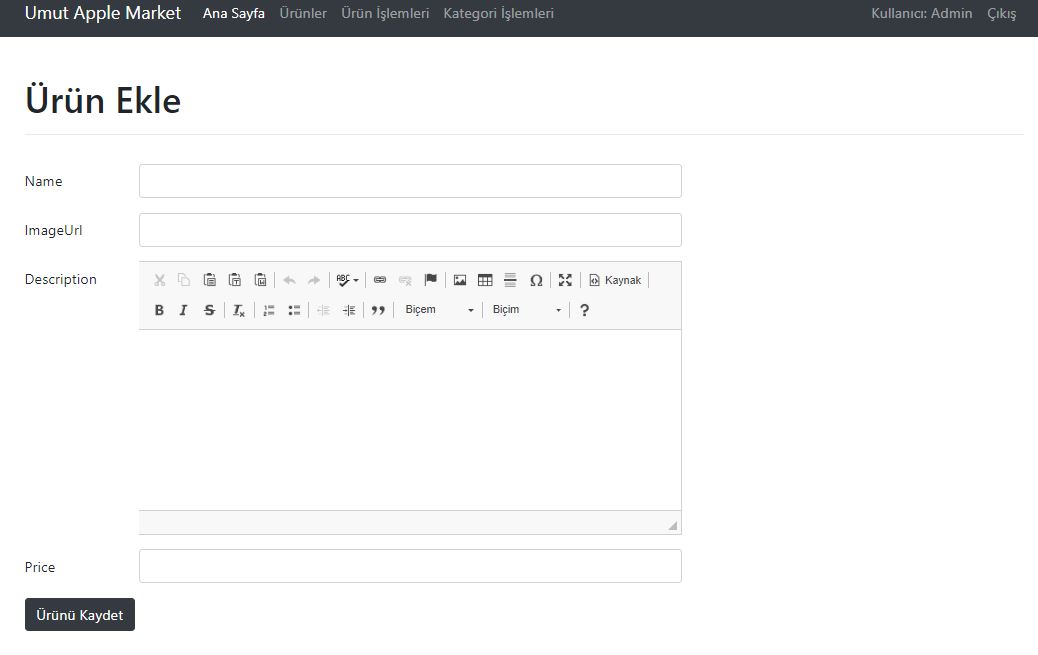
Şekil 35 Admin Kategori Sil

Evet, bu işlemimizin de başarıyla sonuçlandığını gördük şimdi son olarak ürün eklemeyi, silmeyi ve güncellemeyi deneyeceğiz. İlk önce ürün işlemleri linkine navbar aracılığıyla ulaşıyoruz. Ve karşımıza çıkan sayfa aşağıdaki gibi görünmektedir.



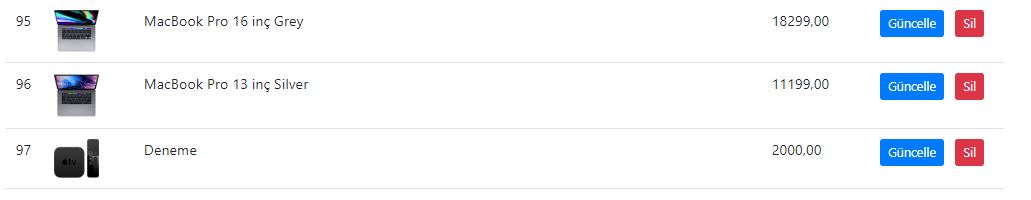
Şekil 36 Admin Ürün İşlemleri

Burada bütün ürünler listelenmektedir. Buradan herhangi bir ürünü ekleyebilir, güncelleyebilir ya da silebiliriz. İlk önce ürün eklemeyi deneyelim. Sağ üst köşedeki ürün ekle butonuna bastığımızda gelen sayfa şu şekildedir.



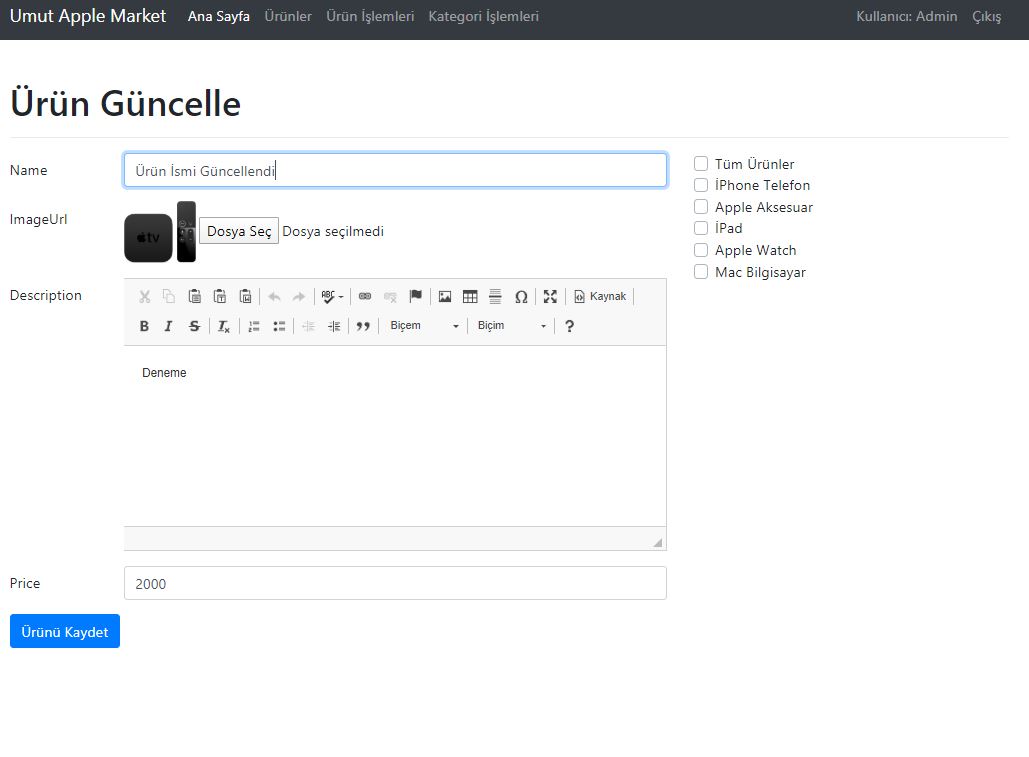
Şekil 37 Admin Ürün Ekle

Sırasıyla bütün alanları doldurmalıyız. Name alanına ürünümüzün adını yazacağız. Şimdilik biz bu işlemi deneme amaçlı yaptığımızdan dolayı ekleyeceğimiz ürünün ismine deneme diyelim. Ürünün açıklamasına da deneme yazalım. Price (Fiyat) kısmına 2000 yazalım. Son olarak ImageUrl (resim url’i) kısmına wwwroot/img klasörünün altındaki herhangi bir resmin url’ini koyalım. Eğer yeni resim eklemek istersek şuan için bu klasöre resmi ekleyip url’ini buraya yapıştırmamız gerekecek şekilde kodlanmıştır uygulamamız. Sonra ürünü kaydet’e tıklayıp ürün sayfamızın en altında ürünümüzün geldiğini görelim.



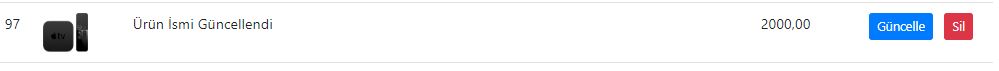
Şekil 38 Admin Ürün Eklendi

Ürünümüzün geldiğini gördük. Şimdi ismini güncelle kısmına girerek değiştirelim.



Şekil 39 Admin Ürün Güncelle

Evet, bu sayfada ekle sayfasından farklı olarak ImageUrl’i wwwroot klasöründen seçiyoruz. Bunun için dosya seçe tıklayıp açılan pencereyi kullanıyoruz. Ekstra olarak sağ tarafta ürünün kategorisini de güncelleyebiliyoruz. Ürün ismini “Ürün İsmi Güncellendi” şeklinde yaptıktan sonra ürünü kaydede tıklayarak ürünler sayfasından değiştirdiğimiz ürünün güncellenip güncellenmediğini kontrol ediyoruz.



Şekil 40 Admin Ürün Güncellendi

Ürünümüz güncellenmiş durumda. Son olarak sil butonuna basıp silindiğini de görüyoruz. Admin kısmı içinde işlemlerimizi bitirdik.

# **SONUÇ**

C# ile Asp.net Core teknolojisi kullanılarak bir web site ve bu web sitenin yönetimi için basit bir kontrol yönetim sistemi yapılmıştır. Site için Apple Market senaryosu uygulanmıştır. Sitede ana sayfa, ürünler, kategoriler, roller, e-mail aktivasyonu, üyelik sistemi, ödeme ve sipariş geçmişi bulunmaktadır.

Projemiz için Microsoft.EntityFramework.Core paketini yükleyerek Asp.net Core teknolojisini aktifleştirdik.

Veri tabanımız için Microsoft.EntityFramework.Core.Design ve Microsoft.EntityFrameworkCore.SqlServer paketinden yararlanılmıştır. Bu paketin yetenekleri sayesinde Visual Studio’nun verilerimizi görmek amaçlı localimizde kullanmamıza izin verdiği SQL Server Object Explorer üzerinde görebileceğimiz bir veri tabanı oluşturduk ve bağlantılarını sağladık. Veri takibimizi buradan gerçekleştirdik.

Kullanıcı tablolarımız bunların veri tabanında tutulması için gerekli yapıyı Microsoft.AspNetCore.Identity paketinden gelen yapıyla oluşturduk. Bu bize birçok tablo oluşturdu. Gerekli olanlarını kullanacağımız modelleri oluşturduk ve bu tabloların üzerine verilerimizi yazdık. Rol özelliğini de bu paketten gelen bir tablo üzerinden gerçekleştirdik.

Ödeme sistemi için IyzipayCore paketini projemizin WebUI katmanına ekledik. Bu sayede sitelerinden sağladıkları api-key ile projemize ödeme sistemi dahil ettik. Aldığımız verileri pakette istenilen değişkenlere ekledikten sonra bize özel apimiz ile gerekli fonksiyonu yazarak işlemimizi gerçekleştirdik.

E-mail onay sistemi içinde SendGrid paketini projemizin WebUI katmanına ekledik. Bu sayede sitelerinden sağladıkları api-key ile projemize e-mail sistemini dahil ettik. Sitemize üye olduğumuz e-postadan sitemizden kullanıcıya ait e-postaya istediğimiz maili gönderebildik.

**KAYNAKLAR**

# [1] .Net Core(Web Api) Middleware ile Authentication Kontrolü

<https://medium.com/@hakantopuz/net-core-middleware-ile-authentication-kontrol%C3%BC-web-api-423e4ccc59bd>

[2] Asp.Net Mvc Nedir? Ne İşe Yarar?

<https://medium.com/@mustafakaraca_51610/asp-net-mvc-nedir-ne-i%CC%87%C5%9Fe-yarar-3327a54c36f7>

[3] C# Repository Design Pattern(Repository Tasarım Deseni)

<https://www.gencayyildiz.com/blog/c-repository-design-patternrepository-tasarim-deseni/>

[4] MVC Code First Nedir ve Code First İle Veri Tabanı Modelleme Nasıl Yapılır?

<https://www.mshowto.org/mvc-code-first-nedir-ve-code-first-ile-veri-tabani-modelleme-nasil-yapilir-bolum-20.html>

[5] Entity Framework Context Sınıfı Nedir? Ne İş Yapar?

<https://snanalagoz.wordpress.com/2018/12/07/entity-framework-ders3-context-sinifi-nedir-ne-is-yapar/>

[6] Asp.NET Core Identity – Identity Altyapısı Kurulumu

<https://www.gencayyildiz.com/blog/asp-net-core-identity-identity-altyapisi-kurulumu-ii/>

[7] Entity Framework Nedir?

<https://medium.com/t%C3%BCrkiye/entity-framework-nedir-bff3943bec72>

[8] SQL Server Data Tools (SSDT) Nedir?

<http://www.mesutx.com/sql-server-data-tools-ssdt-nedir/>

[9] SendGrid E-mail Servis Sağlayıcı

<https://sendgrid.com/>

[10] Iyzico Ödeme Servis Sağlayıcı

<https://sandbox-merchant.iyzipay.com/auth>

[11] Boostrap Dokümantasyon

<https://getbootstrap.com/docs/4.4/getting-started/introduction/>

[12] Entity Framework Core araçları başvurusu- .Net CLI Dokümantasyon

<https://docs.microsoft.com/tr-tr/ef/core/miscellaneous/cli/dotnet>

[13] Sıfırdan İleri Seviye E-Ticaret Sitesi Yapma

<https://www.udemy.com/course/sifirdan-ileri-seviye-php-pdo-e-ticaret-web-sitesi-yapma/>

[14] Uygulama Geliştirerek C# Öğrenin: A’dan Z’ye Eğitim Seti

<https://www.udemy.com/course/sifirdan-ileri-seviye-csharp-programlama/>

[15] Komple ASP.NET Web Geliştirme Eğitimi 2019

<https://www.udemy.com/course/angular-6-ve-aspnet-core-mvc-bastan-sona-proje-gelistirme/>

[16] E-Ticaret Kursu: Katmanlı Mimari ile Core MVC Dersleri

<https://www.udemy.com/course/aspnet-core-mvc-kursu/>

[17] C# ile OOP temelleri: Adım Adım Katmanlı Mimari

<https://www.udemy.com/course/csharp-ile-adim-adim-katmanli-mimari/>

[18] Entity Framework Code First ile Grid.MVC

<https://www.udemy.com/course/entity-framework-code-first-ile-gridmvc/>

[19] Sıfırdan Bootstrap 4 Öğrenme

<https://www.udemy.com/course/sifirdan-bootstrap-4-dersleri-bootstrap-egitimi/>

[20] Sıfırdan Temel HTML Öğrenin

<https://www.udemy.com/course/sifirdan-profesyonel-html-ogrenin/>

[21] AspNet Core Inversion Of Control(IoC) / Dependency Injection

<https://www.udemy.com/course/aspnet-core-inversion-of-control-ioc-dependency-injection/>

[22] Mvc Ders 1 Giriş Başlamadan Önce İzlenmesi Gereken Seriler

<https://www.youtube.com/watch?v=ROMDAwb0cOM&list=PLKnjBHu2xXNNRPqfdZC6hNmJKOqIIpqNj>

[23] What is MVC | What is Asp.Net MVC | What is MVC in .Net | MVC Tutorial for Beginners in .net C#

<https://www.youtube.com/watch?v=fN0PZMI5VhU&list=PLaFzfwmPR7_JuVN71I9pEpN8JadDTh0rg>

[24] Bootstrap 4 ile Responsive Web Tasarımı Dersleri 2017

<https://www.youtube.com/watch?v=Qo3Uzem0Jv4&list=PL-Hkw4CrSVq-xwLUQ0-OIA46mKwYeaxXR>

[25] What is Entity Framework

<https://www.youtube.com/watch?v=Z7713GBhi4k&list=PL6n9fhu94yhUPBSX-E2aJCnCR3-_6zBZx>

[26] ASP NET Core Tutorial

<https://www.youtube.com/watch?v=4IgC2Q5-yDE&list=PL6n9fhu94yhVkdrusLaQsfERmL_Jh4XmU>

[27] How to Send Email with C# and Twilio SendGrid

<https://www.youtube.com/watch?v=ddSymc0hE0A&t=153s>

[28] Php ile Iyzico Ödeme Sistemi Entegrasyonu

<https://www.youtube.com/watch?v=e5Yse6fWpHM>

ÖZGEÇMİŞ

**KİŞİSEL BİLGİLER**

|  |  |
| --- | --- |
| Adı Soyadı | : Umut Demiray |
| Doğum Tarihi ve Yeri | : 01.07.1997 / İSTANBUL |
| Yabancı Dili | : İngilizce |
| E-posta | : umutdemiray97@gmail.com |

**ÖĞRENİM DURUMU**

|  |  |  |  |
| --- | --- | --- | --- |
| **Derece** | **Alan** | **Okul/Üniversite** | **Mezuniyet Yılı** |
| Lisans | Bilgisayar Müh. | Düzce Üniversitesi | 2020 |
| Lise | Sayısal | Orhangazi Anadolu İHL | 2015 |