BLP4210 PLATFORM BAĞIMSIZ PROGRAMLAMA

**SORU #1**

Graphical user interface

Description automatically generated with low confidenceIlk olarak Django Framework’u kurmak için terminale ‘python -m pip install Django’ komutunu yazıyoruz.

Ardindan projeyi olusturmak istediğimiz dizinin içinde ‘django-admin startproject proje\_adi’ komutunu çalıştırıyoruz.

Graphical user interface

Description automatically generated with medium confidence

Projemiz oluşmuş olduk, çalıştırmak için terminale ‘python manage.py runserver’ komutunu yazıyoruz.

Text

Description automatically generated

Projeyi ilk çalıştırdığımızda Django bize projenin sorunsuz bir şekilde çalıştığını gösteren hazır bir websitesi taslağı görüntülüyör.

Şuan projenin ana dosyalarını oluşturmuş olduk fakat uygulamanın çalışacağı back-end ve front-end kodların yazılması gerekek uygulama dosyalarını oluşturmadık. Bunun için terminale ‘python manage.py startapp app\_name‘ komutunu yazıyoruz.

Graphical user interface, text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Oluşturduğumuz bu app’in settings.py dosyası içinde Django’ya tanıtmamız lazım. Bunun için INSTALLED\_APPS kısmında bunu belirtiyoruz.

Text

Description automatically generated

Daha sonra oluşturduğumuz bu uygulamanın urlini proje dosyamız olan ecom içinde bulunan urls.py içerisinde tanımlamamız lazım.

Text

Description automatically generated

Yukarıdaki görüntüde urlpatterns içerisinde 2. Satırda bulunan path uygulamamızın ana sayfasını temsil ediyor. Yani kullanıcı websitesini ilk açtığı anda herhangi bir dizin belirtmeden anasayfaya ulaşmış olacak.

Text

Description automatically generated

Burada iste app dizininde oluşturduğumuz urls.py dosyasında index sayfasının yolunu, arkaplanda kullanacağı fonksiyonu ve adını tanımladık.

Bu sayfanın html kısmını oluşturmak için Templates isimli bir klasör oluşturuyoruz ve sayfanın ön yüzünü burada tanımlayacağız.

A screenshot of a computer

Description automatically generated with low confidence

Bu html sayfalarında kullanmak için oluşturacağımız static dosyalarını ise (css, js, img) static isimli bir klasör içinde tanımlıyoruz.

Uygulamaya resim upload işlemi gerçekleşeceğinden ve oluşturduğumuz bu static dizininin çalışabilmesi için settings.py dosyasında urlini belirtmemiz lazım.

Text

Description automatically generated

Template kısmında sayfanın görünümünü ayarladıktan sonra veritabanı tablolarımızı oluşturmamız lazım. Django’da veritabanı işlemleri models.py dosyasından yapılıyor. Örneğin Product tablosunun oluşturulması bu şekilde:

Text

Description automatically generated

Daha sonrasında bu veritabanı tablolarına ulaşmak için importlamamız yeterli.

Django’da back-end işlemleri views.py dosyasında gerçekleşiyor. Örneğin /products.html isimli bir sayfam olsun ve bu sayfada ürünlerim listelenecek olsun. Bunun için views.py dosyasında bununla ilgili bir fonksiyon yazıyorum. Fonksiyon içinde yapmak istediğim işlemleri gerçekleştirdikten sonra o html sayfasını renderlıyorum.

Text

Description automatically generated

Yukarıdaki örnekte seçilen kategorideki ürünlerin bilgilerini önce ‘category = Category.objects.all()‘ diyerek daha sonra ‘products = Product.objects.filter(category\_id=id)’ diyerek ilişkisel bir şekilde aldım. Bu aldığım bilgileri html dosyasına alabilmek için context şeklinde yollamamız gerekiyor.

Context ile yolladığımız bilgileri html dosyasında şu şekilde alabiliyorum:

Text

Description automatically generated

Graphical user interface

Description automatically generated

Web sitesinin ana sayfası bu şekilde görünüyor. Ödevler için kısıtlı bir sürem olduğundan website taslağını hazır kullandım. Fakat arkaplanda çalışan kodların büyük bir kısmı bana ait.

Kullanıcı kategori menüsünden bir kategori seçtiğinde o kategoriye ait ürünlerin listelendiği sayfaya gidebiliyor.

Graphical user interface, website

Description automatically generated

Diğer sayfalarda olduğu gibi login ve signup için de bir html dosyası oluşturdum ve urls.pyda djangoya tanıttım. Html dosyasında basit bir form oluşturduktan sonra forms.py dosyasında custom bir kullanıcı kayıt formu oluşturuyorum.

Graphical user interface, application

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Ardından views.py dosyasında oluşturduğum model’ı ve formu importluyorum. Signup\_view fonksiyonunda ise signup.html dosyasında bulunan formdaki bilgileri SignUpForm şeklinde alıp UserProfile tablosuna kaydediyorum. Bu formun kontrollerini(kullanıcı adı, parola uyumu gibi işlemler) Django kendisi sağlıyor ve bir hata meydana geldiğinde kullanıcıya uyarı gösterebiliyor. Eğer kullanıcının girdiği bilgiler uygunsa veritabanına kaydedildikten sonra sisteme otomatik olarak login oluyor.

Text

Description automatically generated

Login fonksiyonunda da benzer şekilde kullanıcının girdiği bilgiler User tablosu ile control edilip sisteme girişi sağlanıyor.

Graphical user interface, application

Description automatically generated

Text

Description automatically generated

Kullanıcı sisteme giriş yaptıktan sonra Login butonu yerine profil butonlar geliyor. Bunun kontrolü ise template dosyalarında if-else komutları ile gerçekleştirilebiliyor.

Graphical user interface, application

Description automatically generated

Graphical user interface

Description automatically generated with low confidence

Kullanıcı ürünü incele butonuna bastığında o ürünün detay sayfasına gidebiliyor.

Graphical user interface, website

Description automatically generated

Text

Description automatically generated

Sepete ekle butonuna bastığı anda ise addtocart() fonksiyonu çalışıyor ve seçilen ürünün idsi ve miktarı gibi bilgileri alıp ShopCart tablosuna ekliyor.

Text

Description automatically generated

Text

Description automatically generated

Burada kullanıcı girişi şartı var eğer kullanıcı giriş yapmadan bir ürünü sepete eklemeye çalışırsa login sayfasına yönlendiriliyor.

Kullanıcının sepete eklediği ürünler ise sepet sayfasında bu şekilde görüntüleniyor:

A screenshot of a computer

Description automatically generated

Sağ tarafta bulunan x işaretine basıldığında deletefromcart() fonksiyonu çalışıyor ve o ürünü o kişinin sepetinden(tablodan) siliyor.

Text

Description automatically generated

A screenshot of a computer

Description automatically generated

Admin sayfasından yeni kategori, ürün ekleme gibi işlemler yapılabiliyor.

A screenshot of a computer

Description automatically generated with medium confidence

A picture containing text, screenshot, indoor, screen

Description automatically generated