



**Teknoloji
Fakültesi**

DERS:

BİLGİSAYAR PROGRAMLAMA 2

PROJE ADI:

MAEDRE - ÖZEL TASARIM ÜRÜNLER İÇİN İLAN TEKLİF SİSTEMİ

GİTHUB LİNKİ:

<https://github.com/EnesZulfikaroglu/Maedre>

HAZIRLAYAN:

ENES ZÜLFİKAROĞLU 170418033

PROJE ÖZETİ

Özel tasarım ürünler için ilan-teklif sitesi. Özel tasarım ürün isteyen kullanıcıların ilan verdiği, ürünü sağlayabilecek kullanıcıların da bu ilanlara teklif yapabileceği bir sisteme sahip olan internet sitesidir.

GİRİŞ

1. Projenin Çözmeyi Amaçladığı Sorun:

Günümüzde çok fazla internet üzerinden satış sitesi bulunmaktadır. Ama özel tasarım ürün isteyen kullanıcılar için hazırlanmış bir site bulunmamakta. Bu sebeple özel tasarım ürün isteyen kullanıcılar, bu işi yapan firmaların hepsiyle ayrı ayrı görüşmek zorunda kalıyor. Bu da kullanıcıların zamanını harcayıp fazla uğraşmalarına sebep oluyor. Projenin çözmeyi amaçladığı sorun budur.

2. Kullanılan Yöntem ve Teknolojiler:

Bu projenin Back-End kısmı python tabanlı olan Django Web Framework ile geliştirilmiştir. Database işlemleri için SQLite database kullanılmaktadır. Aynı zamanda projenin Front-End kısmında html – css – Bootstrap teknolojilerinden yararlanılmaktadır.

3. Kullanılan Yöntemin Detaylı Bilgisi:

Django hakkında detaylı bilgi vermek gerekirse, bir araya getirilmiş uygulamalardan oluşmaktadır. Bu uygulamalar büyük projeler için modülerlik sağlar ve projenin karmaşıklığını büyük ölçüde azaltır. Aynı zamanda tekrar kullanılabilirlik sağlar. Bu uygulamalarda kullandığımız dosyalar aşağıda açıklanmıştır.

3.1 urls.py Dosyaları:

Proje urls.py isimli, URL mapini oluşturduğumuz python dosyasından başlar. Bu dosyada belirli URL' ler geldiği zaman, views.py kısmında oluşturduğumuz, çalıştırılacak python methodlarını gösteririz.

3.2 views.py Dosyaları:

views.py isimli python dosyası, asıl python işlemlerini yaptığımız dosyadır. Proje kapsamında bir URL' e gidildiğinde, o URL' in karşıladığı views.py methodu çağrılır. Belirli işlemler yapıldıktan sonra, views.py methodları render() fonksiyonu ile bir html sayfasını renderlar. Böylece kullanıcı ilgili sayfa gösterilir.

3.3 models.py Dosyaları:

models.py isimli python dosyası, veritabanı modellerini oluşturduğumuz python dosyasıdır. Burada oluşturduğumuz modeller, SQL sorgularına dönüştürülür ve veritabanı tablosu olarak kullandığımız veritabanına kaydedilir.

3.4 forms.py Dosyaları:

forms.py isimli dosyada, proje kapsamında kullanacağımız formlar oluşturulur. Bu formlar views.py methodları tarafından render edilecek html dosyalarına gönderilir ve kullanıcıya ulaştırılır.

3.5 settings.py Dosyası:

settings.py isimli dosya, Django tarafından hazır olarak oluşturulur. Projenin ayarları bu dosyadan yapılır. Örnek olarak; oluşturulan uygulamaların import edilmesi, kullanılacak static dosyalarının dizininin ayarlanması gibi işlemler bu dosyadan yapılır.

3.6 Templates Dosyaları:

Bu dosya dizininde projede kullanılan html dosyaları oluşturulur.

3.7 Uygulamalar

Proje içinde oluşturulan uygulamalar; **ilan**, **user** ve **message** uygulamalarıdır. 3 uygulama bulunmaktadır.

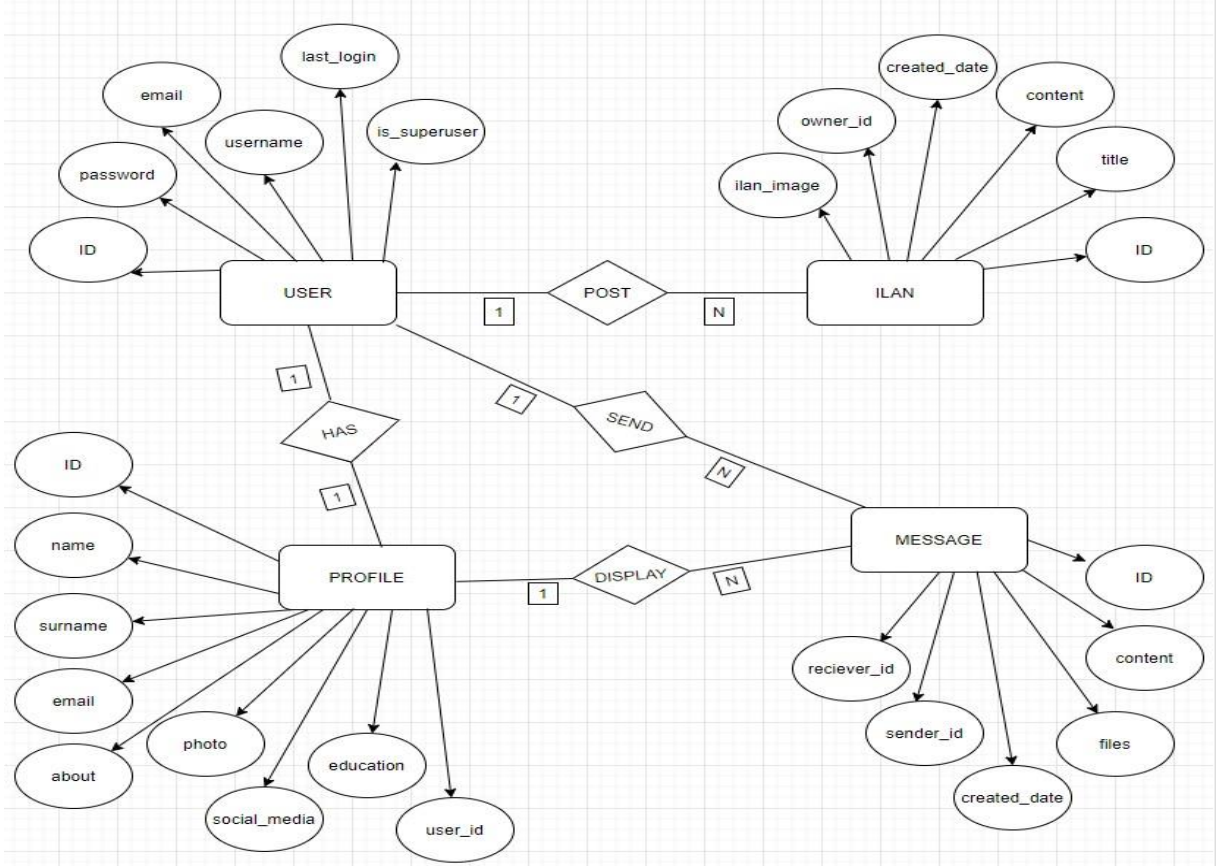
ilan uygulamasında; ilan oluşturma, ilanları görüntüleme gibi ilan işlemleri bulunmaktadır.

user uygulamasında; kullanıcı kaydı, giriş yapma, profil işlemleri gibi kullanıcı işlemleri bulunmaktadır.

Message uygulamasında; mesaj gönderme, gelen kutusunu görme, giden kutusunu görme gibi mesaj işlemleri bulunmaktadır.

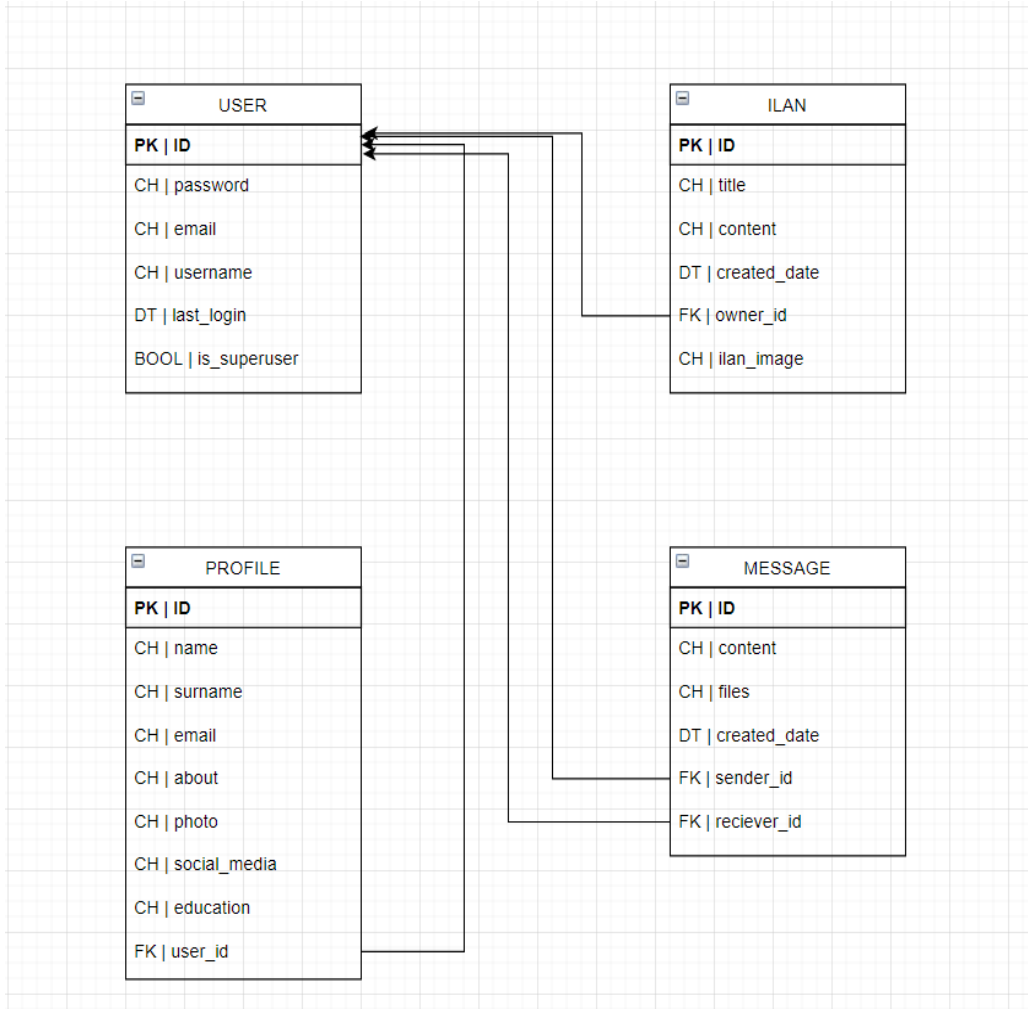
VERİTABANI YAPISI

1. Varlık-ilişki Şeması:



Projenin varlık-ilişki şeması yukarıdaki görselde verilmiştir. Veritabanı, 5 tablodan oluşmaktadır. Bunlar **user**, **ilan**, **profile**, **message** ve **comment** tablolarıdır. Tabloların sütunları ve diğer tablolarla ilişkileri yukarıdaki görselde belirtilmiştir.

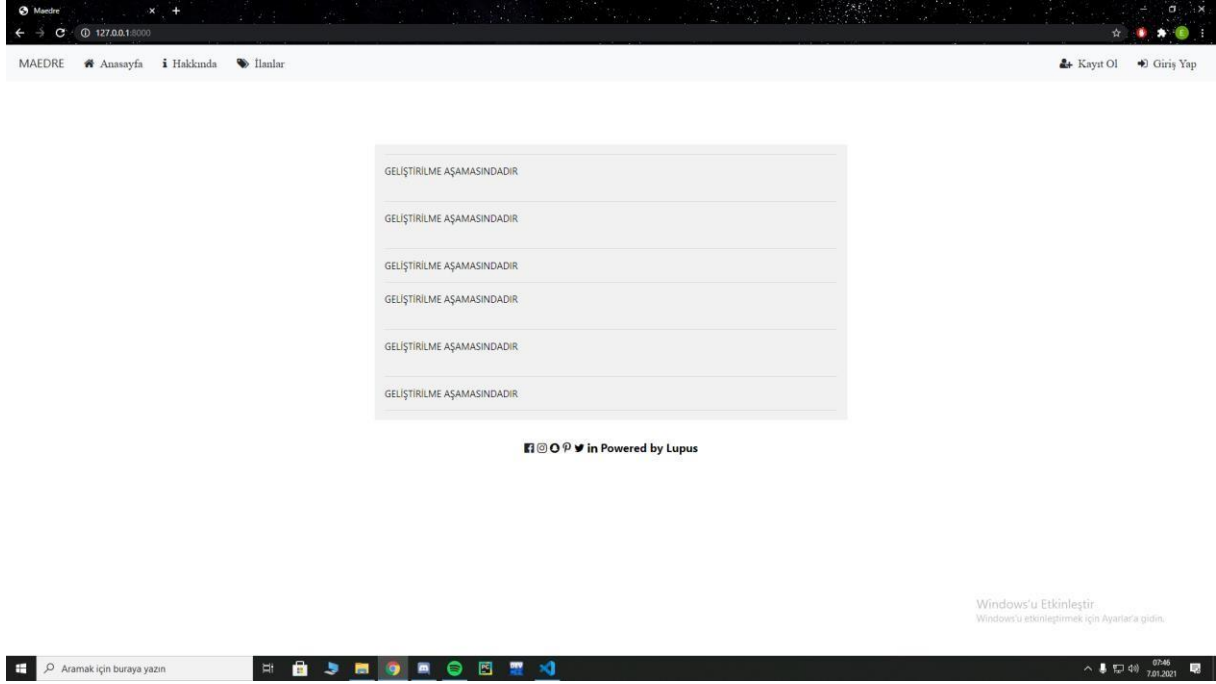
2. Genel Veritabanı Şeması:



- **Message** tablosu “sender_id” ve “reciever_id” foreign keyleri ile **user** tablosuna bağlanır.
- **İlan** tablosu “owner_id” foreign key’ i ile **user** tablosuna bağlanır.
- **Profile** tablosu “user_id” foreign key’ i ile **user** tablosuna bağlanır.
- **Comment** tablosu “comment_user_id” foreign key’ i ile **user** tablosuna bağlanır.
- **Comment** tablosu “ilan_id” foreign key’ i ile **İlan** tablosuna bağlanır.

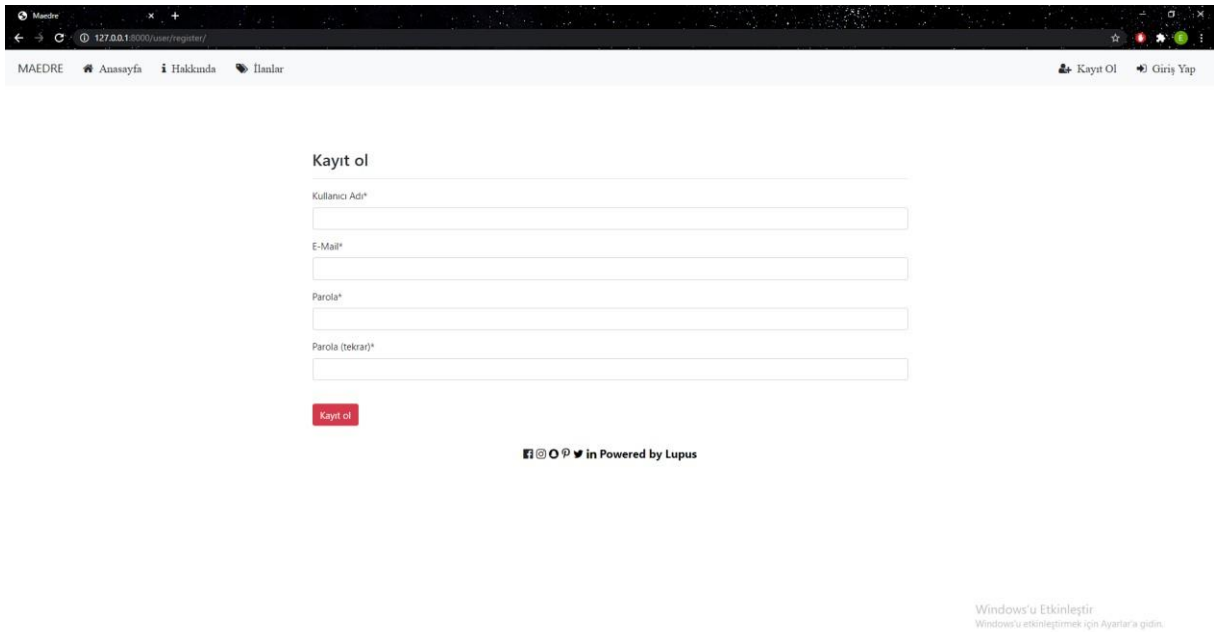
KULLANICI ARAYÜZLERİ

1. ANA SAYFA:



Projeme internet adresi aracılığı ile erişim sağladığımızda bizi karşılayan ilk sayfadır. Henüz giriş yapılmadığı için bu sayfa üzerinden kayıt ol butonu ve giriş yap butonu da görünmektedir. Uygulamaya giriş yaptığımızda bu butonların yerini uygulamaya kaydolmuş kişilerin erişebileceği butonlar alacaktır.

2. KAYIT OL SAYFASI:



Kayıt ol sayfası, /user/forms.py dosyasında oluşturulmuş RegisterForm isimli formu kullanıcıya sunan sayfadır. Form doldurulduktan sonra /user/views.py dosyasında “register” isimli methodda kullanıcı ve default oluşturulan profili veritabanına kaydedilir. Daha sonra bu profil düzenlenebilir. Kaydolan kullanıcı giriş yapılmış bir şekilde ana sayfaya yönlendirilir.

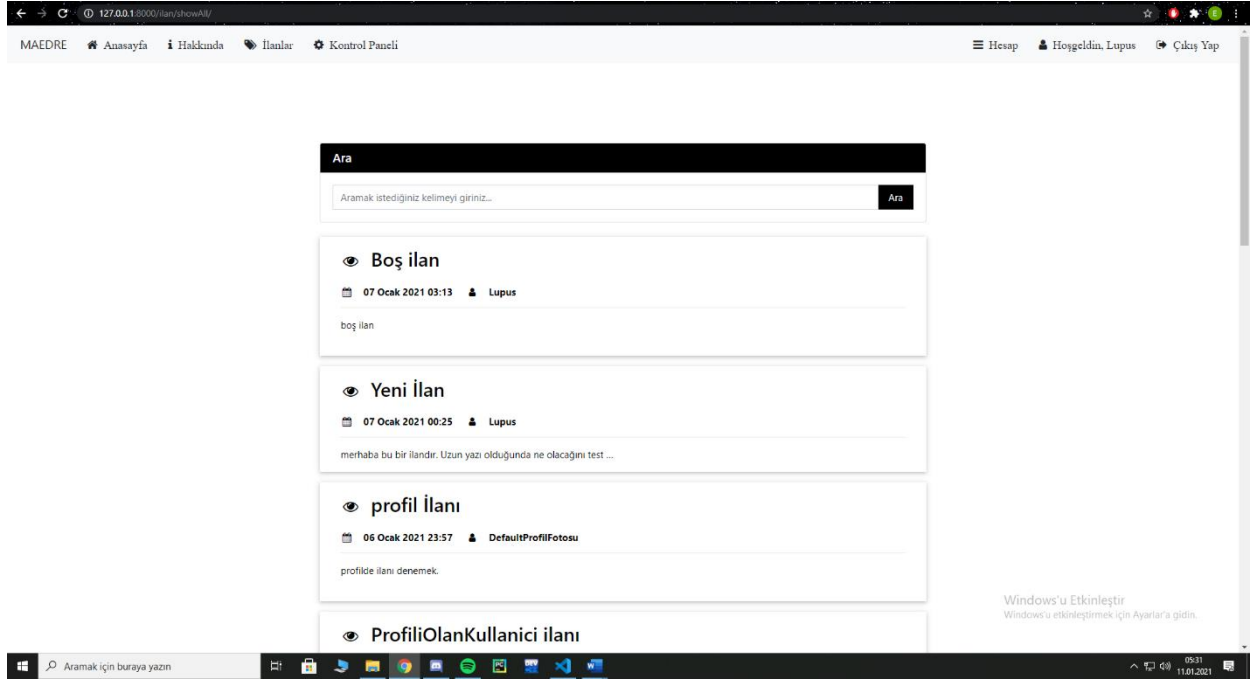
3. GİRİŞ YAP SAYFASI:

Giriş Yap sayfası, /user/forms.py dosyasında oluşturulmuş LoginForm isimli formu kullanıcıya sunan sayfadır. Form doldurulduktan sonra /user/views.py dosyasında “login” isimli methodda kullanıcının girdiği bilgiler kontrol edilir. Yanlış bilgi girilmişse bir hata mesajı döndürülür. Bilgiler doğru ise giriş yapılmış bir şekilde ana sayfaya yönlendirilir.

4. HAKKINDA SAYFASI:

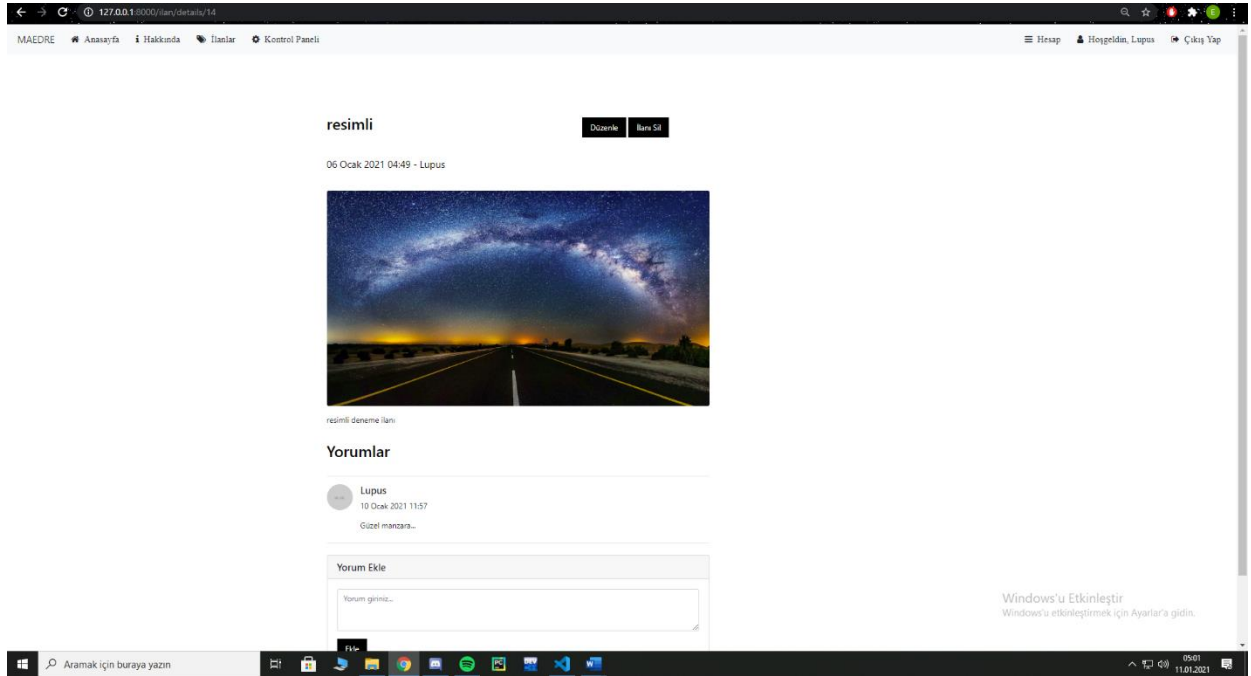
Bu sayfada geliştirmekte olduğum projenin amacıyla ve kullanılan teknolojilerle ilgili açıklamaların yer aldığı kısa bilgilendirmeler gösterilmektedir.

5. İLANLAR SAYFASI:



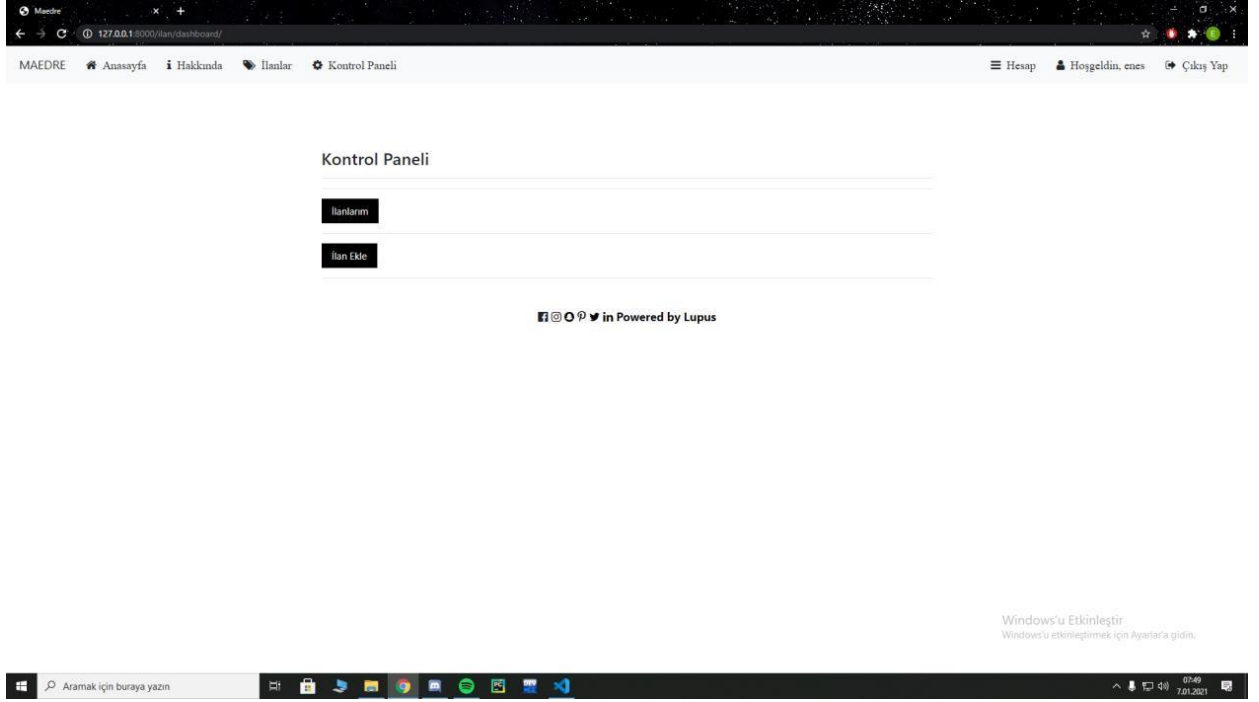
İlanlar sayfası, verilen ilanların yer aldığı sayfadır. Bu sayfada ilanlar listelenir ve ilan detaylarına gitmek için ilana tıklanması gerekmektedir. Arama aygıtı bulunmaktadır. Arama aygıtına girilen kelime URL üzerinde “keyword” ismi ile eklenir. Daha sonra sadece başlığında aranan pattern bulunan ilanlar gösterilir.

6. İLAN DETAY SAYFASI:



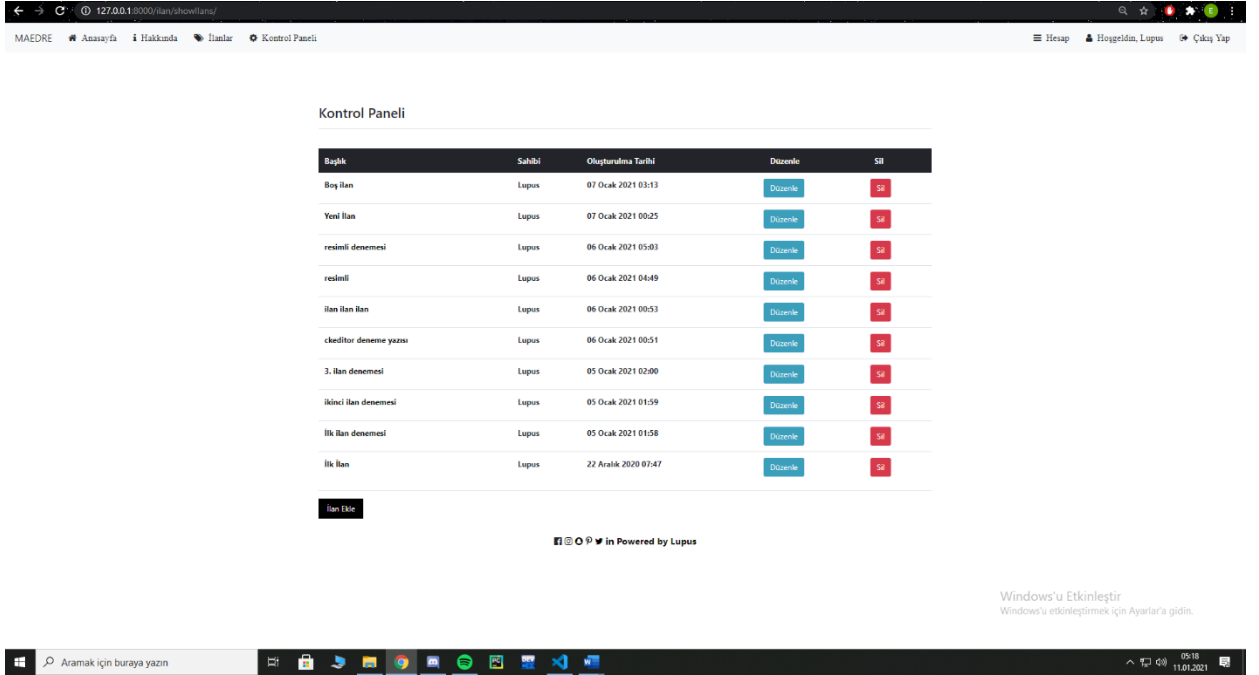
İlan detay sayfası, ilgili ilanın gösterildiği sayfadır. Dinamik URL kullanılmıştır. Bu sayfada ilanları görüntülemenin yanında yorum da yapılabilir. Eğer ilanın sahibi görüntülüyorsa düzenleme ve silme butonları da gösterilir. Eğer başka bir kullanıcının ilanı görüntüleniyorsa, “Mesaj Gönder” butonu gösterilir. Böylece kullanıcılar ilgisini çeken ilanın sahibine direkt mesaj gönderebilir.

7. KONTROL PANELİ SAYFASI:



Kontrol paneli sayfasında; giriş yapan kullanıcıya kendi ilanlar tablosunu görüntülemek, ya da ilan ekleme işlemleri için butonlar gösterilir.

8. İLANLARIM SAYFASI:



İlanlarım sayfasında; kullanıcı ilanlar tablosunu görür. Bu sayfa aracılığıyla ilan düzenleme sayfasına ya da ilan ekleme sayfasına erişebilir, silmek istediği ilanları silebilir.

9. İLAN EKLE SAYFASI:

İlan Ekle

Başlık*

Content*

Başlık*

Content*

Görsel Ekle:

Dosya Seç Dosya seçilmedi

İlan Ekle

Windows'u Etkinleştir

İlan Ekle sayfası, /ilan/forms.py dosyasında oluşturulmuş İlanForm isimli formu kullanıcıya sunan sayfadır. Form doldurulduktan sonra /ilan/views.py dosyasında “addİlan” isimli methodda kullanıcının girdiği bilgiler kontrol edilir. Form geçerli değilse hata döndürür. Geçerli ise ilan veritabanına kaydedilir, başarı mesajı döndürülür.

10. İLAN DÜZENLE SAYFASI:

İlan Güncelle

Başlık*

Boş İlan

Content*

Başlık*

Content*

Görsel Ekle:

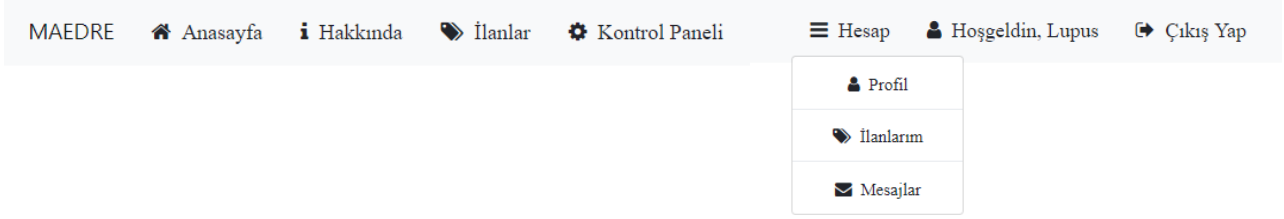
Dosya Seç Dosya seçilmedi

Güncelle

Windows'u Etkinleştir

İlan Düzenle sayfası, dinamik URL ile tanımlanmıştır. /ilan/forms.py dosyasında oluşturulmuş İlanForm isimli formu kullanıcıya eski bilgileri yazılı olarak sunan sayfadır. Form doldurulduktan sonra /ilan/views.py dosyasında “updateİlan” isimli methodda kullanıcının girdiği bilgiler kontrol edilir. Form geçerli değilse hata döndürür. Geçerli ise ilan veritabanında güncellenir, başarı mesajı döndürülür.

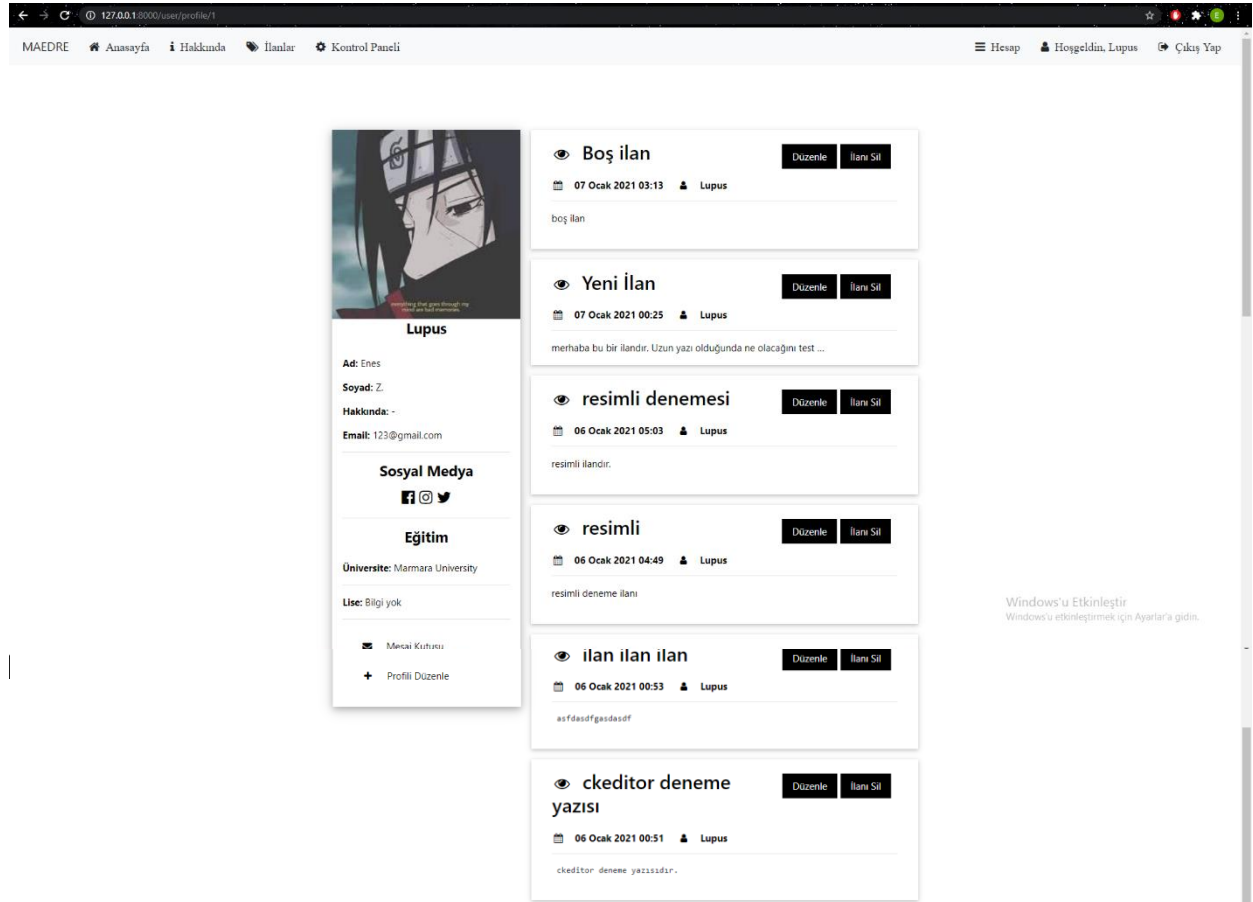
11. NAVBAR DETAYLARI:



Navbar, giriş yapıldı/yapılmadı durumlarına göre dinamik olarak değişir. Giriş yapılmamışsa, Kontrol paneli Hesap, Çıkış Yap gibi butonlar gözükmez. Giriş yapılmışsa, kayıt ol, giriş yap gibi butonlar gösterilmez.

Hesap butonuna tıklandığı zaman aşağı doğru 3 elemanlı bir liste açılır. Bu liste aracılığıyla kullanıcı kendi profiline, ilanlarına ve mesaj kutusuna gidebilir.

12. PROFİL SAYFASI:



Profil sayfası dinamik URL ile tanımlanmıştır. Profil sayfasında kullanıcı hakkında daha önceden kullanıcının belirttiği bilgileri gösterilir.

Eğer profili görüntüleyen kullanıcı profilin sahibi ise verdiği ilanların yanında “Düzenle” ve “İlanı Sil” butonlarını görür. Aynı zamanda bilgilerinin altında da “Mesaj Kutusu” ve “Profili Düzenle” butonları görüntülenir.

Eğer profili görüntüleyen kullanıcı profilin sahibi değilse, bu butonların yerini Mesaj Gönder butonu alır. Kullanıcı, profiline baktığı başka bir kullanıcıya direkt mesaj atabilir.

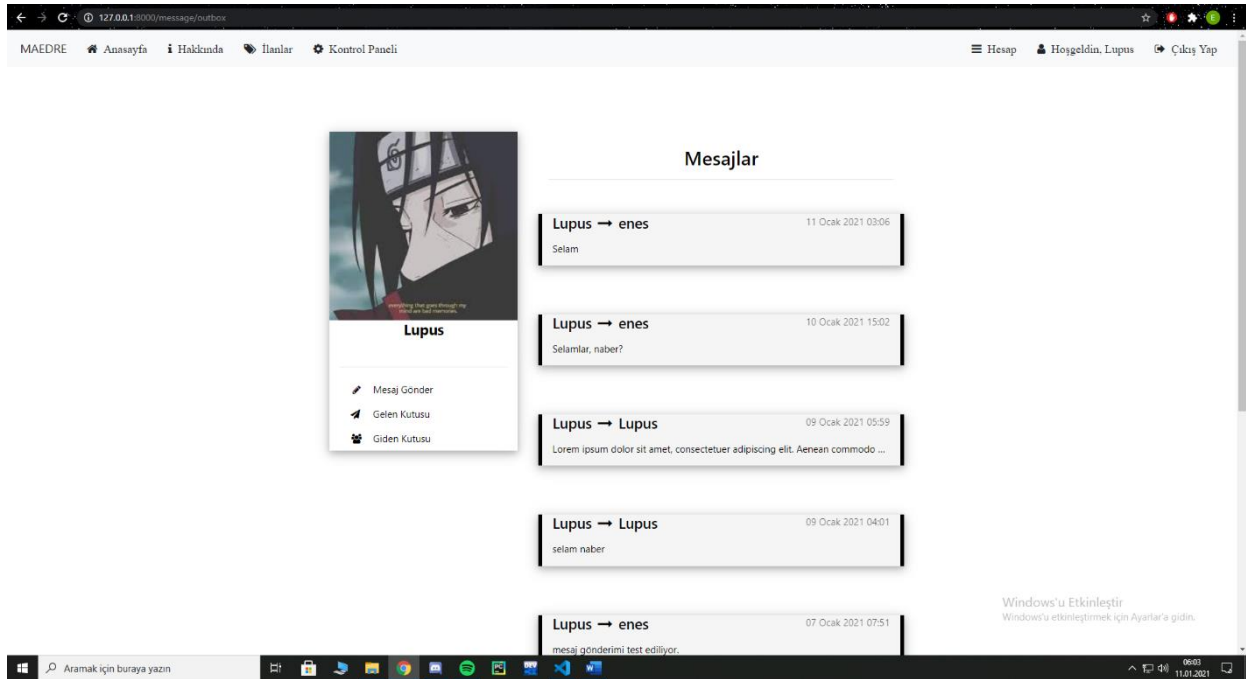
13. PROFİLİ DÜZENLE SAYFASI:

Profil Düzenle sayfası, dinamik URL ile tanımlanmıştır. /user/forms.py dosyasında oluşturulmuş ProfileForm isimli formu kullanıcıya eski bilgileri yazılı olarak sunan sayfadır. Form doldurulduktan sonra /user/views.py dosyasında “updateProfile” isimli methodda kullanıcının girdiği bilgiler kontrol edilir. Form geçerli değilse hata döndürür. Geçerli ise profil veritabanında güncellenir, başarı mesajı döndürülür. Daha önce hiç düzenlenmemiş profillerde default bilgiler bulunur.

14. MESAJLAR – GELEN KUTUSU SAYFASI:

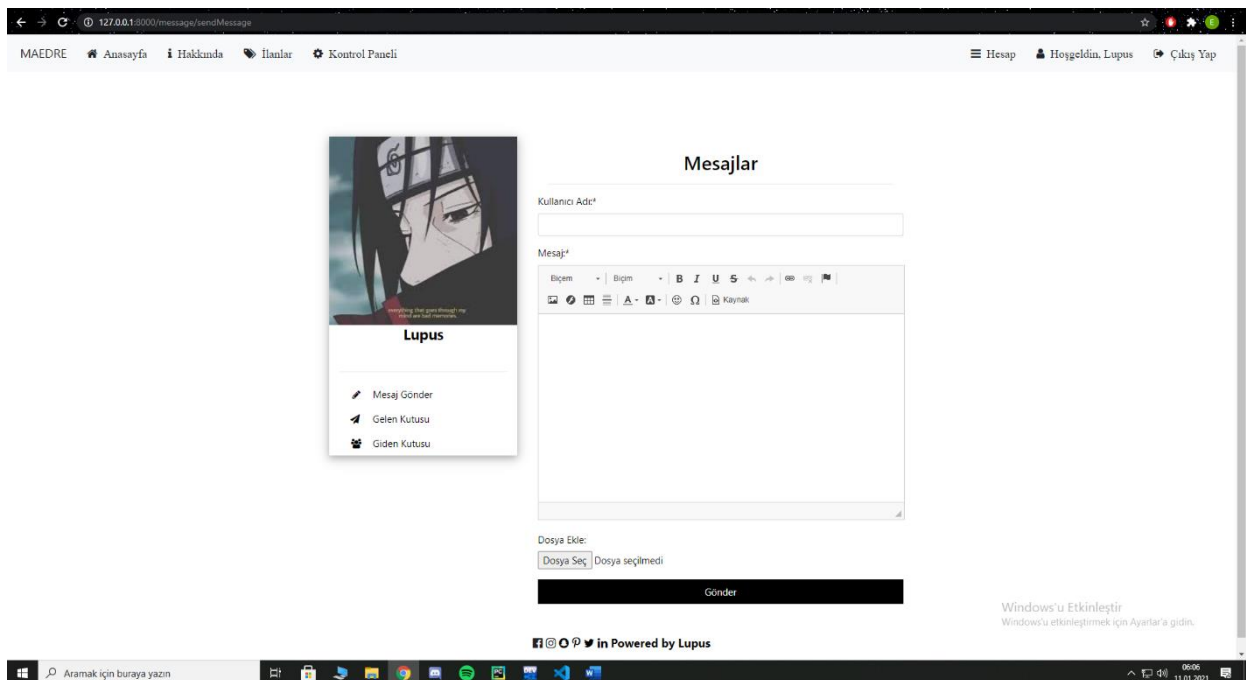
Gelen kutusu sayfasında, kullanıcının aldığı mesajlar gösterilir. message objelerinin reciever_id bilgileri kontrol edilir. Reciever_id değeri giriş yapan kullanıcının id değerine eşit olan mesajlar gösterilir.

15. MESAJLAR – GİDEN KUTUSU SAYFASI:



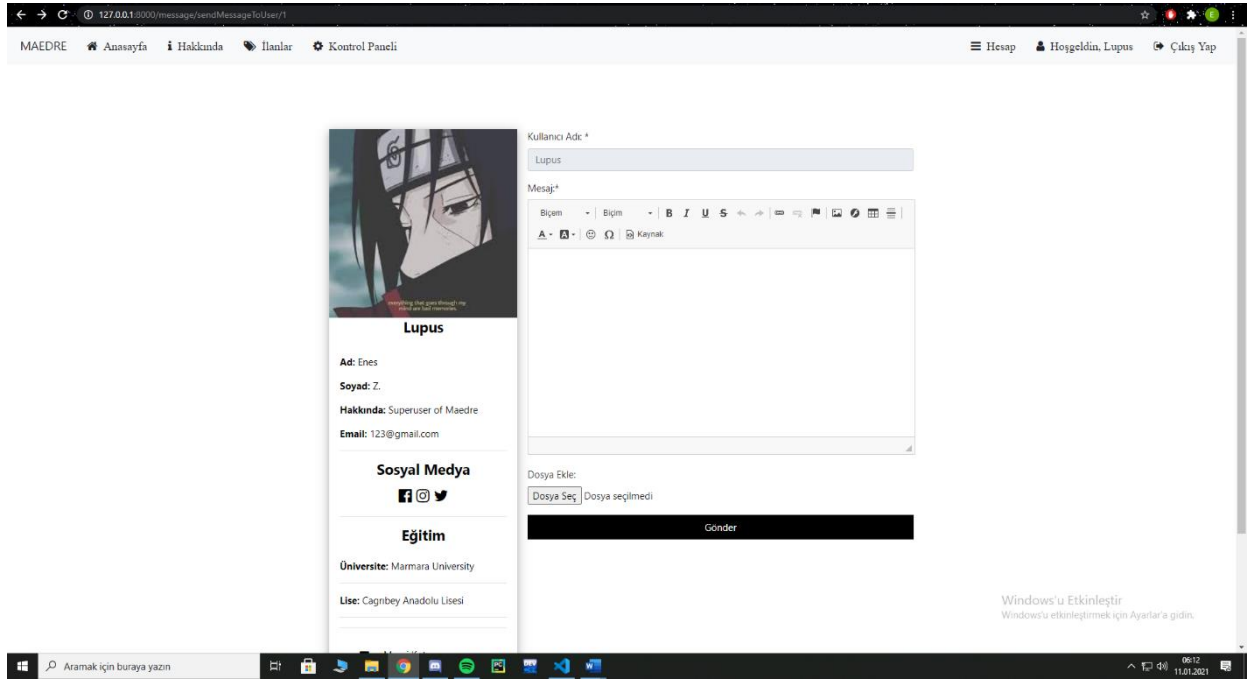
Gelen kutusu sayfasında, kullanıcının gönderdiği mesajlar gösterilir. message objelerinin sender_id bilgileri kontrol edilir. sender_id değeri giriş yapan kullanıcının id değerine eşit olan mesajlar gösterilir.

16. MESAJLAR – MESAJ GÖNDER SAYFASI:



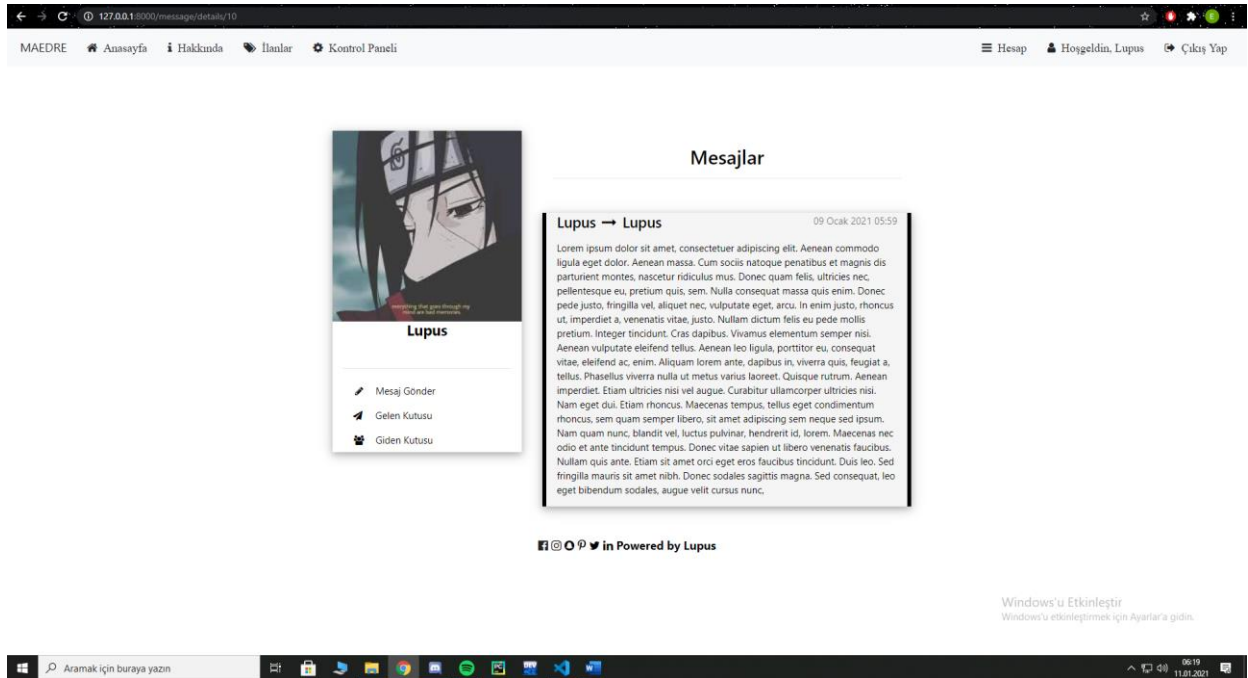
Mesaj Gönder sayfası, sabit URL ile tanımlanmıştır. /message/forms.py dosyasında oluşturulmuş MessageForm isimli formu kullanıcıya sunan sayfadır. Form doldurulduktan sonra /message/views.py dosyasında “sendMessage” isimli methodda kullanıcının girdiği bilgiler kontrol edilir. Form geçerli değilse hata döndürür. Geçerli ise mesaj, kullanıcı ismi girilen kullanıcıya gönderilir. Veritabanına kaydedilir.

17. MESAJLAR – DİNAMİK MESAJ GÖNDER SAYFASI:



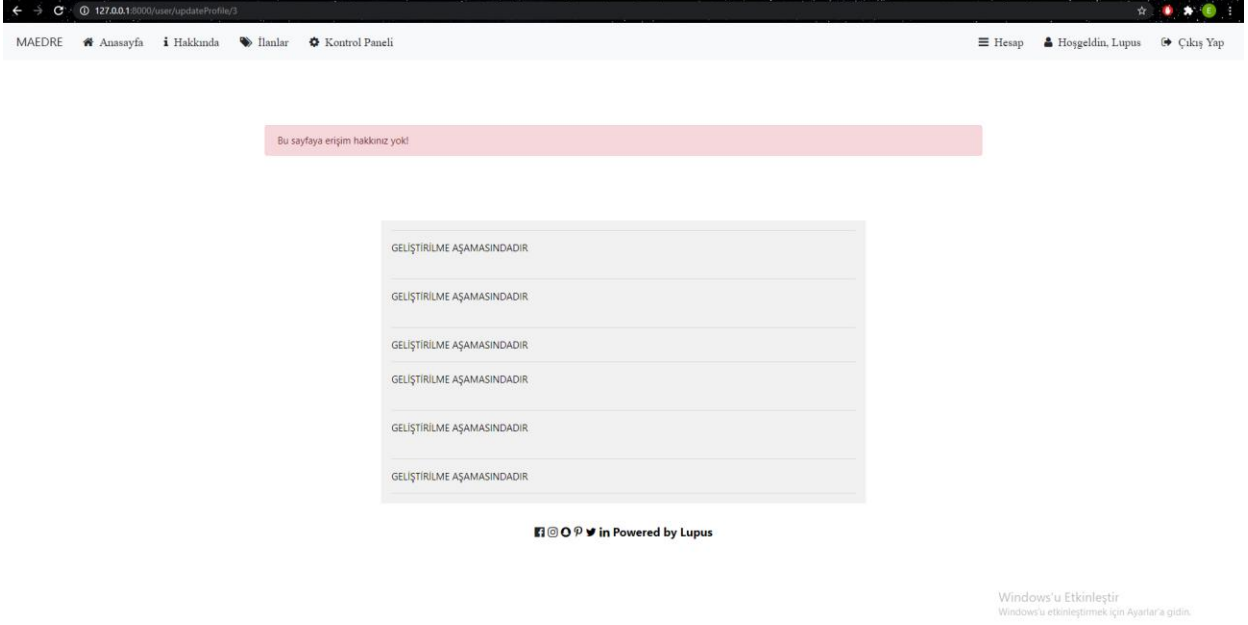
Normal Mesaj Gönder sayfasından farkı, dinamik URL ile tanımlanmasıdır. Bu sayede URL içinde id' si girilen kullanıcının adı, Kullanıcı Adı input bölümüne read-only olarak yazılır. Yani id' si URL içinde girilen belirli bir kullanıcıya mesaj atılırken kullanılır. İlan detaylarının yanında ve profillerde bulunan Mesaj Gönder butonu, dinamik mesaj gönder sayfasına yönlendirir. Böylece kullanıcılar tekrar kullanıcı adı yazmak zorunda kalmaz.

18. MESAJ DETAY SAYFASI:



Bir mesaja tıklandığı zaman, mesajın devamı açılır ve mesaj detay sayfasına geçilir. Dinamik URL olarak tanımlanmıştır. URL' de girilen id değeri, message_id değeri ile aynı olan mesaj gösterilir. Güvenlik açığı oluşmaması amacıyla, giriş yapan kullanıcı, mesajı gönderen ya da alan kullanıcı değil ise mesajı görmesine izin verilmez ve uyarı mesajı gönderilir.

GÜVENLİK AÇIĞI ÖNLEMLERİ



Yapılan bu sitede oldukça fazla dinamik URL kullanılmıştır. Bu durum da kullanışlı olmasıyla birlikte güvenlik açıkları getirir. Önüne geçilen güvenlik açıkları aşağıda listelenmiştir.

1. İlan Güncelleme Önlemi:

İlan güncellemek için girilen URL ***/ilan/updatellan/ilan_id*** şeklindedir. İlanın sahibi olmayan bir kullanıcı, bu URL ile bu sayfaya erişmeye çalışırsa engellenir. Uyarı mesajı gönderilir.

2. İlan Silme Önlemi:

İlan silmek için girilen URL ***/ilan/deletellan/ilan_id*** şeklindedir. İlanın sahibi olmayan bir kullanıcı, bu URL ile izni olmayan bir ilan silmeye çalışırsa engellenir. Uyarı mesajı gönderilir.

3. Mesaj Gizliliği Önlemi:

Bir mesajı görüntülemek için girilen URL ***/message/details/message_id*** şeklindedir. Mesajı alan ya da gönderen kişi olmayan bir kullanıcı, bu URL ile bu sayfaya erişmeye çalışırsa engellenir. Uyarı mesajı gönderilir.

4. Profil Güncelleme Önlemi:

Profil güncellemek için girilen URL ***/user/updateProfile/user_id*** şeklindedir. Profilin sahibi olmayan bir kullanıcı, bu URL ile bu sayfaya erişmeye çalışırsa engellenir. Uyarı mesajı gönderilir.