

Django Projesi İşlev Raporu

1. Proje Genel Bakış

Bu Django tabanlı uygulama, kullanıcıların “ihtiyaç” (Need) ve “teklif” (Offer) gönderebildiği ve eşleştirebildiği bir platformdur.

Başlıca özellikler: - **Kullanıcı yönetimi** ve roller (admin, standart kullanıcı, kurye)

- **REST API** uç noktaları (Django REST Framework)
 - **Gerçek zamanlı bildirimler** (Django Channels ile WebSocket)
 - **Toplu veri işlemleri** (CSV/Excel ile içe/dışa aktarım)
 - **Kurye iş akışları** (isteklerin teslimatı için atamalar)
-

2. Yüklü Uygulamalar

- Django’nin yerleşik uygulamaları:
 - `django.contrib.admin`
 - `django.contrib.auth`
 - `django.contrib.contenttypes`
 - `django.contrib.sessions`
 - `django.contrib.messages`
 - `django.contrib.staticfiles`
 - Üçüncü parti uygulamalar:
 - `rest_framework`
 - `channels`
 - Yerel uygulamalar (proje altındaki):
 - `api`
 - `need`
 - `appuser`
-

3. Veri Modelleri ve İlişkiler

- **Role** (`appuser.models`): Kullanıcı rolleri (admin, standart, kurye) ve izinler.

- **AppUser** (appuser.models): Django **User** ile OneToOne ilişki, **tel**, **address** ve **role** alanlarıyla genişletme.
- **Kind** (need.models): İhtiyaç/teklif kategorisi (örn. gıda, giyim, hizmet).
- **Need** (need.models): Kullanıcının talep ettiği ihtiyaç; **name**, **slug**, **address**, **kind** alanları.
- **Offer** (need.models): Kullanıcının bir **Need** için gönderdiği teklif; **Need** ile ForeignKey ilişkisi.

İlişkiler:

- AppUser → User (OneToOne)
- AppUser → Role (ForeignKey)
- Need → Kind (ForeignKey)
- Offer → Need (ForeignKey)

4. URL Uç Noktaları ve Görünümler

Yol	Görünüm Fonksiyonu	Açıklama
/register/	register_view	Kullanıcı kayıt formu
/login/	login_view	Giriş
/profile/	profile_view	Profil sayfası
/needs/	need_list	İhtiyaç listesi
/needs/create/	need_create	İhtiyaç oluşturma
/needs/<slug:slug>/update/	need_update	İhtiyaç güncelleme
/needs/delete/<slug:slug>/	need_delete	İhtiyaç silme
/offers/	offer_list	Teklif listesi
/offers/create/	offer_create	Teklif oluşturma
/offers/<int:pk>/update/	offer_update	Teklif güncelleme
/offers/delete/<int:pk>/	offer_delete	Teklif silme
/roles/	role_list	Roller listesi (sadece admin)
/roles/create/	role_create	Rol oluşturma
/roles/<slug:slug>/update/	role_update	Rol güncelleme
/roles/delete/<slug:slug>/	role_delete	Rol silme
/import/	import_needs	Toplu içe aktarım (CSV/Excel)
/import/confirm/	import_confirm	İçe aktarımı onaylama
/export/	export_needs	Toplu dışa aktarım (CSV/Excel)
/courier/requests/	courier_request_list	Kurye için bekleyen istekler
/courier/assigned/	my_courier_needs	Kurye atanmış görevler

Yol	Görünüm Fonksiyonu	Açıklama
/api/...	DRF ViewSet'ler	JSON tabanlı CRUD uç noktaları
WebSocket routing (Channels)	—	Gerçek zamanlı bildirim trafiği

5. İşlevselliğin Özeti

1. Kullanıcı Kaydı & Roller:

- Kayıt olurken kullanıcıya **Role** atanır.

- AppUser** profili, iletişim bilgilerini ve rolünü içerir.

2. İhtiyaç & Teklif Süreci:

- Kullanıcı **Need** oluşturur (kategori seçer, detay girer).
- Başka bir kullanıcı **Offer** gönderir ve eşleştirme yapılır.

- Kurye, atanan istekleri teslim etmek üzere görev alır.

3. Yönetici & Roller:

- Adminler yeni **Role** oluşturabilir, izinleri yönetebilir.

- Rol bazlı erişim kontrolleri ile yetkisiz işlemler engellenir.

4. Toplu Veri İşlemleri:

- CSV/Excel ile **Need** içe ve dışa aktarım.

- Dashboard üzerinden işlem istatistikleri.

5. REST API & Gerçek Zaman:

- /api/ yolu altındaki CRUD uç noktaları (DRF).

- WebSocket bildirimleri (Channels) – yeni istek/teklif durumları.

6. Ölçeklenebilirlik & Genişletilebilirlik:

- Modüller **api**, **need**, **appuser** uygulamaları.
- Yeni özellikler kolayca eklenebilir (örn. puanlama, mesajlaşma).

6. Sonraki Adımlar / Öneriler

- Otomatik testler** ekleyin (kayıt, ihtiyaç/teklif akışları, içe/dışa aktarım).

- DRF viewset'lerde **izin kontrolleri** uygulayın.
 - Liste görünümlerinde **select_related/prefetch_related** kullanarak sorgu optimizasyonu yapın.
 - API için **OpenAPI/Swagger** dokümantasyonu oluşturun.
-