🚀 DJANGO PROJESİ OTOMATIK OLUŞTURMA TALİMATNAMESİ

Bu talimatname ile GitHub Copilot'a Django projenizi baştan sona oluşturması için gerekli tüm bilgileri sağlayabilirsiniz. Aşağıdaki alanları doldurun ve bu dosyayı Copilot'a vererek projenizin otomatik olarak oluşturulmasını isteyin.

# 📋 PROJE TEMEL BİLGİLERİ

## 🎯 Proje Tanımı

|  |  |
| --- | --- |
| Proje Adı: | [Projenizin adını yazın - örn: eticaret\_sitesi] |
| Proje Açıklaması: | [Projenin ne yaptığını kısaca açıklayın - örn: Online kitap satış sitesi] |
| Hedef Kitle: | [Kimler kullanacak - örn: Kitap severler, öğrenciler] |
| Ana Özellikler: | [Temel özellikleri listeleyin - örn: Ürün katalogu, sepet, ödeme, kullanıcı kayıt] |
| Proje Türü: | [E-ticaret, Blog, CRM, Portfolio, vb.] |

## 🏗️ Teknik Gereksinimler

|  |  |
| --- | --- |
| Django Versiyonu: | [Örn: 5.0.x, 4.2.x LTS] |
| Python Versiyonu: | [Örn: 3.11, 3.12] |
| Veritabanı: | [PostgreSQL / MySQL / SQLite] |
| Cache: | [Redis / Memcached / Yok] |
| Frontend Framework: | [Bootstrap / Tailwind / Custom CSS / React] |
| API: | [Django REST Framework gerekli mi? Evet/Hayır] |

# 🎨 KULLANICI ARAYÜZÜ VE TASARIM

## 📱 Sayfalar ve İşlevsellik

|  |  |
| --- | --- |
| Ana Sayfalar: | [Hangi sayfalar olacak - örn: Ana sayfa, ürün listesi, ürün detay, sepet, ödeme, profil] |
| Kullanıcı Türleri: | ☐ Misafir kullanıcı\n☐ Kayıtlı kullanıcı\n☐ Admin/Yönetici\n☐ Satıcı |
| Özel İşlevler: | [Ekstra özellikler - örn: Wishlist, yorumlar, puanlama, kupon sistemi, kargo takibi] |
| Navigation Yapısı: | [Menü yapısı, breadcrumb, arama özellikleri] |

## 🎨 Tasarım Tercihleri

|  |  |
| --- | --- |
| Renk Teması: | [Örn: Mavi-beyaz, koyu tema, minimalist] |
| Stil: | [Modern, klasik, minimalist, renkli] |
| Logo/Marka: | [Var/Yok - varsa açıklama] |
| Layout: | [Fixed width, fluid, responsive, mobile-first] |

# 🗄️ VERİTABANI YAPISI VE MODELLER

## 📊 Ana Veri Modelleri

|  |  |
| --- | --- |
| Gerekli Modeller: | [Hangi ana modeller olacak - örn: User, Product, Category, Order, Cart] |
| Model İlişkileri: | [ForeignKey, ManyToMany, OneToOne ilişkileri açıklayın] |
| Özel Alanlar: | [Özel field'lar, validation kuralları, meta bilgiler] |

Örnek Model Yapısı:

Product Model:  
- name: CharField(max\_length=200)  
- description: TextField  
- price: DecimalField(max\_digits=10, decimal\_places=2)  
- stock\_quantity: IntegerField  
- category: ForeignKey(Category)  
- images: ImageField (çoklu resim gerekli mi?)  
- tags: ManyToManyField (etiket sistemi)  
- created\_at: DateTimeField(auto\_now\_add=True)  
- updated\_at: DateTimeField(auto\_now=True)  
- is\_active: BooleanField(default=True)

# 🔐 GÜVENLİK VE KULLANICI YÖNETİMİ

## 👤 Kimlik Doğrulama

|  |  |
| --- | --- |
| Kayıt Sistemi: | [Email/Kullanıcı adı ile kayıt] |
| Giriş Seçenekleri: | [Email/Username + şifre, sosyal medya girişi] |
| Sosyal Giriş: | [Google, Facebook, GitHub - hangilerini istiyorsanız] |
| Email Doğrulama: | [Gerekli/Gerekli değil] |
| Şifre Sıfırlama: | [Email ile/SMS ile] |

## 🛡️ Güvenlik Özellikleri

* ☐ reCAPTCHA entegrasyonu
* ☐ Rate Limiting (DDoS koruması)
* ☐ CSRF Protection
* ☐ XSS Protection
* ☐ SQL Injection koruması
* ☐ Admin panel güvenliği
* ☐ HTTPS zorunluluğu
* ☐ Password strength validation
* ☐ Two-factor authentication
* ☐ Session security

# 💳 ÖDEME VE E-TİCARET

|  |  |
| --- | --- |
| Ödeme Yöntemleri: | [Kredi kartı, PayPal, Stripe, iyzico, vb.] |
| Para Birimi: | [TL, USD, EUR] |
| Kargo Sistemi: | [Kargo entegrasyonu gerekli mi?] |
| Stok Takibi: | [Otomatik/Manuel stok yönetimi] |
| İndirim Sistemi: | [Kupon, promosyon kodu sistemi] |

# 🔧 ENTEGRASYON VE API

|  |  |
| --- | --- |
| REST API: | [Django REST Framework gerekli mi?] |
| API Dokümantasyonu: | [Swagger/OpenAPI otomatik oluşturulsun mu?] |
| Dış Servisler: | [Google Maps, AWS S3, SendGrid, vb.] |
| Social Media: | [Facebook, Twitter, Instagram entegrasyonu] |
| Analytics: | [Google Analytics, özel tracking] |
| Email Servisi: | [Gmail SMTP, SendGrid, AWS SES] |

# 🚀 DEPLOYMENT VE HOSTING

|  |  |
| --- | --- |
| Hosting Platform: | [Heroku, DigitalOcean, AWS, VPS] |
| Domain: | [Özel domain var mı?] |
| SSL Sertifikası: | [Let's Encrypt, paid certificate] |
| CDN: | [CloudFlare, AWS CloudFront] |
| CI/CD: | [GitHub Actions, GitLab CI] |
| Environment: | [Development, Staging, Production] |

# 🎯 ÖZEL GEREKSİNİMLER

Projenize özel, yukarıda yer almayan gereksinimleri buraya yazın:

* ☐ QR kod ile ürün tarama
* ☐ AI destekli ürün önerisi
* ☐ Canlı chat sistemi
* ☐ Video streaming
* ☐ Dosya upload/download sistemi
* ☐ Multi-vendor marketplace
* ☐ Subscription sistemi
* ☐ Booking/Rezervasyon sistemi
* ☐ Çok dilli destek
* ☐ PWA (Progressive Web App)
* ☐ Push notifications
* ☐ Offline çalışma

Diğer özel gereksinimler:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# 📋 PROJE TESLİM FORMATI

* ☐ Tüm kaynak kodlar (GitHub repository)
* ☐ Requirements.txt dosyası
* ☐ README.md dosyası (kurulum kılavuzu)
* ☐ Environment variables şablonu (.env.example)
* ☐ Database schema ve initial data
* ☐ Admin kullanıcısı (username/password)
* ☐ Test verileri ve test kullanıcıları
* ☐ Deployment kılavuzu
* ☐ API dokümantasyonu (varsa)
* ☐ Kullanıcı kılavuzu

# 🤖 GITHUB COPILOT TALİMATI

Yukarıdaki formu doldurduktan sonra, aşağıdaki talimatı GitHub Copilot'a verin:

"Bu Django proje talimatnamesindeki tüm gereksinimlere göre eksiksiz bir Django projesi oluştur.  
  
PROJE YAPISI:  
1. Django projesini başlat ve gerekli ayarları yap  
2. Belirtilen modelları oluştur ve migration'ları hazırla  
3. Admin panel konfigürasyonunu yap  
4. Views, URLs ve templates'leri oluştur  
5. Forms ve validation'ları ekle  
6. Authentication ve authorization sistemini kur  
7. Gerekli static files ve CSS'leri hazırla  
8. Test dosyalarını oluştur  
9. Requirements.txt ve deployment dosyalarını hazırla  
10. README.md ve dokümantasyonu oluştur  
  
GEREKSINIMLER:  
- Her adımı detaylı olarak açıkla  
- Working code'lar üret  
- Production-ready olsun  
- Secure olsun  
- Scalable olsun  
- Best practices uygula  
- Comprehensive error handling  
- Proper logging  
  
DOSYA YAPISI:  
- Clean architecture  
- Proper separation of concerns  
- Modular design  
- Easy to maintain  
- Well documented"

# 📝 KULLANIM TALİMATLARI

## 🎯 Nasıl Kullanılır

1. 1. Formu Doldurun: Yukarıdaki tüm alanları projenize göre doldurun
2. 2. Copilot'a Verin: Doldurduğunuz formu GitHub Copilot'a yapıştırın
3. 3. Talimat Ekleyin: "COPILOT TALİMATI" bölümünü de ekleyin
4. 4. İsteyin: "Bu talimatlara göre Django projemi oluştur" deyin
5. 5. Test Edin: Oluşturulan projeyi test edin ve eksikleri belirtin

## 💡 İpuçları

* • Belirsiz alanları boş bırakabilirsiniz, Copilot uygun varsayılanları seçer
* • Karmaşık projeler için adım adım ilerleyin
* • Her aşamada test edin ve eksikleri tamamlayın
* • Dokümantasyonu güncel tutun
* • Güvenlik ayarlarını production'da mutlaka kontrol edin
* • Performance testing yapın
* • Code review sürecini atlamamayın

## ⚠️ Önemli Notlar

* • Bu talimatname kapsamlı projeler için tasarlanmıştır
* • Küçük projeler için gerekli bölümleri kullanabilirsiniz
* • Güvenlik ayarlarını production'da mutlaka yapın
* • Backup stratejinizi unutmayın
* • Performance metrikleri izleyin
* • Kullanıcı geri bildirimlerini toplayın

Son Güncelleme: 27 Ağustos 2025 | Versiyon: 2.0 | GitHub Copilot Assistant