



Uçak Üretim Yönetim Sistemi Dokümanı

Proje Sahibi: Hakan SANDIKÇI

Proje Tanımı

Bu proje, uçak parçalarını yönetmek, üretim süreçlerini kontrol etmek ve ekip iş birliğini kolaylaştırmak için geliştirilmiş bir uçak üretim yönetim sistemidir. Django framework'ü kullanılarak geliştirilmiştir ve önyüzlerde Tailwind CSS kullanılmıştır. Proje Docker ile ayağa kalkmış olup veritabanında postgresql ile bağlantı sağlanmıştır.

Özellikler

- Kullanıcı Kayıt ve Giriş Sistemi
- Ekip ve Parça Yönetimi
- Parça Üretimi ve Stok Takibi
- Geri Dönüşüm Yönetimi
- Montaj Süreci ve Eksik Parça Kontrolü
- Üretilen Uçakların Listesi

Kullanılan Teknolojiler

- **Backend:** Django Framework
- **Frontend:** Tailwind CSS
- **Veritabanı:** PostgreSQL
- **Testler:** Unit testler projeye dahil edilmiştir.
- **Docker:** PostgreSQL ve uygulama için Docker konteynerleri kullanılmıştır.

Model Yapısı

Part Modeli

- **name:** Parça türü (Kanat, Gövde, Kuyruk, Aviyonik)
- **stock:** Stok miktarı

Aircraft Modeli

- **name:** Uçak adı

Team Modeli

- **name:** Ekip türü (Kanat Takımı, Gövde Takımı, Kuyruk Takımı, Aviyonik Takımı, Montaj Takımı)

Employee Modeli

- **user_id:** Kullanıcı ile birebir ilişki.
- **team_id:** Çalışanın takımı (Team modeli ile ilişki).

ProducedAircraft Modeli

- **name:** Üretilen uçağın adı.
- **produced_at:** Üretim tarihi.
- **quantity:** Üretilen uçak adedi.

AircraftPart Modeli

- **aircraft_id:** Uçağın bilgisi (Aircraft modeli ile ilişki).
- **part_id:** Parçanın bilgisi (Part modeli ile ilişki).
- **quantity:** Parçanın miktarı.
- **deleted_at:** Geri dönüşüm bilgisi.

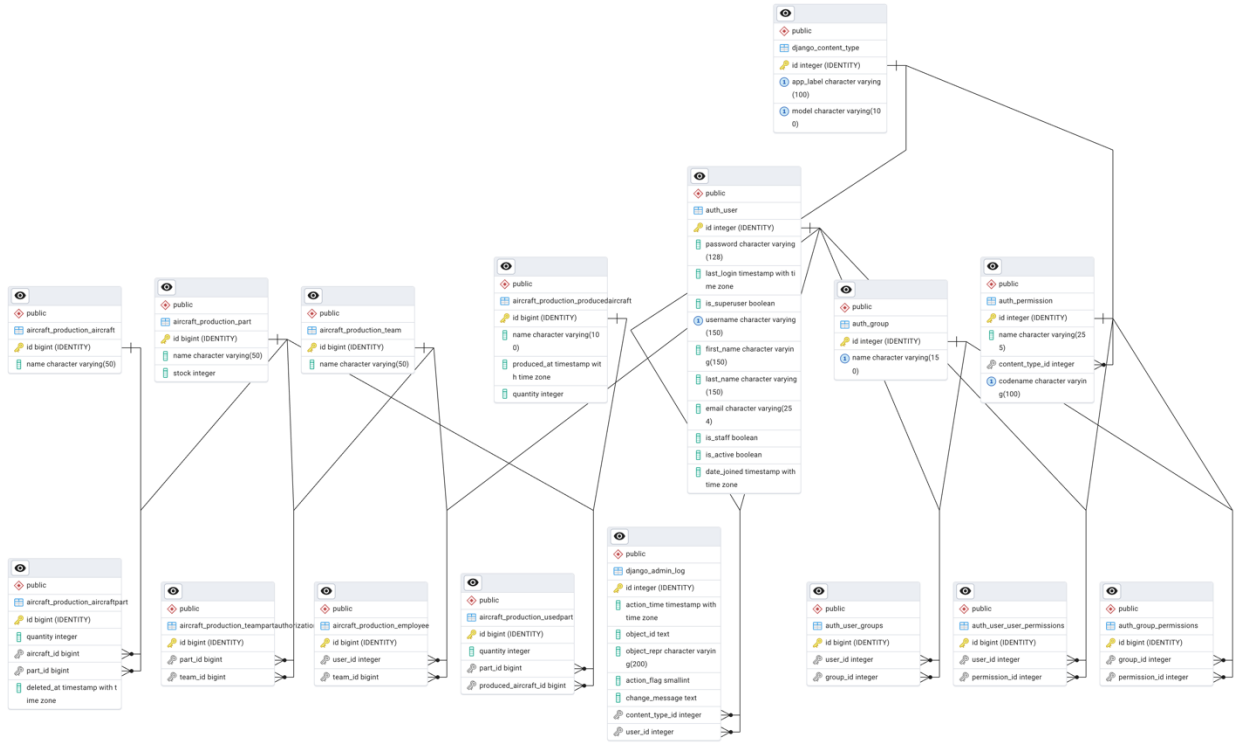
TeamPartAuthorization Modeli

- **team_id:** Takım bilgisi (Team modeli ile ilişki).
- **part_id:** Yetkilendirilen parça bilgisi (Part modeli ile ilişki).

UsedPart Modeli

- **produced_aircraft_id:** Kullanıldığı uçak bilgisi (ProducedAircraft modeli ile ilişki).
- **Part_id:** Kullanılan parça bilgisi (Part modeli ile ilişki).
- **quantity:** Kullanılan miktar.

Veritabanı Diyagramı



Görünümler (Views)

Kullanıcı Yönetimi

- **Kayıt:** Kullanıcıların sisteme kaydolmasını sağlar.
- **Giriş:** Kullanıcıların sisteme giriş yapmasını sağlar.
- **Çıkış:** Kullanıcıların sistemden çıkış yapmasını sağlar.

Dashboard

Kullanıcının bağlı olduğu takıma göre üretilecek parçaların bilgilerini gösterir.

Parça Yönetimi

- **Parça Üretimi:** Kullanıcının takımına uygun parça üretimi yapılır ve stok güncellenir.
- **Parça Azaltma:** Parça stoğundan miktar azaltılır veya geri dönüşüme atılır.
- **Parça Silme:** Parçayı silme işlemi.

Montaj

- **Montaj Sayfası:** Tüm uçakların listelendiği montaj sayfası.
- **Uçak Üretimi:** Gerekli parçalar kontrol edilerek uçak üretimi gerçekleştirilir.

Üretilen Uçaklar

Üretilen uçakların listesi.

Testler

Proje içerisinde aşağıdaki senaryolar için unit testler yazılmıştır:

- Parça üretimi ve stok güncellemeleri
- Parça geri dönüşüm işlemleri
- Üretilen uçakların listelenmesi

Kurulum

Gereksinimler

- Python 3.11+
- Docker
- PostgreSQL

Adımlar

1. Depoyu klonlayın:

```
git clone https://github.com/hakansndkc5/aircraft_production_app.git
cd aircraft_production_app
```

2. Docker konteynerlerini başlatın:

```
docker-compose up --build
```

3. Django uygulamasını yeniden başlatın: Docker konteynerleri build edildikten sonra django_app konteynerini bir kez yeniden başlatmanız gerekmektedir.

4. Projeye erişim: Tarayıcınızda <http://127.0.0.1:8000> adresini açın.

Önyüz Tasarımı

Projenin önyüzü, kullanıcı dostu ve modern bir görünüm sağlamak için Tailwind CSS ile tasarlanmıştır. Arka plan renkleri canlı tutulup butonlarda ufak animasyonlar kullanılarak arayüzün verimi artırmaya odaklı olmasına özen gösterilmiştir.