🚀 CODRescue - Système de Gestion E-commerce

Système de gestion complet pour e-commerce avec préparation de commandes, logistique et supervision

🎯 Aperçu du projet

CODRescue est une solution complète de gestion e-commerce développée avec Django, conçue pour optimiser les processus de préparation de commandes, de logistique et de supervision. Le système offre une interface moderne et intuitive pour gérer efficacement toutes les étapes du cycle de vie des commandes.

🎨 Interface utilisateur moderne

- Design responsive avec Tailwind CSS
- Interface intuitive et accessible
- Recherche globale intelligente
- Tableaux de bord en temps réel

✨ Fonctionnalités principales

🛒 **Gestion des Commandes**

- Création et suivi des commandes
- États de commande avancés
- Gestion des clients et adresses
- Système de validation automatique

📦 **Préparation de Commandes**

- Interface dédiée aux opérateurs de préparation
- Suivi en temps réel des commandes
- Gestion des articles et variantes
- Système de validation des préparations

🚚 Logistique et Livraison

- Gestion des livraisons
- Suivi des retours
- Service après-vente
- Optimisation des routes

👥 Supervision

- Tableau de bord 360° pour les superviseurs
- KPIs et métriques en temps réel
- Gestion des opérateurs
- Rapports détaillés

🔍 Recherche Intelligente

- Recherche globale multi-critères
- Suggestions automatiques
- Filtres avancés
- Recherche en temps réel

🤖 Chatbot IA

- Assistant SQL intelligent pour l'analyse de données en langage naturel
- Intégration avec Google Gemini Pro
- Contexte adaptatif en fonction de l'interface et de l'utilisateur
- Requêtes SQL sécurisées en lecture seule

🏗️ Architecture

```
CODRescue/
├── 📁 article/              # Gestion des articles et variantes
├── 📁 chatbot/              # Intégration de l'assistant IA
├── 📁 client/               # Gestion des clients
├── 📁 commande/             # Module principal des commandes
├── 📁 kpis/                 # Indicateurs de performance
├── 📁 livraison/            # Gestion des livraisons
├── 📁 operatConfirme/       # Interface opérateur confirmé
├── 📁 operatLogistic/       # Interface logistique
├── 📁 parametre/            # Paramètres et configuration
├── 📁 Prepacommande/        # Interface préparation
├── 📁 Superpreparation/     # Interface supervision
├── 📁 synchronisation/      # Synchronisation externe
├── 📁 templates/            # Templates HTML
├── 📁 static/               # Fichiers statiques
└── 📁 config/               # Configuration Django
```

🚀 Installation

Prérequis

- Python 3.13+
- PostgreSQL (recommandé) ou SQLite
- Node.js (pour Tailwind CSS)
- Git

1. Cloner le projet

```
git clone https://github.com/votre-username/CODRescue.git
cd CODRescue
```

2. Créer un environnement virtuel

```
python -m venv env
# Windows
env\Scripts\activate
# Linux/Mac
source env/bin/activate
```

3. Installer les dépendances

```
pip install -r requirements.txt
```

4. Configuration de la base de données

```
python manage.py makemigrations
python manage.py migrate
```

5. Créer un superutilisateur

```
python manage.py createsuperuser
```

6. Installer les dépendances de Tailwind CSS

```
npm install
```

7. Lancer le build de Tailwind CSS

```
npm run build
```

8. Lancer le serveur

```
python manage.py runserver
```

⚙ Configuration

Variables d'environnement

Créez un fichier `.env` à la racine du projet :

```
SECRET_KEY=votre_clé_secrète
DEBUG=True
DATABASE_URL=postgresql://user:password@localhost:5432/codrescue
REDIS_URL=redis://localhost:6379/0
```

Configuration de la base de données

Modifiez `config/settings.py` selon votre configuration :

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'codrescue',
        'USER': 'votre_utilisateur',
        'PASSWORD': 'votre_mot_de_passe',
        'HOST': 'localhost',
        'PORT': '5432',
    }
}
```

🖩 Modules disponibles

Applications installées

Le projet est organisé en plusieurs applications Django :

- `django.contrib.admin`
- `django.contrib.auth`
- `django.contrib.contenttypes`
- `django.contrib.sessions`
- `django.contrib.messages`
- `django.contrib.staticfiles`
- `django.contrib.humanize`
- `rest_framework`
- `django_filters`
- `crispy_forms`
- `widget_tweaks`
- `corsheaders`
- `django_extensions`
- `tailwind`
- `theme`
- `django_browser_reload`
- `commande`
- `article`
- `client`
- `livraison`
- `parametre`
- `operatConfirme`
- `operatLogistic`
- `synchronisation`
- `Prepacommande`
- `Superpreparation`
- `kpis`

🔧 Utilisation

Accès aux interfaces

- **Administration** : `/admin/`
- **Préparation** : `/operateur-preparation/`
- **Logistique** : `/operateur-logistique/`
- **Supervision** : `/super-preparation/`
- **Paramètres** : `/parametre/`

Commandes utiles

```
# Créer des données de test
python manage.py loaddata fixtures/initial_data.json

# Synchroniser avec Google Sheets
python manage.py sync_google_sheets
```

```
# Générer des rapports
python manage.py generate_reports

# Nettoyer les données
python manage.py cleanup_old_data
```

📊 Technologies utilisées

Backend

- **Django 5.1.7** - Framework web principal
- **PostgreSQL** - Base de données relationnelle
- **Redis** - Cache et sessions
- **Celery** - Tâches asynchrones
- **Django REST Framework** - API REST

Frontend

- **Tailwind CSS** - Framework CSS
- **JavaScript ES6+** - Interactivité
- **Chart.js** - Graphiques et visualisations
- **Font Awesome** - Icônes

Outils de développement

- **Django Debug Toolbar** - Débogage
- **Pytest** - Tests unitaires
- **Black** - Formatage de code
- **Flake8** - Linting

Déploiement

- **Gunicorn** - Serveur WSGI
- **Nginx** - Serveur web
- **Docker** - Conteneurisation
- **GitHub Actions** - CI/CD

🤝 Contribution

Nous accueillons les contributions ! Voici comment contribuer :

1. **Fork** le projet
2. Créer une branche pour votre fonctionnalité (`git checkout -b feature/AmazingFeature`)
3. **Commit** vos changements (`git commit -m 'Add some AmazingFeature'`)
4. **Push** vers la branche (`git push origin feature/AmazingFeature`)
5. Ouvrir une **Pull Request**

Guidelines de contribution

- Suivre les conventions de code Python (PEP 8)
- Ajouter des tests pour les nouvelles fonctionnalités
- Documenter les changements importants
- Respecter la structure du projet

Licence

Ce projet est sous licence MIT. Voir le fichier LICENSE pour plus de détails.

Support

- **Email** : codsuitebackend@gmail.com
- **Téléphone** : +212 779 635 687 / +212 694 528 498
- **Issues** : GitHub Issues

Si ce projet vous aide, n'hésitez pas à lui donner une étoile !

Stars repo not found

Forks repo not found