

MPJFUMS - Parfumerie Gabonaise en ligne

MPJFUMS est une application web de parfumerie gabonaise développée avec Django, offrant une expérience utilisateur élégante et performante.

🌸 Fonctionnalités

- Catalogue de parfums avec catégorisation
- Optimisation des images avec django-imagekit
- Interface administrateur personnalisée
- Multilingue (français par défaut)
- Responsive design pour mobile et desktop

🔧 Technologies utilisées

- **Backend:** Django 5.2
- **Base de données:** PostgreSQL (production), SQLite (développement)
- **Déploiement:** Docker, Railway
- **CI/CD:** GitHub Actions
- **Monitoring:** Sentry
- **Statiques:** WhiteNoise
- **Optimisations:**
 - Traitement d'images: django-imagekit
 - Mise en cache: django-cache-memoize

🚀 Installation et démarrage

Prérequis

- Python 3.10+
- Docker et Docker Compose (optionnel)

Installation locale

```
# Cloner le dépôt
git clone https://github.com/votre-utilisateur/mpjfums-.git
cd mpjfums-

# Installer les dépendances
pip install -r requirements.txt

# Configurer la base de données
cd shop
python manage.py migrate
```

```
# Démarrer le serveur  
python manage.py runserver
```

Avec Docker

```
# Construire et démarrer les conteneurs  
docker-compose -f shop/docker-compose.yml up
```

L'application sera disponible à l'adresse <http://localhost:8000>

Déploiement

L'application est configurée pour un déploiement facile sur Railway. Consultez le fichier [railwaydoc.md](#) pour les instructions détaillées.

Pipeline CI/CD

Un pipeline CI/CD est configuré avec GitHub Actions pour les tests et l'analyse de code, et Railway pour le déploiement automatique. Voir [CICD.md](#) pour plus de détails.

Tests

```
cd shop  
python manage.py test
```

Documentation

- [Guide de déploiement Railway](#)
- [Configuration CI/CD](#)

Contribution

Les contributions sont les bienvenues ! N'hésitez pas à soumettre des pull requests.

Licence

Ce projet est sous licence [MIT](#).