

Stack Technologique du Projet E-commerce de Mode – *Jaelle Shop*

Back-end

- **Django** – Framework Python robuste pour gérer la logique métier (produits, utilisateurs, commandes)
 - **Django REST Framework (DRF)** – Pour exposer une API RESTful destinée au front-end
 - **PostgreSQL** – Base de données relationnelle performante et sécurisée
 - **Docker** – Conteneurisation du projet pour une portabilité optimale
 - **Docker Compose** – Orchestration des services (Django, PostgreSQL, etc.)
 - **Gunicorn** – Serveur WSGI efficace pour déployer Django en production
 - **Nginx** – Reverse proxy pour la gestion du trafic et du HTTPS
 - **Celery + Redis** (*optionnel*) – Pour la gestion des tâches en arrière-plan (emails, notifications, etc.)
-

Front-end (séparé via API)

- **React.js** – Librairie JavaScript pour construire une interface utilisateur réactive
 - **Vite** – Outil de build ultra-rapide pour initialiser et développer l'app React
 - **Tailwind CSS** – Framework CSS utilitaire pour un design moderne et responsive
 - **Axios** – Pour interagir efficacement avec l'API Django
 - **React Router** – Pour gérer la navigation entre les pages de manière fluide
 - **React Hook Form** (*optionnel*) – Pour une gestion simple et puissante des formulaires
 - **Redux Toolkit** ou **Context API** – Pour la gestion globale de l'état (panier, utilisateur, etc.)
-

Média & Stockage

- **Cloudinary** – Solution cloud pour le stockage, la gestion et l'optimisation des images (produits, utilisateurs...)
-

Paiement (développement prévu plus tard)

- **Stripe** – Pour des paiements en ligne sécurisés et conformes
 - **Django Stripe / Webhooks** – Intégration côté serveur pour la validation des paiements
 - (*ou PayPal selon les préférences ou le marché cible*)
-

Déploiement & Hébergement

- **Railway** – Plateforme cloud simple et rapide pour déployer :
 - L'API Django + PostgreSQL
 - L'application React
 - La configuration des variables d'environnement

- **GitHub** – Pour le versionnage du code et la collaboration
 - **GitHub Actions** – Pour automatiser les workflows de CI/CD (déploiement, tests...)
-

Sécurité

- **HTTPS** (via Let's Encrypt avec Railway ou Nginx) – Pour chiffrer les échanges de données
- **Django Allauth** ou **Djoser + JWT** – Pour une gestion robuste de l'authentification