

Riziky-Boutic - Documentation Complète

Vue d'Ensemble du Projet

Riziky-Boutic est une plateforme e-commerce moderne et complète développée avec React/TypeScript pour le frontend et Node.js/Express pour le backend. Cette application offre une expérience d'achat en ligne professionnelle avec toutes les fonctionnalités essentielles d'un commerce électronique moderne.

Objectif Principal

Créer une plateforme e-commerce robuste, sécurisée et évolutive permettant aux entreprises de vendre leurs produits en ligne avec une expérience utilisateur exceptionnelle.

Public Cible

- **Clients finaux** : Particuliers et professionnels souhaitant acheter des produits en ligne
- **Administrateurs** : Gestionnaires de boutique pour la gestion des produits, commandes et clients
- **Développeurs** : Équipe technique pour la maintenance et l'évolution du système

Architecture Générale

Stack Technologique

Frontend (Client)

React 18.3.1	→ Framework UI moderne
TypeScript 5.0+	→ Sécurité des types
Tailwind CSS 3.3+	→ Framework CSS utilitaire
Shadcn/UI	→ Composants UI pré-construits
React Router 6.26+	→ Routing côté client
Axios	→ Client HTTP
Socket.io-client	→ Communication temps réel

Backend (Serveur)

Node.js 18+	→ Runtime JavaScript serveur
Express.js 4.18+	→ Framework web
Socket.io	→ WebSocket serveur
JWT	→ Authentification
Helmet.js	→ Sécurité headers
Multer	→ Upload fichiers

Base de Données

JSON Files → Stockage actuel (développement)
PostgreSQL/MongoDB → Migration prête (production)

Structure des Répertoires

```
riziky-boutic/  
  src/                                # Code source frontend React  
    components/                       # Composants réutilisables  
    pages/                           # Pages de l'application  
    hooks/                           # Hooks personnalisés  
    services/                         # Services API  
    contexts/                        # Contextes React  
    types/                           # Définitions TypeScript  
  server/                            # Code source backend Node.js  
    routes/                          # Routes API  
    services/                        # Logique métier  
    middlewares/                    # Middlewares Express  
    data/                           # Fichiers JSON (base de données)  
    uploads/                         # Fichiers uploadés  
  public/                            # Fichiers statiques  
  docs/                             # Documentation projet
```

Démarrage Rapide

Prérequis

- Node.js 18+ installé
- NPM ou Yarn
- Git
- Navigateur web moderne

Installation

1. Cloner le projet

```
git clone [URL_DU_DEPOT]  
cd riziky-boutic
```

2. Installer les dépendances frontend

```
npm install
```

3. Installer les dépendances backend

```
cd server  
npm install
```

```
cd ..
```

4. Démarrer en mode développement

Terminal 1 (Backend) :

```
cd server  
npm run dev
```

Terminal 2 (Frontend) :

```
npm run dev
```

5. Accéder à l'application

- Frontend : `http://localhost:8080`
- Backend API : `http://localhost:10000`

Comptes de Test

Administrateur : - Email : `admin@riziky-boutic.com` - Mot de passe : `Admin123!`

Client : - Email : `vianey.jean@gmail.com` - Mot de passe : [voir fichier `users.json`]

Utilisation de la Plateforme

Pour les Clients

1. Inscription et Connexion

- Créer un compte via la page d'inscription
- Se connecter avec email/mot de passe
- Récupération de mot de passe disponible

2. Navigation et Shopping

- Parcourir le catalogue par catégories
- Utiliser la recherche avancée
- Filtrer par prix, disponibilité, etc.
- Ajouter des produits au panier
- Gérer sa liste de favoris

3. Processus d'Achat

- Réviser le contenu du panier
- Saisir les informations de livraison
- Choisir le mode de paiement
- Confirmer la commande

- Recevoir la confirmation par email

4. Suivi des Commandes

- Consulter l'historique des commandes
- Suivre l'état de livraison
- Laisser des avis produits
- Demander un remboursement si nécessaire

Pour les Administrateurs

1. Accès à l'Administration

- Se connecter avec un compte administrateur
- Accéder au panel d'administration via `/admin`

2. Gestion des Produits

- Créer/modifier/supprimer des produits
- Gérer les catégories
- Upload d'images produits
- Gestion des stocks
- Configuration des promotions

3. Gestion des Commandes

- Traiter les nouvelles commandes
- Mettre à jour les statuts de livraison
- Gérer les remboursements
- Communiquer avec les clients

4. Gestion des Utilisateurs

- Consulter la liste des clients
- Gérer les comptes utilisateurs
- Modérer les avis et commentaires

Configuration et Personnalisation

Variables d'Environnement

Créer un fichier `.env` à la racine :

```
VITE_API_BASE_URL=http://localhost:10000
NODE_ENV=development
JWT_SECRET=votre_secret_jwt_securise
PORT=10000
```

Personnalisation du Design

Couleurs de la Marque Modifier `src/index.css` :

```
:root {
  --primary: 0 84% 60%;           /* Rouge principal */
  --primary-foreground: 0 0% 98%;
  --secondary: 0 0% 96%;
  --accent: 0 84% 70%;           /* Rouge accent */
}
```

Logo et Images

- Remplacer `public/images/Logo/Logo.png`
- Mettre à jour `public/favicon.ico`
- Ajouter des images dans `public/images/`

Configuration des Emails

Modifier `server/services/email.service.js` pour configurer l'envoi d'emails.

Fonctionnalités Principales

Côté Client (Frontend)

Authentification

```
// Hook d'authentification
const { user, login, logout, isAuthenticated } = useAuth();

// Utilisation
await login(email, password);
```

Gestion du Panier

```
// Hook du panier
const { cart, addToCart, removeFromCart, updateQuantity } = useCart();

// Ajouter un produit
await addToCart(productId, quantity);
```

Recherche de Produits

```
// Hook des produits
const { products, searchProducts, filterProducts } = useProducts();
```

```
// Rechercher  
const results = await searchProducts(query, filters);
```

Côté Serveur (Backend)

Routes Principales

```
// Authentification  
POST /api/auth/login  
POST /api/auth/register  
POST /api/auth/logout  
  
// Produits  
GET /api/products  
GET /api/products/:id  
POST /api/products (admin)  
PUT /api/products/:id (admin)  
  
// Commandes  
GET /api/orders  
POST /api/orders  
PUT /api/orders/:id/status (admin)  
  
// Panier  
GET /api/panier  
POST /api/panier/add  
DELETE /api/panier/remove
```

Services Métier

```
// Service des produits  
const productsService = require('./services/products.service');  
  
// Obtenir tous les produits  
const products = await productsService.getAllProducts();  
  
// Créer un produit  
const newProduct = await productsService.createProduct(productData);
```

Sécurité

Authentification JWT

- Tokens sécurisés avec expiration
- Refresh tokens automatiques
- Protection contre les attaques par force brute

Protection des Routes

```
// Route protégée
<ProtectedRoute>
  <AdminDashboard />
</ProtectedRoute>

// Route sécurisée avec IDs obfusqués
<SecureRoute>
  <ProductDetail />
</SecureRoute>
```

Validation des Données

- Validation côté client avec Zod
- Sanitisation côté serveur
- Protection XSS automatique

HTTPS et Sécurité Headers

```
// Headers de sécurité (Helmet.js)
app.use(helmet({
  contentSecurityPolicy: true,
  crossOriginEmbedderPolicy: true
}));
```

Maintenance et Débogage

Logs et Monitoring

Logs Frontend

```
// Console logs structurés
console.log('Action:', action, { data, timestamp: new Date() });
```

Logs Backend

```
// Logs serveur
const logger = require('./core/logger');
logger.info('User authenticated', { userId, timestamp });
logger.error('Database error', { error: err.message });
```

Tests

Tests Frontend

```
# Tests unitaires
```

```
npm run test
```

```
# Tests avec couverture
```

```
npm run test:coverage
```

Tests Backend

```
cd server
```

```
npm run test
```

Base de Données

Sauvegarde des Données JSON

```
# Copier les fichiers de données
```

```
cp server/data/*.json backup/
```

Migration vers PostgreSQL

1. Installer PostgreSQL
 2. Créer la base de données
 3. Exécuter les scripts de migration
 4. Mettre à jour la configuration
-

Déploiement Production

Préparation

1. Build Production

```
npm run build
```

2. Variables d'Environnement Production

```
NODE_ENV=production
```

```
VITE_API_BASE_URL=https://api.votre-domaine.com
```

```
JWT_SECRET=secret_production_securise
```

```
DATABASE_URL=postgresql://user:pass@host:port/db
```

3. Optimisations

- Compression Gzip activée
- Cache headers configurés
- CDN pour les assets statiques

Hébergement Recommandé

Frontend

- Vercel, Netlify, ou AWS S3 + CloudFront

Backend

- Heroku, DigitalOcean, ou AWS EC2

Base de Données

- PostgreSQL (Heroku Postgres, AWS RDS)
 - MongoDB Atlas pour NoSQL
-

Ce qu'il NE FAUT PAS Faire

Sécurité

- Exposer les clés API dans le code client
- Stocker des mots de passe en clair
- Ignorer la validation des entrées utilisateur
- Utiliser HTTP en production

Performance

- Charger tous les produits d'un coup
- Faire des requêtes API dans des boucles
- Oublier la mise en cache
- Négliger l'optimisation des images

Maintenance

- Modifier directement les fichiers JSON en production
 - Déployer sans tests
 - Supprimer les logs d'audit
 - Ignorer les sauvegardes
-

Bonnes Pratiques

Développement

- Utiliser TypeScript pour la sécurité des types
- Suivre les conventions de nommage
- Écrire des tests pour les fonctionnalités critiques
- Documenter les modifications importantes

Performance

- Implémenter la pagination pour les listes
- Utiliser le lazy loading pour les images
- Mettre en cache les données fréquemment utilisées
- Optimiser les requêtes base de données

UX/UI

- Maintenir une cohérence visuelle
 - Implémenter des états de chargement
 - Fournir des messages d'erreur clairs
 - Assurer la responsivité mobile
-

Support et Contact

Documentation Technique

- Architecture : docs/ARCHITECTURE_TECHNIQUE.md
- API : docs/API_DOCUMENTATION.md
- Composants : docs/COMPOSANTS_GUIDE.md

Résolution de Problèmes

1. Vérifier les logs console (F12)
2. Consulter les logs serveur
3. Vérifier la connectivité API
4. Valider les données d'entrée

Contact Équipe

- Email technique : dev@riziky-boutic.com
 - Documentation : docs@riziky-boutic.com
 - Support : support@riziky-boutic.com
-

Cette documentation constitue le guide principal pour comprendre, utiliser et maintenir la plateforme Riziky-Boutic. Elle doit être mise à jour à chaque modification importante du système.