

# Projet E-commerce

Vente de téléphones portables

**Réalisé par :**

Amir ZOUARI

**Technologies :**

Node.js – Express – MongoDB – React – Redux

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Présentation générale du projet</b>	<b>3</b>
2.1	Fonctionnalités principales . . . . .	3
2.2	Architecture technique . . . . .	3
<b>3</b>	<b>Modélisation des données</b>	<b>4</b>
3.1	User . . . . .	4
3.2	Profile . . . . .	4
3.3	Product . . . . .	4
3.4	Category . . . . .	5
3.5	Panier . . . . .	5
<b>4</b>	<b>Relations entre les entités</b>	<b>6</b>
4.1	User – Profile (1 :1) . . . . .	6
4.2	User – Panier (1 :1) . . . . .	8
4.3	User – Product (1 :N) . . . . .	10
4.4	Category – Product (1 :N) . . . . .	12
4.5	Panier – Product (N :N) . . . . .	14
<b>5</b>	<b>Conclusion</b>	<b>16</b>

# Chapitre 1 Introduction

Ce projet consiste à développer une application e-commerce spécialisée dans la vente de téléphones portables. Il s'agit d'une application web **full-stack** composée d'un backend basé sur Node.js et Express, et d'un frontend développé avec React et Redux.

L'objectif principal est de mettre en œuvre une architecture claire, une gestion des utilisateurs sécurisée et des relations cohérentes entre les différentes entités métier.

# Chapitre 2 Présentation générale du projet

## 2.1 Fonctionnalités principales

- Authentification des utilisateurs (inscription, connexion, JWT)
- Gestion des produits (CRUD pour les administrateurs)
- Gestion des catégories de téléphones
- Panier utilisateur
- Profil utilisateur

## 2.2 Architecture technique

- **Backend** : Node.js, Express, MongoDB (Mongoose)
- **Frontend** : React, Redux, Bootstrap
- **Sécurité** : JSON Web Tokens (JWT)

# Chapitre 3 Modélisation des données

Le projet repose sur **cinq entités principales** :

- User
- Profile
- Product
- Category
- Panier

## 3.1 User

L'entité **User** représente les utilisateurs de la plateforme.

- name
- email
- password
- phone
- isAdmin

## 3.2 Profile

L'entité **Profile** permet de stocker des informations supplémentaires liées à un utilisateur.

- fullName
- address
- phone
- preferences

## 3.3 Product

L'entité **Product** représente les téléphones portables vendus sur la plateforme.

- name
- price
- description

- category
- addedBy

## 3.4 Category

L'entité **Category** permet de classer les téléphones (Samsung, Apple, Xiaomi, etc.).

- name
- description

## 3.5 Panier

L'entité **Panier** représente le panier actif d'un utilisateur.

- items (productId, quantity)
- addedBy

# Chapitre 4 Relations entre les entités

## 4.1 User – Profile (1 :1)

Chaque utilisateur possède un seul profil. La relation est matérialisée par une référence vers l'identifiant de l'utilisateur.

**Type :** One-to-One (logique)

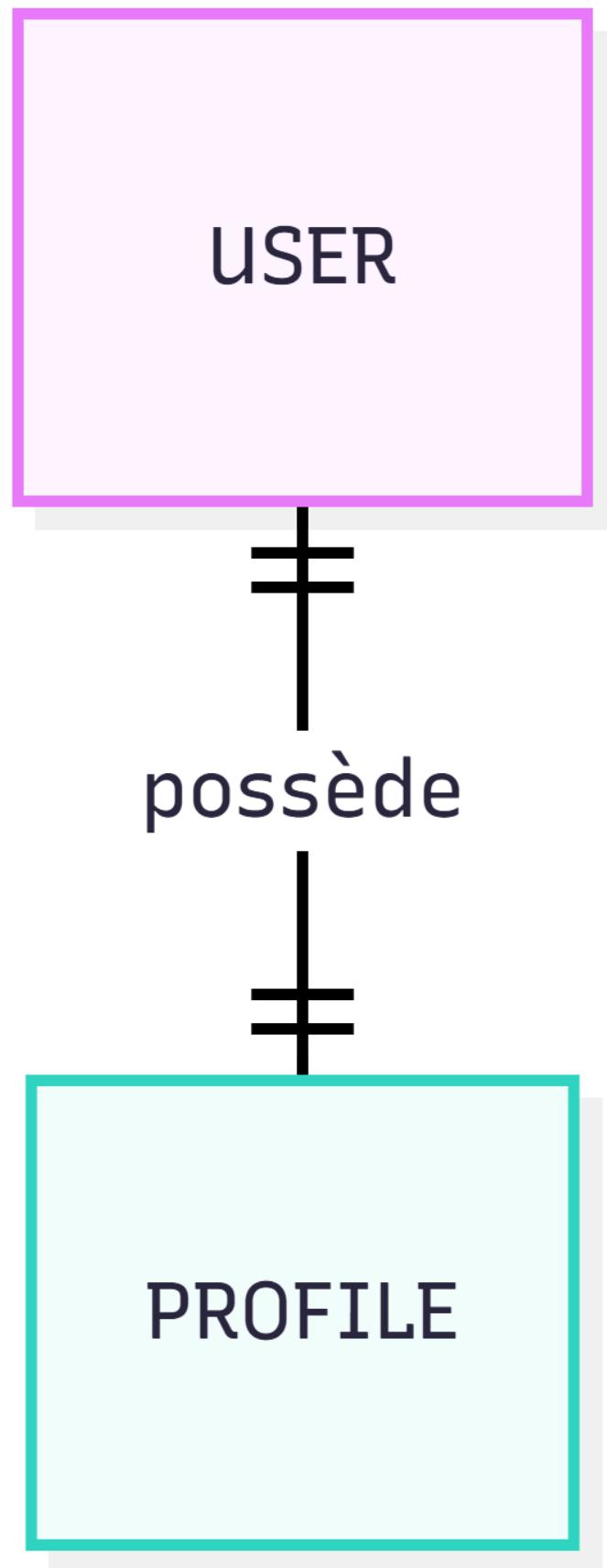


FIGURE 4.1 – Relation User – Profile (1 :1)

## 4.2 User – Panier (1 :1)

Chaque utilisateur possède un panier actif unique. Le panier est identifié par le champ `addedBy`.

**Type :** One-to-One (logique)

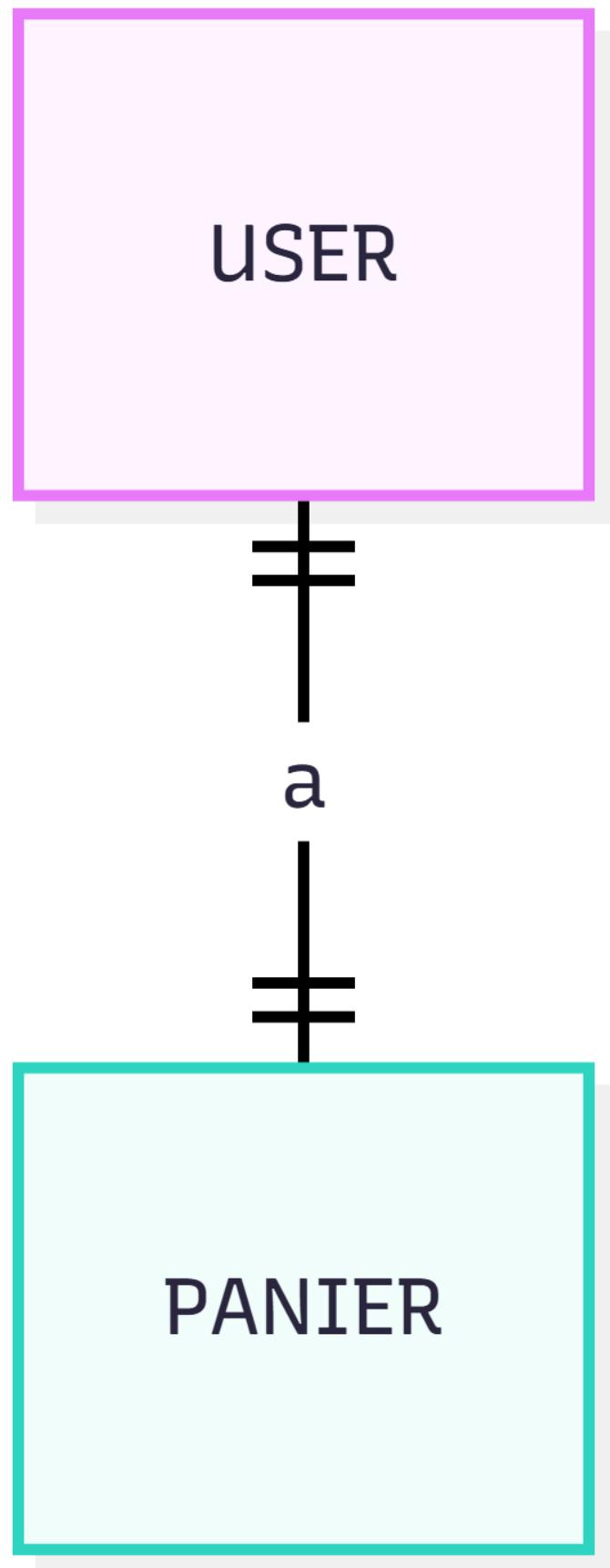


FIGURE 4.2 – Relation User – Panier (1 :1)

### 4.3 User – Product (1 :N)

Un utilisateur administrateur peut ajouter plusieurs produits.

**Type :** One-to-Many

USER



ajoute

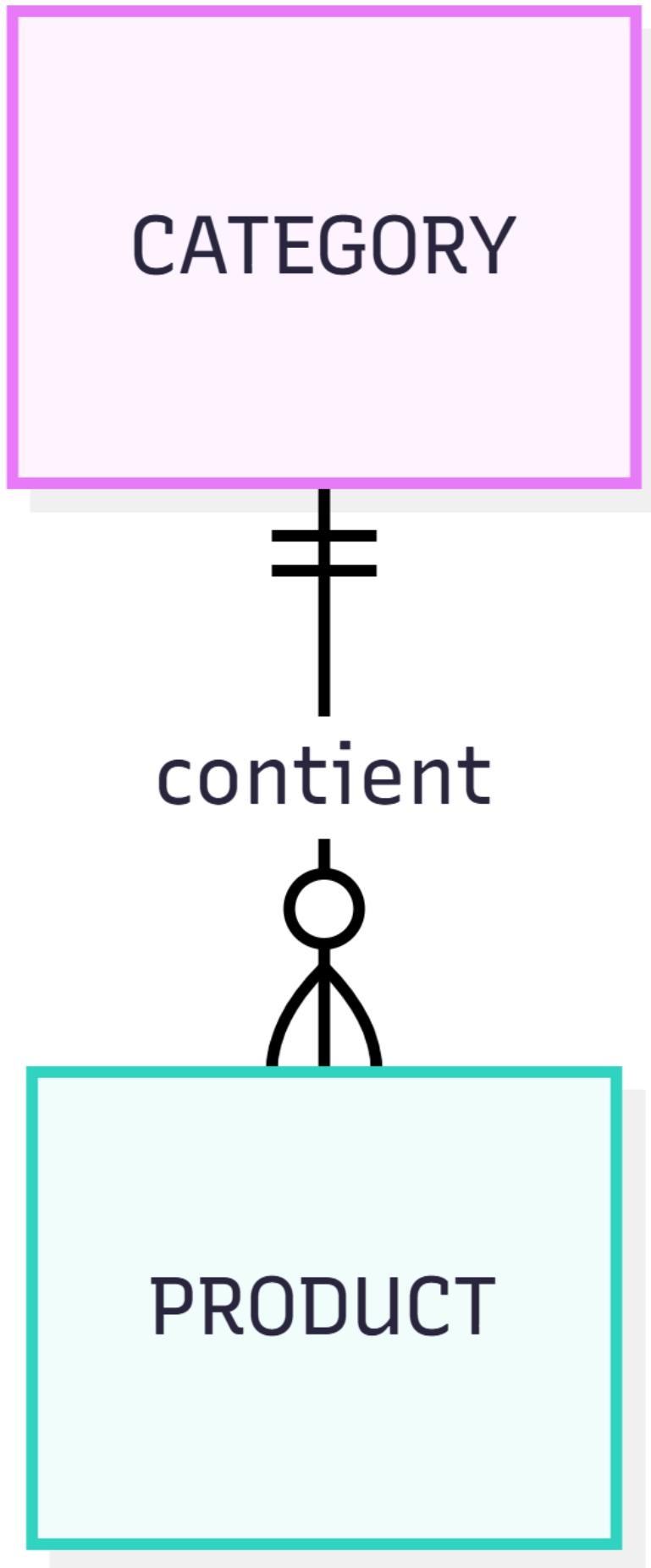


PRODUCT

## 4.4 Category – Product (1 :N)

Une catégorie regroupe plusieurs produits.

**Type :** One-to-Many



## 4.5 Panier – Product (N :N)

Un panier peut contenir plusieurs produits, et un produit peut apparaître dans plusieurs paniers.

**Type :** Many-to-Many (via items)

PANIER



contient



PRODUCT

## Chapitre 5 Conclusion

Ce projet e-commerce met en œuvre une architecture moderne et claire, avec une séparation nette entre les responsabilités. La modélisation des données et les relations entre entités assurent la cohérence et l'évolutivité de l'application.

Ce travail constitue une base solide pour des améliorations futures telles que la gestion des commandes, des paiements et des avis clients.