

Rapport de Projet :

Moteur de Recherche de Produits E-commerce

Réaliser par : Chaabani Najeh

Encadré par : Mr Bassem Boughzallah

Année universitaire **2024-2025**

Table des Matières

- 1. Introduction
- 2. Description Technique
 - 2.1 Backend
 - 2.2 Frontend
- 3. Mise en Œuvre
 - 3.1 Installation et Lancement
 - 3.2 Tests des Endpoints
- 4. Difficultés Rencontrées
 - 4.1 CORS
 - 4.2 Validation des Données
- 5. Conclusion
- 6. Améliorations Futures
- 7. Annexes
 - 7.1 Code Source
 - 7.2 Fichiers de Projet

1. Introduction

Ce projet consiste en la conception et la mise en œuvre d'une application de recherche de produits dans un contexte de commerce électronique. L'objectif principal est de fournir un système qui permet aux utilisateurs de :

Ajouter de nouveaux produits à une base de données.

Rechercher des produits en fonction de critères tels que le nom, la catégorie et la fourchette de prix.

La solution est divisée en deux parties :

Backend : Fournit une API REST créée avec Node.js, Express, et MongoDB.

Frontend : Une interface utilisateur simple développée en HTML, CSS, et JavaScript permettant de consommer l'API.

2. Description Technique

2.1 Backend

Le backend est un serveur Node.js basé sur Express qui gère deux opérations principales :

Ajout de produits : Endpoint : `/add-product`

Méthode : POST

Données entrées (JSON) :

```
{  
  "name": "Nom du produit",  
  "category": "Catégorie",  
  "price": 100,  
  "description": "Description du produit"  
}
```

Recherche de produits : Endpoint : `/search-products`

Méthode : GET

Paramètres de requête :

`name` : Filtrer par nom (optionnel)

`category` : Filtrer par catégorie (optionnel)

`min_price` : Filtrer par prix minimum (optionnel)

`max_price` : Filtrer par prix maximum (optionnel)

Technologies Utilisées

- ❖ Node.js : Environnement d'exécution.
- ❖ Express.js : Framework pour la gestion des routes et des API.
- ❖ Mongoose : ODM pour l'interaction avec MongoDB.
- ❖ MongoDB : Base de données NoSQL pour stocker les informations des produits.
- ❖ CORS : Middleware pour gérer les problèmes de partage des ressources entre origines.

Fonctionnalités Clés

- ❖ Validation des entrées pour éviter des erreurs de saisie.
- ❖ Gestion des erreurs pour renvoyer des messages explicites à l'utilisateur.

2.2 Frontend

L'interface utilisateur a été conçue pour être simple et intuitive, permettant aux utilisateurs d'interagir avec l'API.

Fonctionnalités Principales

- ❖ Ajout de produits :

Formulaire pour entrer les détails du produit.

Bouton pour envoyer les données au backend via une requête POST.

- ❖ Recherche de produits :

Formulaire avec des champs pour le nom, la catégorie, le prix minimum et le prix maximum.

Affichage dynamique des résultats avec des cartes de produit (nom, catégorie, prix, description).

Technologies Utilisées

- ❖ HTML : Structure de la page.
- ❖ CSS : Styles pour une présentation propre.
- ❖ JavaScript : Gestion des interactions utilisateur et des appels API via `fetch`.

3. Mise en Œuvre

3.1 Installation et Lancement

Prérequis

Node.js (v20.15.0 ou supérieur)

MongoDB (en local ou dans le cloud)

Étapes

Installation des dépendances backend :

```
npm install express
npm install body-parser
npm install mongoose
npm install cors
```

Lancer le serveur backend :

```
node index.js
```

Ouvrir l'interface frontend : Double-cliquez sur `index.html` pour l'ouvrir dans le navigateur.

3.2 Tests des Endpoints

Vous pouvez tester l'API backend avec Postman ou un autre outil similaire :

Ajouter un produit :

URL : `http://localhost:3000/add-product`

Méthode : POST

Corps :

```
{
  "name": "Laptop",
  "category": "Electronics",
  "price": 1200,
  "description": "High performance laptop"
}
```

Rechercher des produits :

URL : `http://localhost:3000/search-products?name=Laptop&category=Electronics&min_price=1000&max_price=2000`

Méthode : GET

4. Difficultés Rencontrées

4.1 CORS

Un problème de partage de ressources entre origines a été rencontré lors des appels API. Cela a été résolu en utilisant le middleware `cors` dans le backend.

4.2 Validation des Données

Des erreurs de validation ont été identifiées lors de l'enregistrement des produits. Cela a été corrigé en ajoutant des validations au schéma Mongoose.

5. Conclusion

Le projet a permis de créer une solution fonctionnelle pour la gestion et la recherche de produits dans un contexte de commerce électronique. Les deux parties (backend et frontend) fonctionnent de manière transparente pour offrir une expérience utilisateur fluide.

Ce projet met en évidence les compétences en :

Développement backend (Node.js, Express, MongoDB).

Développement frontend (HTML, CSS, JavaScript).

Intégration API REST.

6. Améliorations Futures

Implémentation d'une authentification pour sécuriser l'ajout de produits.

Ajout de la pagination et du tri dans les résultats de recherche.

Conception d'une interface utilisateur plus avancée avec des frameworks comme React ou Vue.js.

Hébergement du backend et du frontend pour un accès en ligne.

7. Annexes

Code Source

Backend : Voir fichier `back-end.zip`.

Frontend : Voir fichier `front-end.zip`.

Bibliographie/Webographie

1. Node.js Documentation

<https://nodejs.org/en/docs/>

Documentation officielle de Node.js, utilisée pour la mise en place de l'environnement backend.

2. Express.js Documentation

<https://expressjs.com/>

Guide détaillant le framework Express.js pour la création d'applications backend avec Node.js.

3. MongoDB Documentation

<https://www.mongodb.com/docs/>

Documentation officielle de MongoDB, utilisée pour la gestion des bases de données NoSQL dans le projet.

4. Mongoose Documentation

<https://mongoosejs.com/docs/>

Documentation sur Mongoose, utilisé pour interagir avec MongoDB via un ODM (Object Data Modeling).

5. CORS Middleware - npm

<https://www.npmjs.com/package/cors>

Page du module CORS pour résoudre les problèmes de partage de ressources entre origines (Cross-Origin Resource Sharing).

6. MDN Web Docs - Fetch API

https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API

Documentation de la Fetch API pour effectuer des requêtes HTTP côté frontend en JavaScript.

7. HTML & CSS Documentation - MDN

<https://developer.mozilla.org/en-US/docs/Web/HTML>

<https://developer.mozilla.org/en-US/docs/Web/CSS>

Ressources pour la conception d'interfaces frontend avec HTML et CSS.

8. JavaScript Documentation - MDN

<https://developer.mozilla.org/en-US/docs/Web/JavaScript>

Documentation complète sur le langage JavaScript, utilisé pour gérer les interactions frontend et l'appel API.