

# Tharindu PERERA BUT3 – Groupe Geryon

## 1. Présentation du projet audité

### Objectif du projet

Le projet "Pigment" est un selecteur multimodale qui permet de gérer, de rechercher, et de personnaliser des palettes de couleurs. Il vise à offrir une solution intuitive et performante pour la création de palettes adaptées à des projets graphiques, photos ou vidéos.

### Technologies utilisées

- Frontend : React.js, TailwindCSS pour le design
- Backend : Node.js, Express.js
- Base de données : MongoDB

### Fonctionnalités principales

L'application propose plusieurs fonctionnalités en commençant par la recherche de couleurs en fonction de divers tags ou (ou mots clés ça dépend de comment vous l'appellez) comme "Chaleureux, créatif" ou autres. En plus de ça nous avons la possibilité de créer nos propres palettes de couleurs et y ajouter les tags que l'ont souhaite et ensuite les palettes seront automatiques ajoutés et on pourra les retrouver par la suite.

### Utilisateurs cibles

Je pense que l'application s'adresse principalement aux professionnels du design, et autres métiers du design comme les graphistes etc . Le contexte d'utilisation est principalement dans des environnements de création visuelle.

## 2. Analyse technique

### Qualité du code

Le code est structuré de manière claire et logique, celui lis est bien lisible et structuré ; parce que le projet est bien organisé avec des dossiers clairs pour le frontend et le backend rendant le tout plus clair. De plus, au vu de la structure même du projet, celui-ci propose une évolution facile donc facilement améliorable.

### Performance

L'application présente des performances optimisées grâce à l'utilisation de MongoDB pour accélérer les recherches ce qui est très pratique

## Sécurité

Sauf erreur de ma part l'application n'a pas l'air d'avoir de mesures de sécurité particulière

## **3. Évaluation fonctionnelle**

### Objectif initial

Le projet remplit son objectif en permettant la recherche, la gestion, et la création de palettes de couleurs donc l'application répond bien aux besoins des utilisateurs

### Facilité d'utilisation et ergonomie

L'interface est intuitive, cependant peut être que l'interface de création de palette peut être améliorée pour la rendre plus simple à comprendre.

### Bugs et dysfonctionnements

Aucun bug n'a été observé durant les tests. Et même dans l'installation absolument aucun soucis

## **4. Documentation et maintenabilité**

### Commentaires et documentation

Le code est suffisamment commenté pour en faciliter la compréhension et le README est complet et fournit les informations nécessaires à l'installation.

Cependant pour l'installation si on utilise docker il faudrait peut être précisé qu'il faut faire des commandes spécifiques comme pour mon cas

`docker run -d --name mongo -p 27017:27017 mongo` car en utilisant mon mongo de base cela ne fonctionnait pas

## **5. Recommandations et conclusions**

### Points forts

- Architecture solide : Une séparation claire entre le frontend et le backend.
- Optimisation : La gestion des requêtes avec des index et mongoDB rends le tout plus rapide que si ils avaient utilisés une autre BDD

### Points faibles

- Consommation mémoire ? : Bien que celle-ci fonctionne bien, si on crée beaucoup de palette peut être que cela va poser des soucis de mémoire

### Propositions d'amélioration

La seule amélioration que je pourrais voir concerne celle de l'optimisation du tri donc trouver des techniques de recherche plus efficaces pour si la base de données devient trop grande.

### **Conclusion**

Le projet "Pigment" est bien conçu, performant et répond aux objectifs avec une interface utilisateur fluide et simple d'utilisation.