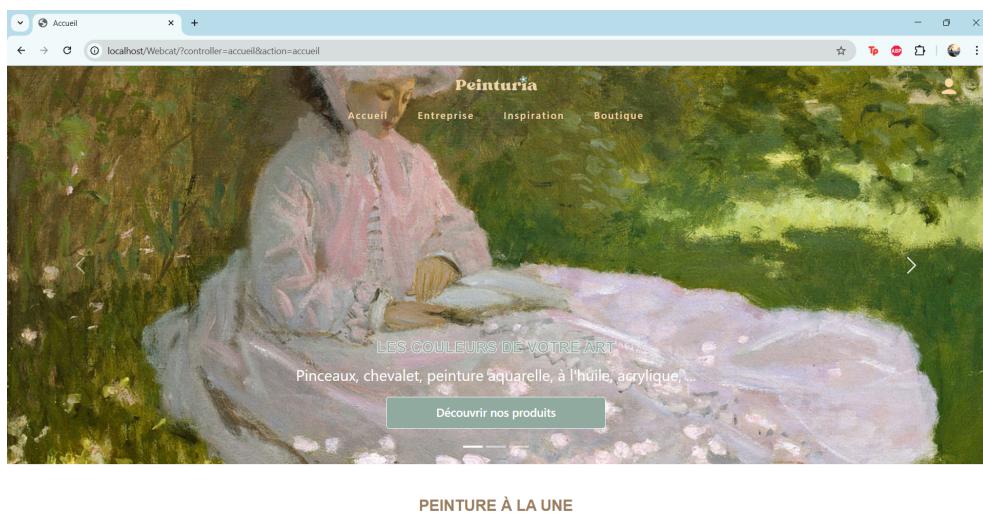


Sélecteur multimodal : du texte à l'image



Webcat - AUGIAS

R5.A.01
Développement avancé

Sommaire

1. Organisation du travail d'équipe - C3

1.1. Communication au sein de l'équipe	2
1.2. Principes de développement collaboratifs	3

2. Ingénierie logicielle - C1

2.1. Fonctionnement et choix architecturaux	4
2.2. Technologies utilisées	5
2.3. Lien vers le dépôt GitHub	6
2.4. Démonstration vidéo de l'application	6

3. Optimisation - C2

3.1. Justifications de l'algorithme de sélection choisi	6
3.2. Avantages et inconvénients de l'algorithme	7
3.3. Métriques de rapidité de réponse	8
3.4. Complexité et formalisation de l'algorithme	9

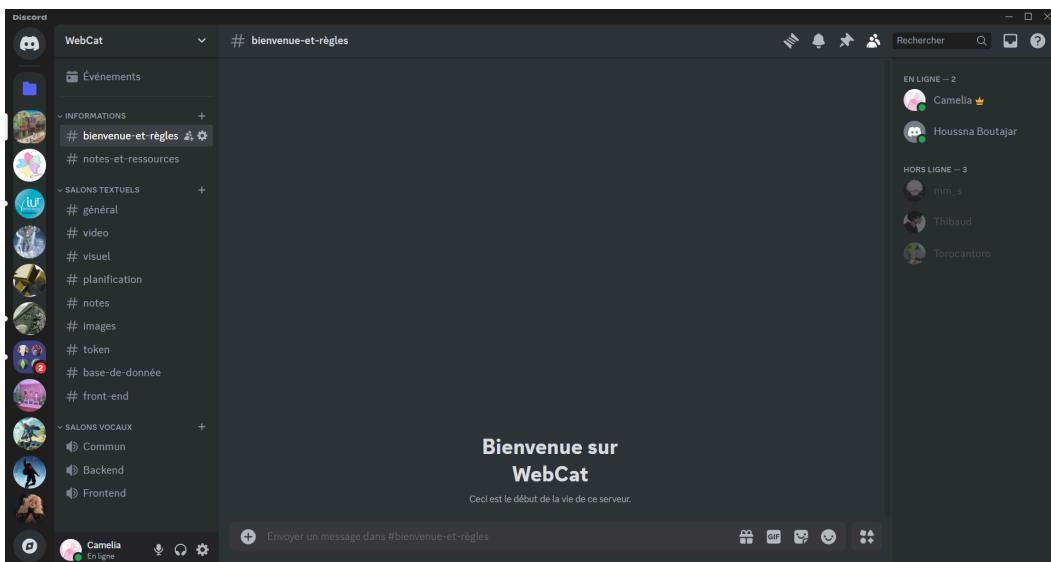
4. Aspects communicationnels - C3

4.1. Logo	10
4.2. Branding	10
4.3. Positionnement marketing	12

1. Organisation du travail d'équipe - C3

1.1. Communication au sein de l'équipe

Pour garantir une communication efficace et une organisation structurée, plusieurs outils et stratégies ont été mis en place. Un **serveur Discord** a été créé avec des **salons textuels dédiés**, comme un salon général pour les discussions communes et des salons spécifiques pour le frontend, le backend, la base de données et les planifications. Des **salons vocaux** adaptés aux besoins de chaque équipe ont aussi été créés. Il y a un salon pour l'équipe frontend, un pour l'équipe backend et un salon commun pour les échanges globaux. Le serveur permet aussi d'organiser des "événements" comme les réunions afin d'envoyer une notification à l'équipe avant le début de la réunion. Un **drive partagé** a également été créé pour centraliser tous les documents liés à l'organisation et la création de la vidéo de prévention. On y retrouve les notes de réunion et les ressources importantes par exemple.



Capture d'écran - Serveur Discord de Webcat

Lors de chaque séance, nous avons effectué un **point écrit où chacun partageait l'avancée de ses tâches**, complété par une **réunion hebdomadaire**, organisée en vocal ou en présentiel, pour discuter des priorités, des blocages éventuels et des ajustements à faire. La cheffe de projet intervenait pour redistribuer les tâches si nécessaire, mais lorsque l'organisation initiale était suffisante, chacun poursuivait le travail prévu dès le début du projet. Chaque membre pouvait aussi se proposer pour des tâches qu'il souhaite. Notre stratégie de communication repose sur une **approche flexible et collaborative**. Nous avons adopté un **management participatif**, où les décisions étaient prises collectivement, permettant à chaque membre de s'impliquer activement. Les équipes étaient également non fixes, laissant la possibilité à un membre spécialisé dans le frontend de contribuer temporairement à des tâches backend, selon ses compétences ou ses envies. Cette organisation a permis de maintenir une bonne dynamique de travail tout en favorisant l'entraide et la polyvalence au sein de l'équipe.

1.2. Principes de développement collaboratifs

La méthodologie **Agile** a été utilisée, combinée à des outils et pratiques de versionnement comme **GitHub**. Le projet a été découpé en **sprints hebdomadaires**, permettant de structurer les différentes étapes du développement tout en restant flexibles face aux besoins et aux imprévus. À la fin de chaque sprint, une **revue** était organisée pour valider les fonctionnalités implémentées et discuter des ajustements nécessaires pour les prochaines étapes. Ensuite, pour garantir la stabilité et l'organisation du code, nous avons utilisé **GitHub avec une stratégie basée sur des branches dédiées** : une pour le frontend, une pour le backend et une pour la base de données. Il y a aussi la branche main qui est la branche principale. Cette structure a permis à chaque équipe de **travailler en parallèle** tout en minimisant les risques de conflits. Avant chaque fusion sur la branche principale, le code était systématiquement revu en équipe pour garantir la qualité et éviter les erreurs. Par ailleurs, un fichier README.md a été créé pour documenter le projet, les fonctionnalités implémentées et les coordonnées de l'équipe. Dès le début du projet, une **documentation technique** a été créée pour formaliser nos idées initiales, lister les fonctionnalités obligatoires et poser les bases de la structure du projet. Des **maquettes** ont été élaborées dès le départ pour définir une vision claire de l'interface utilisateur et des interactions, même si ces maquettes ont évolué au fil du projet en fonction des nouvelles idées et des retours de l'équipe. De plus, pour l'équipe frontend, une attention particulière a été portée à la **standardisation des valeurs** en définissant des variables partagées. Cela a permis de garantir une cohérence visuelle et d'accélérer les ajustements futurs. Cette approche structurée a offert une base solide pour le développement tout en laissant de la place à **l'innovation et à l'amélioration continue**.

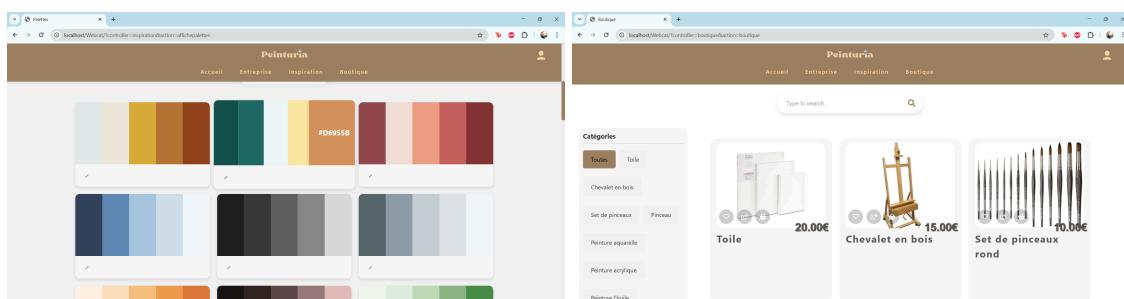


Maquettes des pages Accueil, Entreprise et Boutique

2. Ingénierie logicielle - C1

2.1. Fonctionnement et choix architecturaux

L'application est une **application web locale** conçue pour offrir une expérience utilisateur optimisée et intuitive. Ce site web est un site web commercial pour l'entreprise **Peinturia** qui **vend des peintures de tous types et des matériaux** en rapport avec la peinture. Cette application propose une recherche efficace basée sur des mots-clés et des catégories, permettant aux utilisateurs de trouver rapidement les articles de la boutique mais aussi d'une barre de recherche. Grâce à une **architecture responsive**, les images représentant les produits s'affichent de manière dynamique, offrant une navigation fluide et agréable sur tous les types d'appareils. L'application comprend plusieurs pages. La page **d'accueil** introduit l'utilisateur à l'univers de la plateforme et met en avant du contenu (articles et œuvres). La page **entreprise** présente l'histoire, la mission et les valeurs de Peinturia, renforçant ainsi l'identité de la marque. La page **inspiration** est dédiée à fournir des idées créatives, mettant en avant des **œuvres** d'art et des **palettes** de couleurs que les utilisateurs peuvent explorer et utiliser. La **boutique** offre une expérience complète grâce à un sélecteur multimodal qui combine une barre de recherche performante et une navigation par catégories à l'aide d'étiquettes. En complément, une page de **connexion** permet aux utilisateurs de s'identifier et d'accéder à leur page compte, où ils peuvent gérer leurs favoris et leurs informations personnelles. Dans le cas où ce serait un compte administrateur, la page redirige vers des formulaires pour ajouter du contenu à la boutique. L'application intègre également plusieurs fonctionnalités innovantes. Les utilisateurs peuvent **copier-coller directement les codes couleurs hexadécimaux des palettes proposées**, facilitant ainsi leur utilisation dans d'autres contextes. Une fonctionnalité de pipette permet de **récupérer les couleurs directement depuis les œuvres** affichées sur la page inspiration. Dans la boutique, une **recherche multimodale combine une barre de recherche traditionnelle et une recherche par catégorie**, offrant ainsi une expérience de navigation enrichie. La boutique permet également d'afficher les **fiches techniques des produits en un clic**. Les utilisateurs peuvent naviguer dans les œuvres et produits grâce à une **pagination fluide**. Enfin, les **comptes clients** permettent aux utilisateurs de personnaliser leur expérience et de conserver leurs favoris, leur panier et leur historique de navigation. Bien que l'ajout d'articles aux favoris et au panier ne soit pas encore entièrement implémenté, la structure est prête pour ces ajouts futurs. Dans le cas où l'utilisateur serait un **administrateur**, il est renvoyé sur une page pour ajouter des articles.



Pages Palette et Boutique

Le développement de l'application repose sur le **modèle MVC** (Modèle Vue Contrôleur), une architecture logicielle qui sépare clairement les différentes couches de l'application. Le **modèle** gère la logique métier ainsi que l'interaction avec la base de données **PostgreSQL**, garantissant une gestion efficace des données. La **vue**, qui utilise **CSS et JavaScript**, est responsable de l'affichage et de la présentation des informations aux utilisateurs, avec un design soigné et interactif. Le **contrôleur** joue le rôle d'intermédiaire, gérant les requêtes des utilisateurs, interagissant avec le modèle pour récupérer les données, et renvoyant une réponse sous forme de vue. Ces choix technologiques ont été faits pour garantir une application à la fois performante et évolutive. L'utilisation de **PostgreSQL** assure une gestion robuste et fiable des données, même avec des requêtes complexes. L'intégration de **PHP** pour le backend offre une solution flexible et compatible avec de nombreux environnements. Enfin, le choix de **CSS et JavaScript** pour le frontend permet de créer une interface visuelle attrayante tout en conservant une grande réactivité. Cette combinaison de technologies et d'approches garantit un produit de qualité, adapté aux besoins des utilisateurs et aux objectifs de l'entreprise. Ces choix permettent aussi une meilleure collaboration au sein de l'équipe car le modèle est séparé en plusieurs fichiers et tous les langages utilisés sont connus de l'équipe au complet.

2.2. Technologies utilisées

Pour le développement de l'application, une combinaison de **langages de programmation**, de **librairies** et de **frameworks** a été utilisée afin de répondre aux besoins spécifiques du projet, tout en garantissant une performance optimale et une expérience utilisateur agréable. **JavaScript** a été utilisé pour garantir une **cohérence totale entre le backend et le frontend**, en facilitant les interactions dynamiques et les fonctionnalités complexes de la plateforme. Le **HTML** et le **CSS** ont été employés pour concevoir un design **épuré, responsive et accessible**, répondant ainsi aux standards modernes du web. Le langage **PHP** a servi à gérer le backend, en se chargeant des interactions avec la base de données et des traitements côté serveur. Enfin, **Python** a été utilisé pour la **récupération des données** nécessaires à l'alimentation de la base de données, comme les informations sur les œuvres, les artistes, les descriptions et les palettes de couleurs. Python a également été utilisé pour **générer des fichiers SQL** afin de faciliter l'importation des données dans la base de données. Quant à la base de données, **PostgreSQL** est utilisée, choisie pour ses capacités de gestion de données complexes et sa performance optimale dans les environnements transactionnels. Elle a permis de gérer efficacement les informations des utilisateurs, des produits et des fonctionnalités de la boutique en ligne. Un des avantages majeurs de PostgreSQL est sa capacité à gérer des **relations complexes entre les différentes entités de la base de données**, ce qui a facilité la structuration des données liées, comme les relations entre les utilisateurs et leurs favoris, les produits et les catégories, ainsi que les œuvres et les couleurs associées. Cela garantit une intégrité des données et des performances accrues lors des requêtes impliquant des jointures complexes.

Pour accélérer le développement et garantir une interface moderne, nous avons utilisé **Bootstrap**, une bibliothèque CSS populaire qui permet de créer rapidement une mise en page responsive et esthétique. Plusieurs autres librairies ont été intégrées pour enrichir les fonctionnalités du site, telles que **Swiper.js** pour les carrousels d'images, offrant ainsi une navigation fluide et interactive. **Boxicons** et **Font Awesome** ont été utilisés pour les icônes, garantissant une interface visuellement attrayante. **Venobox** a été intégré pour afficher des galeries d'images et des contenus multimédias dans des modales élégantes. **JQuery et AJAX** (surtout pour la recherche multimodale) a aussi été intégré afin de simplifier la **manipulation du DOM** et de rendre l'interaction avec l'utilisateur plus dynamique. Il a permis d'optimiser certaines fonctionnalités, comme la gestion des événements et les animations, tout en assurant une compatibilité avec différents navigateurs. Grâce à **jQuery**, nous avons pu améliorer l'expérience utilisateur en facilitant l'ajout d'interactions et en réduisant la complexité du code **JavaScript**. Enfin, une **feuille de style CSS personnalisées** a été créée pour ajuster le design en fonction de l'identité visuelle de Peinturia.

2.3. Lien vers le dépôt GitHub

Lien vers le GitHub de l'équipe Webcat : <https://github.com/webcat24/Webcat/tree/main>

2.4. Démonstration vidéo de l'application

Courte vidéo de démonstration de l'application et de ses fonctionnalités :

<https://drive.google.com/file/d/1n9Q8sob4SuOUkZMzDXqOO2mcG22qyn6E/view?usp=sharing> Pour avoir les données, il faut exécuter tables.sql puis data.sql et changer les credentials (fichier dans le dossier Models) afin de relier votre base de données.

3. Optimisation - C2

3.1. Justifications de l'algorithme de sélection choisi

Après des recherches sur des **boutiques e-commerce**, nous avons découvert deux manières de concevoir l'algorithme de recherche. La première consiste à effectuer une **recherche directe** par input, où l'entrée utilisateur est comparée à un nom de produit spécifique (par exemple, en utilisant preg_match). La seconde méthode repose sur la **division de la requête utilisateur en tokens (mots-clés)**. Cette approche permet de détecter certains mots-clés correspondant à des catégories ou critères spécifiques, comme une couleur, une quantité ou un type de matériel. Nous avons opté pour cette seconde possibilité, en **traitant les mots individuellement dans la requête d'entrée**. Chaque mot est testé par rapport à **differents critères**, comme une correspondance avec une couleur, un type de matériau ou une quantité. Lorsqu'un mot correspond à l'un de ces critères, il est **enregistré et utilisé pour construire une requête SQL dynamique**. Cette requête interroge la base de données pour récupérer les éléments correspondant aux tokens détectés. Par exemple, lorsqu'un utilisateur cherche une couleur, le mot correspondant est stocké comme token, et une requête spécifique est générée pour renvoyer les peintures associées à cette couleur.

Le site étant codé en PHP, une **API** a été développée pour gérer ce processus. Elle prend la **requête utilisateur**, effectue la “**tokenisation**”, génère une **requête SQL** basée sur les catégories détectées, **interroge la base de données** et renvoie les **Résultats sous forme de JSON**. Pour intégrer cette API dans l’application, nous utilisons JavaScript avec des **fonctions asynchrones**. Ces fonctions, via **AJAX**, **envoient la requête à l’API**, **récupèrent la réponse JSON**, et **mettent à jour dynamiquement l’affichage** des produits dans la boutique en fonction des résultats. Enfin, notre base de données est structurée selon les principes de normalisation, afin d’éviter les redondances et de maximiser les performances lors des recherches. Elle permet de gérer efficacement des **critères** comme les types de peinture ou les caractéristiques des produits, garantissant ainsi une recherche **rapide et précise**. En résumé, ce système repose sur la décomposition des requêtes utilisateur en tokens, la construction de requêtes dynamiques adaptées, et une API performante pour assurer une expérience utilisateur fluide et intuitive.

3.2. Avantages et inconvénients de l’algorithme

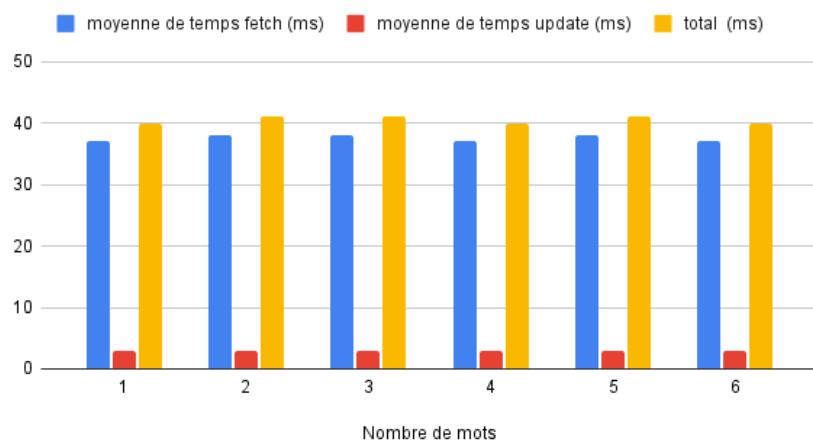
	Avantages	Inconvénients
Traitement des données	Traite divers types de requêtes, même celles non explicitement définies dans la base de données, comme la recherche "pas cher" en utilisant un token lié au prix	Nécessite de prédéfinir les catégories et les correspondances possibles, ce qui limite les mots qui ne sont pas catégorisés ou non prévus
Filtrage intelligent	Ignore les mots inutiles (déterminants, mots de liaison) et se concentre sur les mots pertinents	Certains cas nécessitent de tester des combinaisons de mots pour établir des liens logiques, ce qui augmente la complexité
Adaptation aux phrases	Transforme les requêtes complexes en mots-clés catégorisés pour extraire les informations utiles	Ajout de nouveaux critères ou tokens nécessite une vérification pour éviter les conflits
Rapidité	Réduit les comparaisons inutiles grâce à la priorisation des tests de correspondance et optimise le temps de traitement	Sur des machines moins puissantes, la mise à jour via JavaScript peut entraîner des ralentissements
Requêtes SQL dynamiques	Permet de construire des requêtes adaptées aux critères détectés, sans avoir besoin d’en créer une pour chaque type de recherche	Les modifications de certaines parties de l’algorithme sont complexes en raison des interactions entre les différents points
Performance BDD	Exploite les algorithmes d’optimisation intégrés aux bases de données pour maximiser la rapidité de récupération	L’algorithme est conçu spécifiquement pour ce projet et ne peut pas être facilement réutilisé pour d’autres projets sans modifications importantes

Affichage des résultats	Les tokens permettent d'organiser les recherches par priorité (exemple : tester d'abord les critères les plus probables comme "peinture")	L'ordre d'affichage des résultats doit être défini selon des priorités naturelles, ce qui peut ne pas convenir à tous les utilisateurs
-------------------------	---	--

3.3. Métriques de rapidité de réponse

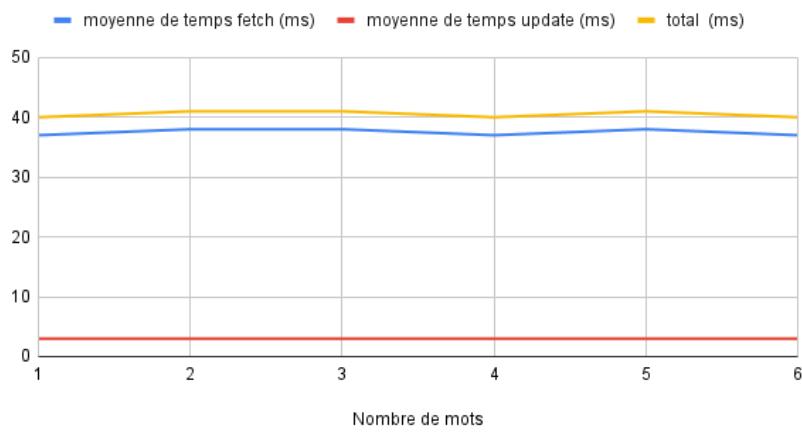
On remarque par ces deux formes de graphiques que nous avons des **temps constants** pour la recherche **peu importe la longueur** de cette dernière.

Moyennes de temps d'exécution selon la taille de la recherche



Graphique à colonnes

Moyennes de temps d'exécution selon la taille de la recherche



Graphique en courbe

Test réalisé sur processeur AMD Ryzen 7 5800X 8-Core 4.20 GHz

3.4. Complexité et formalisation de l'algorithme

Pour la partie “**tokenisation**”, la **complexité est linéaire ($O(n)$)** car elle dépend directement du nombre de mots dans l’entrée utilisateur. Plus l’entrée est grande, plus il y aura de mots à traiter pour vérifier une correspondance. Cependant, dans le cadre d’une boutique e-commerce, les recherches utilisateur restent souvent courtes, ce qui rend cet aspect négligeable en pratique. En ce qui concerne la **création des requêtes SQL**, la complexité est **constante ($O(1)$)**. Les requêtes dépendent uniquement des tokens collectés lors de la tokenisation, et les catégories à prendre en compte sont préalablement définies. Ce processus ne varie pas selon la longueur de l’entrée. De plus, lorsque la recherche est affinée, le nombre de résultats à afficher diminue, ce qui améliore encore les performances. Pour la partie **base de données**, la complexité est déterminée par le système de gestion de bases de données (SGBD). Les **données étant indexées**, les recherches dans la base ont une complexité logarithmique (**$O(\log n)$**), ce qui garantit des temps de réponse rapides et efficaces pour récupérer les résultats correspondant à la requête générée.

L’algorithme repose sur une **structure optimisée pour traiter les recherches utilisateur de manière rapide et efficace**. Les variables utilisées sont soigneusement nommées en fonction de leur rôle afin d’en améliorer la lisibilité et d’optimiser le traitement. Par exemple, les tokens possèdent des valeurs par défaut, comme "couleur" qui englobe toutes les couleurs disponibles, et ces valeurs ne sont mises à jour que si une correspondance avec une recherche utilisateur est détectée, évitant ainsi des calculs superflus. **Une fonction exécutée lors du chargement de la page récupère des données essentielles**, telles que la liste des couleurs disponibles dans la base de données, et les stocke dans des constantes pour **accélérer le processus de “tokenisation”** en réduisant le besoin de requêtes répétées. L’algorithme débute par la “**tokenisation**”, où les mots de l’entrée utilisateur sont **parcourus dans une boucle**. Pour chaque mot, des **conditions if** permettent de tester des correspondances avec des catégories prédéfinies, comme la couleur, le prix ou le type de produit. Dès qu’une correspondance est trouvée, l’instruction **“continue”** est utilisée pour passer au mot suivant, évitant tout traitement inutile. Les tokens collectés servent ensuite à générer dynamiquement une requête SQL adaptée aux critères identifiés, permettant de **récupérer uniquement les données pertinentes dans la base**. La base de données, optimisée avec **des index**, garantit une recherche rapide grâce à une complexité logarithmique comme précisé plus tôt, même pour des ensembles de données volumineux. Pour maximiser la précision et la rapidité, l’algorithme utilise des techniques comme les **expressions régulières** pour les correspondances, des structures conditionnelles pour affiner le matching, et des constantes préchargées pour limiter les appels à la base. Ce **fonctionnement modulaire**, combinant **boucles et conditions**, exploite les capacités des systèmes de gestion de bases de données (SGBD) pour répondre efficacement aux besoins des utilisateurs, **tout en maintenant une complexité globale raisonnable**.

4. Aspects communicationnels - C3

4.1. Logo

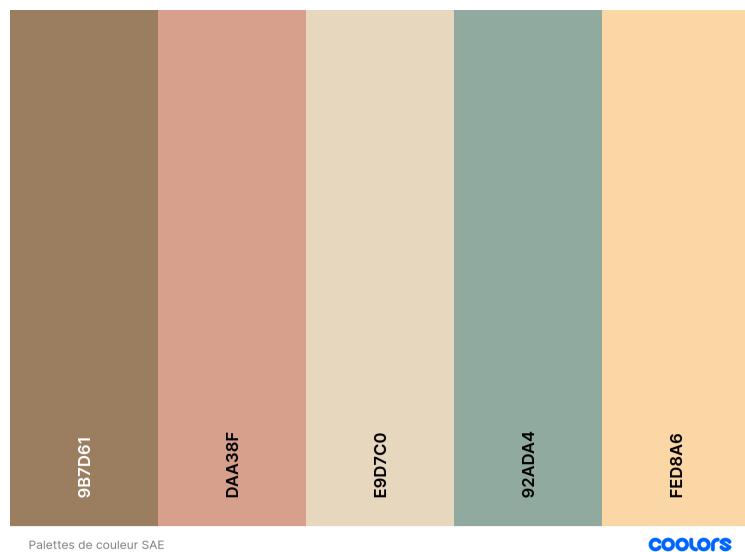
Le logo de Peinturia a été conçu sur **Canva** pour refléter l'essence même de l'entreprise, qui combine **créativité, élégance et expertise**. Il intègre un pinceau en train de peindre le nom "Peinturia" avec une touche artistique, symbolisant l'univers de la peinture et de l'art. On y retrouve également le slogan de l'entreprise "**Les couleurs de votre art**". Les couleurs choisies pour le logo sont issues d'une **palette naturelle**, équilibrant des teintes douces comme le beige et des couleurs vibrantes telles que le turquoise ou le rose, transmettant une image à la fois professionnelle et inspirante tout en captant l'attention des amateurs et professionnels de la peinture. Il existe également un **logo texte, ou logo long**, qui se limite au nom de l'entreprise, orné d'une fleur sur le "i" pour évoquer la nature et un style champêtre, en référence aux peintures impressionnistes comme celles de Monet. Ce logo long est positionné dans le header, au-dessus du menu, afin de mettre en avant le nom de l'entreprise tout en adoptant un aspect élégant et chic. Le logo principal, plus détaillé, se trouve dans le footer du site. Ce dernier est également prévu pour être utilisé sur les packagings, les produits et les supports publicitaires de Peinturia, renforçant ainsi l'identité visuelle de la marque à travers tous les points de contact.



Les logos de Peinturia

4.2. Branding

La **palette de couleurs** de Peinturia incarne des valeurs de simplicité, de douceur et d'harmonie. Chaque teinte a été choisie avec soin pour refléter la nature et transmettre un message cohérent avec l'identité de l'entreprise. Le marron clair symbolise la stabilité et la connexion à la terre, traduisant l'engagement écologique de Peinturia. Le rose terne apporte une touche chaleureuse et humaine, inspirant la créativité, tandis que le beige clair, minimaliste et épuré, évoque l'élégance et l'accessibilité. Le turquoise, apaisant et équilibré, renforce les valeurs durables et naturelles de l'entreprise, tandis que le jaune pâle insuffle une énergie lumineuse et stimulante. Ces teintes, associées à une **typographie inclusive et élégante**, reflètent une approche universelle et accessible, tout en consolidant le professionnalisme de la marque.



Palette de couleur du site web et du logo

L'identité visuelle de Peinturia s'articule autour d'une alliance subtile entre **nature, créativité et expertise**. Inspirée par l'**univers artistique** et les **paysages champêtres**, elle intègre des éléments distinctifs comme la fleur du logo texte/long et le pinceau dans le logo principal. Ces détails évoquent l'univers impressionniste de peintres tels que Monet, comme cité plus tôt, créant un lien naturel avec la créativité et l'harmonie. Le **site web**, quant à lui, adopte un **design épuré et apaisant** qui reflète fidèlement cette identité, offrant une expérience visuelle cohérente et élégante. Une grande attention a aussi été portée quant au choix des images du site web. Cette attention portée à la **cohérence entre branding et supports visuels** permet de captiver autant les amateurs que les professionnels, en soulignant l'accessibilité et l'inspiration offertes par Peinturia. Les choix esthétiques ne se limitent pas à l'apparence : ils traduisent aussi les engagements écologiques et inclusifs de l'entreprise (fictif). La palette naturelle et les éléments visuels mettent en lumière son partenariat avec des producteurs locaux et son respect de l'environnement. La simplicité et l'universalité de la typographie renforcent également son ouverture à tous, quels que soient les besoins ou particularités des utilisateurs. Le design doux et fluide du site web incarne ces valeurs, créant une expérience intuitive et accueillante. Ces **valeurs et engagements** sont énoncés sur la page entreprise.



Différentes images du site web

4.3. Positionnement marketing

Le positionnement marketing de Peinturia est bien pensé et répond aux attentes modernes des consommateurs. En se positionnant comme une **marque premium accessible**, Peinturia réussit à combiner des produits de qualité supérieure tout en restant abordable. Cela attire une clientèle à la recherche de **peintures haut de gamme** sans se tourner vers des options inaccessibles. Ce positionnement répond à une tendance actuelle où la qualité et le prix juste sont des critères décisifs dans le choix d'un produit. L'**engagement écologique** de l'entreprise constitue un autre atout majeur. Dans un contexte où la consommation responsable est de plus en plus importante, mettre en avant des produits respectueux de l'environnement permet à Peinturia de capter l'attention des clients soucieux de leur impact écologique. Ce choix de valoriser des produits durables renforce l'image de la marque et répond à une demande croissante pour des alternatives écologiques dans tous les secteurs. De plus, la collaboration avec des **producteurs locaux français** permet de valoriser le **savoir-faire artisanal** et de répondre à la demande croissante pour des produits authentiques et **made in France**. Ce partenariat renforce la crédibilité de la marque en tant que défenseur du **produit local**, et répond à la tendance de plus en plus forte en faveur de la production responsable. Le **storytelling** autour de l'entreprise, fondée par cinq anciens camarades passionnés d'art, apporte une **dimension humaine et émotionnelle** à la marque. Ce lien authentique permet aux clients de s'identifier à l'histoire de l'entreprise, favorisant ainsi la **fidélité** et l'engagement. Une telle approche devient un véritable atout dans un marché où les consommateurs cherchent à soutenir des marques avec une **histoire forte**. Enfin, la stratégie de ciblage de Peinturia, en se concentrant sur des segments spécifiques tels que les **artistes professionnels**, les **amateurs exigeants** et les **enseignants en arts plastiques**, permet à la marque d'être plus précise dans ses campagnes publicitaires. Elle peut ainsi répondre aux besoins spécifiques de ces groupes, créant ainsi une **relation de proximité** avec ses consommateurs. Les **faux réseaux sociaux** et **comptes clients** permettent d'entretenir une relation continue avec les clients et de les fidéliser, tout en cultivant un sentiment de **communauté exclusive**. L'utilisation d'**influenceurs dans le monde artistique** permet de renforcer l'image de marque et d'attirer de nouveaux clients. Le positionnement de Peinturia repose donc sur une **stratégie complète et cohérente**, combinant qualité, engagement écologique, authenticité et proximité avec le client. Cette approche permet à la marque de se différencier et de construire une base de clients fidèle et engagée, tout en attirant de nouveaux consommateurs grâce à des campagnes ciblées et des partenariats.



The screenshot shows a web browser window with the URL localhost/Websit/ in the address bar. The page title is "A Propos". The header features the brand name "Peinturia" with a small image of a painting. Below the header, there are navigation links for "Accueil", "Entreprise", "Inspiration", and "Boutique". A user icon is also present. The main content area is titled "Nos Valeurs". It contains a paragraph of text about the company's values, mentioning its commitment to environmental respect, local producers, diversity of talents and cultures, equality of opportunities, partnerships with French brands, support for local craftsmanship, and a passion for art and creation. Two small images are displayed: one showing people in a workshop or studio setting, and another showing a field of flowers.

Page Entreprise