

Documentation High Level – Fonctions PBS pour Dream Habitat

1. Interface Utilisateur (UI) et Expérience Utilisateur (UX)

1.1. Page d'Accueil

- **Exigence 1** : Présentation générale du projet avec une UI attrayante.
 - *Description* : La page d'accueil doit offrir une vue d'ensemble claire et attrayante du projet Dream Habitat, mettant en avant ses principales fonctionnalités et avantages. Une interface utilisateur (UI) soignée et moderne doit capter l'attention des visiteurs dès leur arrivée.
- **Exigence 2** : Images de haute qualité et design interactif.
 - *Description* : Utiliser des images haute résolution pour illustrer les concepts de design intérieur. Le design interactif doit permettre une navigation fluide et engageante, avec des éléments visuels et des animations qui améliorent l'expérience utilisateur.

1.2. Module de Capture de Photos

- **Exigence 3** : Permettre aux utilisateurs de capturer et télécharger des photos de leur espace intérieur.
 - *Description* : Fournir un module intégré qui permet aux utilisateurs de prendre des photos de leur espace de vie via leur appareil et de les télécharger directement sur la plateforme pour une personnalisation future.
- **Exigence 4** : Interface intuitive pour une UX optimale.
 - *Description* : L'interface doit être simple et intuitive, guidant l'utilisateur à travers le processus de capture et de téléchargement de photos sans friction, assurant une expérience utilisateur (UX) optimale.

1.3. Saisie des Préférences de Style

- **Exigence 5** : Interface intuitive pour entrer les préférences de style.
 - *Description* : Offrir un formulaire intuitif où les utilisateurs peuvent entrer leurs préférences de style de design intérieur. L'interface doit être claire et facile à utiliser.
- **Exigence 6** : Catégories de styles prédéfinies avec des visuels attractifs.
 - *Description* : Présenter des catégories de styles prédéfinies avec des visuels attractifs pour aider les utilisateurs à choisir et visualiser leurs préférences de manière efficace.

1.4. Galerie de Designs

- **Exigence 7** : Exemples de designs générés par l'IA présentés de manière interactive.
 - *Description* : Fournir une galerie interactive de designs générés par l'intelligence artificielle (IA) qui montre les différentes possibilités de design intérieur en fonction des préférences de l'utilisateur.
- **Exigence 8** : Visualisation sous différents angles avec une expérience fluide.
 - *Description* : Permettre aux utilisateurs de visualiser les designs sous différents angles et perspectives, offrant une expérience de navigation fluide et immersive.

1.5. Processus de Personnalisation

- **Exigence 9 :** Interface utilisateur pour personnaliser les éléments de design en temps réel.
 - *Description :* Offrir une interface utilisateur où les utilisateurs peuvent personnaliser les éléments de leur design en temps réel, avec des options de modification instantanée et de prévisualisation.
- **Exigence 10 :** Expérience utilisateur fluide et réactive pendant la personnalisation.
 - *Description :* Assurer une expérience utilisateur fluide et réactive, minimisant les temps de latence et les interruptions lors de la personnalisation des designs.

2. Intégration de l'IA

2.1. Algorithme de Fusion Créative

- **Exigence 11 :** Développer une solution pour fusionner de manière créative différents éléments de design intérieur.
 - *Description :* Pour offrir des solutions de design intérieur personnalisées, j'ai intégré deux API distinctes qui permettent de générer des images en fonction des photos de pièces. Ces API permettent de modifier les éléments de design en respectant les préférences de style des utilisateurs.
 - **API 1 :** Cette API permet de générer des images à partir de photos de pièces déjà meublées. Elle permet d'ajuster ou de modifier les éléments présents dans une pièce, tout en respectant les préférences de style de l'utilisateur.
 - **API 2 :** Cette API permet de créer des images de pièces non meublées, en ajoutant des meubles et des éléments décoratifs de manière créative, selon les préférences de l'utilisateur pour le style et l'ambiance de la pièce.
- **Exigence 12 :** Assurer l'adaptation aux préférences changeantes des utilisateurs.
 - *Description :* La solution est conçue pour s'adapter aux préférences changeantes des utilisateurs. Lorsqu'un utilisateur modifie ses préférences, comme le style, la solution ajuste automatiquement ses propositions en fonction de ces changements.

Cela inclut la possibilité de réagir aux modifications des utilisateurs, qu'il s'agisse de nouvelles idées de décoration ou d'un changement dans le style souhaité.

3. Gestion de Contenu et API

3.1. Système de Gestion de Contenu et API

- **Exigence 13 :** Développer une API pour la gestion des contenus en utilisant Laravel ou Node.js avec Express.
 - *Description :* Créer une API robuste pour gérer le contenu de la plateforme, en utilisant des technologies telles que Laravel ou Node.js avec Express pour assurer une performance et une scalabilité optimales.

- **Exigence 14 :** Assurer la compatibilité multi-utilisateur avec différents niveaux d'accès.
 - *Description :* Garantir que le système prend en charge plusieurs utilisateurs avec différents niveaux d'accès, permettant une gestion efficace des rôles et des permissions.
- **Exigence 15 :** Utiliser Swagger pour documenter toutes les API.
 - *Description :* Utiliser Swagger pour générer une documentation complète et à jour des API, facilitant l'intégration et l'utilisation par les développeurs.

4. Sécurité et Performance

4.1. Sécurité du Site

- **Exigence 16 :** Mettre en place des mesures de sécurité pour protéger les données des utilisateurs et prévenir les attaques.
 - *Description :* Implémenter des protocoles de sécurité rigoureux pour protéger les données des utilisateurs contre les violations et les attaques, incluant le chiffrement des données sensibles.
- **Exigence 17 :** Utiliser le protocole HTTPS pour toutes les communications.
 - *Description :* Assurer que toutes les communications entre les utilisateurs et la plateforme se font via HTTPS pour garantir la confidentialité et l'intégrité des données échangées.

4.2. Performance et Scalabilité

- **Exigence 18 :** Optimiser le site pour des temps de chargement rapides.
 - *Description :* Améliorer les performances du site en minimisant les temps de chargement, utilisant des techniques telles que la compression des ressources, la mise en cache et le chargement asynchrone.
- **Exigence 19 :** Assurer la scalabilité pour gérer un grand nombre d'utilisateurs simultanés.
 - *Description :* Concevoir l'architecture du site pour qu'elle soit scalable, capable de gérer un nombre croissant d'utilisateurs simultanés sans dégradation des performances.

5. DevOps

5.1. Intégration Continue (CI)

- **Exigence 20 :** Mettre en place des pipelines CI pour automatiser les tests et les builds à chaque modification de code.
 - *Description :* Configurer des pipelines d'intégration continue (CI) pour automatiser les processus de test et de build à chaque modification du code, assurant une qualité constante du code livré.

5.2. Déploiement Continu (CD)

- **Exigence 21 :** Automatiser le déploiement des applications sur les environnements de développement, de test et de production.

- *Description* : Mettre en place des pipelines de déploiement continu (CD) pour automatiser le déploiement des applications sur différents environnements, de la phase de développement à la production.

5.3. Surveillance et Logging

- **Exigence 22** : Mettre en place des outils de surveillance pour suivre les performances et la disponibilité du site et de l'API.
 - *Description* : Déployer des outils de surveillance pour monitorer en temps réel les performances et la disponibilité du site et des API, permettant de détecter et résoudre rapidement les problèmes.
- **Exigence 23** : Mettre en place des systèmes de logging pour suivre et diagnostiquer les erreurs en temps réel.
 - *Description* : Implémenter des systèmes de logging complets pour enregistrer et diagnostiquer les erreurs en temps réel, facilitant la maintenance et l'amélioration continue du système.

Livrables du Projet

1. **Documentation du Code**
 - Inclure la documentation complète et à jour du code avec spécification des dépendances.
2. **Documentation Utilisateur**
 - Fournir des instructions claires et compréhensibles pour les utilisateurs finaux.
3. **Rapports de Tests**
 - Inclure les résultats des tests unitaires, d'intégration, et d'acceptation.
4. **Versions du Site Web et de l'API**
 - Livrer des versions stables du site web et de l'API à la fin de chaque sprint.
5. **Configuration des Pipelines CI/CD**
 - Documentation et configuration des pipelines d'intégration et de déploiement continu.
6. **Rapports de Surveillance et de Logging**
 - Rapports sur les performances et les erreurs collectés à partir des outils de surveillance et de logging.
7. **Documentation Swagger de l'API**
 - Documentation détaillée des endpoints API, des méthodes, des paramètres et des réponses.
8. **Wireframes et Prototypes**
 - Versions basse et haute fidélité des designs pour validation et tests.
9. **Rapports de Tests Utilisateurs**
 - Résultats des tests utilisateurs pour guider les améliorations UI/UX.