Documentation Img2brick

SOMMAIRE :

1. Analyse Fonctionnelle

1.1. Objectif du projet

Img2Brick est une application web conçue pour convertir des images téléchargées par les utilisateurs en modèles de mosaïques en briques. Le système gère l'intégralité du processus, de l'importation de l'image originale à la commande d'un kit de construction physique.

1.2. Parcours utilisateur

Le flux de l'application est structuré selon les étapes suivantes :

- **Accueil et Importation :** L'utilisateur arrive sur une interface de dépôt (drag & drop) pour soumettre son image.
- **Recadrage (Crop) :** L'outil permet de redimensionner l'image selon des ratios spécifiques (Carré, 4:3, 16:9) ou libres pour l'adapter aux dimensions des plaques de base (32x32, 64x64, 96x96 tenons).
- **Génération de Mosaïques :** Le système propose plusieurs variantes de rendu : une version avec nuances de bleu accentuées, une version avec nuances de rouge, et une version en noir et blanc.
- **Commande et Paiement :** Une fois le modèle choisi, l'utilisateur saisit ses informations de livraison et procède à un paiement simulé pour valider sa commande.

1.3. Gestion du multilingue

L'application intègre un système de traduction complet stocké en base de données. Elle supporte actuellement le français et l'anglais, permettant une adaptation dynamique de l'interface (boutons, messages d'erreur, labels de formulaire) selon les préférences de l'utilisateur.

2. Analyse Technique

2.1. Architecture logicielle

Le projet est développé en **PHP 8.4** et suit une structure organisée pour séparer les responsabilités :

- **Point d'entrée (`index.php`) :** Gère la redirection initiale vers la vue principale.

- **Contrôleurs (`control/`) :** Contiennent la logique métier, notamment la gestion des sessions, le chargement des variables d'environnement via `phpdotenv` et la configuration globale.
- **Modèles (`models/`) :** Assurent l'interface avec la base de données pour les utilisateurs, les images et les commandes.
- **Vues (`views/`) :** Regroupent les fichiers d'interface utilisateur (HTML/PHP), les feuilles de style CSS et les scripts JavaScript pour l'interactivité (notamment pour le module de recadrage et la gestion des fichiers).

2.2. Structure de la base de données

Le schéma SQL (`aissyne_img2brick`) repose sur six tables principales :

- **`users` :** Stocke les informations de compte (email, username, mot de passe haché) et l'état de validation (2FA).
- **`images` :** Stocke les données binaires des images (`longblob`) et les identifiants associés aux utilisateurs.
- **`commande` :** Lie un utilisateur, une image et des coordonnées de livraison, tout en enregistrant le montant et la date de la transaction.
- **`cord_banquaire` :** Conserve les informations de paiement (numéro de carte et CVV hachés).
- **`translations` :** Table pivot pour le support multilingue (clé, langue, texte).
- **`tokens` :** Gère les codes de vérification temporaires pour la sécurité des comptes.

2.3. Sécurité et composants tiers

- **Gestion des dépendances :** Le projet utilise **Composer** pour gérer les bibliothèques externes.
- **Bibliothèques utilisées :** * `vlucas/phpdotenv` : Pour la gestion sécurisée des variables de configuration (`.env`).
    - `phpmailer/phpmailer` : Pour l'envoi sécurisé des emails de confirmation et des codes de vérification.
    - `symfony/polyfill-ctype` : Pour assurer la compatibilité des fonctions de vérification de caractères.
- **Sécurité des données :** Les mots de passe et les données bancaires sensibles sont hachés (utilisation de `bcrypt` visible dans le dump SQL) avant stockage. Un système de **Captcha** est également présent pour sécuriser les formulaires de connexion et d'inscription.