Guide de maintenabilité du projet

1. Objectif de la Restructuration

L'ancienne architecture du projet (éclatée et peu organisée) a été réorganisée en suivant une structure MVC (« Model-View-Controller ») pour :

- Améliorer la lisibilité et la maintenabilité du code.
- Séparer les différentes responsabilités (logique, présentation, stockage).
- Faciliter l'ajout de nouvelles fonctionnalités et la gestion des données.

2. Changements Clés

Ancienne Structure

- Beaucoup de fichiers PHP dispersés dans plusieurs dossiers sans réelle distinction des responsabilités.
- Fichiers de base de données (« .sql ») mélangés avec des fichiers applicatifs.
- Fichiers PHP directement accessibles dans plusieurs sous-dossiers (« inscriptions » dans ud22, ud29, etc.).
- Absence de clarté entre la logique et l'affichage.

Nouvelle Structure

1. Dossier `app/` *(Nouvelle logique applicative)*

- **`controllers/`** : Contient `DisplayController.php`, centralisant la logique de contrôle pour l'affichage.
- **`views/`** : Stocke les fichiers de présentation HTML/PHP (ex : `bzh.php`, `Accueil.html`).
- **Les sous-dossiers `bzh/ud22`, `bzh/ud29`, etc.** sont maintenant rangés dans `views/`, clarifiant qu'ils concernent l'affichage.

2. Dossier `config/` *(Nouvelle gestion de la configuration)*

- Ajout de `app.php`, `database.php`, `init.php` et `ui.php` pour centraliser les paramètres de l'application et la connexion à la base de données.

3. Dossier `lib/` *(Fonctions réutilisables)*

- Contient des fichiers PHP servant de bibliothèque de fonctions (`front.php`, `mysql_functions.php`, etc.).
- Facilite la gestion et la réutilisation du code.

4. Dossier `public/` *(Fichiers accessibles publiquement)*

- Contient `index.php` comme point d'entrée principal.
- Un sous-dossier `assets/` pour stocker les images, icônes et ressources statiques.

5. Dossier `BDD/` *(Bases de données)*

- Contient uniquement les fichiers `.sql`, maintenant clairement séparés du code applicatif.

3. Bonnes Pratiques

- **Respecter la structure actuelle** : Ajouter de nouveaux fichiers dans les dossiers appropriés.
- **Utiliser le `controller` pour la logique** : Ne pas mettre de logique directement dans les fichiers de `views/`.
- **Factoriser le code** : Placer les fonctions récurrentes dans `lib/` pour éviter la duplication.
- **Centraliser les configurations** : Toute modification des paramètres doit passer par `config/`.