

Documentation Technique du Projet "Tout pour un Nouveau-Né"

1. Introduction

Ce document technique décrit l'architecture, les composants, et la configuration du projet "Tout pour un Nouveau-Né". Le site est déployé sur Heroku, utilisant Stackhero MySQL pour la base de données relationnelle et MongoDB Atlas pour la base de données NoSQL.

2. Environnement de Développement

2.1 Outils Utilisés

- **IDE** : Visual Studio Code
- **Serveur Local** : XAMPP/MAMP (utilisé pour le développement local)
- **Système de Versioning** : Git (repos stocké sur GitHub)
- **Déploiement** : Heroku
- **Base de Données Relationnelle** : Stackhero MySQL pour Heroku
- **Base de Données NoSQL** : MongoDB Atlas
- **Langage de Programmation** : PHP 7.x
- **Front-End** : HTML5, CSS3, JavaScript, Bootstrap 4.x

2.2 Configuration de l'Environnement

1. Cloner le Dépôt :

```
git clone <url-du-repository>
```

```
cd nom-du-projet
```

2. Configuration de l'Environnement Local :

- Configurer le serveur local (XAMPP/MAMP) pour pointer vers le dossier public/ comme racine du serveur.
- Créer une base de données MySQL localement en utilisant le fichier init.sql fourni.

3. Installation des Dépendances :

- Assurez-vous que les extensions PHP pour MySQL et MongoDB sont activées.

4. Fichier .env :

- Créez un fichier .env à la racine du projet pour stocker les variables d'environnement nécessaires au projet, notamment les informations de connexion à la base de données MySQL et MongoDB.

```
DB_HOST=<hostname>
```

```
DB_USER=<username>
```

```
DB_PASSWORD=<password>
```

```
DB_NAME=<database_name>
```

```
MONGODB_URI=<mongodb_connection_string>
```

5. Lancer le Serveur :

- Ouvrez XAMPP/MAMP et démarrez les services Apache et MySQL.
- Accédez à `http://localhost/<nom-du-projet>/public/` dans votre navigateur.

3. Déploiement sur Heroku

3.1 Prérequis

- Installer l'Heroku CLI.
- Créer un compte Heroku et configurer votre projet.

3.2 Déploiement

1. Création d'une Application Heroku :

```
heroku create nom-du-projet
```

2. Ajout de l'Add-On Stackhero MySQL :

```
heroku addons:create stackhero-mysql
```

3. Configuration de la Base de Données MongoDB Atlas :

- Créez une base de données MongoDB sur Atlas.
- Ajoutez votre chaîne de connexion MongoDB comme variable d'environnement :

```
heroku config:set MONGODB_URI=<mongodb_connection_string>
```

4. Déploiement du Code :

```
bash
```

```
Copier le code
```

```
git push heroku main
```

5. Migration de la Base de Données :

- Utilisez l'interface MySQL de Stackhero pour importer votre base de données ou utilisez une migration automatisée si nécessaire.

4. Architecture du Projet

4.1 Structure des Dossiers

└─ /config/

└─ src/

➔ Dans src il y a les dossiers :

- Models ;
- Public ;
- Views ;
- Admin, Doctor, Parent ;

4.2 Structure de la Base de Données

4.2.1 Base de Données MySQL (Stackhero)

- **Utilisateurs :**
 - id (Primary Key, Auto Increment)
 - nom_utilisateur (VARCHAR)
 - email (VARCHAR)
 - mot_de_passe (VARCHAR)
 - role_id (INT)
 - date_creation (TIMESTAMP)

- **Guides :**

- id (Primary Key, Auto Increment)
- titre (VARCHAR)
- contenu (TEXT)
- auteur_id (Foreign Key référant à Utilisateurs)
- date_creation (TIMESTAMP)

- **Commentaires :**

- id (Primary Key, Auto Increment)
- guide_id (Foreign Key référant à Guides)
- utilisateur_id (Foreign Key référant à Utilisateurs)
- contenu (TEXT)
- date_creation (TIMESTAMP)
- statut (ENUM: 'approuvé', 'en attente', 'rejeté')

- **Avis Médicaux :**

- id (Primary Key, Auto Increment)
- titre (VARCHAR)
- contenu (TEXT)
- medecin_id (Foreign Key référant à Utilisateurs)
- date_creation (TIMESTAMP)

- **Autres Tables :**

- quizzes, questions, reponses, resultats, etc., gérant les quiz et les scores.

4.2.2 Base de Données MongoDB (MongoDB Atlas)

- **scores** (Collection)

- user_id (Référence à l'ID utilisateur MySQL)
- total_score (Nombre total de points)
- quiz_scores (Array d'objets contenant quiz_id et score)
- created_at (Date de création du document)
- updated_at (Date de la dernière mise à jour du document)

5. Sécurité

5.1 Sécurité des Données

- **CSRF Protection** : Utilisation de jetons CSRF pour sécuriser les formulaires.
- **Validation des Données** : Sanitation et validation des entrées utilisateur côté serveur.
- **Hashing des Mots de Passe** : Les mots de passe sont hachés avec `password_hash()`.

5.2 Gestion des Sessions

- **Sessions PHP** : Les sessions sont utilisées pour la gestion des connexions utilisateur. Les identifiants de session sont régénérés après chaque connexion pour prévenir les attaques de fixation de session.

6. Maintenance et Mise à Jour

6.1 Maintenance

- **Mises à jour du Code** : Les mises à jour peuvent être déployées via Git avec `git push heroku main`.
- **Mises à jour des Packages** : Composer et npm (si utilisé pour le front-end) doivent être utilisés pour gérer les dépendances.
- **Sauvegarde de la Base de Données** : Configurer des sauvegardes régulières pour Stackhero MySQL et MongoDB Atlas.

6.2 Support Technique

- **Logs** : Les logs peuvent être visualisés via `heroku logs --tail`.
- **Monitoring** : Utilisation des outils de monitoring fournis par Heroku pour surveiller les performances de l'application.

Cette documentation technique offre une vue d'ensemble du projet, de son architecture, de sa configuration, ainsi que des meilleures pratiques de sécurité et de maintenance. Elle sert de référence pour comprendre et maintenir le site web "Tout pour un Nouveau-Né" dans le cadre d'un déploiement sur Heroku.