

Dossier Projet

Développeur Web et Web Mobile

Session Janvier 2025

AMAR Jean-pierre

Sommaire

Introduction

- I. Compétences du référentiel couvertes par le projet
 - A. Développer la partie front-end d'une application web et web mobile sécurisée
 - B. Développer la partie back-end d'une application web et web mobile sécurisée
- II. Contexte du projet
 - A. Présentation
 - B. Cahier des charges
 - C. Contraintes et livrables attendus
 - D. Environnement humain et technique
- III. Réalisations côté front-end
 - A. Maquettes de l'application
 - B. Captures d'écran des interfaces utilisateur
 - C. Extraits de code des interfaces statiques
 - D. Extraits de code de la partie dynamique
 - E. Choix techniques et sécurité
- IV. Réalisation côté back-end
 - A. Présentation de la base de données
 - B. Extraits de code des composants d'accès aux données
 - C. Éléments de sécurité de l'application
 - D. Jeu d'essai et analyse des résultats
 - E. Choix techniques et sécurité
- V. RGPD
- VI. Veille effectuée

Conclusion

Annexes

- A. Maquettes des interfaces utilisateur
- B. Capture d'écran et code correspondant
- C. Code des composants métier significatifs
- D. Code des composants d'accès aux données significatifs

I. Compétences du référentiel couvertes par le projet

A. Développer la partie front-end d'une application web et web mobile sécurisée

1. Installer et configurer son environnement de travail en fonction du projet web ou web mobile

- **IDE** : PhpStorm
- **Logiciel de prise de notes** : Obsidian
- **Environnement de serveur local** : Mamp
 - Dans le future, je souhaite utiliser Docker pour lancer le serveur web
- **Logiciel de maquettage** : Figma
- **Logiciel pour le MCD** : Jmerise
- **Hébergement** : En Locale
 - Dans le future, je prévu de l'héberger sur mon serveur O2switch
- **Logiciel de versionning** : Git sur Github
 - Dans le future, j'ai prévu de stocker le versionning sur mon serveur

2. Maquetter des interfaces utilisateur web ou web mobile

- Utilisation de l'application Figma pour le maquettage des interfaces
- Inspiration web

3. Réaliser des interfaces utilisateur statiques web ou web mobile

- Utilisation du langage de balisage HTML, CSS, et le framework CSS Tailwind

B. Développer la partie back-end d'une application web et web mobile sécurisée

4. Mettre en place une base de données relationnelle

- Création d'un schéma MCD a l'aide de Jmerise
- Utilisation de MySQL

5. Développer des composants d'accès aux données SQL et NoSQL

- J'ai développé des méthodes en suivant le **CRUD** (create, read, update, delete) qui est un moyen mnémotechnique pour les quatre fonctions de base du stockage persistant.

6. Développer des composants métier côté serveur

- J'ai créé des **Controller** et des **Models** pour suivre mon architecture MVC

7. Documenter le déploiement d'une application dynamique web ou web mobile

- Mon application est pour le moment en local pour pouvoir effectuer le développement plus simplement, et lors du passage en production, je l'hébergeait sur mon serveur O2Switch

II. Contexte du projet

1. Présentation

Le projet **Baby Tracker** vise à concevoir et développer une application web et web mobile permettant aux parents de suivre les activités et les progrès de leurs bébés. L'application offre une solution simple, sécurisée et intuitive pour gérer des informations clés comme les repas, les siestes, les changements de couches, et les mesures (poids, taille).

Elle a pour objectif de simplifier le quotidien des jeunes parents en leur proposant une plateforme centralisée pour consigner et consulter ces données à tout moment.

Cette application web peut également être déployée auprès des collectivités, notamment des crèches, assistante maternelle... En effet, ses informations sont communiquées aux parents mais dans certains cas, le support utilisé peut être obsolète. Dans le cas de la crèche dans laquelle mon enfant est, ses informations sont communiquées sur papier.

2. Cahier des charges

Objectifs Fonctionnels :

- Permettre l'authentification sécurisée des utilisateurs.
- Offrir une interface intuitive pour ajouter, modifier ou supprimer des bébés.
- Enregistrer et suivre les activités quotidiennes des bébés :
 - **Repas** : heures, quantité et notes associées.
 - **Siestes** : durée et qualité du sommeil.
 - **Changements de couches** : fréquence et notes spécifiques.
- Gérer les données médicales :
 - Suivi des mesures (poids, taille).
 - Calcul automatique de l'âge

Objectifs Non Fonctionnels :

- Assurer la sécurité des données :
 - Hashage des mots de passe.
 - Protection contre les attaques et SQL Injection.
- Développer une architecture claire et modulaire (MVC) pour assurer la maintenabilité.
- Garantir la compatibilité avec les appareils mobiles (responsive design).
- Proposer une solution extensible pour ajouter de nouvelles fonctionnalités à l'avenir.

3. Contraintes et livrables attendus

Contraintes Techniques :

- Développement back-end en **PHP** sans framework, en respectant le modèle MVC.
- Base de données relationnelle MySQL, avec gestion efficace des relations entre les tables.
- Utilisation de **HTML, CSS (via Tailwind CSS)**, pour le front-end.
- Environnement de développement local basé sur **MAMP**.

Contraintes de Sécurité :

- Les données des utilisateurs doivent être protégées contre les accès non autorisés.
- Les formulaires doivent inclure des mécanismes de validation pour éviter les failles de sécurité.

Livrables Attendus :

- Une application fonctionnelle déployée sur un environnement local ou distant.
- Une documentation technique détaillant :
 - La structure du projet.
 - La configuration de la base de données.
 - Les étapes de déploiement.
- Une démonstration des principales fonctionnalités.

4. Environnement humain et technique

Environnement Humain :

- **Développeur Web :**
 - Responsable de la conception, du développement et de l'intégration des fonctionnalités.
 - Garant de la qualité du code et de la sécurité de l'application.
- **Utilisateur Final :**

- Parents ou employés de crèche, qui utilisent l'application pour suivre leurs activités.

Environnement Technique :

- **Outils de Développement :**
 - **PhpStorm** : IDE utilisé pour coder efficacement en PHP.
 - **MAMP** : Serveur local pour tester et exécuter l'application.
 - **Git** : Système de gestion de version pour le suivi des modifications.
- **Technologies Utilisées :**
 - **Back-end** : PHP avec une architecture MVC.
 - **Base de données** : MySQL pour le stockage relationnel.
 - **Front-end** : HTML, CSS (avec Tailwind CSS).
 - **Bibliothèques** : PDO pour les requêtes SQL sécurisées.
- **Hébergement (pour la production) :**
 - Serveur O2Switch.
- **Environnement de Test :**
 - Tests manuels sur différents navigateurs (Chrome, Firefox, Safari).

III. Réalisations Côté Front-End

1. Maquettes de l'Application

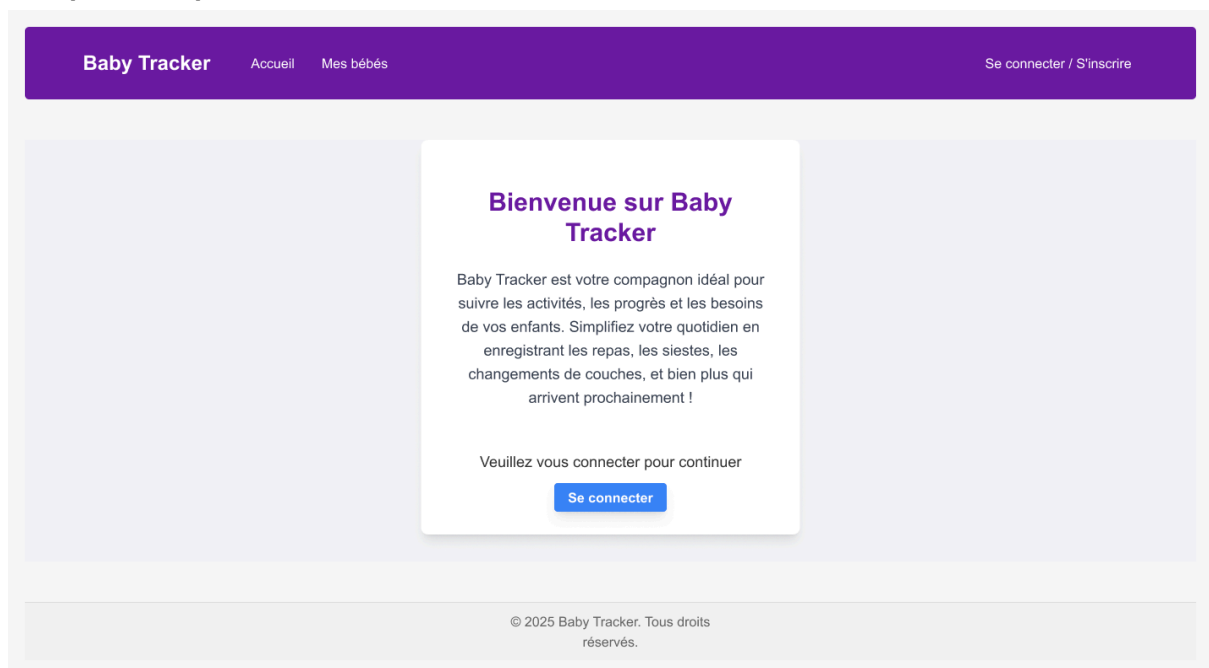
Présentation :

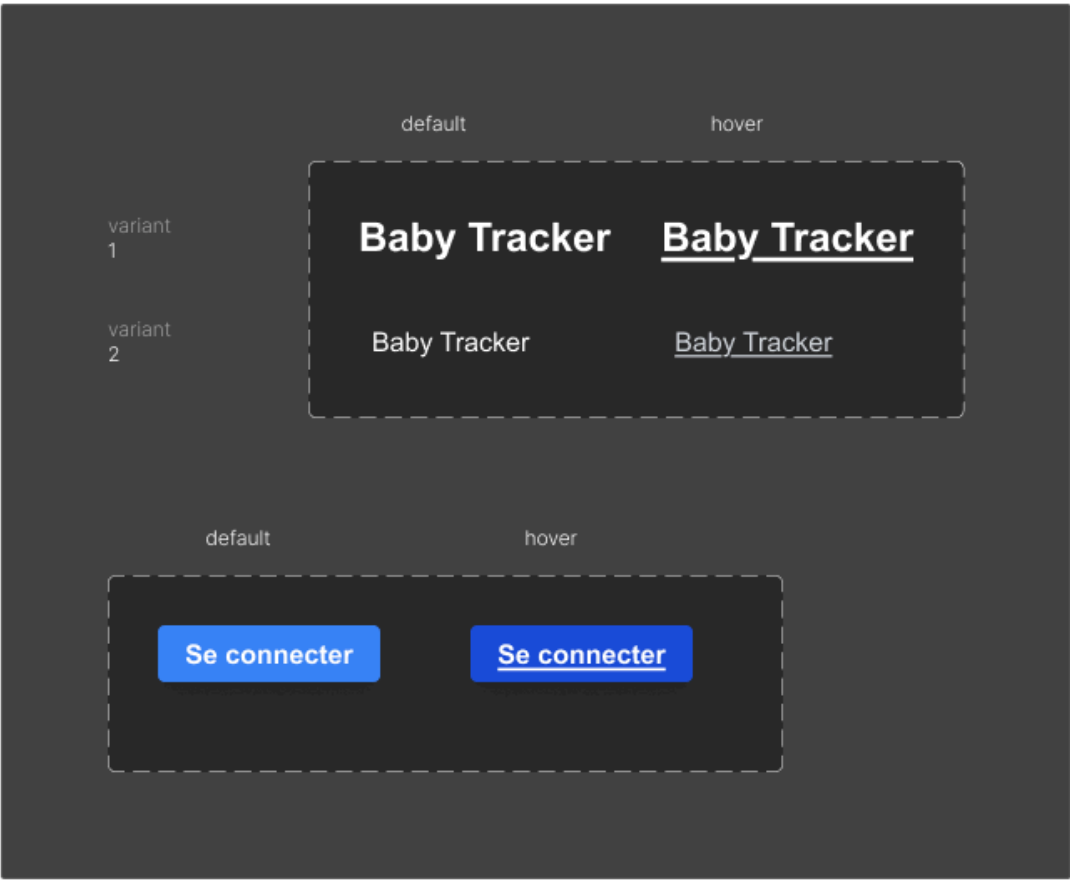
Avant le développement, des maquettes ont été réalisées pour définir l'ergonomie et le design des interfaces utilisateur. Ces maquettes permettent de visualiser la page d'accueil pour avoir une charte graphique.

Outils Utilisés :

- **Figma** : Pour créer les maquettes interactives.
- **Ipad** : Esquisses initiales à la main.

Exemple de Maquettes :



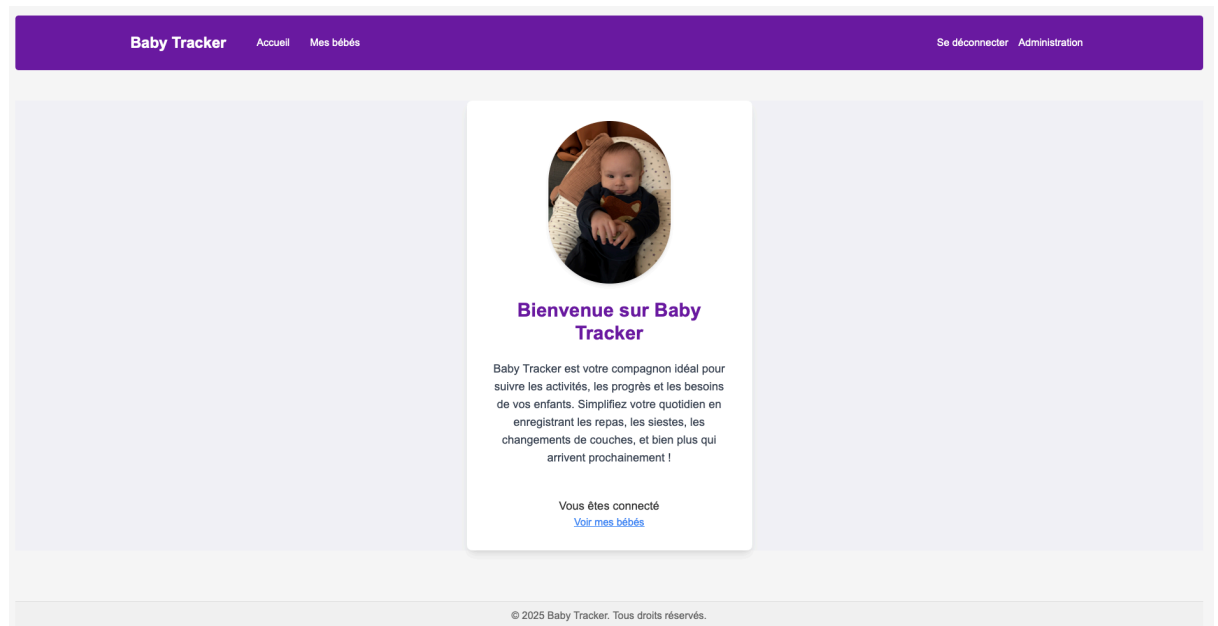


2. Captures d'Écran des Interfaces Utilisateur

Pages Principales :

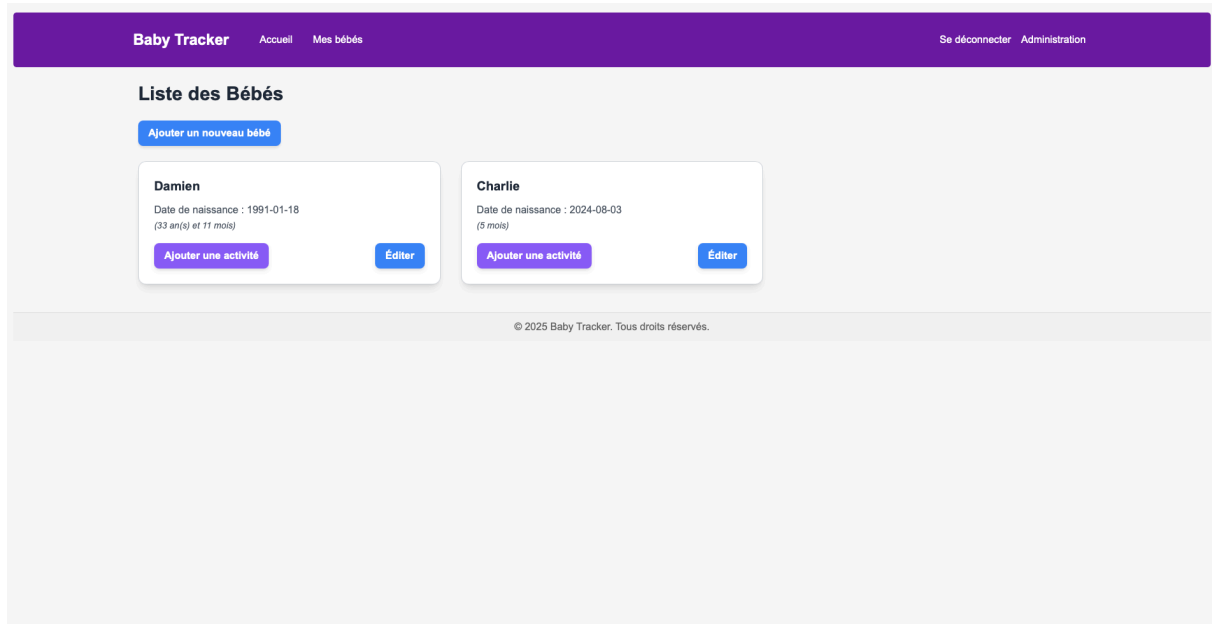
1. Page d'accueil :

- Un design accueillant avec une image, un message de bienvenue, et un bouton pour se connecter.



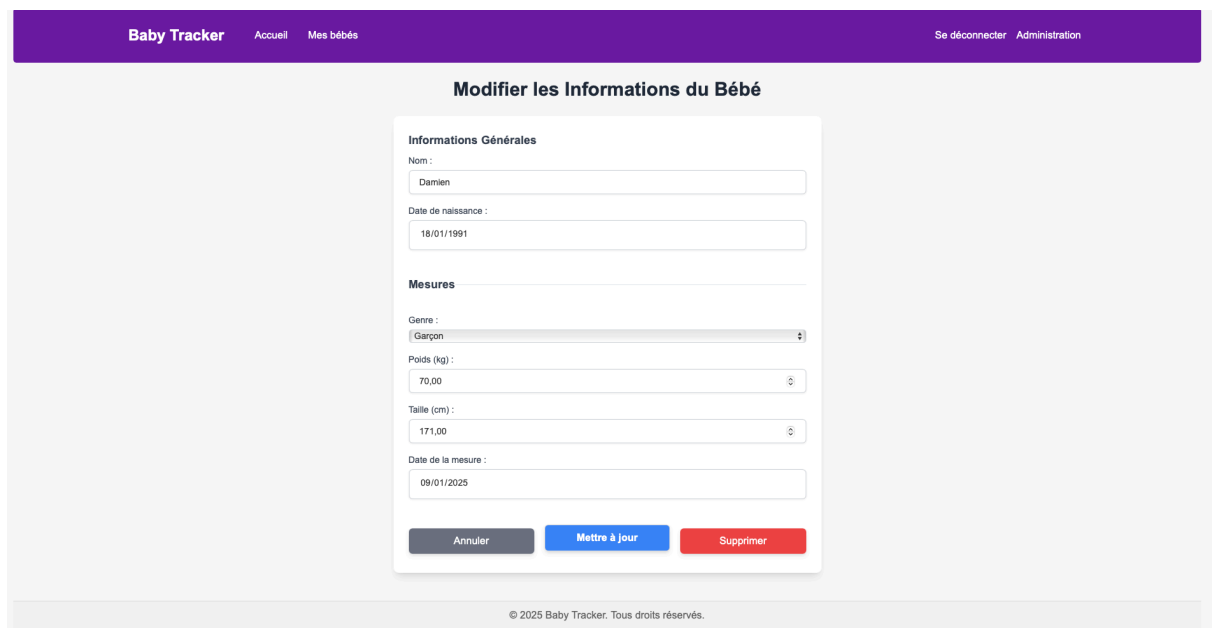
2. Page de gestion des bébés :

- Présente les bébés sous forme de cartes avec leurs informations (nom, date de naissance, âge).



3. Formulaires :

- Exemple : Formulaire d'ajout/modification d'un bébé avec des champs pré-remplis.



4. Page d'authentification :

- Interface avec les champs email/mot de passe et un bouton de connexion.

The screenshot displays the login interface for 'Baby Tracker'. At the top, a purple navigation bar contains the site name 'Baby Tracker' on the left, and links for 'Accueil' and 'Mes bébés' in the center. On the right side of the bar is a link that says 'Se connecter / S'inscrire'. Below the navigation bar, the main content area has a light gray background. In the center of this area is a white login card with a purple title 'Connexion'. The card contains two input fields: 'Email' and 'Mot de passe'. Below these fields is a prominent blue button labeled 'Se connecter'. Underneath the button are two smaller, blue links: 'Mot de passe oublié ?' and 'Créer un compte'. At the bottom of the page, a thin gray footer bar contains the copyright notice '© 2025 Baby Tracker. Tous droits réservés.'

3. Extraits de Code des Interfaces Statiques

Les interfaces statiques sont conçues en **HTML** et **CSS (avec Tailwind CSS)** pour garantir une présentation cohérente et un design responsive.

Exemple : Carte d'un bébé (HTML et Tailwind CSS)

```
<!-- Card -->
<div class="p-6 bg-white border border-gray-300 rounded-xl
shadow-lg hover:shadow-2xl transform transition-all
duration-300">
  <!-- Nom du bébé -->
  <h2 class="text-xl font-bold text-gray-800 mb-3"><?=
htmlspecialchars($baby['name']) ?></h2>

  <!-- Date de naissance et âge du bébé -->
  <p class="text-gray-600">
    Date de naissance : <?=
htmlspecialchars($baby['birth_date']) ?>
    <br>
    <span class="italic text-sm"><?=
Baby::calculateAge($baby['birth_date']) ?></span>
  </p>

  <!-- Boutons -->
  <div class="mt-4 flex justify-between">
    <a href="index.php?ctrl=Activity&action=create&baby_id=<?=
($baby['id']) ?>"
      class="px-4 py-2 bg-violet-500 text-white font-semibold
rounded-lg shadow-md hover:bg-violet-600 focus:outline-none
focus:ring-2 focus:ring-green-400 focus:ring-opacity-75">
      Ajouter une activité
    </a>
    <a href="index.php?ctrl=Baby&action=edit&id=<?=
($baby['id']) ?>"
      class="px-4 py-2 bg-blue-500 text-white font-semibold
rounded-lg shadow-md hover:bg-blue-600 focus:outline-none
focus:ring-2 focus:ring-blue-400 focus:ring-opacity-75">
      Éditer
    </a>
  </div>
</div>
```

5. Choix Techniques et Sécurité

Choix Techniques :

1. **Utilisation de Tailwind CSS :**
 - Pour un design rapide et responsive.

2. **Responsive Design :**
 - Les grilles CSS (via grid-cols-*) permettent une adaptation automatique sur les écrans mobiles et desktops.

3. **Séparation des Composants :**
 - Les parties récurrentes comme la navigation sont isolées dans des fichiers inclus (navbar.php).

Sécurité :

1. **Échappement des Données :**
 - Utilisation de htmlspecialchars() pour éviter les failles XSS (Cross-Site Scripting).

2. **Validation des Entrées :**
 - Toutes les données saisies dans les formulaires sont validées côté serveur avant d'être utilisées.

3. **Fichiers Statiques Sécurisés :**
 - Les ressources comme les images et les scripts sont stockées dans le dossier *public* pour éviter les accès non autorisés au code source.

IV. Réalisation Côté Back-End

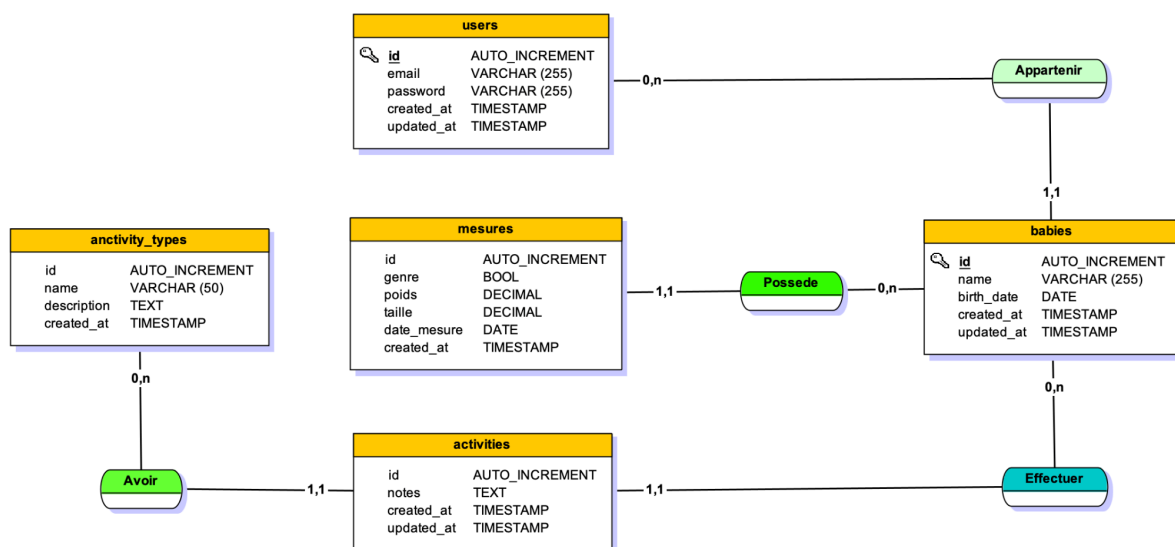
1. Présentation de la Base de Données

La base de données a été conçue en **MySQL** en suivant le modèle relationnel pour répondre aux besoins de l'application **Baby Tracker**. Elle comprend les tables suivantes :

1. users : Gère les utilisateurs inscrits (parents).
2. babies : Stocke les informations des bébés.
3. activities : Enregistre les activités des bébés.
4. activity_types : Définit les différents types d'activités (repas, sommeil, etc.).
5. mesures : Permet de suivre les données médicales des bébés (poids, taille).

Modèle Conceptuel de Données (MCD) :

- Le MCD montre les relations entre les entités (par exemple, un utilisateur peut avoir plusieurs bébés, chaque bébé peut avoir plusieurs activités).



2. Extraits de Code des Composants d'Accès aux Données

Les composants d'accès aux données sont implémentés dans le modèle (Model) en utilisant **PDO** pour garantir la sécurité des requêtes SQL.

Exemple 1 : Récupérer les informations d'un bébé par son ID

BabyController :

```
public function index()
{
    require_once 'config/Database.php';
    $pdo = Database::getConnection();
    $this->babyModel = new Baby($pdo);
    $babies = $this->babyModel->findAllByUser($_SESSION['user_id']);
    include "views/baby/index.php";
}
```

BabyModels :

```
public function findById($id)
{
    $query = $this->db->prepare('SELECT * FROM babies WHERE id = :id');
    $query->bindParam(':id', $id, PDO::PARAM_STR);
    $query->execute();
    return $query->fetch(PDO::FETCH_ASSOC);
}
```


Exemple 2 : Ajouter une activité pour un bébé

ActivityController :

```
public function create()
{
    // Récupérer les types d'activités
    $activityTypes = $this->activityType->findAll();

    // Inclure la vue en passant les données
    include 'views/activity/create.php';
}
```

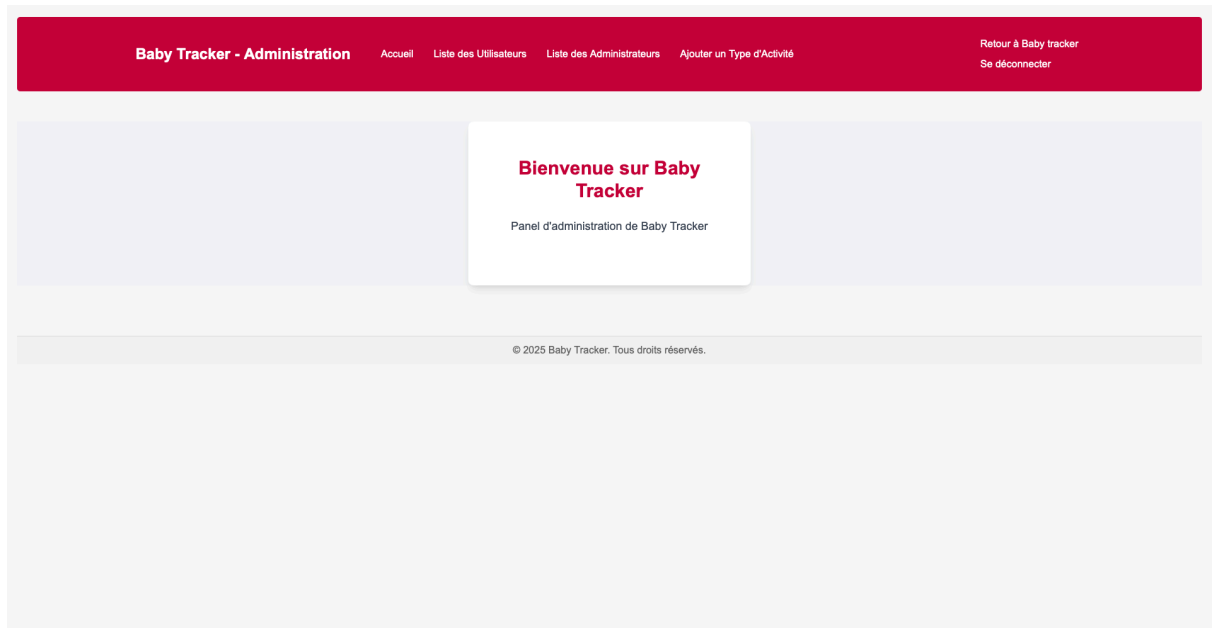
ActivityModels :

```
public function create($babyId, $type, $date, $notes)
{
    $query = $this->db->prepare('INSERT INTO activities (baby_id,
activity_type_id, created_at, notes) VALUES (:baby_id, :type, :date,
:notes)');
    $query->bindParam(':baby_id', $babyId, PDO::PARAM_INT);
    $query->bindParam(':type', $type, PDO::PARAM_STR);
    $query->bindParam(':date', $date, PDO::PARAM_STR);
    $query->bindParam(':notes', $notes, PDO::PARAM_STR);
    $query->execute();
}
```

3. Éléments de Sécurité de l'Application

Création d'un espace Administration :

- Création de rôles pour la gestion des utilisateurs (users, admin)
- Création d'un Dashboard administrateur qui permet d'afficher la liste des utilisateurs, des administrateurs, et d'ajouter des activités.



Authentification et Protection des Données :

1. Hashage des mots de passe :

- Les mots de passe sont hashés avec password_hash() lors de l'inscription.
- Lors de la connexion, ils sont vérifiés avec password_verify().
- Création d'une méthode qui vérifie si l'utilisateur a le bon rôle :

```
protected function requireSessionAdmin()
{
    if (!isset($_SESSION['user_role']) || $_SESSION['user_role'] !== 'admin')
    {
        $_SESSION['error'] = "Accès non autorisé. Veuillez vous connecter en tant qu'administrateur.";
        header('Location: index.php?ctrl=Auth&action=showLoginForm');
        exit;
    }
}
```

Protection Contre les Attaques :

1. **Prévention des Injections SQL :**
 - Toutes les requêtes SQL utilisent des paramètres préparés via **PDO**.
2. **Échappement des Données :**
 - Les données affichées dans les vues sont échappées avec `htmlspecialchars()`.

4. Jeu d'Essai et Analyse des Résultats

Jeu d'Essai Utilisé :

1. **Création d'un utilisateur :**
 - Email : test@mail.com
 - Mot de passe : test123
2. **Ajout d'un bébé :**
 - Nom : Charlie
 - Date de naissance : 2023-08-01
3. **Enregistrement d'une activité :**
 - Type : Repas
 - Date : 2023-02-01 12:30:00
 - Notes : 210 ml de lait
4. **Ajout de mesures :**
 - Genre : Garçon
 - Poids : 6.5 kg
 - Taille : 69 cm
 - Date : 2025-01-05

Résultats Obtenus :

1. Les données ont été correctement enregistrées dans les tables correspondantes (users, babies, activities, mesures).

2. Les fonctionnalités d'affichage ont correctement récupéré les données pour les afficher dans les vues (listes et formulaires).
3. Les messages de validation (succès/erreur) se sont affichés comme attendu.

5. Choix Techniques et Sécurité

Choix Techniques :

1. **PHP avec PDO :**
 - Choisi pour sa simplicité et sa sécurité.
 - Paramètres préparés pour protéger contre les injections SQL.
2. **Architecture MVC :**
 - Facilite la séparation des responsabilités entre les données, la logique métier et l'interface utilisateur.

Sécurité :

1. **Sécurité des Données :**
 - Hashage des mots de passe et validation des entrées.
2. **Session Management :**
 - Les sessions utilisateur sont validées pour chaque requête sensible.

V. RGPD

Pour suivre les recommandations RGPD, j'ai ajouter dans ma base de donnée, dans la table *User* une colonne *lastLogin*

Grâce à cette requête :

```
$query = $this->db->prepare('UPDATE users SET lastLogin = NOW() WHERE id = :id');  
$query->bindParam(':id', $user['id'], PDO::PARAM_INT);  
$query->execute();
```

Lorsqu'un utilisateur se connecte il met à jour la date de *lastLogin* à la date du jour.
Et dans la vue de ma liste des utilisateurs dans mon Dashboard Admin, j'ai l'affichage de la date de dernière connexion des utilisateurs :

```
<p class="text-gray-500 mb-2">  
  <strong>Nom :</strong> <?= htmlspecialchars($user['lastname']) ?? 'Non  
disponible'> ?>  
</p>  
<p class="text-gray-500 mb-2">  
  <strong>Prénom :</strong> <?= htmlspecialchars($user['firstname']) ??  
'Non disponible'> ?>  
</p>  
<p class="text-gray-500 mb-2">  
  <strong>Téléphone :</strong> <?= htmlspecialchars($user['phone']) ??  
'Non disponible'> ?>  
</p>  
<p class="text-gray-500 mb-2">  
  Dernière connexion : <?= htmlspecialchars($user['lastLogin']) ?? 'Non  
disponible'> ?>  
</p>
```

Ce qui va me permettre de supprimer leurs comptes après un an de non connexion.

VI. Veille effectuée

Pour la veille, j'ai utilisé mon expérience personnelle de nouveau parent, en effet j'ai eu le bonheur d'accueillir mon fils qui est arrivé le 3 août pendant la formation.

J'ai eu le besoin de suivre l'heure et la quantité des biberons, ses temps de sommeil et ses changes.

Pour cela j'ai testé plusieurs applications.

Arrivé à la crèche, j'avais le retour des employés via un cahier (sous format papier) qui récapitule ces informations, et je me suis dit qu'il y avait une possibilité d'améliorer cela en créant un outil numérique qui permettra de gérer ces données tout en permettant au parents d'être au courant des différentes activités de leur enfant en temps et en heure.

Conclusion

Malgré les péripéties et retournement de situation durant mon stage, j'ai pris beaucoup de plaisir à faire ce projet.

J'ai encore beaucoup d'idée pour faire évoluer l'application :

- Possibilité de partager les bébés :

Le but est de proposer l'application aux crèches, sachant qu'elles sont plusieurs à s'occuper de nombreux enfants il faut qu'elles puissent accéder au même enfant et que l'on puisse suivre qui à fait qu'elle action.

- Possibilité pour les parents de suivre en direct les actions (biberons, sommeil etc...) via la mise en ligne de l'application.

- Création de statistiques via les données récoltés :

Par exemple, affichage des quantités de lait par jour, du temps de sommeil, graphique de temps de sommeil sur le mois etc...

-