

LO07 – TechnOLOGIE DU WEB

Projet : Conception d’un site Web de mise en relation de nounous avec des parents

Adrien Lebret – Jérémie Marotte – ISI02

Automne

Automne

Table des matières

[LO07 – TechnOLOGIE DU WEB 1](file:///C:\Users\adrie\Documents\UTT\ISI02\LO07\Projet\Rapports\LO07%20-%20Rapport%20-%20Adrien%20Lebret%20-%20Jérémie%20Marotte.docx#_Toc517686080)

[Introduction 4](#_Toc517686081)

[Résumé du sujet du projet 4](#_Toc517686082)

[Problématiques à résoudre 4](#_Toc517686083)

[I) Analyse des contraintes et des besoins du cahier des charges 5](#_Toc517686084)

[II) Modélisation des fonctionnalités demandées 6](#_Toc517686085)

[Fonctionnalité 1 : Administrateur du site 6](#_Toc517686086)

[Fonctionnalité 2 : Gestion des nounous 6](#_Toc517686087)

[Fonctionnalité 3 : Services pour les parents 6](#_Toc517686088)

[Diagramme de fonctionnement général 7](#_Toc517686089)

[III) Etapes de conception du projet 9](#_Toc517686090)

[Outils utilisés 9](#_Toc517686091)

[Phases de conception et de création 10](#_Toc517686092)

[IV) Etape de conception de la base de données 14](#_Toc517686093)

[V) Problèmes rencontrés 19](#_Toc517686094)

[Notre avancée par rapport aux objectifs du projet 19](#_Toc517686095)

[Difficultés rencontrées lors de la production 20](#_Toc517686096)

[Conclusion 21](#_Toc517686097)

[Annexe : Fichier de création de la base de données 22](#_Toc517686098)

Table des illustrations

[Figure 1 : Tableau des contraintes et des besoins 5](#_Toc517689564)

[Figure 2 : Diagramme de fonctionnement général (administrateur) 7](#_Toc517689565)

[Figure 3 : Diagramme de fonctionnement général (nounou) 7](#_Toc517689566)

[Figure 4 : Diagramme de fonctionnement général (parent) 8](#_Toc517689567)

[Figure 5 : Looking For Nounou - Page d'accueil 10](#_Toc517689568)

[Figure 6 : Looking For Nounou - Inscription Nounou 11](file:///E:\UTT\LO07%20-%20Rapport%20-%20Adrien%20Lebret%20-%20Jérémie%20Marotte.docx#_Toc517689569)

[Figure 7 : Looking For Nounou - Inscription Parent 11](file:///E:\UTT\LO07%20-%20Rapport%20-%20Adrien%20Lebret%20-%20Jérémie%20Marotte.docx#_Toc517689570)

[Figure 8 : Looking For Nounou - Inscription Enfant(s) 11](file:///E:\UTT\LO07%20-%20Rapport%20-%20Adrien%20Lebret%20-%20Jérémie%20Marotte.docx#_Toc517689571)

[Figure 9 : Looking For Nounou - Partie Administrateur 12](#_Toc517689572)

[Figure 10 : Looking for Nounou - Profil Parent 12](#_Toc517689573)

[Figure 11 : Looking For Nounou - Recherche d'une nounou 13](#_Toc517689574)

[Figure 12 : Looking For Nounou - Réservation d'une nounou 13](#_Toc517689575)

[Figure 13 : Diagramme Entité-Association 14](#_Toc517689576)

[Figure 14 : Base de données - Table nounou 15](#_Toc517689577)

[Figure 15 : Base de données - Table ‘parent' 15](#_Toc517689578)

[Figure 16 : Base de données - Table 'admin' 15](#_Toc517689579)

[Figure 17 : Base de données - Table 'enfant' 16](#_Toc517689580)

[Figure 18 : Base de données - Table 'disponibilité' 16](#_Toc517689581)

[Figure 19 : Base de données - Table 'ville' 16](#_Toc517689582)

[Figure 20 : Base de données - Table 'langue' 17](#_Toc517689583)

[Figure 21 : Base de données - Table 'parle' 17](#_Toc517689584)

[Figure 22 : Base de données - Table 'reserve' 18](#_Toc517689585)

[Figure 23 : Base de données - Table 'lie' 18](#_Toc517689586)

[Figure 24 : Base de données - Table 'garde' 18](#_Toc517689587)

[Figure 25 : Table - Avancée du projet 19](#_Toc517689588)

# Introduction

## Résumé du sujet du projet

Dans le cadre de l’UE LO07 (Technologie du Web) qui vise à réaliser une production illustrant les langages assimilés dans le cours magistral tels que PHP, HTML, CSS et Javascript, notre binôme (Adrien Lebret et Jérémie Marotte) doit créer un site web permettant la mise en relation de nounous avec des parents.

Les parents choisissent dès lors les nounous en fonction de plusieurs critères comme notamment la disponibilité, les compétences ou les évaluations réalisées par les autres parents.

Notre site web devra différencier trois types d’utilisateur, à savoir les nounous et les parents ainsi que l’administrateur du site.

L’objectif est ainsi de nous faire suivre et appliquer nous-même toutes les étapes de la production d’une conception web.

## Problématiques à résoudre

A travers ce projet, nous souhaitons faciliter à l’administrateur du site la gestion des nounous et les services pour les parents. Cela passe notamment par l’inscription des utilisateurs sur le site et à la gestion des différents types de gardes que peut avoir une nounou (garde d’enfant ponctuelle, régulière, en langues étrangères).

# Analyse des contraintes et des besoins du cahier des charges

Tout d’abord, après lecture du cahier des charges, nous pouvons résumer et examiner les différentes contraintes et les besoins du site web.

|  |  |
| --- | --- |
| Nom de la contrainte / du besoin | Description de la contrainte / du besoin |
| Contraintes techniques obligatoires | * Un serveur Apache * Des pages HTML * Des feuilles de styles CSS pour toutes les pages de votre site * Des scripts PHP * Des instructions JavaScript * Une base de données relationnel |
| Contraintes optionnelles | * Bootstrap * JQuery * Framework |
| Différenciation des utilisateurs | Il existe trois types d’utilisateurs :   * Les nounous * Les parents * L’administrateur du site |
| Adresse | Il faut utiliser la ville comme adresse pour les nounous et les parents |
| Recrutement des nounous | L’administrateur doit pouvoir accepter ou refuser les candidatures. |
| Bloquer l’activité des nounous | L’administrateur peut être amené à bloquer temporairement l’activité d’une nounou. |
| Parents | * Sont identifiés par leur nom, ville, email, liste des enfants avec leur prénom, date de naissance, restrictions alimentaires. * Peuvent déclarer des élément importants pour l’organisation |
| Nounous | * Sont identifiés par leur nom, prénom, ville, email, portable, liste des langues parlées, photo, âge, expérience et les évaluation des parents. * Peuvent déclarer leurs disponibilités et voir leur planning |

Figure 1 : Tableau des contraintes et des besoins

# Modélisation des fonctionnalités demandées

Le projet se décompose dès lors en différentes fonctionnalités qui sont réparties suivant l’utilisateur en question.

## Fonctionnalité 1 : Administrateur du site

Cette fonctionnalité décrit le rôle et les objectifs de l’administrateur du site. Nous devons en effet lui confectionner un compte particulier qui lui permettra d’obtenir des informations stratégiques pour la gestion du site Web.

De plus, lorsqu’une nounou aura rempli le formulaire d’inscription, elle ne sera pas inscrite tant que l’administrateur n’a pas accepté sa candidature. De même, comme l’administrateur a accès à l’ensemble des nounous inscrites (celles refusées étant supprimer de la base de données).

Pour se faire, nous devons mettre en place une page permettant l’affichage des informations tirées de la base de données, mais aussi les traiter (exemple : affichage des utilisateurs actifs).

## Fonctionnalité 2 : Gestion des nounous

Dans cette deuxième fonctionnalité, nous devons créer un formulaire d’inscription pour les nounous qui récoltera toutes les informations nécessaires pour un bon suivi de l’administrateur.

La candidature validée par l’administrateur, le site web permettra à la nounou de choisir ses différentes disponibilités. Ces dernières peuvent être des disponibilités simples, c’est-à-dire pour une période, un créneau précis, ou récurrentes entre deux dates.

Pour chaque disponibilité, elle précise ses horaires de disponibilité. Elle peut cumuler les disponibilités afin de s’approcher au plus près de son emploi du temps. Une fois ses disponibilités précisées, une nounou peut accéder à la liste des disponibilités qu’elle a saisie. De la même manière, elle peut voir sur quels créneaux il lui a été demandé de faire une garde.

## Fonctionnalité 3 : Services pour les parents

Enfin, pour la troisième fonctionnalité, à l’image des nounous, nous devons mettre en place un formulaire d’inscription pour les parents pour qu’ils puissent ainsi chercher une nounou pour une garde ponctuelle, régulièrement et éventuellement en langue étrangère.

Là où les nounous peuvent proposer des disponibilités, les parents peuvent demander des gardes. Pour ce faire, ils remplissent un court formulaire afin de connaître le type de garde, la date et les enfants concernés. Une fois cela fait, la liste des nounous correspondantes leur est affichée pour donner suite à une requête sur la base de données.

Quand la garde est terminée, il est demandé aux parents de saisir le nombre d’heures effectivement réalisé ainsi qu’une courte évaluation de la nounou. Là encore, cette fonctionnalité passe par l’usage d’un formulaire.

## Diagramme de fonctionnement général

Ainsi, nous obtenons les diagrammes de fonctionnement général suivant (A noter que notre fichier database.php est associé à l’ensemble de nos fichiers php) :

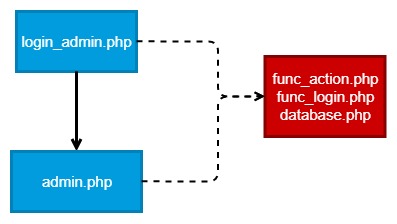


Figure 2 : Diagramme de fonctionnement général (administrateur)

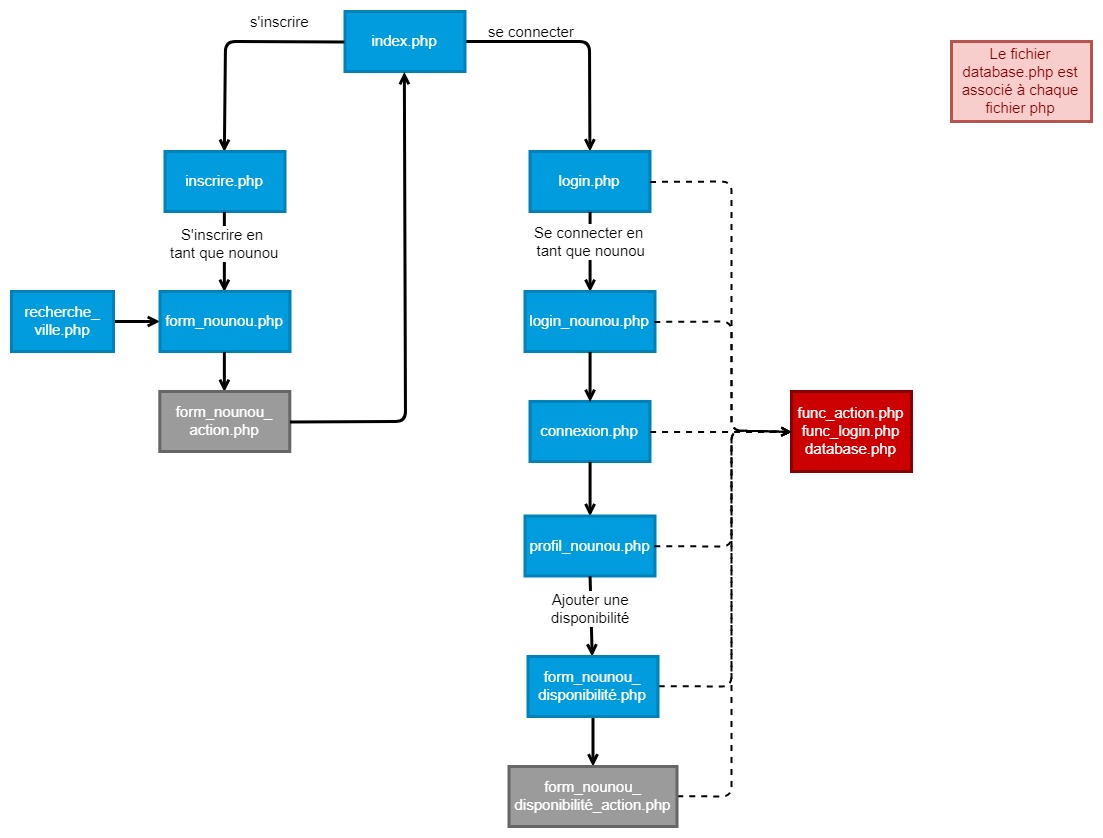


Figure 3 : Diagramme de fonctionnement général (nounou)

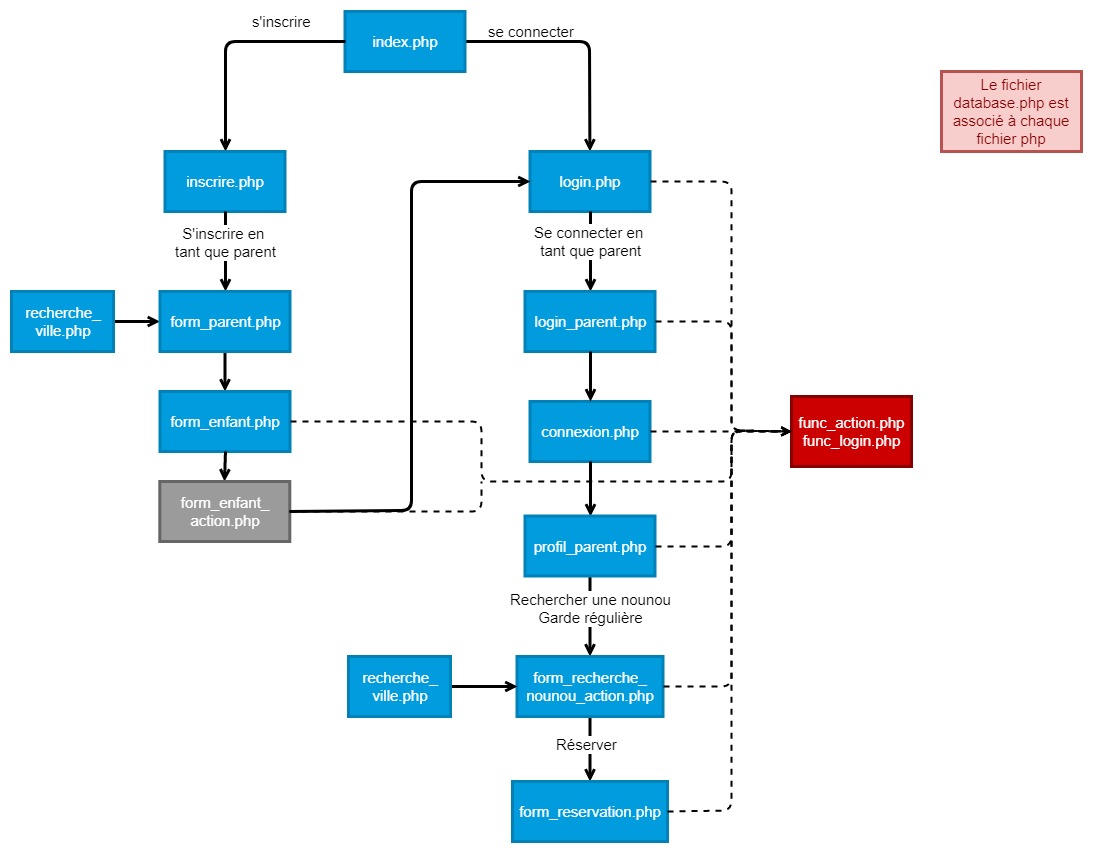


Figure 4 : Diagramme de fonctionnement général (parent)

# Etapes de conception du projet

Avant toute chose, nous avons décidé de diviser notre travail en plusieurs étapes de conception et de modélisation. À partir du moment où les étapes de conception ont été réalisés nous avons pris la décision de diviser notre travail d’implémentation en plusieurs étapes :

1. Formulaires, CSS, HTML.
2. Lien avec la base de données, PHP, dynamisme.
3. Détection des erreurs, JQuery, JavaScript.
4. Commentaires et lisibilité du code, suppression de fichiers inutiles.

## Outils utilisés

Pour notre projet, nous avons utilisé les différents outils cités ci-dessous afin de nous aider à créer et concevoir notre site web :

**Bootstrap** : La possibilité d’utiliser un Framework s’ouvrait à nous. Nous avons donc décidé de travailler avec Bootstrap, le Framework le plus connu pour le développement de sites web.

Nous avons choisi de travailler avec un Framework pour les raisons suivantes :

- Facilité d’utilisation : Bootstrap est un Framework complet, mais aussi facile à prendre en main. Ce Framework dispose aussi d’une documentation accessible facilement

- Vitesse de codage : Bootstrap nous a permis de coder avec des objets CSS, HTML, JavaScript déjà prédéfinis ce qui nous a permis de travailler plus rapidement et plus proprement sur l’ensemble du projet. Ce Framework nous a permis de réaliser un frontend bien structuré.

- Expérience gagné : Travailler avec un Framework est une qualité requise dans le domaine du développement web. L’expérience acquise avec ce Framework sera très utile pour la suite de notre carrière.

**WAMP (Serveur Apache / SQL / PHP)**: Travailler en local a été la meilleure solution pour la réalisation de ce projet. Pour cela nous avons utilisé WAMP (Windows) pour créer nos serveurs locaux nous permettant d’exécuter notre code PHP. Cette solution s’avérait particulièrement pratique car nous avons libre accès à toutes les fonctionnalités du “serveur”. WAMP/MAMP permettent aussi de gérer un serveur local Apache ainsi que gérer nos bases des données grâce à phpMyAdmin.

**PhpMyAdmin**: Pour gérer et stocker notre base de données nous avons travaillé avec phpMyAdmin, outil inclus gratuitement dans les logiciels MAMP/WAMP/LAMP. Ce logiciel nous permettait de gérer la base de données et les tables pour notre projet. Ainsi on vérifie le bon fonctionnement de nos structures.

**HTML / CSS / Javascript / PHP / jQuery**: L’emploi de fichiers HTML et PHP était nécessaire pour mettre en place les pages web de notre projet. De plus, comme nous avons utilisé un Bootstrap et un Template CSS (modifié), nous avons travaillé avec le langage JavaScript, CSS et jQuery.

**GitHub** : Pour favoriser le travail collaboratif, nous allons utiliser le collecticiel GitHub qui permettra à notre groupe de partager des documents à distance tout en assurant une synchronisation des fichiers et des échanges. Le logiciel nous servira également pour observer les différentes avancées du projet grâce à l’historique associé.

## Phases de conception et de création

Dans un premier temps, nous avons décidé de modéliser la base de données qui nous permettra d’assurer la persistance des données des utilisateurs et administrateurs. Les détails concernant la structure de la base de données seront présentés dans la partie « [Etape de conception de la base de données](#_Etape_de_conception)» ci-après.

Dans un second temps, une fois la base de données modélisée nous avons commencé le développement du projet. Nous avons dès lors créé un repository sur GitHub, une fois cela fait nous l’avons intégré à Netbeans.

Avant de passer au développement à proprement parler nous avons consulté la documentation de Bootstrap afin de procéder à son intégration dans le projet. Tout le site possède un header, ainsi qu’un *footer* reprenant des informations sur le site.

La première fonctionnalité développée fut la seule accessible à tout le public à savoir **la page d'accueil**. Nous y trouvons un carrousel qui est une fonctionnalité de Bootstrap. Tout au long du projet c’est grâce à Bootstrap que nous avons pu éditer l’aspect de notre site web.

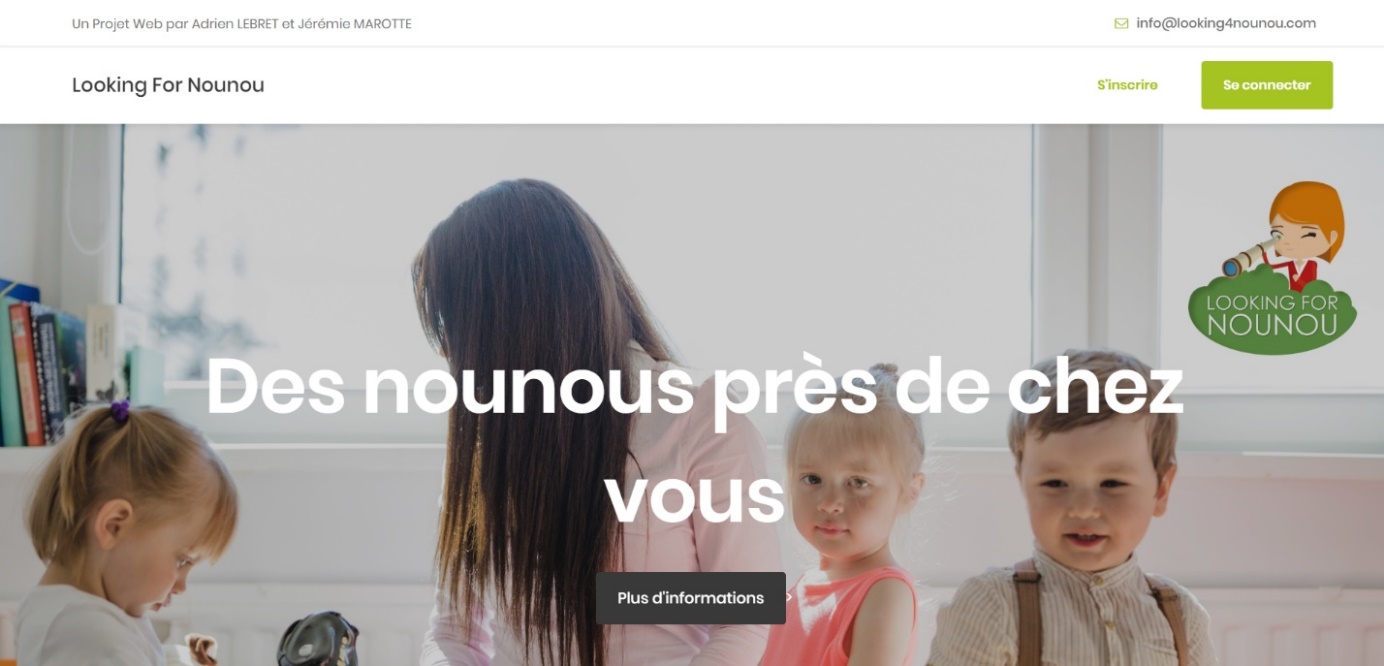


Figure 5 : Looking For Nounou - Page d'accueil

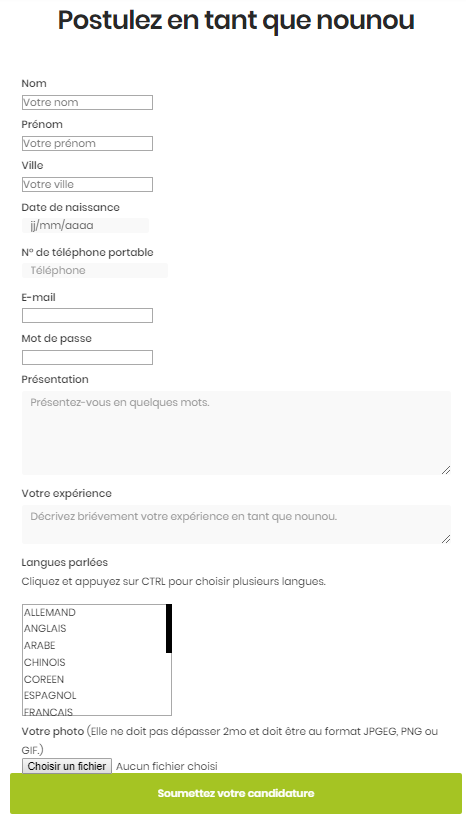


Figure 6 : Looking For Nounou - Inscription Nounou

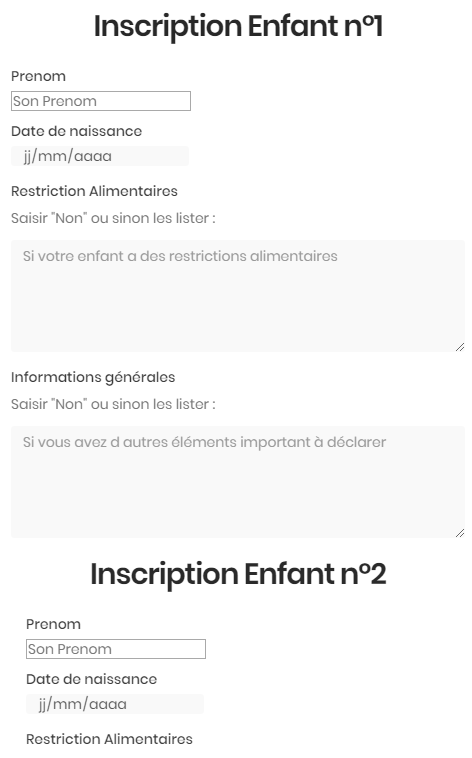
Une fois l’index.php réalisé, nous avons travaillé sur les **formulaires d'inscription** et leurs pages de traitement. Une fonction JavaScript permet de vérifier la validité les informations saisies une fois le bouton submit cliqué, et cela avant le traitement PHP. 8. Dans notre site web, nous offrons 2 possibilités d’inscription : une pour s’inscrire en tant que nounou et une autre pour s’inscrire en tant que parent.

Figure 7 : Looking For Nounou - Inscription Parent

Concernant l’inscription des parents, nous sommes parties de l’idées qu’une famille, en moyenne, ne possède pas plus de 5 enfants. Le nombre d’enfants déterminés, un nouveau formulaire apparaît pour l’inscription des X enfants de la famille en question. A noter que ce formulaire réalise également le traitement des données saisies pour l’inscription du parent.

De ce fait, lorsque le parent a saisi les différentes informations concernant son(ses) enfant(s), un seul bouton d’envoie insérera les informations dans la base de données.

Figure 8 : Looking For Nounou - Inscription Enfant(s)

L’inscription terminée, la (future) nounou est renvoyée sur la page d’accueil car elle doit attendre que l’administrateur accepte son adhésion. Pour se faire, l’administrateur, préalablement connecté, voit apparaître sur son tableau de bord l’inscription d’une nouvelle nounou et fait le choix de l’accepter ou non et ainsi de lui offrir la possibilité de déterminer ses différents créneaux de disponibilités.

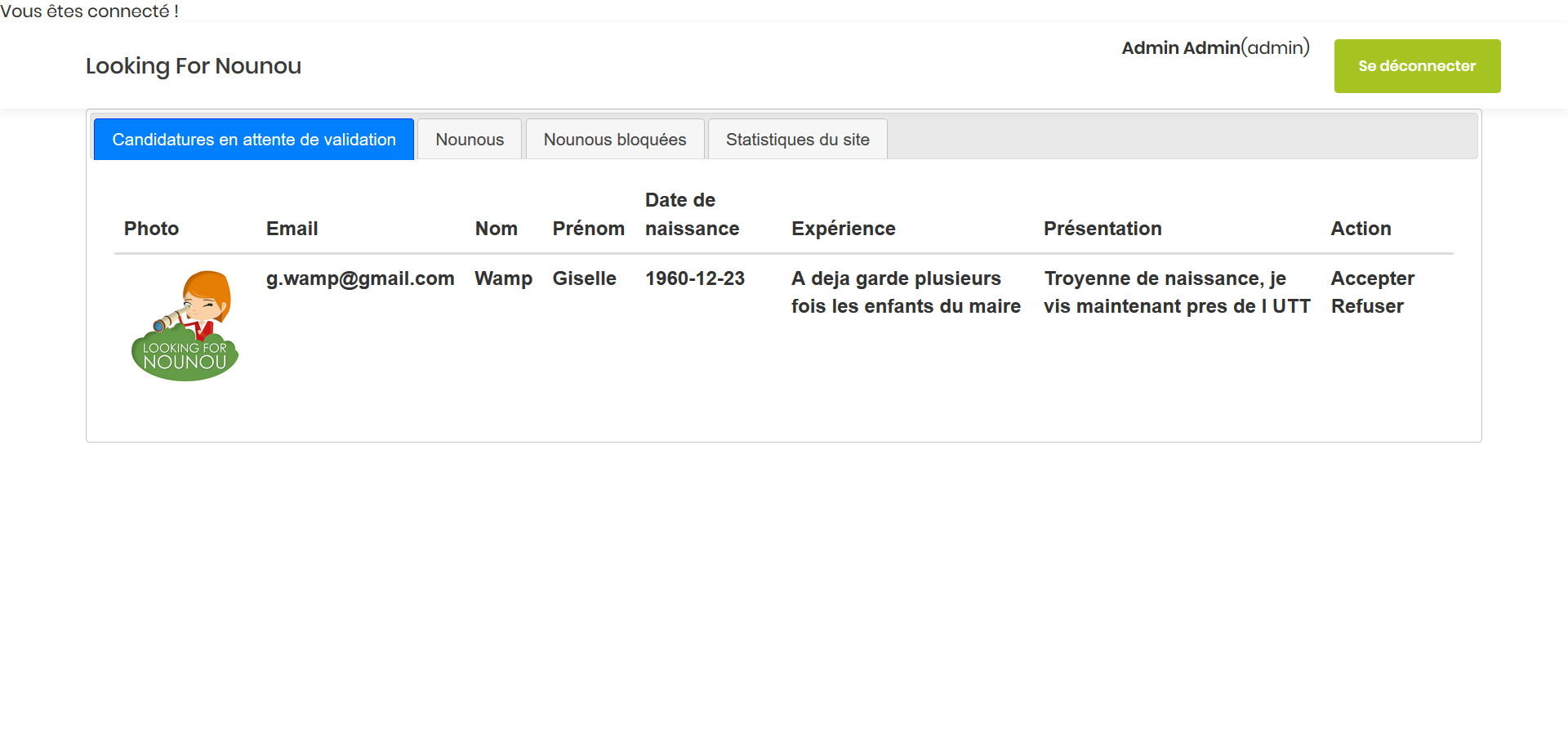


Figure 9 : Looking For Nounou - Partie Administrateur

Du côté des parents, suite à leur inscriptions, le site les envoie vers leur page de profil avec les données suivantes :

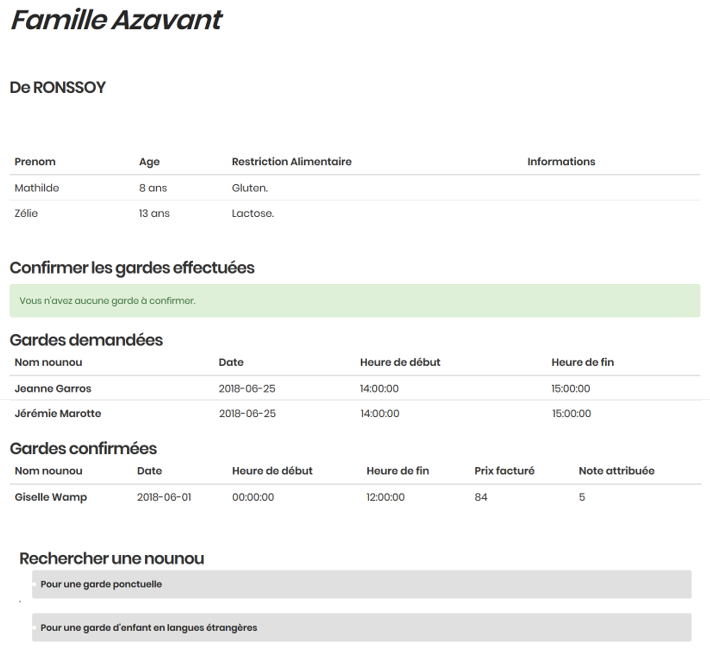


Figure 10 : Looking for Nounou - Profil Parent

Par la suite, nous avons réalisé un fichier PHP concernant la recherche d’une nounou en fonction de la date, heures de début et de fin, recherchées par les parents, mais aussi de la ville.

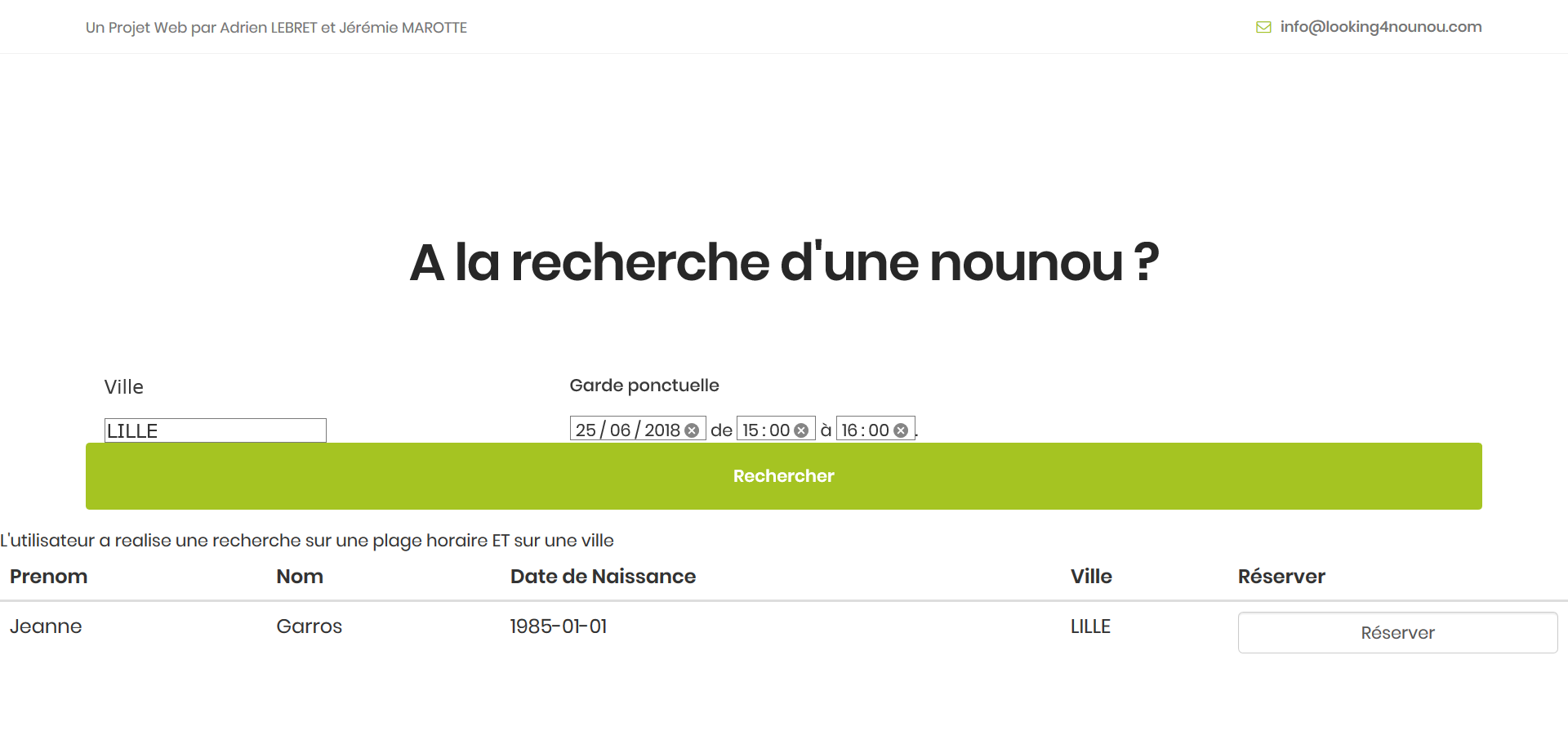


Figure 11 : Looking For Nounou - Recherche d'une nounou

Enfin, nous avons mis en place le système de réservation.



Figure 12 : Looking For Nounou - Réservation d'une nounou

# Etape de conception de la base de données

La conception de la base de données est la première chose que nous avons réalisé dans ce projet. Nous avons essayé de cerner les impératifs et avons construit notre base de données pour répondre aux demandes du sujet.

Pour se faire, nous avons créé une base de données qui s’appelle « nounou ». Dans celle-ci nous disposons de 11 tables qui nous permettent de stocker toutes les informations dont nous avons besoin. Chaque nouvelle information est dès lors stockée dans une de ces tables, et les requêtes SQLnous permettent de venir écrire et lire des données sur ces tables, ainsi nous pouvons faire perdurer tous les renseignements dont nous avons besoin pour faire fonctionner notre site internet.

Ainsi, nous avons modélisé la base de données sur papier. Ensuite nous l’avons transposée numériquement à l’aide de MySQL Workbench. Une fois notre base de données modélisée sur Workbench nous avons pu la transformer en base de données au format SQL et ainsi l’intégrer avec PhpMyAdmin et nous obtenons le diagramme entité-association ci-dessous.

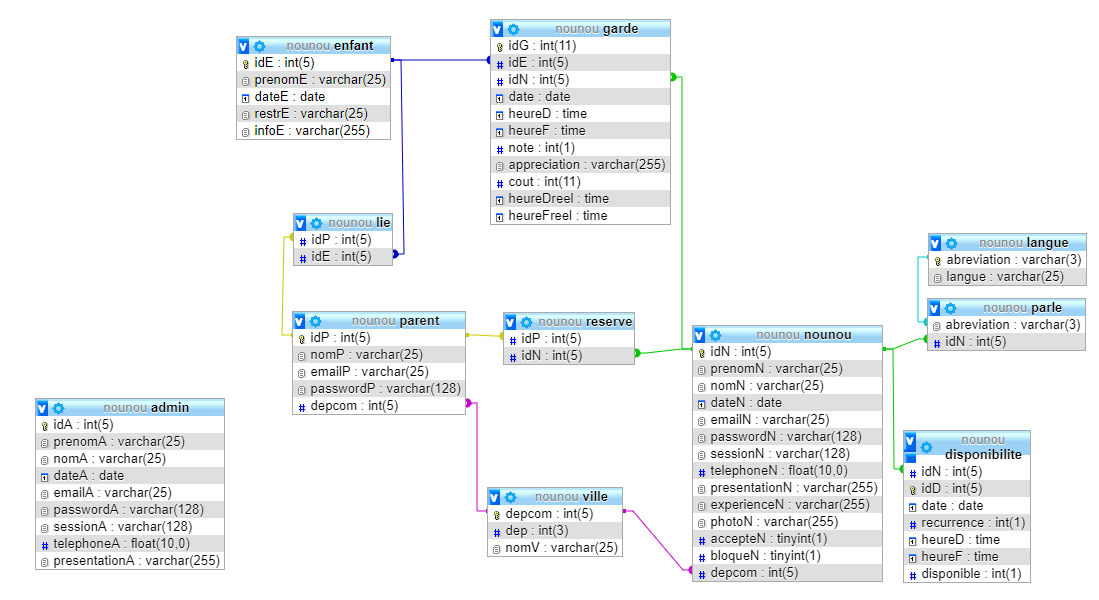


Figure 13 : Diagramme Entité-Association

Concernant les tables en elles-mêmes, nous avons tout d’abord créé les tables correspondant à chaque type d’utilisateur : **Nounou / Parent / Administrateur**. Nous avons voulu séparer les utilisateurs en 3 tables distinctes pour avoir une vision plus globale de la base de données.

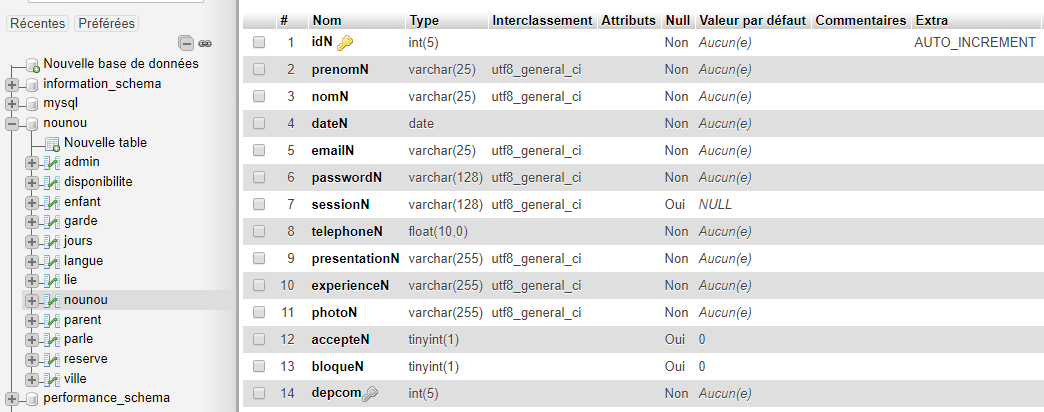


Figure 14 : Base de données - Table nounou

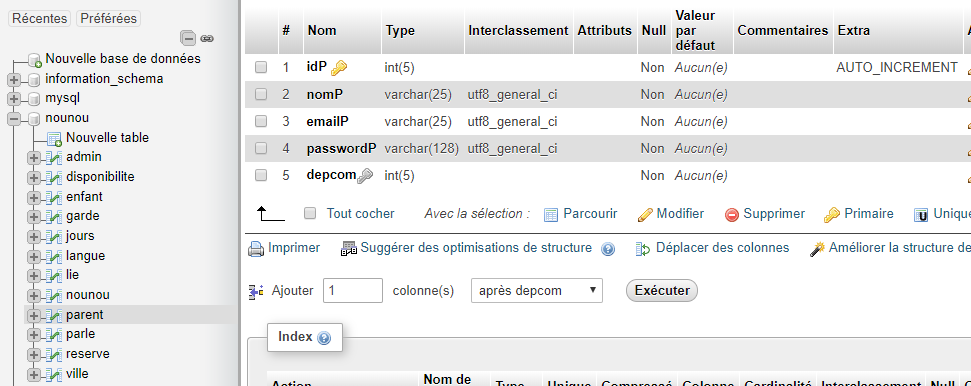


Figure 15 : Base de données - Table ‘parent'

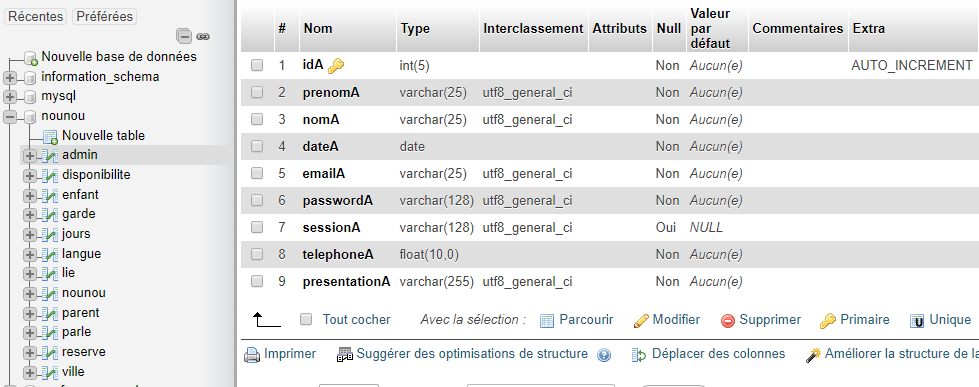


Figure 16 : Base de données - Table 'admin'

Ensuite nous avons mis en place des tables dites « secondaires » tel que la table enfant, disponibilité, ville et langue.

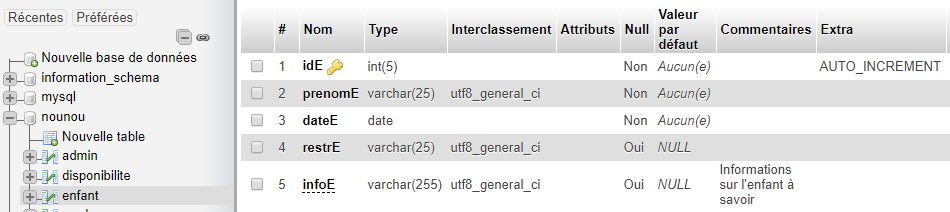


Figure 17 : Base de données - Table 'enfant'

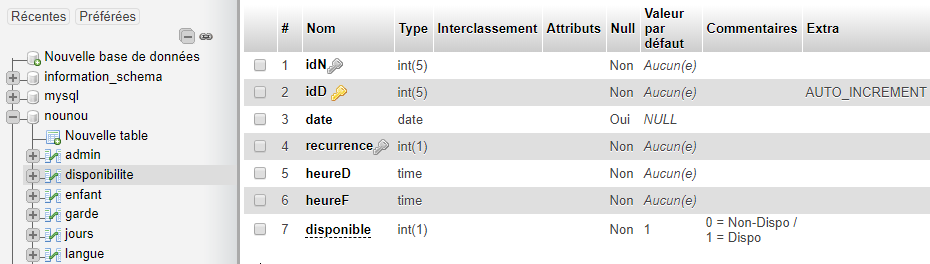


Figure 18 : Base de données - Table 'disponibilité'

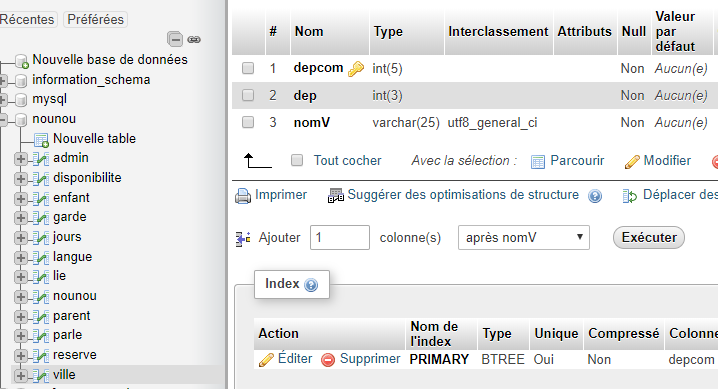


Figure 19 : Base de données - Table 'ville'

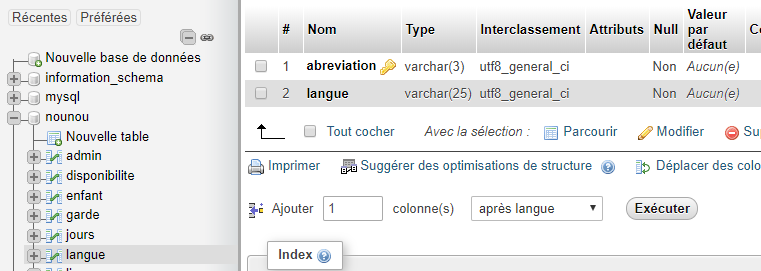


Figure 20 : Base de données - Table 'langue'

Enfin, nous avons relié les tables entre elles avec d’autres tables, là encore pour une meilleure vision de la base de données comme vous pouvez le voir sur le diagramme entité-association exposé plus haut.

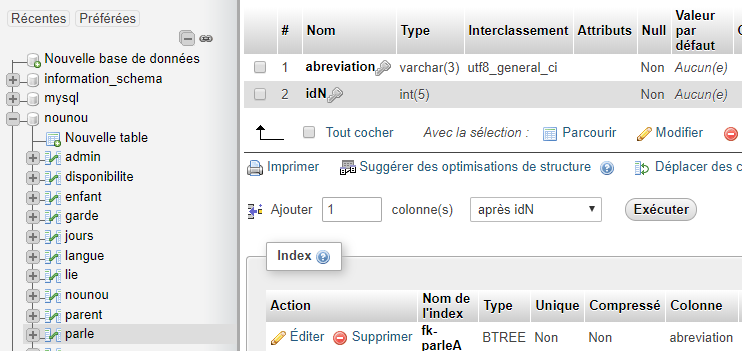


Figure 21 : Base de données - Table 'parle'

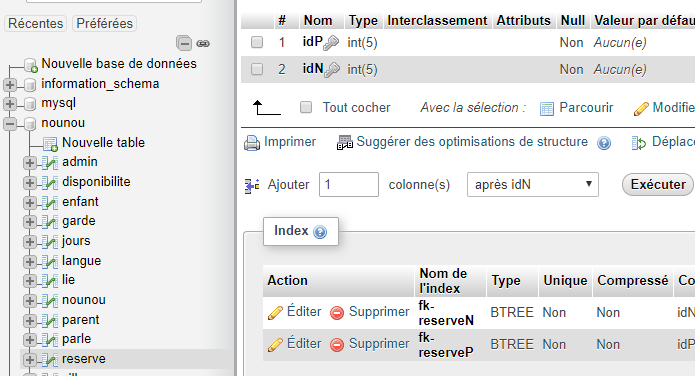


Figure 22 : Base de données - Table 'reserve'

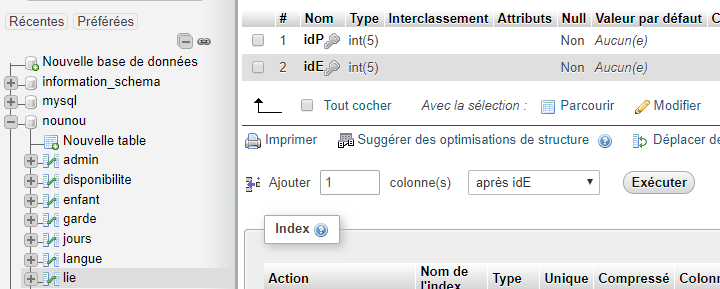


Figure 23 : Base de données - Table 'lie'

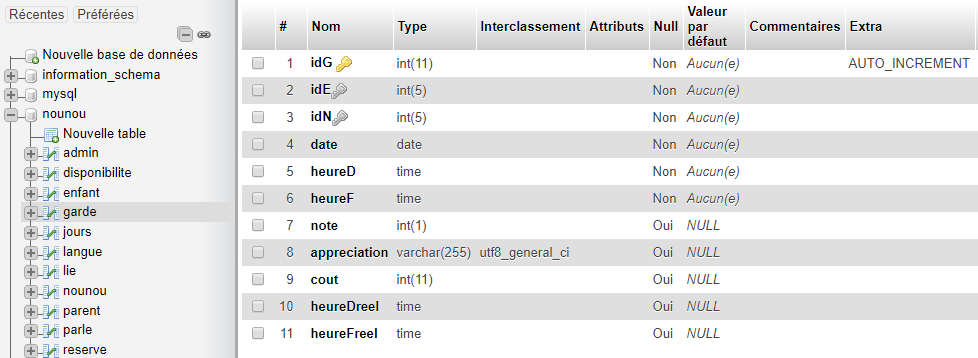


Figure 24 : Base de données - Table 'garde'

# Problèmes rencontrés

## Notre avancée par rapport aux objectifs du projet

Au moment où nous rédigeons ce rapport, nous devons faire état de l’avancée du projet par rapport aux attentes du sujet. Au vue de l’ampleur du sujet, nous n’avons pas eu le temps de traiter toutes les fonctionnalités demandées, cela du projet notamment à une trop grande précipitation. En effet, voulant mettre en pratique nos connaissances acquises tout au long du semestre, nous nous sommes vite dispersés ce qui nous a conduit à devoir nous réorganiser et trouver une manière plus simple de réaliser ce site internet.

|  |  |
| --- | --- |
| **Administrateur** | Administrateur du site web |
| Recrutement des nounous |
| **Nounou** | Inscription des nounous |
| Déclarer ses disponibilités |
| Accéder à son planning (disponibilités, gardes) |
| **Parent** | Inscription des parents et des enfants |
| Garde ponctuelle |
| Garde d’enfant en langues étrangères |
| Garde d’enfant régulière (sorties d’école, de crèche) |
| Evaluation des gardes et gestion des salaires |
| **Atteint – Partiellement Atteint – Non Atteint** | |

Figure 25 : Tableau - Avancée du projet

Les fonctionnalités manquantes sont interdépendantes les unes des autres, en résoudre une permet de terminer l’autre. C’est par exemple le cas pour l'évaluation des gardes et l'administration du site web. De la même manière, une meilleure gestion des dates de gardes, avec un calendrier précis pour la consultation des gardes, permettrait de finir la gestion des gardes.

## Difficultés rencontrées lors de la production

Tout au long de la réalisation de ce projet, nous nous sommes confrontés à des difficultés à implémenter certaines fonctionnalités, souvent en raison de la méthode que nous avons choisi d’utiliser. Tout d’abord, nous avons mis du temps à se familiariser avec le Template Bootstrap que nous avons choisi. Nous avons dès lors mis plus de temps que si nous avions réalisé ce projet avec nos propres fichiers HTML/CSS.

De plus, nous avons rencontrés des difficultés à implémenter des fonctions JavaScript, jQuery ce qui nous a pris plus du temps pour la réalisation du site.

Faire le lien avec la base de données a été aussi compliqué et nous avons dû nous tourner vers notre professeur de TD pour nous aider. La méthode trouvée, ce fut plus simple de faire le lien entre le PHP et le SQL.

# Conclusion

Nous devions concevoir après analyse du cahier des charges, un site web dynamique ayant pour caractéristique de permettre la mise en relation de nounous avec des parents. Les parents choisissent ainsi en fonction de différents critères comme la ville et la disponibilité de la nounou, pour faire garder leur(s) enfant(s).

A fortiori, notre site permet donc cette mise en relation, le tout gérer par un administrateur.

Ce projet nous a offert la possibilité de comprendre LO07 dans son ensemble et donc d’assimiler les différents enjeux et les différentes étapes de la conception web. De plus, il nous a fait découvrir et vivre le rôle d’un ingénieur en développement web ce qui a été très formateur. Nous avons pu aussi développer nos compétences managériales puisque cette réalisation web est le fruit d’un travail en binôme. Nous nous sommes partagés les tâches équitablement : création de la base de données, pages PHP et formulaires, partie liaison base de données et PHP, partie javascript, partie import et export divisées en deux. Seuls le CSS (Bootstrap) et la partie liaison base de données et PHP en elle-même n’ont pas été séparés en deux (Adrien Lebret pour la liaison base de données – PHP et Jérémie Marotte pour le CSS, Bootstrap) mais ont été l’objet de réflexion et de discussion en commun.

Enfin, sur le long terme, nous pouvons imaginer différentes fonctionnalités complémentaires pour une meilleure utilisation de notre site web. Par exemple permettre la connexion grâce aux réseaux sociaux Facebook, twitter. Pouvoir modifier ses disponibilités en tant que nounou mais encore proposer la suppression du compte, ou unifier les espaces de connexions et mettre en place un système de paiement.

# Annexe : Fichier de création de la base de données

-- phpMyAdmin SQL Dump

-- version 4.7.9

-- https://www.phpmyadmin.net/

--

-- Hôte : 127.0.0.1:3306

-- Généré le : Dim 24 juin 2018 à 17:59

-- Version du serveur : 5.7.21

-- Version de PHP : 5.6.35

SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";

SET AUTOCOMMIT = 0;

START TRANSACTION;

SET time\_zone = "+00:00";

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!40101 SET NAMES utf8mb4 \*/;

--

-- Base de données : `nounou`

--

-- Structure de la table `admin`

--

DROP TABLE IF EXISTS `admin`;

CREATE TABLE IF NOT EXISTS `admin` (

`idA` int(5) NOT NULL AUTO\_INCREMENT,

`prenomA` varchar(25) NOT NULL,

`nomA` varchar(25) NOT NULL,

`dateA` date NOT NULL,

`emailA` varchar(25) NOT NULL,

`passwordA` varchar(128) NOT NULL,

`sessionA` varchar(128) DEFAULT NULL,

`telephoneA` float(10,0) NOT NULL,

`presentationA` varchar(255) NOT NULL,

PRIMARY KEY (`idA`)

) ENGINE=InnoDB AUTO\_INCREMENT=2 DEFAULT CHARSET=utf8;

--

-- Déchargement des données de la table `admin`

--

INSERT INTO `admin` (`idA`, `prenomA`, `nomA`, `dateA`, `emailA`, `passwordA`, `sessionA`, `telephoneA`, `presentationA`) VALUES

(1, 'Admin', 'Admin', '1999-12-31', 'admin@admin.fr', '$2y$10$b4x.iDHUMtPkX4DHSSo5IeGvkOR0jlqpKDEDVRRmJ4lK7gxQL7TEC', NULL, 0, '0');

--

-- Structure de la table `disponibilite`

--

DROP TABLE IF EXISTS `disponibilite`;

CREATE TABLE IF NOT EXISTS `disponibilite` (

`idN` int(5) NOT NULL,

`idD` int(5) NOT NULL AUTO\_INCREMENT,

`date` date DEFAULT NULL,

`recurrence` int(1) NOT NULL,

`heureD` time NOT NULL,

`heureF` time NOT NULL,

`disponible` int(1) NOT NULL DEFAULT '1' COMMENT '0 = Non-Dispo / 1 = Dispo',

PRIMARY KEY (`idD`),

KEY `fk-idN` (`idN`),

KEY `fk-idJ` (`recurrence`)

) ENGINE=InnoDB AUTO\_INCREMENT=22 DEFAULT CHARSET=latin1;

--

-- Déchargement des données de la table `disponibilite`

--

INSERT INTO `disponibilite` (`idN`, `idD`, `date`, `recurrence`, `heureD`, `heureF`, `disponible`) VALUES

(4, 1, '2018-06-20', 0, '08:30:00', '14:00:00', 1),

(4, 2, '2018-06-21', 0, '08:00:00', '12:00:00', 1),

(4, 3, '2018-06-19', 0, '11:00:00', '14:00:00', 1),

(4, 4, '2018-06-21', 0, '15:30:00', '20:00:00', 1),

(4, 6, '2018-06-23', 0, '18:00:00', '23:45:00', 1),

(1, 14, '2018-06-23', 0, '18:00:00', '23:30:00', 1),

(5, 15, '2018-06-22', 0, '12:18:00', '15:40:00', 1),

(5, 16, '2018-06-26', 0, '13:00:00', '15:00:00', 1),

(1, 17, '2018-06-25', 0, '13:00:00', '19:00:00', 1),

(5, 18, '2018-06-25', 0, '14:00:00', '17:30:00', 0),

(4, 19, '2018-06-25', 0, '12:00:00', '15:00:00', 1),

(5, 20, '2018-06-25', 0, '20:30:00', '23:50:00', 1),

(5, 21, '2018-06-01', 0, '00:00:00', '15:00:00', 0);

--

-- Structure de la table `enfant`

--

DROP TABLE IF EXISTS `enfant`;

CREATE TABLE IF NOT EXISTS `enfant` (

`idE` int(5) NOT NULL AUTO\_INCREMENT,

`prenomE` varchar(25) NOT NULL,

`dateE` date NOT NULL,

`restrE` varchar(25) DEFAULT NULL,

`infoE` varchar(255) DEFAULT NULL COMMENT 'Informations sur l''enfant à savoir',

PRIMARY KEY (`idE`)

) ENGINE=InnoDB AUTO\_INCREMENT=24 DEFAULT CHARSET=utf8;

--

-- Déchargement des données de la table `enfant`

--

INSERT INTO `enfant` (`idE`, `prenomE`, `dateE`, `restrE`, `infoE`) VALUES

(14, 'Tito', '2009-12-09', 'Non', 'Aime les churros'),

(15, 'Titi', '2007-07-07', 'Allergie tapas', 'Non'),

(16, 'Tita', '2012-12-12', 'Non', 'Non'),

(17, 'boby', '2002-06-07', 'NON', 'Non'),

(18, 'Titouan', '2009-12-06', 'Non', 'Aime regarder la télé avant d\'aller se coucher.'),

(19, 'Tim', '2018-02-12', 'Non', 'Gentil'),

(20, 'Tam', '2019-05-04', 'Non', 'Sommeil léger'),

(21, 'Mathilde', '2010-01-01', 'Gluten.', 'Non'),

(22, 'Zélie', '2005-01-02', 'Lactose', 'Non'),

(23, 'Juliette', '2017-09-02', 'Poisson.', 'Non');

--

-- Structure de la table `garde`

--

DROP TABLE IF EXISTS `garde`;

CREATE TABLE IF NOT EXISTS `garde` (

`idG` int(11) NOT NULL AUTO\_INCREMENT,

`idE` int(5) NOT NULL,

`idN` int(5) NOT NULL,

`date` date NOT NULL,

`heureD` time NOT NULL,

`heureF` time NOT NULL,

`note` int(1) DEFAULT NULL,

`appreciation` varchar(255) DEFAULT NULL,

`cout` int(11) DEFAULT NULL,

`heureDreel` time DEFAULT NULL,

`heureFreel` time DEFAULT NULL,

PRIMARY KEY (`idG`),

KEY `fk-gardeN` (`idN`),

KEY `fk-gardeE` (`idE`)

) ENGINE=InnoDB AUTO\_INCREMENT=8 DEFAULT CHARSET=utf8;

--

-- Déchargement des données de la table `garde`

--

INSERT INTO `garde` (`idG`, `idE`, `idN`, `date`, `heureD`, `heureF`, `note`, `appreciation`, `cout`, `heureDreel`, `heureFreel`) VALUES

(4, 22, 1, '2018-06-25', '14:00:00', '15:00:00', NULL, '', NULL, NULL, NULL),

(7, 21, 5, '2018-06-01', '00:00:00', '12:00:00', 5, 'Giselle s\'est très bien occupée de nos enfants.', 84, '00:00:00', '12:00:00');

--

-- Structure de la table `jours`

--

DROP TABLE IF EXISTS `jours`;

CREATE TABLE IF NOT EXISTS `jours` (

`idJ` int(1) NOT NULL,

`jours` varchar(10) NOT NULL,

PRIMARY KEY (`idJ`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Structure de la table `langue`

--

DROP TABLE IF EXISTS `langue`;

CREATE TABLE IF NOT EXISTS `langue` (

`abreviation` varchar(3) NOT NULL,

`langue` varchar(25) NOT NULL,

PRIMARY KEY (`abreviation`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--

-- Déchargement des données de la table `langue`

--

INSERT INTO `langue` (`abreviation`, `langue`) VALUES

('AL', 'ALLEMAND'),

('AN', 'ANGLAIS'),

('AR', 'ARABE'),

('CH', 'CHINOIS'),

('CO', 'COREEN'),

('ES', 'ESPAGNOL'),

('FR', 'FRANCAIS'),

('HU', 'HINDI'),

('IT', 'ITALIEN'),

('JA', 'JAPONAIS'),

('PO', 'PORTUGAIS'),

('RU', 'RUSSE'),

('TH', 'THAI'),

('TU', 'TURC'),

('VI', 'VIETNAMIEN');

--

-- Structure de la table `lie`

--

DROP TABLE IF EXISTS `lie`;

CREATE TABLE IF NOT EXISTS `lie` (

`idP` int(5) NOT NULL,

`idE` int(5) NOT NULL,

KEY `fk-lieE` (`idE`),

KEY `fk-lieP` (`idP`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--

-- Déchargement des données de la table `lie`

--

INSERT INTO `lie` (`idP`, `idE`) VALUES

(16, 14),

(16, 15),

(16, 16),

(17, 17),

(17, 18),

(18, 19),

(18, 20),

(20, 21),

(20, 22),

(21, 23);

--

-- Structure de la table `nounou`

--

DROP TABLE IF EXISTS `nounou`;

CREATE TABLE IF NOT EXISTS `nounou` (

`idN` int(5) NOT NULL AUTO\_INCREMENT,

`prenomN` varchar(25) NOT NULL,

`nomN` varchar(25) NOT NULL,

`dateN` date NOT NULL,

`emailN` varchar(25) NOT NULL,

`passwordN` varchar(128) NOT NULL,

`sessionN` varchar(128) DEFAULT NULL,

`telephoneN` float(10,0) NOT NULL,

`presentationN` varchar(255) NOT NULL,

`experienceN` varchar(255) NOT NULL,

`photoN` varchar(255) NOT NULL,

`accepteN` tinyint(1) DEFAULT '0',

`bloqueN` tinyint(1) DEFAULT '0',

`depcom` int(5) NOT NULL,

PRIMARY KEY (`idN`),

KEY `fk-habiteN` (`depcom`)

) ENGINE=InnoDB AUTO\_INCREMENT=6 DEFAULT CHARSET=utf8;

--

-- Déchargement des données de la table `nounou`

--

INSERT INTO `nounou` (`idN`, `prenomN`, `nomN`, `dateN`, `emailN`, `passwordN`, `sessionN`, `telephoneN`, `presentationN`, `experienceN`, `photoN`, `accepteN`, `bloqueN`, `depcom`) VALUES

(1, 'Jeanne', 'Garros', '1985-01-01', 'jgarros@gmail.com', '$2y$10$has7fNx5XDj1P0zA1EqSdOLTRrFOjNkw.bSaqV7Cd8DKOemcD2gZK', NULL, 630959808, 'J\'ai toujours aimé garder des enfants depuis mon plus tendre jeune âge.', 'Baby-sitting quand j\'étais jeune.', '0f0d55d5b773ce54f32e9a9ad570087f.jpg', 0, 0, 59350),

(4, 'Jérémie', 'Marotte', '1997-11-07', 'jeremie.marotte@gmail.com', '$2y$10$RB5yK/c.2SqjdbOvZYKpneVmR0ByLwezVkJvmYl7tFHZaf0W6Jt12', NULL, 750247808, 'Picard de naissance, je vis à Troyes pour les études.', '-', '388859d5de1241b7d9a01d96e66d38d7.jpg', 1, 0, 80679),

(5, 'Giselle', 'Wamp', '1960-12-23', 'g.wamp@gmail.com', '$2y$10$kyvgPpxqg5by3JQ48AlDeujfSIsoAF1DpzlcheHGt7ZA51bo4350C', NULL, 687567744, 'Troyenne de naissance, je vis maintenant pres de l UTT', 'A deja garde plusieurs fois les enfants du maire', 'db505117684ee9741034c6c6a2c2e2adpng', 0, 0, 10325);

--

-- Structure de la table `parent`

--

DROP TABLE IF EXISTS `parent`;

CREATE TABLE IF NOT EXISTS `parent` (

`idP` int(5) NOT NULL AUTO\_INCREMENT,

`nomP` varchar(25) NOT NULL,

`emailP` varchar(25) NOT NULL,

`passwordP` varchar(128) NOT NULL,

`depcom` int(5) NOT NULL,

PRIMARY KEY (`idP`),

KEY `fk-habiteP` (`depcom`)

) ENGINE=InnoDB AUTO\_INCREMENT=22 DEFAULT CHARSET=utf8;

--

-- Déchargement des données de la table `parent`

--

INSERT INTO `parent` (`idP`, `nomP`, `emailP`, `passwordP`, `depcom`) VALUES

(16, 'Chicharito', 'mexique@gmail.com', '$2y$10$UokgSh7th1MST.8IAdaDBerCCjLgybh5Ezn0yNcvakBnPU.Npxe4a', 1244),

(17, 'Bernales', 'bernales@gmail.com', '$2y$10$JhxLfHWrWeS8xddHjz6Wbe/YZ9/2wUJ8UHZkF.G/QIVsKDN/5/fdi', 19122),

(18, 'Galliot', 'galliot@gmail.com', '$2y$10$SXpLcGSLOvEc7VQfMjVTz.ta.OHKXgR5oKTdPXGakrGlTgRDfZFXC', 14462),

(20, 'Azavant', 'fanny.azavant@gmail.com', '$2y$10$Uhj6OAUGR3X.AlsRq6pEceaKYikO/lPKQ74xaE9dEKFqC5MMpnkNi', 80679),

(21, 'Texaco', 'texaco@gmail.com', '$2y$10$vXqbmh.COq4393b9lnjzk.41.UanZwDQnm4KGLJT9BmxxjRxeGFx6', 75056);

--

-- Structure de la table `parle`

--

DROP TABLE IF EXISTS `parle`;

CREATE TABLE IF NOT EXISTS `parle` (

`abreviation` varchar(3) NOT NULL,

`idN` int(5) NOT NULL,

KEY `fk-parleA` (`abreviation`),

KEY `fk-parleN` (`idN`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--

-- Structure de la table `reserve`

--

DROP TABLE IF EXISTS `reserve`;

CREATE TABLE IF NOT EXISTS `reserve` (

`idP` int(5) NOT NULL,

`idN` int(5) NOT NULL,

KEY `fk-reserveN` (`idN`),

KEY `fk-reserveP` (`idP`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--

-- Structure de la table `ville`

--

DROP TABLE IF EXISTS `ville`;

CREATE TABLE IF NOT EXISTS `ville` (

`depcom` int(5) NOT NULL,

`dep` int(3) NOT NULL,

`nomV` varchar(25) NOT NULL,

PRIMARY KEY (`depcom`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--

-- Déchargement des données de la table `ville`

--

INSERT INTO `ville` (`depcom`, `dep`, `nomV`) VALUES

(2, 2, 'AFA'),

(1001, 1, 'ABERGEMENT-CLEMENCIAT'),

(1002, 1, 'ABERGEMENT-DE-VAREY'),

(…Ajouter le reste des villes via CSV...)

--

-- Contraintes pour la table `disponibilite`

--

ALTER TABLE `disponibilite`

ADD CONSTRAINT `fk-idN` FOREIGN KEY (`idN`) REFERENCES `nounou` (`idN`) ON DELETE CASCADE ON UPDATE CASCADE;

--

-- Contraintes pour la table `garde`

--

ALTER TABLE `garde`

ADD CONSTRAINT `fk-gardeE` FOREIGN KEY (`idE`) REFERENCES `enfant` (`idE`) ON DELETE CASCADE ON UPDATE CASCADE,

ADD CONSTRAINT `fk-gardeN` FOREIGN KEY (`idN`) REFERENCES `nounou` (`idN`) ON DELETE CASCADE ON UPDATE CASCADE;

--

-- Contraintes pour la table `lie`

--

ALTER TABLE `lie`

ADD CONSTRAINT `fk-lieE` FOREIGN KEY (`idE`) REFERENCES `enfant` (`idE`) ON DELETE CASCADE ON UPDATE CASCADE,

ADD CONSTRAINT `fk-lieP` FOREIGN KEY (`idP`) REFERENCES `parent` (`idP`) ON DELETE CASCADE ON UPDATE CASCADE;

--

-- Contraintes pour la table `nounou`

--

ALTER TABLE `nounou`

ADD CONSTRAINT `fk-habiteN` FOREIGN KEY (`depcom`) REFERENCES `ville` (`depcom`) ON DELETE CASCADE ON UPDATE CASCADE;

--

-- Contraintes pour la table `parent`

--

ALTER TABLE `parent`

ADD CONSTRAINT `fk-habiteP` FOREIGN KEY (`depcom`) REFERENCES `ville` (`depcom`) ON DELETE CASCADE ON UPDATE CASCADE;

--

-- Contraintes pour la table `parle`

--

ALTER TABLE `parle`

ADD CONSTRAINT `fk-parleA` FOREIGN KEY (`abreviation`) REFERENCES `langue` (`abreviation`) ON DELETE CASCADE ON UPDATE CASCADE,

ADD CONSTRAINT `fk-parleN` FOREIGN KEY (`idN`) REFERENCES `nounou` (`idN`) ON DELETE CASCADE ON UPDATE CASCADE;

--

-- Contraintes pour la table `reserve`

--

ALTER TABLE `reserve`

ADD CONSTRAINT `fk-reserveN` FOREIGN KEY (`idN`) REFERENCES `nounou` (`idN`) ON DELETE CASCADE ON UPDATE CASCADE,

ADD CONSTRAINT `fk-reserveP` FOREIGN KEY (`idP`) REFERENCES `parent` (`idP`) ON DELETE CASCADE ON UPDATE CASCADE;

COMMIT;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;