

INTERFACE ADMIN




Réaliser une interface admin de gestion des clients.

Sommaire

- Présentation
- Cahier des charges
- Les outils technologiques à utiliser
- Création de la base de données
- Architecture MVC
- Modèle
- Vue
- Contrôleur
- Conclusion

Présentation

- Ce site admin a pour but de permettre aux administrateurs des Maisons des ligues, de recenser leurs clients. Pour ce faire, il faut créer une interface admin qui permet d'ajouter un client, modifier son profile ou le supprimer.

⚙ Maison des ligues : Interface admin				
Identifiant	Nom	Prenom	Age	Adresse Mail
 Modifier ou ✕ supprimer	Eich	Branden	65	brand@gmail.com
 Modifier ou ✕ supprimer	Ledorf	Rasmus	55	rasmus@php.net
 Modifier ou ✕ supprimer	Berners Lee	Tim	70	tim@w3c.net
+ Ajouter un client				

Cahier des charges

- Créer toute la partie gestion :
 - Pour les enregistrements à publier il faut créer une page avec un formulaire de 4 champs : Nom, Prénom, Age et adresse mail. Tous les champs sont obligatoires. Afficher un warning dans le cas l'administrateur ne renseigne pas un champ.
 - Police de caractères : 'Open Sans', sans-serif
 - Iconographies obligatoires Couleur des boutons : #09599A(blue), #00614E(vert) et #d63031(rouge), La couleur des thèmes doivent être indexée dans des variables css.
 - Utilisez Visual Studio Code et phpMyAdmin pour la bdd.

Outils technologiques



Base de données

- La création de la bdd nous permet de stocker les informations des clients.

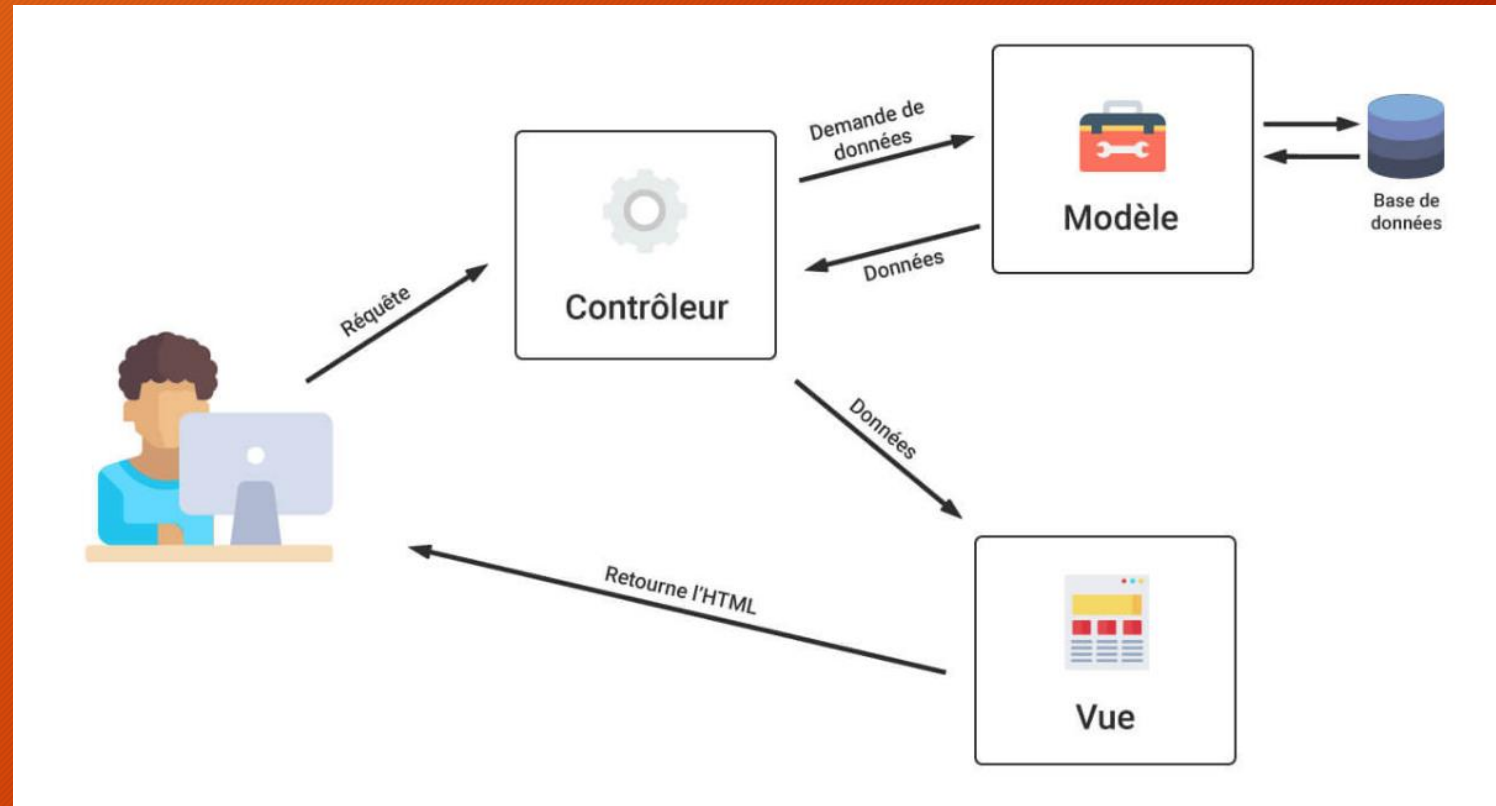


The screenshot shows a database management interface with a toolbar at the top containing buttons for 'Parcourir', 'Structure', 'SQL', 'Rechercher', 'Insérer', 'Exporter', 'Importer', 'Privileges', 'Opérations', and 'Déclencheurs'. Below the toolbar, a table lists the columns of the 'clients' table. The table has columns for '#', 'Nom', 'Type', 'Interclassement', 'Attributs', 'Null', 'Valeur par défaut', 'Commentaires', 'Extra', and 'Action'.

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/> 1	id_client	int(11)			Non	Aucun(e)		AUTO_INCREMENT	Modifier Supprimer Plus
<input type="checkbox"/> 2	nom	varchar(128)	utf8_general_ci		Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/> 3	prenom	varchar(128)	utf8_general_ci		Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/> 4	age	int(4)			Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/> 5	email	varchar(128)	utf8_general_ci		Non	Aucun(e)			Modifier Supprimer Plus

Architecture MVC

- M : Modèle
- V : Vue
- C : Contrôleur



Modèle 1 / 3

- Contient tout le back-end (connexion bdd, les fonction permettant l'ajout, la modification et la suppression des clients).

Connexion à la base de données :

```
<?php
class ConnexionBdd{

    function seConnecter(){
        try{// On se connecte à MySQL
            $pdo_options[PDO::ATTR_ERRMODE] = PDO::ERRMODE_EXCEPTION;
            $bdd = new PDO('mysql:host=localhost;dbname=client_ligue', 'root', '',
                array(PDO::MYSQL_ATTR_INIT_COMMAND => 'SET NAMES utf8',$pdo_options));
            return $bdd;
        }
        catch(Exception $e){
            // En cas d'erreur précédemment, on affiche un message et on arrête tout
            die('Erreur : '.$e->getMessage());
        }
    }
}
```

Affichage du client :

```
<?php
require_once("../src/modele/connexionBdd.inc.php");

class AffichageClient extends ConnexionBdd{

    function publierLeClient(){
        // On récupère tout le contenu
        $reponse = $this->seConnecter()->query('SELECT * FROM `clients` LIMIT 20'); /* LIMIT 1 : limite la liste à afficher*/

        // On affiche chaque entrée une à une
        while ($donnee = $reponse->fetch()){
            echo <table>
            <tr>
                <td>
                    <i class="fa fa-pencil"></i>
                    <a href="index.php?action=mettreAJour&idClient='.$donnee['id_client']. ' ">Modifier</a> ou
                    <i class="fa fa-times"></i>
                    <a href="index.php?action=supprimer&supprimer='.$donnee['id_client']. ' ">Supprimer</a>
                </td>
                <td>'.$donnee['nom'].</td>
                <td>'.$donnee['prenom'].</td>
                <td>'.$donnee['age'].</td>
                <td>'.$donnee['email'].</td>
            </tr>
            </table>;
        }
        $reponse->closeCursor(); // Termine le traitement de la requête
    }
}
```


Modèle 2/3

- Ajout du client :
- Modification du client :

```
<?php
include_once("../src/modele/connexionBdd.inc.php");

class AjouterClient extends ConnexionBdd{

    function insererClient(){
        if(isset($_POST['nom']) || isset($_POST['prenom']) || isset($_POST['age']) || isset($_POST['email'])){

            if(!isset($_POST['nom']) || !isset($_POST['prenom']) || !isset($_POST['age']) || !isset($_POST['email'])){
                print "<p class='warning'>Vous avez oublié de remplir un ou plusieurs champs !</p>";
            }
            else if(!is_numeric($_POST['nom']) || !is_numeric($_POST['prenom'])){
                print "<p class='warning'>Vous devez saisir des caractères...</p>";
            }
            else if(!is_numeric($_POST['age'])){
                print "<p class='warning'>Veuillez saisir des nombres pour votre âge...</p>";
            }
            else{
                $req = $this->seConnecter()->prepare('INSERT INTO clients(nom, prenom, age, email)VALUES(?,?,?,?)');
                $req->execute(array($_POST['nom'], $_POST['prenom'], $_POST['age'], $_POST['email']));

                echo "<p class='success'>Merci le client a été ajouté ;)<br></p>";
            }
        }
    }
}
```

```
<?php
require_once("../src/modele/connexionBdd.inc.php");

class MettreAJourClient extends ConnexionBdd{

    function clientAJour($id_client, $nom, $prenom, $age, $email)
    {
        $reponse = $this->seConnecter()->query("SELECT * FROM `clients` WHERE id_client = ' . $id_client . '");
        $donnees = $reponse->fetch();

        $requete = $this->seConnecter()->prepare('UPDATE clients SET  nom = :nom, prenom = :prenom, age = :age, email = :email
        WHERE id_client = ' . $id_client . ' ');
        $requete->execute(array(
            'nom' => $nom,
            'prenom' => $prenom,
            'age' => $age,
            'email' => $email,
        ));
    }
}
```

Modèle 3/3

- Méthodes permettant de récupérer chaque infos de la bdd :

```
function afficherLeNom($id_client){
    $reponse = $this->seConnecter()->query("SELECT * FROM 'clients' WHERE id_client = " . $id_client . " ");
    $donnees = $reponse->fetch();
    return $donnees['nom'];
}

function afficherLePrenom($id_client){
    $reponse = $this->seConnecter()->query("SELECT * FROM 'clients' WHERE id_client = " . $id_client . " ");
    $donnees = $reponse->fetch();
    return $donnees['prenom'];
}

function afficherAge($id_client){
    $reponse = $this->seConnecter()->query("SELECT * FROM 'clients' WHERE id_client = " . $id_client . " ");
    $donnees = $reponse->fetch();
    return $donnees['age'];
}

function afficherEmail($id_client){
    $reponse = $this->seConnecter()->query("SELECT * FROM 'clients' WHERE id_client = " . $id_client . " ");
    $donnees = $reponse->fetch();
    return $donnees['email'];
}

function supprimerClient($id_client){
    $requete = $this->seConnecter()->prepare('DELETE FROM clients WHERE id_client = :id_client');
    $requete->execute(array('id_client' => $id_client));
}
}
```

- Suppression du client :

```
<?php







require_once("../src/modele/connexionBdd.inc.php");

class SuppressionClient extends ConnexionBdd{

    function supprimerClient($id_client){
        $requete = $this->seConnecter()->prepare('DELETE FROM clients WHERE id_client = :id_client');
        $requete->execute(array('id_client' => $id_client));
    }
}
```

Vue 1/2

- Contient tout le Front-end, l'interface utilisateur (Landing page, les formulaires permettant l'ajout, la modification et la suppression des clients).
- Page d'accueil :

⚙ Maison des ligues : Interface admin				
Identifiant	Nom	Prenom	Age	Adresse Mail
 Modifier ou  supprimer	Elch	Branden	65	brand@gmail.com
 Modifier ou  supprimer	Ledorf	Rasmus	55	rasmus@php.net
 Modifier ou  supprimer	Berners Lee	Tim	70	tim@w3c.net
+ Ajouter un client				

Vue 2/2

- Formulaire créer client :


VOIR LA LISTE DES UTILISATEURS >

Nom :

Prenom

Age:

Adresse mail:

Créer 

- Formulaire modifier ou supprimer client :


VOIR LA LISTE DES UTILISATEURS >


Nom :

Prenom

Age:

Adresse mail:

Mettre à jour 

Supprimer 

Contrôleur

- Etablie la communication entre le Front-end et le Back-end.

```
<?php
require_once("../src/modele/ajouterClient.inc.php");
require_once("../src/modele/affichageClient.inc.php");
require_once("../src/modele/supprimerClient.inc.php");
require_once("../src/modele/mettreAJourClient.inc.php");

function afficherClient(){
    $afficherLeClient = new AffichageClient();
    require("../vue/publier_utilisateur.php");
}

function ajouterClient(){
    $ajouterLeClient = new AjouterClient();
    require("../vue/creer_utilisateur.php");
}

function supprimerClient(){
    $supprimer = new SuppressionClient();
    $supprimer->supprimerClient($_GET['supprimer']);
    header('Location:index.php');
}

function mettreAJourClient(){
    $mettreAJour = new MettreAJourClient();
    require("../vue/mettre_a_jour.php");
}

function mettreAJourBdd()
{
    $mettreAJour = new MettreAJourClient();
    $mettreAJour->clientAJour($_GET['idClient'], $_POST['nom'], $_POST['prenom'], $_POST['age'], $_POST['email']);
    header('Location:index.php');
}
```

Conclusion

- J'ai réalisé un projet d'interface de gestion des clients permettant d'insérer des clients dans une base de données, les modifier ainsi que les supprimer.
- J'ai mis en place une architecture MVC (Modèle Vue Contrôleur) pour réaliser ce projet.