Compte rendu du projet Hotel

Cédric Leroy

Analyse du site

En premier lieu, et avant toute modification, il est nécessaire d'analyser l'ensemble du site. Comme le veut l'exercice, j'ai distingué les éléments statiques des éléments qui pouvaient figurer dans une base de données.

On observe tout d'abord que le site présente un header avec le logo de l'hôtel ainsi que les liens de la navigation. On peut déjà noter que cette partie sera l'un des éléments du template par défaut qui encadrera le contenu de chaque page.

La première « page » est une description textuelle de l'hôtel et de ses prestations, suivie de plusieurs images représentant les « plus » de l'établissement.
Cette partie ne nécessite pas d'être intégrée à une base de données car elle ne présente pas d'éléments redondants et ne semble pas susceptible de changer souvent.

La deuxième page est celle des chambres : là encore un texte décrivant les équipements de l'hôtel.
Tout comme la page « Hôtel », la page « Chambre » ne semble pas vouée à changer fréquemment. Une base de données n'est pas nécessaire.

En troisième page, une galerie de photos des différentes chambres et de l'hôtel.
Ici, on peut se dire que les gérants voudraient pouvoir changer régulièrement les photos, les mettre à jour… On peut alors noter qu'une table « Galerie » serait intéressante à créer.

Ensuite vient la page « Tarifs ». Cette liste des chambres et options associées fait l'objet d'une consigne particulière : celle de la rendre modifiable via une base de données.
Cette page fera l'objet d'une partie dédiée plus loin dans le compte rendu.

La page « Tarifs » et la page « Réservations » sont entrecoupées par une grande photo de la façade de l'hôtel. N'étant pas réellement une page, elle sera pour le moment mise de côté en vue d'être attribuée à une page plus tard.

A noter que cette photo prend l'emplacement de la page « Tourisme » dans l'ordre de la navigation. La page « Tourisme », elle, est une page parallèle au site on page. Elle fera l'objet d'une partie du compte rendu. Notons tout de même que son contenu sera intégré dans une table de notre base de données.

La page « Réservations », quant à elle, ne fera pas l'objet de beaucoup de changements. La réservation sera gérée par une centrale externe.

Vient ensuite la page « Infos pratiques ». Cette page contient les informations de l'hôtel : son adresse, numéro de téléphone, emplacement sur la carte et les horaires d'ouverture et fermeture. Ces informations n'étant pas amenées à changer, pas de base de données ici.

Pour finir, on retrouve un footer : un petit bandeau comportant le nom des créateurs du site ainsi qu'un lien vers les mentions légales.
D'ailleurs, cette page n'est pas mentionnée dans les consignes, mais elle sera bien évidemment elle aussi intégrée au projet.

Découpage et Routage

Cette analyse préliminaire m'a permis de faire un découpage du site et d'en déterminer les différentes pages :

- La page hôtel
- La page chambres
- La page galerie
- La page tarifs suivie de la grande photo de l'hôtel : c'est l'emplacement le plus adapté, car mettre la photo juste avant les réservations rendrait cette partie moins visible
- La page réservations
- La page tourisme
- La page infos pratiques
- La page des mentions légales
- La page erreur 404 spécifique au routeur utilisé

Une fois ce découpage effectué, j'ai pu démarrer la mise en place du routage : création de la page index.php, liste des différentes routes du site et création du fichier .htaccess.
Après avoir testé que l'url était correctement traitée par le .htaccess, j'ai initialisé Composer pour l'autoload, et pour le router. Il a ensuite fallu calibrer AltoRouter. J'ai dû évidemment adapter les éléments que vous m'aviez fourni et au bout de plusieurs essais, le router fonctionnait comme prévu.

J'ai donc pu m'attaquer au découpage html : à partir du site statique, j'ai conservé le header avec la nav ainsi que le footer dans un fichier template.php.
Le reste a ensuite été réparti dans les différentes vues : v_hotel.php, v_galerie.php, …
Tant qu'à faire, j'ai profité de cette étape pour transférer les dossiers du site statique dans mon dossier public (images, css et js).

Une fois les différentes vues créées, j'ai commencé à créer les différents controllers, et en premier HotelController. J'ai repris la structure faite sur le blogMVC pour aller plus vite.
Après m'être assuré que le controller fonctionnait, j'ai pu directement faire le controller principal. J'ai directement implémenté l'ensemble de méthodes ob_start() pour pouvoir contrôler que le controller appelait bien la vue v_hotel.

Là encore, après avoir vérifié le bon fonctionnement des choses, j'ai pu copier/coller pour les controllers suivants.

J’ai bien entendu ajouter chaque route à mon index. Et le découpage du site était fini : le router comprenait bien toutes les routes internes du site, le bon controller était appelé par la bonne route, et il appelait bien la bonne vue. Le site statique était bel et bien reproduit à l’identique.

public > index.php > ...

```php
<?php
use App\Router;


$url = ($_GET['url'])??null;

require '../vendor/autoload.php';

define('hotelMVC', '/');
$router = new Router();

$router->get('', 'Hotel#index');
$router->get('chambres', 'Chambres#index');
$router->get('pratique', 'Pratique#index');
$router->get('reservation', 'Reservation#index');
$router->get('galerie', 'Galerie#index');
$router->get('tarifs', 'Tarifs#index');
$router->get('mentions', 'Mentions#index');
$router->get('tourisme', 'Tourisme#index');
$router->get('admin', 'Admin#index');
$router->get('edit', 'Edit#detailTarif');
$router->post('updateTarif', 'Edit#updateTarif');



$match = $router->run();

```

Image de l’utilisation d’AltoRouter

Il était temps de s’occuper des pages manquantes.

Les pages en plus : Tourisme, mentions légales et erreur 404

La consigne principale étant de reproduire le site à l'identique, et avant d'intégrer la base de données, j'ai pris soin de rajouter les pages manquantes au site fourni. A savoir : la page Tourisme et la page Mentions légales. En bonus : j'ai fait une petite page erreur 404 pour coller au routeur que vous m'avez fourni.

C'est par cette dernière que j'ai commencé : j'ai repris la structure de la page hôtel et j'ai simplement mis le message de l'erreur ainsi qu'un lien de redirection vers l'accueil (page hôtel).

```
<div id="index-hotel-titre">
    <h2><label>ERREUR 404</label></h2>
</div>
<br>
<div class="barre-array-rouge"></div>
<div class="arrow-down-rouge"></div>
<div class="barre-array-rouge"></div>

<br>
<br>

<div id="index-label-description-hotel">
    <h3>
        <label>
            NOUS SOMMES DÉSOLÉS, CETTE PAGE EST INTROUVABLE.
            <br>
            <br>
            <div class="tourisme-cell-droite"><a target="" href="/">> CLIQUEZ ICI POUR REVENIR A L'ACCUEIL DU SITE</a></div>
            <br>
            <br>

        </label>
    </h3>
</div>
```

Code de la page Erreur 404

Pour la page mentions légales, je n'ai eu qu'à reprendre le code source de la page du vrai site et la rajouter dans ms vues, dans mes controllers et dans mon router.

Hôtel du Relais
2 rue de la Croix Camus
79100 Thouars
Tél : 05.49.66.29.45
Fax : 05.49.66.29.33

Directeur de la publication :
Hôtel du Relais

Réalisation éditoriale :
Hôtel du Relais

Responsable technique :
Nexti // Nexti

Responsable graphique :
Univers'elles // Univers'elles

Avertissement
Les informations contenues dans le site n'ont qu'une valeur indicative et sont susceptibles d'évoluer en fonction des modifications législatives et réglementaires. Nous ne pouvons être tenue responsable de l'interprétation que vous pourriez faire des informations contenues dans ce site. Tous les documents accessibles en téléchargement sont fournis à titre informatif et n'ont pas de valeur légale. Seul un document délivré à l'entreprise peut être utilisé comme référence.

Page des mentions légales

Et c'a été la même chose pour la page tourisme : toutes les images étaient déjà présentes dans le dossier fourni. Je n'ai eu qu'à copier/coller le code source de la page.

Le site est à présent complet et identique au vrai site, à l'exception de l'aspect one page bien entendu.

Je peux donc m'atteler à la partie base de données.

Base de données

Grâce à l'analyse en début de projet, j'ai déjà une bonne idée des différentes tables de ma base de données Hotel : les tarifs, la galerie photos, et les trois parties de la page tourisme.

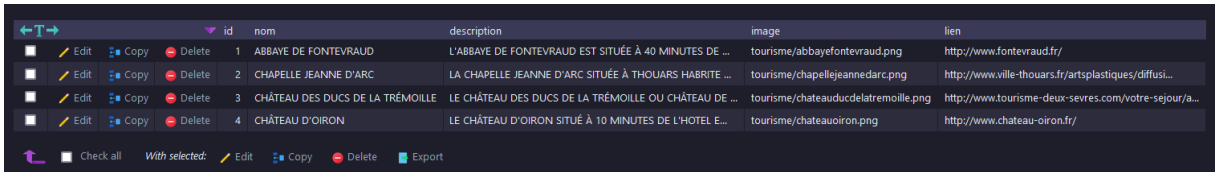
J'ai commencé par la page tourisme, puisque je venais de l'intégrer au site.
J'ai donc délimiter 3 parties : les sorties, les monuments, et les villes. J'ai conscience que, contrairement aux sorties, les villes et monuments risquent de ne pas trop changer (voir même pas du tout) mais certains éléments comme les descriptions, les liens et les photos peuvent être amenés à être modifiés par les gérants. Et puis tant qu'à mettre de la base de données dans une page, autant tout faire : ça fait de l'entraînement.

J'ai donc créé via phpMyAdmin la table sorties, la table monuments et la table villes ; et j'ai repris pour chaque ligne le nom, la description, la photo et le lien.
Notez que chaque table que j'évoquerai dans ce compte rendu sera bien entendu munie d'une colonne id qui s'auto incrémente et qui est la primary key de la table.

	id	nom	description	image	lien
Edit Copy Delete	1	CENTER PARCS	LE CENTER PARCS - DOMAINE DU BOIS AUX DAIMS EST À ...	tourisme/centerparcs.png	https://www.centerparcs.fr/fr-fr/france/fp_BD_vaca...
Edit Copy Delete	2	CHÂTEAU DE SAINT MESMIN	LE CHÂTEAU DE SAINT MESMIN EST À 45 MINUTES DE ROU...	tourisme/stmesmin.png	https://www.chateau-saintmesmin.com/
Edit Copy Delete	3	FUTUROSCOPE	LE FUTUROSCOPE EST UN PARC D'ATTRACTION TRAVAILLAN...	tourisme/futuroscope.png	LE FUTUROSCOPE EST UN PARC D'ATTRACTION TRAVAILLAN...
Edit Copy Delete	4	KARTING VAL D'ARGENTON	LE KARTING D'ARGENTON-LES-VALLÉES	tourisme/kartingloudun.png	https://pks-loisirs.fr/
Edit Copy Delete	5	KARTING FRANCK TALON	LE KARTING DE LOUDUN (FRANCK TALON)	tourisme/kartingloudun.png	https://www.loudun-karting.com/
Edit Copy Delete	6	MONGOLFIADES	NOUS SOMMES À PROXIMITÉ DU CENTRE DES MONGOLFIADES...	tourisme/mongolfiade.png	http://www.tourisme-pays-thouarsais.fr/evenement/l...
Edit Copy Delete	7	PARC DE LA VALLÉE	LE PARC DE LA VALLÉE À PROXIMITÉ DE THOUARS, EST U...	tourisme/parcdelavallee.png	https://parcdelavallee.fr/
Edit Copy Delete	8	LE PUY DU FOU	LE PUY DU FOU EST À 1H DE ROUTE DE L'HOTEL. CE PAR...	tourisme/puydufou.png	https://www.puydufou.com/fr
Edit Copy Delete	9	THÉÂTRE DE THOUARS	LE THÉÂTRE DE THOUARS ACCUEILLE DE NOMBREUX SPECTA...	tourisme/theatrethouars.png	https://www.theatre-thouars.com/index_m.html
Edit Copy Delete	10	ZOO DE DOUÉ LA FONTAINE	LE ZOO DE DOUÉ LA FONTAINE EST SITUÉ À 25 MINUTES.	tourisme/zoodoue.png	http://www.bioparc-zoo.fr/fr/

Check all *With selected:* Edit Copy Delete Export

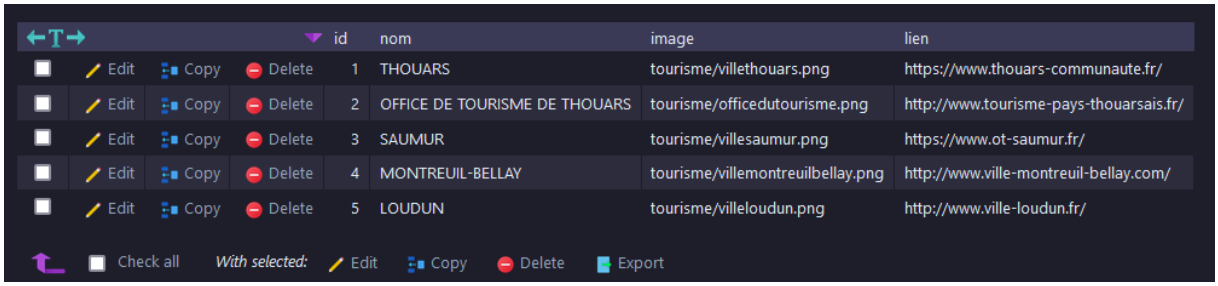
Table des sorties



	id	nom	description	image	lien
Edit Copy Delete	1	ABBAYE DE FONTEVRAUD	L'ABBAYE DE FONTEVRAUD EST SITUÉE À 40 MINUTES DE ...	tourisme/abbayefontevraud.png	http://www.fontevraud.fr/
Edit Copy Delete	2	CHAPELLE JEANNE D'ARC	LA CHAPELLE JEANNE D'ARC SITUÉE À THOUARS HABRITE ...	tourisme/chapellejeannedarc.png	http://www.ville-thouars.fr/artsplastiques/diffusi...
Edit Copy Delete	3	CHÂTEAU DES DUCS DE LA TRÉMOILLE	LE CHÂTEAU DES DUCS DE LA TRÉMOILLE OU CHÂTEAU DE ...	tourisme/chateauducdelatremoille.png	http://www.tourisme-deux-sevres.com/votre-sejour/a...
Edit Copy Delete	4	CHÂTEAU D'OIRON	LE CHÂTEAU D'OIRON SITUÉ À 10 MINUTES DE L'HOTEL E...	tourisme/chateauoiron.png	http://www.chateau-oiron.fr/

Check all *With selected:* Edit Copy Delete Export

Table des monuments



	id	nom	image	lien
Edit Copy Delete	1	THOUARS	tourisme/villethouars.png	https://www.thouars-communaute.fr/
Edit Copy Delete	2	OFFICE DE TOURISME DE THOUARS	tourisme/officedutourisme.png	http://www.tourisme-pays-thouarsais.fr/
Edit Copy Delete	3	SAUMUR	tourisme/villesaumur.png	https://www.ot-saumur.fr/
Edit Copy Delete	4	MONTREUIL-BELLAY	tourisme/villemontreuilbellay.png	http://www.ville-montreuil-bellay.com/
Edit Copy Delete	5	LOUDUN	tourisme/villeloudun.png	http://www.ville-loudun.fr/

Check all *With selected:* Edit Copy Delete Export

Table des villes

		id	image	alt
Edit Copy Delete		1	55.JPG	facade
Edit Copy Delete		2	105.jpg	SALON
Edit Copy Delete		3	603.jpg	Chambre N°3
Edit Copy Delete		4	612.jpg	Chambre Twin
Edit Copy Delete		5	611.jpg	Chambre familiale
Edit Copy Delete		6	04.jpg	Chambre familiale partie enfant
Edit Copy Delete		7	DSCN5989.jpg	Chambre Bain 5
Edit Copy Delete		8	604.jpg	Chambre N°4
Edit Copy Delete		9	DSCN5990.jpg	Salle de bain Ch 5
Edit Copy Delete		10	06.jpg	Chambre 15
Edit Copy Delete		11	12.jpg	Chambre 11
Edit Copy Delete		12	07.jpg	Chambre 7
Edit Copy Delete		13	17.jpg	Extérieur
Edit Copy Delete		14	Petit dejeune.jpg	Petit déjeuné
Edit Copy Delete		15	Ville.jpg	Ville de Thouars

Check all *With selected:* Edit Copy Delete Export

Table des photos

		id	item	prix
Edit Copy Delete		0	Chambre 1 personne (avec douche)	61.2
Edit Copy Delete		1	Chambre 1 personne (avec baignoire)	64
Edit Copy Delete		2	Chambre 2 personnes (avec douche)	65
Edit Copy Delete		3	Chambre 2 personnes (avec baignoire)	66
Edit Copy Delete		4	Chambre Twin (2 lits simples)	66
Edit Copy Delete		5	Chambre triple (1 lit de 160 - 1 lit de 90)	78
Edit Copy Delete		6	Chambre communicante (1 lit de 160 et 2 lits de 90...	84
Edit Copy Delete		7	Chambre communicante (1 lit de 160 et 2 lits de 90...	88
Edit Copy Delete		8	Suite familiale (2 salles d'eau) pour 2 personnes	70
Edit Copy Delete		9	Suite familiale (2 salles d'eau) pour 3 personnes	86
Edit Copy Delete		10	Suite familiale (2 salles d'eau) pour 4 personnes	90
Edit Copy Delete		11	Petit dejeuner buffet	8.5
Edit Copy Delete		12	Supplement service en chambre	1.4
Edit Copy Delete		13	Soiree etape	78
Edit Copy Delete		14	Menu (du lundi au jeudi)	15
Edit Copy Delete		15	Animaux (sur demande)	5
Edit Copy Delete		16	Taxe de séjour	0.7

Check all *With selected:* Edit Copy Delete Export

Table des tarifs

Une fois les tables créées, j'ai intégré la class Database au dossier et j'ai préparé le model HotelModel.php.
En reprenant les éléments faits sur le blog, je n'ai eu qu'à adapter la requête pour trois fonctions : getSorties, getMonuments et getVilles.

Page HotelModel.php

Je ramène le résultat de ces requêtes dans le TourismeController et je regroupe les tableaux associatifs donnés par les fetchAll en variables $sorties, $monuments et $villes.
J'utilise la fonction compact() pour les réunir dans la variable $data. Celle-ci sera le second paramètre de la fonction render du controller principal.

src > controllers > Controller.php > PHP Intelephense > Controller > a

```php
<?php
namespace Hotel\controllers;

abstract class Controller
{

    public function render(string $view, $data=[])
    {
        ob_start();

        extract($data);
        // extract($prix);

        require('../views/'.$view.'.php');

        $contenu = ob_get_clean();

        require('../views/template/template.php');

    }
```

HotelController.php

Dans ce controller, je n'ai plus qu'à utiliser extract() pour pouvoir afficher les données voulues dans la vue souhaitée.

Cela me permet pour finir d'afficher grâce à un foreach (enfin 3, un pour chaque table) tous les enregistrements des tables.

La page tourisme a cependant une petite subtilité : l'affichage des différentes sorties, monuments et villes ne se fait pas sous la forme d'un tableau mais d'une succession de disposées en flex. J'ai donc pu m'inspirer des premiers exercices fait en PHP, notamment le fameux échiquier, et en modifiant un peu les paramètres, j'ai réussi l'alternance des couleurs de background.

```php
<div class="tourisme-bloc-flex">
<?php
    $count = 0;
    $classeCSS = 'tourisme-flex-blanche';
    $cote = '';
    foreach($sorties as $sortie):
        $count++;

        $classeCSS = ($count % 4 == 0 || $count % 4 == 1) ? 'tourisme-flex-beige' : 'tourisme-flex-blanche';

        if($cote=='gauche'){ $cote = 'droite'; } else { $cote = 'gauche'; }

    echo '<div class=" '. $classeCSS .' ">';
        echo '<div class="tourisme-contenu-'.$cote.'">';
            echo '<img class="tourisme-img-'.$cote.'" src="pics/'. $sortie['image'].'">';
            echo '<div class="tourisme-table-'.$cote.'">';
                echo '<div class="tourisme-cell-gauche"><label class="tourisme-sous-titre">'. $sortie['nom'].'</label></div>';
                echo '<div class="tourisme-cell-droite"><a target="_blank" href="'. $sortie['lien'].' ">> SITE</a></div>';
            echo '</div>';
            echo '<label class="tourisme-label-gauche">'. $sortie['description'].'</label>';
        echo '</div>';


    if ($count % 2 == 0) {
        echo '</div><div class="tourisme-bloc-flex">';
    } else {
        echo '</div>';
    }

    endforeach; ?>
```

Extrait du code de la page tourisme

Les autres tables ont été plus simples à intégrer au site.

Les photos de la galerie s'affichent également grâce à un foreach.

La page des tarifs est à mon goût un peu moins « propre » car la façon dont sont présentés les différents tarifs n'est pas récurrente : on retrouve des items en gras, d'autres (les sous-options) sont écrites dans une taille plus petite, certaines options sont par nuit, etc…

N'ayant pas de pattern à répéter, j'ai simplement affiché chaque tarif selon l'id qu'il détient dans la table de données.

```php
<div class="tablerow">
    <div class="index-tarifs-sous-titre"><label>CHAMBRE 1 PERSONNES</label></div>
</div>
<div class="tablerow">
    <div class="tablecell-gauche">
        <label class="tarifs-details">-avec douche</label>
    </div>
    <div class="tablecell-droite">
        <label class="tarifs-details"><?php echo $data[0]['prix']; ?> €</label>
    </div>
</div>


<div class="tablerow">
    <div class="tablecell-gauche">
        <label class="tarifs-details">-avec baignoire</label>
    </div>
    <div class="tablecell-droite">
        <label class="tarifs-details"><?php echo $data[1]['prix']; ?> €</label>
    </div>
</div>
```

Affichage des tarifs

Cela conclue la partie sur la base de données.

Après cela, j'ai pu m'attaquer à la partie admin et la modification des tarifs via cette interface.

Interface admin

En préambule de cette dernière partie, je tiens à m'excuser : je n'ai pas su reproduire le login et le logout censés protéger la page admin.php. Je me souviens de ce qui a été vu en cours mais je dois avouer que l'appliquer au modèle MVC m'a pas mal embrouillé.

Je ne peux donc pas présenter le login et le logout. J'ai en revanche réussi à intégrer une page admin avec une page edit permettant de modifier les tarifs du site via la base de données. Etant donné qu'il n'y a pas d'accès protégé, on accède à la page admin en tapant 'admin' dans l'url.

Pour se faire, j'ai commencé par ajouter deux nouvelles requêtes dans HotelModel : la fonction getTarifById qui me permettra d'afficher l'item qu'on souhaite modifier ; et la fonction setNewTarif pour modifier la table des tarifs.

```php
public function getTarifs()
{

    return $this->db->query('SELECT id, item, prix FROM tarifs')->fetchAll();

}

public function getTarifById($id)
{

    // return $this->db->query("SELECT id, item, prix FROM tarifs WHERE id=$id")->fetch();
    return $this->db->query('SELECT id, item, prix FROM tarifs WHERE id=?',[$id])->fetch();
}

public function setNewTarif($newtarif, $id)
{
    return $this->db->query('UPDATE tarifs SET prix =? WHERE id=?', [$newtarif,$id]);
}
```

Fonctions propres à la récupération des tarifs

Je créé ensuite deux nouveaux controllers : Admin et Edit. Dans le premier, je reprends la fonction listeTarifs pour récupérer un tableau clair de tous les items et le prix associé. Et dans Edit je récupère la requête du tarif selon l'id, et la fonction pour modifier le prix.

src > controllers > AdminController.php > ...

```php
<?php
namespace Hotel\controllers;

use Hotel\models\HotelModel;

class AdminController extends Controller
{
    private $hotel;

    public function __construct()
    {

        $this->hotel = new HotelModel();

    }

    public function listeTarifs(){

        return $this->hotel->getTarifs();
    }

    public function index()
    {
        $tarifs = $this->listeTarifs();
        $this->adminRender('v_admin', $tarifs);

    }

}
```

AdminController.php

src > controllers > EditController.php > PHP Intelephense > EditController

```php
<?php
namespace Hotel\controllers;

use Hotel\models\HotelModel;

class EditController extends Controller
{
    private $hotel;

    public function __construct()
    {

        $this->hotel = new HotelModel();

    }

    public function detailTarif(){
        $id = $_GET['id'];

        $tarif = $this->hotel->getTarifById($id);
        $this->adminRender('v_edit', $tarif);


    }

    public function updateTarif() {

        $newtarif = $_POST['prix'];

        $id = $_POST['id'];
        $this->hotel->setNewTarif($newtarif, $id);
        $this->adminRender('v_admin');
    }

}
```

EditController.php

Après cela, j'adapte mon controller principal : je n'ai pas besoin d'afficher le template de la partie visiteur, alors je créé la fonction adminRender qui fonctionnera comme render() mais qui n'affichera que la vue demandée, sans header et footer.

```php
public function adminRender(string $view, $data=[])
{
    ob_start();

    extract($data);

    $contenu = ob_get_clean();

    require('../views/'.$view.'.php');
}
```

Fonction adminRender()

Pour la partie vue, la page v_admin affiche un tableau de quatre colonnes : l'id, l'option et son prix, ainsi que le lien pour modifier la ligne choisie.

```php
<body>
    <h1>Liste des tarifs</h1>
    <table>
        <tr>
            <th>id</th>
            <th>Item</th>
            <th>Prix</th>
        </tr>
            <?php foreach($data as $prix):
            echo '<tr>';
            echo '<td>'.$prix['id'].'</td>';
            echo '<td>'.$prix['item'].'</td>';
            echo '<td>'.$prix['prix'].'</td>';
            echo '<td><a href="edit?id='.$prix['id'].'">Modifier tarif</td>';
            echo '</tr>';
            endforeach;
            ?>

    </table>
</body>
```

La vue v_admin.php

Ce lien nous mène à la seconde vue : v_edit qui comporte un formulaire avec un seul input de type number avec un step de 0.1 permettant d'entrer un nouveau prix.
Le bouton submit appelle la fonction updateTarif.

```
<body>
    <form method="post" action="updateTarif">
        <label for="prix">Nouveau Prix</label>
        <input type="number" step="0.1" name="prix" id="prix" value="<?php echo $data['prix']; ?>"><br>
        <input type="hidden" step="0.1" name="id" id="id" value="<?php echo $_GET['id']; ?>"><br>
        <input type="submit">
    </form>
</body>
```

La vue v_edit.php

Ce n'est pas la partie la plus aboutie, il faut bien le reconnaître : on doit taper manuellement admin dans l'url pour revenir au tableau modifié car je n'ai pas réussi à manipuler l'action du formulaire comme je le souhaitais.

Par ailleurs, j'ai mis beaucoup de temps avant de m'apercevoir que la class du Router comportait une méthode post en plus de la méthode get.

J'arrive donc à un résultat fonctionnel, mais loin de ce que j'aurais voulu. Il en est de même pour le côté esthétique, bien que cela soit plus accessoire.

Conclusion

Pour conclure sur ce projet, je dirais que j'ai bien assimilé le modèle MVC et ses avantages. Le routage et le découpage ont été les parties les moins compliquées.

J'ai été satisfait de réussir à maîtriser les fonctions compact() et extract() : j'ai réussi à les mettre en application et je vois clairement aujourd'hui ce que ça permet de faire.

Les bases de données n'ont pas représenté un obstacle significatif, même si je sais que j'aurais sans doute pu pousser un peu plus la conception, avec des jointures par exemple. Je n'ai pas vraiment vu où je pouvais m'en servir.

Enfin, la partie admin et modification des tarifs est celle qui m'a donné le plus de fil à retordre. Même en ayant une bonne idée de ce que j'avais à faire, l'appliquer au découpage MVC m'a posé beaucoup de problèmes. Ce qui fait que cette partie n'est pas totalement aboutie.