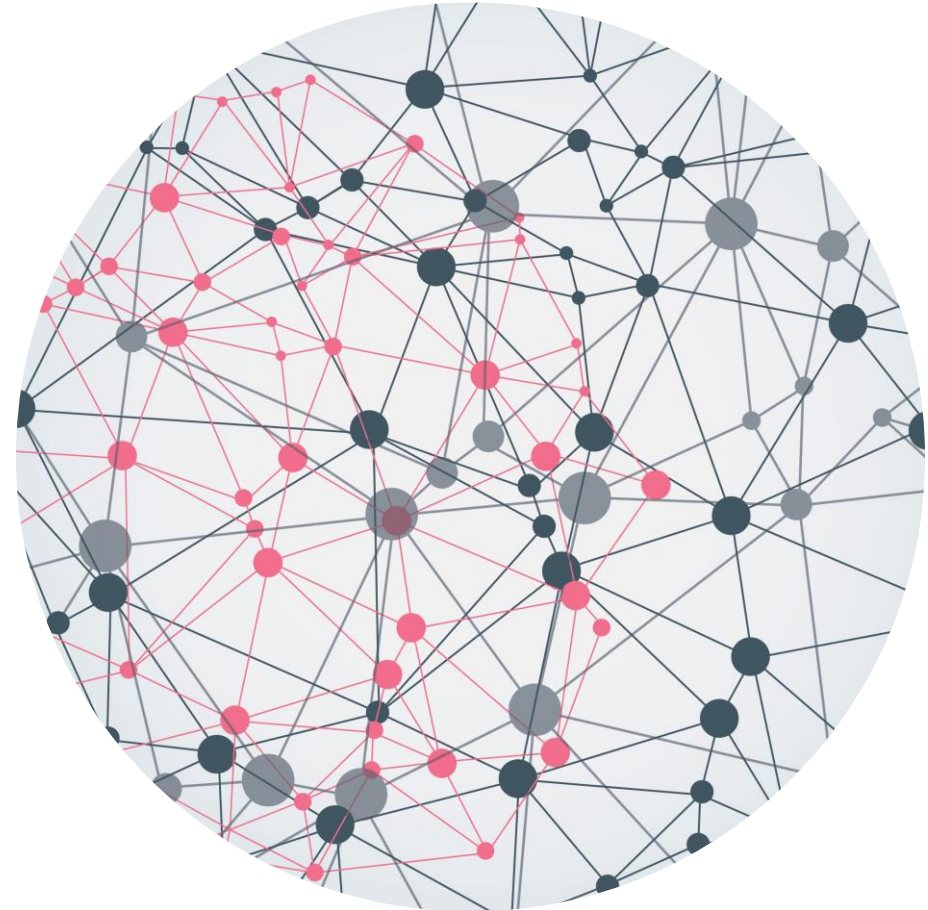


L007 – PROJET « PATRIMOINE »

Semestre P24

MIKOU Ali – LUCAT Enguerrand



INTRODUCTION

MIKOU - LUCAT | / | non-connecté Innovations ▾ Connexion ▾

Accueil

Bienvenue dans notre projet de LO07 intitulé "Patrimoine" ! Il a été réalisé durant le semestre P24.

• Projet LO07 - UTT - Ali MIKOU / Enguerrand LUCAT



Programme imitant les fonctionnalités d'un site de gestion de patrimoine

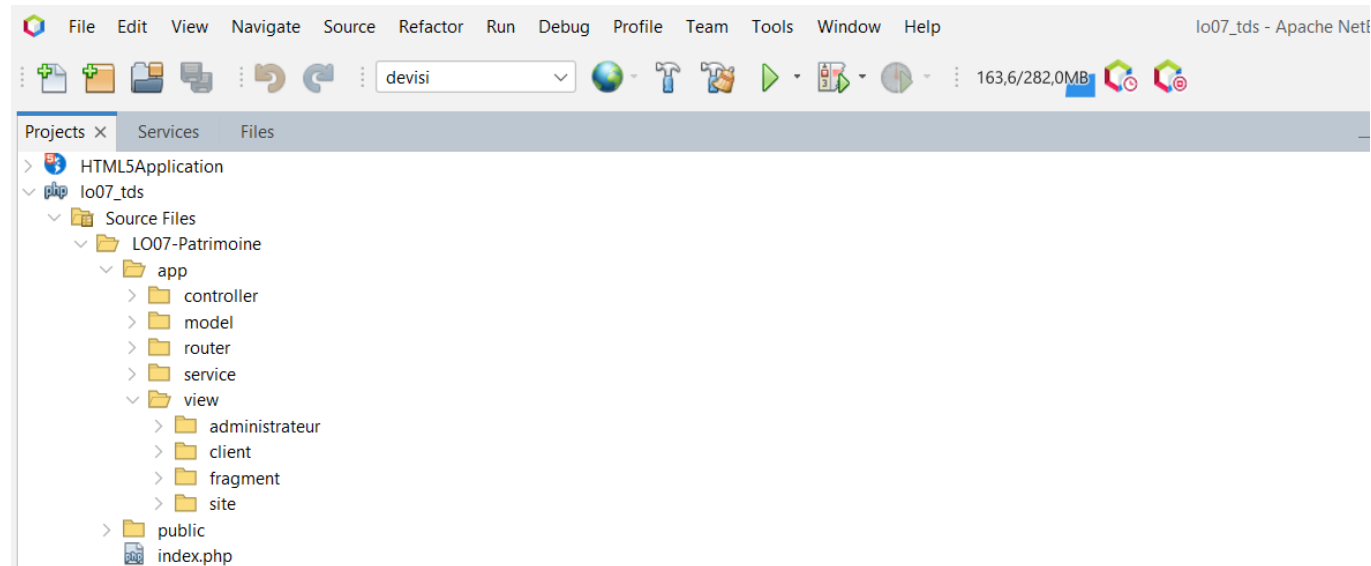


Construction du code en utilisant le modèle MVC



Programme devant assurer la persistance des données

STRUCTURE DU PROJET & SOLUTIONS TECHNIQUES



➡ Structure du code respectant le modèle MVC

➡ Sous dossiers organisant la view en fonction des profils utilisateurs

STRUCTURE DU PROJET & SOLUTIONS TECHNIQUES



Model : Gère l'accès + les interactions avec la base de données + structure les classes de données avec une notion de persistance.

Dans notre code, chaque fichier modèle possède une classe contenant un constructeur et des méthodes qui permettent d'interagir avec la base de données.

```
ControllerClient.php x viewAccueil.php x ModelBanque.php x Model.php x config.php x
History
class ModelBanque{
    private $id, $label, $pays;

    public function __construct($id = NULL, $label = NULL, $pays = NULL){
        if (!is_null($id)){
            $this->id = $id;
            $this->label = $label;
            $this->pays = $pays;
        }
    }

    public static function getAll(){
        try {
            $database = Model::getInstance();
            $query = "select * from banque";
            $statement = $database->prepare($query);
            $statement->execute();
            $results = $statement->fetchAll(PDO::FETCH_CLASS, "ModelBanque");
            return $results;
        } catch (PDOException $e) {
            printf("%s - %s<br/>\n", $e->getCode(), $e->getMessage());
            return NULL;
        }
    }
}
```



View : Affiche les données à l'utilisateur tout au long de sa navigation, permet d'interagir avec celui-ci.

Dans notre code, chaque fichier view possède du code php permettant d'afficher différents éléments : que ce soient des informations ou des formulaires.

```
viewAchatResidence.php x viewChoixResidence.php x ControllerClient.php x viewAccueil.php x ModelBanque.php x Mo
History
<form role="form" method="get" action="router2.php">
<div class="form-group">
    <?php
    echo "<input type='hidden' name='residenceChoisi' value='".$residenceChoisi->getId()."> ";
    <?>

    <input type="hidden" name="action" value="residenceAchete">
    <select class="form-select" name="compteChoisiAcheteur" required id="compteChoisiAcheteur">
        <?php
        foreach ($comptesAcheteur as $element){
            echo "<option value='". $element->getId() ."'> ". $element->getLabel() . "</option> ";
        }
    </select><br>
    <select class="form-select" name="compteChoisiVendeur" required id="compteChoisiVendeur">
        <?php
        foreach ($comptesVendeur as $element){
            echo "<option value='". $element->getId() ."'> ". $element->getLabel() . "</option> ";
        }
    </select><br>
    <?php
    echo "<h4>Prix de la résidence : ".$prix."</h4>";
    echo "<input type='hidden' name='prix' value='".$prix."> ";
    <?>
</div>
</form>
```

STRUCTURE DU PROJET & SOLUTIONS TECHNIQUES



Router : Gère les URL entrantes en les associant aux actions des contrôleurs.

Dans notre code, on retrouve un switch qui permet d'appeler les méthodes dans les controllers.

```
viewAccueil.php x ModelBanque.php x Model.php x
// --- Liste des méthodes autorisées
switch ($action) {

    case "adminReadAll":
    case "clientReadAll":
    case "compteReadAll":
    case "residenceReadAll":
    case "banqueCreate":
    case "banqueCreated":
    case "banqueSelect":
    case "banqueSelected":
    case "banqueReadAll":
        ControllerAdministrateur::$action();
        break;

    case "login":
    case "logout":
    case "amelioration":
    case "fonctionnalite":
    case "register":
    case "createaccount":
        ControllerSite::$action();
        break;

    case "compteCreate":
    case "compteCreated":
    case "transfertInterCompte":
```



Controller : Orchestre le modèle MVC en traitant les actions de l'utilisateur, consultant et modifiant les données du model et en faisant appel aux views.

Dans notre code, chaque fichier Controller est représenté sous forme d'une classe dont les méthodes peuvent accéder aux méthodes des modèles et transmettre les données aux views.

```
ControllerClient.php x viewAccueil.php x ModelBanque.php x Model.php x config.php x Controller
class ControllerClient
{
    public static function patrimoineClient()
    {
        include 'config.php';
        $client = ModelPersonne::getOneById($_SESSION['id'])[0];
        $fullName = " ". $client->getNom(). " ". $client->getPrenom(). " ";
        $results_residence = ModelResidence::getAllById($_SESSION['id']);
        $results_compte = ModelCompte::getByPersonne($_SESSION['id']);
        $total_comptes = array_reduce($results_compte, function($sum, $compte) {
            return $sum + $compte->getMontant();
        }, 0);

        $total_residences = array_reduce($results_residence, function($sum, $residence) {
            return $sum + $residence->getPrix();
        }, 0);

        $total_patrimoine = $total_comptes + $total_residences;
        $vue = $root . '/app/view/client/viewPatrimoine.php';
        if (DEBUG)
            echo ("ControllerAdministrateur : accueil : vue = $vue");
        require($vue);
    }
}
```

INNOVATION PROJET/DATA

MIKOU - LUCAT | / non-connecté | innovations • connexion

Nouvelle Fonctionnalité : Recommandations de résidences

Une utilisation originale des données sert d'entrée un système de recommandations d'achats. Son fonctionnement est le suivant :
À partir d'un conseil client connecté, l'argent disponible sur ses comptes est estimé et, à l'aide du résultat de ce calcul, nous pourrions lui suggérer l'achat d'un ensemble de résidences pertinentes qui correspondent à son budget.

Exemple de mise en place

Pour montrer cette nouvelle fonctionnalité, nous utiliserons par défaut le cas de PRIOR BEATRICE. Cependant, si vous souhaitez vous connecter à un autre compte client, cette fenêtre s'adaptera et affichera le cas du client connecté.

Client actuel : PRIOR Beatrice

Actuellement le client dispose de 670000 euros en cumulé sur ses comptes.

Avec cette somme, notre fonctionnalité permet de proposer à PRIOR Beatrice les résidences suivantes :

id	label	ville	prix	personne_id
101	Appartement Marie Curie	Troyes	90000	1007
101	Appartement Curie de Troyes	Troyes	300000	1002
102	Parking rue Daulnay	Troyes	10000	1002
103	Maison de la rue Voltaire	Troyes	300000	1002
104	Appartement de la rue des 15-20	Troyes	100000	1002
101	Appartement Ambroise Petit	Reims	400000	1003
104	Appartement de la rue Charlevoix	Reims	275000	1003
101	Cabane au fond du jardin	Rhône	1000	1004

Projet L007 - UTT - All MIKOU / Enguerrand LUCAT



Fonctionnalité additionnelle conçue pour améliorer la praticité du site



Système de suggestion d'achat de résidence en fonction des moyens du client connecté

INNOVATION MVC

MIKOU - LUCAT | / | non-connecté Innovations ▾ Connexion ▾

Amélioration du MVC

Durant le développement de ce projet, nous avons utilisé le MVC. Bien que ce soit très utile, des améliorations peuvent lui être ajoutées.

En effet, nous avons remarqué qu'une partie conséquente de la logique dans le code se répète d'un contrôleur à l'autre. De ce fait, nous pensons que ces logiques de code peuvent être extraites et mises sous la forme de fonctions utilisées dans toutes les parties du code qui en ont besoin.

Ces nouvelles fonctions seraient mises dans un document mis à part et auquel tous les autres fichiers auraient accès. En structurant ainsi notre code, toutes les logiques récurrentes sont rassemblées et peuvent être réutilisées sur les différents fichiers du projet.

Nous avons également remarqué que la gestion des erreurs se répète souvent dans le code. Nous pourrions aussi essayer de centraliser cette gestion des erreurs et des exceptions afin d'améliorer la robustesse de l'application.

• Projet LO07 - UTT - Ali MIKOU / Enguerrand LUCAT



Rassembler dans un unique fichier les logiques de code utilisées par plusieurs Controllers



Centraliser de la gestion d'erreurs et d'exceptions pour améliorer la robustesse de l'application

CONCLUSION

Liste des banques vue par l'administrateur

id	label	pays
1	Crédit Agricole	France
2	La Banque Postale	France
3	Caisse d'épargne	France
4	Société Générale	France
5	BNP Paribas	France
6	Le Crédit Lyonnais	France
7	CIC	France
8	Crédit Mutuel	France
11	Wells Fargo Bank	Etats-Unis
12	JPMorgan Chase Bank	Etats-Unis
13	Citibank	Etats-Unis
14	Bank of America	Etats-Unis
15	Atijari Wafa Bank	Maroc
16	Banque Populaire	Maroc
17	Crédit Troyen	France
18	Crédit Lyonnais	France
19	Crédit Parisien	France

Liste des banques enregistrées

Liste des administrateurs

id	nom	prénom	status	login	password
0	Le	Boss	0	boss	secret
1008	Ali	Mikou	0	mikouali	secret
1011	Lucat	Enguemand	0	lucatang	secret

Liste des administrateurs enregistrés

Liste des clients

id	nom	prénom	status	login	password
1001	PRIOR	Beatrice	1	beatrice	secret
1002	EATON	Tobias	1	tobias	secret
1003	MATTHEWS	Jeanine	1	jeanine	secret
1004	PRIOR	Caleb	1	caleb	secret
1007	SWIFT	Taylor	1	taylor	secret
1009	BOND	James	1	MyNameIs	BondJamesBond
1010	Emploi	Paul	1	Jaipes	Dargent

Liste des clients enregistrés

Liste des comptes

id	Titulaire	Personne_id	Banque	Banque_id	Label	Montant
76	BOND James	1009	CIC	7	ArgentDePoche	1000
21	EATON Tobias	1002	CIC	7	Compte chèque de Tobias	1234
12	PRIOR Beatrice	1001	La Banque Postale	2	Livret A	2500
13	PRIOR Beatrice	1001	La Banque Postale	2	Livret DD	3500
74	SWIFT Taylor	1007	Crédit Agricole	1	Mes vacances en jet privé	10000
31	MATTHEWS Jeanine	1003	Citibank	13	Compte chèque de Jeanine	154800
11	PRIOR Beatrice	1001	Crédit Agricole	1	Compte chèque de Béatrice	161000
14	PRIOR Beatrice	1001	La Banque Postale	2	Livret B	503500
71	SWIFT Taylor	1007	Wells Fargo Bank	11	The Eras Tour	9637000
75	BOND James	1009	Bank of America	14	DefinitelyNotCIA	10000000
72	SWIFT Taylor	1007	JPMorgan Chase Bank	12	Red Tour	50003000
73	SWIFT Taylor	1007	Citibank	13	Reputation Stadium Tour	167000000

Liste des comptes enregistrés

Liste des résidences

id	Titulaire	Personne_id	Label	ville	Prix
401	PRIOR Caleb	1004	Cabane au fond du jardin	Rôme	1000
202	EATON Tobias	1002	Parking rue Daumenay	Troyes	10000
101	SWIFT Taylor	1007	Appartement Marie Curie	Troyes	90000
204	EATON Tobias	1002	Appartement de la rue des 15-20	Troyes	100000
201	EATON Tobias	1002	Appartement Curie de Troyes	Troyes	200000
304	MATTHEWS Jeanine	1003	Appartement de la rue Charteraine	Reims	275000
102	PRIOR Beatrice	1001	Maison Promenade des Anglais	Nice	400000
301	MATTHEWS Jeanine	1003	Appartement Ambroise Petit	Reims	400000
203	EATON Tobias	1002	Maison de la rue Voltaire	Troyes	500000
303	MATTHEWS Jeanine	1003	Maison de la rue Toumehonneau	Reims	900000
704	SWIFT Taylor	1007	Bungalow à Los Angeles	Los Angeles	1780000
701	SWIFT Taylor	1007	Appartement Penthouse de Music Row	Nashville	2000000
703	SWIFT Taylor	1007	Northumberland Estate	Nashville	2500000
702	SWIFT Taylor	1007	Domaine de Beverly Hills	Los Angeles	3550000
708	SWIFT Taylor	1007	Maison Cornelia Street	New York	9750000
705	SWIFT Taylor	1007	Mancor Watch Hill Seaside Estate	Rhode Island	17750000
706	SWIFT Taylor	1007	Duples à NY (Loft de Tribeca)	New York	20000000
707	SWIFT Taylor	1007	Mancor de Beverly hills	Beverly Hills	25000000

Liste des résidences enregistrées