

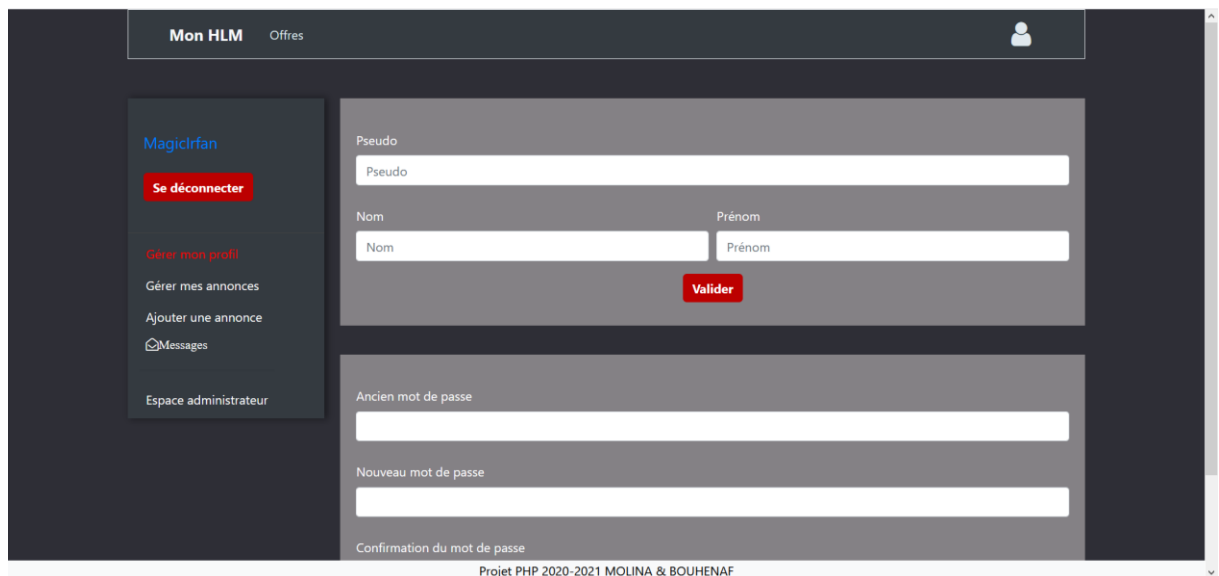
COMPTE-RENDU du projet de PHP (Mon HLM)

I- Introduction et présentation rapide du site

Dans le cadre du cours de M3104, nous devons faire un site d'annonces immobilières alliant le HTML/CSS et le PHP avec un système d'installation mais aussi un Framework afin de respecter l'architecture MVC (Modèle, Vue, Contrôleur). De ce fait, notre site utilise CodeIgniter 4 comme Framework

Pourquoi « Mon HLM ? »

Car nous aimons le chanteur Renaud et il a fait une musique s'intitulant « Mon HLM », nous avons donc décidé de lui rendre hommage.



The screenshot displays the 'Mon HLM' website interface. At the top, there is a navigation bar with 'Mon HLM' and 'Offres' links, and a user profile icon. On the left, a sidebar menu for 'Magiclrfan' includes links for 'Se déconnecter', 'Gérer mon profil', 'Gérer mes annonces', 'Ajouter une annonce', 'Messages', and 'Espace administrateur'. The main content area contains a login form with fields for 'Pseudo' and 'Mot de passe', and a 'Valider' button. Below this is a registration form with fields for 'Ancien mot de passe', 'Nouveau mot de passe', and 'Confirmation du mot de passe'. The footer indicates 'Projet PHP 2020-2021 MOLINA & BOUHENAF'.

L'espace personnel du site

II- Analyse du problème et choix techniques

Pour la réalisation du projet, nous avons dû utiliser l'architecture MVC grâce à un framework de notre choix. Le MVC permet de séparer les données à afficher, la présentation de l'interface graphique et la logique concernant les actions effectuées par l'utilisateur

Dans notre site, il y a 3 modèles : mAnnonces, mUtilisateur, mMessages, ces dernières contiennent les fonctions permettant de manipuler les données, ce sont les contrôleurs qui utilisent ces fonctions et les contrôleurs transfèrent ces données aux vues pour afficher les données grâce à des tableaux associatifs. Les modèles contiennent les requêtes SQL utiles pour le site internet.

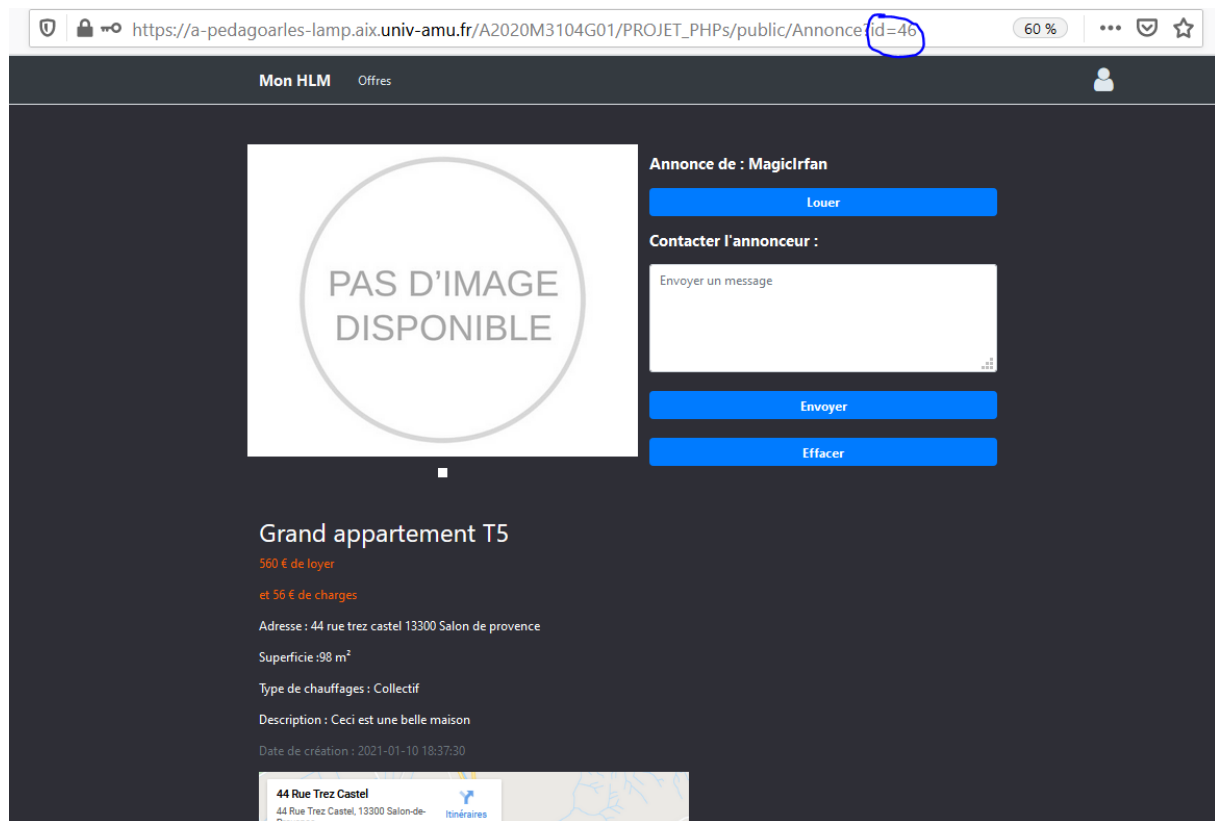
Il y a aussi 4 contrôleurs :

- « Pages » qui affiche les vues (sauf pour afficher les messages d'erreurs) et transfère les données des modèles vers les Vues afin de faciliter les redirections
- « cMessages » qui gère les Messages
- « cAnnonces » qui gère les annonces immobilières
- « cUtilisateurs » qui gère les utilisateurs

Pour afficher les annonces (les 6 dernières annonces par exemple) on trie les annonces par la date de publication et on récupère les 6 premières retournées par la requête SQL qui affiche les annonces et on les affiche grâce à une boucle foreach() dans la page d'accueil

Lorsqu'on clique sur une annonce, on voit ses caractéristiques (localisation, les photos, l'auteur etc...), c'est parce qu'on transfère l'id de l'annonce par l'URL grâce à la superglobale GET et grâce à son id, on peut afficher toutes les caractéristiques de l'annonce, c'est la même chose pour la modification mais aussi la suppression d'une annonce car en transférant l'identifiant de l'annonce, on peut modifier ou supprimer la bonne annonce.

Exemple :



Affichage de l'annonce grâce à son identifiant qui est transféré grâce à la variable GET (ici, id=46)

Aussi, pour savoir si un utilisateur est connecté, on démarre une session ou on reprend une session existante et cela permet de stocker les données de l'utilisateur, par exemple le pseudo ou bien l'email, on détruit la session lorsqu'on se déconnecte du site

Il y a aussi un script Javascript qui permet de cacher ou ajouter des radios boutons lors de l'ajout d'une annonce.

III- Arborescence du site

a) Parcours récursive des répertoires



b) Les pages de notre site

IV- Conception et mise en place de la base de données

Lors de la conception de la base de données, nous devons trouver les clés étrangères des différentes tables grâce aux cardinalités. Initialement, il y a les clés étrangères suivantes :

Dans T_annonce :

- E_id_engie qui référence la table T_Energie ;
- T_type qui référence la table T_typeMaison

Dans T_message :

- A_idannonce qui référence la table T_annonce
- U_mail qui référence la table T_utilisateur

Ce sont des clés primaires-étrangères car T_message est une table d'association

Dans T_photo :

- A_idannonce qui référence la table T_annonce

Nous avons le schéma relationnel suivant :

T_Energie(E_id_engie,E_description) ;

T_annonce(A_idannonce, A_titre, A_cout_loyer, A_cout_charges, A_type_chauffage, A_superficie, A_description, A_adresse, A_ville,A_CP, A_date, A_publiee, #E_id_engie, #T_type,#U_mail) ;

T_typeMaison(T_type, T_description) ;

T_photo(P_idphoto, P_titre, P_nom, #A_idannonce) ;

T_utilisateur(U_mail, U_mdp, U_pseudo, U_nom, U_prenom, U_date_inscription, U_publication) ;

T_message(M_idmessage, #A_idannonce, #U_mail, M_destinataire, M_dateheure_message, M_texte_message, M_lu) ;

Nous avons donc ajouté les colonnes suivantes car cela nous semblait nécessaire :

- Ajout de la clé étrangère U_mail dans la table annonce afin de marquer les annonces de l'auteur.
- Ajout de M_destinataire et de M_lu dans la table afin de bien envoyer les messages au destinataire et savoir si le message a été lu ou non
- Ajout de A_date et A_publiee dans la table T_annonce afin trier les annonces et pour savoir si l'annonce est archivée ou non
- Ajout de U_date_inscription et U_publicaton dans la table T_utilisateur afin de savoir qui est l'administrateur (vu que c'est le premier compte inscrit) et pour savoir aussi si l'utilisateur peut publier ou non les annonces

Au moment de l'installation de la base de données, des valeurs par défaut seront également insérées puisque ce ne sont pas les utilisateurs qui vont les intégrer, ils vont simplement les utiliser :

- Les différents types de maison (T1 jusqu'à T5), accompagnés de leur description.
- Les différents types d'énergies.

V – Choix des améliorations et détails d'implémentation

Pour les deux bonus, nous avons choisi la localisation grâce à Google Maps et la récupération des mots de passes car cela semblait à notre portée.

Localisation :



Pour afficher la localisation grâce à Google Maps, nous avons utilisé la balise `<iframe>` qui représente un contexte de navigation imbriqué qui permet en fait d'obtenir une page HTML intégrée dans la page courante. Dans le paramètre `src` de la balise, nous avons remplacé l'adresse et la ville lors de l'affichage de l'annonce afin de trouver la localisation correcte, si la localisation est fausse, cela affiche la carte du monde.

Balise HTML qui nous sert à afficher la localisation d'un logement :

```
<div class="mapouter">
  <div class="gmap_canvas"><iframe width="550" height="500" id="gmap_canvas"
  src="https://maps.google.com/maps?q=<?=esc($annonce>A_adresse)?>,<?=esc($annonce>A_ville)?>e&t=&z=13&ie=UTF8&iwloc=&output=embed" frameborder="0" scrolling="no" marginheight="0" marginwidth="0"></iframe>
  </div>
</div>
```

Mot de passe oublié :

Nous avons créé une page accessible par un bouton « Mot de passe oublié » dans la page de connexion, composé d'un formulaire avec un seul champ pour rentrer le mail du compte. Si l'email existe dans la BDD, un mail est envoyé vers cet adresse email grâce à la fonction mail() permettant d'envoyer un mail. Lorsque l'utilisateur clique sur le lien de la récupération du mot de passe, cela affiche une fenêtre où l'utilisateur peut entrer le nouveau mot de passe et la confirmation, si c'est bon, l'utilisateur pourra se connecter.

VI – Difficultés rencontrées et solutions éventuelles

Difficulté concernant HTML/CSS :

Nous avons voulu au départ créer un site sans utiliser Bootstrap, avec l'unique utilisation des flexbox pour qu'il soit responsive.

Au bout d'un moment on s'est rendu compte que ça devenait compliqué avec le grand nombre d'éléments, nous avons donc repris avec Bootstrap sans forcément avoir modifié la structure du site, d'où le manque de responsive design sur certaines pages.

Une refonte complète de la structure du site pour un ajout des éléments responsive de Bootstrap pourrait régler ce problème.

Pagination :

La pagination des annonces ne fonctionnait pas, en effet, les annonces se dupliquaient lorsqu'on parcourait les pages, on a donc abandonné l'idée de faire la pagination

CodeIgniter :

Nous avons eu du mal à utiliser CodeIgniter surtout au début car il y avait beaucoup de notions à apprendre en même temps, mais au fil du temps, on s'est plus ou moins habitués à l'utilisation de CI.

L'envoi et l'affichages des messages :

Lorsqu'un utilisateur envoyait un message, le destinataire ne le recevait pas, nous avons donc décidé d'ajouter une colonne M_destinataire pour que le destinataire reçoit les messages.

VII – Conclusion

Ce projet est l'un des plus ambitieux que nous avons pu faire jusqu'à présent, cela s'est ressenti par le contenu du cahier des charges du site, même si nous avons sous-estimé le projet, cela ne nous a pas déplu car nous avons envie de nous orienter dans le développement web