

Manuel d'utilisation du site

Perrier Julien

Collic Johan



Présentation du site

Page d'accueil :

[Accueil](#)[Recherche](#)[Notre Agence](#)[Mon compte](#)

La page d'accueil est constituée d'une banderole défilante de 5 biens mis à la une ainsi que des 10 dernières annonces ajoutées sur le site. Ces images sont cliquables et amènent sur la vue de l'annonce en question

Les pages incluent un menu permettant de naviguer sur les différentes pages du site ainsi que d'un footer affichant les différents contacts de l'agence.

Page Annonce:

Une fois arrivé sur la vue d'une annonce on a accès à toutes les informations la concernant. Si on est logué et admin on peut modifier l'image de l'annonce via cette page.

Page Recherche:

La page recherche affiche toutes les annonces triées par date de publication les plus récente. Il est possible d'affiner la recherche en choisissant le type de recherche, vente ou location, en indiquant un prix min et max ou encore en rentrant un nom de ville.

☒ Vente

☐ Location

Prix Min :

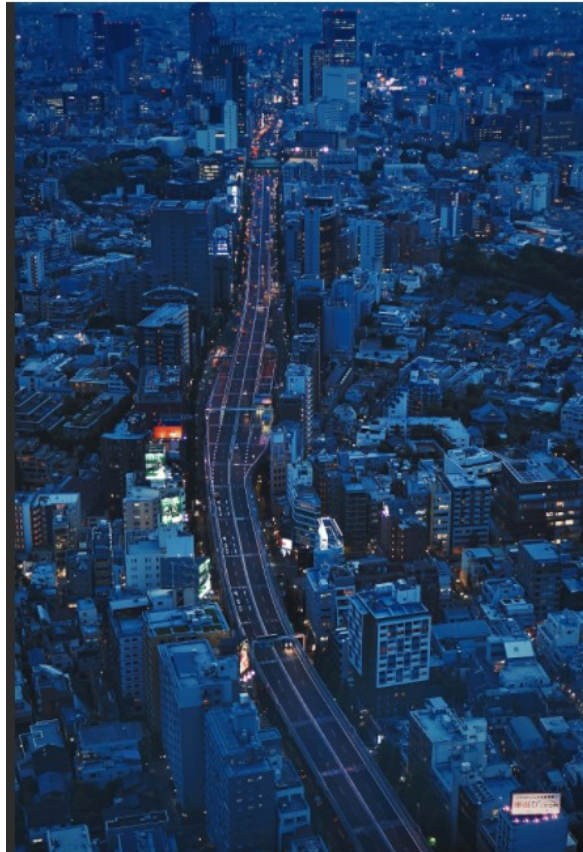
Prix Max :

Ville:

Rechercher

Page Ajout:

Les admins peuvent ajouter des annonces grâce à un formulaire.



AJOUTER UNE ANNONCE

Titre :

Superficie :

prix :

Type de vente :



Vente



Location

Adresse :

ville :

Code postal :

Numéro étage :

Nombre de pièces :

Options :



Meublé



Ascenseur



Terasse



Balcon

Description :

Réinitialiser

Ajouter annonce

Admin:

Le site est sensé pouvoir accueillir deux types d'utilisateur, les clients et les admin. Le rôle d'un utilisateur est définie par le boolean `is_admin` dans la base de données. Le système d'authentification ne fonctionne pas car le pattern mvc n'arriva pas à chargé la classe utilisateur...

Le fichier `util.php` contient les fonctions permettant de déterminer si l'utilisateur est logué et s'il est admin dans le but d'afficher certaines informations ou non mais cela ne fonctionne donc pas (le site s'affiche comme si vous étiez admin)

viewRecherche.php

util.php X

public > util.php > ...

```
1  <?php
2
3  function initPhpSession(): bool
4  {
5      if (!session_id()) {
6          session_start();
7          session_regenerate_id();
8          return true;
9      }
10     return false;
11 }
12
13 function isLoggedIn(): bool
14 {
15     if (isset($_SESSION['username'])) {
16         return true;
17     } else {
18         return false;
19     }
20 }
21
22 function isAdmin(): bool
23 {
24     if (isLoggedIn())
25     {
26         if (isset($_SESSION['isAdmin']) && $_SESSION['isAdmin'] == 1)
27         {
28             return true;
29         }
30         return false;
31     }
32 }
33
34 function clean_php_session(): void
35 {
36     session_unset();
37     session_destroy();
38 }
```

Pattern MVC

Le site est entièrement construit en suivant le pattern « modèle, vue, Contrôleurs ». Le Router charge automatiquement les classes du dossier modèle grâce à la fonction `spl_autoload_register` de php . La fonction `routeReq` détermine le contrôleur à utiliser en fonction des données passées dans l'url.

Modèles :

Le dossier modèle contient toutes les classes correspondant aux tables de la base de données. Les constructeurs des classes utilisent la fonction `hydrate()` afin de maximiser l'encapsulation et ainsi la sécurité du site. La classe `Model` dont tout les managers des classes héritent, effectue la connexion à la bdd et contient les requêtes faisant appel à celle-ci.

Vues:

Chaque page dispose d'une vue propre qui détermine le contenu à afficher ainsi que d'un fichier template qui gère la mise en forme. Tout cela est géré par les fonctions contenues dans le fichier `View.php` appelées à l'aide par les contrôleurs.

Contrôleurs:

Les contrôleurs gèrent les actions et les fichiers vues à générer en fonction des données de l'url. De cette manière on reste toujours sur la même page et c'est seulement l'action qui détermine ce qui va s'afficher. De plus grâce au fichier `htaccess`, on peut masquer les informations que l'utilisateur n'a pas besoin de voir.

FichierEditionSélectionAffichageAtteindreExécuterTerminalAideControllerPost.php - htdocs - Visual Studio Code

EXPLOREUR

HTDOCS

controllers

ControllerAccueil.phpControllerAgence.phpControllerContact.phpControllerLogin.phpControllerPost.phpControllerRecherche.phpRouter.php

models

Appartement.phpAppartementManager.phpContact.phpContactManager.phpImageManager.phpInfo.phpInfoManager.phpModel.phpUser.phpUserManager.php

public

css

footer.cssform.cssloginform.cssmenu.csspost.cssrecherche.cssstyle.css

images

agenceappartements

viewRecherche.phputil.phpviewMenu.php# menu.cssControllerPost.php

controllers > ControllerPost.php > ControllerPost > storeImage

```
1 <?php
2
3 require_once 'views/View.php';
4
5 class ControllerPost
6 {
7     private $_appartementManager;
8     private $_view;
9
10    public function __construct()
11    {
12        if (isset($_url) && count($_url) < 1) {
13            throw new \Exception("Page Introuvable");
14        } else if (isset($_GET['create'])) {
15            $this->create();
16        } else if (isset($_GET['addImage'])) {
17            $this->addImage();
18        } else if (isset($_GET['status']) && $_GET['status'] == "new") {
19            $this->store();
20        } else if (isset($_GET['status']) && $_GET['status'] == "newimg") {
21
22            $this->storeImage();
23        } else {
24            $this->appartement();
25        }
26    }
27
28    //fonction pour afficher un article
29    private function appartement()
30    {
31        if (isset($_GET['id'])) {
32            $this->_appartementManager = new AppartementManager;
33            $appartement = $this->_appartementManager->getAppartement($_GET['id']);
34            $this->_view = new View('SinglePost');
35            $this->_view->generatePost(array('appartement' => $appartement));
36        }
37    }
38 }
```

Ce qu'il manque :

Le but était que le site soit déployable pour n'importe quelle agence, on aurait du pouvoir changer le logos, les descriptions , les liens des différents réseaux sociaux de l'agence etc, directement depuis le site en tant qu'admin mais chaleureusement le site ne semble pas réussir à charger les classes nécessaires.

```
hydrate($data); } // HYDRATION public function hydrate(array $data) { foreach ($data as $key => $value) { $method = 'set' . ucfirst($key); if (method_exists($this, $method)) { $this->$method($value); } } } // GETTERS public function getId() { return $this->_id; } public function getTailleImg() { return $this->_taille_img; } public function getNomImg() { return $this->_nom_img; } public function getDescription() { return $this->_description; } //SETTERS public function setId($id) { $id = (int) $id; $this->_id = $id; } public function setTaille_img($taille) { $taille = (int) $taille; $this->_taille_img = $taille; } public function setNom_img($nom) { if(is_string($nom)) $this->_nom_img = $nom; } public function setPrenom($description) { if(is_string($description)) $this->_description = $description; } public function enregistrer() { global $bdd; $req = $bdd->prepare("INSERT INTO infos VALUES (null, :taille, :nom, :description)"); $data = [ ':taille' => $this->getTailleImg(), ':nom' => $this->getNomImg(), ':description' => $this->getDescription(), ]; $insertOk = $req->execute($data); $sql="SELECT id as LastID from infos where id_annonce = @@Identity:"; // @@Identity permet de recuperer la dernière valeur entrée $stmt=$bdd->query($sql); while ($idInfo = $stmt->fetch(PDO::FETCH_OBJ)) { $this->setId($idInfo->LastID); } }  
Fatal error: Uncaught Error: Class 'Info' not found in C:\xampp\htdocs\models\Model.php:49 Stack trace: #0 C:\xampp\htdocs\models\InfoManager.php(7): Model->getAll2('infos', 'Info') #1 C:\xampp\htdocs\controllers\ControllerAgence.php(22): InfoManager->getInfos() #2 C:\xampp\htdocs\controllers\ControllerAgence.php(15): ControllerAgence->infos() #3 C:\xampp\htdocs\controllers\Router.php(41): ControllerAgence->__construct(Array) #4 C:\xampp\htdocs\index.php(7): Router->routeReq() #5 {main} thrown in C:\xampp\htdocs\models\Model.php on line 49
```

La page login ne fonctionnant pas, l'utilisation de sessions est impossible, il y aurait du avoir un système de panier ou il aurait été possible de consulter les annonces passées en favoris par un client et aussi d'imprimer ces annonces en pdf.