Jacot-dit-Montandon Ludovic, Loïc Burnand

M306 - Annonces

Documentation

TABLE DES MATIERES

[INTRODUCTION 2](#_Toc3808911)

[La maquette 2](#_Toc3808912)

[ANALYSE FONCTIONNELLE 3](#_Toc3808913)

[FONCTIONNELITES DISPONIBLE 3](#_Toc3808914)

[En tant qu’utilisateur 3](#_Toc3808915)

[DESCRIPTION DE L’INTERFACE ET DE SON FONCTIONNEMENT 3](#_Toc3808916)

[Barre de navigation 3](#_Toc3808917)

[Le menu des catégories 3](#_Toc3808918)

[Page « index.php » 4](#_Toc3808919)

[Formulaire « Ajouter une annonce » 5](#_Toc3808920)

[Formulaire « Se connecter » 6](#_Toc3808921)

[ANALYSE ORGANIQUE 7](#_Toc3808922)

[ARCHITECTURE DU PROJET 7](#_Toc3808923)

[DESCRIPTION DES FONCTIONS 7](#_Toc3808924)

[Fonction « formLogin » 7](#_Toc3808925)

[Fonction « connectDB » 7](#_Toc3808926)

[Fonction « getAllAnonces » 7](#_Toc3808927)

[Fonction « getAllTypes » 7](#_Toc3808928)

[Fonction « getAllUsers » 7](#_Toc3808929)

[Fonction « getTypes » 7](#_Toc3808930)

[Fonction « getTypesById » 7](#_Toc3808931)

[Fonction « getAnnonces » 8](#_Toc3808932)

[Fonction « getAnnoncesByType » 8](#_Toc3808933)

[Fonction « insertPost » 8](#_Toc3808934)

[Fonction « userExist » 8](#_Toc3808935)

[Fonction « getUserId » 8](#_Toc3808936)

# INTRODUCTION

Le projet consiste à créer un site Internet qui permet aux personnes qui l’utilisent de consulter des annonces de tous type et d’un ajouter des nouvelles via un formulaire.

L’utilisateur qui possède un compte sur le site pourra, après s’y etre connecté :

* Consulter une annonce
* Avoir accès aux annonces triées par leur type
* Ajouter une nouvelle annonce

Pour créer ce site, nous avons utilisé un template Bootstrap pour rendre le site adapté à l’écran de l’utilisateur, que ce soit sur mobile, sur un ordinateur, sur une tablette, etc. Il est nommé « Large » et disponible à l’adresse suivante : <https://getbootstrap.com/docs/4.2/examples/blog/>

Ce document permet de mieux comprendre le fonctionnement du site, et comment il a été développé.

## La maquette

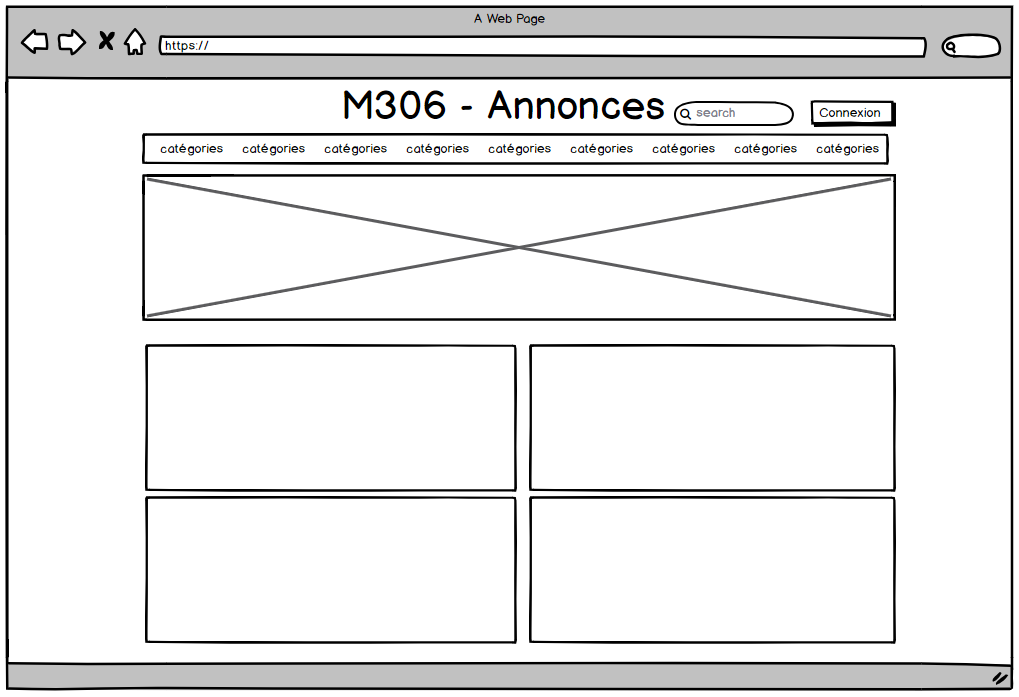


Figure 1: La maquette

Nous avons imaginé en premier le lieu le site de cette façon.

# ANALYSE FONCTIONNELLE

Cette partie de la documentation de la documentation technique explique ce que le projet doit pouvoir faire.

## FONCTIONNELITES DISPONIBLE

### En tant qu’utilisateur

* Se connecter au site
* Consulter les différentes annonces, trié par type
* Créer de nouvelles annonces

## DESCRIPTION DE L’INTERFACE ET DE SON FONCTIONNEMENT

Cette partie de la documentation permettra d’expliquer les différentes pages du site.

### Barre de navigation

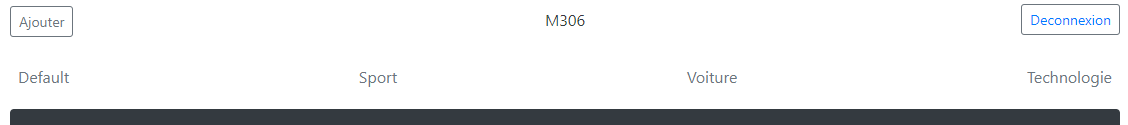


Lorsque l’utilisateur n’est pas connecté, la barre de navigation s’affiche comme ceci. Elle permet à l’utilisateur de se connecter ou consulter les annonces par type.



Lorsque qu’un utilisateur est connecté, la barre de navigation s’affiche de cette manière. Elle permet à l’utilisateur d’ajouter une annonce, de consulter les annonces par type ou de se déconnecter.

### Le menu des catégories



En dessous de la barre de navigation se trouve les différentes catégories qui trient les annonces par type.

### Page « index.php »

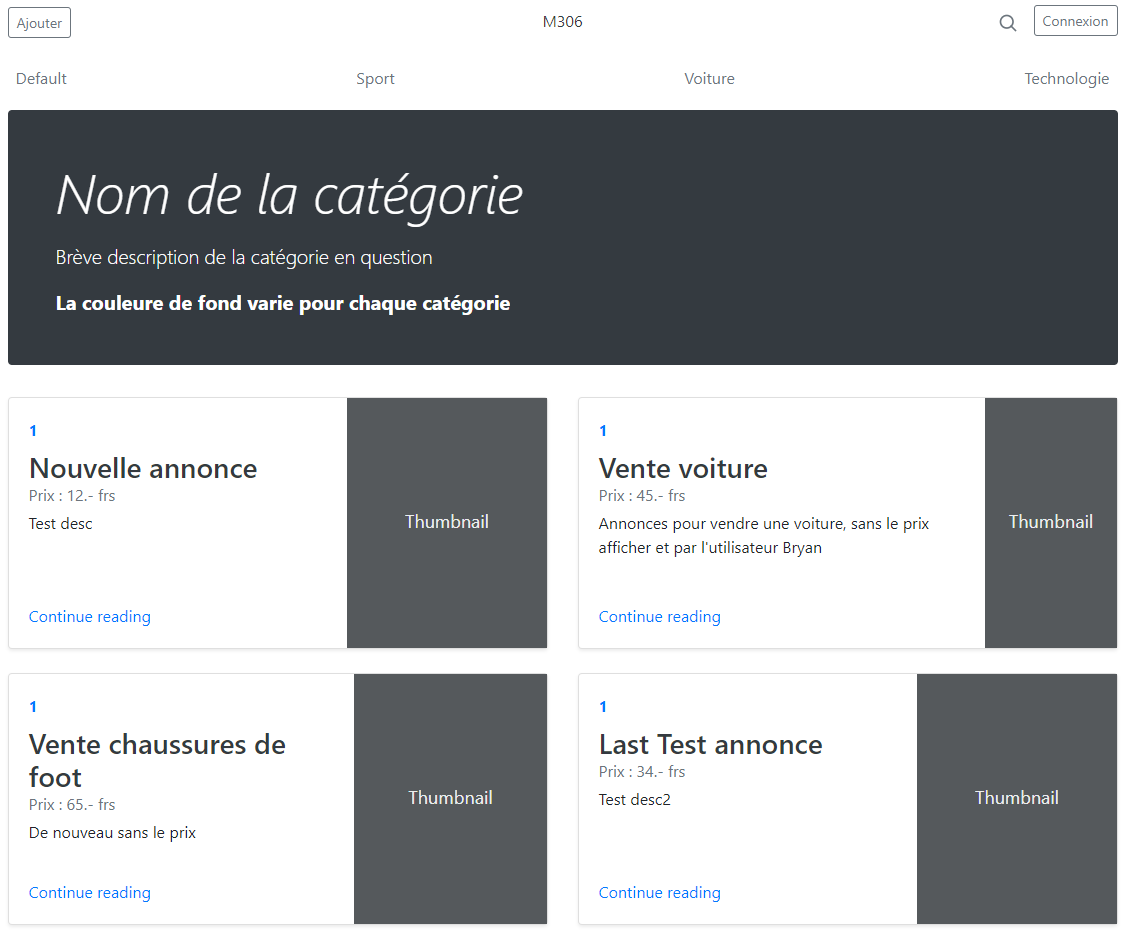


Figure 2: index.php

Cette page affiche toute les annonces en fonction du type en paramètre. Il possible de cliquer sur chacune d’elle pour y découvrir plus d’information.

### Consulter une annonce

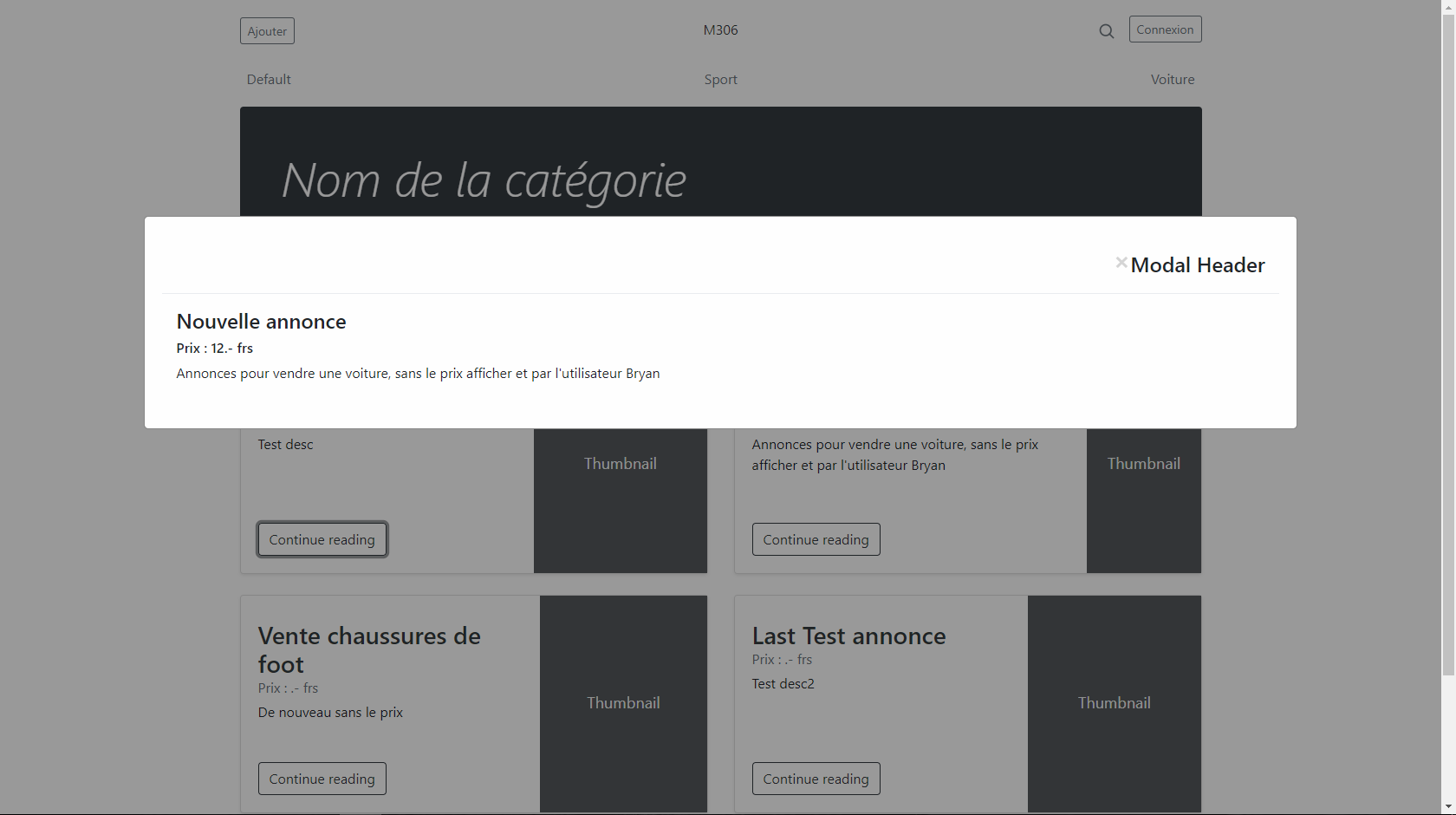


Figure : Consuler une annonce

Le bouton pour consulter une annonce ouvre une pop-up avec les informations que celle-ci contient.   
Le titre, le prix et la description sont affichés.

### Formulaire « Ajouter une annonce »

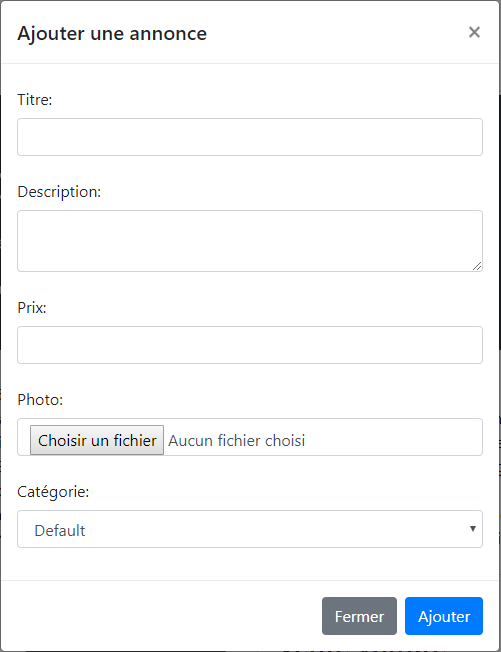


Figure 4: Formulaire pour ajouter une annonce

Le formulaire est composé dans total de 5 champs : le titre, la description, le prix, l’image et le type. Ceux-ci sont tous indispensable. Le champs « titre » est de type texte, le champs « description » est de type texte, le champ prix est de type numéro, le champs « image » est de type file et le champs type est un ascenseur provenant de la base. Lors de l’appui sur le bouton « Ajouter », la page se recharge et récupère les différentes valeurs avec un filtre adéquat.

### Formulaire « Se connecter »

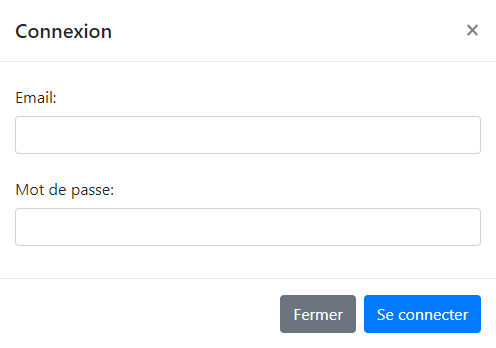


Figure 5: Formulaire de connexion

Le formulaire présente 2 champs : l’email et le mot de passe. Le champs « email » est de type email et le champs « password » de type password. Lors de l’appui sur le bouton « Connexion », une vérification avec la base de données est effectuée et l’utilisateur est connecté en cas de succès.

# ANALYSE ORGANIQUE

## ARCHITECTURE DU PROJET

Les fichiers du site sont séparés entre plusieurs dossiers :

* Doc : ce dossier contient toute les document Word tel que le mandat et la documentation ainsi que les images et capture présent dans cette dernière.
* css : ce dossier contient les fichier Bootstrap et le fichier style.css
* A la racine du site, il y a toutes les différentes pages du site qui sont atteignable par les personnes naviguant sur le site.

## DESCRIPTION DES FONCTIONS

### Fonction « formLogin »

Cette fonction est déclarée dans le fichier « phptohtml.php » et sert à afficher le formulaire de login. Elle ne reçois aucun paramètres. Cette fonction echo du code html.

### Fonction « connectDB »

Cette fonction, comme toutes les suivantes, est déclarée dans le fichier « bdd.php » et sert à établir la connexion à la base de données. Elle ne reçoit pas de paramètre, mais utilise cinq constantes déclarée dans le même fichier. Ces constantes sont « SERVER » pour l’hote du serveur, « DB\_NAME » pour le nom de la base de données, « PSEUDO » pour l’utilisateur qui se connecte, « PWD » pour le mot de passe utilisé lors de la connexion et « PORT » pour le port utilisé. Cette fonction retourne une intance de l’objet PDO.

### Fonction « getAllAnonces »

Cette fonction sert à récupérer toutes les annonces sans tri. Elle ne reçoit pas de paramètre et revoie un tableau contenant des objets Annonces.

### Fonction « getAllTypes »

Cette fonction récupère tous les type. Elle ne reçoit pas de paramètre et renvoie un tableau contenant des objets Type.

### Fonction « getAllUsers »

Cette fonction récupère l'id, le pseudo et l'email de tous les utilisateurs. Elle ne reçoit pas de paramètre et renvoie un tableau contenant des objets User.

### Fonction « getTypes »

Cette fonction récupère l’id et le nom des types. Elle ne reçoit pas de paramètre et renvoie un tableau contenant des objet Types.

### Fonction « getTypesById »

Cette fonction récupère le nom des types et leurs descriptions. Elle reçoit l’id en paramètre et renvoie les informations de la base.

### Fonction « getAnnonces »

Cette fonction récupère le nom, la description, le nom de l’image, le type d’image, le prix, l’id utilisateur et l’id type. Elle ne reçoit pas de paramètre et revoie un tableau des objets Annonces.

### Fonction « getAnnoncesByType »

Cette fonction récupère le nom, la description, le nom de l’image, le type d’image, le prix, l’id utilisateur. Elle reçoit paramètre l’id du type et revoie un tableau des objets Annonces.

### Fonction « insertPost »

Cette fonction ajoute une annonce. Elle reçoit en paramètre le nom, la description, le prix, l’id de l’utilisateur et l’id du type et insert dans la base de données ce qu’ils contiennent.

### Fonction « userExist »

Cette fonction vérifie si l’utilisateur existe dans la base. Elle reçoit en paramètre l’email et le mot de passe en sha1 puis vérifie dans la base de données.

### Fonction « getUserId »

Cette fonction récupère l’id de l’utilisateur. Elle reçoit en paramètre l’email et revoie l’id correspondant.

# CONCLUSION

Le fait de réaliser ce projet nous a permis de nous rendre compte que réaliser l’intégralité d’un projet informatique n’est pas chose aisée et plus complexe que ce que nous pensions. En effet, en plus de la production du code, il y a la documentation technique qu’il faut produire, et tout ceci prend un temps considérable.

Néanmoins, nous avons beaucoup apprécié faire ce projet, et cela nous a été très instructif. Nous avons pu créer un projet entier, avec la production du code et la documentation nécessaire.

## COMPARAISON ENTRE LE TRAVAIL ACCOMPLI ET LES OBJECTIFS

Si l’on compare le travail effectué avec les objectifs prévus, on peut constater que nous avons respecté ce dernier dans l’ensemble et que toutes les fonctionnalités énumérées sont disponibles et fonctionnelles.

## POINTS POSITIFS

Nous sommes très contents du travail que nous avons réussi à fournir. Nous trouvons que le code que nous avons produit est propre et respecte les règles de codage qui nous ont été enseignées au CFPT. L’interface est aussi facile à utiliser.

## POINT NEGATIFS

Un camarade a malheureusement abandonné la formation au cours du projet et nous avons alors dû nous répartir le travail de façon optimal pour ne pas retarder tout le projet.