

# **PROJET WEB**

## **LES BONS BAILS**

**Victor MEAS ,Jean-Baptiste SKUTNIK, Hugo TRACHINO,  
Yassir CHEKOUR**



# Sommaire

## **Partie 1 : Approche et contexte**

Introduction .....	p.1
Répartition des tâches .....	p.3

## **Partie 2 : L'implémentation**

Explication Générale .....	p.4
Dépendance des modules.....	p.7
Problèmes rencontrés .....	p.9
Contraintes respectées .....	p.10

<b>Conclusion</b> .....	p.11
-------------------------	------

# Partie 1 - Approche et contexte

## 1- Introduction



Ce projet Web nous offrait beaucoup de liberté vis à vis du sujet à choisir. Nous ne voulions pas nous orienter dans le sens des sujets déjà établis par l'équipe pédagogique, pensant que l'intégration des OAs, l'agenda numérique et la communication entre élèves (Sujet twitter proposé) étaient déjà assurés par les serveurs IRC, iiens.net et les chats Discord de l'école respectivement. Nous avons donc décidé de nous lancer dans l'idée d'un site unique, mais toujours dans l'optique d'aider les élèves.

Le but de ce projet est donc de modéliser et réaliser un site nommé "Les Bons Bails". Similaire à d'autres sites comme "Le Bon Coin" et "Craigslist", il mettra en lien les élèves de l'ENSIIE par moyen d'annonces (d'offres ou de demandes) mais cependant dans l'optique de partager des corrigés de partiels, des TD, des cours, etc.

Le site aurait pour but alors d'engager les élèves à publier des annonces aux informations diverses tel que le sujet, le module, le semestre, la date ou encore le choix de rémunération (PAY: payant, SWAP: échange/service, FREE: pro bono). D'autres élèves pourraient alors rechercher et visualiser ces annonces en fonction de leurs propres intérêts ou disponibilités, et choisir de contacter les publieurs d'annonces en allant chercher l'information de contact sur leur profil (numéro de téléphone, facebook, email, etc.).

Nous n'aurons alors aucune responsabilité quant au copiage qui pourrait advenir lors de ces échanges car nous n'aurions que pour rôle de faciliter un premier contact. Toute tricherie serait alors à la discrétion des élèves.

Ainsi, il s'agissait dans ce projet de mettre en place les différents modules, fonctions et B.D. permettant de créer un site fonctionnel.

Il s'agirait ainsi de séparer le travail en trois axes régisseurs:

- Les pages web présentant le site, les annonces et les utilisateurs (Vue)
- Les formulaires pour l'authentification, la modification d'informations personnelles et la recherche/création d'annonces
- Les fonctions pour interagir avec la base de donnée et récupérer les informations imparties par les utilisateurs (Modèle et Contrôleur)

On rappelle que le sujet proposera au minimum:

- Une authentification
- Un compte administrateur donnant les droits à certaines fonctionnalités (au choix)
- Un profil utilisateur éditabile
- Une base de données relationnelle :
  - au moins 3 tables
  - au moins une table de jointure ( n...n)
  - au moins une jointure dans une requête
- des INSERT, DELETE, UPDATE, SELECT
- Un CRUD (Create Read Update Delete)

L'explication de ces différentes tâches et de leurs codes se trouveront dans la deuxième partie.

L'intégralité de ce projet se fera en PHP (intégration HTML, CSS, etc.), PosteGreSQL et PDO, en s'appuyant sur le principe du responsive WebDesign et en sollicitant des bases de données. Notre code devra être présentable et accomplir les tâches qu'il incombe.

Les dépôts de code et notre progrès pourront être retrouvés sur le Gitlab de notre groupe.

## 2 - Répartition des tâches

Le site comportant des tâches variées et parfois interdépendantes, il était nécessaire de réunir certaines d'entre elles lors de notre travail. Yassir et Victor (Front end) se sont focalisés sur l'apparence du site et le layout général tandis que Hugo et Jean-Baptiste se sont attaqués au fond du site (Back end). Ceci a donné lieu à l'organisation suivante:

J-B Skutnik (Back end )	Hugo Trachino (Back end)	Victor Meas (Front end)	Yassir Chekour (Front end)
Authentification de l'utilisateur ou création de compte (Sign Up)	Formulaire de recherches d'annonces	Rapport de projet	Fonctions d'affichage de header
Création de Base de Donnée, implementation	Création du main et fonctions de vues de base	Formulaire de création d'annonces et insertion des annonces dans la BD	Page d'index au début du site

des tags			
Page admin	Sidebar	Fonction d’affichage de footer, de page d’erreur, compléments de CSS	Pages a propos, contact

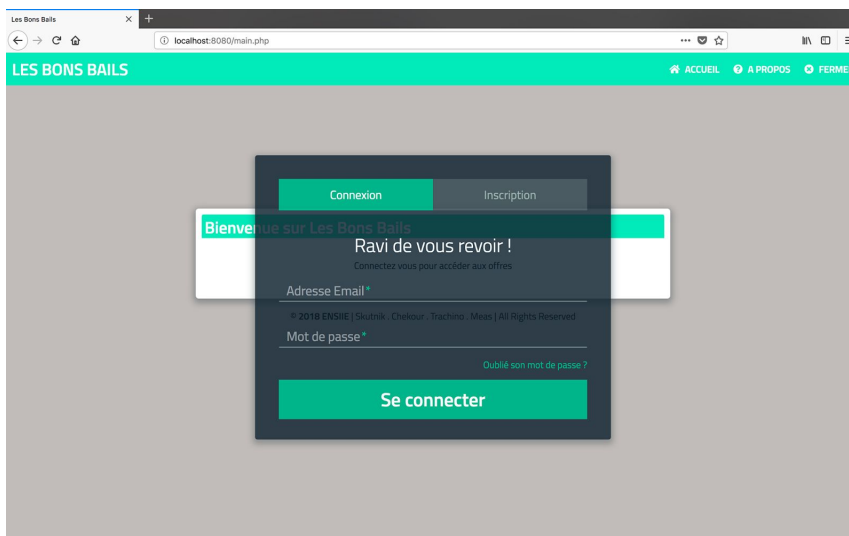
## Partie 2 - L’implémentation

### 1 - Explication Générale et détails du code

#### a) Répartition et organisation du site

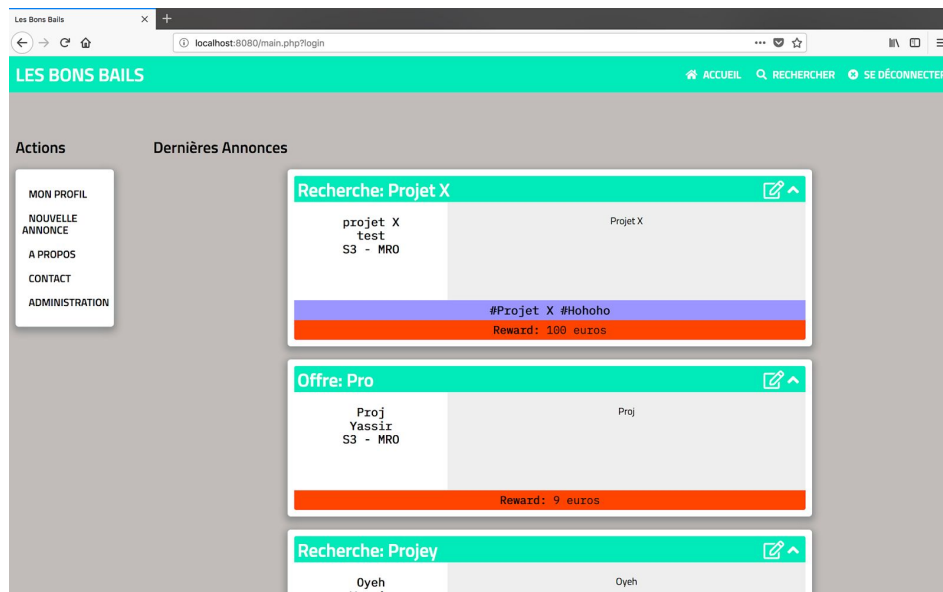
Le site sera segmenté en 8 pages, dont une pour l'espace perso/administrateur, une qui agit comme index/accueil pour naviguer les annonces existantes, une pour rechercher, une pour créer une annonce de son choix, les 3 dernières sont pour les sections “A propos” “Contact” et une en cas d’erreur de recherche. Ces différentes pages pourront tous être accessibles par le header et la sidebar omniprésent sur la page, suite à l’authentification au départ (le header comportera une option “Déconnecter” également pour retourner au Login).

#### Login/Sign Up



Apparition d’une page initiale de présentation (index.php). Page de formulaires situées dans src/login/login.php et createUser.php, utilise les fonctions d’accessdb.php, apparait en module au dessus de l’accueil (main.php)

#### Accueil

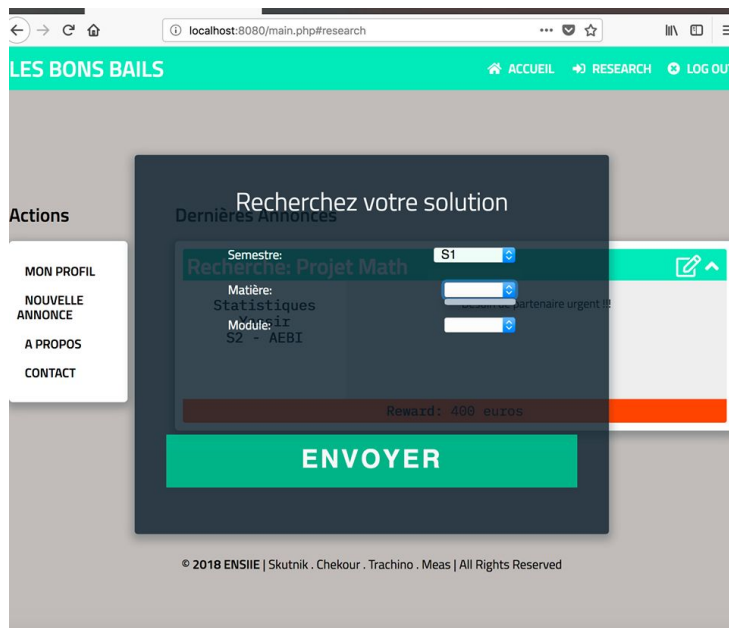


Main.php, comporte l’affichage des annonces, utilise viewfunctions.php à ce fin pour aussi afficher le header et le footer, ainsi que sidebar.php pour afficher les actions à effectuer en permanence. On y trouve les modules de pages pour l’espace perso et administrateur.

### Annonces création

Formulaire faisant appel à createform.php et renvoi vers la cible edit.php/create.php

### Annonces recherche



Formulaire faisant appel à form.html et renvoi vers la cible research.php

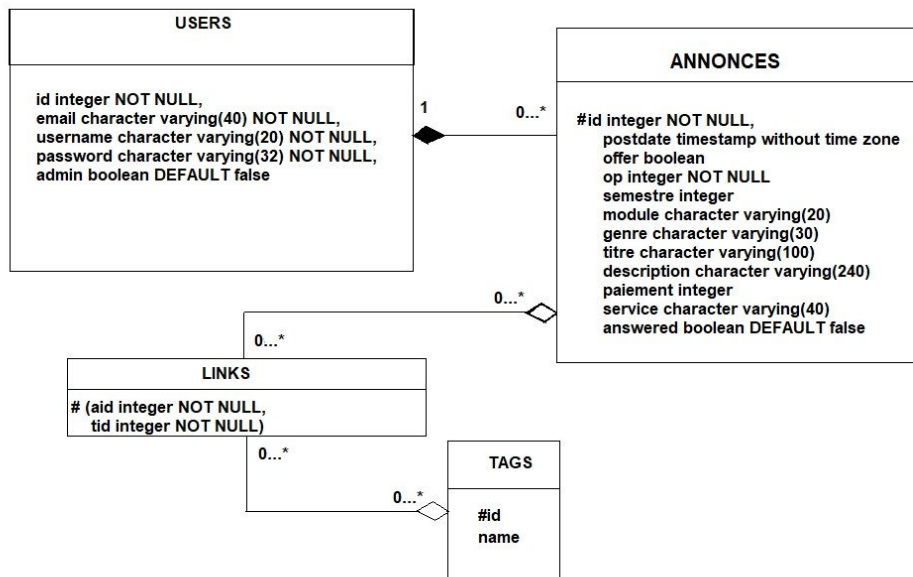
### A Propos/Contact/Erreur

Évident,comporte le layout de base avec des simples paragraphes et liens (apropos.php, contact.php, nondroit.php).

### Espaces utilisateurs et administrateurs

Récupère les données de l'utilisateur/administrateur et grâce aux formulaires respectifs, permettent de modifier des informations, supprimer des annonces, etc.

## b) La base de donnée



Notre base de donnée (data/db.sql) sera séparée en 4 tables:

1. users(#id,email,username,password,admin)
2. annonce(#id,postdate,offer,op,semestre,module,genre,titre,description,paiement,service, answered)
3. tags(#id,name)
4. lien(#(tid=>tags),(aid=>annonce))

Chacune permet de récupérer les informations pertinentes vis à vis des utilisateurs (clé id, email obligatoirement @ensiie.fr, username unique, mot de passe, et booléen admin pour savoir s'il s'agit du compte administrateur ou non) et des annonces.

Le système de tags fonctionne similairement à ce qui avait été demandé dans le sujet de twitter, et permet de filtrer les annonces lors de leurs recherches et publications, précisant des choses supplémentaires (ex: #urgent, #pasdecopiagesvp,etc.). La table lien comme elle est nommée, sert à mettre en lien les annonces et les tags correspondantes lors des recherches et filtrages, par le biais d'une jointure sur les clefs étrangères ci-dessus

## c) Le code des pages, formulaires et fonctions

### **Modèle**

Regroupe des fichiers variés dans public et src, cette partie est moins bien définie que les deux autres car ses éléments peuvent se trouver dans des fichiers d’affichage (vue) ou de traitement de données (contrôleur) mais regroupe en général les actions effectuées pour récupérer des informations suite à un formulaire (ex: recherche d’annonces, visualisation de la page perso. , etc.)

### **Vue**

Tout l’aspect affichage du site se retrouve dans le répertoire /public et regroupe à la fois les pages (contact.php, apropos.php, etc.) et regroupe également les fichiers CSS, JS, SCSS et les images pour peaufiner la présentation du site. La majorité de ces fichiers sont simples et font largement appels au modules des parties modèle et contrôleur, notamment viewfunctions.php qui regroupe toutes les fonctions d’affichage (ex: footer(), header(titre), etc.) . On peut y trouver la présentation des formulaires aussi.

### **Contrôleur**

Cette partie du site se trouve principalement dans /src et englobe les fichier PHP cibles des formulaires (annonce.php, form.html) ainsi que accessdb.php. Accessdb.h est le module principal d’accès et de communication à la base de donnée, utilisant des fonctions de PostGreSQL et PDO (PHP Data Objects) pour faciliter la tâche d'appeler et récupérer des informations de la BD.



## 2 - Interdépendance des modules

Comme nous l'avons énoncé précédemment, nos codes, séparés en différents modules présentent une forme d'interdépendance en lien avec les fonctions et structures dont ils nécessitent.

Module	Dépendance	Explication
/src/accessdb.php	'../vendor/autoload.php'	Pas beaucoup de dépendance, module de base pour accéder à la BD.
/src/annonce.php	'../src/accessdb.php'	Pour faire appel à la BD car il s'agit des fonctions du formulaire de création d'annonce createForm.php et create/edit.php
/src/config.php	aucune	Initialisation de variables pour la BD
/src/form.html	aucune	formulaire de recherche d'annonces
/src/main.php	include("annonce.php"); include("viewfunctions.php"); include("sidebar.php");	le main.php dans public est le même, fait appel aux fonctions de Vue, et à annonce.php pour récupérer les annonces récentes
/src/sidebar.php	aucune	Affiche la sidebar à gauche de la page avec des liens pour renvoyer vers les pages d'actions conséquentes.
/src/viewfunctions.php	include ("login.php");	Présente les fonctions de vue, nécessite de savoir si l'utilisateur est connecté pour certaines fonctions d'affichage (notamment pour l'affichage des annonces ou non, car le login est un overlay sur l'accueil)
/src/login.php et	require 'accessdb.php';	Cible des formulaires de login

createUser.php		et signup respectivement, besoin d'insérer les infos utilisateurs dans la BD
/src/tags.php et user.php	aucune	Fonctions pour récupérer les informations sur les utilisateurs et tags. On utilise directement les commandes PostgreSQL
/public/create.php	include("../src/accessdb.php"); include("../src/annonce.php");	Appelé après le formulaire creation annonce (createForm) et est donc dépendant de la cible annonce.php
/public/createForm.php	include("../src/annonce.php"); include("../src/sidebar.php"); include("../src/viewfunctions.php");	Viewfunctions pour l'affichage du site (idem pour sidebar) annonce.php car il s'agit de la cible de ce formulaire de création d'annonces.
/public/edit.php	include("../src/accessdb.php"); include("../src/annonce.php");	Appelé pour modifier une annonce, et donc nécessite l'accès aux informations de l'annonce et à la BD
/public/perso.php	include("../src/viewfunctions.php"); include("../src/annonce.php"); include("../src/sidebar.php"); include("../src/user.php");	Affichage de toutes les infos de l'utilisateur d'où user.php ainsi que ses annonces d'où annonce.php
/public/recherche.php	include("../src/annonce.php"); include("../src/viewfunctions.php"); include("../src/sidebar.php");	Va chercher les informations sur les annonces, donc pas besoin d'accès à la BD, seulement aux informations des formulaires.
/public/nondroit.php, contact.php, index.php et apropos.php	include("../src/viewfunctions.php"); include("../src/sidebar.php");	Uniquement les fonctions d'affichage.
/public/editUser.php	include("../src/accessdb.php"); include("../src/user.php");	Appelé pour modifier les informations de l'utilisateur, et donc nécessite l'accès à la BD et aux informations récupérées par le formulaire
/src/login.html,	Aucune	Modules d'affichage de

userForm.php createForm.php	et		formulaire de login/SignUp/création d'annonces (cibles: main.php?login ou createUser,editUser.php et edit.php/create.php)
admin.php		include("../src/viewfunctions.php"); include("../src/annonce.php"); include("../src/sidebar.php"); include("../src/user.php");	Affichage de toutes les annonces et des utilisateurs.

### 3- Problèmes rencontrés



**-Redéfinition de la base de donnée** Initialement, la table d'annonces et d'utilisateurs nous suffisait, mais nous avons remarqués qu'il nécessitait au moins 3 tables. Nous avons donc décidés d'utiliser un système de tags qui s'est montré utile en terme de filtrage d'annonce. La difficulté était de trouver comment définir les tables suivantes pour ne pas enfreindre l'ordre de notre base de donnée existante.

#### **Conflits de CSS et problèmes d'affichage**

Travailler en groupe peut poser problème lorsque chacun veut donner une apparence unique à sa page, il était au départ difficile de se mettre d'accord sur une apparence unifiée pour le site, et fut l'objet de nombreuses discussions, corrections et engueulades. A ceci se rajoutait les fichiers CSS/SCSS qui pouvait présenter des conflits, en oubliant par inadvertance de les définir dans des classes ou des div/span.

### 4 - Contraintes respectées

#### **Authentification**

Le site demande bien une authentification à son entrée et assure la confidentialité des données par encodage.

#### **Profil Utilisateur**

Le header présente un onglet (lien) pour accéder à l'espace personnel de l'utilisateur (sidebar), lui permettant de modifier les données de son compte (username, mot de passe entre autres).

### **Compte Administrateur**

Nous avons implémenté un compte administrateur qui peut accéder à "l'espace administrateur" afin de visualiser toutes les annonces/utilisateurs et les modifier, ainsi que de donner le titre d'administrateur à d'autres utilisateurs.

### **Base de donnée: jointures et CRUD**

Par le biais du système de tags (cf. partie Base de Données) notre site respecte la consigne selon quoi il fallait avoir des jointures dans nos requêtes. Nous respectons aussi les directives de CRUD, car notre site propose les mesures et fonctionnalités adéquates pour créer (sign up et création d'annonces), lire (affichage d'annonces), modifier (espaces personnel d'utilisateurs) et supprimer (option de supprimer compte et annonces) les données de notre base (informations utilisateurs et d'annonces).

## **Conclusion**

En conclusion, nous avons pu finir le projet dans son intégralité. L'organisation de départ et l'établissement des modules nous semblait initialement faciles, mais nous nous plîrent devant le besoin de redéfinir certaines fonctions et structures faces aux problèmes de appels de la base de donnée, de facilité d'accès et d'ergonomie du site et de conflits de CSS. Le projet est certes accompli dans sa totalité, mais la fonctionnalité et la présentation peuvent laisser à désirer: si nous avions plus de temps ou si nous désirions poursuivre ce projet pour en faire un véritable outil pour les élèves nous voudrions améliorer la présentation du site, ajouter des fonctionnalités pour rendre le site plus intuitif et le rapprocher d'un vrai réseau social (capacité de chat/messages suite à la mise en relation des élèves) et étoffer la sécurité liée à l'authentification des utilisateurs et l'accès à la base de donnée (protection contre failles SQLi, etc.).