# TP PHP Idea machine (boite à idée)

## Objectifs du test

* analyser votre capacité à développer une application en PHP et HTML/CSS/JS regroupant des briques fonctionnelles courantes (authentification, autorisation, CRUD - Create Read Update Delete)
* Analyser votre capacité à structurer le code d’une application dans sa globalité (et pas uniquement sur une page / fonction) et notamment en
  + en HTML
  + en CSS
  + en PHP
  + en persistance (BDD / Fichiers …)
  + en JS

## Fonctionnalités

Pour cela, vous allez devoir réaliser un site web permettant de simuler une “boite à idées”. C’est à dire que ce site permettra :

* Pour un utilisateur anonyme de :
  + consulter toutes les idées
  + se creer un compte (email / mdp)
  + se connecter
* Pour un utilisateur connecté de :
  + consulter toutes les idées
  + lister les idées dont il est l’auteur
  + modifier les idées dont il est l’auteur
  + supprimer les idées dont il est l’auteur
  + se déconnecter
  + ajouter une idée
  + voter (upvote OU downvote) pour des idées (**une seule fois par idée**)

Concernant Les idées :

* elles sont composées d’un titre (champs texte) et d’une description (textearea), d’une image (uploadée par le user), d’une date de création (automatiquement initialisée en php lors de l’enregistrement), et d’un auteur (correspondant au mail de l’utilisateur)
* elles sont affichées a deux endroits :
  + sur la home du site, triée selon leur note moyenne de upvote/downvote. Le texte est tronqué si nécessaire (selon design).
  + sur une page détail “idée”, affichant l’intégralité des info (intégralité du texte / date creation / upvote-downvote / auteur)
* elles ne peuvent être modifiées ou supprimées que par leur auteur.
* Seules 6 idées sont affichées sur la home, pour afficher les autres idées, un bouton “charger plus” doit être présent et permet de récupérer **en ajax** les 6 idées suivantes (etc).

## Contraintes techniques

### côté backend :

le projet doit-être développé en PHP (7 ou 8) **SANS framework** et sans librairie tierces (pas de twig ou d’ORM)

### côté Persistence de données :

Pour le stockage des données (notamment les utilisateurs et les idées), vous devez utiliser **SQLite** **OU un fichier à plat** au format de votre choix (txt / csv / json).

### côté Frontend :

HTML CSS et JS natif. Vous ne devez utiliser ni librairie (ex bootstrap, SCSS, etc), ni framework (vuejs, jquery, …).

Des wireframes sont à votre disposition pour clarifier le besoin fonctionnel, mais elles ne doivent pas être vues comme un carcan. Vous êtes donc libre d’adapter le design tant que vous respectez les fonctionnalités (et notamment le système de pagination).

A noter : les wireframes n’illustrent pas la fonctionnalité de création de compte, mais celle-ci est attendue dans votre rendu.

Vous êtes bien sûr libre de rendre le tout joli, selon votre inspiration (il y’aura une attention lors de la notation sur l’esthétique et l’utilisabilité de vos interfaces).

- le site doit être **responsive** (breakpoint à 1000px de large). Pensez à adapter la navigation (ex burger menu).

## Rendu

Vous devrez fournir :

* une archive (zip) contenant votre code source et les fichiers de DATA (SQLite ou fichiers à plat)
* Merci d’être vigilant au codage de vos urls dans vos fichiers. A titre d’info je testerai votre site en le faisant fonctionner à la racine d’une domaine (ex “<http://site.local>”) donc éviter d’hardcodé des url type “localhost/tp-php-6/index.php”

## Critères de notation

* respects des consignes
* respect des fonctionnalités
* qualité du code (architecture / DRY / KISS / absence d’erreurs / …)
* design (ergonomie)

N’hésitez pas à commenter votre code, et à m’informer de tout ce que vous jugerez utile.

Bon courage !