Pour les 3 projets proposés, le patron **MVC** doit etre utilisé. La gestion d'erreurs doit être complète. (champs vérifiés, connexion à la BD, etc.). Concernant l'interface, toutes les idées sont les bienvenues, la qualité de l'ergonomie de l'interface sera prise en compte.  
Les plus téméraires peuvent aller plus loin (ORM ?, templating avec Twig ?, etc)

Le projet est à présenter et à rendre en semaine 7 (TP).

**Projet A: ToDo List**

Pour ce projet, il s’agit d’une simple application de listes de tâches. Pour cette application il y aura deux acteurs : les visiteurs (non connectés) et les utilisateurs (connectés).

La seule différence entre ces deux acteurs se trouvera au niveau de leurs listes de tâches, en effet, les utilisateurs connectés pourront créer des listes de tâches qui seront privées et qu’eux seuls pourront voir. En revanche les visiteurs ont seulement accès à des listes de tâches publiques.

Chaque liste et chaque tâche doit être sauvegardée en base de données (pensez à la relation entre les deux). Voici le fonctionnement de l’application :

* Le visiteur peut voir/ajouter/supprimer des listes et des tâches publiques sans se connecter.
* Il faut créer un espace pour se connecter à l’application (si vous avez du temps, faire une partie inscription également).
* Une fois l’utilisateur connecté, il a accès aux listes publiques (comme le visiteur), mais également à ses listes privées.
* Toutes les listes de tâches ajoutées par un utilisateur sont privées par défaut afin de simplifier l’application. Il peut bien entendu supprimer ses listes également. Il faut penser à la relation entre les listes de tâches et l’utilisateur en base de données.
* Chaque tâche pourra être complétée via une case à cocher, ajoutez un bouton pour valider en dessous de la liste des tâches.
* Pour les plus téméraires, essayez de compléter/dé-compléter des tâches via des requêtes AJAX à la place du bouton valider (optionnel).

La gestion des droits doit être complète, un visiteur ne doit pas pouvoir accéder aux listes des utilisateurs ou les supprimer, idem pour la complétion des tâches.

En général dans une application de gestion de tâches, les tâches complétées sont barrées afin d’être distinguées des tâches restantes. Cela est faisable via CSS pur (i.e. : la page n’a pas besoin d’être actualisée, et l’utilisation de JavaScript n’est pas requise).