

| TP noté | Création d'un site utile |

RENDU le 1^{er} avril à 23h59

ORAL dans la semaine du 1^{er} avril

Objectifs du TP

- Approfondir le Framework Symfony
- Créer sa deuxième application en Symfony
- Travailler en équipe de 3 ou 4
- Présenter son travail devant un groupe

Rendu (par email à alexis@onylrocks.com)

- Support Présentation orale dans le dossier nommé ci-dessus
 - o Présentation du contexte (faut vendre votre appli !)
 - o Diagramme de classe
 - o Structuration du projet (si intéressant)
 - o Problèmes sur la réalisation / leviers utilisés
 - o Démo
 - o Retour d'expérience (quel bilan « je » et « nous » faisons)
- Lien du Repo Git de la forge de l'Université

Note

- Les points sont notés par fonctionnalités
- 1 point pour l'effort sur le CSS
- 1 point sur la structure globale, le nommage des variables, etc.
- Les bonus peuvent majorer la note (+ 1 point max) sans dépasser 20 mais surtout montrer votre compréhension du Framework, ses possibilités et votre implication dans ce cours
- La présentation du groupe sera notée en plus de cette note

Contexte

Jack Asseur a eu l'idée d'une application pour sa future « jeune entreprises innovantes à fort potentiel de croissance » de développer le site web qui règlera le problème dans les ménages : un site de gestion des listes de courses ! En ces temps de forte inflation, les français.e.s (et autres pays) sont fortement intéressé.e.s pour gérer au mieux leurs achats quotidiens, hebdomadaires et même mensuels.

Pour sa V0 (vu qu'il n'a pas d'argent), il a embauché une équipe de 2 ou 3 stagiaires de BUT pour la réaliser. Vous êtes cette équipe.

Pour aller voir les financeurs et leur vendre sa POC il aimerait que vous réalisiez différentes fonctionnalités présentées plus bas.

Il vous remercie par avance de votre travail et espère bien rentabiliser cet investissement.

Fonctionnalités attendues

Lors de l'arrivée de l'utilisateur sur le site <http://liste2courses.local>, l'utilisateur doit renseigner un pseudo qui sera *propriétaire* des listes de courses qu'il créera avec sa *session courante* ou en remettant le *même pseudo* [ndpⁱ : 1 point].

Dans le deuxième écran, et après une redirection suite au succès de la première page, l'utilisateur pourra *consulter l'ensemble de ses listes de courses existantes* [ndp : 2 points] (avec un bouton pour aller voir le détail de ses listes). Un *bouton d'ajout* d'une nouvelle liste de course est aussi à prévoir sur cette page ainsi que la *suppression* d'une liste de courses [1 point].

Suite au clic sur le bouton « Ajouter une liste » (et avoir renseigné un champ *input* pour donner un nom à la nouvelle liste [ndp : 1 point]), nous arrivons sur un troisième écran qui permet *d'ajouter des articles déjà définis* [ndp : 1 point] par un admin avec *une quantité* pour l'article sélectionné. Sur cet écran, l'utilisateur pourra *supprimer un article* ou *marquer cet article comme acheté* [ndp : 2 points].

Enfin, sur le deuxième écran, au-dessus de sa liste de liste de courses, Jack souhaite voir des statistiques relevant au minimum : la *totalité des sommes dépensées*, la *moyenne*, *l'article le plus cher* et *celui moins cher* et la *répartition des dépenses* par type d'article (sucré/salé) [ndp : 1 point/stat obligatoire].

Pour l'admin (qui a le pseudo *Minh*), il aura accès au CRUD pour gérer les articles ainsi que les types d'articles [ndp : 2 points/CRUD + 2 points gestion admin].

Bonus

- Récupération des articles non achetés pour les ajouter dans la liste en cours de création
- Ajouter la notion de « Magasin » qui sera lié aux articles afin de gérer des statistiques par magasin par exemple
- Ajouter des images aux articles
- Gérer une véritable authentification
- Permettre à plusieurs utilisateurices de collaborer sur la même liste de courses
- Restreindre l'accès aux listes de courses aux propriétaires

ⁱ Ndp : Note du Prof (à ne pas confondre avec *ndlr* qui est la *note de la rédaction*)