TP Symfony

Objectifs

- Être capable de créer un projet Symfony et de le versionner.
- Savoir manipuler les routes et les contrôleurs.
- Savoir créer et exploiter les entités.
- Savoir créer des formulaires et gérer les données utilisateurs avec les validateurs.
- Maitriser Doctrine avec le mapping, les migrations et le manager.
- Être à l'aise avec Twig.
- Envoyer des mails avec Symfony Mailer.

Modalités

Ce projet est à réaliser les journées du 27/01/21 et 29/01/21. Vous devrez versionner votre projet avec git et le rendre disponible sur Github pour le formateur. Vous ferez un maximum de commits avec chaque étape de vos travaux. Idéalement, vous pouvez travailler avec les Pull Requests sur Github mais ce n'est pas une obligation (Création de branches et *merge* dans la branche *master*).

Réalisation

Nous souhaitons créer un mini site e-Commerce. Vous avez quelques maquettes à disposition.

Vous créerez les routes suivantes ainsi que les contrôleurs et méthodes associées :

- / → IndexController::home
- /contact → IndexController::contact
- /product → ProductController::list
- /product/iphone-xs → ProductController::show($slug)
- /admin/product/new → ProductController::new
- /admin/product/edit/1 → ProductController::edit($id)
- /admin/product/delete/1 → ProductController::delete($id) (uniquement en POST)
- /category/smartphone → CategoryController::show($slug)
- /login → SecurityController::login()
- /register → SecurityController::register()
- /cart → CartController::index()

Il est évident que chaque contrôleur devra renvoyer une réponse correcte ainsi que la vue Twig associée.

Nous aurons besoin d'une entité métier *Product*. Cette entité contiendra les attributs suivants :

- Identifiant unique
- Nom du produit
- Slug du produit
- Description du produit
- Prix du produit
- Date de création
- Coup de cœur (permet de dire si le produit est un coup de cœur ou non)
- Une liste de couleurs (à stocker sous forme de *array* ou de *json* avec Doctrine)
- Image du produit (On devra pouvoir uploader une image)
- Promotion (entier contenant une promotion en pourcentage à calculer avec le prix)

Le nom du produit devra contenir au moins 3 caractères et la description au moins 10 caractères. Le prix du produit devra être un entier numérique compris entre 99 et X. Le slug du produit sera généré automatiquement à partir du nom du produit. Vous pouvez utiliser une bibliothèque comme Slugify ou le composant String de Symfony.

Vous vous aiderez de *fixtures* pour remplir la base de données une fois l'entité créée : https://symfony.com/doc/master/bundles/DoctrineFixturesBundle/index.html

Vous êtes libre dans la création de produits.

Pour les contrôleurs :

- La page *d'accueil* affichera 3 produits aléatoires de la base de données dans un carousel. Attention, il n'y a pas de ORDER BY RAND() dans Doctrine. Elle affichera également un produit coup de cœur aléatoire de la BDD. On affichera aussi les derniers produits créés ainsi que les meilleurs produits. Les meilleurs produits seront ceux avec les meilleures notes donc vous devez attendre de faire l'entité Review avant de faire cela.
- La page *products* affichera tous les produits dans la base de données. Une pagination sera présente, on affiche 6 produits par page. Une *sidebar* sur la gauche permettra de filtrer les produits par couleurs (On restera bien sur la même page, c'est un formulaire). La liste de catégories affichera des liens vers chaque catégorie de la boutique, vous devez attendre de faire l'entité Category. On affichera le dernier produit créé dans la BDD.
- La page *category/smartphone* affichera également la liste des produits mais seulement de la catégorie concernée.
- La page *products/iphone-xs* affichera le produit iPhone XS de la base de données. La description du produit pourra être du markdown transformé en HTML.

L'entité Review contiendra le nom d'un utilisateur, la date de création, la note, une description et bien sûr elle sera liée à l'entité Product. A partir de la fiche produit, on pourra ajouter une Review. L'administrateur du site pourra gérer les Review à partir du back office.

- La page contact est un simple formulaire de contact où l'utilisateur saisit les informations, on les valide et on envoie un mail à l'administrateur du site.

- La page */admin/product/new* affichera un formulaire permettant d'ajouter un produit dans la base de données, vous créerez le formulaire dans les règles de l'art en séparant la création de celui-ci dans une classe externe (Un *FormType*). L'entité devra être validée.
- La page */admin/product/edit* permettra de modifier le produit dont l'*id* est située dans l'URL. Vous utiliserez le même formulaire.
- La page */admin/product/delete* permettra de supprimer un produit. Cette page ne sera accessible que par la méthode POST de la requête HTTP. On devra donc passer par un formulaire pour supprimer le produit et pas par un simple lien.

Vous ferez la même chose pour les catégories, les reviews et les utilisateurs.

Pour la partie graphique, vous êtes libre de modifier les maquettes. La liste des fonctionnalités n'est pas exhaustive et vous êtes libre d'ajouter ou modifier des fonctionnalités en fonction de votre imagination.

Le but n'est pas de terminer le TP au plus vite mais bien de manipuler Symfony et d'être à l'aise avec le framework. Chacun avancera comme il le peut sur chaque partie.

BONUS

Vous implémenterez la partie sécurité avec l'inscription d'un utilisateur et le login. A la connexion, l'utilisateur verra son email apparaitre dans le menu et pourquoi pas son avatar ?

Vous implémenterez un système de panier en utilisant les sessions. Le panier devra donc être conservé durant la visite d'un utilisateur et celui-ci pourra ajouter / modifier ou supprimer un produit du panier.

Vous implémenterez la possibilité de rechercher d'un produit.

Bon courage.