Projet
Développement et architecture WEB

L'objectif du projet est de réaliser une application WEB todo list. Le projet est à faire par groupe de deux. Dans vos fichiers composer.json, pensez à bien renseigner les DEUX auteurs du groupe, avec vos noms, prénoms et adresses mail. La date du rendu sera fixée ensemble, probablement vers la mi-janvier 2022.

Si des manipulations spécifiques sont nécessaires pour faire fonctionner vos projets, vous pouvez ajouter une documentation sous la forme d'un readme.md à la racine. MAIS je m'attends à pouvoir faire fonctionner votre projet en faisant un `docker-compose up`

1. Une application MVC

L'objectif ici est de réaliser le site WEB "Todo List". Elle comptera pour 14 points sur la note globale du projet.

Cette application devra respecter les contraintes suivantes :

- Langage de programmation: PHP 8 / XHTML / CSS / JS
- Utilisation de composer pour les dépendances PHP
- Utilisation du Framework Slim 4
- Accès à la base de données MySQL avec PDO
- Utilisation du framework d'injection de dépendances PHP-DI
- Utilisation de variables d'environnement pour la configuration de l'app
- Utilisation de NPM pour les dépendances node
- Utilisation du framework Bootstrap 5
- Site web responsive design utilisable sur 3 types de device: smartphone, tablette et desktop
- Utilisation du moteur de template Twig
- Utilisation du pré-processeur de feuille de style Sass
- Utilisation du framework javascript jQuery
- Mise en place d'une structure MVC (modèle - vue - contrôleur)
- Utilisation de docker et de docker compose pour l'infrastructure

Fonctionnalités attendues :

- Fonctionnalité de login et mise en session de l'utilisateur.

Une fois connecté (et seulement une fois connecté), les fonctionnalités suivantes doivent être mise à disposition de l'utilisateur

- Listing des tâches

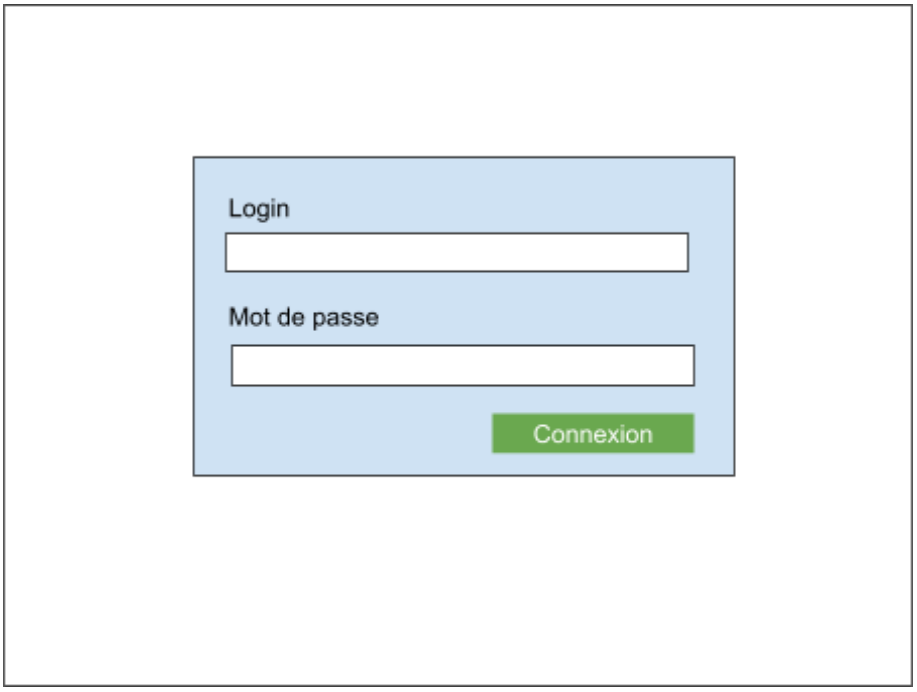
- Filtrage des tâches par utilisateur
- Filtrage des tâches par mots clés
- Affichage du détail d'une tâche, avec la liste des commentaires associés
- Ajout d'une nouvelle tâche (voir champs dans le fichier SQL TP n°1)
- Ajout d'un nouveau commentaire à une tâche
- Fonctionnalité de déconnexion

Cela représente en tout 6 use cases (connexion, puis liste / recherche de tâches, détail d'une tâche, ajout d'une tâche, ajout d'un commentaire et déconnexion)

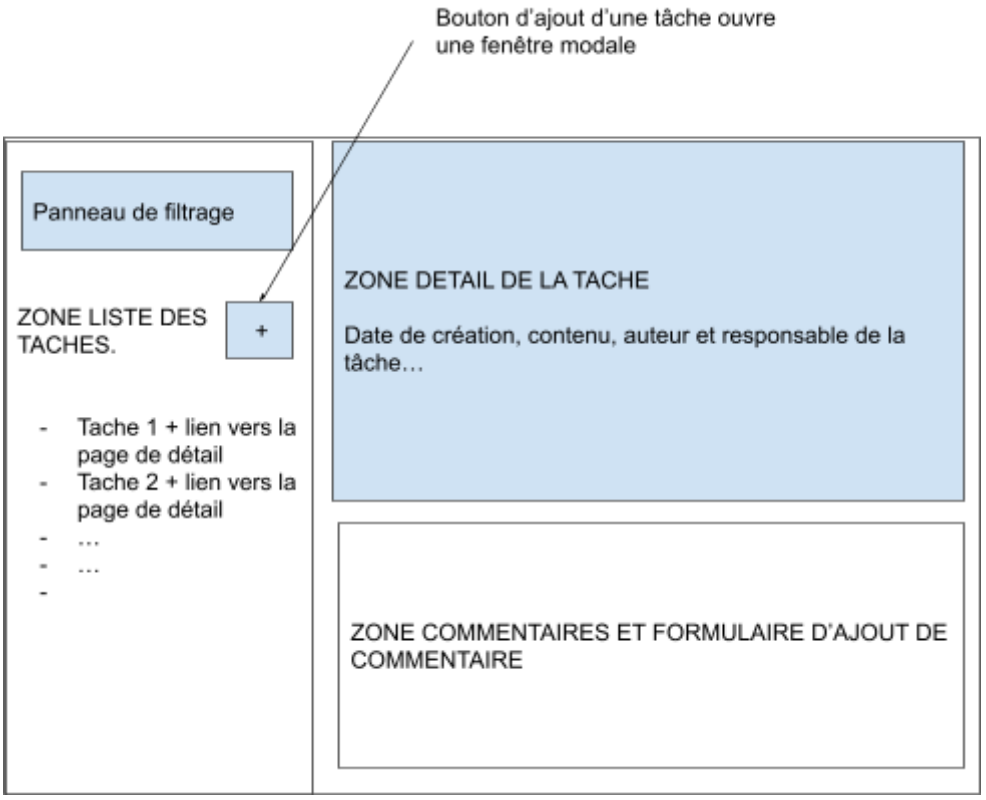
Une contrainte supplémentaire est de réaliser une des deux fonctionnalités d'ajout de tâches et/ou de commentaires en ajax (Javascript asynchrone) avec jQuery. La fonctionnalité d'ajout doit être réalisée au travers d'une fenêtre modale de Bootstrap 5.

L'expérience utilisateur et l'ergonomie sont à votre discrétion, cependant voici un certain nombre d'idées que vous pourrez utiliser pour réaliser votre projet.

Pour le login



Pour la fenêtre une fois connecté



La notation se fera de la manière suivante :

- 9 points pour les use cases
- 3 points liés au respect des consignes
- 2 points sur les aspects lisibilité, cohérence, sécurité, performances et tests

2. Une API REST JSON intégrée

Cette partie comptera pour 6 points dans la note du projet.

Le business souhaite que l'équipe de développement publie une application mobile native basée sur les mêmes fonctionnalités que le site web. Nous ne développerons pas l'application mobile mais devons exposer les données aux équipes chargées de leur développement via une API REST Json. Par soucis de timing, nous intégrerons cette API dans le projet WEB existant et ne bootstrapons pas un nouveau projet.

En plus des contraintes précédentes, l'API todo devra respecter les instructions suivantes:

- Authentification à l'API par token header non chiffré
- Format d'échanges clients / serveur JSON
- Architecture hexagonale
- Routes spécifiques commençant par www.domaine.com/**api**/xxx

Fonctionnalités attendues :

- Endpoint de listing / recherche des tâches (pour simplifier, pas de pagination)

- Endpoint de création d’une tâche

Concrètement, l’objectif est de mettre en place un nouveau répertoire “api” à la racine du projet, qui contiendra les deux use-cases mentionnés ci-dessus. Ces deux use cases doivent respecter une architecture hexagonale. Les mêmes repositories, models, représentation des requêtes métier doivent ensuite être partagés et utilisés par les contrôleurs de votre site web.

Le site web et l’API todo partagent la même base de données.

La notation se fera sur la base des corrections que vous aurez apporté au premier rendu:

- Développement de l’authentification : 1.5 point
- Développement des use cases de l’API todo: 3 points
- Refactorisation du site web : 1.5 point