CÉGEP SHAWINIGAN
Du savoir et des gens

TP2

420-W1V-SW - Développement d'applications Web transactionnelles avancées

Pondération de toutes les parties : 17.5%

Travail à effectuer

Votre application doit répondre à un besoin. Vous pouvez choisir quel genre d'application vous ferez. Le besoin peut être sérieux, réel, imaginaire ou loufoque.

Exemple :

- Magasin : Notre application vise à rendre des produits ou des services plus accessibles.
- ToDo list : Notre application sert à centraliser la création de tâche et la gestion de tâche pour une équipe dans une industrie.
- Carnet de voyage : Notre application vise à permettre aux voyageurs de cataloguer et de partager leurs profil de voyageur.

L'application doit répondre à quelques éléments de base :

- L'application doit être adaptée aux petits et aux gros écrans (Responsiveness).
- Il doit y avoir 3 types d'utilisateurs :
    - **Utilisateur** : Un utilisateur classique. Peut être un anonyme ou un utilisateur enregistré "client". Ce sont les personnes qui vont utiliser l'application et interagir avec les systèmes. L'utilisateur devrait être le membre qui effectue une tâche.
    - **Gestionnaire** : Ce type représente les utilisateurs qui ont les accès pour gérer certains éléments. Par exemple, un gestionnaire peut ajouter et retirer des produits d'un magasin en ligne. Sinon, d'un point de vue équipe, il peut être celui qui crée les tâches. Il peut avoir le contrôle sur le rôle utilisateur et gestionnaire.
    - **Administrateur** : Représente une personne parmis un groupe d'utilisateur qui à presque tous les droits dans l'application. Il a le contrôle sur les rôles utilisateur, gestionnaire et administrateur.

Partie 1 : Modèle de données

Vous devez produire les modèles de données dont vous avez besoin pour votre application. Vous pourrez en tout temps demander des retours à l'enseignant sur vos modèles de données. Je vous invite à les faire dans un tableau ou un diagramme. Je vous invite ensuite à vous faire un plan de travail pour votre projet. Vous pouvez monter ce plan de travail dans un format pseudo-SRS pour conserver une bonne idée de la direction de votre projet tout au long de sa durée.

X	Modèle de données	Note /4
	Liste des entitées	
	Relation entre les entités	
	Un plan de travail à été fait	
	TOTAL :	

Partie 2 : Maquettes

Vous devez produire des maquettes de l'application à l'aide d'un outil de Figma. Ces maquettes serviront de base à l'implémentation de l'interface en VueJs. Pensez à faire une maquette pour chacune des fenêtres que vous aurez besoin de faire pendant l'implémentation. Ces maquettes doivent respecter les principes de design:

- Hiérarchie : taille et effets sur la police, le contraste et l'espacement.
- Constance : Style graphique, les boutons ont des comportements similaires.
- Contraste : Les éléments critiques sont visible et distinct
- Proximité : Les éléments qui sont liés sont proches les un des autres.
- Alignement : Facilite la lecture et augmente la prédictibilité.

X	Maquettes	Note /4
Couverture de l'application		
	Il y a une maquette pour chaque visuel de l'application	
	Les maquettes sont groupé selon leur fonctionnalités	
	Une composante qui partage de l'information clairement aux utilisateurs.	
Hiérarchie		
	La taille et les effets de la police changent selon l'importance de la fonctionnalité.	
	Le contraste entre le texte et les couleurs facilite la lecture.	
	Le contraste entre les composantes valorise l'importance des fonctionnalités.	
	Les composantes sont espacées pour permettre la distinction des composantes.	
Constance		
	Le style graphique est constant dans la maquette.	
	Les composantes similaires ont un comportement similaire dans la maquette.	
Contraste & Proximité		
	Les éléments critiques se distinguent par leur emplacement et/ou leurs couleurs.	
	Les éléments qui sont en relation sont proches les un des autres.	
Alignement		
	Les composantes sont alignées ensemble de façon cohérente.	
	TOTAL :	

Partie 3 : Implémenter l'application

L'application **Serveur** doit implémenter les fonctionnalitées suivantes :

X	Application Serveur	Note /4
Authentification		
	Une authentification est implémentée.	
	L'authentification redirige la requête à /login si l'utilisateur doit être authentifié.	
	L'authentification redirige la requête à l'url initial de la requête après le login.	
Rôle		
	L'accès au routes CRUD sont restreintes aux gestionnaires et aux administrateurs.	
	Des routes spécifiques existent pour permettre aux utilisateurs d'interagir avec les éléments souhaités.	
ARCHITECTURE MVC		
	Logique métier se trouve dans les modèles	
	Les intéractions avec la BD se trouve dans les modèles	
	Les communications client-serveur sont dans les contrôleurs ou les routes.	
GESTION DES ERREURS		
	Middleware d'erreur	
I18N		
	Les traductions sont dans les fichiers de la langue spécifiée.	
	Les réponses dépendent de la langue spécifiée dans le header de la requête.	
Console.log non justifié (utilisez le debugger)		
	TOTAL :	

L’application **Client** doit respecter les éléments suivants :

X	Application Client	Note /4
Respect des maquettes		
	Chacune des pages correspond à sa maquette.	
Authentification		
	Le token est enregistré dans les cookies.	
I18N		
	Les traductions sont dans les fichiers de la langue spécifiée.	
	Les labels dépendent de la langue de l’utilisateur.	
Stores		
	Il y a un store par entité visible.	
	Le store fetch un nombre limité de données par requête (pagination).	
Pagination		
	Les listes affichées sont limitées par un certain chiffre.	
	Il est possible de naviguer dans les données avec la pagination.	
Responsive		
	L’affichage est adéquat sur petit écran (cell, tablette).	
	L’affichage est bon sur écran traditionnel (1920x1080).	
Router		
	La gestion des pages est faite par un router.	
Bonnes pratiques		
	Lorsqu’un élément n’a pas terminé de charger ses données, la composante doit montrer clairement qu’elle effectue un travail (loading component).	
	Les composantes donnent une rétroaction efficace selon leur fonction.	
	Les composantes similaires utilisent une composante générique.	
“Debugger” trouvé dans le code		
	TOTAL :	

Remise

La remise de ce travail s'effectura en 3 parties. Chaque partie doit être présentée à l'enseignant sous forme de démonstration. Vous avez droit à une présentation formative par partie pour avoir de la rétroaction sur votre travail et sur votre présentation. Vous pouvez contacter l'enseignant pour planifier vos présentations. Vous pouvez choisir de passer à la partie suivante avant la présentation d'une partie.
Vous pouvez demander une rencontre à n'importe quel heure du cours, pourvu qu'il est envisageable de faire votre présentation avant la fermeture du zoom.

Vous devez **remettre** votre travail **en format ZIP** sur **LÉA après** votre **présentation finale**. Vous devez exporter vos **maquettes en PDF** et les mettre dans un dossier "/pdf" à la racine de votre dossier de projet. Vous devez **supprimer les "node_modules"** de vos dossiers avant de compresser.

La date limite de la présentation finale est le **8 août à 11:00.**

Évaluation

4	3	2	1	0
Les éléments demandés sont effectués de façon exemplaire.	Les éléments demandés sont respectés et le code est de bonne qualité.	Les éléments demandés sont respectés, mais la qualité du code est à améliorer	Les éléments demandés sont incomplets ou trop peu élaborés.	Les éléments demandés sont absents.

X	Présentation	Note /4
	Fluidité de la présentation	
	Un plan de présentation à été fait	
	Gestion adéquate d'un bug pendant la présentation	
	Compréhension du code	
	TOTAL :	

X		
	Modèle	
	Maquettes	
	Application Client	
	Application Serveur	
	Présentation	
	Critère de remise respectés (/4)	
	TOTAL :	