Samuel Ferragne 2208610  
Liam Brouillard Adjaïlia 2210406  
Web et base de données  
420-4A5-MO

Rapport TP1

Travail présenté à  
Sylvain Labranche

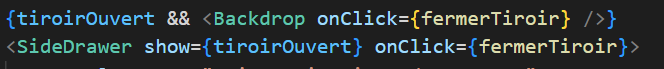
Techniques de l'informatique  
Collège Montmorency  
14 mars 2023

Compréhension du problème

1. Sur chaque page :  
   - Menu de navigation avec des liens pour aller vers l’accueil, la page professeurs et la page des cours offerts  
   - La page actuellement sélectionner doit être démarquée des autres  
   - Si la largeur de la page est trop petite pour afficher les liens on utilise un tiroir pour les afficher  
   - Refermer le tiroir si on clique a coter
2. Page d’accueil :  
   - Logo de montmorency  
   - Liste de fonctionnalités implémentés au site
3. Page professeurs :  
   - Liste de *Card* affichant tous les professeurs travaillant au cégep (avec photo)  
   - Formulaire pour ajouter un nouveau professeur (Doit avoir nom, prénom, date d’embauche et ajouter une photo)   
   - Le nouveau professeur n’a pas de cours attribué
4. Page des cours :  
   - Liste de *Card* affichant tous les cours offerts   
   - Formulaire pour ajouter un nouveau cours (Doit avoir titre, discipline, nombre maximal d’étudiants, date de début et date de fin)  
   - Vérifier que les dates du cours sont comprises dans la session actuelle (1er janvier au 6 juin)  
   - Vérifier que le professeur attribué au cours existe sinon l’ajouter a la liste de professeur qui travaille au cégep  
   - Le nouveau cours n’a pas d’étudiants attribué  
   - Lorsque que l’on clique sur la *Card* d’un cours nous dirige vers la page du cours en particuliers  
   - Filtrer les cours pour voir les cours disponibles jusqu’à 4 session en arrière
5. Page de cours individuel  
   - Liste de *Card* affichant tous les étudiants inscrits  
   - Afficher le professeur qui donne le cours  
   - Formulaire pour inscrire un étudiant au cours (Doit avoir nom, prénom et numéro d’admission)  
   - Vérifier que l’élève n’est pas déjà inscrit (Ne doit pas avoir de numéro d’admission doublon)

Résultat de l’implémentation

1. Text

   Description automatically generatedText

   Description automatically generatedSur chaque page  
   + Nous avons implémenter un menu de navigation comprenant le lien vers l’accueil, la page professeurs et la page des cours offerts. On utilise la fonction Router que React offre pour charger les pages et recharger le DOM lorsque l’utilisateur entre la bonne URL ou clique sur le lien dans le menu.  
   + La page sélectionner se démarque grâce aux CSS suivant :   
   + La page affiche un tiroir lorsque la fenêtre est plus petite que 768px :  
     
     
     
   et referme le tiroir lorsque l’on clique (à coter ou sur le tiroir):
2. Page d’accueil  
   + Le logo montmorency est ajouté en utilisant le lien web de l’image  
   + Les fonctionnalités sont ajoutées en utilisant un composant séparé pour faciliter la lecture du code de la page Accueil.js
3. Page professeurs  
   + Liste de *Card* affichant la photo, le prénom et le nom du professeur ainsi que tous les cours donnés par celui-ci  
   + Les données des professeurs sont stocker sur App.js et sont données en paramètre lors de l’appelle de la page  
   + Le formulaire prend toutes les informations données et l’enregistre grâce a la méthode *setProf* passée en paramètre en même temps que la liste de profs (le professeur créé a une liste de cours vide)  
   + Vérification que tous les champs du formulaire sont remplit
4. Page des cours  
   + Liste de *Card* affichant les cours de la session actuelle avec les informations suivante : titre, discipline, nombres de places maximum, date de début et de fin et nom du professeur  
   - Nous avons dû laisser de coter le filtrage des cours et l’affichage des cours des sessions antérieurs  
   + Les données des cours sont stocker sur App.js et sont données en paramètre lors de l’appelle de la page  
   + Lorsque l’on clique sur un cours nous sommes renvoyé vers la page du cours (*voir 5.)*  
   + Le formulaire prend toutes les informations données et l’enregistre grâce à la méthode *setCours* passée en paramètre en même temps que la liste des cours (le cours créé a une liste d’étudiant vide)  
   + Vérification que les dates sont comprises dans la session actuelle (1er janvier – 6 juin)  
   + Le formulaire permet d’ajouter un nouveau professeur à la liste des professeurs si celui-ci n’existe pas. Ce nouveau professeur aura l’image par défaut comme image et la date actuelle comme date d’embauche
5. Page de cours individuel  
   + Liste de *Card* affichant tous les étudiants inscrit avec leur nom et numéro d’admission  
   + Affiche le professeur du cours tout en haut de la page  
   + Le formulaire prend toutes les informations données et l’enregistre grâce à la méthode *setCours* passée en paramètre en même temps que la liste des cours   
   + Le formulaire s’assure que le numéro de l’élève se faisant ajouter au cours n’est pas déjà présent

Journal de bord

**Samuel Ferragne**

Première conception du rapport (compréhension du problème)   
 -12 mars (0.5 heure)  
Création des dossier de bases du projet  
 -12 mars (1 heure)  
Création du github  
 -12 mars (1 heure)  
Création des pages simplifiées  
 -12 mars (1 heure)  
Création navigation vers les pages des cours individuel  
 -14 mars (0.5 heure)  
Ajout de la liste d’étudiants inscrits au cours  
 -14 mars (1 heure)  
Ajout de la liste de cours de chaque profs  
 -14 mars (0.5 heure)  
Finition du rapport  
 -15 mars (1 heure)

**Liam Brouillard Adjaïlia**

Création des formulaires pour créer un nouveau professeur ou élève  
 -14 mars (1.5 heure)

Débug de différents problèmes liés aux formulaires  
 -14 mars (2 heure)

Création des programmes permettant la création de nouveaux professeurs ou élèves  
 -14 mars (2 heures)

Debug de différents problèmes liés aux programmes  
 -14 mars (1.5 heure)

Création du formulaire pour ajouter un nouveau cours  
 -15 mars (0.5 heure)

Débug de différents problèmes liés au formulaire  
 -15 mars (0.5 heure)

Création du programme permettant la création d’un nouveau cours  
 -15 mars (1 heure)

Débug de différents problèmes liés au programme  
 -15 mars (2 heures)

Débug général des problèmes restants dans l’application  
 -15 mars (0.5 heure)

Ajout de plusieurs fonctionnalités manquante dans l’application  
 -15 mars (1 heure)