Examen 2

INSTRUCTION DE RENDU:

Créer un repository dans GitHub au nom 1017examen2votrenom que vous donnerez en réponse à l’examen dans moodle. Les réponses aux questions se trouveront dans le fichier README du repository.

1 - Créer un application VueJS avec vueCLI

Donner les commandes nécessaires pour créer l’application

Faire un initial commit dans votre repository avec les fichiers créés par le vueCLI

2 - Supprimer la vue HelloWord.vue créée par défaut.

Supprimer le logo de VueJS dans App.vue

Créer un composant représentant votre page d’accueil dans un répertoire views.

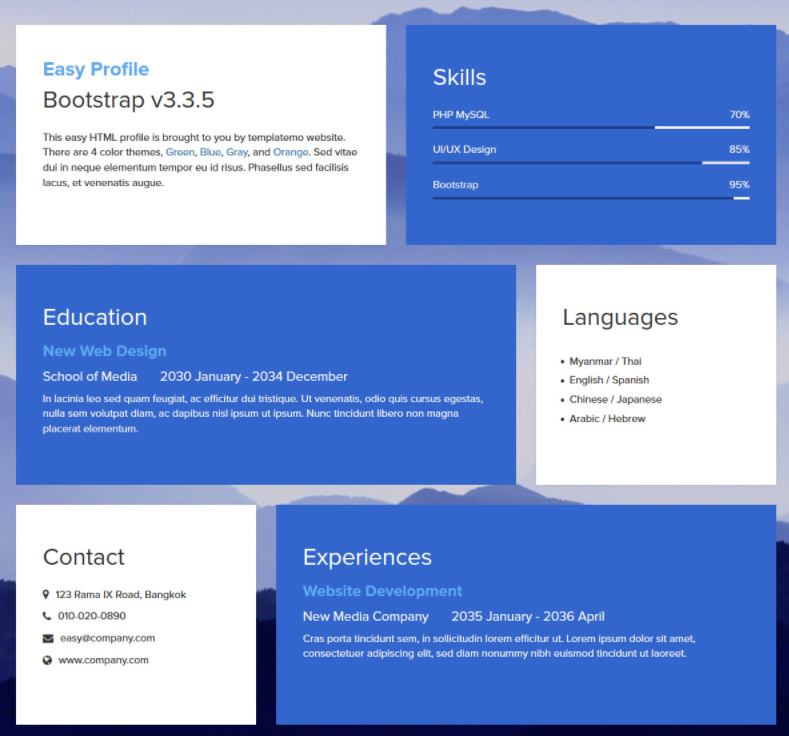
Créer un composant Header et un composant footer dans le répertoire components.

Ajouter ces 3 composants dans votre fichier App.vue.

Commiter votre code avec le message de commit “mes premiers composants”

3 - Utiliser la méthode de Hook beforeCreate qui vérifiera si un utilisateur est connecté et qui affichera une page standard si l’utilisateur est déconnecté ou une page de profil si l’utilisateur est connecté. (on simulera avec un booléen loggedIn - true/false pour savoir s’il est connecté ou non). La page d’accueil restera la même que ce que vous avez créé à la question précédente si l’utilisateur est déconnecté. Si l’utilisateur est connecté on affichera les informations suivantes en dur puisque celles-ci ne prendront pas en compte de vraies informations.

Exemple:



4-

* Faire un composant Projet qui contiendra une liste de projets de votre choix (Vous pouvez mettre les projets que vous avez réalisés au cours de vos sessions)
* Faire un composant contact contenant les champs suivants:
  + Nom
  + Prénom
  + Courriel
  + Catégorie (un dropdown que l’on remplira avec un tableau que l’on parcourra avec une directive v-for)
* On mettra ces composants dans le répertoire views
* Puis commiter votre code avec le message de commit “mes nouveaux composants”

5 - Faire un menu pour accéder aux différents menus. On utilisera un tableau qui contient les menus et on utilisera une directive v-for pour les afficher dans le menu. On utilisera le plugin Vue-routeur pour pouvoir naviguer à travers nos composants.

Puis commiter votre code avec le message de commit “mon menu avec vue router”

6 - Faire un Mixin qui permet de traduire les libellés de notre application.

On utilisera le français ou l’anglais pour les libellés et les libellés se trouveront dans une constante dans le fichier de déclaration du mixin

Exemple:

const TRANSLATIONS = {

en: {

firstName: "Firstname",

age: "Age"

},

fr: {

firstName: "Prénom",

age: "Ager"

}

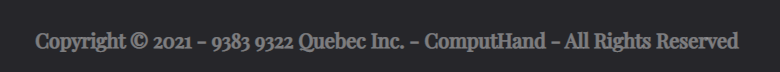
};

On utilisera un filtre pour afficher la traduction.

Puis commiter votre code avec le message de commit “mon mixin translate”

7 - Créer une directive personnalisée qui met la date et le copyright dans le bas de page.

Exemple:



Puis commiter votre code avec le message de commit “ma directive bas de page”