

**420-KB5-LG****Programmation WEB****Projet Final d'Intégration**

## Description

Ce travail consiste à élaborer une application Web à l'aide de JavaScript et de bibliothèque comme jQuery. L'application permettra de faire la gestion simple d'une liste de contact téléphonique. L'image ci-dessous montre un peu le visuel de l'application. Les couleurs et la disposition ne sont pas obligatoirement celles montrées ci-dessous.



The image shows a web application interface for a contact manager. At the top, there is a header bar with the text "Gestionnaire de contacts". Below the header, there is a table with three columns: "Nom", "Téléphone", and "Courriel". The table is currently empty, and the background is a light gray color.

Nom	Téléphone	Courriel
-----	-----------	----------

## Consignes

- Faire usage de CSS dans votre travail.
- Faire usage des techniques de programmation JavaScript que nous avons vu durant la session.
- Faire usage de jQuery.
- Faire des requêtes AJAX à un service Web.

## Partie 1 : Faire l'affichage des données présentes

Pour cette portion, vous devez interroger le service Web en effectuant une requête ajax et faire afficher les données que vous allez récupérer.

L'affichage sera simpliste pour l'instant et devra ce faire par l'intermédiaire d'un tableau. Vous devrez réaliser une fonction qui se nommera « maketable » et qui acceptera le nombre de rangée, le nombre de colonne et un tableau qui contient les données de la liste téléphonique. La fonction créera le tableau selon le nombre de rangée et de colonne et y insérera les données au bon endroit.

Voici quelques informations concernant le service Web (Un gros merci à Nicolas Chourot pour l'élaboration et l'implantation de ce service Web)

Son URL : [http://www.informatique.clg.qc.ca/KB5\\_Contacts/api/Contacts](http://www.informatique.clg.qc.ca/KB5_Contacts/api/Contacts)

Méthode d'envoi : GET

Voici un exemple de données reçues de la part de ce service :

```
[{"Id":1,
  "Name":"Thérèse Cripies",
  "Phone":"(514) 888-939",
  "Email":"therescrispie@gmail.com",
  "Sexe":1,
  "Birthdate":"1950-05-19T00:00:00",
  "Single":true},
{"Id":2,
  "Name":"Armande Zampu",
  "Phone":"(450) 430-3120",
  "Email":"Armande.Zampue@gmail.com",
  "Sexe":1,
  "Birthdate":"1970-02-01T00:00:00",
  "Single":false},
{"Id":3,
  "Name":"Alain Proviste",
  "Phone":"(450) 430-3120",
  "Email":"ALainProviste@hotmail.com",
  "Sexe":0,
  "Birthdate":"1965-01-01T00:00:00",
  "Single":true
}]
```

Le format des données est le suivant :

**Id** : int / non éditable

**Name** : string / obligatoire /

**Phone** : string / obligatoire / (000) 000-0000

**Email** : string / obligatoire /

**Sexe** : int / obligatoire / 0 – masculin , 1 – féminin, 2 – autre

**Birthdate** : string / obligatoire / YYYY-MM-DD

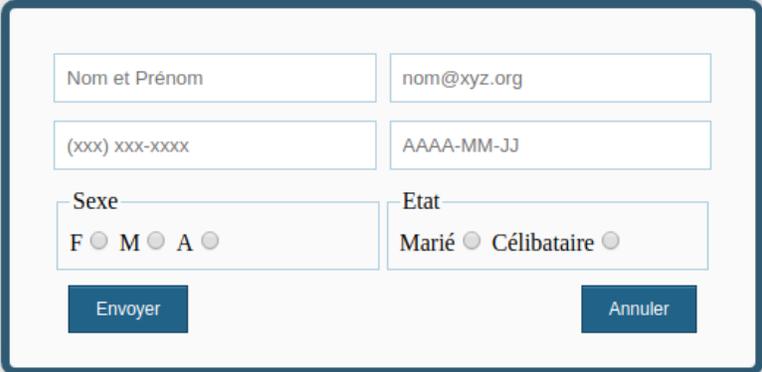
**Single** : bool / true – false

## Partie 2 : Ajout d'une entrée dans le bottin

Pour cette partie, vous devez permettre d'ajouter une nouvelle entrée dans le bottin par l'intermédiaire d'un formulaire.

Vous devrez ajouter un bouton à la fin de la liste des contacts qui permettra d'ajouter une entrée. Ce bouton sera affiché à la fin de la liste des contacts.

Ce bouton fera apparaître un formulaire tel que montré ci-dessous :



The image shows a contact form with the following fields and controls:

- Nom et Prénom**: Text input field containing "nom@xyz.org".
- (xxx) xxx-xxxx**: Text input field for phone number.
- AAAA-MM-JJ**: Text input field for birthdate.
- Sexe**: Radio button group with options F, M, and A.
- Etat**: Radio button group with options Marié and Célibataire.
- Envoyer**: Submit button.
- Annuler**: Cancel button.

Le nom et prénom, le courriel ainsi que le numéro de téléphone sont des champs « input » de type « text ». Le champ pour la date de naissance est un champ « input » de type « date ». Les deux derniers champs « input » sont des boutons radio.

Notez que :

Bouton radio « Sexe » : Représente une valeur entière comme suit : F vaut 1, M vaut 0 et A vaut 2.

Bouton radio « Etat » : Représente une valeur booléenne comme suit : Marié vaut true et Célibataire vaut false.

Il faudra aussi très probablement transformer le champ « date » en « datetime » avant de l'envoyer au service Web.

On vous donne 2 petits cadeaux :

Un fichier CSS est disponible sur le site du cours pour l'utiliser avec le formulaire.

On vous donne ci-dessous le code pour la requête AJAX dans le cas d'un ajout à la base de données du service WEB :

```
$.ajax({
    url: "http://www.informatique.clg.qc.ca/KB5_Contacts/api/Contacts",
    type: 'POST',
    data: contact,
    success: () => { //Refaire afficher le bottin des contacts},
    error: function(jqXHR, textStatus, errorThrown) {
        //Message d'erreur
    }
});
```

Dans cette requête AJAX, le type POST est utilisé pour mentionner que l'on envoie des données au serveur. Les données sont dans un objet « contact » que l'on passe à l'étiquette « data ».

Autrement dit, vous n'avez qu'à bâtir votre objet « contact » et à utiliser l'étiquette « data ».

Pour la fonction success, vous devez refaire afficher la liste complète du bottin. Vous pouvez réutiliser le code que vous avez développé à la partie 1.

Bouton « Envoyer » :

Le formulaire devra préalablement être validé avant de l'envoyer.

Les données sont envoyées, le formulaire disparaît et le bouton « Ajouter » réapparaît.

On fait réafficher le bottin avec la nouvelle entrée.

Bouton « Annuler » :

Fait disparaître le formulaire et fait réapparaître le bouton « Ajouter ».

Validation :

- Tous les champs sont obligatoires.
- Le champ courriel devra être validé avec une expression régulière.
- Le champ téléphone devra être validé avec une expression régulière. On accepte uniquement la forme suivante (xxx) xxx-xxxx où x est un chiffre de 0 à 9 et le code régional est borné par une parenthèse ouvrante et fermante. Il y a aussi un espace entre la parenthèse fermante et le prochain chiffre.

Formulaire :

Vous pouvez appliquer les classes CSS suivantes aux éléments de formulaire en utilisant le fichier CSS que l'on vous a fourni.

**class="field-style field-split align-left**

**class="field-style field-split align-right**

The image shows a form with two columns. The left column is highlighted with a red box and has a red arrow pointing to the text 'class="field-style field-split align-left"'. The right column is also highlighted with a red box and has a red arrow pointing to the text 'class="field-style field-split align-right"'. The form contains the following fields: 'Nom et Prénom', '(xxx) xxx-xxxx', 'Sexe' (with radio buttons for F, M, A), 'nom@xyz.org', 'AAAA-MM-JJ', 'Etat' (with radio buttons for Marié, Célibataire), 'Envoyer', and 'Annuler'.

Pour l'utiliser convenablement, les éléments doivent être dans une balise « div » et « ul » et « li »

Exemple :

```
<div class="form-style-9">  
  Éléments de formulaire ici avec des ul et li  
  <ul>  
    <li>  
      <input ... class="field-style field-split align-left" ...  
    </li>  
    <li>  
      <input ... class="field-style field-split align-left" ...  
    </li>  
  </ul>  
</div>
```

Utilisez align-left ou align-right pour l'alignement à gauche ou à droite.

## Échéancier de remise

Partie 1 – Affichage des données	Jeudi 9 mai 2019 avant 23:59	Vérification en classe Le lundi 13 mai
Partie 2 – Ajout d'une entrée	Jeudi 30 mai 2019 avant 23:59	
Partie 3 – À voir		

## Remise finale du projet

Le jeudi 30 mai 2019 avant 23:59, vous remettez tous vos fichiers qui font partie de votre projet à part les fichiers relatifs à jQuery de googleApi.