# Contrôle de compétence

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nom:**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | **Prénom:**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | **Classe:**  \_\_\_\_\_\_\_\_\_ | **Date:**  \_\_\_\_\_\_\_\_\_\_ | **NB pts:**  \_\_\_\_\_\_\_\_ | **Note:**  \_\_\_\_\_\_ |

# Compétence du module :

Implémenter au moyen d’une technologie définie une interface front-end respectant des standards d’interfaces actuels.

# Durée :

* **4 périodes de 45 minutes** (avec**?? points** maximum)

# Consignes :

* Toutes les attentes de ce test sont stipulées dans ce document.
* N’oubliez pas de remplir les blancs (nom, prénom, etc.) sur chaque feuillet.
* A la fin de cette EP, vous devrez rendre votre donnée complétée.

# Moyens d’aide :

* Notes personnelles
* Exercices du module (selon le contexte des exercices)
* Supports de cours et livres
* Accès internet autorisé en dehors des IA (tel que chatGPT).
* Les natels, smartphones, montres connectées, etc. sont posés sur la table.

# Règles d’évaluation :

Le nombre de points par questions ou tâches est mis en évidence à côté de leurs énoncés. Les règles de corrections sont fixées sur un document annexe à la seule disposition des correcteurs.

Il n’y a pas de pondération prévue.

**Calcul de la note :**

* **Règle de base**

Note = [(Nb points obtenus / Nb points total) \* 5] + 1

* **Condition particulière**

Aucune.

Remarques / compléments :

Aucune

# Partie 1: Application permettant de connaître les prochaines correspondances des CFF

**Contexte** : Vous devez contribuer au développement d’une mini-application Vue.js 3 en utilisant l’API Composition.

**Objectif** : Votre mission consiste à implémenter une fonctionnalité permettant de trouver les correspondances des CFF au départ de la gare A pour une arrivée dans la gare B.

Pour cela vous allez utiliser l’API des CFF !

**Contraintes**:

L’application doit contenir deux **composants** :

* HomeView.vue
* SearchResults.vue

L’application doit utiliser **Vue Router** pour gérer l’affichage des 2 composants.

**Sources** :

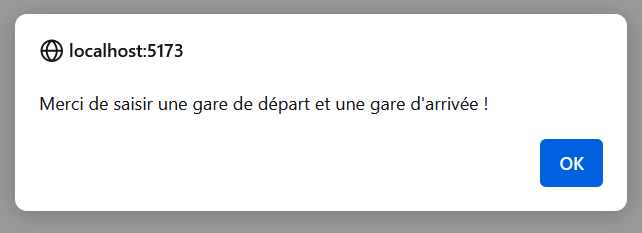
Vous trouverez un code de départ dans le répertoire **Sources**.

## Page d’accueil :

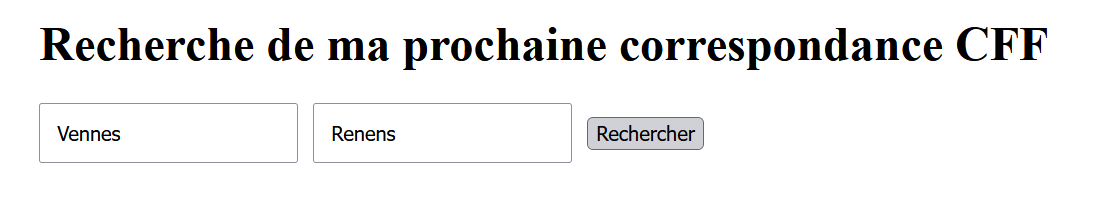
La page d’accueil doit ressembler à cela :

L’utilisateur doit saisir une gare de départ et une gare d’arrivée.

Dans le cas contraire, une « popup » doit s’afficher :



Si l’utilisateur fournit des informations correctes, par exemple :



## Page de résultats de la recherche :

Le résultat devrait être :



Le lien « Retour à la recherche » permet à l’utilisateur de retourner à la page d’accueil pour effectuer une nouvelle recherche.

Dans le cas contraire, si aucune correspondance n’est trouvé :



# Partie 2 : Application des enseignants

Vous vous souvenez avoir développé le backend de l’application des enseignants ?

Le temps est venu de vous développer le frontend en Vue.js 3 – API Composition.

Vous devez implémenter :

* La page d’accueil
* La page du détail d’un enseignant (icone de la loupe)
* La page permettant d’ajouter un enseignant
* La suppression d’un enseignant

Lors de l’ajout et de la suppression d’un enseignant, une redirection sur la Homepage doit être implémentée.

A noter que pour l’ajout d’un enseignant, la validation des données n’est pas demandée.

Le code de l’API REST vous est fourni.

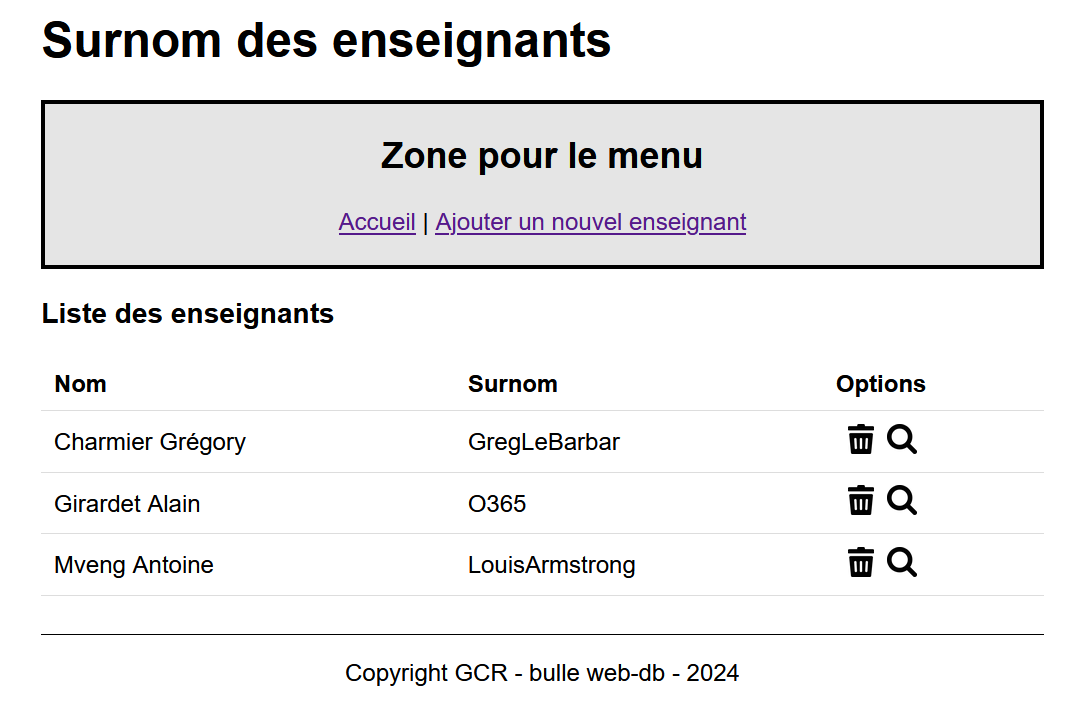
Un code de départ pour l’application Vue.js vous est également fourni

**Sources** :

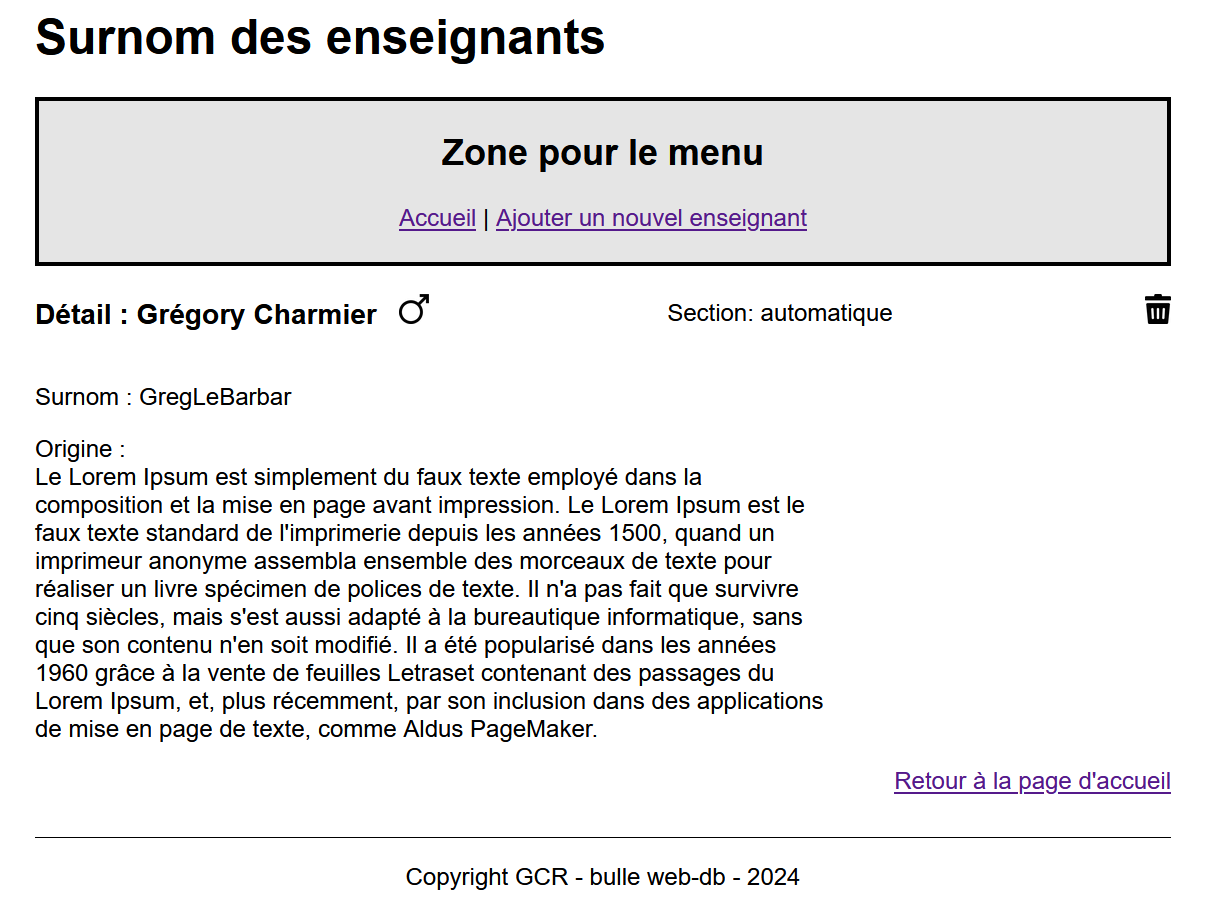
Vous trouverez dans le répertoire **Sources** :

* un code de départ de l’application frontend en Vue.js
* le backend de l’application en Node.js / Express.js

## La page d’accueil :



## La page du détail d’un enseignant



## La page d’ajout d’un enseignant

