

# Projet

# ~/chatbot-io

## 1/ Context

Vous devez réaliser une application de type Chatbot en utilisant le frameworks css [bootstrap](#) et vous devez utiliser **javascript Vanilla** pour la réalisation du projet.

***Vanilla js veut dire que votre projet ne peut pas être réalisé avec un frameworks javascript et qu'il doit être intégralement réalisé en javascript natif.***

## 2/ Le rendu attendu

- Un repo gitlab ~/chatbot-io/
- Une gestion des classes à l'intérieur de votre projet en **POO** et une bonne programmation fonctionnelle. Vous êtes libre d'utiliser tout les concepts vu pendant le cours.
- Vous devez utiliser l'environnement suivant :
  - 1 / Un environnement disposant d'un bundler qui vous allez pouvoir exécuter avec nodejs disponible [à cette adresse](#) (pour configurer l'environnement et utiliser node 20 vous devez lire [ce cours](#) du slide 15 à 21 et configurer votre machine pour vous permettre de lancer votre environnement avec [npm cli](#)).
- La réalisation de toutes les fonctionnalités sont attendus :
  - La liste des contacts (les bots)
  - Vous devez au moins générer minimum 3 bots
  - Chaque bot devra au minimum avoir 3 actions différentes déclenchées par un mot ou un groupement de commande que vous entrez dans la barre des messages que vous envoyer. Bien entendu si aucune action est paramétrée vous devrez ne rien envoyer. Petit conseil, une commande help pour voir les commandes disponible serait grandement appréciée.
  - Les 3 bots auront au moins une action en commun (qui lorsque qu'elle est déclencher permet de faire parler tous vos bots)
  - Les 3 bots devront appeler pour chacun d'eux au moins une api d'un web services online. (<https://developer.uber.com/docs/drivers/introduction>) utiliser [fetch](#) ou [axios](#) par exemple).
  - Vous devez d'utiliser l'API [ChatGpt](#) pour gérer une commande spéciale qui permet de poser des questions à tous vos bots. (vous devez préciser le nom du bot qui va vous répondre)
  - L'intégralité du système de l'envoi d'un message devra être réalisé avec les événements du clavier (touche entrée, clique sur un bouton envoyer).
  - Vous devrez avoir une liste des messages reçus et envoyés.
    - Dans chaque message devra figurer : l'heure, l'avatar, le nom et le message.

- Les message reçu seront à gauche et les messages que vous envoyez seront à droite.
- Vous devrez également réaliser la barre qui vous servira à écrire vos messages et à les envoyer :
  - Vous devez permettre l'envoi de message avec deux possibilités d'actions :
    - La touche entrée
    - Le bouton envoyer
- Pour la partie historique des messages vous devrez faire en sorte que la scrollbar remonte sur le message le plus récent si la totalité de l'historique des messages est complète.
- Vous serez également noté sur le soin apporté à votre application avec l'interface que vous aurez réalisé avec le framework css.
- Vous devez faire en sorte d'utiliser le [localStorage](#) du navigateur pour stocker les messages temporairement pour éviter qu'ils ne soient supprimer au rafraichissement de la page.
- Exemple du rendu attendu bon courage :

