Voici un résumé détaillé et structuré des étapes pour réaliser ce projet, en commençant par les installations nécessaires, et en terminant avec la création du chatbot interactif avec Gradio. Prérequis : avoir un PC avec une carte graphique dédiée.

# Installation de CUDA et cuDNN

## **1.1 Télécharger et installer CUDA**

* **Télécharger CUDA** : Va sur le [site officiel de NVIDIA](https://developer.nvidia.com/cuda-toolkit) et télécharge la version compatible avec ton GPU, par exemple CUDA 12.x.
* **Installer CUDA** : Suis les instructions spécifiques à ton système d'exploitation (Windows, Linux, macOS).

## **1.2 Télécharger et installer cuDNN**

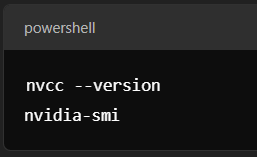
* **Télécharger cuDNN** : Va sur le [site officiel de cuDNN](https://developer.nvidia.com/cudnn) et télécharge la version compatible avec CUDA 12.
* **Décompresser le fichier cuDNN** : Une fois le fichier téléchargé (généralement un fichier .tar.gz ou .zip), décompresse-le.
* **Copier les fichiers cuDNN dans CUDA** : Copie les fichiers bin, lib, et include depuis le dossier décompressé de cuDNN vers le répertoire d'installation de CUDA. Cela permet de lier cuDNN à CUDA.

## **1.3 Vérifier l’installation**

Vérifie l’installation en exécutant dans Powershell

nvcc --version

nvidia-smi



# Installation **de Python et des bibliothèques nécessaires**

[Télécharger Python ici](https://www.python.org/downloads/)

* Prends la **dernière version stable de Python 3.10 ou 3.11** (évite Python 3.12 car certaines bibliothèques peuvent ne pas être compatibles).
* Si tu es sous **Windows**, télécharge l’installateur **Windows x86-64 executable installer**.

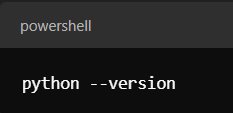
✅ **Étape 1** : Exécute l’installateur.  
✅ **Étape 2** : **Coche "Add Python to PATH"** en bas de la fenêtre d’installation.  
✅ **Étape 3** : Clique sur **"Install Now"**.

📌 **Astuce** : Pour une installation propre, tu peux aussi cliquer sur **"Customize installation"** et vérifier que pip et venv sont sélectionnés.

## **2.1 Vérifier si Python est installé**

* Ouvre **PowerShell** et tape :

python –version



* Si Python n’est pas installé, télécharge-le depuis : [🔗 Télécharger Python](https://www.python.org/downloads/)
* **Pendant l'installation**, coche **Add Python to PATH**.

## **2.2 Activer un environnement virtuel (FACULTATIF)**

Un environnement virtuel permet d’éviter des conflits par la suite.

* **Crée un dossier sur ton bureau**
* **Ouvre PowerShell**
* **Navigue dans le dossier de ton projet** (par exemple, C:\Users\[Nom\_Utilisateur]\Desktop\[Nom\_du\_dossier]), où tu veux stocker ton environnement virtuel.



* Crée un environnement virtuel **dans le dossier de ton projet en exécutant cette**

python -m venv myenv

Une image contenant texte, Police, capture d’écran, blanc

Description générée automatiquement

* Cette commande crée un dossier appelé **myenv** qui contient un environnement isolé pour ton projet.

## **2.4 Activer l’environnement virtuel**

Si tu es sous **Windows**, exécute cette commande pour **activer** l’environnement virtuel :

myenv\Scripts\activate

Une image contenant texte, Police, blanc, capture d’écran

Description générée automatiquement

Attention il faudra rouvrir systématiquement ton environnement si tu veux exécuter ton agent ia / ton chatbot.

## **2.4 Désactiver ton environnement virtuel**

A faire seulement a la fin de ta session.

* Si tu es sous **Windows**, exécute cette commande pour **activer** l’environnement virtuel :

myenv\Scripts\activate

Une image contenant texte, Police, capture d’écran, blanc

Description générée automatiquement

Une fois activé, tu verras quelque chose comme :

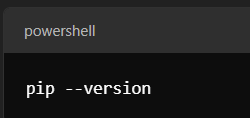
Une image contenant texte, capture d’écran, Police, blanc

Description générée automatiquement

## **2.5 Installer pip**

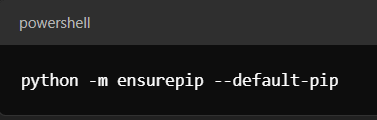
* Vérifie si **pip** est installé :

pip –version



* Si pip n’est pas reconnu, exécute :

python -m ensurepip --default-pip



# Installation **d'Ollama**

## **3.1 Télécharger Ollama**

* Télécharge Ollama depuis leur site officiel [Ollama](https://ollama.com/).

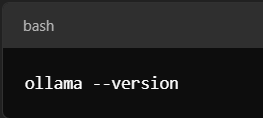
## **3.2 Installer Ollama**

* Exécute le fichier d’installation d'Ollama téléchargé et suis les instructions.

## **3.3 Vérifier l'installation d'Ollama**

* Ouvre PowerShell ou CMD et tape la commande suivante pour vérifier que Ollama est bien installé :

ollama –version

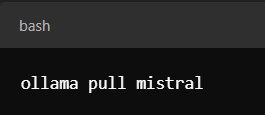


# Téléchargement **et exécution du modèle Mistral**

## **4.1 Télécharger le modèle Mistral**

* Pour télécharger le modèle Mistral, exécute cette commande dans ton terminal (Ollama doit être installé et configuré) :

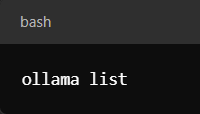
ollama pull mistral



## **4.2 Vérifier les modèles téléchargés**

* Pour lister les modèles disponibles et vérifier si Mistral a bien été téléchargé, utilise cette commande :

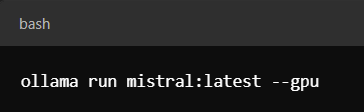
ollama list



## **4.3 Exécuter le modèle Mistral avec GPU**

* Pour exécuter le modèle Mistral avec l’accélération GPU (si tu utilises une carte graphique comme la RTX 3090), lance la commande suivante :

ollama run mistral:latest –gpu

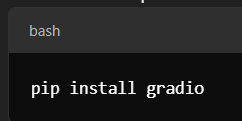


# Créer **un chatbot interactif avec Gradio**

## **5.1 Installer Gradio**

* Installe Gradio pour créer une interface utilisateur interactive avec la commande :

pip install gradio



# **Créer un fichier Python pour le chatbot**

* Crée un fichier Python, par exemple chatbot.py, et colle le code suivant pour initialiser le modèle Mistral et créer une fonction qui répond aux questions :

## **6.1 créer ou copier/coller le fichier python chatbot.py dans le dossier**

* Copier/coller le fichier chatbot .py dans le dossier sur votre bureau
* Lancer Powershell
* Aller dans le dossier ou est hébergé le fichier prompt python :

|  |
| --- |
| cd C:\Users\[Nom\_utilisateur]\Desktop\[Nom\_du\_dossier] |



## **6.2 Nourrir l'IA avec un fichier CSV pour alimenter le modèle (FACULTATIF – cela dépend de votre projet)**

Pour que l'IA puisse utiliser des informations spécifiques dans un fichier CSV (par exemple un règlement ou des règles), tu devras :

* Charger et parser un fichier CSV avec des informations pertinentes.
* Adapter l'agent pour qu’il puisse répondre en se basant sur ces données spécifiques, tout en utilisant le modèle Ollama pour les cas où les informations ne sont pas trouvées dans le CSV.

Un exemple basique pour charger un fichier CSV dans Python :

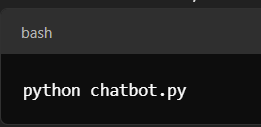
|  |
| --- |
| import pandas as pd  # Charger le fichier CSV  df = pd.read\_csv('chemin/vers/ton/fichier.csv')  # Exemple de recherche dans le CSV (trouver une ligne où une colonne correspond à une question)  def search\_in\_csv(question):  result = df[df['column\_name'].str.contains(question, case=False, na=False)]  return result if not result.empty else "Aucune correspondance trouvée dans le fichier." |



Tu peux ensuite combiner cette recherche avec le modèle Ollama pour offrir une réponse plus précise en fonction des données dans ton fichier CSV.

## **6.3 Lancer le chatbot**

* Pour lancer le chatbot, exécute ce fichier Python avec la commande suivante :



Cela ouvrira une interface web dans ton navigateur où tu pourras poser des questions et interagir avec le modèle Mistral via Ollama.

Et cliquer sur le lien indiqué ou aller sur : <http://127.0.0.1:7862> pour interagir avec votre Chatbot !

# **Conclusion**

Tu as maintenant une vue d'ensemble des étapes pour configurer ton environnement et créer un chatbot interactif avec un modèle d'IA basé sur Ollama. La partie importante de ce projet réside dans l'intégration du modèle Ollama avec Gradio, tout en utilisant les outils de traitement de texte comme les fichiers CSV pour alimenter le modèle de données spécifiques.

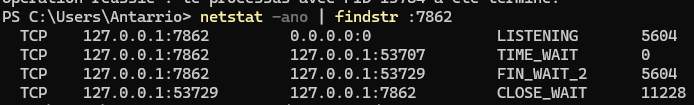
# **Nota Bene**

Dans POWERSHELL si besoin penser à « tuer » le processus entre chaque essaie en faisant :

CTLR + C

Le cas échéant vous pouvez vérifier si vous avez bien arrêter le chatbot  :

|  |
| --- |
| netstat -ano | findstr :7862 |



Indiquer ensuite le processus à tuer s’il ne s’est pas coupé

|  |
| --- |
| taskkill /PID 5604 /F  taskkill /PID 11228 /F |

