

```
!pip install langchain langchain-community pypdf chromadb -q
!pip install langchain_groq -q
!pip install -U langchain-huggingface -q
!pip install -U langchain-chroma -q
!pip install gradio -q
```

ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is changing in pip 24.3. langchain-groq 0.3.8 requires langchain-core<1.0.0,>=0.3.75, but you have langchain-core 1.0.7 which is incompatible. langchain 0.3.27 requires langchain-core<1.0.0,>=0.3.72, but you have langchain-core 1.0.7 which is incompatible.

```
!pip install langchain langchain-core langchain-community langchain-groq chromadb sentence-transformers
```

```
Requirement already satisfied: flatbuffers in /usr/local/lib/python3.12/dist-packages (from onnxruntime==1.14.1->chromadb)
Requirement already satisfied: protobuf in /usr/local/lib/python3.12/dist-packages (from onnxruntime==1.14.1->chromadb) (5
Requirement already satisfied: sympy in /usr/local/lib/python3.12/dist-packages (from onnxruntime==1.14.1->chromadb) (1.13
Requirement already satisfied: importlib-metadata<8.8.0,>=6.0 in /usr/local/lib/python3.12/dist-packages (from opentelemet
Requirement already satisfied: googleapis-common-protos~=1.57 in /usr/local/lib/python3.12/dist-packages (from opentelemet
Requirement already satisfied: opentelemetry-exporter-otlp-proto-common==1.38.0 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: opentelemetry-proto==1.38.0 in /usr/local/lib/python3.12/dist-packages (from opentelemetry
Requirement already satisfied: opentelemetry-semantic-conventions==0.59b0 in /usr/local/lib/python3.12/dist-packages (from
Requirement already satisfied: backoff>=1.10.0 in /usr/local/lib/python3.12/dist-packages (from posthog<6.0.0,>=2.4.0->chr
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.12/dist-packages (from pydantic<3.0.0,>=2.
Requirement already satisfied: pydantic-core==2.33.2 in /usr/local/lib/python3.12/dist-packages (from pydantic<3.0.0,>=2.7
Requirement already satisfied: typing-inspection>=0.4.0 in /usr/local/lib/python3.12/dist-packages (from pydantic<3.0.0,>=
Requirement already satisfied: python-dotenv>=0.21.0 in /usr/local/lib/python3.12/dist-packages (from pydantic-settings<3.
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests<3,>=2->l
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.12/dist-packages (from rich>=10.11.0->chrom
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.12/dist-packages (from rich>=10.11.0->chr
Requirement already satisfied: greenlet>=1 in /usr/local/lib/python3.12/dist-packages (from SQLAlchemychemy<3,>=1.4->langchain)
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-packages (from torch>=1.11.0->sentence-transfo
Requirement already satisfied: networkx>=2.5.1 in /usr/local/lib/python3.12/dist-packages (from torch>=1.11.0->sentence-tr
Requirement already satisfied: Jinja2 in /usr/local/lib/python3.12/dist-packages (from torch>=1.11.0->sentence-transformer
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch>=1.1
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch>=1
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.6.80 in /usr/local/lib/python3.12/dist-packages (from torch>=1.1
Requirement already satisfied: nvidia-cudnn-cu12==9.10.2.21 in /usr/local/lib/python3.12/dist-packages (from torch>=1.11.0
Requirement already satisfied: nvidia-cublas-cu12==12.6.4.1 in /usr/local/lib/python3.12/dist-packages (from torch>=1.11.0
Requirement already satisfied: nvidia-cufft-cu12==11.3.0.4 in /usr/local/lib/python3.12/dist-packages (from torch>=1.11.0-
Requirement already satisfied: nvidia-curand-cu12==10.3.7.77 in /usr/local/lib/python3.12/dist-packages (from torch>=1.11.
Requirement already satisfied: nvidia-cusolver-cu12==11.7.1.2 in /usr/local/lib/python3.12/dist-packages (from torch>=1.11
Requirement already satisfied: nvidia-cusparselt-cu12==12.5.4.2 in /usr/local/lib/python3.12/dist-packages (from torch>=1.11
Requirement already satisfied: nvidia-cusparse-cu12==12.5.4.2 in /usr/local/lib/python3.12/dist-packages (from torch>=1.11
Requirement already satisfied: nvidia-nvtx-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch>=1.11.0->s
Requirement already satisfied: nvidia-nvjitlink-cu12==12.6.85 in /usr/local/lib/python3.12/dist-packages (from torch>=1.11
Requirement already satisfied: nvidia-cufile-cu12==1.11.1.6 in /usr/local/lib/python3.12/dist-packages (from torch>=1.11.0
Requirement already satisfied: triton==3.5.0 in /usr/local/lib/python3.12/dist-packages (from torch>=1.11.0->sentence-tran
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.12/dist-packages (from transformers<5.0.0,>=4.4
Requirement already satisfied: safetensors>=0.4.3 in /usr/local/lib/python3.12/dist-packages (from transformers<5.0.0,>=4.
Requirement already satisfied: click>=8.0.0 in /usr/local/lib/python3.12/dist-packages (from typer>=0.9.0->chromadb) (8.3.
Requirement already satisfied: shellingham>=1.3.0 in /usr/local/lib/python3.12/dist-packages (from typer>=0.9.0->chromadb)
Requirement already satisfied: httpxtools>=0.6.3 in /usr/local/lib/python3.12/dist-packages (from uvicorn[standard]>=0.18.3
Requirement already satisfied: uvloop>=0.15.1 in /usr/local/lib/python3.12/dist-packages (from uvicorn[standard]>=0.18.3->
Requirement already satisfied: watchfiles>=0.13 in /usr/local/lib/python3.12/dist-packages (from uvicorn[standard]>=0.18.3
Requirement already satisfied: websockets>=10.4 in /usr/local/lib/python3.12/dist-packages (from uvicorn[standard]>=0.18.3
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.12/dist-packages (from scikit-learn->sentence-trans
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.12/dist-packages (from scikit-learn->sentence-trans
Requirement already satisfied: cachetools<6.0,>=2.0.0 in /usr/local/lib/python3.12/dist-packages (from google-auth>=1.0.1->
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.12/dist-packages (from google-auth>=1.0.1->
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.12/dist-packages (from google-auth>=1.0.1->kubernetes
Requirement already satisfied: zipp>=3.20 in /usr/local/lib/python3.12/dist-packages (from importlib-metadata<8.8.0,>=6.0-
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.12/dist-packages (from markdown-it-py>=2.2.0->rich>=10
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.12/dist-packages (from sympy->onnxruntime==1.1
Requirement already satisfied: mpyy-extensions>=0.3.0 in /usr/local/lib/python3.12/dist-packages (from typing-inspect<1,>=
Requirement already satisfied: humanfriendly>=9.1 in /usr/local/lib/python3.12/dist-packages (from coloredlogs->onnxruntime
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.12/dist-packages (from Jinja2->torch>=1.11.0->sentence-transformers)
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.12/dist-packages (from requests-oauthlib->kubernetes)
Requirement already satisfied: pyasn1<0.7.0,>=0.6.1 in /usr/local/lib/python3.12/dist-packages (from pyasn1-modules>=0.2.1
Using cached langchain-core-0.3.80-nv3-none-any.whl (450 kB)
```

```
# Importations
from langchain.document_loaders import PyPDFLoader
from langchain.text_splitter import RecursiveCharacterTextSplitter
from langchain.embeddings import HuggingFaceEmbeddings
from langchain.vectorstores import Chroma
from langchain.chains import RetrievalQA
from langchain_groq import ChatGroq
from langchain.chains import RetrievalQA
from langchain.prompts import PromptTemplate

import google.generativeai as genai
```

```
from groq import Client # Assurez-vous d'utiliser le bon SDK Groq
```

```
import os
import gradio as gr
# Définition du chemin d
```

```
# Définition du chemin du fichier PDF
pdf_path = "/content/Journal0132025.pdf"
```

```
# Chargement du document PDF
loader = PyPDFLoader(pdf_path)
```

```
# Extraction des pages
pages = loader.load()
```

```
# Vérification du nombre de pages chargées
print(f"Nombre de pages chargées : {len(pages)}")
```

```
# Affichage du contenu de la première page pour contrôle
print(pages[0].page_content[:500]) # Affiche les 500 premiers caractères de la première page
```

```
Nombre de pages chargées : 17
TRADUCTION FRANÇAISE POUR INFORMATION
```

```
Sommaire
Décrets et arrêtés
```

```
Présidence du Gouvernement
Arrêté du Chef du Gouvernement du 30 janvier 2025, portant organisation et
ouverture d'une session de formation au profit des hauts cadres
administratifs à l'Institut de leadership administratif à l'Ecole nationale
d'administration pour l'année 2025 .....
Nomination de contrôleurs généraux des dépenses publiques ...
```

```
# Définition du découpeur de texte
text_splitter = RecursiveCharacterTextSplitter(
    chunk_size=500, # Taille de chaque chunk
    chunk_overlap=50, # Chevauchement entre les chunks pour conserver le contexte
    separators=["\n\n", "\n", " ", ""], # Séparateurs pour découper propremen
)
```

```
# Découpage des pages en chunks
docs = text_splitter.split_documents(pages)
```

```
# Vérification
print(f"Nombre total de chunks créés : {len(docs)}")
print("\nExemple d'un chunk :\n")
print(docs[0].page_content) # Afficher un exemple de chunk
```

```
Nombre total de chunks créés : 110
```

```
Exemple d'un chunk :
```

```
TRADUCTION FRANÇAISE POUR INFORMATION
```

```
Sommaire
Décrets et arrêtés
```

```
Présidence du Gouvernement
Arrêté du Chef du Gouvernement du 30 janvier 2025, portant organisation et
ouverture d'une session de formation au profit des hauts cadres
administratifs à l'Institut de leadership administratif à l'Ecole nationale
d'administration pour l'année 2025 .....
```

```
import os
os.environ["OPENAI_API_KEY"] = "sk-proj-mHvskn53BAgNiMEbfkbpJJEa-8Auo18Qslgsi3uecQ6e3WLC4IR-rIJBCAftyL2swToecwGI41T3B1bkFJz"
```

```
# Chemin où sauvegarder la base de données
persist_directory = "/kaggle/working/chroma_db"
```

```
# Chargement du modèle d'embeddings open-source
embedding_function = HuggingFaceEmbeddings(model_name="sentence-transformers/all-MiniLM-L6-v2")

# Vérifier si une base Chroma existe déjà
if os.path.exists(persist_directory):
    print("Chargement de la base Chroma existante...")
    vectorstore = Chroma(persist_directory=persist_directory, embedding_function=embedding_function)
else:
    print("Création d'une nouvelle base Chroma...")
    vectorstore = Chroma.from_documents(docs, embedding_function, persist_directory=persist_directory)
    #vectorstore.persist() # Sauvegarde de la base

# Vérification
print(f"Nombre de documents indexés : {vectorstore._collection.count()}")
```

```
/tmp/ipython-input-1062372533.py:5: LangChainDeprecationWarning: The class `HuggingFaceEmbeddings` was deprecated in LangChain
embedding_function = HuggingFaceEmbeddings(model_name="sentence-transformers/all-MiniLM-L6-v2")
/usr/local/lib/python3.12/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
warnings.warn(
modules.json: 100% 349/349 [00:00<00:00, 7.12kB/s]

config_sentence_transformers.json: 100% 116/116 [00:00<00:00, 3.78kB/s]

README.md: 10.5k/? [00:00<00:00, 287kB/s]

sentence_bert_config.json: 100% 53.0/53.0 [00:00<00:00, 3.68kB/s]

config.json: 100% 612/612 [00:00<00:00, 17.9kB/s]

model.safetensors: 100% 90.9M/90.9M [00:01<00:00, 72.4MB/s]

tokenizer_config.json: 100% 350/350 [00:00<00:00, 8.84kB/s]

vocab.txt: 232k/? [00:00<00:00, 4.54MB/s]

tokenizer.json: 466k/? [00:00<00:00, 7.42MB/s]

special_tokens_map.json: 100% 112/112 [00:00<00:00, 3.96kB/s]

config.json: 100% 190/190 [00:00<00:00, 5.49kB/s]
Création d'une nouvelle base Chroma...
Nombre de documents indexés : 110
```

```
os.environ["GROQ_API_KEY"] = "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" # C'est ma clé, mettez la votre à
genai.configure(api_key=os.environ["GROQ_API_KEY"])

api_key = os.getenv("GROQ_API_KEY")

# Initialisation du client Groq avec la clé API depuis l'environnement
api_key = os.getenv("GROQ_API_KEY")
if not api_key:
    raise ValueError("La clé API GROQ_API_KEY n'est pas définie.")

client = Client(api_key=api_key)
```

```
# Initialisation du modèle Groq
groq_llm = ChatGroq(model="llama-3.3-70b-versatile")

# Création du prompt pour Groq basé sur le JORT Tunisien
prompt_template = (
    "Vous êtes un expert en droit tunisien et vous devez répondre avec précision "
    "en vous basant exclusivement sur le Journal Officiel de la République Tunisienne (JORT). "
    "Voici le contexte juridique extrait du JORT : {context} \n\n"
    "Question : {question} \n\n"
    "Fournissez une réponse claire, précise et conforme aux textes législatifs et réglementaires tunisiens."
)

# Définition de la chaîne QA avec le retriever
qa_chain = RetrievalQA.from_chain_type(
    llm=groq_llm,
    chain_type="stuff",
    retriever=vectorstore.as_retriever(),
    return_source_documents=True
)
```

```
# Exemple de question
question = "Pouvez-vous résumer les dernières réformes législatives publiées dans le JORT ?"

# Utilisation de `run()` pour obtenir la réponse et les documents sources
response = qa_chain.invoke({"query": question})

# Extraction des résultats
answer = response['result']
source_documents = response['source_documents']

# Affichage de la réponse et des documents sources
print(f"Réponse : {answer}")
print("Documents sources :")
for doc in source_documents:
    print(doc.page_content) # Assurez-vous que vos documents ont l'attribut `page_content`
```

Réponse : Je ne peux pas résumer les dernières réformes législatives publiées dans le Journal Officiel de la République Tun

Cependant, je peux vous indiquer que les textes fournis font référence à différentes lois et décrets, notamment :

- \* La loi n° 64-44 du 3 novembre 1964, portant réforme de l'école nationale d'administration
- \* La loi n° 83-112 du 12 décembre 1983, portant statut général des personnels de l'Etat
- \* La loi n° 2003-69 du 20 octobre 2003, relative aux centres de vacances d'été et de loisirs pour enfants
- \* Le décret n° 2004-2579 du 2 novembre 2004, portant création du centre 7 novembre de vacances d'été et de loisirs pour enf
- \* L'arrêté du Chef du Gouvernement du 30 janvier 2025, portant organisation et ouverture d'une session de formation au profi
- \* Le décret n° 90-999 du 11 juin 1990, fixant les attributions du ministère des domaines de l'Etat

Il est possible que ces textes soient liés à des réformes législatives récentes, mais je ne peux pas fournir plus d'informat

Documents sources :

Vu la loi n° 64-44 du 3 novembre 1964, portant  
réforme de l'école nationale d'administration,  
Vu la loi n° 83-112 du 12 décembre 1983, portant  
statut général des personnels de l'Etat, des collectivités  
locales et des établissements publics à caractère  
administratif, et l'ensemble des textes qui l'ont  
modifiée ou complétée.  
Vu le décret n° 2007-1885 du 23 juillet 2007,  
fixant l'organisation administrative et financière de  
l'Ecole nationale d'administration et l'ensemble des  
Vu la Constitution,  
Vu la loi n° 2003-69 du 20 octobre 2003, relative aux centres de vacances d'été et de loisirs pour enfants, telle  
que modifiée par la loi n° 2024-24 du 10 mai 2024,  
Vu le décret n° 2004-2579 du 2 novembre 2004, portant création du centre 7 novembre de vacances d'été et de  
loisirs pour enfants à Hammamet et fixant son organisation administrative et financière et les modalités de son  
fonctionnement,  
Vu l'avis du Tribunal administratif,  
TRADUCTION FRANÇAISE POUR INFORMATION

#### Sommaire

Décrets et arrêtés

#### Présidence du Gouvernement

Arrêté du Chef du Gouvernement du 30 janvier 2025, portant organisation et  
ouverture d'une session de formation au profit des hauts cadres  
administratifs à l'Institut de leadership administratif à l'Ecole nationale  
d'administration pour l'année 2025 .....  
social.  
Le Président de la République,  
Vu la Constitution,  
Vu la loi n° 2012-1 du 16 mai 2012, portant loi de finances complémentaire pour l'année 2012, notamment son  
article 30, telle que modifiée par le décret-loi n° 21-2021 du 28 décembre 2021, portant loi de finances pour l'année  
2022,  
Vu le décret n° 90-999 du 11 juin 1990 fixant les attributions du ministère des domaines de l'Etat,  
Prend le décret dont la teneur suit :

```
sample_questions = [
    # Questions sur les lois et réglementations générales
    "Quelle est la dernière loi publiée dans le JORT sur le droit du travail ?",
    "Quelles sont les nouvelles mesures fiscales introduites cette année selon le JORT ?",
    "Quels sont les textes de loi régissant les marchés publics en Tunisie ?",
    "Existe-t-il des modifications récentes du Code des sociétés commerciales ?",
    "Quelle est la réglementation actuelle sur les importations en Tunisie selon le JORT ?",

    # Questions sur le droit du travail et social
    "Quelles sont les nouvelles dispositions du Code du travail publiées dans le JORT ?",
    "Quelle est la durée légale du travail en Tunisie selon les derniers textes officiels ?",
    "Quels sont les droits des employés en cas de licenciement selon le JORT ?",
```

```

"Y a-t-il eu des changements récents concernant le salaire minimum en Tunisie ?",
"Quelles sont les nouvelles mesures de protection sociale adoptées récemment ?",

# Questions sur les marchés publics et entreprises
"Quels sont les critères d'éligibilité pour répondre à un appel d'offres public en Tunisie ?",
"Quelle est la procédure pour enregistrer une nouvelle entreprise en Tunisie selon le JORT ?",
"Quels sont les avantages fiscaux pour les startups en Tunisie ?",
"Quelles sont les obligations légales des entreprises en matière de déclaration fiscale ?",

# Questions sur le droit immobilier et urbanisme
"Quelles sont les nouvelles lois régissant l'acquisition de biens immobiliers en Tunisie ?",
"Quels sont les critères pour bénéficier d'un lotissement social selon le JORT ?",
"Quelles sont les autorisations nécessaires pour construire un immeuble en Tunisie ?",
"Y a-t-il des modifications récentes dans la loi sur le bail commercial ?",

# Questions sur les taxes et impôts
"Quels sont les nouveaux taux d'imposition appliqués aux entreprises en Tunisie ?",
"Quels sont les délais de déclaration fiscale pour les particuliers et entreprises ?",
"Quelles exonérations fiscales sont prévues pour les investisseurs étrangers ?",
"Y a-t-il eu des modifications dans la TVA selon les derniers décrets du JORT ?",

# Questions sur les droits et obligations des citoyens
"Quels sont les droits du consommateur selon les dernières lois publiées ?",
"Comment obtenir une carte de séjour en Tunisie selon le JORT ?",
"Quelle est la réglementation concernant l'importation de véhicules en Tunisie ?",
"Quelles sont les nouvelles règles concernant le commerce électronique en Tunisie ?",

# Questions pour des résumés et synthèses
"Pouvez-vous résumer les dernières réformes législatives publiées dans le JORT ?",
"Quels sont les points clés du dernier décret gouvernemental concernant la fiscalité ?",
"Quelles sont les principales évolutions du droit du travail en Tunisie cette année ?",
"Faites-moi une synthèse des dernières modifications du Code des investissements."
]

```

```

# Fonction de réponse du chatbot
def chatbot(question):
    if not question.strip():
        return "Veuillez entrer une question."

    response = qa_chain.invoke({"query": question})
    answer = response['result']
    sources = "\n\n".join([doc.page_content for doc in response['source_documents']])

    return f"""Réponse : ** {answer} \n\n Sources : ** \n {sources} """

```

```

# Interface Gradio
with gr.Blocks() as interface:
    gr.Markdown("## Chatbot IA - LangChain & Groq\nPosez votre question ou choisissez-en une prédéfinie.")

    with gr.Row():
        question_input = gr.Textbox(placeholder="Posez votre question ici...")
        submit_btn = gr.Button("Envoyer")

    output = gr.Markdown()

    # Boutons pour les questions prédéfinies
    gr.Markdown("### Questions Prédéfinies")
    for q in sample_questions:
        gr.Button(q).click(fn=chatbot, inputs=gr.Textbox(value=q, visible=False), outputs=output)

    # Lien entre l'entrée manuelle et le chatbot
    submit_btn.click(chatbot, inputs=question_input, outputs=output)

# Lancement de l'interface
interface.launch(share=True)

```

Colab notebook detected. To show errors in colab notebook, set debug=True in launch()

\* Running on public URL: <https://78a28c6462b0185d39.gradio.live>

This share link expires in 1 week. For free permanent hosting and GPU upgrades, run `gradio deploy` from the terminal in the

Étapes suivantes :  [Déployer dans Cloud Run](#)

Commencez à coder ou à [générer](#) avec l'IA.