المدرسة الوطنية العليا للذكاء الاصطناعي وعلوم المعطيات - تارودانت
ÉCOLE NATIONALE SUPÉRIEURE DE L'INTELLIGENCE ARTIFICIELLE
ET SCIENCES DES DONNÉES - TAROUDANT

Rapport Projet Bases de données SQL et NoSQL

Première Année Cycle d'Ingénieur : Sécurité IT et Confiance Numérique

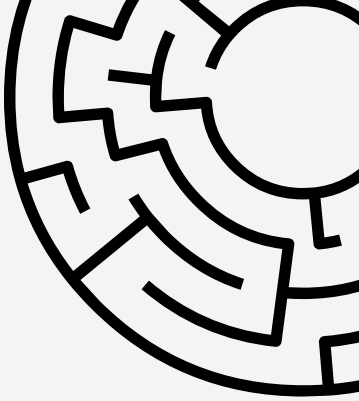
Sujet:
Carnet d'adresses avec MongoDB

Réalisé par :
- BOUTALMAOUINE MOHAMED
- AHOUARI BELAID

Encadré par :
Pr NASSIRI

Année Universitaire 2023-2024

CONTENU



INTRODUCTION

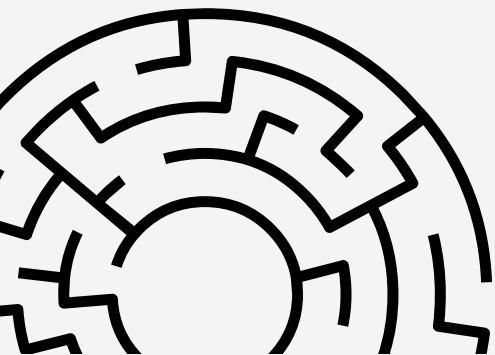
Dans un monde où la communication est essentielle, disposer d'un carnet d'adresses efficace est une nécessité pour rester organisé et connecté. Notre projet de carnet d'adresses avec MongoDB vise à fournir une solution simple mais puissante pour répondre à ce besoin.

Imaginez-vous avec un carnet d'adresses virtuel, où vous pouvez facilement stocker, gérer et retrouver les informations de contact de vos amis, collègues et membres de votre réseau. Ce carnet d'adresses numérique est non seulement pratique, mais aussi sécurisé et évolutif, grâce à l'utilisation de MongoDB comme base de données.

Avec notre application, vous pouvez créer de nouveaux contacts en quelques clics, en saisissant simplement leur nom, numéro de téléphone, adresse e-mail, etc. De plus, vous pouvez consulter votre liste de contacts existante à tout moment, et mettre à jour ou supprimer des contacts selon vos besoins.

L'une des fonctionnalités les plus utiles de notre application est la fonction de recherche, qui vous permet de trouver rapidement des contacts en fonction de leur nom, numéro de téléphone ou adresse e-mail. Que vous ayez besoin de retrouver le numéro d'un ami ou l'adresse e-mail d'un collègue, notre application vous offre une solution rapide et efficace.

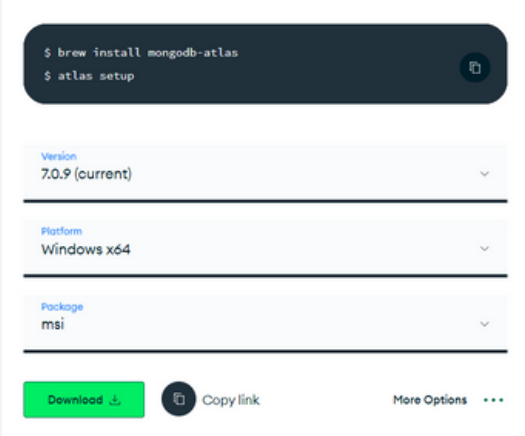
En combinant la simplicité d'utilisation de notre interface utilisateur avec la puissance de MongoDB, nous vous offrons un outil de gestion de contacts complet et fiable. Prêt à organiser vos contacts de manière intelligente et efficace ? Découvrez notre carnet d'adresses avec MongoDB dès aujourd'hui !



Installation de MongoDB

Pour l'installation de MongoDB dans un systèmed'exploitation Windows, nous avons suivi le un processus méthodique suivant :

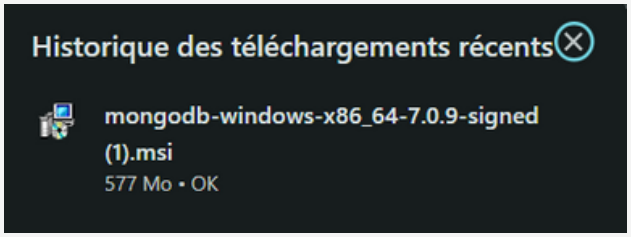
Etape 1 : accéder à la page officiel d'installation de MongoDB :
https://www.mongodb.com/try/download/community

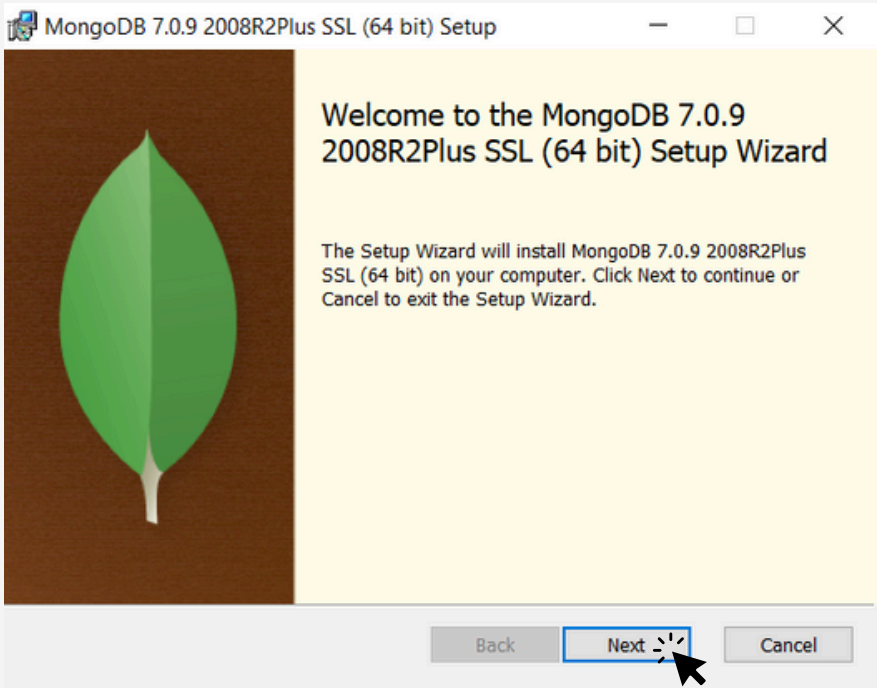


Etape 2 : Choisir la version appropriée. Pour notre installation, nous avons opté pour la dernière version de MongoDB compatible avec Windows (64 bits).

Pour installer la version que nous avons choisie, vous pouvez cliquer ici:
https://fastdl.mongodb.org/windows/mongodb-windows-x86_64-7.0.9-signed.msi

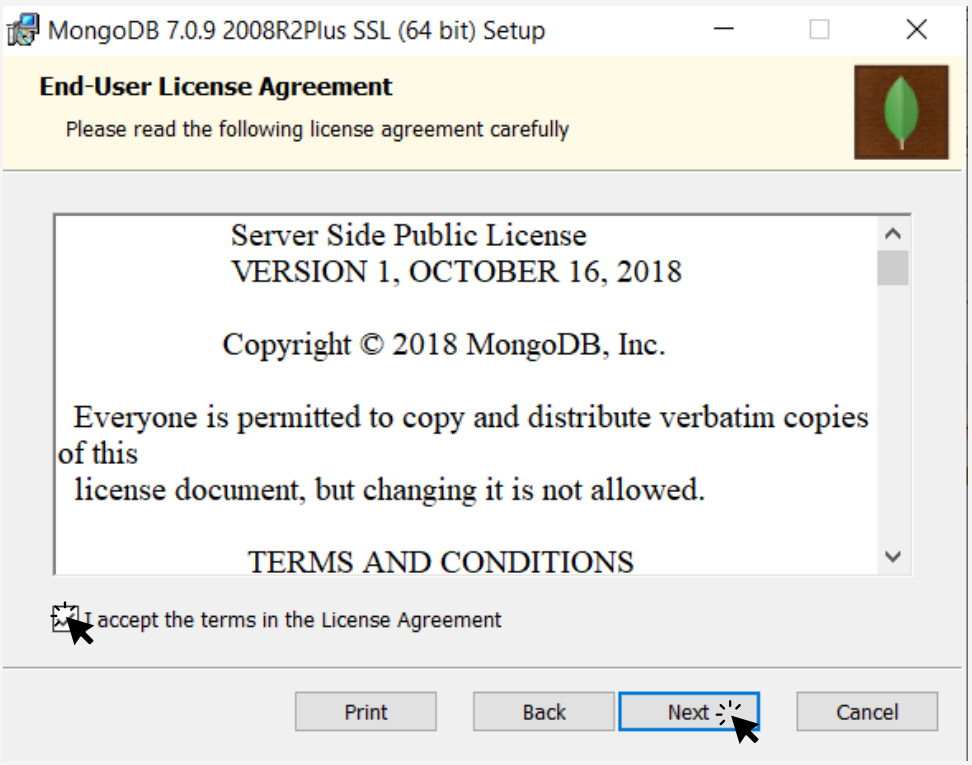
Etape 3 : Après le téléchargement de la version choisie, nous avons ouvert le fichier téléchargé :



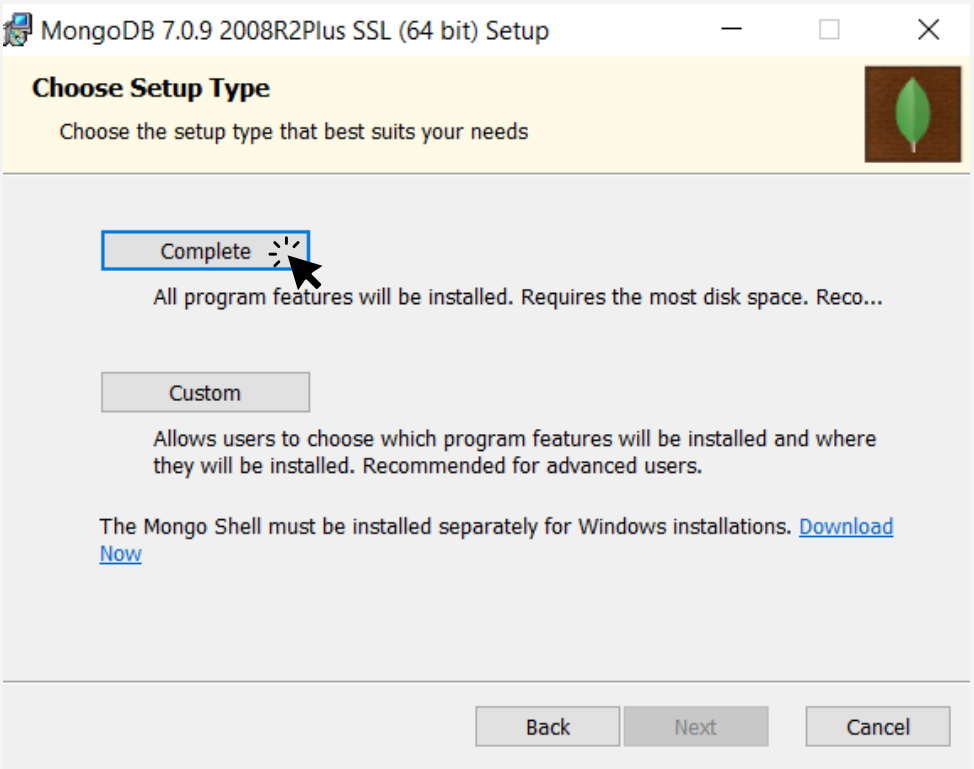


Installation de MongoDB

Etape 4 : Nous avons ensuite accepté les règles d'utilisation.

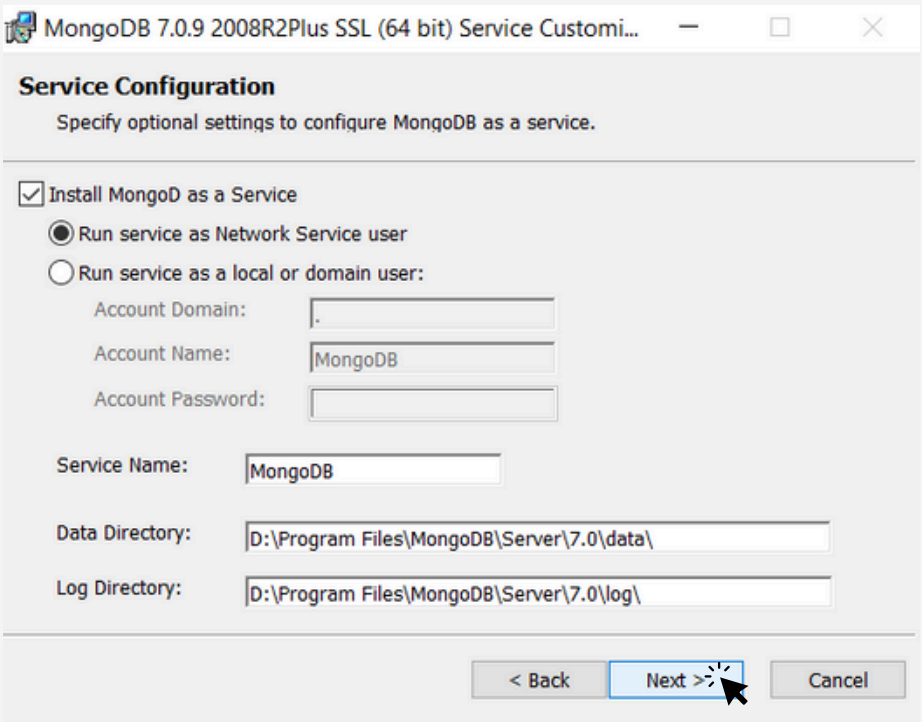


Etape 5 : Après avoir accepté les règles d'utilisation, nous avons sélectionné l'option pour installer la version complète.

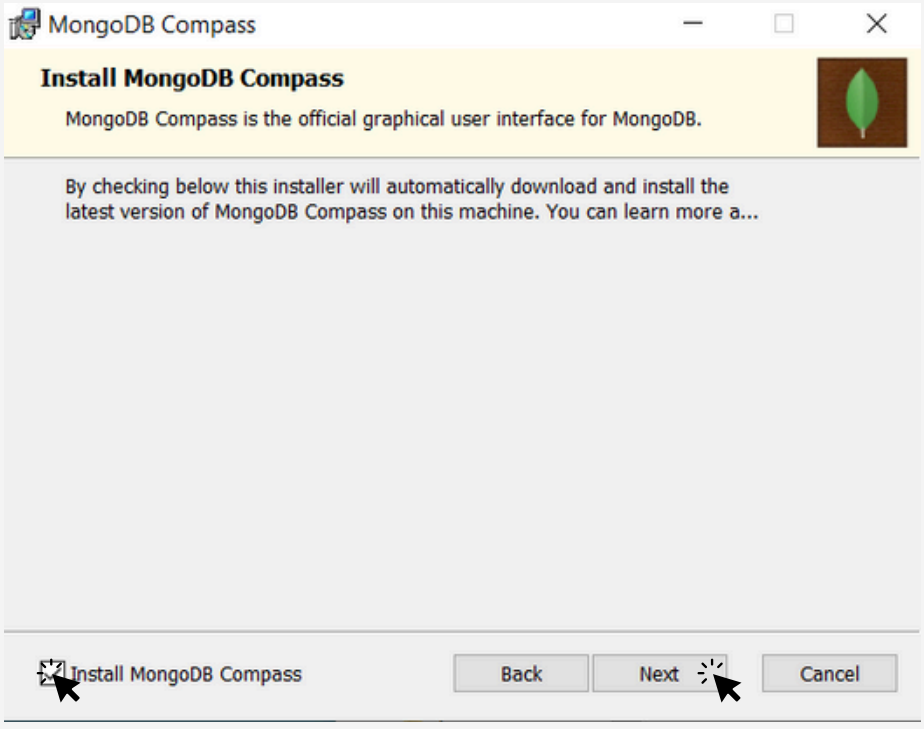


Installation de MongoDB

Etape 6 : Ensuite, nous avons choisi un chemin d'installation pour MongoDB.

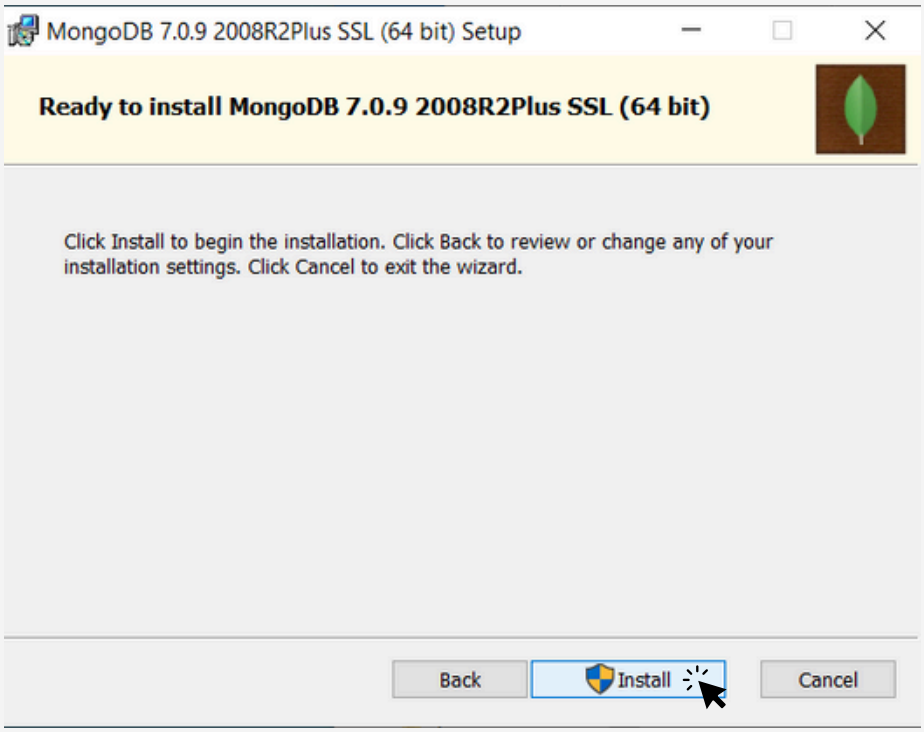


Etape 7 : Nous avons également opté pour l'installation de MongoDB Compass afin de faciliter l'utilisation de MongoDB.

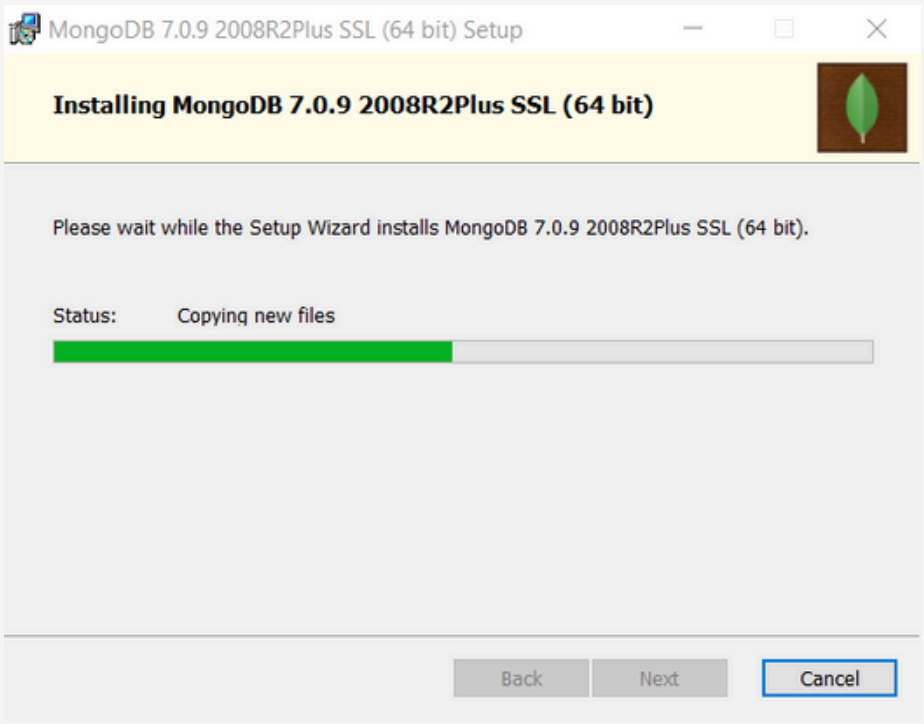


Installation de MongoDB

Etape 8 : Finalement, nous avons appuyé sur le bouton "Installer" pour lancer le processus d'installation.

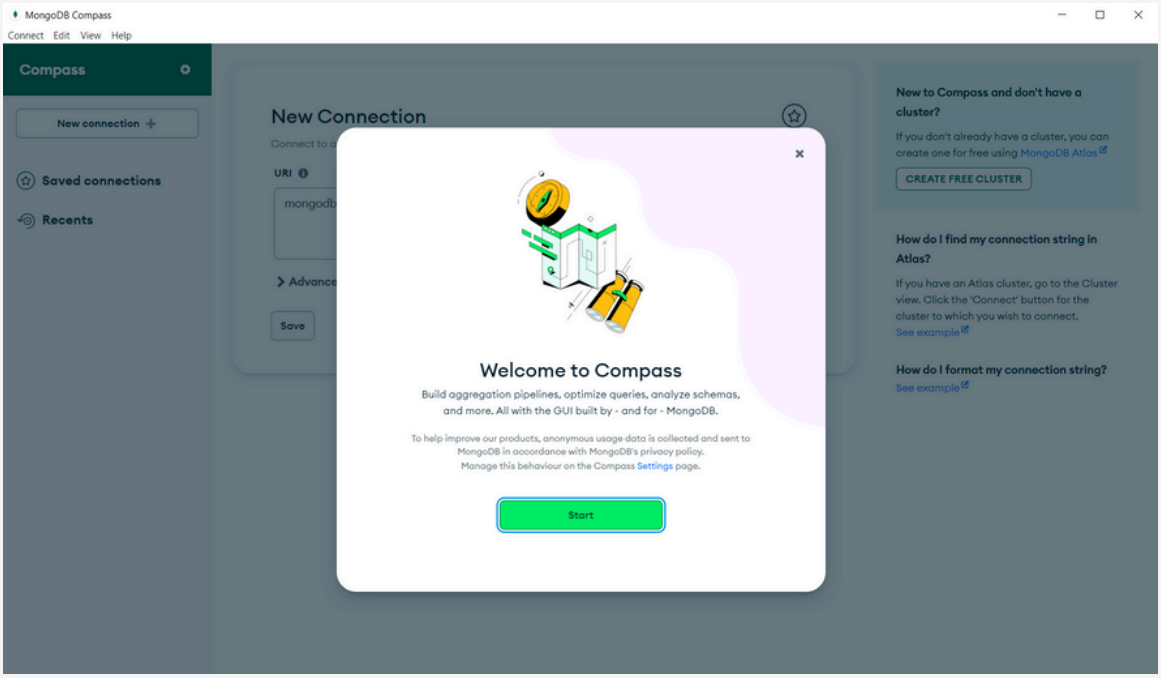


Etape 9 : Nous avons ensuite attendu que l'installation se termine.

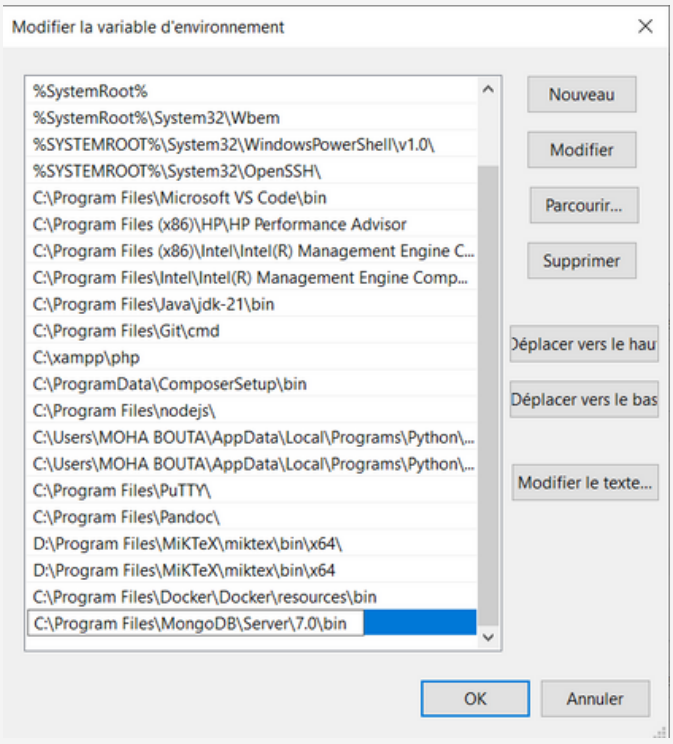


Installation de MongoDB

Etape 10 : Après l'installation, une fenêtre de MongoDB Compass s'est affichée, confirmant ainsi que l'installation s'est déroulée avec succès.

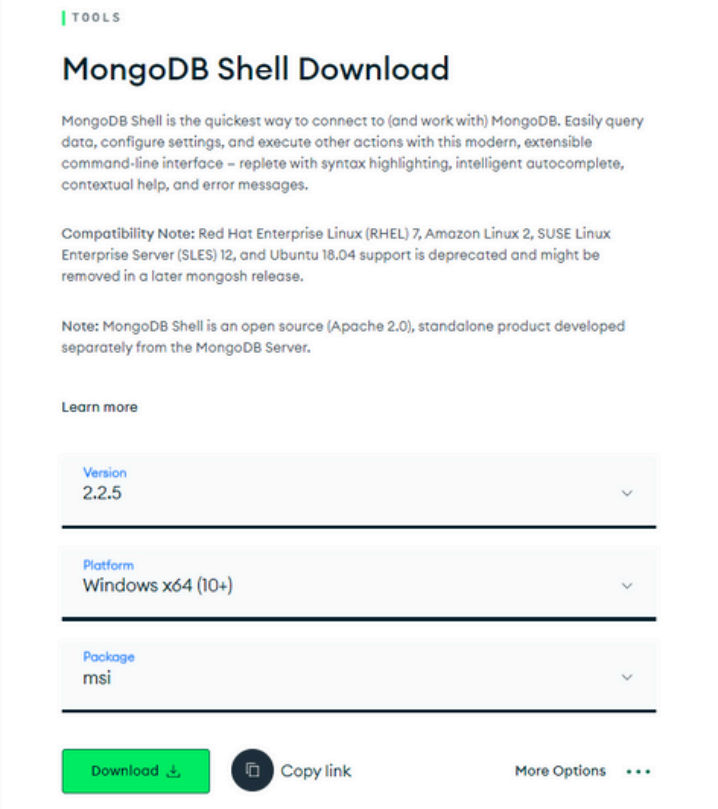


Etape 11 : Corriger les problèmes de chemin d'accès et d'environnement après l'installation de MongoDB

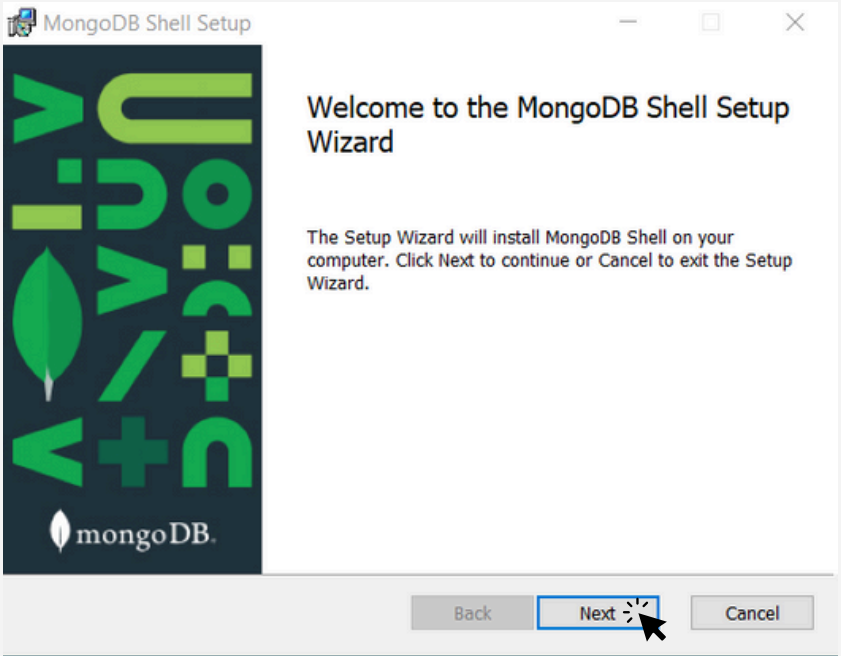


Installation de MongoDB

Etape 12 : Maintenant, nous devons installer le shell MongoDB.
lien : https://www.mongodb.com/try/download/shell



Etape 13 : Après le téléchargement du fichier du shell MongoDB, nous avons obtenu l'interface suivante :

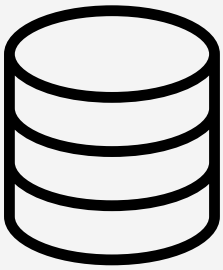


Installation de MongoDB

Etape 14 : Après l'installation du shell MongoDB, nous avons testé son bon fonctionnement en utilisant l'invite de commande (cmd) pour vérifier sa version.

```
C:\Users\MOHA BOUTA>mongod --version
db version v7.0.9
Build Info: {
    "version": "7.0.9",
    "gitVersion": "3ff3a3925c36ed277cf5eafca5495f2e3728dd67",
    "modules": [],
    "allocator": "tcmalloc",
    "environment": {
        "distmod": "windows",
        "distarch": "x86_64",
        "target_arch": "x86_64"
    }
}
```

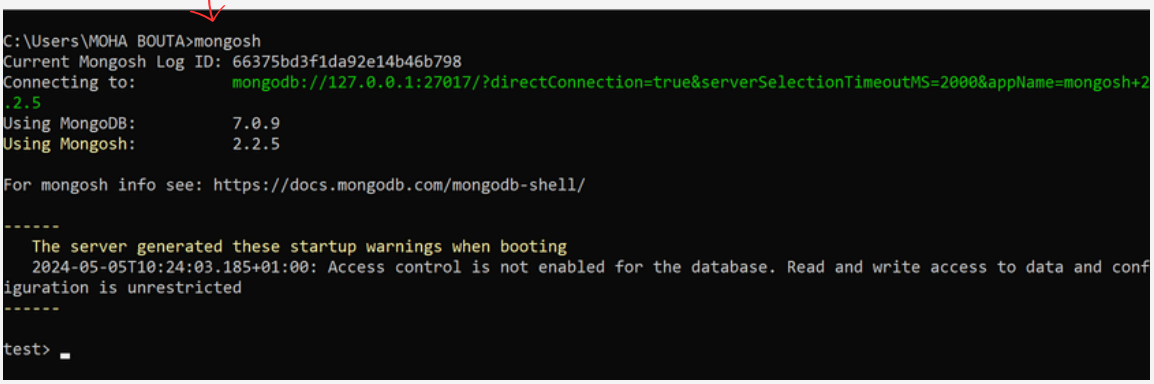
CRÉATION DE LA BASE DE DONEES AVEC MONGODB



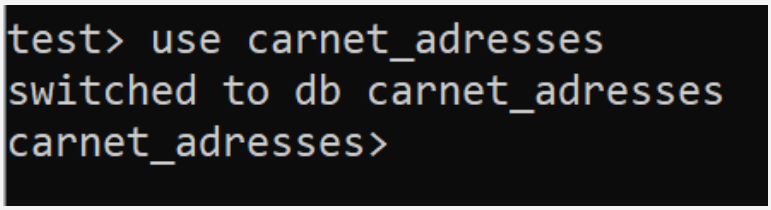
Pour créer une base de données dans MongoDB, deux méthodes simples peuvent être utilisées : le shell de MongoDB ou l'interface Compass.

⏩ méthodes I : shell de mongoDB

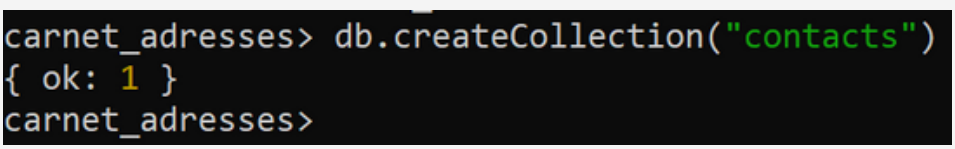
La commande **mongosh** permet d'accéder au shell de MongoDB.



La commande **use carnet_adresses** permet de créer la base de données "carnet_adresses" dans MongoDB.



La commande **db.createCollection("contacts")** permet de créer la collection "contacts" dans la base de données "carnet_adresses" dans MongoDB.



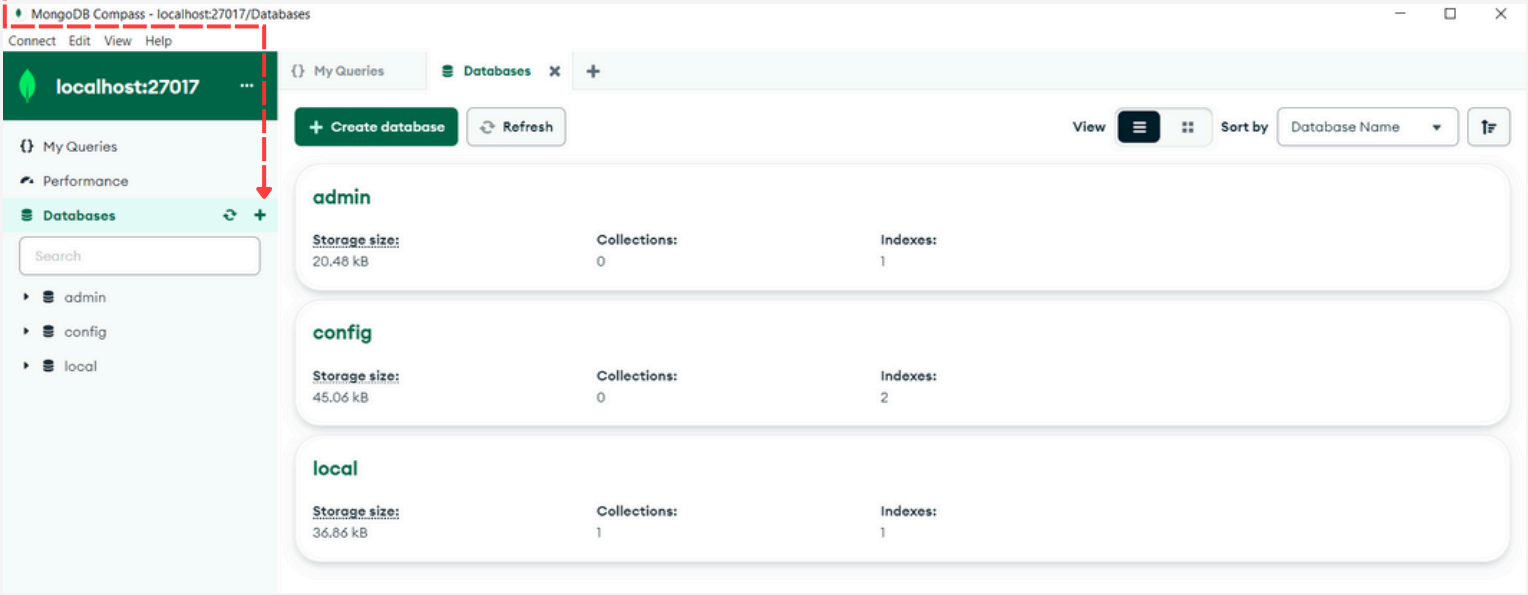
Maintenant que la base de données et la collection ont été créées, la base de données "carnet_adresses" est prête à recevoir des données. Vous pouvez ajouter des documents (contacts) à la collection "contacts" en utilisant les commandes d'insertion appropriées de MongoDB.

CRÉATION DE LA BASE DE DONEES AVEC MONGODB

Pour créer une base de données dans MongoDB, deux méthodes simples peuvent être utilisées : le shell de MongoDB ou l'interface Compass.

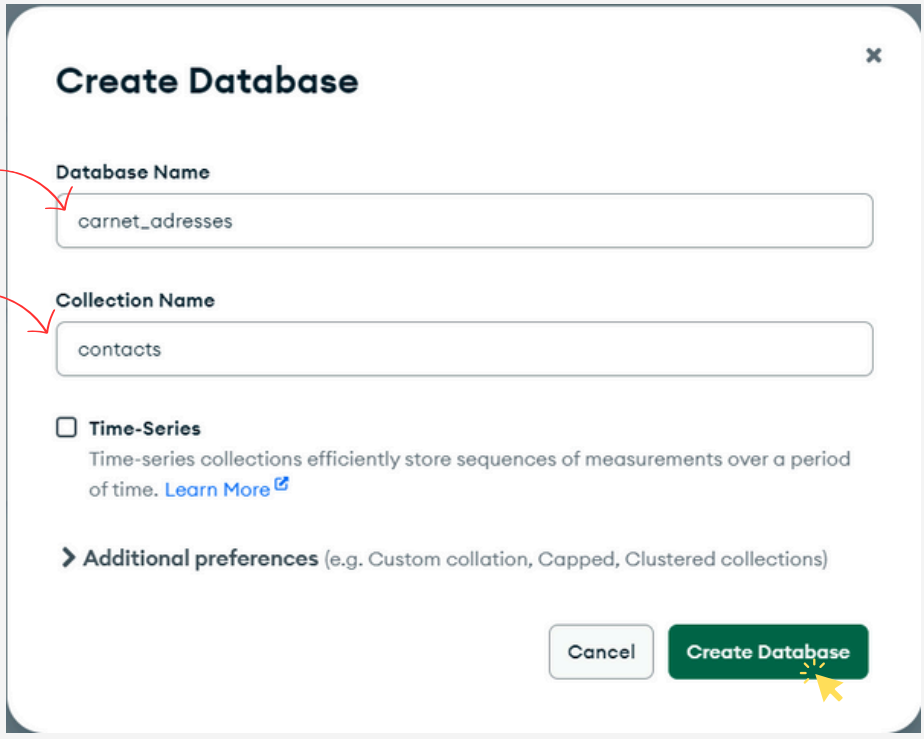
⏩ méthodes II : Utilisation de l'interface Compass

Création de la base de données : Cliquez sur le bouton "Create Database" ou "Créer une base de données" pour créer une nouvelle base de données.



Ajouter le nom de la base de données

Ajouter le nom de la Collection



Maintenant que la base de données et la collection ont été créées, la base de données "carnet_adresses" est prête à recevoir des données. Vous pouvez ajouter des documents (contacts) à la collection "contacts" en utilisant les commandes d'insertion appropriées de MongoDB.

CRÉATION INTERFACE UTULISATEUR

Nous avons développé une application de carnet d'adresses basique en utilisant Python et le module Tkinter pour créer une interface utilisateur conviviale. Cette application permet aux utilisateurs de stocker et de gérer facilement leurs contacts. À travers une interface simple et intuitive, les utilisateurs peuvent ajouter de nouveaux contacts, afficher la liste de tous les contacts enregistrés, mettre à jour les informations d'un contact existant, supprimer des contacts indésirables et effectuer des recherches par nom, numéro de téléphone ou adresse e-mail.

- Alors qu'ona devisé lintefrace en 4 parties :

⏩ Connexion à MongoDB :

Cette partie du code permet à l'utilisateur de se connecter à une base de données MongoDB en fournissant l'URL du serveur, le nom de la base de données et le nom de la collection. Si la connexion est réussie, une interface pour gérer les contacts est initialisée.

```python
# Connexion à MongoDB
def connecter_mongodb():
    global collection
    url = entry_url.get()
    database_name = entry_database.get()
    collection_name = entry_collection.get()

    try:
        client = pymongo.MongoClient(url)
        database = client[database_name]
        collection = database[collection_name]
        messagebox.showinfo("Connexion réussie", "Connexion à MongoDB établie avec succès.")
        initialiser_interface_crud()
    except pymongo.errors.ConnectionFailure:
        messagebox.showerror("Erreur de connexion", "Impossible de se connecter à MongoDB. Vérifiez l'URL et réessayez.")

```

CRÉATION INTERFACE UTULISATEUR

⏩ Initialisation de l'interface CRUD :

Une fois connecté à MongoDB, cette partie du code initialise l'interface utilisateur pour les opérations CRUD (Créer, Lire, Mettre à jour, Supprimer). Elle crée des champs pour saisir le nom, prénom, téléphone, e-mail et adresse d'un contact, ainsi que des boutons pour effectuer ces opérations.

```python
# Initialiser l'interface CRUD
def initialiser_interface_crud():
    frame_connexion.destroy()

    global entry_nom, entry_prenom, entry_telephone, entry_email, entry_adresse, entry_recherche

    frame_crud_contacts = tk.Frame(root)
    frame_crud_contacts.pack(pady=10)

    # ... (autres composants d'interface)

```

⏩ Opérations CRUD :

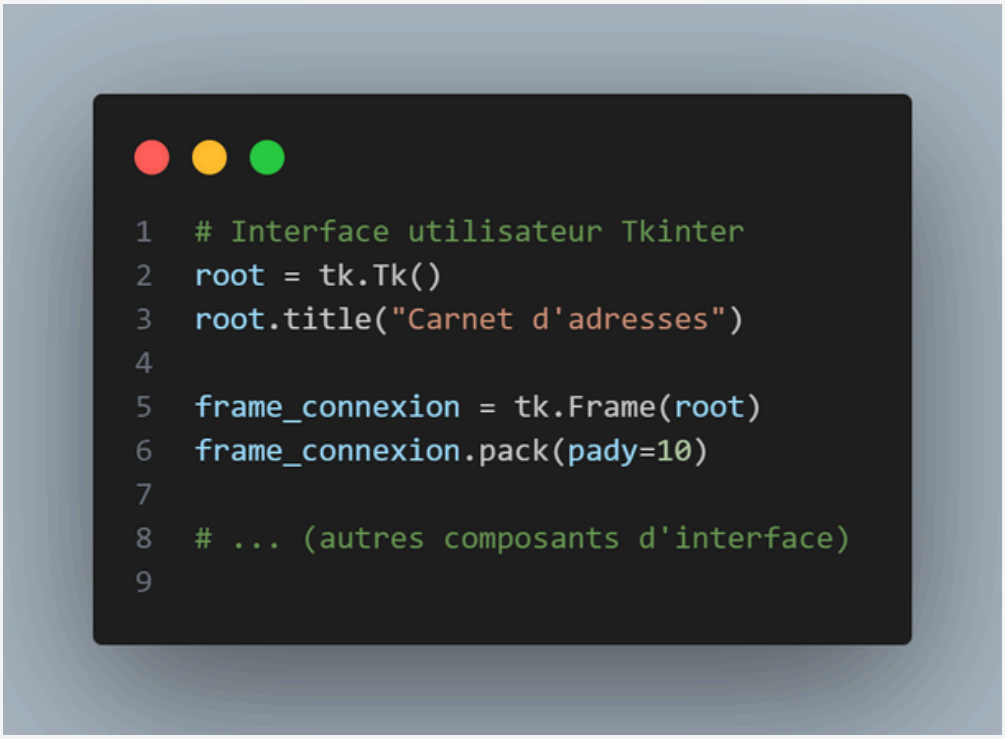
Chaque fonction ici correspond à une opération CRUD spécifique. Par exemple, "creer_contact()" crée un nouveau contact en récupérant les informations saisies par l'utilisateur et en les ajoutant à la collection MongoDB. Les autres fonctions effectuent des opérations similaires pour lire, mettre à jour et supprimer des contacts.

```python
# Opérations CRUD

def creer_contact():
    nom = entry_nom.get()
    prenom = entry_prenom.get()
    telephone = entry_telephone.get()
    email = entry_email.get()
    adresse = entry_adresse.get()

    contact = {"nom": nom, "prenom": prenom, "telephone": telephone, "email": email, "adresse": adresse}
    collection.insert_one(contact)
    messagebox.showinfo("Succès", "Contact créé avec succès.")

# ... (autres fonctions CRUD)

```

CRÉATION INTERFACE UTULISATEUR

Interface utilisateur Tkinter :

Cette partie du code crée une fenêtre Tkinter pour l'interface utilisateur. Elle contient des champs pour saisir les informations de connexion MongoDB, ainsi que des boutons pour effectuer la connexion et démarrer l'application de carnet d'adresses.



```python
# Interface utilisateur Tkinter
root = tk.Tk()
root.title("Carnet d'adresses")

frame_connexion = tk.Frame(root)
frame_connexion.pack(pady=10)

# ... (autres composants d'interface)

```

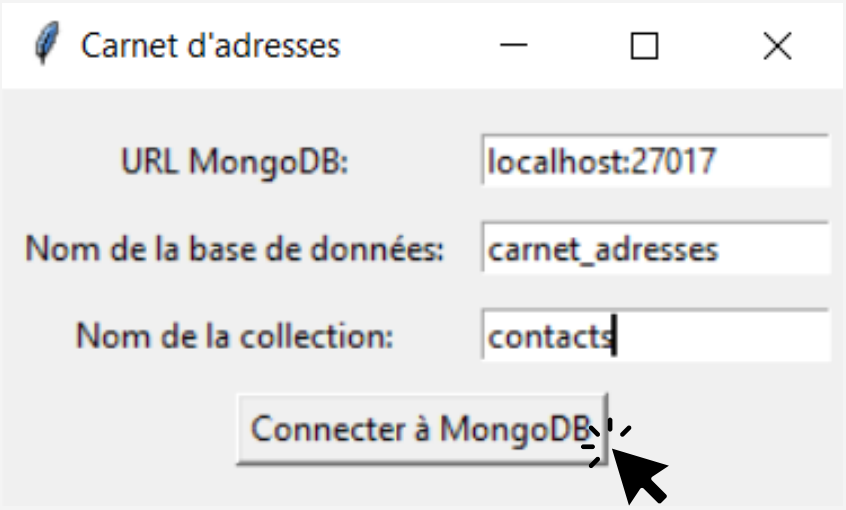
Chaque partie du code contribue à créer une application fonctionnelle de carnet d'adresses avec une interface utilisateur simple, permettant aux utilisateurs de gérer efficacement leurs contacts via une base de données MongoDB.

Vous pouvez consulter le code complet dans notre dépôt Git en suivant le lien ci-dessous : **https://github.com/MOHAMEDBOUTALMAOUINE/Carnet_Adresses-MongoDB**

DEMONSTRATION INTERFACE UTULISATEUR

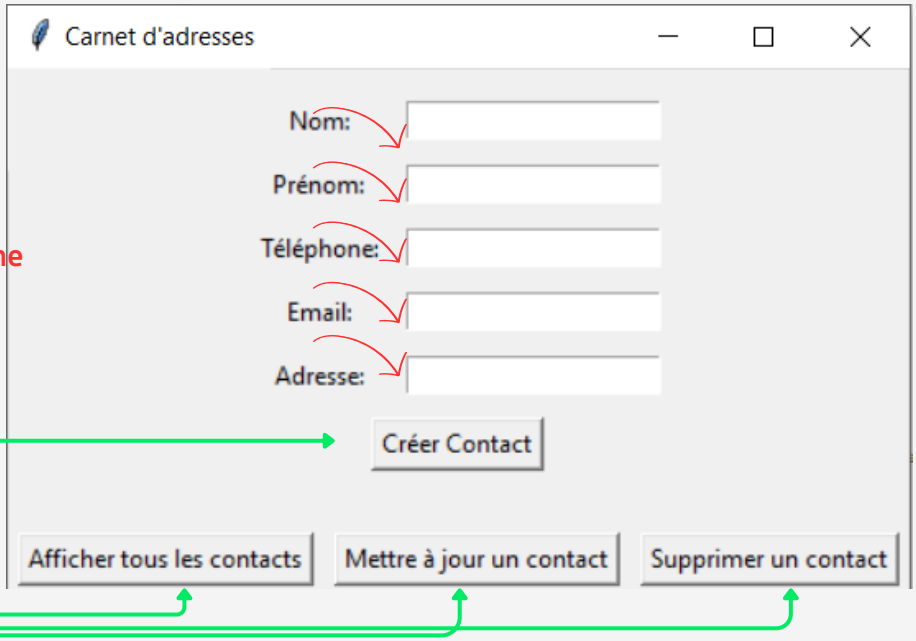
Première exécution de code

interface de connexion a la base de données qui convienne



fournir les informations nécessaire pour la connexion

Après avoir entré les informations nécessaires à la base de donner on aura l'accès à interface utilisateur de crud



Ajouter le Nom

Ajouter le Prenom

Ajouter le Telephone

Ajouter l' Email

Ajouter l' Adresse

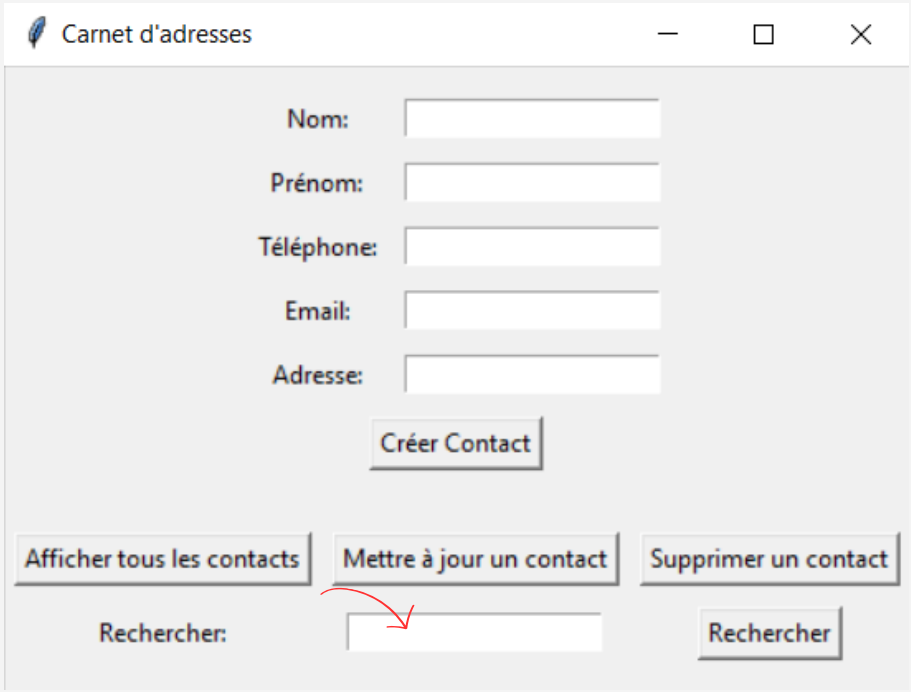
les opérations CRUD

FONCTINNALITES SUPPLÉMANTAIRE : RECHERCHE

En plus des fonctionnalités de base telles que la création, la lecture, la mise à jour et la suppression de contacts, nous avons enrichi notre application en incluant une fonctionnalité de recherche avancée. Cette fonctionnalité permet aux utilisateurs de trouver rapidement des contacts en saisissant des termes de recherche tels que le nom, le prénom, le numéro de téléphone, l'adresse e-mail ou l'adresse

```python
def rechercher_contact():
    recherche = entry_recherche.get()
    query = {"$or": [{"nom": {"$regex": recherche}},
                     {"prenom": {"$regex": recherche}}, # Ajout de la recherche par prénom
                     {"telephone": {"$regex": recherche}},
                     {"email": {"$regex": recherche}},
                     {"adresse": {"$regex": recherche}}]}
    contacts = list(collection.find(query))
    if contacts:
        affichage = "\n".join([f"Nom: {contact['nom']}\nPrénom: {contact['prenom']}\nTéléphone: {contact['telephone']}\nEmail: {contact['email']}\nAdresse: {contact['adresse']}\n" for contact in contacts])
        messagebox.showinfo("Résultat de la recherche", affichage)
    else:
        messagebox.showinfo("Résultat de la recherche", "Aucun contact trouvé.")

```

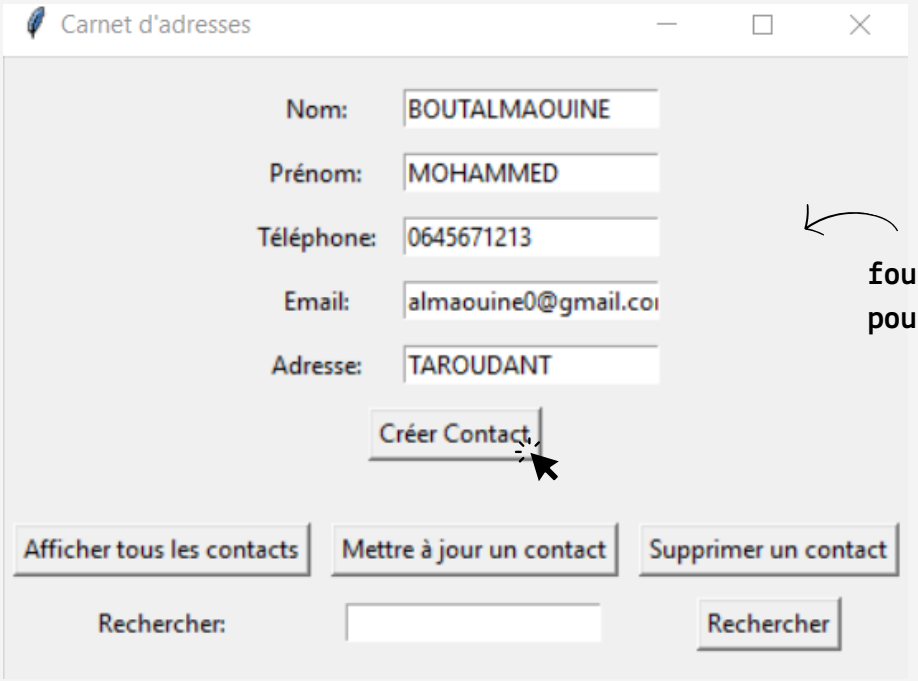
Deuxième exécution de code



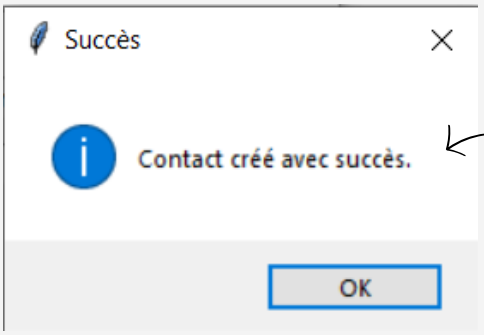
champs de recherche

TESTS D'APPLICATION

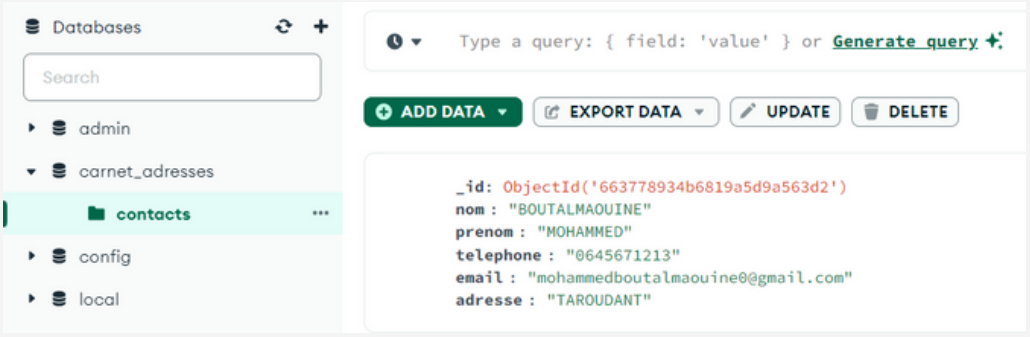
creation de premier contact :



fournir les informations nécessaire pour le contact



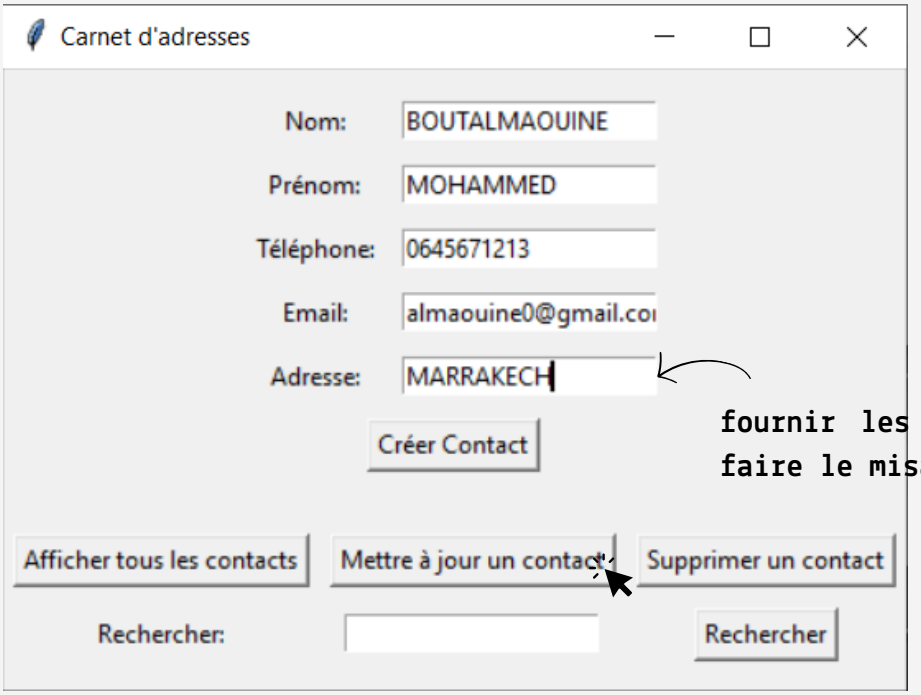
Message de bon enregistrement de contact dans BD



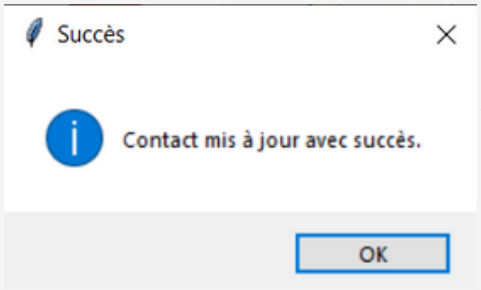
Le bon enregistrement de contact dans BD

TESTS D'APPLICATION

Mise a jour d'un contact :



fournir les informations dont on va faire le misa a jour (Adresse par ex)



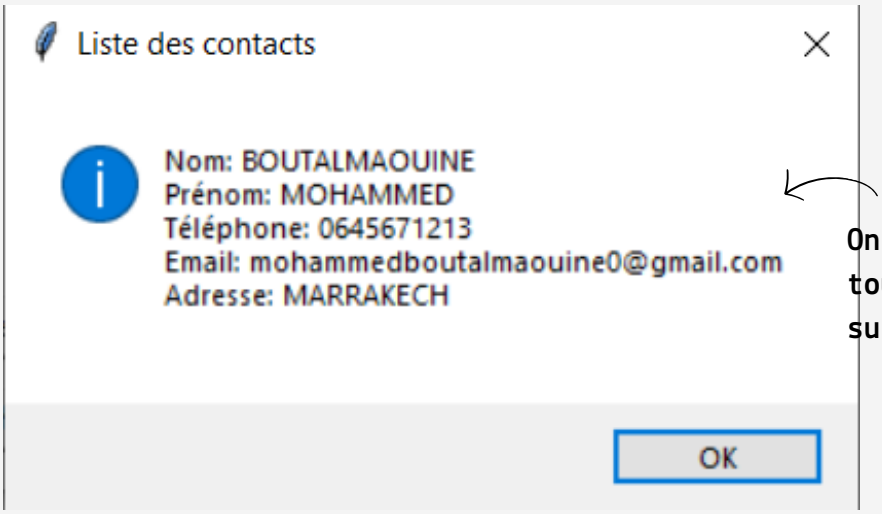
Message de mise à jour de contact dans BD


```
_id: ObjectId('663778934b6819a5d9a563d2')
nom : "BOUTALMAOUINE"
prenom : "MOHAMMED"
telephone : "0645671213"
email : "mohammedboutalmaouine0@gmail.com"
adresse : "MARRAKECH"
```


la mise à jour de contact dans BD avec succès

TESTS D'APPLICATION

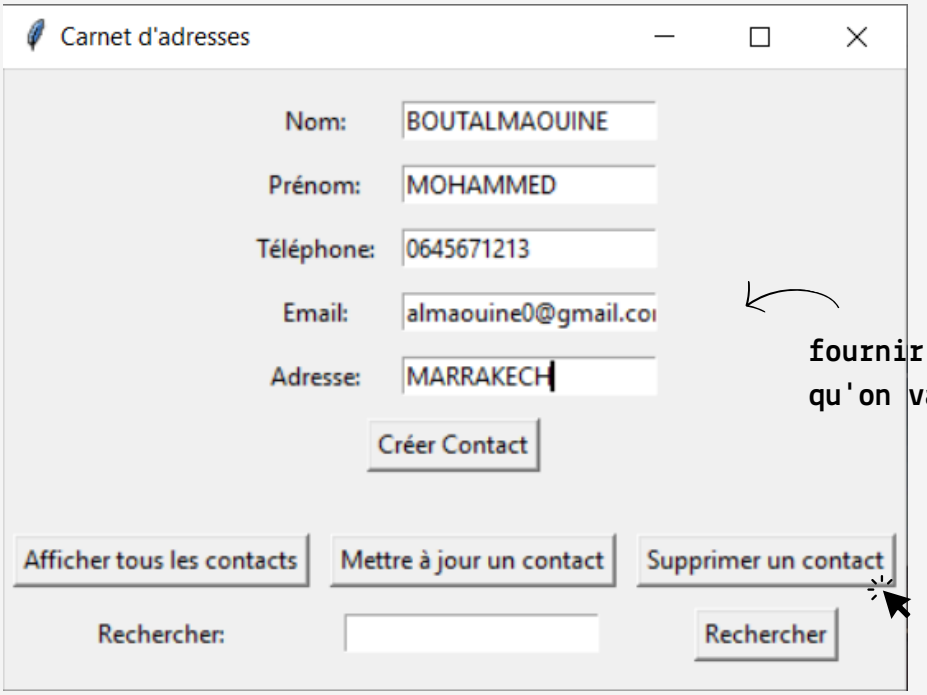
- **Affichage des contacts de la BD :**



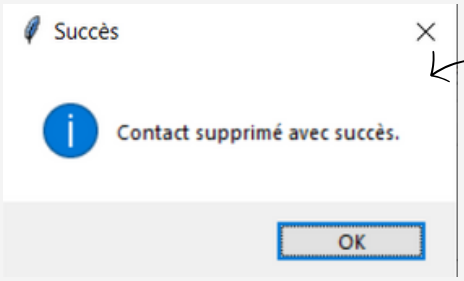
On cliquant sur le button afficher tous les contacts on aura la fenêtre suivante

TESTS D'APPLICATION

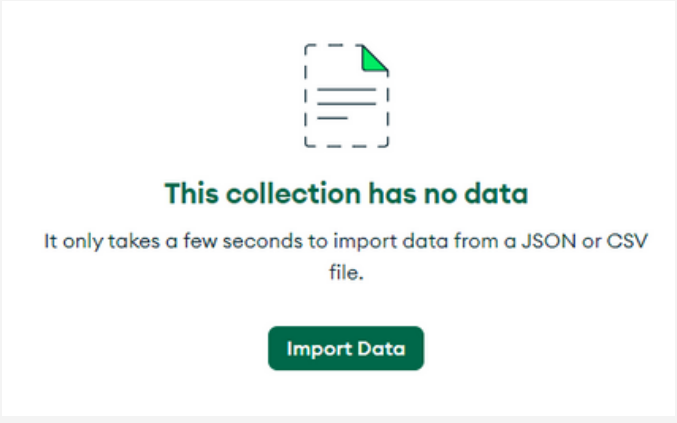
Suppression d'un contact :



fournir les informations du contact qu'on va supprimer



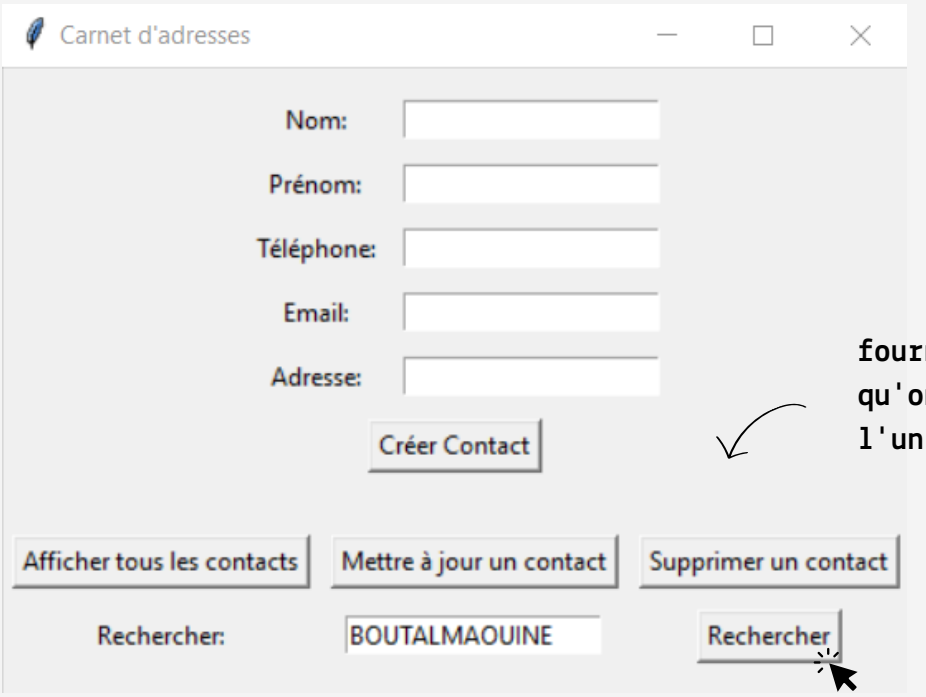
Message de suppression de contact dans BD



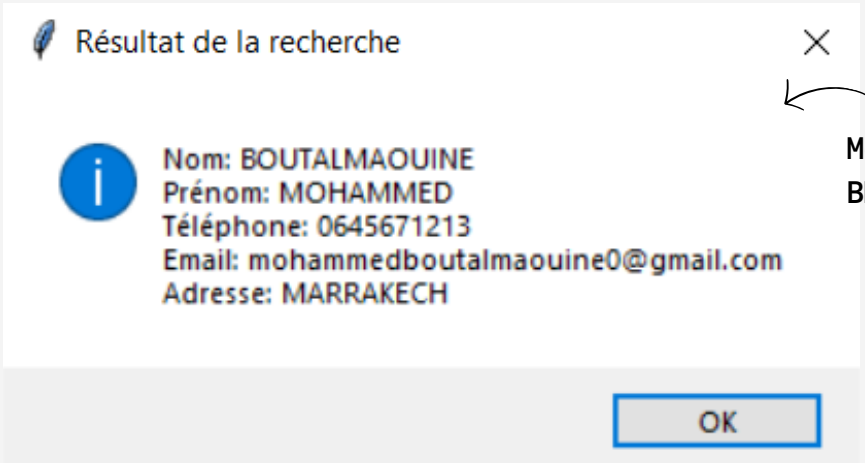
suppression du de contact dans BD avec succès

TESTS D'APPLICATION

⏩ Recherche d'un contact :



fournir les informations du contact qu'on va chercher on peut entrer juste l'un des informations (prénom par ex)



Message de recherche de contact dans BD avec le prénom BOUTALMAOUINE

CONCLUSION

Pour conclure ce projet, nous avons développé une application de carnet d'adresses utilisant Python, Tkinter pour l'interface utilisateur et MongoDB comme base de données. Cette application offre une interface conviviale permettant aux utilisateurs de créer, lire, mettre à jour et supprimer des contacts. De plus, nous avons ajouté une fonctionnalité de recherche avancée, offrant aux utilisateurs la possibilité de trouver rapidement des contacts en fonction de différents critères tels que le nom, le prénom, le numéro de téléphone, l'email ou l'adresse. Grâce à cette application, la gestion des contacts devient plus efficace et intuitive. Elle peut être facilement étendue et personnalisée pour répondre aux besoins spécifiques des utilisateurs. En combinant la puissance de Python, Tkinter et MongoDB, ce projet illustre comment créer une application de gestion de données simple mais fonctionnelle.

Lien GitHub du projet:
https://github.com/MOHAMEDBOUTALMAOUINE/Carnet_Adresses-MongoDB