

RAPPORT DE STAGE

BTS SIO option Slam (Solutions logicielle et

Applications) 1^{ère} année

Du 19 mai 2025 au 15 juillet

2025

Safia Benabdellaoui

Table des matières :

Remerciements	3
Introduction	4
Présentation de l'entreprise NEAR DevHub	5
Environnement de travail et outils utilisés.....	6
Découverte des modèles LLM et étude comparative	7
<i>Création et configuration d'agents IA</i>	8
Utilisation de Git et GitHub en ligne de commande	10
Projet d'agent IA touristique	11
Étude des <i>vector stores</i> et enrichissement des agents.....	13
Tâche finale : création autonome de mon propre agent IA – <i>BookBot</i>	15
Conclusion	18

Remerciements

Avant toute chose, je tiens à exprimer ma profonde gratitude à **Guillermo Gallardo**, mon tuteur de stage au sein de l'équipe **DevRel de NEAR DevHub**, pour son accompagnement bienveillant, sa disponibilité, et la confiance qu'il m'a accordée tout au long de ce stage.

Son expertise, ses conseils avisés et sa pédagogie m'ont permis de progresser rapidement, de mieux comprendre les enjeux liés aux agents intelligents, et de prendre part à un environnement professionnel stimulant, innovant et riche en apprentissages.

Je remercie également toute l'équipe **Near DevHub**, qui m'a chaleureusement accueillie et m'a offert l'opportunité d'évoluer dans un cadre collaboratif, motivant, et profondément tourné vers l'avenir de l'intelligence artificielle.

Introduction

Dans le cadre de mon Brevet Technicien Supérieur Services Informatiques aux Organisations, j'ai eu l'opportunité d'effectuer un stage de plusieurs semaines au sein de l'équipe **NEAR DevHub**, spécialisée dans le développement et la promotion d'agents intelligents via la plateforme **NearAI**. Ce stage, réalisé à **Barcelone**, m'a permis de découvrir un environnement professionnel moderne, collaboratif et stimulant, tout en développant des compétences concrètes et actuelles autour de l'intelligence artificielle.

Encadrée par **Guillermo Gallardo**, et membre de l'équipe DevRel en tant que Senior DevRel Ingineer, j'ai bénéficié d'un **accompagnement quotidien**, durant lequel j'ai pu poser mes questions, partager mes avancées et progresser rapidement.

Ce stage avait pour objectif principal de me familiariser avec les **modèles d'intelligence artificielle (LLM)**, d'apprendre à concevoir des **agents intelligents** personnalisés, et de comprendre le fonctionnement des outils de versionnement comme **GitHub** dans un contexte réel de développement.

Ce rapport retrace de façon détaillée les missions réalisées, les compétences acquises, les difficultés rencontrées ainsi que les enseignements tirés de cette expérience enrichissante, qui m'a permis de combiner théorie et pratique autour de l'IA moderne.

Présentation de l'entreprise NEAR DevHub

NEAR DevHub est une initiative créée pour accompagner les développeurs et créateurs de projets dans l'écosystème **NEAR**, une plateforme dédiée à l'innovation technologique, notamment dans les domaines de l'intelligence artificielle et du développement open source.

Le DevHub fonctionne comme une **communauté dynamique de développeurs**, réunissant des personnes du monde entier autour d'outils et de projets concrets. Il a pour mission de **soutenir, former et faire évoluer** les talents techniques qui souhaitent participer à la construction d'agents intelligents, d'applications IA ou d'outils en lien avec les technologies décentralisées.

Ce programme est soutenu par la **NEAR Foundation**, une organisation à but non lucratif basée en Suisse plus précisément à Zug, qui pilote l'écosystème NEAR. La fondation fournit les ressources nécessaires (financement, outils, documentation, mentors, communauté) pour faire évoluer les projets innovants dans un esprit **collaboratif, éthique et ouvert à tous**.

L'équipe **DevRel (Developer Relations)**, à laquelle j'étais rattachée pendant mon stage, joue un rôle central dans cet accompagnement : elle aide les nouveaux développeurs à comprendre les outils, à créer leurs premiers agents IA, à utiliser efficacement GitHub, à collaborer et à progresser dans leurs projets.



NEAR DevHub

Plateforme appartenant à l'écosystème NEAR

organisme qui finance et structure

Environnement de travail et outils utilisés

J'ai travaillé sur mon propre ordinateur, dans un environnement technique orienté développement Python et intelligence artificielle.

Voici les principaux outils que j'ai utilisés :

- **Visual Studio Code (VS Code)** : éditeur de code dans lequel j'ai écrit tous les fichiers nécessaires aux agents IA (agent.py, README.md, metadata.json).
- **Terminal WSL (Windows Subsystem for Linux)** : pour utiliser les commandes Linux et interagir avec GitHub.
- **Python 3.11** : langage principal utilisé pour développer les agents intelligents et interagir avec les modèles LLM via la librairie nearai.
- **near-ai** : librairie Python développée par NEAR DevHub, permettant de créer, configurer et tester des agents IA.
- **Git et GitHub** : pour le versionnement de mes projets. J'ai appris à utiliser des commandes Git (git add, git status, git commit, git push) pour gérer mon code et publier mon travail.

Mon dépôt Github personnel, utilisé tout au long du stage pour partager mes avancées, est disponible ici :

<https://github.com/SAFyess/safias-amazing-internship>



Découverte des modèles LLM et étude comparative

L'un des premiers objectifs de mon stage a été de **découvrir les différents modèles d'intelligence artificielle (LLM)** que je pouvais utiliser dans mes agents. Un LLM (Large Language Model) est un modèle d'IA entraîné pour comprendre et générer du texte, un peu comme ChatGPT. Ces modèles permettent à un agent intelligent de **répondre, réfléchir ou conseiller** comme un humain.

J'ai étudié plusieurs modèles et leurs différentes versions disponibles sur Near AI. Voici ceux que j'ai testés ou analysés :

Modèle	Points forts	Limites	Utilisation fréquente
LLaMA	Réponses structurées, compréhensibles	Moins rapide sur certaines tâches	Idéal pour des agents simples
DeepSeek	Très fluide	Parfois trop créatif	Agents conversationnels
Qwen	Bon équilibre vitesse/qualité	Moins précis sur les sujets pointus	Utilisation générale
Flux	Rapide, compact	Moins profond	Réponses directes ou techniques
Firesearch	Très bon pour la recherche d'infos sur plusieurs sources	Moins naturel dans la conversation	Recherche documentaire ou touristique

Étude comparative

Pour mieux comprendre leurs différences, j'ai réalisé une **étude comparative** en leur posant les **mêmes questions** sur des sujets variés (ex. : résumé de livre, conseils touristiques, idées de lecture, etc.).

```
+ Hello! What can I do for you? 2 days in Marrakech?
+ Available!
** Day 1: Marrakech (Morning) A mix of culture, cuisine, and leisure!
  - "Morning": 
    - "Breakfast at Riad Al-Mahrouq's scenic terrace overlooking the Jemaa el-Fna square. Enjoy mint tea and pastries while watching the early morning activity below."
    - "Explore the Ben Youssef Madrasa and the Koutoubia Mosque in the heart of the city. Don't miss the intricate tilework and golden minaret at sunset."
  - "Afternoon": 
    - "Visit the Bahia Palace (morning), a grand 19th-century residence with ornate rooms and courtyards. 
    - "Walk through the Souks of Marrakech (afternoon). 
    - "Taste of the Medina": lose yourself in the maze of markets (fezels, souks, bazaars). Don't forget to haggle!
  - "Evening": 
    - "Dine at a local restaurant (like 'Le Petit Marocain') featuring traditional Moroccan cuisine. Try tagine, couscous, or lamb kebab with a side of mint tea or mint tea with mint leaves."
    - "Stay at a riad in the Medina for a traditional night." 

** Day 2: Marrakech & Agadir
  - "Morning": 
    - "Visit the Bahia Palace (morning). Stroll through the red-tiled rooms and lush gardens filled with citrus trees and flowering plants. 
    - "Visit the Koutoubia Mosque (morning). Explore the intricate details of its architecture, including its iconic green and gold minaret." 
    - "Afternoon": 
      - "Visit the Ben Youssef Madrasa (morning). 
      - "Walk through the Souks of Marrakech (afternoon). 
      - "Taste of the Medina": lose yourself in the maze of markets (fezels, souks, bazaars). 
      - "Evening": 
        - "Dinner at a local restaurant (like 'Le Petit Marocain') featuring traditional Moroccan cuisine. Try tagine, couscous, or lamb kebab with a side of mint tea or mint tea with mint leaves." 
        - "Stay at a riad in the Medina for a traditional night." 

  - "Morning": 
    - "Travel to Agadir (train). Explore the coastal town of Agadir, known for its modern architecture and beachfront. 
    - "Visit the Agadir Marina (morning). 
    - "Afternoon": 
      - "Visit the Agadir Marina (morning). 
      - "Visit the Agadir Marina (afternoon). 
      - "Evening": 
        - "Dinner at a local restaurant (like 'Le Petit Marocain') featuring traditional Moroccan cuisine. Try tagine, couscous, or lamb kebab with a side of mint tea or mint tea with mint leaves." 
        - "Stay at a riad in the Medina for a traditional night." 
```

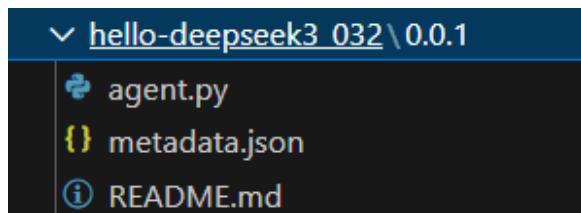
Création et configuration d'agents IA

Une fois les modèles LLM explorés, j'ai appris à **créer mes propres agents IA** sur la plateforme Near AI. Un agent IA est comme une petite intelligence spécialisée : on peut lui dire quoi faire (ex. : recommander un livre, expliquer une notion, donner des conseils...), et il répond automatiquement grâce au modèle choisi.

Étapes de création

Pour chaque agent, je devais suivre ces étapes :

1. **Créer un dossier pour l'agent** sur mon projet (dans Visual Studio Code).
2. **Créer deux fichiers importants :**
 - README.md → pour expliquer le rôle de l'agent.
 - metadata.json → pour configurer son comportement.



3. **Choisir un modèle LLM adapté** (ex. : LLaMA, DeepSeek...) selon le besoin de l'agent.

A screenshot of a code editor showing the 'metadata.json' file. The code is as follows:

```
1 {
2     "name": "hello-deepseek3_0324",
3     "version": "0.0.1",
4     "description": "Testing deepseek3_032 agent",
5     "category": "agent",
6     "tags": [],
7     "details": {
8         "agent": {
9             "defaults": {
10                 "model": "deepseek3_0324",
11                 "model_provider": "fireworks"
12             }
13         }
14     },
15     "show_entry": true
16 }
```

The code defines an object with properties like name, version, description, category, tags, and details. The details section contains an 'agent' object with a 'defaults' property that specifies a model and a model provider.

4. **Tester l'agent** directement dans la commande WSL.

> hello, what are you capable of doing?

Assistant: Hello! I'm thrilled to meet you! I'm a friendly AI agent, and I'm here to help and provide information on a wide range of topics. I'm capable of:

1. **Answering questions**: I can process natural language queries and provide relevant answers, covering everything from science, history, and technology to entertainment, culture, and more.
2. **Providing definitions**: If you're unsure about a word or term, I can define it for you and offer examples to help clarify its meaning.
3. **Generating text**: I can create text based on a prompt, topic, or style, which can be helpful for writing ideas, composing emails, or even creating short stories.
4. **Translation**: I can translate text from one language to another, including popular languages such as Spanish, French, German, Chinese, and many more.
5. **Summarization**: If you'd like me to summarize a long piece of text, I can condense it into a shorter, more digestible version, highlighting key points and main ideas.
6. **Offering suggestions**: If you're looking for ideas or recommendations, I can provide suggestions on topics like books, movies, music, travel destinations, and more.
7. **Playing games**: We can engage in text-based games, such as Hangman, 20 Questions, Word Chain, or even simple puzzles.
8. **Chatting and conversing**: I'm happy to engage in casual conversations, answering your questions, sharing interesting facts, and even cracking jokes (bad puns included)!

Utilisation de Git et GitHub en ligne de commande

Pendant mon stage, j'ai appris à utiliser **Git** (un outil de versionnage de code) et **GitHub** (une plateforme en ligne pour héberger ce code) **en ligne de commande**.

- **Git** permet de suivre l'évolution d'un projet (comme un "historique intelligent").
- **GitHub** permet de **stocker le code en ligne**, de manière sécurisée, accessible partout.
- En ligne de commande, c'est plus rapide et plus direct pour manipuler les fichiers.

Voici les principales commandes que j'ai apprises et utilisées :

Commande	Fonction
git init	Démarrer un nouveau projet Git dans un dossier.
git status	Voir les fichiers modifiés ou ajoutés.
git add .	Ajouter tous les fichiers modifiés pour les enregistrer.
git commit -m "message"	Enregistrer les changements avec un message explicatif.
git remote add origin <URL>	Connecter mon dossier local à mon dépôt GitHub.
git push -u origin main	Envoyer mon code sur GitHub pour la première fois.
git push	Mettre à jour mon code sur GitHub après chaque modification.

Projet d'agent IA touristique

L'un de mes projets a été de créer un **agent intelligent** capable de répondre à des questions touristiques sur une ville spécifique. J'ai commencé par la ville de **Rabat**, la capitale du Maroc, et j'avais prévu de pouvoir ajouter d'autres villes par la suite.

Le but était de créer un agent qui pourrait aider un utilisateur à découvrir une ville :

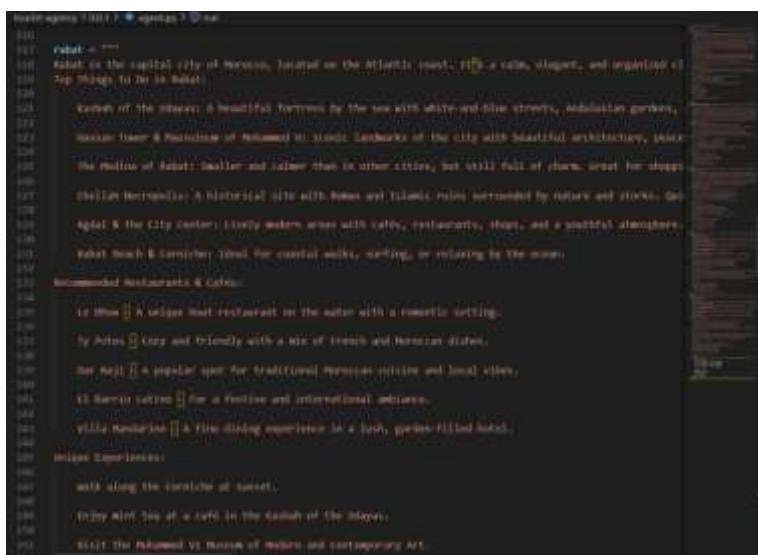
- **Que visiter ?**
- **Quand y aller ?**
- **Quelles spécialités locales goûter ?**
- **Quels conseils pratiques ?**

Comment ça fonctionne ?

L'agent IA s'appuie sur un **modèle LLM** (comme DeepSeek ou Qwen) pour traiter les questions. Mais pour lui permettre de répondre avec **des infos locales précises**, je lui ai fourni **des fichiers personnalisés**.

Voici comment j'ai structuré le projet :

1. **Fichier rabat.md** → J'y ai ajouté des descriptions sur Rabat : monuments, plages, quartiers populaires, gastronomie, culture, conseils pratiques, etc.



2. Fichier metadata.json

→ Ce fichier configure l'agent :

- o nom de l'agent
- o modèle LLM utilisé

```

1   {
2     "name": "tourist-agent",
3     "version": "0.0.1",
4     "description": "A tourist agent",
5     "category": "agent",
6     "tags": [],
7     "details": {
8       "agent": {
9         "defaults": {
10           "model": "llama4scoutinstructbasic",
11           "model_provider": "fireworks"
12         }
13       }
14     },
15     "show_entry": true
16   }

```

3. Test de l'agent en ligne via les commandes WSL

→ J'ai posé des questions comme « Que visiter à Rabat ? » ou « Quelle est la meilleure période pour y aller ? » et j'ai observé les réponses.

```

Salut super agent, est ce que tu pourrais me faire un planning de ce que je pourrais faire à Rabat sur une période de 2 jours ??  

Assistant: Bonjour !  

Rabat, la capitale du Maroc, est une ville charmante qui offre un mélange parfait de modernité, d'histoire et de charme balnéaire. Pour un séjour de 2 jours à Rabat, voici un planning suggéré :  

**Jour 1 :**  

* Matin : Commencez par visiter la Kasbah des Oudayas, une forteresse magnifique située au bord de la mer, avec des rues blanches et bleues, des jardins andalous et des vues panoramiques sur l'océan. C'est l'endroit idéal pour prendre des photos et profiter de l'atmosphère calme.  

* Après-midi : Dirigez-vous vers la Tour Hassan et le Mausolée de Mohammed V, deux monuments emblématiques de la ville avec une architecture magnifique et une atmosphère paisible.  

* Soir : Explorer la Médina de Rabat, plus petite et plus calme que celles des autres villes, mais toujours pleine de charme. Vous pouvez y trouver des boutiques de handicrafts et de produits locaux.  

**Jour 2 :**  

* Matin : Visitez la Nécropole de Chellah, un site historique avec des ruines romaines et islamiques entourées de nature et de cigognes. C'est un endroit calme et verdoyant, parfait pour une promenade.  

* Après-midi : Découvrez l'Agdal et la contre-ville, des zones modernes avec des cafés, des restaurants et des boutiques. Vous pouvez également prendre un tramway pour aller à Salé, la ville sour de Rabat, située de l'autre côté de la rivière.  

* Soir : Terminez votre séjour par une promenade le long de la corniche de Rabat, idéal pour voir le coucher du soleil et profiter de l'atmosphère balnéaire.  

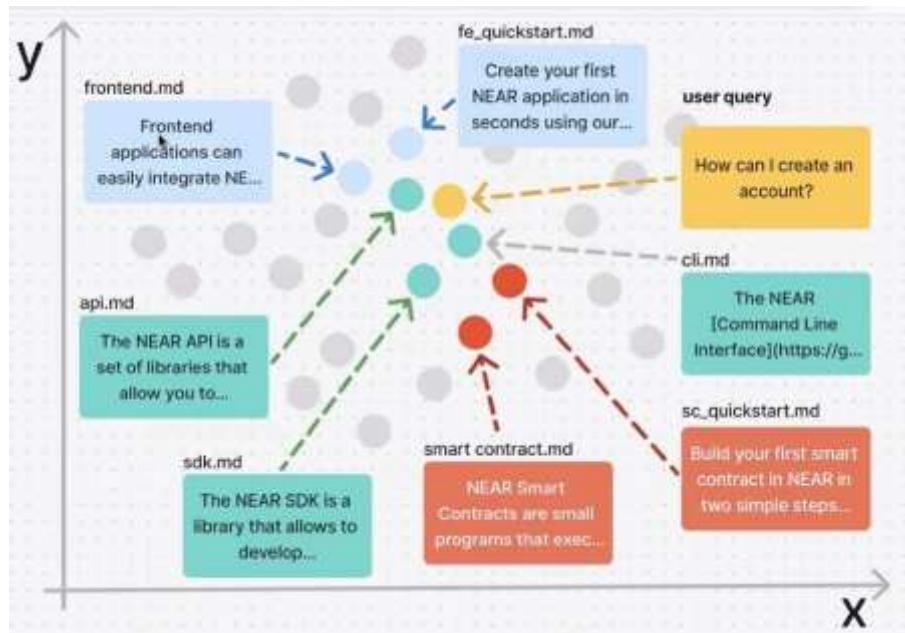
N'oubliez pas de goûter la cuisine locale, notamment les plats traditionnels marocains comme la tagine et le couscous. Vous pouvez également essayer les restaurants et cafés recommandés, comme le Dhow, Ty Potes et han Najj.  

J'espère que ce planning vous aidera à organiser votre séjour à Rabat !>

```

Étude des *vector stores* et enrichissement des agents

j'ai découvert le concept de **vector stores**, une technologie utilisée pour **mémoriser et retrouver des informations** de façon intelligente à partir d'un texte.



C'est une sorte de **base de données spéciale** utilisée dans l'intelligence artificielle. Mais au lieu de stocker des données de manière classique (comme des cases dans un tableau), il transforme les textes en **vecteurs mathématiques**.

Cela permet à l'agent IA de **comparer** les idées, même si elles ne sont pas formulées exactement pareil.

Le but de cette étape était de permettre à mon **agent touristique** de répondre aux utilisateurs avec des informations spécifiques sur différentes villes marocaines, que j'avais rédigées moi-même dans des fichiers .md (Markdown).

Pour cela, j'ai d'abord dû créer un script Python que j'ai nommé `vector_store.py`. Ce fichier a servi à **uploader mes fichiers Markdown dans un vector store**, c'est-à-dire une base de données vectorielle utilisée par NEAR AI pour stocker et retrouver les informations de manière intelligente.

Voici les principales étapes réalisées dans `vector_store.py` (visible sur la 2e image) :

- Chargement de la configuration de mon projet avec NEAR AI.
- Lecture automatique de tous les fichiers présents dans le dossier `cities/` (chaque fichier correspond à une ville).

- Upload de ces fichiers dans le vector store.
 - Création finale du vector store avec le nom "cities".

À la fin de cette opération, j'ai obtenu un **VECTOR_STORE_ID**, une sorte d'identifiant unique généré automatiquement. Cet identifiant est très important, car il permet de faire appel au vector store plus tard, dans mon agent.

Je l'ai ensuite **intégré dans le fichier agent.py** de mon agent IA touristique (voir la 1ère image). Grâce à ce VECTOR_STORE_ID, mon agent peut désormais :

- **Faire une requête intelligente dans le vector store** selon la question posée par l'utilisateur (ligne 10 à 12).
 - **Récupérer automatiquement les fichiers les plus pertinents** contenant les réponses (dans mon cas, les villes correspondant à la demande de l'utilisateur).
 - Créez un prompt enrichi avec ces contenus pour générer une réponse adaptée.

Exemple : si un utilisateur demande « Que visiter à Rabat ? », mon agent va rechercher dans le vector store tous les fichiers qui parlent de Rabat, et utiliser ces données pour construire sa réponse.

```
hello-helloagent_20
hello-mammal20bin...
hello-spawn20js...
hello-spawn20js...
hello-spawn20js...
hello-spawn3_20js...
hello-spawn3_23js...
hello-spawn32js...
hello-vrl_17js...
hello-vrlaffection0417
router-agent ->
threads
cities
  Agadir.mld
  Astanad
  Casablanca.mld
  Launched
  mammalized
  Rabat.mld
  Tanger.mld
  tangied
  launred
  Tetuan.mld
  Zagora.mld
  agentpy
  metadata
  README.mld
vector_storpy
```

vector_storpy X Targeted Manuverized (1) metadata.json hello-world

```
round-trip: 0.013 > vector_storpy > ...
1 import json
2 from glob import glob
3
4 import memory
5 import neural
6
7 # Load Neural Net Configuration
8 config = neural.config.load_config_file()
9 base_url = config.get("api_url", "https://api.manuver.vt") + "/v1"
10 auth = config["auth"]
11
12 client = openai.OpenAI(base_url=base_url, api_key=json.dumps(auth))
13
14 # Get a list of all the markdown files in the "cities" folder
15 md_files = list(glob("./cities/*.md", recursive=True))
16 file_ids = []
17
18 print("All files", md_files)
19
20 for file_path in md_files:
21     print(f"Processing {file_path}")
22
23     with open(file_path, "r", encoding="utf-8") as file:
24         uploaded_file = client.files.create(
25             file=(file_path, file.read()), "text/markdown",
26             purpose="assistant"
27         )
28         file_ids.append(uploaded_file.id)
29
30 vs = client.vector_stores.create(
31     name="cities",
32     file_ids=file_ids,
```

Tâche finale : création autonome de mon propre agent IA - BookBot

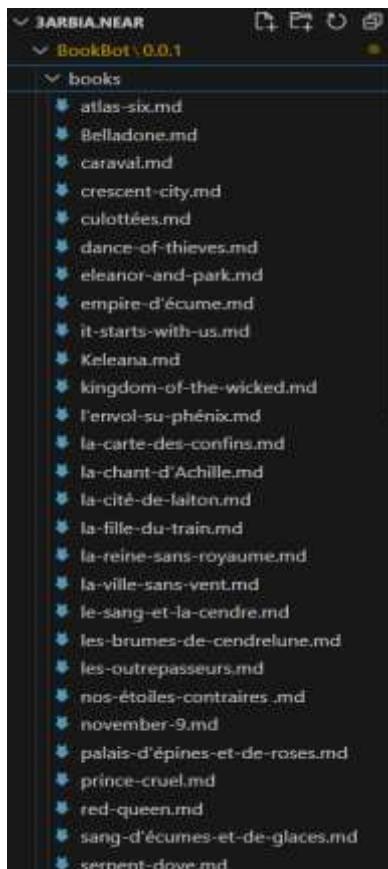
À la fin de mon stage, après avoir découvert et manipulé différents modèles de langage (LLM), appris à utiliser GitHub, enrichi des agents avec des vector stores et compris toute la logique de NEAR AI, j'ai eu l'opportunité de réaliser **mon propre projet en autonomie**.

J'ai alors décidé de créer un **agent intelligent spécialisé dans la recommandation de livres**, un sujet qui me tient énormément à cœur, car la lecture est une vraie passion pour moi. C'était aussi une manière de **lier mes deux centres d'intérêt** : la technologie et la littérature. C'est ainsi que **BookBot** est né.

Mon objectif était de permettre à un utilisateur de décrire le type de livre qu'il aimerait lire (par exemple : un roman de science-fiction, avec une héroïne forte, une ambiance futuriste et un style immersif) et que l'agent propose directement un ou plusieurs titres adaptés à ces critères.

Cela permet d'obtenir des recommandations ciblées, **sans avoir à faire de longues recherches**.

Pour enrichir cet agent, j'ai créé une **base de connaissances personnalisée**, composée de nombreux fichiers .md, chacun correspondant à un livre



J'ai choisi des œuvres destinées principalement aux jeunes et aux adolescents, car ce sont les personnes de ma génération qui se posent souvent ce type de questions. D'ailleurs, aujourd'hui, beaucoup de jeunes lecteurs trouvent leurs prochaines lectures grâce aux réseaux sociaux, notamment dans la communauté **BookTok** sur TikTok. C'est ce phénomène qui m'a inspirée.

J'ai rédigé chaque fiche de livre moi-même, en y intégrant :

- le titre,
- l'auteur,
- la date de publication,
- un résumé global sans spoiler (même pour les sagas),
- le nombre de tomes,
- et d'autres informations utiles comme l'existence d'une adaptation ou l'inspiration de l'auteur.

```
ooklibot > 0.0.1 > books > * caraval.md > # Caraval
1 # Caraval
2
3 Auteur : Stephanie Garber
4 Première publication : 2017 (par Flatiron Books)
5 Genre : Fantasy, romance, aventure, mystère
6 Résumé global (sans spoiler, pour toute la saga)
7
8 Dans un univers où les frontières entre la réalité et l'illusion sont floues, deux sœurs, Scarlett et Donatella Dragna, rêvent depuis toujours de participer à Caraval : un jeu magique organisé une fois par an par le mystérieux maître du jeu, Legend. Quand enfin l'occasion se présente, leur vie bascule.
9
10 Ce qui semble être un jeu féerique s'avère rempli d'épreuves, d'énigmes, de tentations et de dangers bien réels. Le décor est somptueux, mais derrière les apparences se cachent manipulations, secrets de famille, et pouvoirs qui dépassent l'imagination.
11
12 Au cœur de Caraval, l'amour, le sacrifice et la vérité sont mis à l'épreuve. L'histoire mêle ambiance théâtrale, magie troublante, jeux de dupes et romance intense, dans une course contre la montre où tout peut basculer.
13
14 Parfait pour les lectrices et lecteurs qui aiment :
15
16    les univers enchantés et sombres à la fois
17
18    les intrigues sentimentales mêlées à des énigmes
19
20    les rebondissements et les retournements de situation
21
22    les ambiances de carnaval, de spectacle, de rêve... mais inquiétant
23
24 Nombre de tomes :
25
26 Trilogie complète :
27
28     Caraval (2017)
```

Même si je ne peux pas inclure tous les livres du monde, j'ai pris soin de proposer une sélection variée et assez complète pour permettre à l'agent de faire des recommandations cohérentes, même sur des critères un peu complexes.

Enfin, j'ai structuré l'agent comme les précédents, en créant un fichier Python principal (agent.py) et en y connectant un vector store contenant tous mes fichiers .md. J'ai également **publié mon agent BookBot** ainsi que le **tourist-agent** sur la plateforme NEAR AI, ce qui permet à n'importe quel utilisateur de dialoguer directement avec lui et d'obtenir des suggestions personnalisées.

- Lien de mon agent BookBot :

<https://app.near.ai/agents/3arbia.near/hello-agent/latest/run>

- Lien de mon agent Tourist-Agent :

<https://app.near.ai/agents/3arbia.near/tourist-agent/latest/run>

Conclusion

Ce stage chez **NEAR DevHub**, au sein de l'équipe **DevRel**, a été une véritable opportunité pour moi de **découvrir en profondeur le monde de l'intelligence artificielle**, des modèles LLM et du développement Python. En partant d'un niveau débutant, j'ai pu progresser rapidement grâce à des missions concrètes, de l'accompagnement, mais aussi beaucoup de pratique.

J'ai appris à utiliser des outils professionnels (comme Git, GitHub, Visual Studio Code, NEAR AI Hub), à manipuler des modèles LLM variés (LLaMA, Qwen, DeepSeek, etc.), à structurer des agents intelligents, et à enrichir leurs capacités avec des **vector stores**. J'ai aussi compris l'importance de la documentation et de l'organisation du code dans un vrai environnement de travail.

Ce stage m'a permis de **gagner en autonomie**, de développer ma capacité à **chercher, tester, corriger**, mais surtout à **mener un projet complet jusqu'au bout**, comme ce fut le cas avec mon agent final **BookBot**.

Je ressors de cette expérience avec des bases solides, une vraie motivation pour continuer dans ce domaine, et surtout la preuve que **même en partant de zéro, on peut créer des choses utiles, intelligentes et personnelles avec de la volonté et de la rigueur**.