Réalisation d’une application de type RAG

**      **

Certification : Développeur d’application en Intelligence Artificiel

Marseille – Nice

Bloc de compétences 3 - E4

Réaliser une application intégrant un service d’intelligence artificielle

LOMBARDI Joachim

# A6. Compétence : Conception d'une application intégrant un service d'IA

## C14. Analyse du Besoin et Spécifications Fonctionnelles

Il s’agit d’une application permettant de poser des questions sur des articles médicaux issus du site internet PubMed. La méthode Merise a été utilisée pour réaliser la base de données. L’interface utilisateur est simple et lisible. Des messages d’erreurs ou de succès sont affichés afin d’indiquer à l’utilisateur si sa démarche a été réalisée correctement.

## C15. Conception du Cadre Technique

L’architecture du projet est celle d’un projet Django à savoir une architecture MVT (Model-View-Template)

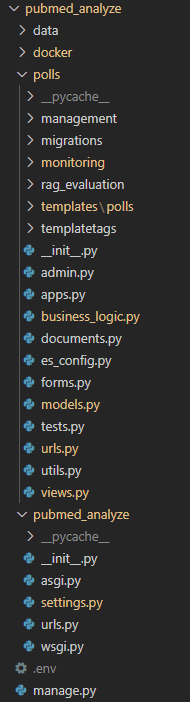


Figure . Arborescence du projet pubmed\_analyze

### Les éléments clés de l’architecture Django

#### manage.py

Un script utilisé pour exécuter des commandes Django comme :

* *python manage.py runserver* (lancer le serveur),
* *python manage.py migrate* (appliquer les migrations),
* *python manage.py createsuperuser* (créer un admin).

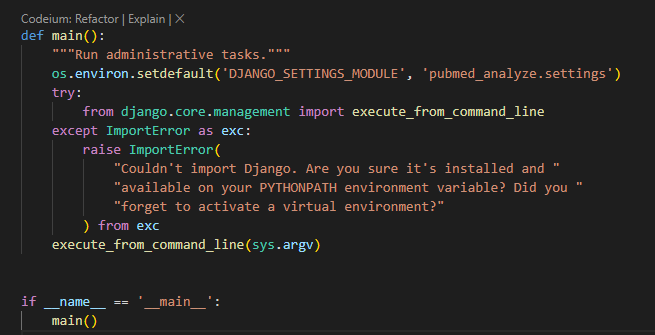


Figure . manage.py

#### 2. settings.py (Configuration du projet)

Ce fichier contient les paramètres globaux du projet Django :

* Configuration de la base de données (PostgreSQL, MySQL, SQLite…).



Figure . settings.py -- bases de données

* Applications installées (INSTALLED\_APPS).

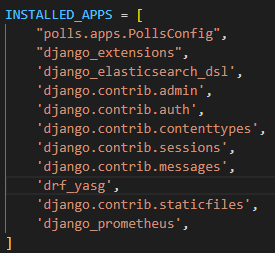


Figure 4. settings.py -- Installed apps

* Middleware (MIDDLEWARE) pour gérer les requêtes.

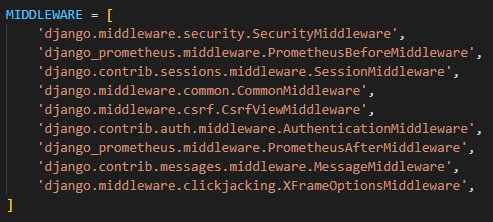


Figure . settings.py -- Middlewares

* Templates et fichiers statiques.

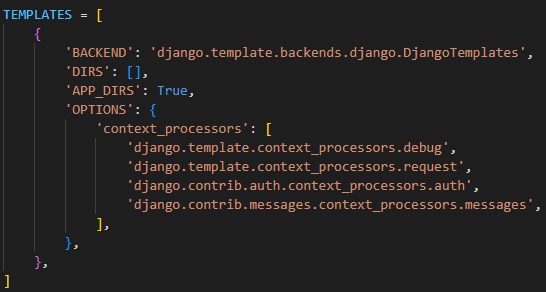


Figure . settings.py -- Templates

#### 3. urls.py (Gestion des URL)

Ce fichier définit le routage de l’application



Figure . urls.py

#### 4. models.py (Gestion de la base de données - ORM)

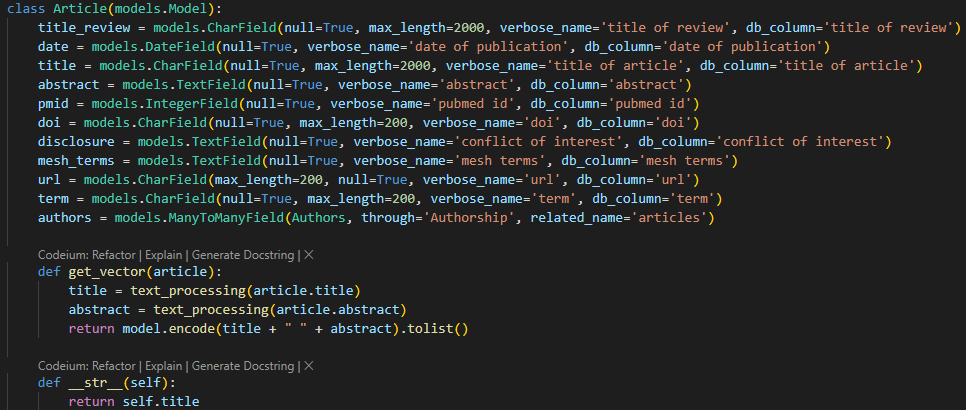


Figure . models.py -- Articles

Les modèles définissent la structure de la base de données en Python. Django génère ensuite automatiquement les tables SQL.

#### 5. views.py (Logique de l'application)

Les vues définissent la logique des pages web ou API.

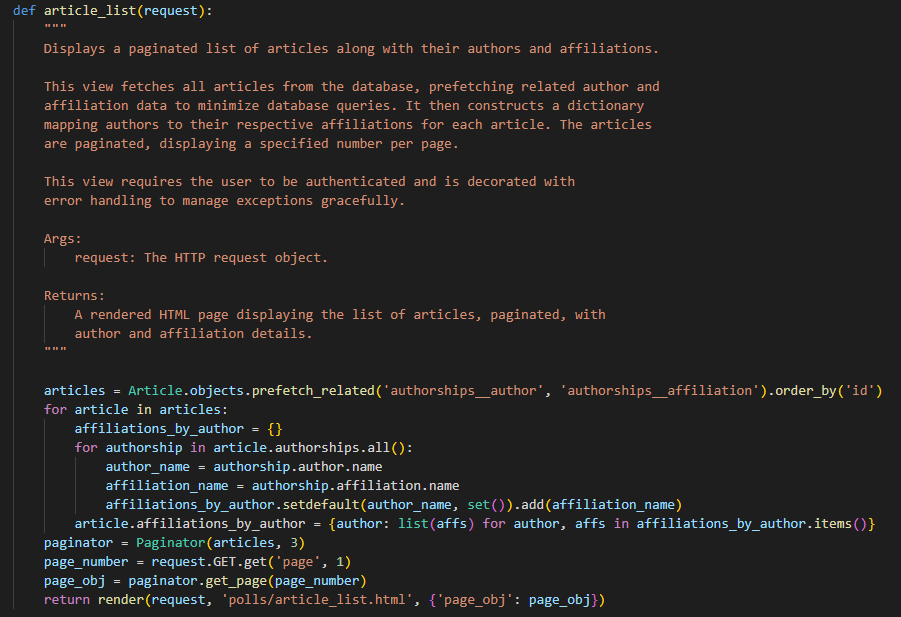


Figure . view.py -- article\_list

#### 6. templates/ (Système de Templates - HTML)

Django utilise un moteur de templates pour générer du HTML dynamique.

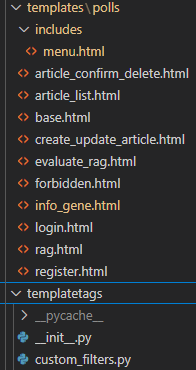


Figure . Arborescence des templates

#### 7. forms.py (Gestion des formulaires)

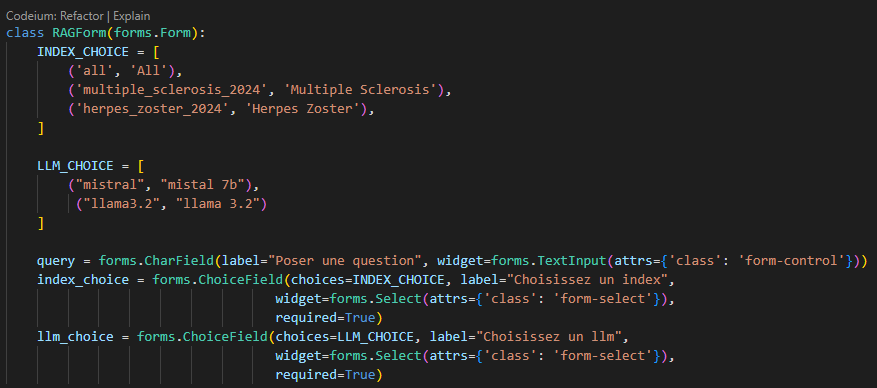


Figure . forms.py – RAGForm

Il y a également éléments spécifiques à mon projet :

#### 8. Docker (Gestion des conteneurs)

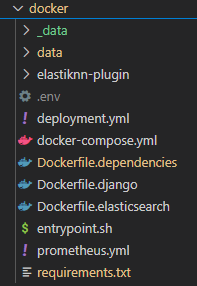


Figure . Fichier docker

C’est aussi là qu’est stocké mon fichier requirements.txt avec toutes les bibliothèques.

#### 9. Management (Commandes Django)

Elles me permettent d’exécuter des fonctions dans l’invite de commande PowerShell (sans requête http)

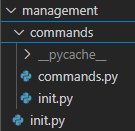


Figure . Fichier management

#### 10. Monitoring (Récupère les métriques du RAG)

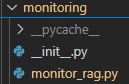


Figure . Fichier monitoring

#### 11. RAG\_evaluation (Evalue le RAG)

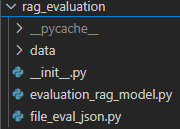


Figure . rag\_evaluation

# A7. Compétence : Développement des interfaces et fonctionnalités

## C16. Coordination de la Réalisation Technique

### Conduite agile de projet.

Afin de faire de la gestion de projet, il existe plusieurs outils dont voici les principaux :  
Voici les principales méthodes de gestion de projet agile :

#### SCRUM

Scrum est une méthodologie Agile qui vise à améliorer la flexibilité, la collaboration et l’efficacité dans le développement de produits, notamment en IT et en Machine Learning. Il repose sur des cycles courts appelés sprints, permettant une adaptation rapide aux changements.

Principaux Rôles dans Scrum :

* Scrum Master → Facilite le processus, élimine les obstacles et assure le respect des principes Scrum.
* Product Owner → Définit les besoins du produit et priorise les tâches dans le backlog.
* Équipe de développement → Développe, teste et livre les fonctionnalités du produit.

Événements Clés de Scrum :

* Sprint Planning → Planification des tâches à accomplir pendant le sprint (1 à 4 semaines).
* Daily Scrum (Stand-up Meeting) → Réunion quotidienne de 15 minutes pour synchroniser l’équipe.
* Sprint Review → Présentation des fonctionnalités développées à la fin du sprint.
* Sprint Retrospective → Analyse des points forts et axes d’amélioration pour le sprint suivant.

Artefacts Scrum :

* Product Backlog → Liste des fonctionnalités à développer.
* Sprint Backlog → Liste des tâches sélectionnées pour un sprint.
* Increment → Version améliorée du produit après chaque sprint.

SCRUM est la méthode agile la plus populaire.

#### Kanban

Kanban est une méthode Agile similaire à Scrum mais plus flexible et sans cycles fixes (sprints). Elle se concentre sur la gestion des flux de travail et l'amélioration continue.

Principaux Concepts :

* Tableau Kanban avec trois colonnes principales : À faire, En cours, Terminé.
* Limite de travail en cours (WIP) : Limiter le nombre de tâches dans une colonne pour éviter la surcharge.
* Amélioration continue : Revue régulière des processus pour améliorer le flux de travail.

Avantages :

* Flexibilité et adaptabilité pour les équipes en évolution.
* Suivi visuel des tâches en temps réel.

Outils populaires :

* Trello (qui propose aussi un mode Kanban)
* Jira

#### XP (Extreme Programming) : Méthode Agile Technique.

XP (Extreme Programming) se concentre sur l'amélioration de la qualité du code et l'agilité technique. C'est une approche fortement utilisée pour les projets de développement logiciel.

Principaux Concepts :

* Développement itératif : Des cycles de développement courts pour livrer rapidement des versions du produit.
* Code de qualité : Utilisation de pratiques comme la revue de code, les tests automatisés et l'intégration continue.

Avantages :

* Haute qualité du code.
* Adaptation rapide aux besoins du client.

Outils populaires :

* Jenkins (pour l'intégration continue)
* GitLab ou GitHub (gestion du code)

La méthode SCRUM est la méthode la plus pratique lors de travail en équipe car elle permet de visualiser et corriger l’évolution du projet à tout instant, cependant dans ce cas précis le projet à été réalisé en solo à titre individuel ainsi il n’y a jamais eu de présentation au client ni à l’équipe. A noter, que ces méthodes peuvent être complémentaires car Kanban permet de visualiser le SCRUM s’il y en a un. Ci-dessous une présentation des principaux outils de visualisation pour gérer les tâches :D.

#### Trello

Trello est un outil de gestion de projet visuel, basé sur la méthodologie Kanban. Il est souvent utilisé pour gérer les sprints Scrum de manière simple et intuitive.

Structure d’un Tableau Trello

* Tableau → Représente un projet.
* Listes → Représentent les étapes du projet (ex: À faire, En cours, Terminé).
* Cartes → Représentent les tâches spécifiques (ex: “Déployer un modèle ML”).
* Étiquettes, deadlines et membres assignés → Facilitent l’organisation des tâches.

Exemple d’Utilisation de Trello dans Scrum

* Product Owner ajoute les tâches dans le Product Backlog.
* L’équipe choisit les tâches pour le sprint et les place dans la liste "À faire".
* Pendant le sprint, les tâches passent par "En cours" puis "Terminé" après validation.

#### Jira

Jira, développé par Atlassian, est l'un des outils les plus populaires dans le domaine Agile et Scrum. Il est particulièrement adapté pour des projets logiciels ou des équipes de développement.

Caractéristiques principales :

* Suivi détaillé des sprints et des backlogs.
* Possibilité d’intégrer des tableaux Kanban et Scrum pour la gestion des tâches.
* Outils de gestion de version et de test intégrés.

Avantages :

* Très adapté aux équipes techniques.
* Suivi détaillé des problèmes, tâches et bugs.

Inconvénients :

* Peut-être complexe pour des projets non techniques.
* Courbe d’apprentissage.

#### Asana : Outil de Gestion de Projet Collaboratif

Asana est une alternative populaire pour la gestion des tâches et la collaboration d’équipe. Il peut être utilisé pour différents types de projets, y compris ceux en Agile et en Scrum.

Caractéristiques principales :

* Suivi des tâches avec des listes de tâches ou des tableaux Kanban.
* Assignation des tâches, priorisation, et suivi des progrès.
* Collaboration en temps réel avec commentaires et pièces jointes.

Avantages :

* Interface conviviale et flexible.
* Collaboration fluide pour les équipes.

Inconvénients :

* Moins de fonctionnalités spécifiques pour les équipes de développement logiciel.
* Manque de certaines intégrations avancées.

Trello est l’outil le plus populaire, il est simple et complet de plus je l’ai déjà utilisé dans le cadre de ma formation. Voici un Trello réalisé dans un projet antérieur :

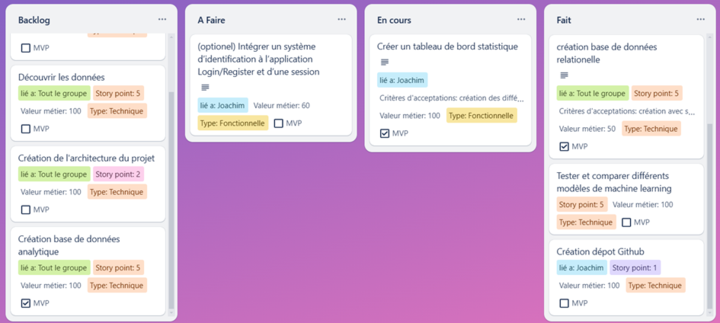
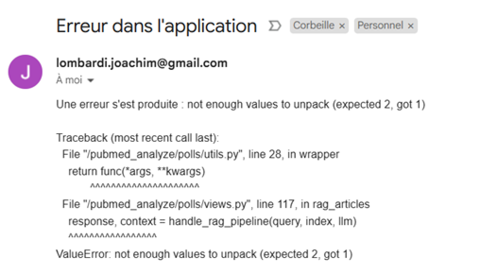


Figure . Kanban

Les dates n’y sont pas présentes mais on peut y voir les différentes tâches à effectuer, celui ou ceux qui ont à l’effectuer, les story points représentant l’effort à fournir pour réaliser la tâche et les valeurs métiers désignant l’importance de la tâche pour le client ou l’entreprise. Également, si la tâche est essentielle au MVP (Produit Minimum Viable) à savoir une version contenant les parties essentielles du projet.

Exemple d’une erreur.

Lorsque mon programme a été exécuté, j’ai reçu l’email suivant :



J’ai donc ouvert un ticket sur GitHub. L’erreur indique que deux valeurs sont attendues mais qu’il n’en a eu qu’une. Ainsi, la fonction handle\_rag\_pipeline renvoie une valeur en sortie mais deux variables sont proposées. Après analyse, il y avait deux erreurs, la première entraînant l’exception à savoir le mot « llm » a été passé dans les paramètres de la fonction generation à la place de « model ». Ce qui a entraîné une exception mais cette exception renvoie une réponse JSON et non deux variables.

## C17. Développement des Composants et Interfaces

Les prérequis pour installer l’environnement de développement sont indiqués sur le README : <https://github.com/JoachimLombardi/Projet_chef_d-oeuvre/blob/master/README.md>

Le Front-End a été développer avec Django + bootstrap.



Figure . extrait du fichier base.html montrant l'appel à bootstrap