



ÉCOLE NATIONALE SUPÉRIEURE
D'INFORMATIQUE ET D'ANALYSE DES SYSTÈMES
- RABAT

BUSINESS INTELLIGENCE AND ANALYTICS

Projet : Automatisation des Pipelines
avec n8n , développement d'une
Application avec Vibe Coding

Élèves :

Salma JENNANE
Salma BOUTANFIT
Yassine MICHAAL
Youssef RIZKI

Enseignant :

N. KERZAZI

25 mai 2025

Remerciements

Nous tenons à remercier chaleureusement notre encadrant, **N. Kerzazi**, pour sa disponibilité, son expertise et son accompagnement tout au long de ce projet. Nous remercions également l'ENSIAS pour le cadre propice à la recherche et à l'innovation, ainsi que nos proches pour leur soutien inestimable.

Résumé

Dans le cadre de ce projet, nous explorons l'automatisation des pipelines MLOps à l'aide de nouvelles approches émergentes telles que l'Agentic AI, la plateforme MCP et le paradigme de développement Vibe Coding. Le projet se décline en deux volets : (1) la reconstruction d'un pipeline MLOps avec n8n, intégrant des modèles de traitement du langage naturel et orchestré via des déclencheurs WhatsApp et des notifications Gmail, et (2) le développement d'une application interactive via Windsurf, mettant en œuvre le Vibe Coding. L'ensemble des interactions avec les IA a été journalisé pour en tirer des enseignements à des fins de recherche.

Abstract

This project explores the automation of MLOps pipelines using emerging approaches such as Agentic AI, the MCP platform, and the Vibe Coding development paradigm. It includes two main components : (1) the reconstruction of an MLOps pipeline with n8n, integrating NLP models and orchestrated through WhatsApp triggers and Gmail notifications, and (2) the development of an interactive application via Windsurf using Vibe Coding. All interactions with AI tools were logged for further research and analysis.

Introduction générale

La montée en puissance de l'intelligence artificielle générative et des plateformes no-code/low-code a profondément modifié la manière de concevoir des solutions intelligentes et automatisées. Dans ce contexte, le concept d'**Agentic AI**, qui consiste à déléguer des tâches complexes à des agents autonomes orchestrés par des instructions de haut niveau, s'impose comme un paradigme révolutionnaire.

Parallèlement, des outils comme **n8n** permettent de construire des workflows d'automatisation personnalisés intégrant diverses API et modèles de machine learning, tandis que la **MCP** (Multi-agent Cloud Platform) facilite le déploiement distribué de ces agents. Le **Vibe Coding**, quant à lui, introduit une nouvelle manière de coder à travers des interfaces conversationnelles ou visuelles, avec un focus sur la collaboration homme-machine.

Ce projet s'inscrit dans cette dynamique d'innovation et vise à construire des pipelines intelligents en exploitant pleinement ces nouveaux outils et concepts. Il se compose de deux parties : une première orientée MLOps avec n8n, et une seconde axée sur le développement d'une application utilisant Vibe Coding avec Windsurf. Ce rapport documente le travail réalisé, les choix techniques, les défis rencontrés, ainsi que les perspectives ouvertes par cette nouvelle génération d'outils IA.

Table des matières

| | |
|--|-----------|
| Remerciements | 1 |
| Résumé | 2 |
| Abstract | 3 |
| Introduction générale | 4 |
| Introduction Générale | 8 |
| 1 Contexte Théorique : Agentic AI, MCP et Vibe Coding | 9 |
| 1.1 Agentic AI | 9 |
| 1.2 MCP (Multi-agent Cloud Platform) | 9 |
| 1.3 Vibe Coding | 9 |
| 2 Système MLOps pour l'analyse automatisée des commentaires YouTube | 10 |
| 2.1 Introduction | 10 |
| 2.2 Architecture du système | 11 |
| 2.3 Workflow d'analyse avec n8n | 11 |
| 2.3.1 Déclenchement et validation | 11 |
| 2.3.2 Extraction des données YouTube | 13 |
| 2.3.3 Stratégie d'échantillonnage et gestion des contraintes d'API | 14 |
| 2.3.4 Préparation des données pour l'analyse | 15 |
| 2.3.5 Analyse multimodale avec IA | 16 |
| 2.3.6 Fusion et synthèse des résultats | 18 |
| 2.3.7 Génération et diffusion du rapport | 18 |
| 2.4 Améliorations futures | 19 |
| 2.5 Conclusion | 20 |
| 3 Développement d'une Application avec Vibe Coding | 21 |
| 3.1 Objectifs de l'application | 21 |
| 3.2 Utilisation de Windsurf et design de l'application | 21 |
| 3.3 Composants de l'Interface Utilisateur | 22 |
| 3.3.1 Bannière d'En-tête | 22 |
| 3.3.2 Panneau Central : Horaires de Prière | 22 |
| 3.3.3 Panneau Latéral Gauche : « Today's Dhikr » (Dhikr du Jour) | 23 |
| 3.3.4 Panneau Latéral Droit : « Today's Hadith » (Hadith du Jour) | 23 |
| 3.4 Conception de l'Interface (UI/UX) | 23 |
| 3.5 Public Cible | 23 |

| | |
|--|-----------|
| 3.6 Fonctionnalités Potentielles (Non Visibles Mais Suggérées) | 24 |
| Conclusion Générale | 25 |
| Références | 26 |

Table des figures

| | | |
|------|--|----|
| 2.1 | Vue d'ensemble du workflow n8n pour l'analyse des commentaires YouTube | 10 |
| 2.2 | Structure des conteneurs Docker du système MLOps | 11 |
| 2.3 | Interface Twilio Sandbox pour WhatsApp montrant le déclenchement du workflow avec la commande "GoDeploy" | 12 |
| 2.4 | Console ngrok montrant le tunnel établi vers notre serveur n8n local | 12 |
| 2.5 | Nœud de validation du déclencheur | 13 |
| 2.6 | Boucle d'extraction des commentaires par vidéo | 14 |
| 2.7 | Extrait code pour sélectionner les commentaires | 14 |
| 2.8 | Exemple d'erreur rencontrée lors du dépassement des limites de l'API Hugging Face | 15 |
| 2.9 | Modèle Hugging Face pour l'analyse de sentiment | 16 |
| 2.10 | Modèle Hugging Face pour la détection d'émotions | 16 |
| 2.11 | Modèle Hugging Face pour détecter les commentaires toxiques | 17 |
| 2.12 | Configuration du nœud HTTP Request pour l'analyse de toxicité via l'API Hugging Face.. . . . | 17 |
| 2.13 | Noeud fusion résultats | 18 |
| 2.14 | Exemple de rapport d'analyse reçu par email1 | 19 |
| 2.15 | Exemple de rapport d'analyse reçu par email2 | 19 |
| 2.16 | Exemple de rapport d'analyse reçu par email3 | 19 |
| 3.1 | Interface de notre application | 22 |

Introduction Générale

L'émergence des intelligences artificielles génératives et des plateformes no-code/low-code a inauguré une ère de transformation profonde dans la conception de solutions numériques. Au cœur de cette métamorphose se trouve le concept d'**Agentic AI** – une vision où des agents autonomes intelligents exécutent des séquences complexes d'opérations en réponse à des directives humaines de haut niveau, franchissant ainsi le seuil entre l'outil passif et le collaborateur actif.

Dans ce paysage en mutation, plusieurs technologies clés cristallisent cette évolution. La plateforme **n8n** permet d'entrelacer APIs, services cloud et modèles d'IA dans des workflows sophistiqués sans expertise approfondie en programmation. Plus révolutionnaire encore, le paradigme du **Vibe Coding** propose une approche conversationnelle où le code émerge naturellement d'un dialogue entre le développeur et l'intelligence artificielle.

Notre projet s'inscrit à cette intersection fertile entre l'automatisation intelligente et la création assistée par IA. Il se déploie selon deux axes complémentaires : d'une part, l'élaboration d'un système MLOps avec n8n pour l'analyse automatisée de commentaires YouTube, illustrant la puissance de l'orchestration d'agents dans un contexte d'application concrète ; d'autre part, le développement d'une application d'horaires de prière via Vibe Coding avec Windsurf, explorant les nouvelles frontières de la collaboration homme-machine dans la conception d'interfaces.

Ce rapport documente notre parcours d'exploration, les défis rencontrés et surmontés, ainsi que les perspectives qu'offrent ces nouveaux paradigmes pour l'avenir de la création numérique et de l'intelligence artificielle appliquée.

Chapitre 1

Contexte Théorique : Agentic AI, MCP et Vibe Coding

1.1 Agentic AI

L'Agentic AI désigne un cadre dans lequel des agents autonomes sont capables d'exécuter des tâches de manière proactive, en utilisant des modèles de langage avancés (comme GPT, Claude, etc.) pour raisonner, planifier et agir. Contrairement à une simple requête utilisateur, un agent agentique possède un cycle de vie complet : il peut analyser une tâche, la décomposer en sous-tâches, interagir avec des outils externes, et itérer jusqu'à obtenir un résultat satisfaisant. Ce paradigme est particulièrement adapté à l'automatisation des workflows intelligents.

1.2 MCP (Multi-agent Cloud Platform)

La MCP est une infrastructure cloud conçue pour l'exécution distribuée d'agents autonomes. Elle permet d'assigner à chaque agent des rôles spécifiques, tout en facilitant leur communication et leur coordination. Dans notre projet, la MCP est utilisée pour héberger les agents impliqués dans le pipeline n8n, ce qui garantit une exécution fluide, scalable et traçable des différentes étapes de traitement.

1.3 Vibe Coding

Le Vibe Coding représente une approche innovante de la programmation, centrée sur l'expérience développeur et la co-crétation avec des intelligences artificielles. Elle privilégie l'utilisation de plateformes conversationnelles, de blocs visuels ou d'outils comme Windsurf pour créer des applications de manière fluide, intuitive et interactive. Dans notre cas, Vibe Coding a permis de développer une application complète tout en enregistrant les suggestions, corrections et interactions IA pour analyse future.

Chapitre 2

Système MLOps pour l'analyse automatisée des commentaires YouTube

2.1 Introduction

Dans ce chapitre, nous présentons un système MLOps complet pour l'analyse automatisée des commentaires YouTube. Ce système utilise une approche d'intégration continue et de déploiement continu (CI/CD) pour l'analyse de sentiment et d'émotion des commentaires de vidéos YouTube, en combinant plusieurs technologies modernes : n8n pour l'orchestration des workflows, des modèles d'intelligence artificielle (Hugging Face et MCP), Twilio pour les notifications.

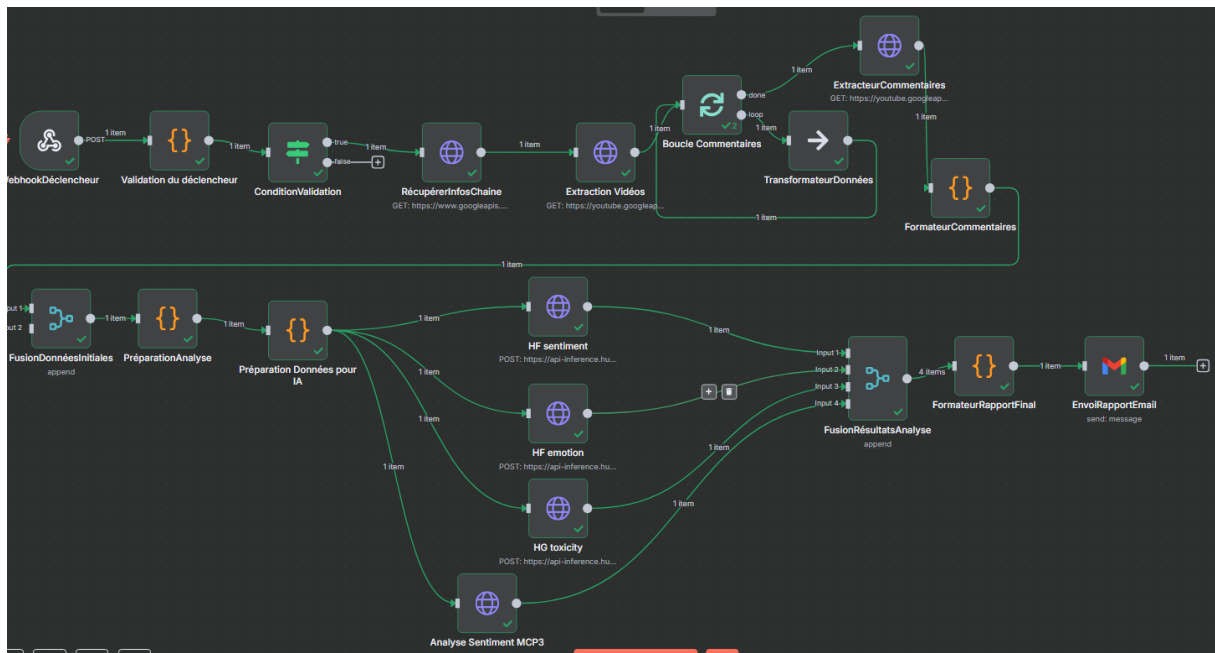


FIGURE 2.1 – Vue d'ensemble du workflow n8n pour l'analyse des commentaires YouTube

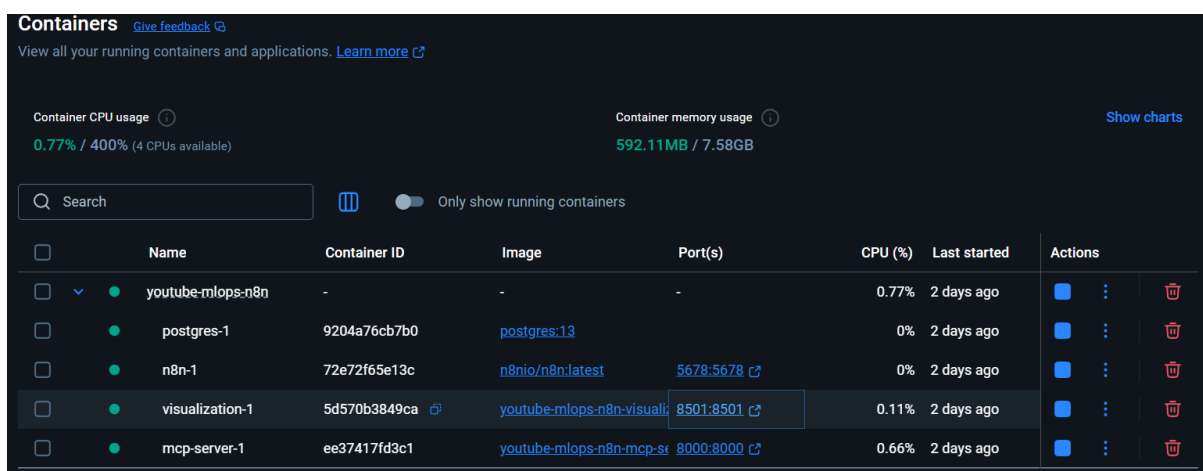
Ce système permet aux créateurs de contenu et aux gestionnaires de communauté d'obtenir des insights précieux sur la réception de leurs vidéos par leur audience, en identifiant

automatiquement les sentiments dominants, les émotions exprimées, et en détectant les commentaires potentiellement problématiques ou spammeurs.

2.2 Architecture du système

Le système MLOps développé est basé sur une architecture microservices déployée avec Docker, comprenant cinq composants principaux :

- **n8n** : Plateforme d'automatisation de workflow, responsable de l'orchestration du processus d'analyse
- **mcp-server** : Service d'analyse de sentiment multi-catégories personnalisé
- **Hugging Face** : Modèles d'IA pour la détection d'émotions, de sentiments et de toxicité
- **postgres** : Base de données pour stocker les résultats et configurations
- **visualization-app** : Application Streamlit pour visualiser les résultats d'analyse



| | Name | Container ID | Image | Port(s) | CPU (%) | Last started | Actions |
|--------------------------|-------------------|--------------|---------------------------|-----------|---------|--------------|---------|
| <input type="checkbox"/> | youtube-mlops-n8n | - | - | - | 0.77% | 2 days ago | |
| <input type="checkbox"/> | postgres-1 | 9204a76cb7b0 | postgres:13 | - | 0% | 2 days ago | |
| <input type="checkbox"/> | n8n-1 | 72e72f65e13c | n8nio/n8n:latest | 5678:5678 | 0% | 2 days ago | |
| <input type="checkbox"/> | visualization-1 | 5d570b3849ca | youtube-mlops-n8n-visuali | 8501:8501 | 0.11% | 2 days ago | |
| <input type="checkbox"/> | mcp-server-1 | ee37417fd3c1 | youtube-mlops-n8n-mcp-s | 8000:8000 | 0.66% | 2 days ago | |

FIGURE 2.2 – Structure des conteneurs Docker du système MLOps

Cette architecture permet une séparation des préoccupations claire, où chaque service est responsable d'une fonction spécifique, tout en facilitant l'évolutivité et la maintenance du système.

2.3 Workflow d'analyse avec n8n

2.3.1 Déclenchement et validation

Le workflow est déclenché par un webhook qui peut être appelé via différentes interfaces. Pour notre implémentation, nous avons privilégié l'utilisation de WhatsApp comme interface principale, offrant ainsi une expérience utilisateur fluide et accessible depuis n'importe quel appareil mobile.

Intégration de Twilio pour WhatsApp

Pour permettre l'interaction via WhatsApp, nous avons intégré la plateforme Twilio, un service cloud de communication qui facilite l'envoi et la réception de messages via

différents canaux.

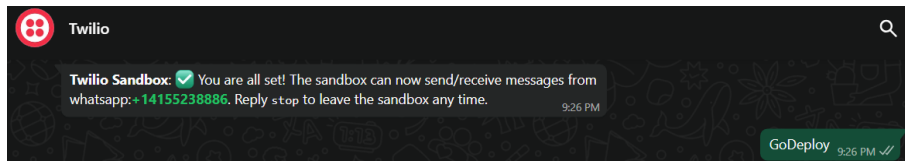


FIGURE 2.3 – Interface Twilio Sandbox pour WhatsApp montrant le déclenchement du workflow avec la commande "GoDeploy"

La configuration de Twilio implique la création d'un sandbox WhatsApp qui fournit un numéro de téléphone virtuel (+14155238886) auquel les utilisateurs peuvent envoyer des messages. Comme illustré dans la Figure 2.3, l'utilisateur envoie simplement le mot-clé "GoDeploy" pour déclencher l'analyse complète des commentaires YouTube.

Exposition sécurisée avec ngrok

Pour que Twilio puisse communiquer avec notre instance n8n hébergée localement, nous avons utilisé ngrok, un outil qui crée un tunnel sécurisé vers notre serveur local et l'expose à Internet via une URL publique.

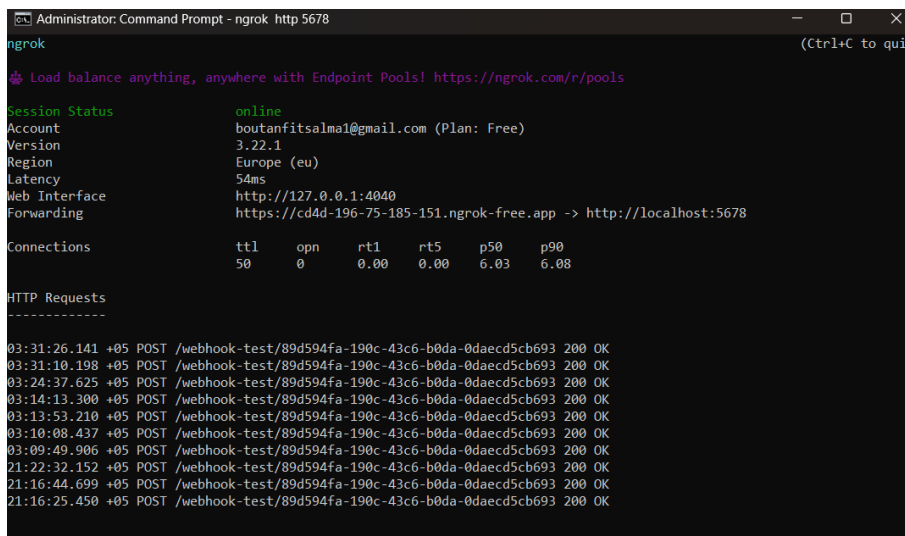


FIGURE 2.4 – Console ngrok montrant le tunnel établi vers notre serveur n8n local

Listing 2.1 – URL de webhook exposée avec ngrok

`https://cd4d-196-75-185-151.ngrok-free.app -> http://localhost:5678`

Cette configuration crée un pont essentiel dans notre architecture : les messages WhatsApp sont d'abord reçus par Twilio, qui les transmet ensuite à notre URL ngrok sous forme de requêtes HTTP POST. Ces requêtes sont automatiquement acheminées vers notre serveur n8n local sur le port 5678, où le nœud WebhookDéclencheur les intercepte et démarre le workflow d'analyse.

L'utilisation de ngrok présente plusieurs avantages dans notre contexte de développement et de déploiement :

- Élimination du besoin d'un serveur avec une adresse IP publique ou un nom de domaine
- Inspection et débogage faciles des requêtes entrantes via le tableau de bord ngrok
- Mise en place rapide d'un environnement de test ou de démonstration
- Sécurisation des communications avec un chiffrement TLS automatique

Cette approche événementielle, où le système n'est activé que lorsqu'un message "GoDeploy" est reçu, garantit une utilisation optimale des ressources et permet une intégration fluide entre les services cloud (Twilio) et notre infrastructure locale (n8n).

Pour sécuriser le système, un nœud de validation est implémenté pour vérifier que le déclencheur contient un mot-clé spécifique ("GoDeploy"). Cette validation simple mais efficace évite les déclenchements accidentels et ajoute une couche de sécurité. Cela a été fait avec le nœud IF.

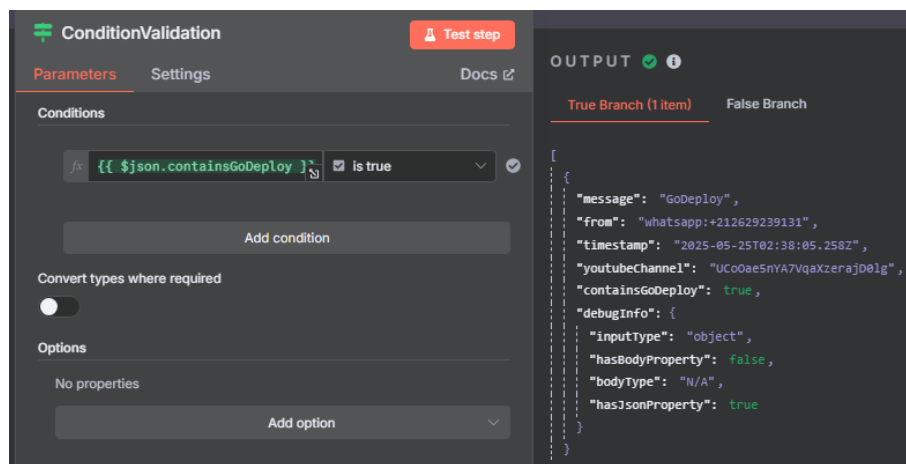


FIGURE 2.5 – Nœud de validation du déclencheur

2.3.2 Extraction des données YouTube

Une fois le déclenchement validé, le workflow procède à l'extraction des données YouTube via l'API YouTube Data. Cette extraction se fait en deux étapes principales :

1. **RécupérerInfosChaine** : Récupère les informations de base sur la chaîne YouTube
2. **Extraction Vidéos** : Obtient la liste des vidéos récentes de la chaîne
3. **ExtracteurCommentaires** : Pour chaque vidéo, extrait les commentaires associés

La récupération des commentaires est implémentée dans une boucle, permettant de traiter chaque vidéo individuellement et d'éviter les limitations de l'API YouTube.

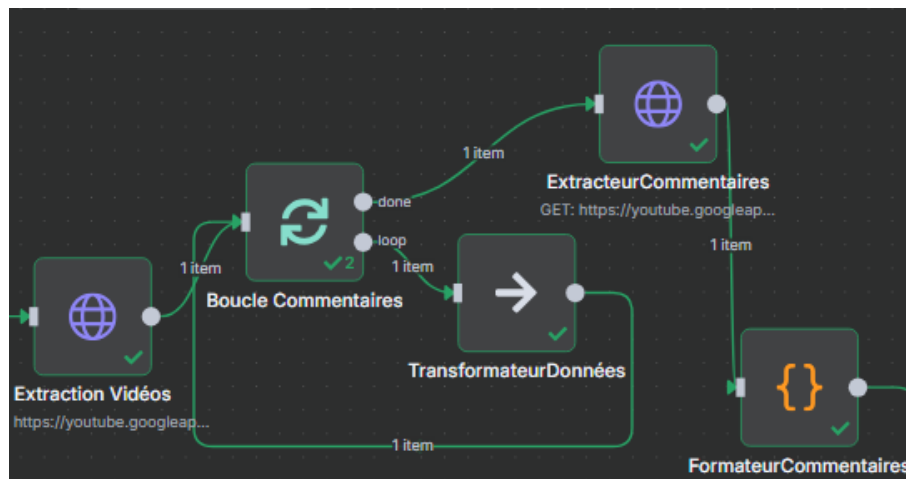


FIGURE 2.6 – Boucle d'extraction des commentaires par vidéo

2.3.3 Stratégie d'échantillonnage et gestion des contraintes d'API

L'analyse de tous les commentaires d'une vidéo YouTube populaire peut représenter un volume considérable de données, ce qui pose deux défis majeurs : la limitation des API gratuites et le temps d'exécution du workflow. Pour résoudre ces problèmes, nous avons implémenté une stratégie d'échantillonnage intelligente.

Sélection des commentaires

```
// 1. Trier les commentaires par différents critères
// Par nombre de likes (les plus populaires)
const popularComments = [...allComments]
  .sort((a, b) => b.likeCount - a.likeCount)
  .slice(0, 5)
  .map(cleanAndTruncateComment);

// Par date de publication (les plus récents)
const recentComments = [...allComments]
  .sort((a, b) => new Date(b.publishedAt) - new Date(a.publishedAt))
  .slice(0, 5)
  .map(cleanAndTruncateComment);

// Filtrer les commentaires courts (moins de MAX_TEXT_LENGTH caractères) pour les aléatoires
const shortComments = allComments.filter(c =>
  (c.text && c.text.length < MAX_TEXT_LENGTH) ||
  (c.textOriginal && c.textOriginal.length < MAX_TEXT_LENGTH)
);

// Commentaires aléatoires courts pour compléter
const randomComments = [...shortComments]
  .sort(() => Math.random() - 0.5)
  .slice(0, 10)
  .map(cleanAndTruncateComment);

// 2. Fusionner en éliminant les doublons
const selectedCommentIds = new Set();
const selectedComments = [];
```

FIGURE 2.7 – Extrait code pour sélectionner les commentaires

Notre algorithme de sélection des commentaires applique les critères suivants : Ce code sélectionne :

- 5 commentaires les plus populaires (avec le plus de likes)
- 5 commentaires les plus récents
- Complète avec des commentaires aléatoires pour atteindre 20 commentaires au total
- Élimine les doublons potentiels

Cette approche équilibrée nous permet d'obtenir un échantillon représentatif et analysable avec les ressources disponibles.

Gestion des limites des API Hugging Face

Les modèles Hugging Face accessibles gratuitement via leur API Inference imposent plusieurs contraintes techniques que nous avons dû gérer :

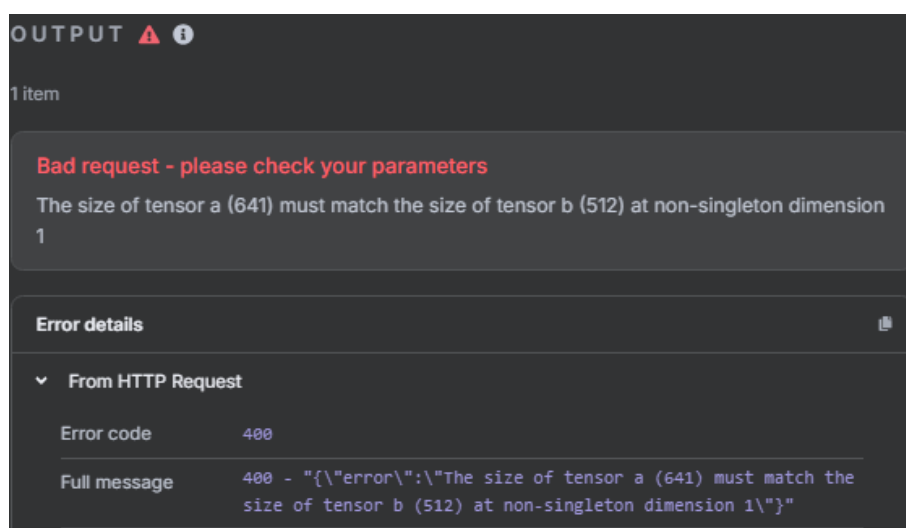


FIGURE 2.8 – Exemple d'erreur rencontrée lors du dépassement des limites de l'API Hugging Face

Ces contraintes incluent :

- **Limite de longueur des textes** : Maximum de 512 tokens pour la plupart des modèles
- **Quotas d'appels API** : Restrictions sur le nombre de requêtes par heure (généralement 60 requêtes/heure pour les comptes gratuits)
- **Temps de réponse variables** : Latence plus élevée pendant les périodes de forte demande

Cette stratégie d'échantillonnage et de gestion des contraintes d'API nous a permis d'obtenir des résultats d'analyse fiables tout en optimisant l'utilisation des ressources disponibles, démontrant ainsi un principe fondamental de MLOps : l'équilibre entre qualité des résultats et efficacité opérationnelle.

2.3.4 Préparation des données pour l'analyse

Les données brutes extraites de YouTube nécessitent une préparation avant d'être soumises aux modèles d'IA. Cette phase est cruciale pour assurer une analyse précise et pertinente.

Le nœud **TransformateurDonnées** normalise les données des commentaires, en extrayant les informations pertinentes telles que le texte, l'auteur, le nombre de likes, et la date de publication.

Le nœud **FormateurCommentaires** structure ces données dans un format compatible avec les APIs d'analyse de sentiment et d'émotion.

La fusion des données initiales (**FusionDonnéesInitiales**) est une étape clé qui consolide les métadonnées de la vidéo avec les commentaires pour maintenir la traçabilité et le contexte de l'analyse.

2.3.5 Analyse multimodale avec IA

Notre système utilise quatre modèles d'IA différents pour obtenir une analyse complète et nuancée des commentaires :

1. **HF sentiment** : Modèle Hugging Face pour l'analyse de sentiment (1 à 5 étoiles)

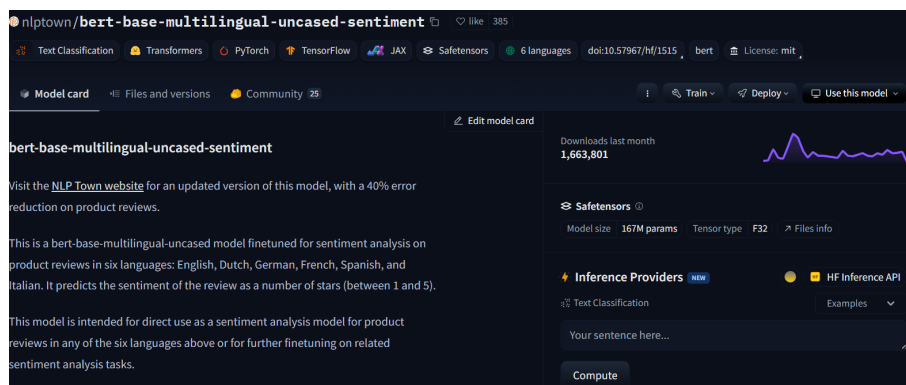


FIGURE 2.9 – Modèle Hugging Face pour l'analyse de sentiment

2. **HF emotion** : Modèle Hugging Face pour la détection d'émotions (joie, tristesse, colère, etc.)

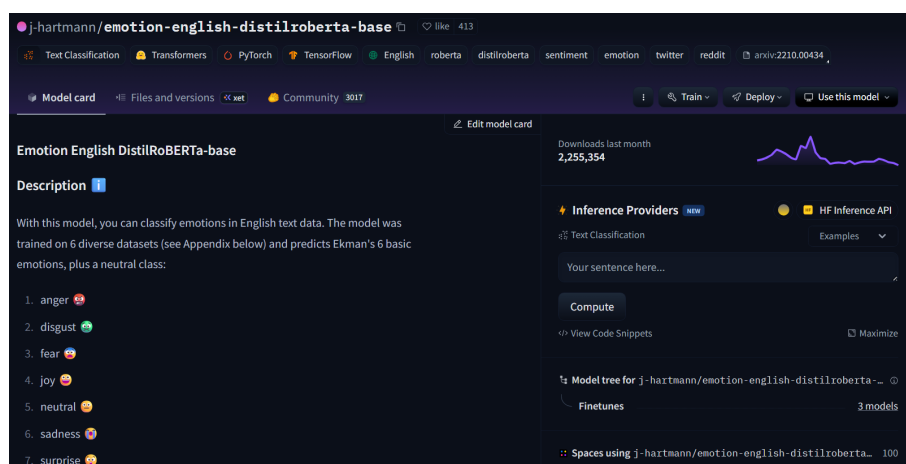


FIGURE 2.10 – Modèle Hugging Face pour la détection d'émotions

3. **HF toxicity** : Modèle Hugging Face pour détecter les commentaires toxiques

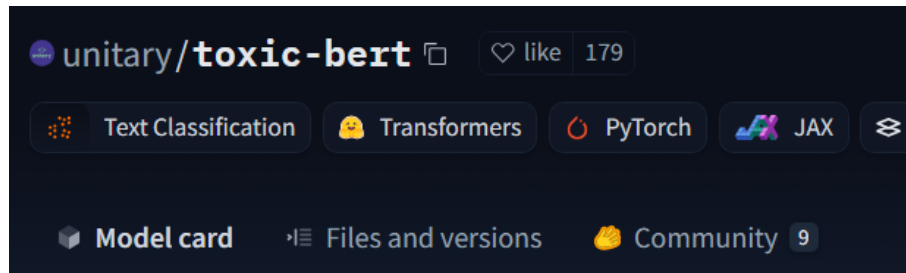


FIGURE 2.11 – Modèle Hugging Face pour détecter les commentaires toxiques

4. **Analyse Sentiment MCP** : Modèle personnalisé Multi-Category Perception pour une analyse de sentiment raffinée (positif/neutre/négatif)

Cette approche multimodale permet de capturer différentes dimensions du langage humain, offrant ainsi une analyse plus robuste et plus nuancée des commentaires.

Configuration des modèles Hugging Face

Pour l'analyse des sentiments, des émotions et de la toxicité des commentaires, nous avons intégré plusieurs modèles de la plateforme Hugging Face. Cette intégration a été réalisée via des nœuds HTTP Request dans n8n, configurés pour communiquer avec l'API Inference de Hugging Face.

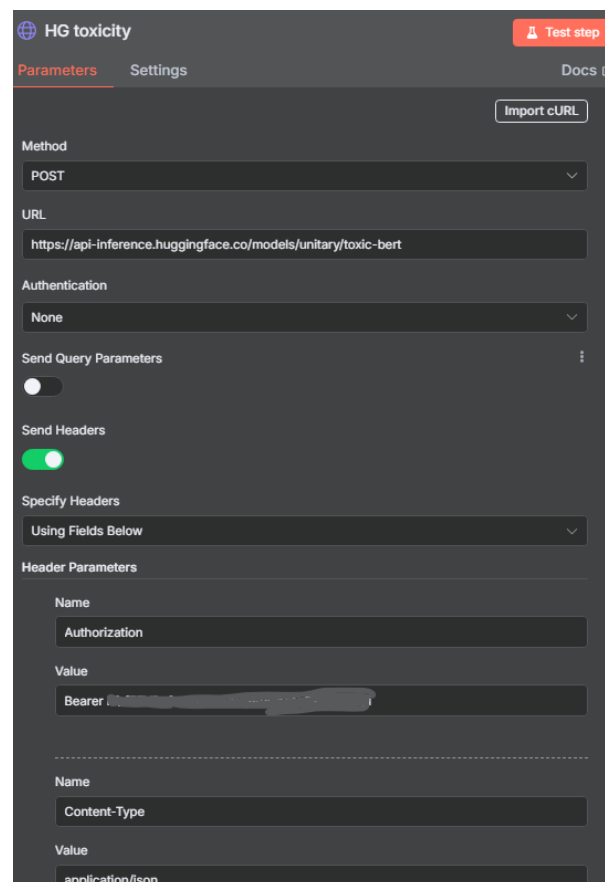


FIGURE 2.12 – Configuration du nœud HTTP Request pour l'analyse de toxicité via l'API Hugging Face..

Comme illustré dans la Figure 2.12, chaque nœud HTTP Request est configuré avec :

- Une méthode HTTP POST pour envoyer les données à analyser
- L'URL spécifique du modèle Hugging Face (dans ce cas, le modèle `unitary/toxic-bert` pour la détection de toxicité)
- Un en-tête d'authentification contenant un token API Hugging Face sécurisé
- Le type de contenu défini comme `application/json` pour la transmission des données

Cette intégration cloud représente un excellent exemple de l'approche MLOps moderne, où nous exploitons des modèles pré-entraînés via des API plutôt que de développer et d'entraîner nos propres modèles, réduisant ainsi considérablement le temps de développement et les ressources de calcul nécessaires.

2.3.6 Fusion et synthèse des résultats

Le nœud **FusionRésultatsAnalyse** joue un rôle critique dans l'agrégation des résultats des différents modèles d'IA.

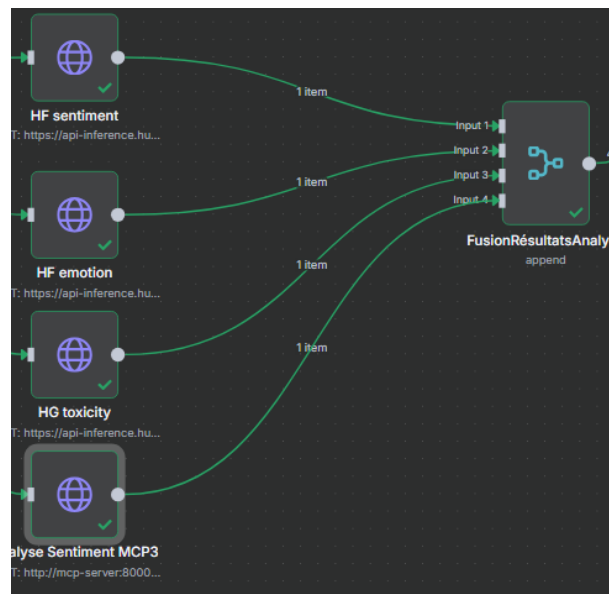


FIGURE 2.13 – Nœud fusion résultats

Cette approche dynamique permet au système de s'adapter à différentes structures de données et de maintenir la robustesse même en cas de changements dans les réponses des modèles.

2.3.7 Génération et diffusion du rapport

La dernière phase du workflow consiste à générer un rapport détaillé et à le diffuser aux parties prenantes.

Le nœud **FormateurRapportFinal** structure toutes les données d'analyse dans un format JSON cohérent, qui servira de base pour les différentes sorties.

Le rapport est ensuite envoyé par email via le nœud **EnvoiRapportEmail**, utilisant un template HTML soigneusement conçu pour présenter les résultats de manière claire.

Parallèlement, un fichier JSON contenant toutes les données d'analyse est sauvegardé dans un dossier partagé pour être utilisé par l'application de visualisation.

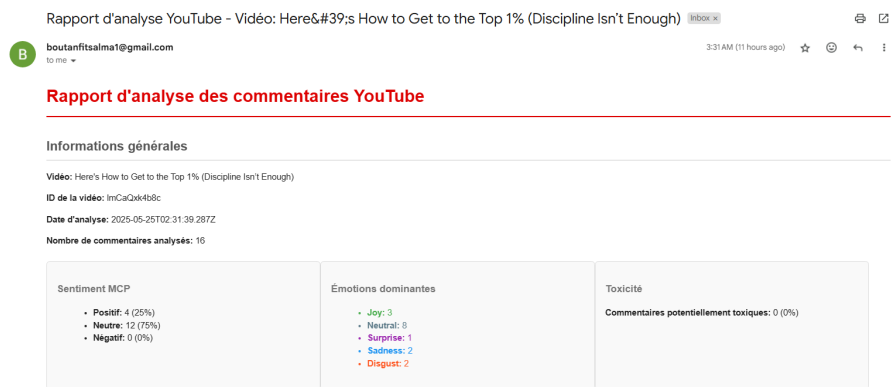


FIGURE 2.14 – Exemple de rapport d'analyse reçu par email1

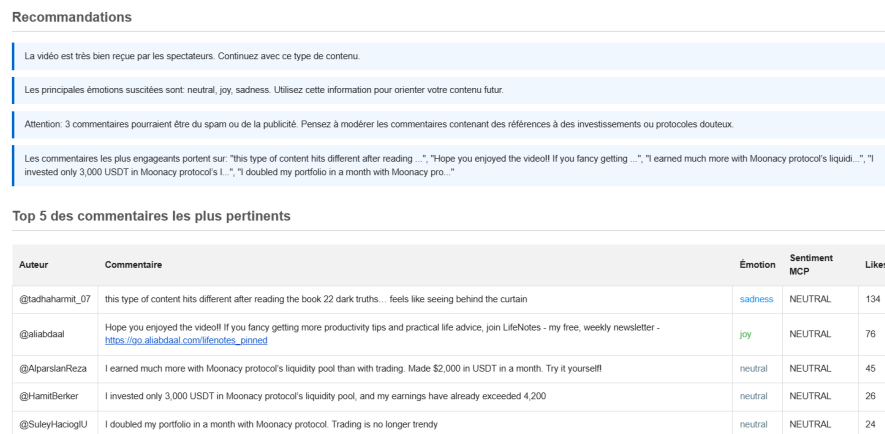


FIGURE 2.15 – Exemple de rapport d'analyse reçu par email2

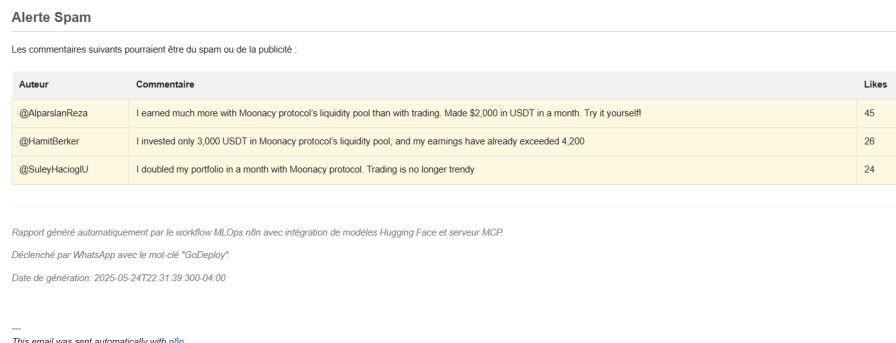


FIGURE 2.16 – Exemple de rapport d'analyse reçu par email3

2.4 Améliorations futures

Plusieurs améliorations sont envisagées pour les versions futures du système :

- **Analyse temporelle** : Implémenter un suivi de l'évolution des sentiments au fil du temps
- **Personnalisation des modèles** : Fine-tuner les modèles d'IA pour des domaines ou des chaînes YouTube spécifiques
- **Actions automatisées** : Suggérer ou implémenter des réponses automatisées aux commentaires basées sur leur analyse

2.5 Conclusion

Le système MLOps pour l'analyse des commentaires YouTube présenté dans ce chapitre démontre comment une architecture microservices moderne peut être utilisée pour créer un pipeline d'analyse complet et automatisé. En combinant des outils d'orchestration comme n8n, des modèles d'IA avancés, et des interfaces de visualisation intuitives, nous avons créé un système qui transforme des données brutes en insights actionnables pour les créateurs de contenu.

Ce système illustre parfaitement les principes MLOps d'automatisation, de reproductibilité et d'intégration continue, tout en offrant une valeur métier concrète. Les défis techniques rencontrés et les solutions mises en œuvre fournissent des leçons précieuses pour le développement de systèmes MLOps similaires dans d'autres domaines d'application.

Chapitre 3

Développement d'une Application avec Vibe Coding

3.1 Objectifs de l'application

L'objectif premier de cette application est d'assister les musulmans dans leur pratique religieuse quotidienne en fournissant des horaires de prière (Salat) précis et actualisés pour leur localisation, ainsi que du contenu spirituel complémentaire.

3.2 Utilisation de Windsurf et design de l'application

L'environnement Windsurf a été utilisé pour coder l'interface utilisateur en collaboration avec un LLM. L'approche Vibe Coding facilite la création de composants visuels par dialogue et correction automatique du code.

3.3 Composants de l'Interface Utilisateur

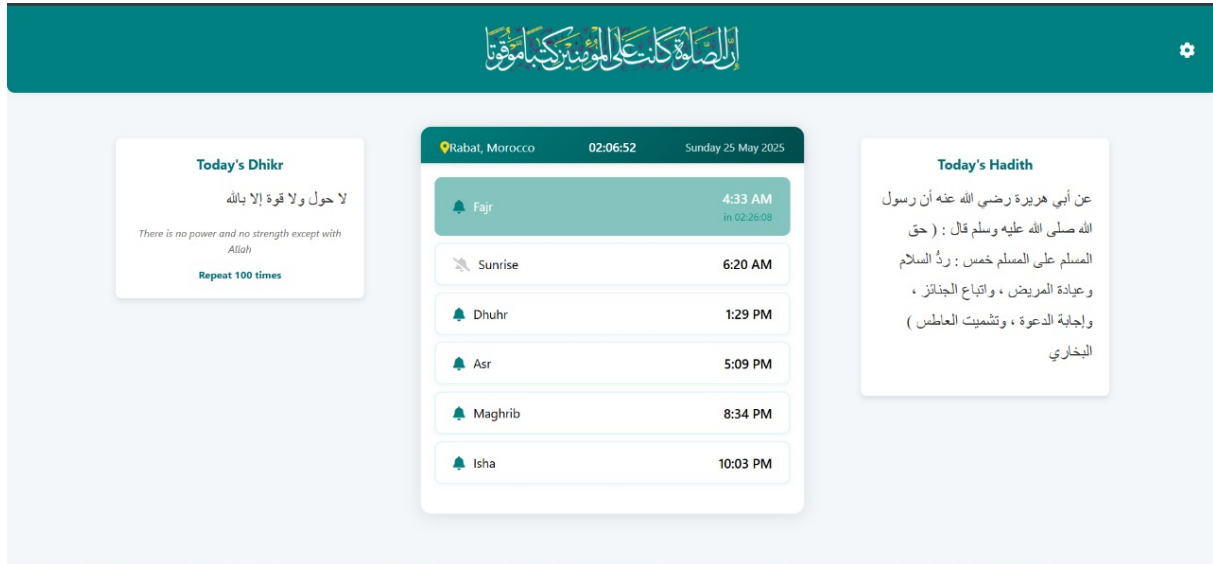


FIGURE 3.1 – Interface de notre application

3.3.1 Bannière d'En-tête

En haut de l'application, on observe le logo de notre application, qui est un verset quoranique. (En vérité la prière a été prescrite pour les croyants en des temps déterminés - Coran 4 :103). Ce verset souligne l'importance de prier à l'heure. Une icône de paramètres (roue dentée) sur la droite, suggérant la possibilité de personnaliser les réglages de l'application les différents audio des adhans.

3.3.2 Panneau Central : Horaires de Prière

C'est l'élément principal de l'application. Il affiche :

- **Localisation et Date/Heure Actuelle :**
 - Localisation : « Rabat, Morocco » (Rabat, Maroc), indiquée par une icône de localisation.
 - Heure actuelle : « 02:06:52 ».
 - Date : « Sunday 25 May 2025 » (Dimanche 25 Mai 2025).
- **Liste des Prières Quotidiennes :**
 - **Fajr** : 4 :33 AM. Cette prière est mise en évidence (couleur de fond distincte), indiquant qu'il s'agit de la prochaine prière. Un compte à rebours « in 02:26:08 » (dans 02h26m08s) est visible. Une icône de cloche est présente, suggérant une notification pour l'appel à la prière (Adhan).
 - **Sunrise (Lever du soleil/Shuruq)** : 6 :20 AM. Une icône de soleil/cloche barrée indique un repère temporel.
 - **Dhuhr** : 1 :29 PM (icône de cloche).
 - **Asr** : 5 :09 PM (icône de cloche).
 - **Maghrib** : 8 :34 PM (icône de cloche).
 - **Isha** : 10 :03 PM (icône de cloche).

3.3.3 Panneau Latéral Gauche : « Today's Dhikr » (Dhikr du Jour)

Ce module propose un rappel spirituel quotidien :

- Titre : « Today's Dhikr ».
- Traduction en anglais (implicitement traduisible en français) : « There is no power and no strength except with Allah » (Il n'y a de puissance ni de force qu'en Allah).
- Suggestion de pratique : « Repeat 100 times » (Répéter 100 fois).

3.3.4 Panneau Latéral Droit : « Today's Hadith » (Hadith du Jour)

Ce module offre une parole ou un acte du Prophète Muhammad (paix et bénédictions sur lui) :

- Titre : « Today's Hadith ».
- Texte en arabe : Le Hadith affiché concerne les droits du musulman sur son frère musulman (cinq droits : rendre le salut, rendre visite au malade, suivre les convois funéraires, répondre à l'invitation et invoquer la miséricorde d'Allah pour celui qui éternue et loue Allah).
- Source du Hadith : indiquant une authenticité élevée.

3.4 Conception de l'Interface (UI/UX)

- **Clarté et Modernité** : L'interface est épurée, avec un fond clair et des accents de couleur turquoise/vert, souvent associés à l'esthétique islamique.
- **Organisation Modulaire** : L'information est structurée en cartes distinctes (Dhikr, Horaires de prière, Hadith), facilitant la lecture.
- **Intuitivité** : La navigation semble simple, les horaires de prière étant au centre de l'attention. Les icônes sont utilisées judicieusement.
- **Lisibilité** : Bon contraste texte/fond, avec une typographie claire pour l'anglais et l'arabe.

3.5 Public Cible

L'application s'adresse aux musulmans désireux de :

- Connaître avec précision les heures de prière.
- Recevoir des rappels pour les prières.
- S'engager dans le Dhikr (souvenir d'Allah) quotidien.
- Lire et méditer sur un Hadith chaque jour.

3.6 Fonctionnalités Potentielles (Non Visibles Mais Suggérées)

Bien que non explicitement affichées, des fonctionnalités courantes pour ce type d'application pourraient inclure :

- Détection automatique de la localisation et temps selon cette localisation.
- Extraction des horaires des prières basant sur la localisation à partir d'une api.
- Personnalisation des notifications de l'appel d'adhan.

Conclusion Générale

Au terme de ce projet multidimensionnel, nous avons exploré deux paradigmes novateurs dans le domaine de l'intelligence artificielle appliquée : d'une part, un système MLOps pour l'analyse automatisée des commentaires YouTube, et d'autre part, une application d'horaires de prière développée selon l'approche révolutionnaire du Vibe Coding avec Windsurf. Cette dualité d'approches nous a permis d'appréhender l'IA tant du point de vue de l'orchestration de modèles pré-entraînés que de celui de la collaboration créative homme-machine. Le contraste entre ces deux méthodologies est particulièrement instructif : alors que le système MLOps représente une approche structurée et industrielle de l'exploitation des modèles d'IA, le Vibe Coding incarne une vision plus intuitive et conversationnelle de la conception logicielle.

Notre système d'analyse de commentaires YouTube illustre comment l'architecture microservices moderne peut orchestrer harmonieusement différentes technologies - n8n, Hugging Face, MCP, Twilio - pour transformer des données non structurées en insights actionnables. L'approche événementielle, où le système n'est activé qu'à la demande via WhatsApp, démontre l'efficacité des principes MLOps dans un contexte d'application réelle.

En parallèle, notre exploration du Vibe Coding avec Windsurf pour la création de l'application d'horaires de prière a ouvert des perspectives fascinantes sur l'avenir du développement logiciel. Cette méthodologie, où le code émerge d'un dialogue naturel avec l'IA, redéfinit profondément la relation entre le développeur et ses outils. L'interface épurée et modulaire de l'application, avec ses fonctionnalités spirituelles intégrées (Dhikr quotidien, Hadith du jour) et sa conception centrée sur l'utilisateur, démontre le potentiel du Vibe Coding pour créer des expériences utilisateur riches et contextuellement pertinentes.

Les défis rencontrés dans les deux volets du projet ont été de nature différente mais complémentaire. Pour le système MLOps, nous avons dû surmonter les limitations des API gratuites, gérer l'intégration entre services cloud et infrastructures locales, et développer des stratégies d'échantillonnage intelligentes. Pour l'application d'horaires de prière, nous avons exploré les frontières de la collaboration avec l'IA générative, appris à formuler des intentions de design sous forme conversationnelle, et développé une sensibilité aux nuances culturelles et spirituelles dans la conception d'interface.

Bibliographie

- [1] Park, J., et al. (2023). Generative Agents : Interactive Simulacra of Human Behavior. Stanford.
- [2] <https://n8n.io>
- [3] Windsurf Documentation, Claude Desktop, 2024.
- [4] Multi-agent Cloud Platform, Internal Documentation, 2025.