

(M) ont démontré une f

AutoFinder vise à répondre à ces limitations en proposant une application web intelligente capable de déceler et d'identifier les anomalies. Utilisant une machine de flexibilité élevée, elle permet de

**Thématique et objectifs**

---

**Thématique**

Le client exprime généralement ses besoins de manière floue ou partielle :

- "livraison automatique"
- "un budget de 80 000 DHS"
- "une demande récente"

**Objectifs principaux :**

- Interpréter correctement ces demandes en langage naturel

- Assurer une architecture

- **Frontend Web** (HTML / CSS / JavaScript)
- **Backend Flask** (API REST)
- **Pipeline IA contrôlé :**
  - Détection d'intention
  - Extraction de contraintes
  - Recherche sémantique (RAG)
  - Filtrage strict
  - Génération contrôlée via LLM local

## 4. Architecture globale

---

## 4.2 Backend Flask

- /chatbot : interface
- /chat : endpoint

## 5. Détection d'intention

---

- **smalltalk** : salutation
  - **car\_search** : recherche de voitures
  - **other** : hors sujet

La détection repose sur des mots-clés (nombres + unités).

## 6. RAG et base de dom

- ## 6.1 Embeddings

**Modèle :** all-MiniLM-L6-v2 (SentenceTransformers)

Chaque voiture est décrite par un texte synthétique : marque, modèle, carburant, transmission

## 6.2 Base vectorielle

  - **Base utilisée :** ChromaDB
  - **Mode :** persistant sur disque
  - Signature du dataset (mtime + taille) pour détecter les changements et reconstruire l'index

## 6.3 Recherche

Lors d'une requête utilisateur :

  1. Le texte est transformé en embedding

## Le RAG est utilisé comme mécanisme de rappel

- Carburant (diesel, essence, électrique)
- Transmission (automatique / manuelle)
- Budget maximum

- Cette extraction repose sur des expressions régulières.

## 2 Filtrage strict

Après la recherche RAG, un filtrage détermine si une interprétation n'est laissée au LLM.

### Rôle clé :

  - Le RAG apporte la pertinence sémantique
  - Le filtrage apporte la précision et la fiabilité

## Génération de réponse via le LLM

### Modèle utilisé

**Modèle local :** Optimus 7B (GGUF)

**Moteur :** llama\_cpp

**Température :** 0.4 pour réduire la créativité

## 2 Contrôle du LLM

Le LLM reçoit uniquement les filtres extraits, sans règles strictes.

Des règles strictes sont imposées via le prompt :

  - Interdiction d'inventer
  - Obligation de référencer les IDs

## Gestion du contexte conversationnel

### Problème identifié

Le smalltalk ("bonjour", "ça va") pollue la recherche.

### Solutions mises en place

Les solutions mises en place améliorent la pertinence du RAG, en :
  - Réinitialisation du contexte lors du passage à une autre question.
  - Limitation stricte de l'historique envoyé au LLM.
  - Conservation uniquement des messages utiles.Les choix améliorent : la pertinence du RAG, la précision et la fiabilité.

## 3. MVP et scénarios de test

### 3.1 Initialisation du système

Le système effectue plusieurs opérations au démarrage :

  - Vérification et réutilisation de la base Chitchat.
  - Chargement du modèle d'embeddings (all-MiniLM-L6-v2).
  - Chargement du LLM local (Optimus 7B) avec les filtres extraits.

```
nadirb@U-NadirB:~$ cd vscode_files/AutoFinder/  
nadirb@U-NadirB:~/vscode_files/AutoFinder$ source venv_au  
(venv_ autofinder) nadirb@U-NadirB:~/vscode_files/AutoFinder$ [warmup] starting heavy loads in background...  
[rag] voitures.json inchangé, réutilisation de la base Ch
```

```
[rag] collection exiting, embed  
[rag] chroma ready | ms=129.0  
[llm] loading model...  
llama_context: n_ctx_per_seq (204  
[llm] model loaded | ms=464.8  
[warmup] done | ms=594.2
```

## 10.2 Cas de test : Recherche sans résultats

Lorsque aucune voiture ne correspond aux critères de l'utilisateur et en suggérant d'ajuster ses critères.

- Requête : "je cherche une voiture avec un budget 100 dh"

```
127.0.0.1 - - [18/Jan/2026 21:02:41] "POST /chat HTTP/1.1" 200 -
[chat] request received | history len=1
```

```
[chat] calling LLM | max_
[chat] LLM done | llm_ms=
127.0.0.1 - - [18/Jan/202
```

Le système extrait correctement les contraintes

- Requête : "je cherche une voiture essence et manuelle"

**Chatbot AutoFinder**

Pose une question et trouve une voiture rapidement.

Utilis

Bot: 14901 | Maruti alto[STD | essence | manuelle | 73129  
climatisation, direction assistée 14905 | Maruti alto[STD |

vitres électriques, climatisation 14921 | Maruti alto km | 18700 DHS | Options: direction assistée, clim essence | manuelle | 73129 km | 18700 DHS | Optio

**Ecris ton message...**

```
[chat] history=[{'role': 'user', 'content': 'je cherche une voiture essence et manuelle'}]
[filters] constraints={'carburant': 'essence', 'transmission': 'manuelle'}
[rag] loading embedding model...
[rag] embedding model loaded | ms=2433.4
[rag] search | k=5 query='je cherche une voiture essence et manuelle'
[chat] history=[{'role': 'user', 'content': 'je cherche une voiture essence et manuelle'}]
```

```
- ID: 14901 | Marut
- ID: 14905 | Marut
- ID: 14910 | Marut
- ID: 14921 | Marut
- ID: 14923 | Marut
[chat] calling LLM
[chat] LLM done | l
127.0.0.1 - - [18/J
```

- Figure 5 :*

- Le LLM présente uniquement les véhicules reçus du
- Aucune hallucination détectée dans les réponses

## 11.1 Coût en ressources

L'exécution locale d'un LLM et d'un RAG

### Solutions adoptées :

- Limitation du contexte

- ## 11.2 Abandon de l'architecture LLM → RAG → LLM

L'architecture initiale prévoyait une analyse complète par LLM, une reformulation finale. Cette approche s'est révélée trop lourde pour les requêtes.

Ce choix améliore la robustesse et la prédictibilité du système.

## 13. Conclusion

---

Le projet AutoFinder démontre qu'il est possible de construire une application IA fiable et contrôlée en combinant intelligemment des modèles de langage, une base de données vectorielle et des mécanismes déterministes.

L'approche adoptée privilégie la fiabilité et la transparence plutôt que la sophistication pure, en reconnaissant les limites des ressources disponibles et en mettant en place des garde-fous appropriés pour garantir des recommandations précises et vérifiables.