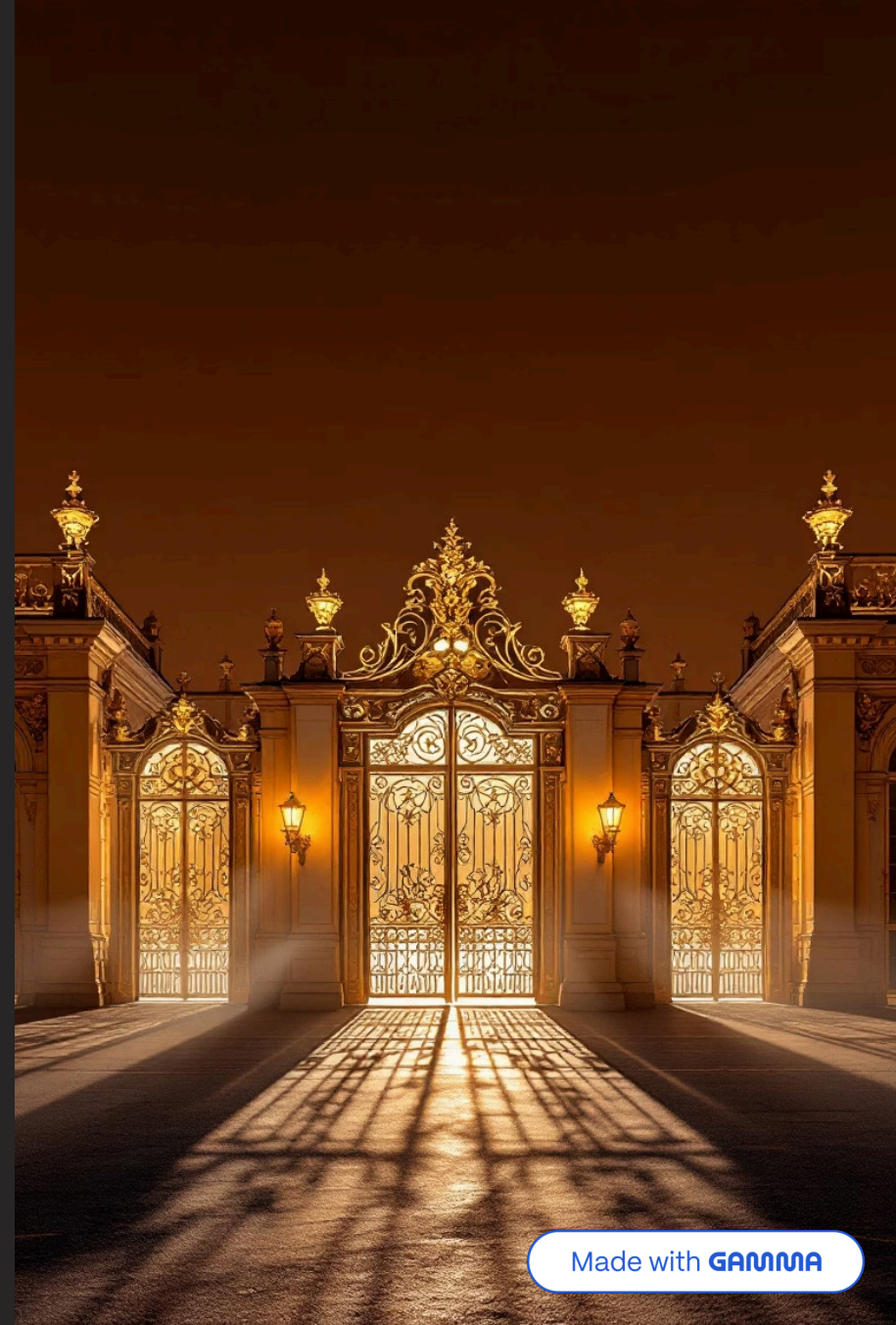


Les Clefs de Versailles

Team: Lumiere

An Agentic Solution for Personalized Visitor Itineraries



The Challenge: Navigating Versailles

1 What is there to see?

Understanding the Palace's complex structure, interconnected gardens, and hundreds of rooms requires expert knowledge that most visitors lack.

2 What are the common visitor problems?

Overwhelming information, difficulty planning for time constraints, weather conditions, crowd management, and personal interests create frustration.

3 How can AI enhance the experience?

By creating an intelligent system that plans custom itineraries through simple natural language conversation, eliminating complexity.





Foundation: Knowledge Base

Data Sources

- **JSONL Files:** 343 complex, nested documents from official website
- **PDF Documents:** Guides, maps, and supplementary documentation

Tools used

- Weather
- Transit Maps
- RAG
- Web Search

Data Enhancement

JSONL Refinement

01

Recursive Text Extraction

Traversed nested JSON structures to extract text from all content types and hierarchical levels.

03

Text Cleaning

Stripped URLs and normalized whitespace for consistent, searchable content structure.

05

Deduplication

Eliminated duplicate text segments while maintaining original document order and integrity.

02

Content Filtering

Removed fragments under 7 words while preserving important headings and contextual information.

04

Smart List Handling

Joined list items with semicolons to preserve relationships and maintain semantic context.

06

Final Formatting

Created concentrated `{"url": "text"}` structure optimized for embedding and retrieval.

Result: 4,700+ JSON entries transformed into a clean, structured, highly searchable dataset.

Data Enhancement

PDF Processing Docling

Layout Detection

Machine Learning models automatically identified document structure, headers, paragraphs, and visual elements with high accuracy.

Table Extraction

Specialized analysis model designed for accurate table extraction, preserving data relationships and formatting.

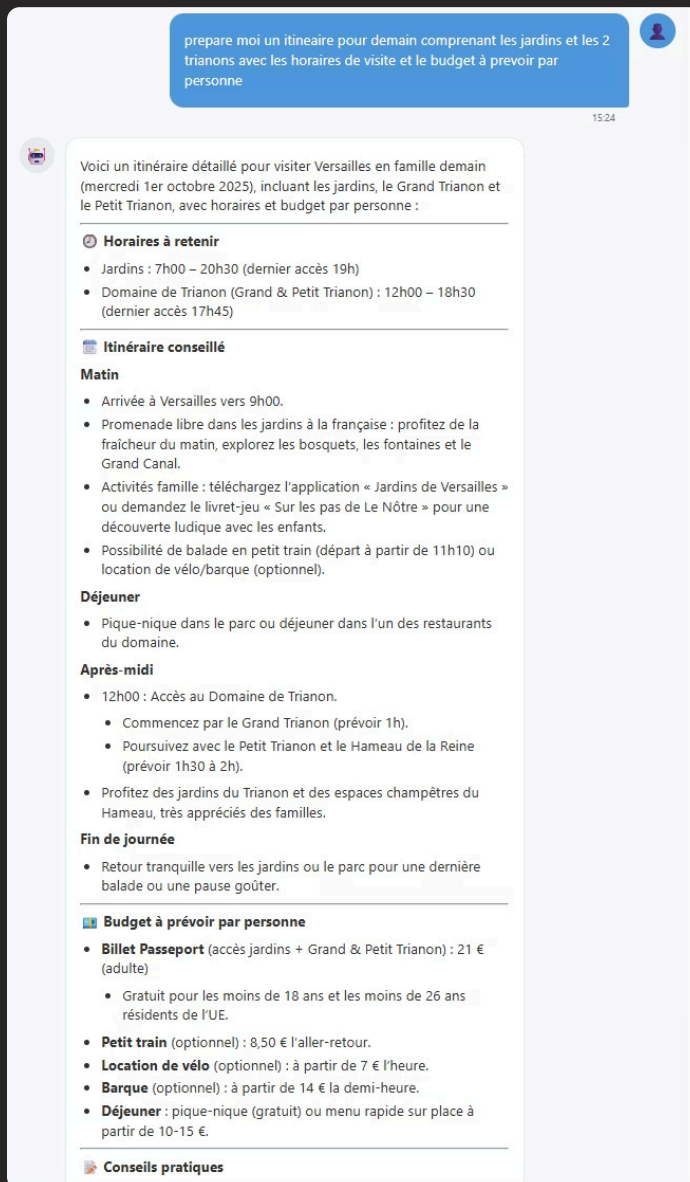
Structure Preservation

Maintained original folder hierarchy and document organization in the output for easy reference and navigation.

📄 **Result:** All hackathon PDFs converted to clean Markdown files, perfectly structured for RAG chunking and embedding processes.

Frontend

A Modern Chat Interface



The User Experience:

- **Intuitive & Responsive Design:** A clean, modern chat interface that works seamlessly on any device.
- **Real-Time Feedback:** Users see live connection status and an animated typing indicator while the agent responds.
- **Agent Transparency:** Visual badges show which tools the agent used (e.g., 🔍 Knowledge Base, 🌤️ Weather), providing insight into its reasoning.
- **Easy Onboarding:** Pre-defined query buttons help users start the conversation and discover the agent's capabilities effortlessly.
- **Persistent Conversations:** A unique conversation ID ensures users can continue their planning session later.
- **Voice Support:** Visitors can naturally speak their questions in French/English, allowing for convenient, hands-free interaction while walking through Versailles. This feature makes the experience more accessible and intuitive for all users.

Tech: React, TypeScript, Vite, Axios

Backend

Modular Agentic Architecture

This modular agentic architecture propels our backend with dynamic intelligence and unwavering accuracy. Specialized tools ensure a clean separation of concerns, keeping the model exceptionally efficient, consistently updated, and ready for any complex query.

Agent

LLM with tool calling capabilities and persistent state using python. Grounded in Versailles knowledge base for all factual queries and recommendations. 2 possible modes:

- 1** **One shot:** Plans a visit based on the user's query without asking for additional information (for the API test)
Conversational: Asks for additional information to infer user intent and preferences. Proposes a complete plan with special guided visits

Tech: OpenAI GPT-4.1 / Mistral Medium 3.1 / LangChain

Tool: RAG

RAG knowledge base for factual questions about locations, schedules, and services.

Tech: Linq-Embed-Mistral / ChromaDB / LangChain

Tool: Weather forecast

Triggers `get_weather_for_versailles_visit()` tool for weather-dependent itinerary planning and recommendations.

Tech: Open Weather API

Backend

Modular Agentic Architecture

Clean separation of concerns through specialized prompt modules enables easy updates and maintenance.

1

Tool: Travel Planner

Handles transportation queries with detailed directions, public transit options, and parking information.

Tech: Google Maps API

2

Web Search

Uses words in the user message to search the web for up to dates information

Tech: Linkup



RAG Pipeline: Knowledge Indexing

1

Raw Data Processing

343 documents from chateauversailles.fr loaded and prepared for analysis

2

Intelligent Chunking

Documents split into 2000-6000 character semantic chunks with 200-character overlap

3

Vector Embedding

Mistral AI converts each chunk into numerical vectors for similarity search

4

Persistent Storage

ChromaDB stores 5,799 searchable chunks with metadata and source URLs

343

Source Documents

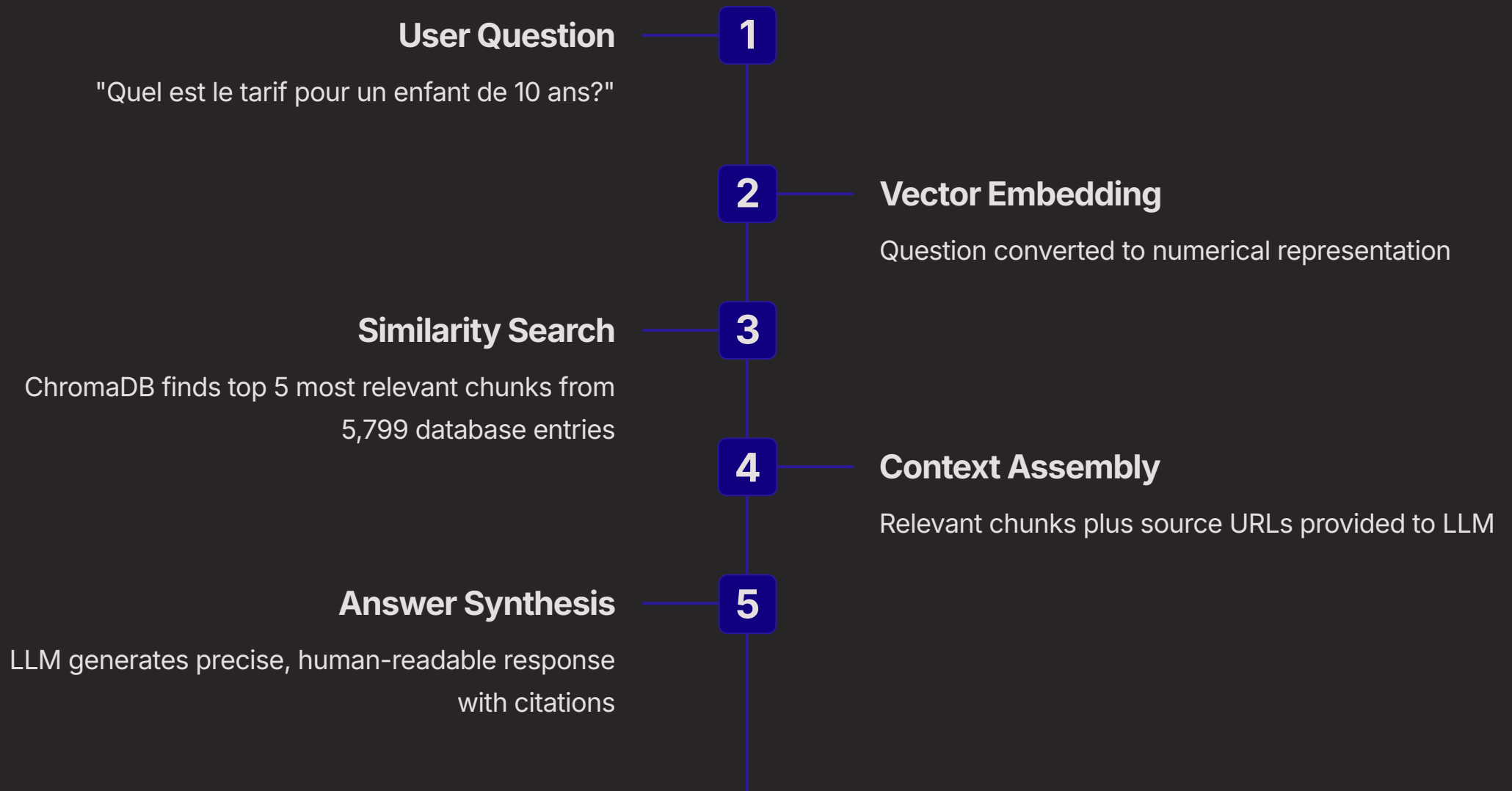
Official Versailles content

5.8K

Searchable Chunks

Semantic text segments

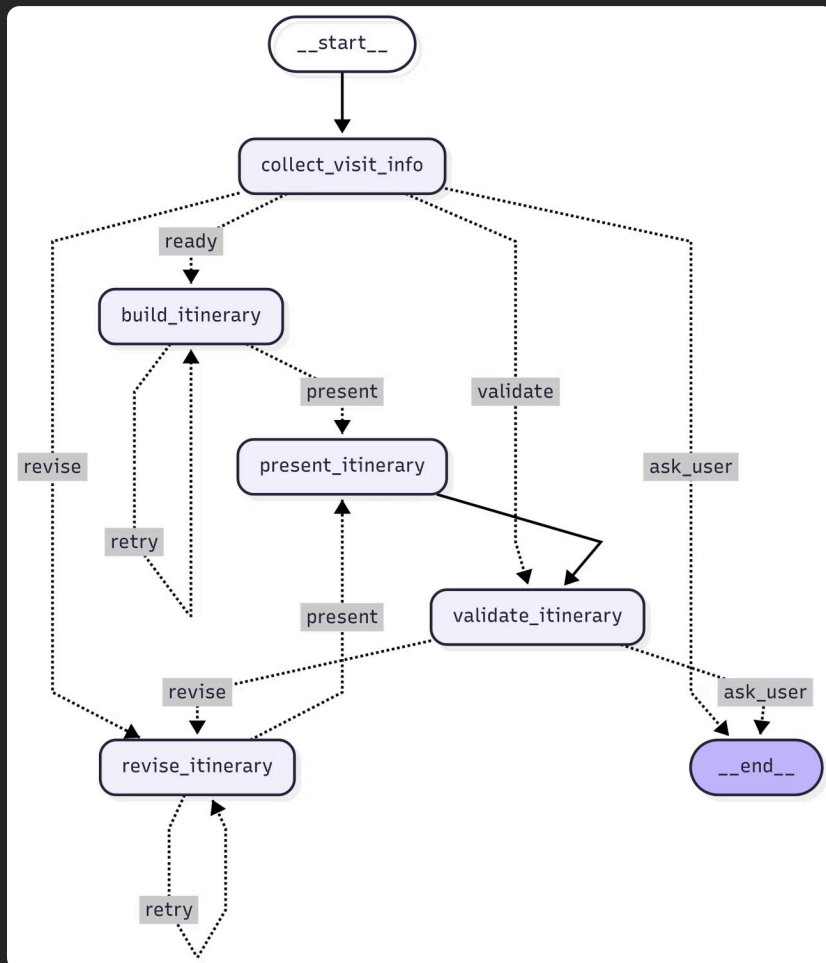
RAG Pipeline: Real-Time Query Processing



📄 **Performance:** Sub-second response times with comprehensive source attribution for every answer.

Alternative testée: LangGraph Agentic Workflow

Système multi agents préparant itinéraire personnalisé (à la journée, ½ journée, ou sur plusieurs jours) à partir d'une base d'activités structurée



Agents :

1. Agent qui **collecte le détail de la visite** (planification): composition du groupe (pers. âgées, enfants...), centres d'intérêts (conférences, parcours sculptures, jeux d'eaux, la vie du Roi...), budget maximum...
2. Agent générant une **proposition d'itinéraire** prenant en compte:
 - météo, centres d'intérêts, calendrier et horaires des activités, contrainte de budget ...
3. Agent prenant en compte les **demandes de modifications** en précisant la demande si nécessaire
4. Agent de **révision de l'itinéraire**
5. Agent pour la **mise en forme cohérente** des itinéraires pour leur présentation (ajouts de conseils et suggestions d'activités en soirée sur Versailles)

Pros:

- Embarque tests de cohérence permettant de limiter les hallucinations (cohérences des horaires et des dates, horaires compatibles entre eux, activités non doublées, centres d'intérêts adaptés et respectés, cohérence météo...). Résultats des tests remontés à l'agent durant la construction / révision du programme de visite
- Déterminisme de l'itinéraire de visite

Cons

- Nécessite base de données structurées recensant les activités proposées avec les champs détaillés (metadata: accès handicapé, activité famille, tarifs, horaires, programmation, lien intérêts....)
- Plus lourd qu'un chatbot

Thank You

Les Clefs de Versailles

Transforming visitor experiences through intelligent, personalized itinerary planning

5,799 Knowledge Chunks

Comprehensive Versailles information

Sub-second Responses

Real-time personalized planning

Source Attribution

Verified, trustworthy answers

Team Lumière Versailles • Hackathon 2025

