Master 2 – Open Data

Journal de Bord du Projet

Dashboard Météo et Agent Conversationnel

Période : Octobre 2025 – Décembre 2025

Auteurs : CARLET Malcom, GUILHEM Adrian

Table des matières

Chapitre 1

Objectif Général du Projet

L'objectif principal de ce projet était de concevoir une application web capable d'aller bien au-delà de l'affichage classique de données météorologiques. L'enjeu n'était pas simplement de montrer une température ou une probabilité de pluie, mais de proposer un véritable accompagnement contextualisé, capable de conseiller l'utilisateur dans ses décisions quotidiennes. L'ambition était donc de créer un tableau de bord moderne, complété par un agent conversationnel reposant sur un modèle de langage, afin d'offrir une expérience intégrée, intuitive et pédagogique.

Cette vision a guidé l'ensemble du développement. L'application devait fournir des prévisions météorologiques fiables en temps réel pour n'importe quelle ville du monde, interpréter ces données de manière accessible, proposer des activités adaptées aux conditions climatiques, et contextualiser les réponses grâce à des informations culturelles ou locales. L'ajout d'un agent conversationnel devait permettre à l'utilisateur d'interagir librement avec l'application, sans se limiter aux interfaces classiques.

Chapitre 2

Architecture Générale de l'Application

L'architecture adoptée repose sur une structure modulaire, pensée pour faciliter la maintenance et l'évolution du projet. Plusieurs modules indépendants ont été créés : un module dédié à la page d'accueil et à la recherche de ville, un autre consacré à la récupération et au traitement des données météorologiques, un agent intelligent utilisant LangChain et l'API Groq, ainsi qu'un moteur de recommandations basé sur les prévisions complètes. Un fichier dédié regroupe les styles et éléments graphiques afin de garantir une cohérence visuelle.

Streamlit a été choisi comme technologie d'interface en raison de sa simplicité de mise en œuvre et de sa compatibilité naturelle avec Python. Le traitement des données est assuré par divers services : Open-Meteo pour les prévisions météorologiques, Photon pour le géocodage, Nominis pour les informations culturelles, une API d'horoscope pour les prédictions quotidiennes, JokeAPI pour les messages ludiques et DuckDuckGo pour la recherche contextuelle. L'intelligence artificielle repose sur LangChain, LangGraph et le modèle LLaMA 3.1 via l'API Groq.

Chapitre 3

Déroulement du Développement

3.1 Phase de Conception et Définition des Besoins

La première étape du projet a consisté à clarifier les attentes fonctionnelles. L'idée d'un simple tableau de bord météo a rapidement évolué vers un système plus ambitieux. Il devenait essentiel de permettre à l'utilisateur de formuler des questions naturelles, d'obtenir des conseils, et de pouvoir comprendre la météo au-delà de chiffres bruts. Cette notion de contextualisation a guidé la conception du chatbot mais aussi celle du moteur de recommandations, chargé de proposer des activités ou comportements adaptés.

Le choix des technologies s'est fait progressivement. Streamlit s'est imposé pour sa rapidité de prototypage. Groq a été retenu pour ses performances exceptionnelles en temps de réponse, essentielles pour un chatbot interactif. Open-Meteo a été préférée à d'autres API commerciales afin d'éviter les limitations d'usage et garantir une grande latitude d'expérimentation.

3.2 Intégration des Données Météorologiques

L'intégration de l'API Open-Meteo a représenté un travail conséquent. La structure des données, parfois complexe et hétérogène selon les villes, imposait une organisation rigoureuse des fonctions de traitement. Il a été nécessaire de gérer simultanément des données actuelles, horaires et journalières, tout en tenant compte du fait que certaines valeurs pouvaient être absentes ou différemment formatées selon les régions du monde. Des fonctions de conversion d'unités ont été ajoutées afin d'homogénéiser les résultats.

La gestion des erreurs réseau a également été essentielle. Des arrêts temporaires de service, des lenteurs de réponse ou des pertes partielles de données ont conduit à implémenter une couche de robustesse grâce à des exceptions spécifiques et des messages explicites à destination de l'utilisateur.

3.3 Géocodage et Résolution des Problèmes de Fiabilité

Le choix initial du service Nominatim s'est avéré problématique en production. Certaines requêtes étaient bloquées, d'autres renvoyaient des réponses vides ou aléatoires. L'origine du problème provenait à la fois de limitations strictes du service et de restrictions liées au déploiement cloud.

Le remplacement de Nominatim par Photon a constitué une avancée déterminante. Photon a permis d'obtenir des résultats fiables, rapides, et surtout exempts de blocages liés au nombre de requêtes. Une difficulté supplémentaire résidait dans l'ordre inhabituel des coordonnées retournées, qu'il a fallu régulièrement inverser.

Chapitre 4

Interface Utilisateur et Visualisations

La page d'accueil a été conçue comme un point d'entrée simple permettant de rechercher une ville ou d'en sélectionner une parmi une liste. Une attention particulière a été portée à l'esthétique générale : dégradés, effets de profondeur, animation légère, et présentation claire des actions disponibles. Une carte interactive a été ajoutée afin de renforcer la dimension visuelle de l'application.

La page dédiée aux données météorologiques est structurée en plusieurs sections thématiques. On y retrouve les conditions actuelles, les prévisions à sept jours, l'analyse du vent, les précipitations, l'ensoleillement, ainsi que des représentations telles que des graphiques polaires, des courbes horaires ou des matrices thermiques. L'objectif était de présenter une grande quantité d'informations tout en conservant une navigation fluide.

Une attention particulière a été portée à la performance. Les graphiques ne sont générés qu'à l'ouverture de leur section, et les données sont stockées temporairement afin d'éviter les appels excessifs aux API. Des indicateurs visuels informent l'utilisateur du chargement des données, ce qui contribue fortement à l'expérience ergonomique du tableau de bord.

Chapitre 5

Conception et Fonctionnement de l'Agent Conversationnel

L'agent conversationnel constitue la partie la plus innovante du projet. Son fonctionnement s'appuie sur LangChain et LangGraph, deux technologies permettant de structurer les interactions entre le modèle de langage et différents outils. Le modèle LLaMA 3.1 a été configuré pour répondre de manière précise aux questions factuelles, tout en conservant un ton naturel.

Un système d'outils spécialisés a été mis en place. L'agent est ainsi capable d'appeler un outil météorologique lorsqu'une question concerne une ville ou une date particulière, un outil de recherche lorsqu'un enrichissement culturel est nécessaire, ou encore des outils annexes pour répondre à des demandes liées à l'horoscope ou aux événements du jour.

Une difficulté majeure a été de gérer les contextes implicites. L'utilisateur peut par exemple demander la météo de demain après avoir demandé celle d'aujourd'hui. Il fallait donc permettre au modèle d'interpréter correctement les références temporelles sans renvoyer une erreur ou redemander la ville à chaque étape. La conversion entre l'historique stocké dans Streamlit et le format attendu par LangChain a représenté un travail important.

Le moteur de recommandations constitue une extension naturelle de l'agent. Il analyse les prévisions complètes et génère un texte structuré comprenant des suggestions d'activités, des conseils vestimentaires, des recommandations en fonction du vent, de la pluie ou de l'ensoleillement, ainsi qu'une mise en contexte propre à la ville étudiée.

Chapitre 6

Difficultés Rencontrées et Solutions Apportées

Plusieurs difficultés ont marqué l'avancement du projet. La première concernait la fiabilité des services externes. Que ce soit pour le géocodage ou l'horoscope, certaines API présentaient des instabilités silencieuses, des retards ou des changements de structure non documentés. Pour y remédier, un système complet de gestion des erreurs a été mis en place, capable de détecter les anomalies les plus fréquentes et d'adapter le comportement de l'application.

La migration de LangChain vers sa version moderne a également représenté un défi important. Les anciennes méthodes de création d'agents ont été supprimées ou profondément modifiées. Il a fallu comprendre le nouveau fonctionnement basé sur LangGraph, revoir entièrement la création des outils, et adapter le format des messages utilisés par le modèle. Cette transition a demandé plusieurs jours de travail, mais elle a considérablement renforcé la stabilité et les capacités de l'agent.

Le traitement de l'horoscope a constitué une autre source de difficulté. L'API utilisée renvoie du texte en anglais et de manière parfois instable. Pour améliorer l'expérience, un système de traduction automatique a été intégré, tout en prévoyant un mécanisme de repli lorsque le service de traduction échouait. Cette approche garantit que l'application reste fonctionnelle en toutes circonstances.

Enfin, le volume des données météorologiques a nécessité une réflexion approfondie. Les modèles de langage sont sensibles à la taille du contexte fourni, ce qui a conduit à concevoir des méthodes d'agrégation spécialement pensées pour l'agent. Au lieu de transmettre des dizaines d'heures de données brutes, l'agent reçoit des résumés synthétiques par période de la journée, ce qui améliore à la fois la performance et la qualité des réponses.

Chapitre 7

Conclusion Générale

Ce projet a permis de développer une application complète, mêlant intelligence artificielle, visualisation avancée et intégration de données externes. Le tableau de bord propose une lecture claire et structurée des conditions météorologiques, tandis que l'agent conversationnel ajoute une dimension dynamique et interactive. L'ensemble constitue un produit cohérent, pensé pour être à la fois informatif, intuitif et personnalisable.

Le travail accompli a permis d'acquérir une expérience solide dans la manipulation d'APIs, la création d'agents intelligents, la gestion d'états dans une application web ainsi que la conception de visualisations adaptées à des données météorologiques complexes. Les perspectives d'évolution sont nombreuses : notifications, historique météorologique, comptes utilisateurs, traduction multilingue, génération de rapports automatisés, et pourquoi pas une version mobile.

Ce journal de bord retrace l'ensemble des choix, difficultés et solutions adoptées tout au long du développement, et constitue une documentation de référence pour toute évolution future de l'application.