

WeatherLive App

1. Introduction

« Bonjour. Je vais vous présenter mon projet, WeatherLive App.

C'est une application météo qui permet de consulter en temps réel la météo d'une ville et de visualiser son évolution sur les 20 derniers jours. À l'origine, je devais créer une application en terminal, mais j'ai finalement développé une interface web, plus moderne et accessible. »

2. Présentation du Projet & Conformité

« Voici ce qui a été réalisé par rapport à mon cahier des charges initial :

- **Récupération des données** : L'application utilise l'API OpenWeatherMap pour obtenir les données actuelles et historiques, comme prévu.
- **Persistance des données** : Les données sont sauvegardées dans une base SQLite locale. J'utilise aussi Pandas pour structurer les données des 20 derniers jours dans des tableaux, ce qui est très pratique pour analyser et créer des graphiques.
- **Visualisation** : Je suis allé plus loin que prévu. Au lieu de graphiques statiques, l'application affiche des courbes interactives.
- **Interface** : C'est le plus grand changement. J'ai choisi Streamlit pour créer l'interface web. C'est un framework Python qui permet de transformer du code en application web très rapidement, sans avoir à apprendre le HTML ou le JavaScript. C'était parfait pour mon projet. »

3. Démonstration & Fonctionnement

« L'application est très simple :

1. **On entre le nom d'une ville.**
2. **La page affiche aussitôt la météo du jour : température, conditions, vent, humidité...**
3. **Sur le côté, les graphiques apparaissent.**

Pour les graphiques, voici comment ça marche en arrière-plan :

Mon code récupère les données de l'API et les organise dans un tableau Pandas.

Ensuite, Streamlit prend ce tableau et, avec une seule instruction, `st.line_chart()`, il génère un graphique interactif. La combinaison Pandas + Streamlit rend la création de ce tableau de bord très simple et efficace.

Le flux est donc : API → Python → Pandas → Streamlit → Votre navigateur. »

4. Difficultés Rencontrées

« J'ai surmonté plusieurs défis :

- Appels API : Comprendre comment fonctionnent les requêtes, surtout pour les données historiques, était nouveau pour moi.
- Base de données SQL : Apprendre à créer une table et à y insérer des données a été un grand pas en avant.
- Gestion du temps : Mon calcul de la durée du jour donnait des valeurs négatives ! J'ai découvert un problème de fuseaux horaires. J'ai dû utiliser la bibliothèque pytz pour le corriger et obtenir une bonne durée entre le lever et le coucher du soleil. »

5. Succès & Améliorations Futures

« Je suis très satisfait d'avoir une application web complète et fonctionnelle. Pour la suite, j'aimerais ajouter une prévision heure par heure et peut-être une comparaison entre plusieurs villes, pour la rendre encore plus proche des applications météo professionnelles. »

6. Conclusion

« Pour résumer, ce projet m'a permis d'apprendre énormément sur le développement Python, les APIs, les bases de données et la création d'interfaces avec Streamlit. J'ai su adapter mon projet initial pour en faire une application web moderne. Merci de votre attention. »

