# Documentation du Projet :

# Visualisation des Données Météorologiques

## Introduction

Ce projet a pour objectif de créer un pipeline de données météorologiques permettant de visualiser des informations sur les températures de différentes villes françaises. L'application utilise des technologies  telles que MongoDB, Python, Dash, Pandas, et Plotly pour offrir une interface interactive et des analyses approfondies.

## Technologies Utilisées

- **Python** : Langage de programmation principal.
- **Dash** : Framework pour le développement d'applications web interactives.
- **Pandas** : Bibliothèque pour la manipulation et l'analyse des données.
- **Plotly** : Bibliothèque pour la création de graphiques interactifs.
- **MongoDB** : Base de données NoSQL utilisée pour le stockage des données.
- **JSON** : Format de fichier utilisé pour les données des villes.

## Description des Données

Pour ce projet, nous avons sélectionné **60 villes les plus peuplées de France**, qui ont été utilisées comme base pour l'analyse et la visualisation des données météorologiques. Les données météorologiques proviennent du site openweathermap.org, extraites grâce à l'API REST. Ces données incluent des informations sur les températures, les vitesses de vent, et les coordonnées géographiques des villes.

### Fichier JSON des Villes

Le fichier JSON contient des détails tels que l'identifiant de la ville, le nom de la ville, la latitude et  la longitude.. Ce fichier est intégré dans le processus de collecte et de visualisation des données.

## Structure de l'Application

L'application Dash est composée de quatre pages principales :

1. **Page d'accueil** : Une introduction au projet avec des informations sur les membres de l'équipe, l'école et les objectifs.
2. **Page 1 - Top 10 des villes les plus chaudes et les plus froides** : Affiche un tableau des 10 villes les plus chaudes et les plus froides en fonction de la température.
3. **Page 2 - Visualisation des températures par ville** : Permet aux utilisateurs de visualiser les températures des villes sélectionnées et d'afficher un graphique interactif.

4. **Page 3 - Corrélations** : Affiche des graphiques illustrant les corrélations entre la température, la vitesse du vent et la latitude.

## Flux de Données

Le projet utilise **MongoDB** pour stocker et gérer les données météorologiques. Les étapes suivantes résument le flux de données :

1. **Collecte des données** : Les données sont collectées à partir d'une source externe (comme OpenWeatherMap) et stockées dans un fichier JSON.
2. **Insertion dans MongoDB** : Les données JSON sont importées dans une base de données MongoDB pour un stockage structuré.
3. **Analyse avec Pandas** : Les données sont traitées avec Pandas pour effectuer des calculs et préparer les données nécessaires à la visualisation.
4. **Visualisation avec Dash et Plotly** : L'application Dash utilise les bibliothèques Plotly et Dash pour créer des graphiques interactifs et des tableaux.

## Conclusion

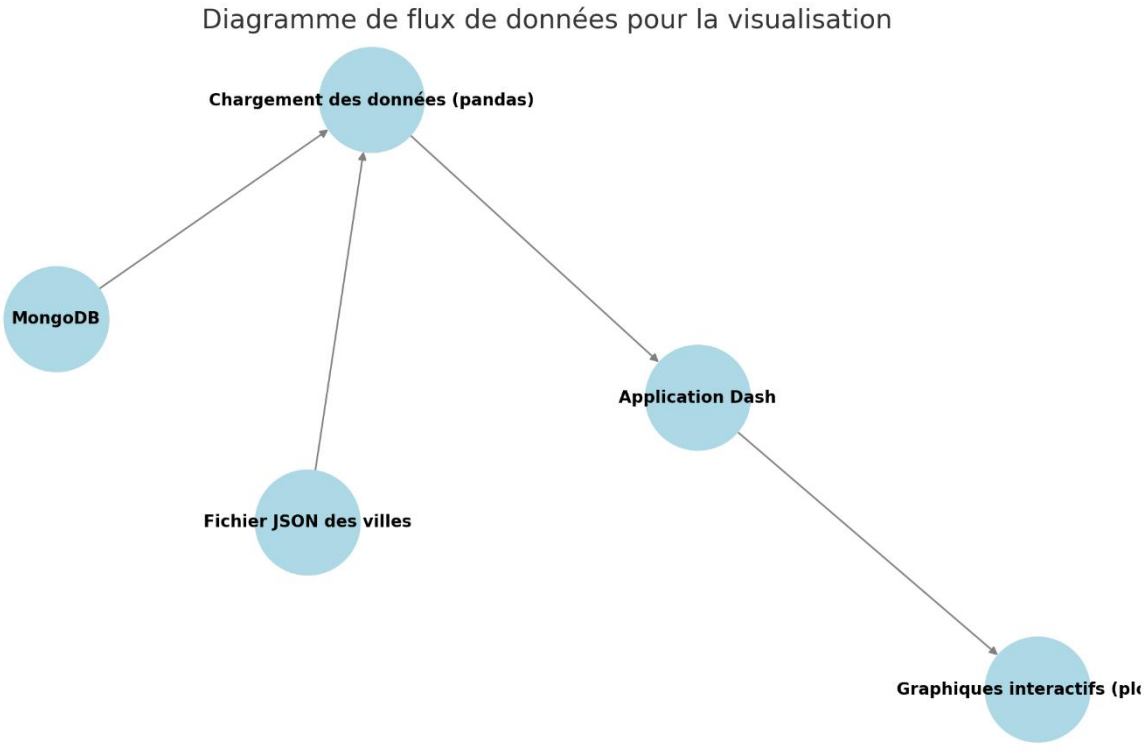
Ce projet offre une vue interactive des conditions météorologiques des villes les plus peuplées de France, permettant aux utilisateurs d'explorer des visualisations des températures, des tendances et des corrélations importantes pour l'analyse des données météorologiques.

## Description du projet

1. **Source de données** :
    - **MongoDB** : Base de données où sont stockées les informations météorologiques.
    - **Fichier JSON des villes** : Contient les données des villes nécessaires à l'affichage des graphiques et à la sélection par l'utilisateur.
2. **Traitement des données** :
    - **Chargement et prétraitement** : Utilisation de `pandas` pour charger et préparer les données issues de MongoDB et du fichier JSON.
3. **Application Dash** :
    - **Pages d'analyse** (Page 1, Page 2, Page 3) : Ces pages utilisent les données traitées pour afficher des graphiques interactifs et des tableaux.
4. **Visualisation des données** :
    - **Graphiques interactifs** : Utilisation de `plotly` pour créer des graphiques basés sur les données de température, la corrélation entre la température et la vitesse du vent, et la température par ville.

## Diagramme

1. **Fichier JSON des villes ↔ Chargement des données (pandas) ↔ Application Dash**
2. **MongoDB ↔ Chargement des données (pandas) ↔ Application Dash**
3. **Application Dash ↔ Graphiques interactifs (plotly)**

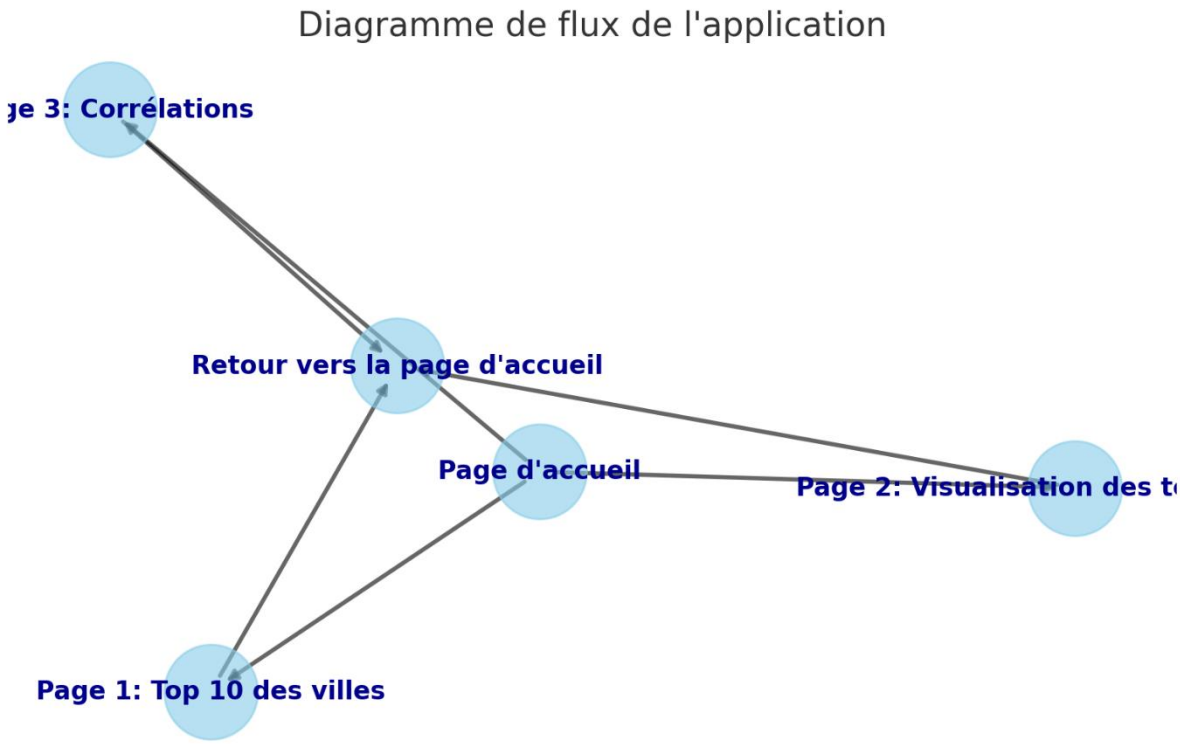


## Structure du projet :

1. **Page d'accueil** :
    - Contient des informations sur le projet, les membres de l'équipe, et un menu de navigation vers les autres pages.
    - **Liens** :
        - Vers **Page 1** : Top 10 des villes selon la température du jour.
        - Vers **Page 2** : Visualisation des températures par ville.
        - Vers **Page 3** : Corrélations (température, vent et latitude).
2. **Page 1** : **Top 10 des villes selon la température du jour**
    - Affiche les 10 villes les plus chaudes et les plus froides.
    - **Retour vers la page d'accueil** : Permet de revenir à la page d'accueil.
3. **Page 2** : **Visualisation des températures par ville**
    - Permet de visualiser les températures par ville sous forme de graphique.
    - Options pour choisir entre afficher toutes les données ou une sélection spécifique de villes.
    - **Retour vers la page d'accueil** : Permet de revenir à la page d'accueil.
4. **Page 3** : **Corrélations (température, vent et latitude)**
    - Affiche les graphiques des corrélations entre la température et la vitesse du vent, ainsi que la température et la latitude.
    - **Retour vers la page d'accueil** : Permet de revenir à la page d'accueil.

## Structure du schéma (par étapes) :

- **Accueil** -> **Page 1** / **Page 2** / **Page 3** : L'utilisateur peut choisir la page à afficher.
- Chaque page d'analyse peut rediriger vers l'accueil.



Voici un diagramme de flux illustrant la structure du projet. Il montre comment les pages sont connectées entre elles, avec des flèches indiquant la direction de navigation entre la page d'accueil et les différentes pages d'analyse :

1. **Page d'accueil** :
    - L'utilisateur peut naviguer vers **Page 1**, **Page 2**, ou **Page 3** via les liens disponibles.
2. **Page 1**, **Page 2**, et **Page 3** :
    - Chaque page d'analyse permet à l'utilisateur de revenir à la **Page d'accueil**.