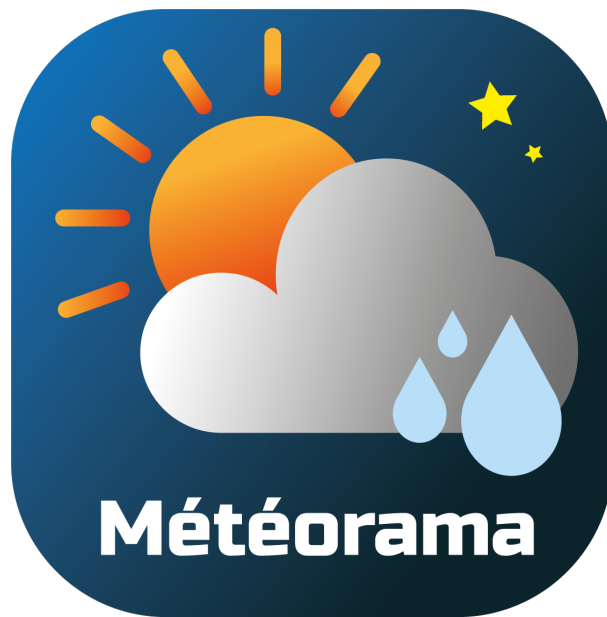


Météorama

Rapport de Soutenance 1



Auteurs :

Hugo MORO - Roy SABRA

Aubin THOMÉ - Sacha VAUDEY

Date :

Février 2025

Table des matières

| | | |
|----------|-----------------------------------------------|-----------|
| 1 | Rappel du projet | 4 |
| 1.1 | Description générale | 4 |
| 1.2 | Le groupe Météorama | 4 |
| 1.2.1 | Hugo MORO | 4 |
| 1.2.2 | Roy SABRA | 5 |
| 1.2.3 | Aubin THOMÉ | 5 |
| 1.2.4 | Sacha VAUDEY | 5 |
| 1.3 | Récupération des données | 6 |
| 1.4 | Traitement des données | 6 |
| 1.5 | Modèle de prévision | 6 |
| 1.6 | Interface et ergonomie | 6 |
| 1.7 | Répartition des tâches | 7 |
| 2 | Réalisations | 8 |
| 2.1 | État d'avancement | 8 |
| 2.2 | Réalisation techniques | 9 |
| 2.2.1 | Fonctionnement général | 9 |
| 2.2.2 | Récupération des données | 9 |
| 2.2.3 | Traitement et nettoyage des données | 10 |
| 2.2.4 | Modèles de prévision | 12 |
| 2.2.5 | Interface graphique | 13 |
| 2.2.6 | Site internet | 15 |
| 3 | Analyse | 19 |
| 3.1 | Points positifs | 19 |
| 3.2 | Points de complexité | 19 |
| 4 | Conclusion | 20 |

Table des figures

| | | |
|---|-----------------------------------------------------------------|----|
| 1 | Exemple de JSON après nettoyage | 10 |
| 2 | Interface d'accueil de l'application | 14 |
| 3 | Interface de présentation de données de l'application | 15 |
| 4 | Page d'accueil du site internet | 16 |
| 5 | Page d'équipe du site internet | 16 |
| 6 | Page de la chronologie du projet du site internet | 17 |
| 7 | Page de ressources du site internet | 17 |

Liste des tableaux

| | | |
|---|--------------------------------------------------------------------|----|
| 1 | Répartition des tâches et avancement | 7 |
| 2 | Avancement des tâches pour la première soutenance | 8 |
| 3 | écart moyen entre les valeurs attendus et les prédictions. | 13 |

Introduction

Ce rapport de soutenance a pour objectif de présenter l'état d'avancement de notre projet Météorama depuis le dépôt et la validation de son cahier des charges. Ce projet s'inscrit dans une démarche structurée, où chaque étape du développement a été planifiée afin d'assurer une progression cohérente et efficace.

Dans un premier temps, nous rappellerons brièvement les objectifs du projet afin d'en situer le contexte et l'importance. Ensuite, nous détaillerons les avancées réalisées jusqu'à présent, en mettant en avant les différentes phases du développement, les choix techniques effectués, ainsi que les défis rencontrés et les solutions mises en place.

Enfin, ce rapport se conclura par une projection des étapes à venir, en soulignant les tâches prioritaires et les ajustements nécessaires pour atteindre les objectifs fixés d'ici la prochaine soutenance. L'ensemble de ces éléments permettra d'évaluer notre progression et d'anticiper les ajustements requis pour la suite du projet.

1 Rappel du projet

1.1 Description générale

Nous avons choisi, pour ce projet du semestre 4, de développer une application météorologique complète, dans laquelle nous concevons nous-mêmes les algorithmes de prévision. L'objectif de ce projet, en accord avec le sujet, est de construire une application basée sur de grands principes algorithmiques.

Cela implique le traitement des données météorologiques sur lesquelles nous nous appuyons pour les prévisions, ainsi que la conception d'un modèle de prévision reposant sur des algorithmes statistiques et des techniques d'interpolation des données. Ce modèle s'appuie sur les relevés météorologiques effectués au cours des jours précédant la prévision souhaitée.

1.2 Le groupe Météorama

1.2.1 Hugo MORO

J'ai découvert la programmation en classe de troisième, au collège, en développant un système d'arrosage automatique contrôlable par téléphone grâce à Arduino. Ce projet, conçu pour les jardins de l'établissement, m'a permis d'allier ma passion pour la technologie et celle de la botanique en apportant une solution de simplification à l'entretien des espaces verts.

Pendant le confinement, j'ai approfondi mes compétences sur Arduino en explorant de nouvelles fonctionnalités et en testant divers capteurs. Par la suite, je me suis tourné vers Python pour élargir mes possibilités tout en poursuivant mes expérimentations.

La météo ayant un impact majeur sur de nombreux domaines, tel que la botanique, ce projet de prévision climatique de groupe m'intéresse particulièrement. Un système capable de prévoir les variations météorologiques pourrait être un outil précieux pour de nombreux utilisateurs. Contribuer à son développement me motive, car il allie technologie et utilité concrète au quotidien.

1.2.2 Roy SABRA

Arrivé en France en 2014, j'ai commencé à apprendre le français et à me familiariser avec une culture à la fois riche et stimulante. Mon parcours est assez particulier, mais j'ai toujours nourri une passion pour le monde de la technologie, de la robotique et de l'informatique.

En ce qui concerne ce projet, à vrai dire, je me suis toujours demandé comment fonctionne réellement un système météorologique, et la manière dont les prévisions s'avèrent souvent correctes et précises. Contrairement à mes camarades, je n'ai jamais eu d'expérience dans ce domaine, mais je suis tout à fait prêt à découvrir le développement, l'analyse et le fonctionnement d'un système météorologique grâce à ce projet.

1.2.3 Aubin THOMÉ

Durant un atelier sur Raspberry Pi dans le Fablab de mon village, j'ai découvert la domotique. J'ai alors très vite déployé mon propre réseau de domotique à l'aide de la plateforme Jeedom. Ce réseau domotique implémente des scénarios qui en fonction de données vont exercer différentes actions sur les éléments connectés. En utilisant les données publiques de météo et les données des capteurs du réseau local, j'ai pu implémenter différentes actions sur les objets de la maison. C'est pourquoi j'ai tout de suite accroché à une proposition de sujet sur la prévision météorologique.

Un autre facteur qui m'a motivé est la réalisation d'un stage d'une semaine à l'INRAE (Institut national de recherche pour l'agriculture, l'alimentation et l'environnement) sur les rives du lac Léman, où les prévisions météo sont un élément essentiel au bon déroulement de leurs recherches en milieu lacustre.

1.2.4 Sacha VAUDEY

Passionné par l'informatique, je me suis également toujours beaucoup intéressé au monde de la météorologie. Depuis longtemps, j'étudie le fonctionnement général des modèles de prévision météorologique mis à notre disposition. J'ai également exploré les nombreuses applications de la météo dans divers domaines.

Une expérience marquante a été ma visite du centre de prévision météorologique de l'armée de l'air sur la base aérienne 942, située au Mont Verdun. Lors de cette visite, j'ai eu l'opportunité de découvrir les différents métiers liés à la météorologie, d'explorer les nombreux modèles de prévision existants, et de comprendre le fonctionnement global d'une prévision météorologique.

1.3 Récupération des données

Les données que nous utilisons sont récupérées grâce à **OpenWeatherMap**, un service mondial, en grande partie gratuit, de collecte de données météorologiques. Ce service nous permet d'accéder à un très grand nombre de données en temps réel, ainsi qu'à des relevés très précis effectués au cours des jours passés.

1.4 Traitement des données

Nous souhaitons ensuite traiter ces données afin de corriger d'éventuels problèmes qu'elles pourraient contenir. Cela inclut la correction des incohérences présentes, le nettoyage du dataset afin de ne conserver que les données utiles, ainsi que le ré-échantillonnage des données pour qu'elles s'adaptent parfaitement à notre modèle de prévision.

1.5 Modèle de prévision

L'objectif central de ce projet est de concevoir un modèle de prévision à part entière. Étant donné que l'utilisation d'une solution de Deep Learning pour ce modèle serait trop complexe, nous avons choisi de nous orienter vers un modèle statistique.

L'objectif est d'exploiter divers algorithmes de prévision statistique afin de générer, à partir des données récupérées et traitées précédemment, une prévision à court terme pour une localisation précise.

1.6 Interface et ergonomie

Enfin, notre objectif est de présenter ce projet à travers une interface moderne et ergonomique, permettant aux utilisateurs d'accéder facilement à toutes les informations utiles en fonction de la localisation indiquée.

Grâce à *Tauri*, nous serons en mesure de développer une interface intuitive et fluide, facilitant la navigation et l'affichage des informations essentielles propres à ce type d'application.

1.7 Répartition des tâches

Pour rappel, nous avons réalisé une répartition des tâches ainsi qu'une prévision d'accomplissement du projet lors de la remise du cahier des charges, que nous rappelons ici.

| Tâche | Hugo | Roy | Aubin | Sacha | S1 | S2 | S3 |
|---------------------------|------|-----|-------|-------|-----|------|------|
| Récupération des données | X | | | | 90% | 100% | 100% |
| Traitement des données | | X | | X | 30% | 60% | 100% |
| Algorithmes de prévisions | X | | X | | 25% | 50% | 100% |
| Interface graphique | | | X | | 0% | 20% | 100% |
| Site Web | | X | | X | 30% | 60% | 100% |

TABLE 1 – Répartition des tâches et avancement

Nous n'avons apporté aucune modification à la répartition des tâches car cette organisation nous a permis d'avancer efficacement sur le développement du projet.

2 Réalisations

Cette partie vise à aborder plus en détail l'ensemble des réalisations qui ont pu être accomplies pour cette première soutenance. Nous commencerons donc par faire un bref résumé de l'avancement, avant d'entrer plus en détail sur les différentes réalisations techniques.

2.1 État d'avancement

| Tâche | Première Soutenance |
|---------------------------|---------------------|
| Récupération des données | 90% |
| Traitement des données | 80% |
| Algorithmes de prévisions | 25% |
| Interface graphique | 60% |
| Site Web | 60% |

TABLE 2 – Avancement des tâches pour la première soutenance

Ce tableau retrace le pourcentage d'avancement actuel de notre projet en fonction des prévisions établies dans le cahier des charges. Comme nous pouvons le constater, les prévisions faites lors de la rédaction du cahier des charges étaient largement inférieures à ce que nous avons pu réaliser pour cette première soutenance.

En effet, la récupération et le traitement des données ont avancé plus rapidement que prévu, tout comme l'interface graphique et le site internet. En ce qui concerne l'algorithme de prévision, nous sommes en phase avec les estimations que nous avons établies au moment de la rédaction du cahier des charges. Nous détaillons dans la suite du rapport les raisons de ces avancées.

2.2 Réalisation techniques

Nous allons maintenant détailler l'ensemble des fonctionnalités et algorithmes implémentés dans notre projet.

2.2.1 Fonctionnement général

Avant de détailler l'ensemble des implémentations que nous avons réalisées jusqu'à présent pour ce projet, nous allons aborder un aspect technique essentiel que nous avons choisi pour son développement.

En effet, ce projet nécessite un grand nombre d'interactions **In/Out**, pouvant être coûteuses en termes de performances. Pour y remédier, nous avons choisi d'utiliser **Tokio**, qui permet de transformer des fonctions en asynchrones. L'objectif est d'éviter les potentielles fonctions **I/O** bloquantes au sein de notre programme, qui pourraient entraîner de lourdes latences. Grâce à une approche asynchrone, nous pouvons exécuter certaines séquences de fonctions sans que l'ensemble du programme soit intégralement bloqué.

Cependant, cette approche entraîne d'autres problématiques, notamment en matière de synchronisation des actions. Étant donné que certaines fonctions principales – dont le retour est crucial pour notre programme – sont asynchrones, nous avons dû nous assurer que leur exécution était bien finalisée avant de lancer d'autres processus dépendant de leurs résultats.

2.2.2 Récupération des données

La première étape que nous avons dû réaliser pour ce projet est de récupérer les données météorologiques utiles pour notre programme. L'utilisateur étant invité à entrer une ville dans le moteur de recherche de l'interface, il faut que le programme soit capable de récupérer en temps réel les données météorologiques de cette ville.

Pour ce faire, nous avons utilisé le service OpenWeatherMap, qui nous permet d'accéder à ce type de données. Nous avons donc mis en place un système de requêtes permettant de récupérer les données météorologiques entre deux dates, avec un écart maximum de quatre jours.

Pour implémenter cette fonction, nous avons utilisé la librairie `reqwest`, qui permet d'effectuer des requêtes en ligne. Le retour de chaque requête est ensuite enregistré dans un fichier JSON afin que nous puissions utiliser ces données dans la suite du code.

2.2.3 Traitement et nettoyage des données

Traitement des données

Le traitement des données se réalise en plusieurs étapes. L'objectif de départ est de restreindre les données à traiter à l'essentiel. De ce fait, nous parcourons le JSON produit de la requête à OpenWeatherMap pour en extraire que les données météorologique utiles. Ainsi, nous obtenons un JSON beaucoup plus simple à traiter comme il est possible de le voir sur l'image suivante

```
sunrise: 1740292237
sunset: 1740331140
▼ weatherData:
  ▼ wd:
    ▼ 0:
      clouds: 0
      day: 19
      dt: 1739970000
      feels_like: 285.07000732421875
      hour: 13
      humidity: 69
      month: 2
      pressure: 1020
      real: true
      temp: 285.92999267578125
      temp_max: 287.2900085449219
      temp_min: 285.42999267578125
      wind_deg: 180
      wind_speed: 10.289999961853027
      year: 2025
```

FIGURE 1 – Exemple de JSON après nettoyage

De plus, nous avons fait le choix d'utiliser une structure personnalisée pour pouvoir traiter rapidement les données. Cette structure est accompagnée également d'une implémentation sous forme de liste afin que nous puissions facilement les ordonner et itérer sur cette structure. Elle nous permet ainsi via le chargement de JSON de créer une structure possédant pour chaque type de donnée un attribut associée. De ce fait, il nous est alors possible de traiter rapidement et facilement les données sans avoir à charger et à sauvegarder un nouveau fichier JSON à chaque fois ce qui aurait représenter une action beaucoup trop coûteuse pour notre programme.

Nettoyage des données

Après avoir traiter les données et les avoir transformer dans un structure facilement utilisable, nous allons pouvoir commencer le nettoyage de ces données.

Ce qu'il faut savoir c'est que OpenWeatherMap renvoie des données qui sont déjà très propres. De ce fait, nous avons implémenter des fonctions de nettoyage de données sur un dataset possédant des données de bonne qualité. Ainsi, elles n'agissent que très peu sur les données (à part sur des exemples précis qui seront détaillés ci-après).

Remplacement des valeurs nulles Le premier algorithme à implémenter est un algorithme permettant de prédire les données manquante. Il peut arriver que dans le dataset fournit par OpenWeatherMap, certaines données peuvent être manquante. Pour palier à ce problème et ainsi éviter des erreurs dans la suite de l'exécution de notre programme, nous allons parcourir l'ensemble des données que nous avons récupéré à la recherche de données manquantes. Si une donnée est manquante, elle sera alors remplacée par la moyenne de ses voisine directes. Cela permet de bloquer efficacement les trou tout en conservant une certaine cohérence sur les données.

Suppression des valeur aberrante Le deuxième algorithme que nous avons implémenter est un algorithme de suppression des valeur aberrantes. L'objectif de cette algorithme est de détecter les valeurs qui n'auraient aucune cohérence par rapport à ses voisines.

Déjà, nous avons appliqué cet algorithme sur des valeurs où il était pertinent de le faire, c'est à dire des valeurs qui ne fluctue que très peu d'une heure à une autre. Par exemple, appliquer cet algorithme sur les températures ou la pression atmosphérique est une bonne solution, mais l'appliquer sur la vitesse de vent n'aurait aucun sens car c'est une donnée qui peut varier grandement dans un intervalle de temps assez réduit.

L'objectif est de parcourir toutes les valeurs de notre struct. Si une valeur est considérée comme aberrante par notre algorithme c'est à dire une donnée qui aurait un écart trop élevé par rapport à ses voisines (écart calculé dynamiquement via une moyenne des écarts entre les valeurs voisines), alors elle est supprimée puis remplacée par la moyenne de ces deux voisines. Au bout d'un passage, les valeur de pression et de température sont donc lissée et plus cohérente.

Normalisation des données Le dernier algorithme que nous avons implémenté est une normalisation des données. L'intérêt de la normalisation est de faire en sorte que les données étudiées se retrouvent sur un intervalle d'étude cohérent pour la suite de notre programme. Pour ce faire, nous allons parcourir l'ensemble de nos données pour en déterminer la moyenne et ainsi un écart type. Toutes les valeurs étant trop éloignées de cet écart type sont alors normalisées pour être plus proche.

Après cette normalisation, nous sommes désormais en mesure de lancer notre algorithme de prévision qui se basera sur des données nettoyées.

Autres implémentations Il nous sera possible par la suite, après avoir complété le système de prédiction, d'implémenter des fonction de traitement de données supplémentaires afin d'obtenir de meilleurs résultats et pouvoir obtenir des niveaux d'erreur plus faibles sur nos prédictions.

2.2.4 Modèles de prévision

L'objectif principal de notre projet est de concevoir un modèle de prévision météorologique basé sur des méthodes statistiques. Étant donné que l'utilisation d'un modèle de Deep Learning est complexe et nécessite une quantité importante de données historiques, nous avons choisi une approche plus accessible et explicable, en nous basant sur des algorithmes de prévision statistique et des techniques d'interpolation.

Nous avons identifié plusieurs méthodes de prédiction qui convienne pour nos données :

Moyenne mobile pondérée : permet de lisser les variations des relevés et d'offrir plus de poids aux observations récentes. ;

Régression linéaire : établie des tendances générales et permet une prédiction qui est plus continue par rapport aux dernières données. ;

Interpolation polynomiale : permet une précision accrue à condition de faire le bon choix pour le polynôme et sa fonction d'interpolation.

À la date de rédaction de ce rapport, seule la fonction de moyenne mobile pondérée a été implémentée. La conception de la structure de gestion des données et des méthodes associées a été essentielle pour cette implémentation. Avant de commencer le développement des fonctions de prédiction, il a été nécessaire de prendre le temps de réfléchir à la meilleure manière d'aborder le problème et d'anticiper les différentes étapes. Bien que cette phase ait pris du temps, elle représente un investissement précieux pour l'ajout futur de nouvelles méthodes de prévision.

La méthode implémentée utilise les données des trois derniers jours aux heures que l'on souhaite prédire, en appliquant des coefficients croissants en fonction du temps. Cependant, il est encore possible d'améliorer cette méthode. Par exemple, on pourrait prendre en compte les heures entourant l'heure h que l'on souhaite prédire, en intégrant les valeurs des heures $h-1$ et $h+1$ des jours précédents, avec des coefficients plus faibles. Une autre piste d'amélioration consisterait à affiner les coefficients utilisés pour mieux refléter l'importance relative de chaque donnée dans la prédiction. Cependant même avec des améliorations la pression sera limitée.

Voici un tableau présentant les écart moyen entre les valeur réelles et les valeurs prédites sur la semaine du 14 au 21 février 2025, pour des prévisions à 24, 48 et 72 heures.

| Prédiction | Humidité | Pression | Température | Temp. max | Temp. min | Temp. ressentie |
|------------|----------|----------|-------------|-----------|-----------|-----------------|
| 24 h | 4.98 | 1.10 | 3.59 | 1.00 | 2.57 | 3.51 |
| 48 h | 5.22 | 3.66 | 4.14 | 0.99 | 5.41 | 5.58 |
| 72 h | 5.35 | 4.60 | 5.89 | 1.00 | 6.13 | 6.47 |

TABLE 3 – écart moyen entre les valeurs attendus et les prédictions.

2.2.5 Interface graphique

Pour l’interface graphique nous sommes passés par le framework logiciel open source *Tauri*. Nous avons fait ce choix pour plusieurs raisons :

- **Consommation mémoire réduite** : Contrairement à *Electron*, qui embarque *Chromium*, *Tauri* utilise le moteur de rendu natif du système, ce qui réduit considérablement l’empreinte mémoire et la taille des binaires.
- **Multiplateforme** : Fonctionne sur *Windows*, *macOS* et *Linux*.
- **Simplicité de prise en main** : le frontend utilise des technologies web (*HTML*, *CSS*, *JavaScript*).

Architecture du projet

Contrairement à ce que l’on pourrait penser, l’interface graphique est centrale dans notre projet. Déjà, car elle permet aux utilisateurs d’interagir avec l’environnement que nous avons développé, mais aussi parce que *Tauri* adopte une architecture de fichiers légèrement différente par rapport à un projet *Cargo* classique.

En effet, les projets *Tauri* possèdent deux dossiers sources. Le premier contient l’ensemble du Front-End, c’est-à-dire le code *HTML*, *CSS* et *JavaScript* permettant la construction de l’interface et le fonctionnement de ses différentes composantes (barre de recherche, affichage dynamique selon les données, etc...). Le deuxième contient l’ensemble du Back-End, c’est-à-dire l’ensemble des fichiers *Rust* nécessaires au fonctionnement logique du projet. C’est ce dossier source qui contient les fonctions nécessaires à la récupération et au traitement des données météorologiques en temps réel, mais aussi l’ensemble des algorithmes du modèle de prévisions.

L’objectif final est donc d’utiliser le code JavaScript du Front-End pour exécuter les fonctions du Back-End nécessaires afin de fournir à l’utilisateur les résultats voulus.

Il nous a donc été particulièrement important d’implémenter cette architecture en premier lieu au sein du projet afin que nous puissions, dès le départ, partir sur une organisation que nous allons garder durant tout le développement de celui-ci.

Organisation des pages de l'UI

L'UI reprend les thèmes du site pour garder une cohérence graphique sur l'ensemble du projet.

Au moment de l'ouverture de l'application, l'utilisateur arrive sur une page d'accueil avec en son centre une barre de recherche dans laquelle il peut saisir le nom d'une ville française. Une liste déroulante s'ouvre et il peut alors sélectionner sa ville. Dans un premier temps, nous avons fait le choix de limiter le choix à la France de manière à simplifier le développement.

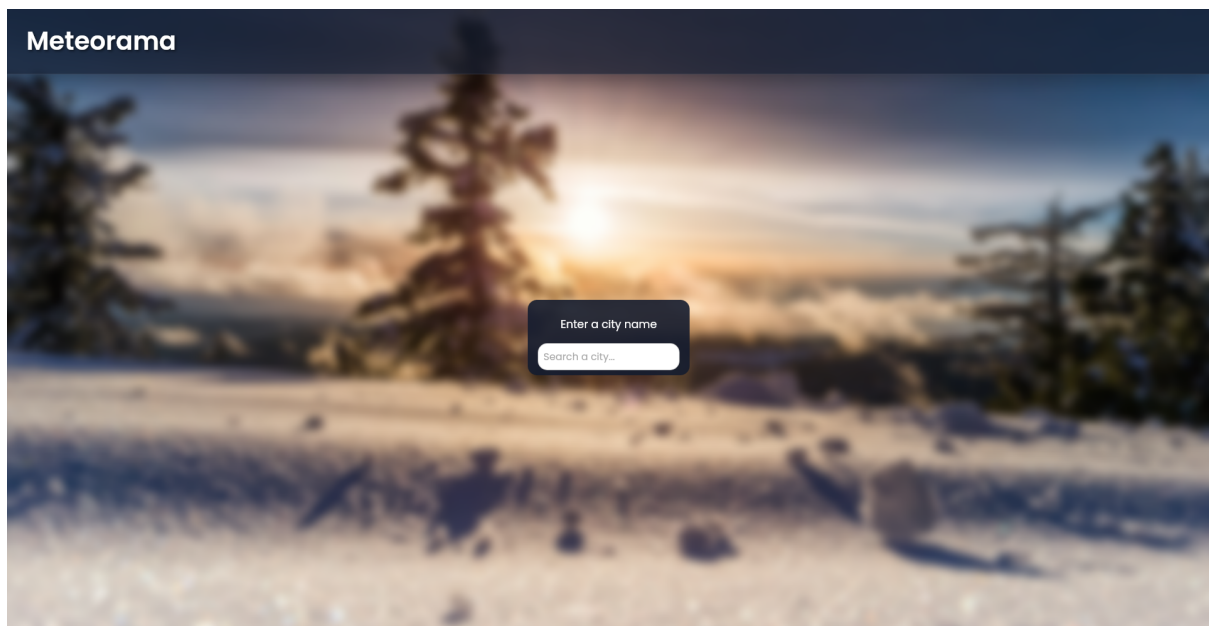


FIGURE 2 – Interface d'accueil de l'application

Une fois la ville sélectionnée, une nouvelle page se charge avec les données météo actuelles de la ville. Celles-ci sont représentées en groupe dans des boxes. La disposition actuelle n'est pas fixe et évoluera sûrement dans les prochaines semaines.

A terme, il sera proposé de naviguer vers une nouvelle fenêtre pour accéder à un onglet où il sera possible de lancer une prévision météo et d'en consulter le résultat.

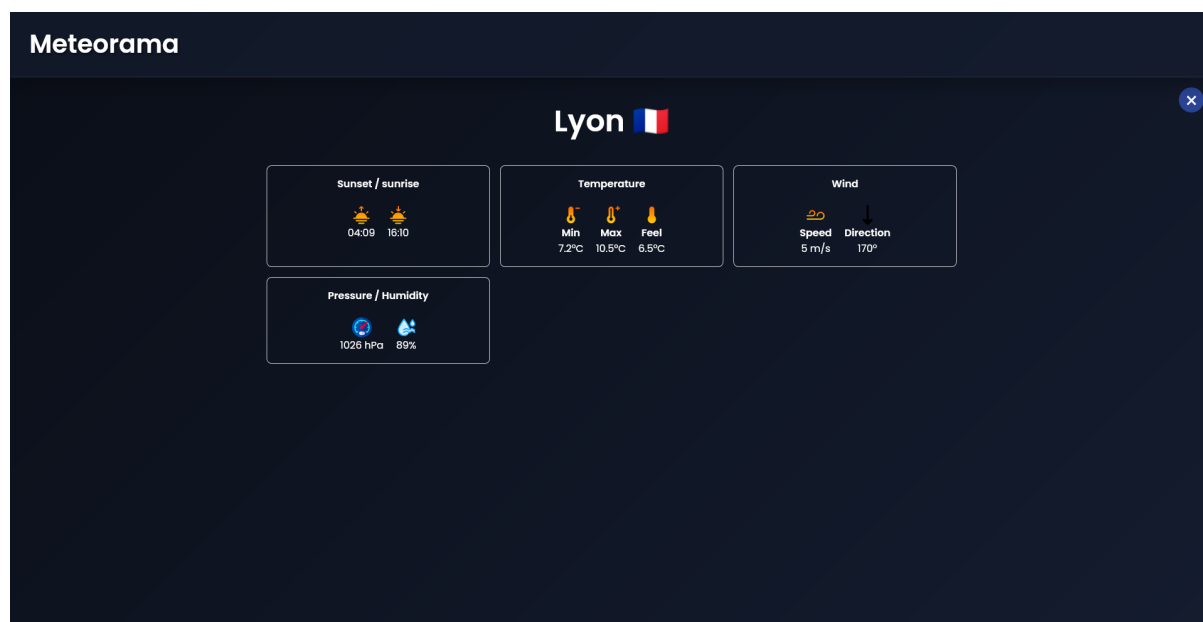


FIGURE 3 – Interface de présentation de données de l'application

Appel des fonctions Rust

Pour obtenir les données météo actuelles, nous avons développé une fonction *Rust* asynchrone qui s'occupe de faire une requête, de traiter et d'arranger les données avant de nous les renvoyer pour pouvoir les afficher. (cf. 2.2.2). La fonction *Rust* marquée avec l'attribut `#[tauri::command(async)]` est appelée depuis le *JavaScript* de la page et est appelée grâce à un `invoke()`. Cet attribut marque la fonction comme une commande asynchrone, permettant de l'exécuter sans bloquer le thread principal. Cela nous permet de ne pas figer l'interface et d'afficher un gif de chargement.

Une fois la partie prévision intégrée à l'interface, d'autres fonctions *Rust* seront appelées.

2.2.6 Site internet

L'objectif pour la réalisation du site internet était que son interface soit cohérente avec celle de l'application que nous vous présentons. Pour ce faire, nous sommes partis sur les mêmes bases de CSS que celle du projet et avons implémenté divers éléments, propre aux sites internet de présentation de projet.

Page d'accueil

Tout d'abord, la page d'accueil du site internet reprend les codes de page internet de présentation de ce genre d'application. Elle est simple et minimaliste contenant des informations claires et concises sur notre projet, quelques encadrés permettant de détailler les fonctionnalités que celui-ci possède et la possibilité de naviguer sur les différentes pages qui permettront de récupérer les diverses ressources du projet.

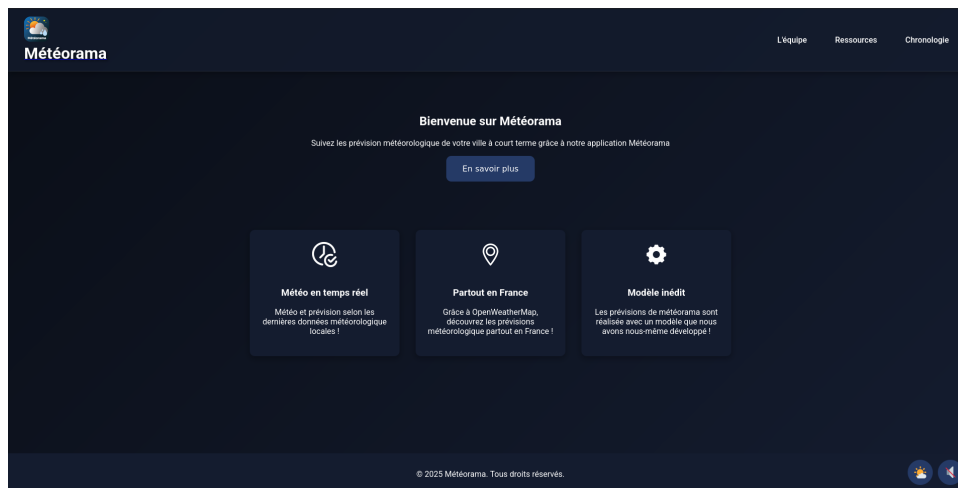


FIGURE 4 – Page d'accueil du site internet

Équipe La page "Équipe" résume notre groupe de projet. Il est possible d'y retrouver une courte présentation du groupe ainsi que de chacun des membres qui le compose. L'objectif était d'obtenir une page minimaliste et moderne contenant les informations nécessaires pour en savoir plus sur notre groupe.

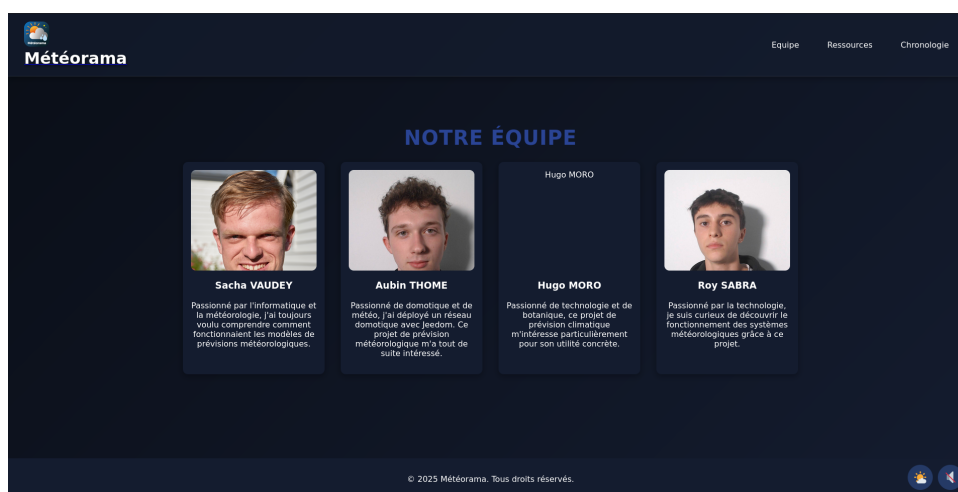


FIGURE 5 – Page d'équipe du site internet

Chronologie Cette page retrace l’histoire du projet et les différentes avancées que nous avons pu faire. Là encore, l’objectif est de reprendre les codes de site internet similaire afin de construire une page cohérente et moderne.



FIGURE 6 – Page de la chronologie du projet du site internet

Ressources Enfin, notre site contient une page de ressources permettant de se procurer l’application, les rapports mais aussi de se rendre sur le repo GitHub de notre projet afin d’en consulter le code source.



FIGURE 7 – Page de ressources du site internet

Fonctionnalités supplémentaires Afin que le site internet corresponde au mieux à l’esprit du projet, nous avons implémenté une fonctionnalité permettant de changer la météo affichée. Cela se traduit par l’apparition, en arrière-plan du site, de la pluie, de la neige ou même du beau temps, grâce à un petit bouton placé en bas de la page pour

activer cette fonctionnalité. Un effet sonore accompagne également chaque type de météo affiché en arrière-plan.

Nous avons voulu créer un site simple et intuitif, où tout est facile à trouver. Que ce soit pour comprendre notre projet, découvrir l'équipe ou accéder aux ressources, tout est fait pour que la navigation reste fluide et agréable. Nous offrons un espace clair et accessible où on trouve facilement toutes les infos dont on a besoin.

3 Analyse

3.1 Points positifs

Comme abordée lors du tableau d'avancement, nous avons largement dépassé nos attentes en terme de développement. En effet, nous avons pu nous approprier plus rapidement que prévu les différentes fonctionnalités de RUST dont nous avons besoins, notamment la structure de projet Tauri qui est la plus complexe d'entre elle ce qui nous a permis de prendre de l'avance sur notre programme.

Grâce à cela, il nous sera possible de prévoir de nouvelles fonctionnalités, qui n'étaient pas prévu dans le cahier des charges afin de compléter notre projet si nous nous rendons compte que nous avons finit ce qui était prévu initialement. Il est encore trop tôt pour pouvoir nous engager sur de potentielles fonctionnalités supplémentaires car nous ne savons pas si sur la suite du développement du projet, certaines choses prendrait plus de temps, mais nous pourrons détailler des fonctionnalités supplémentaire sur le rapport de la deuxième soutenance, puis que nous aurons alors plus de visibilité sur le développement général du projet.

Outre l'avancée générale du projet, nous pouvons aussi souligner la cohésion du groupe qui a été plutôt bonne dès le départ. En effet, nous n'avons jamais travaillé ensemble dans les projets précédents à EPITA, mais cela n'a été en aucun cas un frein au développement et à notre efficacité sur celui-ci.

3.2 Points de complexité

Nous avons eu lors du développement certains problèmes, en particulier avec l'utilisation de Tauri. En effet, Tauri permet de développer une interface d'application grâce aux langage Web. Cela apporte certes beaucoup d'avantage notamment à la facilité de mise en place d'une interface complexe qui est le propre des langages Web, mais apporte une certaine complexité sur l'exécution de fonction Rust depuis du code JavaScript. De ce fait, nous avons rencontrer pas mal de difficulté avant de pouvoir exécuter correctement notre code. Cela s'est en plus couplé à l'utilisation d'asynchrone sur certaine fonctions IO de notre code, notamment pour le traitement de JSON et le traitement de requêtes API puisque cela n'est pas directement supporté par Tauri. Nous avons donc composer avec tout cela afin de pouvoir réaliser le projet correctement.

4 Conclusion

En somme, nous avons réussi, pour cette première soutenance, à vous présenter un projet à la hauteur de nos attentes, voire même au-delà. D'un point de vue technique, nous avons largement réussi à réaliser ce que nous avons prévu lors du dépôt du cahier des charges. Cela nous a permis, en quelques semaines, d'approfondir considérablement nos connaissances du langage RUST, en plus des compétences acquises lors des TP, afin de pouvoir implémenter des fonctionnalités et des algorithmes complexes.

Au-delà de cela, ce projet nous a permis, une fois de plus, d'approfondir nos capacités à travailler en équipe, car nous n'avions jamais travaillé ensemble auparavant. Grâce à ce projet, nous avons pu constater que notre dynamique de travail était plutôt bonne, comme en témoigne l'avancée et le dépassement des objectifs que nous nous étions fixés.