

JOURNAL DE BORD

Projet AurorAlerte

Dashboard Interactif de Surveillance des Aurores Boréales

Université de Montpellier Paul Valéry | Master 2 MIASHS | Décembre 2025

Réalisé par :

Adjimon Jérôme Vitoffodji
Alvin Ingabire

Table des matières

1 Contexte et Objectifs du Projet	3
1.1 Contexte Académique	3
1.2 Inspiration et Choix du Sujet	3
1.3 Objectifs du Projet	3
2 Session de Développement du 29 Novembre 2025	4
2.1 Vue d'Ensemble de la Session	4
2.2 Problème 1 : Optimisation de la Carte Mondiale	4
2.2.1 Symptômes Observés	4
2.2.2 Diagnostic	4
2.2.3 Solutions Explorées	5
2.2.4 Solution Finale Retenue	5
2.2.5 Résultats et Améliorations	5
2.3 Amélioration 2 : Système de Recherche de Villes Dynamique	5
2.3.1 Besoin Identifié	5
2.3.2 Architecture Implémentée	5
2.3.3 Fonctionnalités Clés	6
2.3.4 Statistiques Enrichies	6
2.3.5 Éléments Pédagogiques Ajoutés	7
2.4 Amélioration 3 : Système d'Alertes Email Automatisé	7
2.4.1 Problématique Initiale	7
2.4.2 Solution : Calcul Automatique du Seuil Kp	8
2.4.3 Nouvelle Interface Utilisateur Simplifiée	8
2.4.4 Avantages de l'Approche Automatisée	9
2.4.5 Correction d'un Bug Architectural Critique	9
2.5 Amélioration 4 : Enrichissement des Emails d'Alerte	10
2.5.1 Modification de l'Architecture Email	10
2.5.2 Nouveau Contenu Email Personnalisé	10
2.5.3 Structure de l'Email Amélioré	10
2.6 Tests et Validation du Système d'Alertes	11
2.6.1 Création d'Outils de Test Dédiés	11
2.6.2 Méthodes de Test Documentées	11
2.6.3 Guide de Configuration Gmail	11
2.6.4 Documentation des Problèmes Courants	12
2.7 Livrables de la Session	12
2.7.1 Fichiers de Développement Crées	12
2.7.2 Documentation Technique Produite	13
2.8 Statistiques de la Session	13
3 Technologies et APIs	14
3.1 Stack Technique	14
3.2 APIs Intégrées	14

4 Fonctionnalités Implémentées	15
5 Tests et Validation	16
5.1 Tests Fonctionnels	16
6 Améliorations Futures	17
7 Conclusion	18
7.1 Bilan des Objectifs	18
7.2 Compétences Acquises	18
7.3 Réflexion Finale	18
A Statistiques Finales du Projet	20
B Ressources et Liens	21

Chapitre 1

Contexte et Objectifs du Projet

1.1 Contexte Académique

Ce projet s'inscrit dans le cadre du Master 2 **Open Data et Web des Données** à l'université de Montpellier Paul Valéry, formation Master 2 MIASHS. L'objectif était de créer une application web interactive exploitant des données en temps réel provenant d'APIs publiques, en mettant l'accent sur la visualisation de données scientifiques et météorologiques.

1.2 Inspiration et Choix du Sujet

Le projet est inspiré du dashboard "Aurora Monitor" de **Sheila Géa**.

1.3 Objectifs du Projet

- Créer un dashboard interactif de surveillance des aurores boréales en temps réel
- Intégrer multiple APIs publiques (NOAA, Open-Meteo, OpenWeatherMap, Sunrise-Sunset)
- Traduire l'application en français avec adaptation culturelle complète
- Ajouter des fonctionnalités pédagogiques (descriptions, explications scientifiques)
- Implémenter un système d'alertes par email intelligent et automatisé
- Créer une carte mondiale interactive des probabilités d'observation
- Documenter professionnellement le projet pour présentation académique

Chapitre 2

Session de Développement du 29 Novembre 2025

2.1 Vue d'Ensemble de la Session

Cette session intensive de développement a duré environ 6 heures et a permis d'implémenter plusieurs améliorations majeures au dashboard AurorAlerte. Les travaux ont porté sur trois axes principaux :

1. **Carte mondiale interactive** : Résolution de problèmes d'affichage et ajout d'un système de recherche de villes dynamique
2. **Système d'alertes automatisé** : Implémentation d'un calcul automatique du seuil Kp selon la localisation de l'utilisateur
3. **Amélioration des emails** : Enrichissement du contenu des alertes avec informations contextuelles

2.2 Problème 1 : Optimisation de la Carte Mondiale

2.2.1 Symptômes Observés

La carte mondiale des aurores boréales s'affichait de manière compressée avec beaucoup d'espace blanc inutilisé. Plusieurs tentatives d'augmentation de la hauteur n'avaient aucun effet visible sur l'affichage final.

Observation : La carte apparaissait aplatie horizontalement avec une grande zone blanche en bas, rendant difficile la lecture des informations géographiques et des limites de visibilité des aurores.

2.2.2 Diagnostic

Analyse approfondie des causes du problème :

1. **Projection inadaptée** : La projection cartographique "natural earth" coupait les bords du monde et ne convenait pas pour un focus sur l'hémisphère nord
2. **Hauteur insuffisante** : Les valeurs de 600-700 pixels ne permettaient pas un affichage optimal de la zone géographique pertinente
3. **Plage de latitude excessive** : La plage 40-90°N incluait des zones inutiles pour l'observation des aurores (Afrique du Nord, Amérique du Sud)
4. **Absence de focus géographique** : L'affichage global du monde ne permettait pas de se concentrer efficacement sur l'hémisphère nord où les aurores sont observables

2.2.3 Solutions Explorées

Trois approches progressives ont été testées pour résoudre le problème :

Version	Hauteur	Caractéristiques
Version 1	900px	Augmentation de la hauteur, plage latitude 30-90°N
Version 2	1200px	Taille extra large, plage latitude 25-90°N
Version 3 (finale)	800px	Focus hémisphère nord 40-85°N, projection Mercator, zoom automatique 1.5x

TABLE 2.1 – Versions testées de la carte mondiale

2.2.4 Solution Finale Retenue

La version finale utilise une combinaison optimale de paramètres :

- **Projection Mercator** : Meilleure lisibilité pour l'hémisphère nord que la projection equirectangular
- **Focus 40-85°N** : Exclusion des zones non pertinentes (hémisphère sud, tropiques)
- **Hauteur 800px** : Compromis optimal entre lisibilité et espace écran
- **Zoom 1.5x** : Agrandissement automatique de la zone d'intérêt
- **Centrage lat=60°, lon=0°** : Position optimale pour l'Europe du Nord et l'Arctique

2.2.5 Résultats et Améliorations

Améliorations Obtenues :

- La carte remplit 100% de la hauteur disponible sans espace blanc inutilisé
- Affichage exclusif des zones pertinentes : Europe du Nord, Amérique du Nord, Groenland, Arctique
- Suppression des zones non pertinentes : Afrique, Amérique du Sud, Antarctique, Océan Indien
- Bandes de latitude plus denses (tous les 1° au lieu de 5°) pour une meilleure précision visuelle
- Amélioration significative de la lisibilité des villes et des limites de visibilité des aurores

2.3 Amélioration 2 : Système de Recherche de Villes Dynamique

2.3.1 Besoin Identifié

Le dashboard affichait initialement 9 villes principales pré-configurées (Longyearbyen, Tromsø, Reykjavik, Stockholm, Oslo, Édimbourg, Londres, Paris, Berlin). Les utilisateurs souhaitaient pouvoir ajouter leurs propres localisations d'intérêt sans avoir à modifier le code source de l'application.

2.3.2 Architecture Implémentée

Mise en place d'un système à deux niveaux de villes :

1. **Villes principales (9 villes fixes)** : Toujours affichées automatiquement
 - Représentation : Cercles noirs
 - Couleur dynamique : Vert si aurores visibles, Rouge si invisibles
 - Taille : 16 pixels si visible, 12 pixels si invisible
2. **Villes personnalisées (maximum 5)** : Ajoutées par recherche utilisateur

- Représentation : Losanges dorés
- Couleur dynamique : Jaune si aurores visibles, Orange si invisibles
- Taille : 14 pixels uniformément

2.3.3 Fonctionnalités Clés

Géocodage Intelligent

Le système utilise l'API de géocodage Open-Meteo pour convertir automatiquement les noms de villes entrés par l'utilisateur en coordonnées géographiques précises (latitude, longitude). Cette approche permet :

- Support de noms de villes en plusieurs langues
- Résolution automatique des ambiguïtés géographiques
- Gestion des variantes orthographiques
- Récupération des coordonnées exactes pour positionnement précis sur la carte

Détection Intelligente des Doublons

Système sophistiqué de détection des doublons utilisant deux méthodes complémentaires :

1. **Vérification par nom** : Comparaison insensible à la casse des noms de villes pour détecter les entrées identiques
2. **Vérification par proximité géographique** : Détection de villes situées à moins de 0.5° de distance (environ 55 km), ce qui permet d'éviter l'affichage multiple de villes très proches (exemple : Paris et sa banlieue proche)

Cette double vérification garantit qu'aucune ville n'apparaît deux fois sur la carte, améliorant la clarté de la visualisation.

Messages de Feedback Utilisateur

Système complet et contextuel de retour d'information à l'utilisateur :

- **Message de doublons** : "Ville(s) déjà présente(s) : Stockholm, Paris. Veuillez saisir d'autres villes."
- **Message de succès** : "3 ville(s) ajoutée(s) sur la carte!"
- **Message mixte** : Affichage combiné des deux types de messages si certaines villes sont ajoutées et d'autres rejetées
- **Message d'échec** : "Aucune ville ajoutée. Vérifiez les noms saisis." si aucune ville n'a pu être géocodée

2.3.4 Statistiques Enrichies

Ajout de quatre métriques calculées en temps réel sous la carte mondiale :

Métrique	Description
Latitude Limite	Latitude minimale à partir de laquelle les aurores sont théoriquement visibles, calculée selon l'indice Kp actuel
Distance à Limite	Distance en kilomètres entre la localisation choisie par l'utilisateur et la limite de visibilité, avec indication de direction (dans la zone visible ou distance vers le nord)
Aurores Ici	Indicateur binaire (OUI / NON) indiquant si les aurores sont actuellement visibles depuis la localisation choisie
Villes Visibles	Nombre et pourcentage de villes affichées sur la carte depuis lesquelles les aurores sont actuellement visibles

TABLE 2.2 – Statistiques affichées sous la carte mondiale

2.3.5 Éléments Pédagogiques Ajoutés

Légende Interactive

Création d'une légende visuelle composée de 5 cartes colorées avec dégradés CSS (hauteur 130 pixels) :

- Zone Verte : Région où les aurores sont actuellement visibles
- Ligne Dorée : Limite de visibilité correspondant à l'indice Kp actuel
- Zone Rouge : Région où les aurores ne sont pas visibles actuellement
- Villes Principales : Représentation des 9 villes pré-configurées (cercles)
- Villes Personnalisées : Représentation des villes ajoutées par l'utilisateur (losanges)

Tableau d'Interprétation Kp

Tableau pédagogique de 10 lignes (Kp 0 à 9) avec mise en forme conditionnelle dynamique :

- **Ligne actuelle** (Kp en cours) : Surlignée en gradient vert pour identification immédiate
- **Lignes passées** (Kp inférieur à l'actuel) : Fond vert clair indiquant des conditions déjà atteintes
- **Lignes futures** (Kp supérieur à l'actuel) : Fond rouge clair indiquant des conditions nécessitant plus d'activité géomagnétique

Colonnes du tableau : Indice Kp, Latitude limite, Régions géographiques visibles, Fréquence d'observation

2.4 Amélioration 3 : Système d'Alertes Email Automatisé

2.4.1 Problématique Initiale

Dans la version précédente du système d'alertes, l'utilisateur devait accomplir manuellement trois étapes :

1. Entrer son adresse email
2. Choisir manuellement un seuil Kp sur une échelle de 0 à 9
3. Deviner empiriquement quel indice Kp correspondait à sa ville

Problème d'Expérience Utilisateur Identifié :

L'utilisateur moyen ne possède pas les connaissances scientifiques nécessaires pour déterminer quel seuil Kp choisir pour sa localisation géographique. Cette situation créait une confusion : "Est-ce que Kp 5.0 est approprié pour Stockholm ? Pour Paris ? Pour Tromsø ?"

Cette ambiguïté rendait le système d'alertes difficile à configurer correctement et peu intuitif, limitant son adoption et son efficacité.

2.4.2 Solution : Calcul Automatique du Seuil Kp

Principe Fondamental

Développement d'un algorithme qui calcule automatiquement l'indice Kp minimum nécessaire pour observer les aurores boréales à une latitude géographique donnée. Cette approche élimine complètement le besoin pour l'utilisateur de comprendre la relation entre indice Kp et latitude d'observation.

Table de Correspondance Scientifique

Utilisation d'une table de correspondance basée sur les modèles scientifiques de géomagnétisme :

Indice Kp	Latitude (°N)	Exemples de Villes
0	66.5	Cercle arctique
1	64.5	Nord de l'Islande
2	62.4	Laponie
3	60.4	Rovaniemi (Finlande)
4	58.3	Stockholm, Helsinki
5	56.3	Écosse
6	54.2	Nord de l'Angleterre, Danemark
7	52.2	Londres, Amsterdam
8	50.1	Bruxelles, Nord de la France
9	48.1	Paris, Munich

TABLE 2.3 – Correspondance KpLatitude minimale

Exemples de Calcul Automatique

Le système calcule automatiquement le seuil optimal selon la ville choisie :

- **Stockholm** (59.33°N) → Kp minimum calculé : 4
- **Tromsø** (69.65°N) → Kp minimum calculé : 1
- **Paris** (48.85°N) → Kp minimum calculé : 9
- **Londres** (51.51°N) → Kp minimum calculé : 8

2.4.3 Nouvelle Interface Utilisateur Simplifiée

Workflow Optimisé

Le processus de configuration des alertes est désormais réduit à 4 étapes simples :

1. L'utilisateur sélectionne sa ville dans l'interface (exemple : Stockholm)
2. Le système calcule automatiquement et silencieusement : Kp minimum = 4
3. L'utilisateur entre son adresse email et valide
4. Le système affiche un message contextuel : "Bonne localisation ! Aurores régulières (Kp >= 4)"

Messages Contextuels Intelligents

Le système génère des messages personnalisés en fonction du Kp minimum calculé, offrant une évaluation qualitative de la localisation pour l'observation des aurores :

Kp Min	Type Message	Message Affiché
0-2	Succès (vert)	"Excellente localisation ! Aurores fréquentes. Vous en verrez souvent !"
3-5	Info (bleu)	"Bonne localisation ! Les aurores sont régulièrement visibles ici."
6-7	Avertissement (jaune)	"Aurores rares ici. Tempêtes géomagnétiques nécessaires. Profitez de cette occasion !"
8-9	Erreur (rouge)	"Aurores très rares. Événements géomagnétiques extrêmes requis. Conseil : Voyagez plus au nord !"

TABLE 2.4 – Messages personnalisés selon le Kp minimum

Paramètres Avancés pour Utilisateurs Expérimentés

Un panneau extensible "Paramètres Avancés" permet aux utilisateurs expérimentés de :

- Activer la personnalisation manuelle du seuil Kp (désactivée par défaut)
- Modifier l'intervalle minimum entre deux alertes (0.5 à 6 heures, défaut 1 heure)
- Visualiser le calcul automatique pour comprendre le raisonnement du système

Par défaut, ce panneau reste caché et le mode automatique intelligent est utilisé, simplifiant l'expérience pour 95% des utilisateurs.

2.4.4 Avantages de l'Approche Automatisée

Améliorations Significatives de l'Expérience Utilisateur :

- Simplicité accrue : Élimination de la configuration manuelle du seuil Kp technique
- Cohérence logique : Alignement avec le reste du dashboard où la localisation est centrale
- Intelligence contextuelle : Calcul automatique précis basé sur des données scientifiques
- Fiabilité maximale : Impossibilité de mal configurer le système
- Valeur pédagogique : L'utilisateur comprend pourquoi ce seuil spécifique est approprié
- Professionnalisme : L'application s'adapte à l'utilisateur, pas l'inverse

2.4.5 Correction d'un Bug Architectural Critique

Erreur DéTECTée

Une erreur système apparaissait lors du chargement de l'application : `NameError: name 'kp_zones' is not defined`

Analyse de la Cause

Investigation révélant que la variable globale `kp_zones` était définie à l'intérieur d'un bloc conditionnel spécifique (onglet "Carte mondiale") mais utilisée dans une section exécutée avant (système d'alertes).

Principe d'architecture violé : En Python/Streamlit, toute variable utilisée dans plusieurs endroits du code doit être définie GLOBALEMENT au début de l'application, avant toute logique conditionnelle ou de navigation par onglets.

Solution Implémentée

Déplacement de la définition de `kp_zones` vers le début du fichier principal, immédiatement après la configuration de la page Streamlit, transformant ainsi cette table en constante globale accessible depuis n'importe quel point de l'application.

Cette correction a également amélioré la maintenabilité du code en clarifiant que cette table est une donnée scientifique fondamentale du projet, pas une configuration spécifique à un onglet.

2.5 Amélioration 4 : Enrichissement des Emails d'Alerte

2.5.1 Modification de l'Architecture Email

Le système d'envoi d'email a été enrichi pour inclure automatiquement le contexte de localisation de l'utilisateur, transformant une alerte générique en notification personnalisée et pédagogique.

2.5.2 Nouveau Contenu Email Personnalisé

Section Informative Contextuelle

Ajout d'une section dédiée dans chaque email expliquant le contexte géographique :

- **Pour localisations excellentes** (Kp min 0-2) : "Excellente nouvelle ! Votre localisation est idéale pour observer les aurores. Vous en verrez souvent !"
- **Pour localisations bonnes** (Kp min 3-5) : "Bonne localisation ! Les aurores sont régulièrement visibles ici."
- **Pour localisations rares** (Kp min 6-7) : "Les aurores sont rares à cette latitude. Profitez de cette occasion exceptionnelle !"
- **Pour localisations très rares** (Kp min 8-9) : "Événement exceptionnel ! Les aurores sont très rares ici. Ne manquez pas ce spectacle unique !"

Nouvelles Métriques Affichées

L'email HTML enrichi présente désormais 5 métriques clés au lieu de 4 :

1. **Indice Kp Actuel** : Valeur en temps réel de l'activité géomagnétique (sur 9)
2. **Kp Minimum Requis** : Seuil calculé spécifiquement pour cette localisation (NOUVEAU)
3. **Score de Probabilité** : Score composite prenant en compte Kp, météo et obscurité (sur 1.0)
4. **Ciel Dégagé** : Pourcentage de ciel sans nuages
5. **Obscurité** : Indicateur nuit/jour avec icône appropriée

2.5.3 Structure de l'Email Amélioré

Format HTML Professionnel

L'email utilise un template HTML responsive avec :

- **En-tête dynamique** : Couleur de fond adaptée au score (vert pour excellent, jaune pour bon, rouge pour moyen)
- **Section d'information contextuelle** : Fond jaune clair avec bordure pour attirer l'attention sur le message personnalisé

- **Métriques visuelles** : Présentation en grille des 5 indicateurs clés avec typographie hiérarchisée
- **Conseils d'observation** : Liste détaillée (meilleure période, lieu idéal, direction, patience, conseils photo)
- **Bouton d'action** : Lien cliquable vers le dashboard complet
- **Footer informatif** : Timestamp UTC précis et informations sur l'origine de l'alerte

Version Texte Alternative

Pour garantir la compatibilité avec tous les clients email, chaque email inclut également une version texte brut contenant toutes les informations essentielles dans un format lisible sans HTML.

2.6 Tests et Validation du Système d'Alertes

2.6.1 Crédit d'Outils de Test Dédiés

Pour faciliter les tests du système d'alertes email sans nécessiter l'activation complète du dashboard, un script Python dédié a été développé : `test_email.py`

Fonctionnalités du Script de Test

- Chargement automatique de la configuration SMTP depuis le fichier de secrets
- Interface interactive en ligne de commande pour entrer l'email de test
- Validation automatique du format de l'adresse email
- Envoi d'un email avec des données de test prédéfinies (Kp 6.5, Stockholm, score 0.82)
- Affichage de messages de debug détaillés pour faciliter le diagnostic
- Fourniture automatique de solutions en cas d'erreur d'envoi

2.6.2 Méthodes de Test Documentées

Trois approches complémentaires de test ont été documentées :

Méthode	Description et Usage
Script <code>test_email.py</code>	Script Python standalone avec interface interactive, idéal pour tests rapides et debugging
Test depuis Dashboard	Modification temporaire du seuil Kp à 0.0 pour forcer l'envoi immédiat d'une alerte de test
Test interactif Python	Console Python avec import manuel, utile pour tests unitaires de la fonction d'envoi

TABLE 2.5 – Méthodes de test disponibles

2.6.3 Guide de Configuration Gmail

Processus de Configuration Documenté

Un guide détaillé a été créé pour configurer Gmail avec mot de passe d'application :

1. Accéder aux paramètres de sécurité du compte Google
2. Activer l'authentification à deux facteurs (prérequis obligatoire)
3. Générer un mot de passe d'application spécifique pour "AurorAlerte"
4. Copier le mot de passe généré (16 caractères) dans le fichier de configuration
5. Tester la connexion avec le script de test

Note Importante de Sécurité :

Il est impératif d'utiliser un **mot de passe d'application** Gmail et non le mot de passe principal du compte. Cette pratique garantit :

- Meilleure sécurité (le mot de passe peut être révoqué sans changer le mot de passe principal)
- Traçabilité des accès
- Conformité avec les recommandations de sécurité de Google

2.6.4 Documentation des Problèmes Courants

Création d'un guide de résolution des erreurs fréquemment rencontrées :

Erreur Rencontrée	Solution Documentée
Authentification échouée	Vérifier l'utilisation d'un mot de passe d'application et l'activation de l'authentification à deux facteurs
Connection refused / Timeout	Désactiver temporairement l'antivirus, autoriser Python dans le pare-feu, vérifier que le port 587 n'est pas bloqué
Invalid sender	Confirmer que l'adresse email expéditeur correspond exactement au compte Gmail utilisé
Email non reçu	Attendre 5-10 minutes, vérifier minutieusement le dossier SPAM/Courrier indésirable

TABLE 2.6 – Guide de résolution des problèmes

2.7 Livrables de la Session

2.7.1 Fichiers de Développement Crées

Au total, 11 fichiers ont été créés pendant cette session :

Fichier	Contenu et Utilité
carte_mercator.py	Première version corrigée (900px)
carte_mercator_1200px.py	Version expérimentale extra large
carte_hemisphere_nord.py	Version finale optimisée recommandée
carte_avec_recherche_villes.py	Carte complète avec système de recherche
alertes_avec_validation.py	Système d'alertes avec validation email
alertes_automatiques.py	Version finale automatisée complète
alerts_ameliore.py	Fichier model/alerts.py enrichi
kp_zones_definition.py	Définition globale de la table Kp-Latitude
test_email.py	Script de test du système d'alertes

TABLE 2.7 – Fichiers Python créés

2.7.2 Documentation Technique Produite

Document	Contenu
INTEGRATION_ALERTES_AUTO.md	Guide complet d'intégration du système automatisé (40+ pages)
GUIDE_TEST_EMAIL.md	Documentation complète des tests et résolution de problèmes (25+ pages)

TABLE 2.8 – Documentation créée

2.8 Statistiques de la Session

Métrique	Valeur
Durée totale de la session	environ 6 heures
Problèmes majeurs résolus	4
Nouvelles fonctions créées	3
Lignes de code ajoutées/modifiées	environ 500
Fichiers créés (code + docs)	11
Captures d'écran analysées	3
Versions de carte testées	3
Types de messages utilisateur	8
Pages de documentation	65+

TABLE 2.9 – Statistiques détaillées de la session

Chapitre 3

Technologies et APIs

3.1 Stack Technique

Technologie	Version	Utilisation
Python	3.10+	Langage principal du backend
Streamlit	1.28+	Framework web interactif
Plotly	5.17+	Visualisations graphiques interactives
Pandas	2.1+	Manipulation et analyse de données
Requests	2.31+	Appels APIs REST
smtplib	built-in	Envoi d'emails via SMTP
PIL/Pillow	10.0+	Traitement d'images

TABLE 3.1 – Technologies principales utilisées

3.2 APIs Intégrées

- NOAA Space Weather Prediction Center (Kp, prévisions, animations OVATION)
- Open-Meteo (météo, géocodage)
- OpenWeatherMap (météo actuelle, icônes)
- Sunrise-Sunset (obscurité)

Chapitre 4

Fonctionnalités Implémentées

- 3 jauge interactives temps réel
- Historique Kp 4 heures avec export CSV
- Carte mondiale Mercator focalisée hémisphère nord (40-85°N)
- Recherche de 5 villes personnalisées + détection automatique des doublons
- Calcul automatique du seuil Kp selon la localisation
- Messages contextuels personnalisés (4 niveaux)
- Emails HTML enrichis avec 5 métriques
- Documentation technique complète (65+ pages)

Chapitre 5

Tests et Validation

5.1 Tests Fonctionnels

Test Effectué	Résultat
Traduction complète français	Passé
4 APIs fonctionnelles	Passé
Géocodage français (Suède, Norvège)	Passé
Alertes email (test forcé)	Passé
Carte zones Kp 3, 5, 7, 9	Passé
Recherche villes + doublons	Passé
Calcul Kp auto (3 villes testées)	Passé
Export CSV	Passé
Responsive 2 résolutions	Passé
Gestion erreurs	Passé

TABLE 5.1 – Résultats tests fonctionnels

Chapitre 6

Améliorations Futures

- Historique observations utilisateur (BDD), Alertes SMS Twilio, Favoris localisations multiples
- Mode hors ligne (cache), Webcams additionnelles, Support multilingue (EN, SV, NO), Timeline Kp 24h
- Prédictions ML, Application mobile, Partage social

Chapitre 7

Conclusion

7.1 Bilan des Objectifs

Le projet AurorAlerte a dépassé ses objectifs initiaux en créant un dashboard professionnel combinant :

- 4 APIs temps réel parfaitement intégrées
- Traduction française complète et localisée
- 15+ emplacements pédagogiques détaillés
- Système alertes intelligent automatisé
- Carte mondiale avec recherche dynamique
- Export données multiples formats
- 9 problèmes techniques majeurs documentés et résolus
- Guide test complet (65+ pages documentation)

7.2 Compétences Acquises

Compétences Techniques : Python/Streamlit avancé, APIs REST multiples, Plotly visualisation géographique, SMTP/Gmail sécurisé, Pandas manipulation données, Gestion états Streamlit, Projections cartographiques, Géocodage multilingue

Méthodologie Professionnelle : Debugging systématique, Documentation technique exhaustive, Gestion version Git, Architecture logicielle modulaire, Tests fonctionnels, Scripts test autonomes, Résolution problèmes utilisateur

Soft Skills Développés : Adaptation code existant, Traduction/localisation culturelle, Pédagogie scientifique, Gestion projet académique, UX design centré utilisateur, Feedback contextuel, Communication technique

7.3 Réflexion Finale

Ce projet s'est étendu sur plusieurs semaines avec une session intensive finale le 29 novembre 2025. L'approche choisie d'adapter un code existant plutôt que repartir de zéro s'est révélée extrêmement formatrice, reflétant fidèlement les pratiques professionnelles réelles.

La session du 29 novembre a particulièrement démontré l'importance d'une méthodologie rigoureuse : analyse détaillée des problèmes avec documentation visuelle, exploration de solutions multiples avant décision, tests systématiques à chaque étape, et documentation exhaustive pour reproductibilité.

L'aspect le plus gratifiant du projet a été sa dimension pédagogique : transformer des concepts scientifiques complexes (géomagnétisme, météorologie, projections géographiques) en expérience utilisateur accessible et engageante.

La refonte du système d'alertes illustre parfaitement le principe central du design centré utilisateur : le système doit s'adapter intelligemment à l'utilisateur, jamais l'inverse.

AurorAlerte représente désormais un outil opérationnel et pédagogique complet, prêt au déploiement pour passionnés d'aurores boréales francophones.

Le code source documenté, le README détaillé et ce journal de bord exhaustif permettent à d'autres développeurs de contribuer et d'étendre le projet.

L'application démontre qu'il est possible de créer des outils scientifiques qui soient simultanément accessibles, esthétiques, rigoureux et pédagogiques en combinant APIs publiques, frameworks modernes et attention méticuleuse au design d'expérience utilisateur.

Annexe A

Statistiques Finales du Projet

Métrique	Valeur
Lignes de code Python	environ 2000
Lignes de documentation	environ 2500
Fonctions développées	30+
APIs intégrées	4
Langues supportées	2 (FR + EN pays)
Onglets interface	7
Visualisations Plotly	20+
Temps développement total	environ 50 heures
Commits Git	40+
Problèmes techniques résolus	9
Tests fonctionnels documentés	10
Pages documentation technique	65+

TABLE A.1 – Statistiques complètes (mise à jour 29/11/2025)

Annexe B

Ressources et Liens

- Site web : <https://web-production-ff2d6.up.railway.app/#aura-hunter>
- Version Github : https://github.com/JeromeVitoff/Open_Data-Web_donnees
- NOAA SWPC : <https://www.swpc.noaa.gov/>
- Open-Meteo : <https://open-meteo.com/>
- OpenWeatherMap : <https://openweathermap.org/>
- Sunrise-Sunset : <https://sunrise-sunset.org/api>
- Streamlit : <https://docs.streamlit.io/>

Document réalisé par :

Adjimon Jérôme Vitoffodji et Alvin Ingabire

Master 2 MIASHS, Open Data et Web des Données