

# JOURNAL DE BORD

## Projet AurorAlerte

### Dashboard Interactif de Surveillance des Aurores Boreales

Master 2 Open Data et Web des Donnees  
Universite de Montpellier Paul Valéry | Master 2 MIASHS | Novembre 2025

#### Realise par :

Adjimon Jerome Vitoffodji  
Alvin Ingabire

# Table des matières

<b>1</b>	<b>Contexte et Objectifs du Projet</b>	<b>3</b>
1.1	Contexte Academique . . . . .	3
1.2	Inspiration et Choix du Sujet . . . . .	3
1.3	Objectifs du Projet . . . . .	3
<b>2</b>	<b>Session de Developpement du 29 Novembre 2025</b>	<b>4</b>
2.1	Vue d'Ensemble de la Session . . . . .	4
2.2	Probleme 1 : Optimisation de la Carte Mondiale . . . . .	4
2.2.1	Symptomes Observees . . . . .	4
2.2.2	Diagnostic . . . . .	4
2.2.3	Solutions Explorees . . . . .	5
2.2.4	Solution Finale Retenue . . . . .	5
2.2.5	Resultats et Ameliorations . . . . .	5
2.3	Amelioration 2 : Systeme de Recherche de Villes Dynamique . . . . .	5
2.3.1	Besoin Identifie . . . . .	5
2.3.2	Architecture Implementee . . . . .	5
2.3.3	Fonctionnalites Cles . . . . .	6
2.3.4	Statistiques Enrichies . . . . .	6
2.3.5	Elements Pedagogiques Ajoutes . . . . .	7
2.4	Amelioration 3 : Systeme d'Alertes Email Automatise . . . . .	7
2.4.1	Problematiche Initiale . . . . .	7
2.4.2	Solution : Calcul Automatique du Seuil Kp . . . . .	8
2.4.3	Nouvelle Interface Utilisateur Simplifiee . . . . .	8
2.4.4	Avantages de l'Approche Automatisee . . . . .	9
2.4.5	Correction d'un Bug Architectural Critique . . . . .	9
2.5	Amelioration 4 : Enrichissement des Emails d'Alerte . . . . .	10
2.5.1	Modification de l'Architecture Email . . . . .	10
2.5.2	Nouveau Contenu Email Personnalise . . . . .	10
2.5.3	Structure de l'Email Ameliore . . . . .	10
2.6	Tests et Validation du Systeme d'Alertes . . . . .	11
2.6.1	Creation d'Outils de Test Dedies . . . . .	11
2.6.2	Methodes de Test Documentees . . . . .	11
2.6.3	Guide de Configuration Gmail . . . . .	11
2.6.4	Documentation des Problemes Courants . . . . .	12
2.7	Livrables de la Session . . . . .	12
2.7.1	Fichiers de Developpement Crees . . . . .	12
2.7.2	Documentation Technique Produite . . . . .	13
2.8	Statistiques de la Session . . . . .	13
<b>3</b>	<b>Technologies et APIs</b>	<b>14</b>
3.1	Stack Technique . . . . .	14
3.2	APIs Integrees . . . . .	14

<b>4</b>	<b>Fonctionnalites Implementees</b>	<b>15</b>
<b>5</b>	<b>Tests et Validation</b>	<b>16</b>
5.1	Tests Fonctionnels . . . . .	16
<b>6</b>	<b>Ameliorations Futures</b>	<b>17</b>
<b>7</b>	<b>Conclusion</b>	<b>18</b>
7.1	Bilan des Objectifs . . . . .	18
7.2	Competences Acquises . . . . .	18
7.3	Reflexion Finale . . . . .	18
<b>A</b>	<b>Statistiques Finales du Projet</b>	<b>20</b>
<b>B</b>	<b>Ressources et Liens</b>	<b>21</b>

# Chapitre 1

## Contexte et Objectifs du Projet

### 1.1 Contexte Academique

Ce projet s'inscrit dans le cadre du Master 2 **Open Data et Web des Données** a 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 Gea**.

### 1.3 Objectifs du Projet

- Créer un dashboard interactif de surveillance des aurores boreales en temps reel
- Integrer multiple APIs publiques (NOAA, Open-Meteo, OpenWeatherMap, Sunrise-Sunset)
- Traduire l'application en francais avec adaptation culturelle complete
- Ajouter des fonctionnalites pedagogiques (descriptions, explications scientifiques)
- Implementer un systeme d'alertes par email intelligent et automatise
- Créer une carte mondiale interactive des probabilites d'observation
- Documenter professionnellement le projet pour presentation academique

## Chapitre 2

# Session de Developpement du 29 Novembre 2025

### 2.1 Vue d'Ensemble de la Session

Cette session intensive de developpement a dure environ 6 heures et a permis d'implementer plusieurs ameliorations majeures au dashboard AurorAlerte. Les travaux ont porte sur trois axes principaux :

1. **Carte mondiale interactive** : Resolution de problemes d'affichage et ajout d'un systeme de recherche de villes dynamique
2. **Systeme d'alertes automatise** : Implementation d'un calcul automatique du seuil Kp selon la localisation de l'utilisateur
3. **Amelioration des emails** : Enrichissement du contenu des alertes avec informations contextuelles

### 2.2 Probleme 1 : Optimisation de la Carte Mondiale

#### 2.2.1 Symptomes Observes

La carte mondiale des aurores boreales s'affichait de maniere compressee avec beaucoup d'espace blanc inutilise. 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 geographiques et des limites de visibilite des aurores.

#### 2.2.2 Diagnostic

Analyse approfondie des causes du probleme :

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

### 2.2.3 Solutions Explorees

Trois approches progressives ont ete testees pour resoudre le probleme :

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

TABLE 2.1 – Versions testees de la carte mondiale

### 2.2.4 Solution Finale Retenue

La version finale utilise une combinaison optimale de parametres :

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

### 2.2.5 Resultats et Ameliorations

#### Ameliorations Obtenues :

- La carte remplit 100
- Affichage exclusif des zones pertinentes : Europe du Nord, Amerique du Nord, Groenland, Arctique
- Suppression des zones non pertinentes : Afrique, Amerique du Sud, Antarctique, Ocean Indien
- Bandes de latitude plus denses (tous les 1 degrees au lieu de 5 degrees) pour une meilleure precision visuelle
- Amelioration significative de la lisibilite des villes et des limites de visibilite des aurores

## 2.3 Amelioration 2 : Systeme de Recherche de Villes Dynamique

### 2.3.1 Besoin Identifie

Le dashboard affichait initialement 9 villes principales pre-configurees (Longyearbyen, Tromso, Reykjavik, Stockholm, Oslo, Edimbourg, Londres, Paris, Berlin). Les utilisateurs souhaitent pouvoir ajouter leurs propres localisations d’interet sans avoir a modifier le code source de l’application.

### 2.3.2 Architecture Implementee

Mise en place d’un systeme a deux niveaux de villes :

1. **Villes principales (9 villes fixes)** : Toujours affichees automatiquement
  - Representation : 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 Cles

#### Geocodage Intelligent

Le système utilise l'API de geocodage Open-Meteo pour convertir automatiquement les noms de villes entrées 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 degrés 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 :

Metrique	Description
Latitude Limite	Latitude minimale a partir de laquelle les aurores sont theoriquement visibles, calculee selon l'indice Kp actuel
Distance a Limite	Distance en kilometres entre la localisation choisie par l'utilisateur et la limite de visibilite, 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 affichees sur la carte depuis lesquelles les aurores sont actuellement visibles

TABLE 2.2 – Statistiques affichees sous la carte mondiale

### 2.3.5 Elements Pedagogiques Ajoutes

#### Legende Interactive

Creation d'une legende visuelle composee de 5 cartes colorees avec degradés CSS (hauteur 130 pixels) :

- Zone Verte : Region ou les aurores sont actuellement visibles
- Ligne Doree : Limite de visibilite correspondant a l'indice Kp actuel
- Zone Rouge : Region ou les aurores ne sont pas visibles actuellement
- Villes Principales : Representation des 9 villes pre-configurees (cercles)
- Villes Personnalisees : Representation des villes ajoutes par l'utilisateur (losanges)

#### Tableau d'Interpretation Kp

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

- **Ligne actuelle** (Kp en cours) : Surlignee en gradient vert pour identification immediate
- **Lignes passees** (Kp inferieur a l'actuel) : Fond vert clair indiquant des conditions deja atteintes
- **Lignes futures** (Kp superieur a l'actuel) : Fond rouge clair indiquant des conditions necessitant plus d'activite geomagnetique

Colonnes du tableau : Indice Kp, Latitude limite, Regions geographiques visibles, Frequence d'observation

## 2.4 Amelioration 3 : Systeme d'Alertes Email Automatise

### 2.4.1 Problematique Initiale

Dans la version precedente du systeme d'alertes, l'utilisateur devait accomplir manuellement trois etapes :

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



**Probleme d'Experience Utilisateur Identifie :**

L'utilisateur moyen ne possede pas les connaissances scientifiques necessaires pour determiner quel seuil Kp choisir pour sa localisation geographique. Cette situation creait une confusion : "Est-ce que Kp 5.0 est approprie pour Stockholm ? Pour Paris ? Pour Tromso ?"

Cette ambiguïté rendait le systeme d'alertes difficile a configurer correctement et peu intuitif, limitant son adoption et son efficacite.

**2.4.2 Solution : Calcul Automatique du Seuil Kp****Principe Fondamental**

Developpement d'un algorithme qui calcule automatiquement l'indice Kp minimum necessaire pour observer les aurores boreales a une latitude geographique donnee. Cette approche elimine completement 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 basee sur les modeles scientifiques de geomagnetisme :

Indice Kp	Latitude (degreés 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	Ecosse
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 Kp – Latitude minimale

**Exemples de Calcul Automatique**

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

- **Stockholm** (59.33 degrees N) – Kp minimum calcule : 4
- **Tromso** (69.65 degrees N) – Kp minimum calcule : 1
- **Paris** (48.85 degrees N) – Kp minimum calcule : 9
- **Londres** (51.51 degrees N) – Kp minimum calcule : 8

**2.4.3 Nouvelle Interface Utilisateur Simplifiee****Workflow Optimise**

Le processus de configuration des alertes est desormais reduit a 4 etapes simples :

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

## Messages Contextuels Intelligents

Le systeme genere des messages personnalisés en fonction du Kp minimum calcule, offrant une evaluation qualitative de la localisation pour l'observation des aurores :

Kp Min	Type Message	Message Affiche
0-2	Succes (vert)	"Excellente localisation! Aurores frequentes. Vous en verrez souvent!"
3-5	Info (bleu)	"Bonne localisation! Les aurores sont regulierement visibles ici."
6-7	Avertissement (jaune)	"Aurores rares ici. Tempetes geomagnetiques necessaires. Profitez de cette occasion!"
8-9	Erreur (rouge)	"Aurores tres rares. Evenements geomagnetiques extremes requis. Conseil : Voyagez plus au nord!"

TABLE 2.4 – Messages personnalisés selon le Kp minimum

## Parametres Avances pour Utilisateurs Experimentes

Un panneau extensible "Parametres Avances" permet aux utilisateurs experimentes de :

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

Par default, ce panneau reste cache et le mode automatique intelligent est utilise, simplifiant l'experience pour 95

### 2.4.4 Avantages de l'Approche Automatisee

#### Ameliorations Significatives de l'Experience Utilisateur :

- Simplicite accrue : Elimination de la configuration manuelle du seuil Kp technique
- Coherence logique : Alignement avec le reste du dashboard ou la localisation est centrale
- Intelligence contextuelle : Calcul automatique precis base sur des donnees scientifiques
- Fiabilite maximale : Impossibilite de mal configurer le systeme
- Valeur pedagogique : L'utilisateur comprend pourquoi ce seuil specifique est approprie
- Professionnalisme : L'application s'adapte a l'utilisateur, pas l'inverse

### 2.4.5 Correction d'un Bug Architectural Critique

#### Erreur Detectee

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

#### Analyse de la Cause

Investigation revelant que la variable globale `kp_zones` etait definie a l'interieur d'un bloc conditionnel specifique (onglet "Carte mondiale") mais utilisee dans une section executee avant (systeme d'alertes).

**Principe d'architecture viole** : En Python/Streamlit, toute variable utilisee dans plusieurs endroits du code doit etre definie GLOBALEMENT au debut de l'application, avant toute logique conditionnelle ou de navigation par onglets.

### Solution Implementee

Deplacement de la definition de `kp_zones` vers le debut du fichier principal, immediatement apres 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 egalement ameliore la maintenabilite du code en clarifiant que cette table est une donnee scientifique fondamentale du projet, pas une configuration specifique a un onglet.

## 2.5 Amelioration 4 : Enrichissement des Emails d'Alerte

### 2.5.1 Modification de l'Architecture Email

Le systeme d'envoi d'email a ete enrichi pour inclure automatiquement le contexte de localisation de l'utilisateur, transformant une alerte generique en notification personnalisee et pedagogique.

### 2.5.2 Nouveau Contenu Email Personnalise

#### Section Informative Contextuelle

Ajout d'une section dediee dans chaque email expliquant le contexte geographique :

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

#### Nouvelles Metriques Affichees

L'email HTML enrichi presente desormais 5 metriques cles au lieu de 4 :

1. **Indice Kp Actuel** : Valeur en temps reel de l'activite geomagnetique (sur 9)
2. **Kp Minimum Requis** : Seuil calcule specifiquement pour cette localisation (NOUVEAU)
3. **Score de Probabilite** : Score composite prenant en compte Kp, meteo et obscurite (sur 1.0)
4. **Ciel Degage** : Pourcentage de ciel sans nuages
5. **Obscurite** : Indicateur nuit/jour avec icone appropriee

### 2.5.3 Structure de l'Email Ameliore

#### Format HTML Professionnel

L'email utilise un template HTML responsive avec :

- **En-tete dynamique** : Couleur de fond adaptee 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 personnalise

- **Metriques visuelles** : Presentation en grille des 5 indicateurs cles avec typo graphie hierarchisee
- **Conseils d'observation** : Liste detaillee (meilleure periode, lieu ideal, direction, patience, conseils photo)
- **Bouton d'action** : Lien cliquable vers le dashboard complet
- **Footer informatif** : Timestamp UTC precis et informations sur l'origine de l'alerte

### Version Texte Alternative

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

## 2.6 Tests et Validation du Systeme d'Alertes

### 2.6.1 Creation d'Outils de Test Dedies

Pour faciliter les tests du systeme d'alertes email sans necessiter l'activation complete du dashboard, un script Python dedie a ete developpe : `test_email.py`

#### Fonctionnalites 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 donnees de test predefinies (Kp 6.5, Stockholm, score 0.82)
- Affichage de messages de debug detailles pour faciliter le diagnostic
- Fourniture automatique de solutions en cas d'erreur d'envoi

### 2.6.2 Methodes de Test Documentees

Trois approches complementaires de test ont ete documentees :

Methodes	Description et Usage
Script <code>test_email.py</code>	Script Python standalone avec interface interactive, ideal pour tests rapides et debugging
Test depuis Dashboard	Modification temporaire du seuil Kp a 0.0 pour forcer l'envoi immediat 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 – Methodes de test disponibles

### 2.6.3 Guide de Configuration Gmail

#### Processus de Configuration Documente

Un guide detaille a ete cree pour configurer Gmail avec mot de passe d'application :

1. Acceder aux parametres de securite du compte Google
2. Activer l'authentification a deux facteurs (prerequis obligatoire)
3. Generer un mot de passe d'application specifique pour "AurorAlerte"
4. Copier le mot de passe genere (16 caracteres) dans le fichier de configuration
5. Tester la connexion avec le script de test

**Note Importante de Securite :**

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 securite (le mot de passe peut etre revoque sans changer le mot de passe principal)
- Tracabilite des acces
- Conformite avec les recommandations de securite de Google

## 2.6.4 Documentation des Problemes Courants

Creation d'un guide de resolution des erreurs frequemment rencontrees :

Erreur Rencontree	Solution Documentee
Authentification echouee	Verifier l'utilisation d'un mot de passe d'application et l'activation de l'authentification a deux facteurs
Connection refused / Timeout	Desactiver temporairement l'antivirus, autoriser Python dans le pare-feu, verifier que le port 587 n'est pas bloque
Invalid sender	Confirmer que l'adresse email expéditeur correspond exactement au compte Gmail utilise
Email non reçu	Attendre 5-10 minutes, verifier minutieusement le dossier SPAM/Courrier indesirable

TABLE 2.6 – Guide de resolution des problemes

## 2.7 Livrables de la Session

### 2.7.1 Fichiers de Developpement Crees

Au total, 11 fichiers ont ete crees pendant cette session :

Fichier	Contenu et Utilite
carte_mercator.py	Premiere version corrige (900px)
carte_mercator_1200px.py	Version experimentale extra large
carte_hemisphere_nord.py	Version finale optimisee recommandee
carte_avec_recherche_villes.py	Carte complete avec systeme de recherche
alertes_avec_validation.py	Systeme d'alertes avec validation email
alertes_automatiques.py	Version finale automatisee complete
alerts_ameliore.py	Fichier model/alerts.py enrichi
kp_zones_definition.py	Definition globale de la table Kp-Latitude
test_email.py	Script de test du systeme d'alertes

TABLE 2.7 – Fichiers Python crees

### 2.7.2 Documentation Technique Produite

Document	Contenu
INTEGRATION_ALERTES_AUTO.md	Guide complet d'integration du systeme automatisé (40+ pages)
GUIDE_TEST_EMAIL.md	Documentation complete des tests et resolution de problemes (25+ pages)

TABLE 2.8 – Documentation creee

## 2.8 Statistiques de la Session

Metrique	Valeur
Duree totale de la session	environ 6 heures
Problemes majeurs resolus	4
Nouvelles fonctions creees	3
Lignes de code ajoutees/modifiees	environ 500
Fichiers crees (code + docs)	11
Captures d'ecran analysees	3
Versions de carte testees	3
Types de messages utilisateur	8
Pages de documentation	65+

TABLE 2.9 – Statistiques detaillees 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 donnees
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 utilisees

### 3.2 APIs Integrees

- NOAA Space Weather Prediction Center (Kp, previsions, animations OVATION)
- Open-Meteo (meteo, geocodage)
- OpenWeatherMap (meteo actuelle, icones)
- Sunrise-Sunset (obscurite)

## Chapitre 4

# Fonctionnalites Implementees

- 3 jauges interactives temps reel
- Historique Kp 4 heures avec export CSV
- Carte mondiale Mercator focalisee hemisphere nord (40-85 degrees N)
- Recherche de 5 villes personnalisees + detection automatique des doublons
- Calcul automatique du seuil Kp selon la localisation
- Messages contextuels personnalises (4 niveaux)
- Emails HTML enrichis avec 5 metriques
- Documentation technique complete (65+ pages)



## Chapitre 5

# Tests et Validation

### 5.1 Tests Fonctionnels

Test Effectue	Resultat
Traduction complete francais	Passe
4 APIs fonctionnelles	Passe
Geocodage francais (Suede, Norvege)	Passe
Alertes email (test force)	Passe
Carte zones Kp 3, 5, 7, 9	Passe
Recherche villes + doublons	Passe
Calcul Kp auto (3 villes testees)	Passe
Export CSV	Passe
Responsive 2 resolutions	Passe
Gestion erreurs	Passe

TABLE 5.1 – Resultats 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
- Predictions ML, Application mobile, Partage social

## Chapitre 7

# Conclusion

### 7.1 Bilan des Objectifs

Le projet AurorAlerte a depasse ses objectifs initiaux en creant un dashboard professionnel combinant :

- 4 APIs temps reel parfaitement integrees
- Traduction francaise complete et localisee
- 15+ emplacements pedagogiques detailles
- Systeme alertes intelligent automatise
- Carte mondiale avec recherche dynamique
- Export donnees multiples formats
- 9 problemes techniques majeurs documentes et resolus
- Guide test complet (65+ pages documentation)

### 7.2 Competences Acquises

**Competences Techniques :** Python/Streamlit avance, APIs REST multiples, Plotly visualisation geographique, SMTP/Gmail securise, Pandas manipulation donnees, Gestion etats Streamlit, Projections cartographiques, Geocodage multilingue

**Methodologie Professionnelle :** Debugging systematique, Documentation technique exhaustive, Gestion version Git, Architecture logicielle modulaire, Tests fonctionnels, Scripts test autonomes, Resolution problemes utilisateur

**Soft Skills Developpes :** Adaptation code existant, Traduction/localisation culturelle, Pedagogie scientifique, Gestion projet academique, UX design centre utilisateur, Feedback contextuel, Communication technique

### 7.3 Reflexion Finale

Ce projet s'est etendu sur plusieurs semaines avec une session intensive finale le 29 novembre 2025. L'approche choisie d'adapter un code existant plutot que repartir de zero s'est revelee extremement formatrice, refletant fidelement les pratiques professionnelles reelles.

La session du 29 novembre a particulièrement demontré l'importance d'une methodologie rigoureuse : analyse detaillee des problemes avec documentation visuelle, exploration de solutions multiples avant decision, tests systematiques a chaque etape, et documentation exhaustive pour reproductibilite.

L'aspect le plus gratifiant du projet a ete sa dimension pedagogique : transformer des concepts scientifiques complexes (geomagnetisme, meteorologie, projections geographiques) en experience utilisateur accessible et engageante.

La refonte du systeme d'alertes illustre parfaitement le principe central du design centre utilisateur : le systeme doit s'adapter intelligemment a l'utilisateur, jamais l'inverse.

**AurorAlerte** represente desormais un outil operationnel et pedagogique complet, pret au deploiement pour passionnes d'aurores boreales francophones.

Le code source documente, le README detaille et ce journal de bord exhaustif permettent a d'autres developpeurs de contribuer et d'etendre le projet.

L'application demontre qu'il est possible de creer des outils scientifiques qui soient simultanement accessibles, esthetiques, rigoureux et pedagogiques en combinant APIs publiques, frameworks modernes et attention meticuleuse au design d'experience utilisateur.

## Annexe A

# Statistiques Finales du Projet

Metrique	Valeur
Lignes de code Python	environ 2000
Lignes de documentation	environ 2500
Fonctions developpees	30+
APIs integrees	4
Langues supportees	2 (FR + EN pays)
Onglets interface	7
Visualisations Plotly	20+
Temps developpement total	environ 50 heures
Commits Git	40+
Problemes techniques resolus	9
Tests fonctionnels documentes	10
Pages documentation technique	65+

TABLE A.1 – Statistiques completes (mise a jour 29/11/2025)

## Annexe B

# Ressources et Liens

- Projet original : <https://github.com/sheilageorge/aurora-monitor>
- Version adaptee AurorAlerte : [votre repository GitHub]
- 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 realise par :**

Adjimon Jerome Vitoffodji et Alvin Ingabire

Master 2 Open Data et Web des Donnees