# Documentation du Projet - Application Météo Vocale

## Contexte du Projet

L'objectif est de développer une application météo permettant une interaction fluide et intuitive grâce à la reconnaissance vocale via Azure Speech-to-Text. L’expérience utilisateur (UX) doit être ergonomique et ludique, intégrant des animations et une interface accessible.

## Problèmes et Solutions Associées

Voici les principaux défis et leurs solutions techniques adaptées pour assurer fiabilité et performance.

## Tableau des Problèmes et Solutions

|  |  |  |
| --- | --- | --- |
| Problème | Solution Technique | Impact et Bénéfices |
| Reconnaissance vocale imprécise | Azure Speech-to-Text + fallback vers Whisper AI | Améliore la précision en environnement bruyant |
| Identification du lieu et horizon flous | NLP avec Azure LUIS | Meilleure extraction des intentions |
| Latence élevée des réponses | Mise en cache Redis | Réduction du temps de réponse |
| Interface peu engageante | Carte interactive Leaflet.js et animations CSS | Améliore l'interaction et l'ergonomie |
| Sécurité des données | Authentification JWT et validation des entrées | Protection contre les attaques courantes |

## Conception Ludique et Accessible

Pour rendre l'application plus interactive et engageante, plusieurs éléments seront intégrés :  
1. Animation vocale : Affichage d'une onde sonore dynamique.  
2. Carte interactive : Zoom dynamique sur les prévisions météo.  
3. Thème sombre et clair pour le confort visuel.  
4. Visualisation intuitive des prévisions avec graphiques dynamiques.  
5. Avatar assistant IA pour une meilleure accessibilité.

## Feuille de Route et Calendrier des Tâches

|  |  |  |  |
| --- | --- | --- | --- |
| Jour | Tâche | Technologies utilisées | Objectif clé |
| Jour 1-2 | Conception de l'architecture | FastAPI, Azure, Streamlit, SQLAlchemy | Définir les interactions backend/frontend |
| Jour 3-4 | Intégration de la reconnaissance vocale | Azure Speech-to-Text | Transcrire les requêtes vocales |
| Jour 5 | Connexion API météo | OpenWeatherMap API | Obtenir les prévisions météo |
| Jour 6 | Stockage en base de données | Azure SQL, ORM SQLAlchemy | Gérer les requêtes utilisateurs |
| Jour 7-8 | Sécurisation (Top 10 OWASP) | JWT, HTTPS, validation | Protection contre les attaques |
| Jour 9 | Développement UI interactive | Streamlit, TailwindCSS, Leaflet.js | Créer une interface attrayante |
| Jour 10-11 | Visualisation des données | Plotly, Chart.js | Graphiques interactifs et cartes météo |
| Jour 12 | Dockerisation et CI/CD | Docker, GitHub Actions, Azure App Services | Déploiement automatisé |
| Jour 13 | Tests d'intégration et corrections | PyTest, Selenium | Vérifier le bon fonctionnement |
| Jour 14 | Démo finale et livraison | GitHub, Streamlit Cloud | Finalisation du projet |

## Conclusion

L'approche proposée met l'accent sur la fiabilité, l'ergonomie et la sécurité. Grâce à une intégration fluide des services Azure, un backend performant et un frontend interactif, cette application offrira une expérience utilisateur optimisée et agréable.