

Rapport Individuel — TerraSafe AI (Oil & Gas)

Rôle : Tech Lead Agent & Backend

Nom : Mohamed Abdellahi Sidiya Filière : DAI (L3)

5 février 2026

1 Contexte et objectif

Le projet TerraSafe AI est une application web qui évalue, explique et anticipe les risques environnementaux et sanitaires liés à la qualité de l'eau et du sol dans des zones influencées par le pétrole et le gaz. Le MVP doit permettre : (1) saisie de mesures, (2) calcul d'un score de risque, (3) carte & dashboard, (4) génération de rapport via agents numériques.

2 Ma mission (livrables)

Back-end & API

- Concevoir l'API REST (Flask) : Zones, Mesures (Samples), Scores (RiskResult).
- Mettre en place la base SQLite et les modèles (Zone, Sample, RiskResult).
- Implémenter le moteur de scoring côté serveur : appel à `risk_engine.py`.
- Mettre en place la sécurité minimale : validation d'entrées, logs, gestion d'erreurs.

Agents numériques (workflow)

- Agent Analyse : calculer C, B, G, P puis $R = w_1C + w_2B + w_3G + w_4P$.
- Agent Reporting : générer un rapport (résumé + technique) à partir d'un JSON standard.
- Agent Alerte : règles de seuil + notifications (MVP : bannière/icône).

3 Plan Agile personnel (4 sprints)

- Sprint 0 : schéma DB + routes API + dataset simulé minimal.
- Sprint 1 : CRUD complet + calcul R v1 + endpoints `/risk/calc`.
- Sprint 2 : historique + endpoints d'agrégation (moyenne par zone, tendance).
- Sprint 3 : intégration agent reporting (prompt + format JSON stable) + doc API.

4 Prompts ChatGPT à utiliser (méthode RC-TC)

Chaque prompt suit : Rôle / Contexte / Tâche / Contraintes. Je colle le résultat dans le repo.

Prompt 1 — Spécification API

Rôle : Agis comme architecte backend senior (Flask).

Contexte : Je développe TerraSafe AI. Entités : Zone, Sample, RiskResult.

Tâche : Propose une API REST complète (routes, méthodes, payload JSON).

Contraintes : Réponds en JSON avec champs {route, method, request, response, errors}.

Pas de blabla.

Prompt 2 — Schéma DB

Rôle : Ingénieur base de données.

Contexte : SQLite MVP TerraSafe AI.

Tâche : Définir tables, types, clés, index.

Contraintes : Fournis DDL SQL complet + justification courte (3 lignes max).

Prompt 3 — Gestion erreurs & validation

Rôle : Ingénieur qualité logiciel.

Contexte : API reçoit ph, conductivite, turbidite, nitrates_hc, bio_risk_level, proximity_index.

Tâche : Proposer règles de validation + messages d'erreur.

Contraintes : Format YAML. Inclure bornes, type, et exemples invalides.

Prompt 4 — Format JSON pour Agent Reporting

Rôle : Prompt engineer.

Contexte : Agent Reporting doit générer rapport à partir d'un JSON stable.

Tâche : Définir un schéma JSON minimal pour rapport (titre, zone, scores, risques, recommandations, limites).

Contraintes : Donne JSON Schema + 1 exemple de JSON rempli.

Prompt 5 — Tests unitaires

Rôle : Développeur Python senior.

Contexte : Module risk_engine.py calcule C,B,G,P,R.

Tâche : Proposer une suite de tests (pytest) avec cas sain/industriel/haut risque.

Contraintes : Donne code pytest complet. Si incertain, propose comment vérifier.

5 Compétence MOOC à obtenir

Plateforme : Coursera

Cours : Prompt Engineering for ChatGPT (Vanderbilt)

Lien : <https://www.coursera.org/learn/prompt-engineering>

Compétences visées

- Écrire des prompts structurés et robustes.
- Réduire les hallucinations via contraintes, formats, vérifications.
- Construire des “applications à prompts” (reporting, extraction, synthèse).

6 Preuves (e-réputation)

- Repo GitHub : API + risk_engine + tests + README.
- Capture d'écran Postman / Swagger.
- Certificat MOOC + résumé (1 page) dans le repo.

7 Risques et garde-fous

- L’agent ne doit pas agir sans validation humaine (risques santé).
- Sources : si une recommandation est incertaine, l’agent doit le dire.
- Pas de données personnelles sensibles dans le MVP.