

Instructions pour le test technique - Responsable Technique

Data, Stratégie et Intégrations

Vue d'ensemble

Vous allez créer un système simplifié de gestion des points d'eau qui reflète les défis réels d'Uduma. Ce test évalue vos compétences techniques, votre réflexion stratégique et votre capacité à travailler de manière autonome.

Durée: test à compléter d'ici **lundi 18 août 12h GMT**

Présentation: 15 minutes lors de l'entretien

Usage IA: Autorisé avec transparence (voir directives ci-dessous)

Votre mission:

1. **Calculer** la consommation mensuelle pour chaque point d'eau
2. **Analyser** la cohérence entre consommation et revenus
3. **Identifier** les anomalies dans les relevés de compteurs
4. **Concevoir** une base de données pour stocker ces informations
5. **Créer** des KPIs pertinents basés sur ces calculs et les visualiser

Description des [données test](#) à utiliser

Vous trouverez un fichier CSV contenant les relevés de compteurs d'eau collectés dans la région de Sikasso au Mali (données exemple, non réelles). Ces données incluent des informations sur des points d'eau et leurs relevés mensuels.

- **Région:** Sikasso et environs, Mali
 - **Période:** Relevés mensuels janvier à juillet 2024
 - **Index compteurs:** En m3 cumulés depuis installation
 - **Tarification:** Variable selon le type d'installation
 - **Anomalies présentes**
-

Le défi

Contexte

Uduma gère des points d'eau dans la région de Sikasso au Mali. Actuellement, les données proviennent de rapports formulaires mobiles. Votre mission est de créer un système démo avec une base de données, des endpoints API et une page pour visualiser les KPIs commerciaux importants.

Partie 1: Base de données & API

Exigences techniques

1. **Technologies:**
 - **Backend:** Python OU JavaScript (framework libre, choix à expliquer)
 - **Frontend:** Python OU JavaScript OU R OU third party service (choix à expliquer)
 - **Base de données:** PostgreSQL ou SQLite
 - **Vous devez utiliser les DEUX langages d'une façon ou d'une autre** (Python ET JavaScript)
2. **Structure du système:**
 - Créer une base de données avec les tables appropriées pour stocker les données fournies (2 modèles minimum, par exemple: water points, meter_readings, consumption etc)
 - Développer une API REST avec au minimum 2 endpoints
 - Construire une interface web qui consomme vos API endpoints
3. **Endpoints API suggérés:**
 - **GET /water_points** - liste tous les points d'eau
 - **GET /water_points/{id}/consumption** - consommation pour un point spécifique
 - **GET /consumption** - données de consommation avec filtres optionnels
4. **Données fournies:**
 - Utilisez les données partagées
 - À vous de définir la structure de base de données optimale
 - À vous d'importer et organiser ces données

Partie 2: Tableau de bord web analytique

Exigences

Vous êtes libre de développer vous même une interface pour la visualisation si vous le souhaitez (Python, Javascript ou R) ou bien d'utiliser un service existant. Vous justifierez votre choix durant la présentation.

1. **Page unique**
2. **2 visualisations des KPIs de votre choix**, par exemple:
 - Consommation moyenne mensuelle par point d'eau, commune
 - Revenus totaux par mois
 - Relevés par employé(e)
3. **Fonctionnalités** (au choix, indicatif):
 - Liste/tableau des points d'eau
 - Graphique simple

- Filtrage basique (par commune, statut, ou période)
 - 4. **Connexion à votre API** pour récupérer les vraies données ou à la base de données
-

Directives pour l'usage de l'IA: transparence obligatoire

Si vous utilisez l'IA pour cet exercice, veuillez le documenter:

- Quelles tâches ont bénéficié d'assistance IA
- Quelles questions/prompts vous avez utilisés
- Comment vous avez validé/modifié le code généré
- Ce que vous avez construit entièrement vous-même

Structure de présentation (ligne indicative mais non stricte) (15 minutes)

Partie 1: Base de données & API (6 minutes)

- Structure des tables, architecture du système
- Démonstration des endpoints API
- Import des données d'exemple et manipulation effectuée

Partie 2: Démonstration dashboard (6 minutes)

- Fonctionnalités du tableau de bord et choix du stack
- Logique de calcul des KPIs
- Fonctionnalités interactives

Partie 3: IA & questions (3 minutes)

- Usage IA
- Questions et discussion technique

Critères d'évaluation

Excellence technique

- **Qualité du code:** Propre, lisible, maintenable
- **Architecture:** Technologies appropriées, sécurisé
- **Fonctionnalité:** Système qui fonctionne de bout en bout
- **Base de données:** Conception appropriée, relations correctes
- **Évaluation des risques:** Identifie les problèmes potentiels

Communication

- **Documentation:** Claire, complète, professionnelle
- **Présentation:** Bien structurée, gestion du temps
- **Intégration IA:** Usage réfléchi, transparence honnête
- **Gestion des questions:** Démonstre une compréhension approfondie

Soumission de l'exercice

Repository GitHub / Bitbucket / Gitlab (au choix)

- Lien du repository avec README clair
- Historique des commits montrant votre processus
- Inclure le log d'usage IA
- Lien si projet déployé

Documentation

- Instructions d'installation/configuration
- Décisions architecturales
- Limitations connues et améliorations futures

Échéance

Soumettre le lien repository et la documentation avant lundi 18 août 12h GMT.

Ce test reflète les défis réels que notre Responsable Technique rencontrera. Nous avons hâte de voir votre approche et de discuter vos solutions lors de l'entretien!