# 1. Présentation du projet

**Nom du projet** : Réalisation d’un Data Lake interopérables et une Application Web de Visualisation

**Objectif** : Centralisation, traitement et visualisation de données multi-sources (type santé), avec visualisation sur application web (Next.js + D3.js) connectée à un backend Data Lake (Spark, Hive, Mongo, HBase).

**Situation actuelle** : Actuellement, l’entreprise ne dispose que de 2 bases PostgreSQL hébergeant des données de santé. Cependant, l’ambition du projet est de construire une architecture capable de s’adapter à l’avenir : jusqu’à 100 sources différentes, de types variés (PostgreSQL, fichiers Excel, CSV, etc.)

# 2. Périmètre du projet

Le projet est organisé en **5 phases** principales :

## **1. Préparation & Installation**

**Objectifs : Poser les bases du projet (outils, apprentissage, environnement).**

Tâches :

* Prise en main du projet.
* Apprentissage des technologies Next.js, D3.js.
* Étude de l’architecture d’un Data Lake.
* Étude des types de réseaux et protocoles de sécurité.
* Conception du schéma d’architecture réseau (interconnexion établissements → Data Lake).
* Création de VM avec Vagrant.
* Installation des environnements (Hadoop, Hive, MongoDB, Spark, etc.).
* Vérification de l’interopérabilité entre les outils installés.

## **2. Backend – Construction du Data Lake**

Objectifs : Mettre en place un pipeline d’ELT et une architecture modulaire.

Tâches :

* Définition du schéma logique / schéma pivot.
* Développement Spark pour :
* Connexion aux bases sources.
* Ingestion des données vers HDFS (Zone RAW).
* Logging et gestion des erreurs.
* Nettoyage des données :
  + Suppression de doublons, gestion des formats.
  + Normalisation des valeurs.
* Transformation des données selon le schéma pivot basé sur des standards (**FHIR**)
* Création des tables Hive.
* Intégration de MongoDB et HBase (accès rapide + historique).
* Génération d’UUID pour l’identification unique des patients.
* Exposition des données via API (Hive, MongoDB, HBase).

## **3. Frontend – Next.js + D3.js**

Objectifs : Concevoir une interface utilisateur pour la visualisation des données.

Tâches :

* Initialisation du projet avec Next.js.
* Mise en place de l’authentification (NextAuth + JWT).
* CRUD utilisateurs (gestion des comptes).
* Connexion aux API backend (Hive, MongoDB, HBase).
* Développement de 4 graphiques dynamiques avec D3.js pour cette première version.
  + A définies à partir des indicateurs issus du Rapport Mensuel d’Activité (RMA)

## **4. Tests & Validation**

Objectifs : Vérifier le bon fonctionnement et la cohérence du système.

Tâches :

* Construction de bases de données fictives pour test.
* Vérification de la cohérence des données (UUID, transformation, agrégation...).
* Tests d'interfaces et d'API (Next.js → Hive/Mongo).
* Corrections des erreurs identifiées (backend + frontend).

## **5. Déploiement & Livraison**

Objectifs : Préparer une version stable et documentée du système.

Tâches :

* Exportation de la VM Vagrant au format. box.
* Test d'installation sur un autre poste.
* Déploiement de l’application web.
* Rédaction de la documentation technique et utilisateur.
* Formation utilisateur.

# 3. Livrables attendus

* VM Vagrant exportable contenant l’environnement complet.
* Code source du backend (ETL Spark, API).
* Code source du frontend (Next.js + D3.js).
* Documentation technique (installation, structure, API).
* Documentation utilisateur (guide de l’interface).
* Rapport de tests et corrections.
* Graphiques interactifs dans l’application.

# 4. Technologies utilisées

| **Domaine** | **Technologies** |
| --- | --- |
| Frontend | Next.js, D3.js, NextAuth, Tailwind CSS |
| Backend | Apache Spark, Hive, MongoDB, HBase, API REST |
| Stockage | HDFS, Hive Tables |
| Virtualisation | Vagrant, VirtualBox |
| Langages | Typescript, Python, Scala (Spark), Shell |
| Sécurité | JWT, Protocoles sécurisés réseau |

5. Contraintes & Sécurité

* Respect des normes RGPD en matière de traitement des données de santé.
* Interopérabilité entre les outils (Spark, Hive, MongoDB...).
* Sécurisation des accès via tokens (JWT) et protocoles réseau (VPN/SSL)
* Génération d’identifiants anonymes (UUID) pour les patients.

6. Planning prévisionnel

| **Phase** | **Durée estimée** |
| --- | --- |
| Préparation & Installation | 2 semaines |
| Backend | 3 semaines |
| Frontend | 2 semaines |
| Tests & Validation | 1 semaine |
| Livraison & Documentation | 1 semaine |

7. Équipe projet

**Chef de projet** : Mr. Rabemanoela Harena

**Développeur Backend** : Ramanantsafidy Mariella Jonah Fitia

**Développeur Frontend** : Ramanantsafidy Mariella Jonah Fitia

**Administrateur Systèmes / VM** : Ramanantsafidy Mariella Jonah Fitia