

Intégration des Données Agricoles avec Interopérabilité Sémantique



Module : Data spaces

**Préparé par (Groupe 9) : El Meziane Chaima & Ohmad
Abdessamade & Adnane Riyadi & Yasmina Hachhouch &
Othmane Fahmi**

Supervisé par : PR. Ouafae Baida

Introduction

Dans un contexte où la digitalisation de l'agriculture devient essentielle pour améliorer la productivité et la durabilité, l'intégration des données issues de différentes sources représente un défi majeur. Ce projet vise à concevoir une plateforme de gestion des données agricoles, permettant d'assurer l'interopérabilité entre divers systèmes et applications.

L'objectif est de créer un environnement où les données provenant de capteurs, bases de données et API externes peuvent être intégrées, harmonisées et exploitées de manière efficace grâce à des standards sémantiques. À travers une architecture bien définie et l'utilisation de technologies adaptées, cette plateforme offrira une vue unifiée des données agricoles, facilitant ainsi la prise de décision pour les acteurs du secteur.

Ce rapport détaille les différentes étapes du projet, depuis l'analyse des besoins jusqu'au développement et à la mise en œuvre de la solution, en mettant l'accent sur les défis rencontrés et les choix technologiques effectués.

Objectifs Principaux

Ce projet vise à répondre aux défis d'intégration et d'interopérabilité des données dans le domaine agricole en mettant en place une plateforme innovante. Les principaux objectifs sont :

- **Créer un écosystème numérique unifié** permettant de centraliser, structurer et gérer efficacement les données agricoles issues de diverses sources (capteurs, bases de données, API, fichiers, etc.).
- **Assurer l'interopérabilité sémantique des données** en adoptant des standards sémantiques et des ontologies, facilitant ainsi l'échange et l'utilisation cohérente des informations entre différents systèmes.
- **Améliorer la prise de décision basée sur les données**, en offrant des analyses pertinentes et des outils d'aide à la décision pour les agriculteurs, chercheurs et autres parties prenantes.
- **Renforcer la coordination entre les acteurs du secteur agricole**, en favorisant un accès simplifié et structuré aux informations, améliorant ainsi la collaboration et l'efficacité des interventions.

Cette approche vise à optimiser la gestion des ressources agricoles, accroître la productivité et contribuer à une agriculture plus intelligente et durable.

1. Analyse des Besoins

1.1 Identification des Parties Prenantes

- Ministère de l'Agriculture
- Agriculteurs et coopératives agricoles
- Instituts de recherche agronomique
- Fournisseurs de services agricoles
- Organismes de financement agricole

Acteur	Données
Agence du Bassin Hydraulique de Souss Massa	barrage_souss_massa.csv
Analystes	Accès à toutes les données
Chercheurs en climatologie	Temperature_change_on_land.csv
Département de l'Agriculture	cheptel_2010_2021.csv
Agriculteurs	Producing_AnimalsSlaughtered.csv
Investisseurs	Credit_to_Agriculture.csv
Ministère de l'Emploi	Agriculture_and_agrifood_systems.csv

1.2 Fonctionnalités Principales

Intégration des données

- **Collection automatisée** : Acquisition des données à partir de diverses sources (capteurs, bases de données, API, fichiers, etc.).
- **Transformation standardisée** : Normalisation et structuration des données selon des formats unifiés pour assurer leur cohérence.
- **Enrichissement sémantique** : Utilisation d'ontologies et de standards sémantiques pour améliorer l'interprétation et l'exploitation des données.

Visualisation des données

- **Tableaux de bord interactifs** : Affichage dynamique des données via des graphiques et indicateurs clés pour faciliter l'analyse.
- **Rapports personnalisables** : Génération de rapports adaptés aux besoins des utilisateurs, avec des filtres et options d'exportation.

Gestion des accès

- **Authentification sécurisée** : Mise en place de mécanismes d'authentification robustes pour garantir la protection des données.
- **Contrôle d'accès basé sur les rôles** : Attribution de permissions spécifiques en fonction des profils utilisateurs pour une gestion fine des accès.

Analyse des données

- **Analyses croisées** : Exploration et comparaison de différentes sources de données pour identifier des corrélations et tendances.
- **Suivi des tendances** : Surveillance des évolutions des données au fil du temps afin d'anticiper les besoins et améliorer la prise de décision.

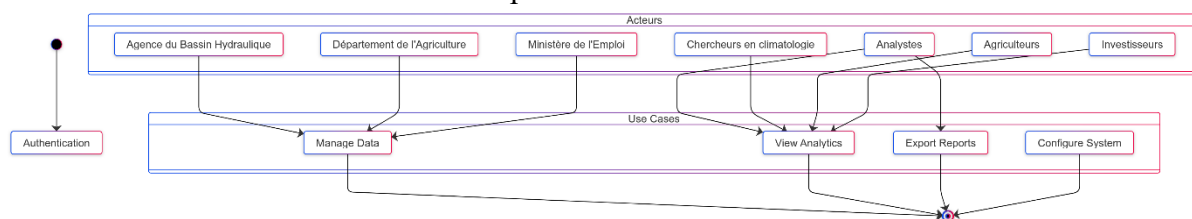
2. Conception de l'Architecture

2.1 Sources de Données

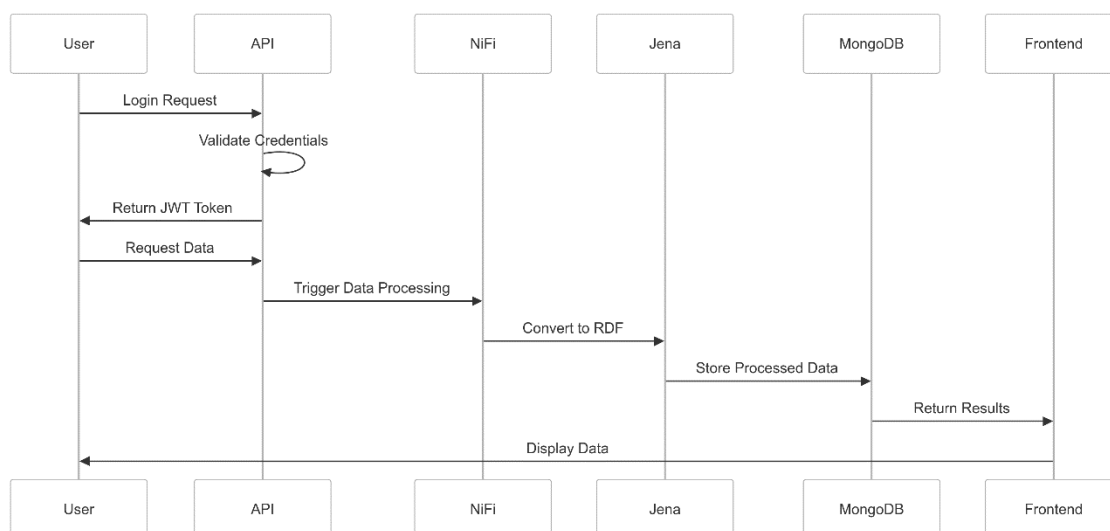
- **FAO (Food and Agriculture Organization)** : Données standardisées et fiables via FAOSTAT.
- **data.gov.ma** : Portail National des données ouvertes du Maroc.
- **Capteurs environnementaux** : Mesures en temps réel sur l'humidité, la température, etc.

2.2 Conception

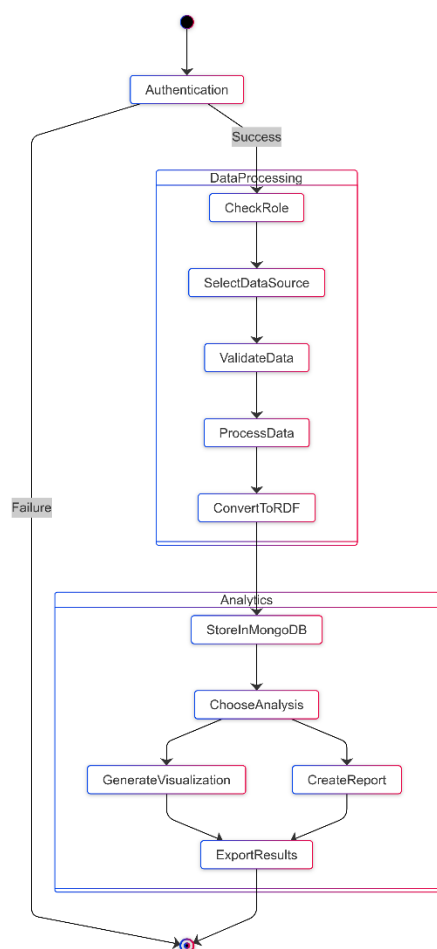
Les cas d'utilisation décrivent les interactions entre les utilisateurs et le système, permettant de définir les fonctionnalités clés et les processus métiers.



Les diagrammes de séquence représentent les interactions entre les composants du système au fil du temps.



Les diagrammes d'activité montrent les différentes étapes et processus dans le système.



2.3 Technologies Utilisées

L'application est déployée dans un environnement **Docker**, assurant une gestion efficace des conteneurs et une portabilité optimale. Les principales technologies utilisées sont :

- **Apache NiFi** : Orchestration et gestion des flux de données, permettant l'extraction, la transformation et le chargement (ETL) des données provenant de diverses sources.
- **Apache Jena** : Traitement et gestion des données RDF, facilitant l'interopérabilité sémantique et l'exploitation des ontologies.
- **MongoDB** : Base de données NoSQL utilisée pour stocker et organiser les données structurées et semi-structurées de manière flexible et évolutive.
- **Express.js** : Framework backend léger pour la création d'une API RESTful, permettant l'interaction entre la plateforme et les données intégrées.
- **React.js** : Bibliothèque front-end permettant de concevoir une interface utilisateur dynamique, interactive et réactive pour l'affichage et la manipulation des données.

Cette architecture modulaire et évolutive garantit une intégration fluide des différentes technologies et une interopérabilité optimale des données agricoles.

2.4 Architecture Sémantique

L'architecture sémantique de l'application repose sur des principes fondamentaux permettant d'assurer l'interopérabilité et la cohérence des données. Les éléments clés de cette architecture sont :

- **Utilisation de RDF (Resource Description Framework)** : Le modèle RDF est employé pour la représentation des données sémantiques, permettant de structurer l'information sous forme de triplets (sujet, prédicat, objet) et assurant une meilleure interprétation des relations entre les entités.
- **Définition d'une ontologie** : Une ontologie spécifique au domaine agricole est définie pour standardiser les formats de données et faciliter l'échange d'informations entre les systèmes. Cela permet de représenter les concepts, les relations et les règles de gestion des données de manière unifiée.
- **Exploitation de SPARQL** : Le langage de requêtes SPARQL est utilisé pour interroger et manipuler les données RDF, offrant ainsi des capacités avancées de recherche et d'analyse sémantique sur les données intégrées.

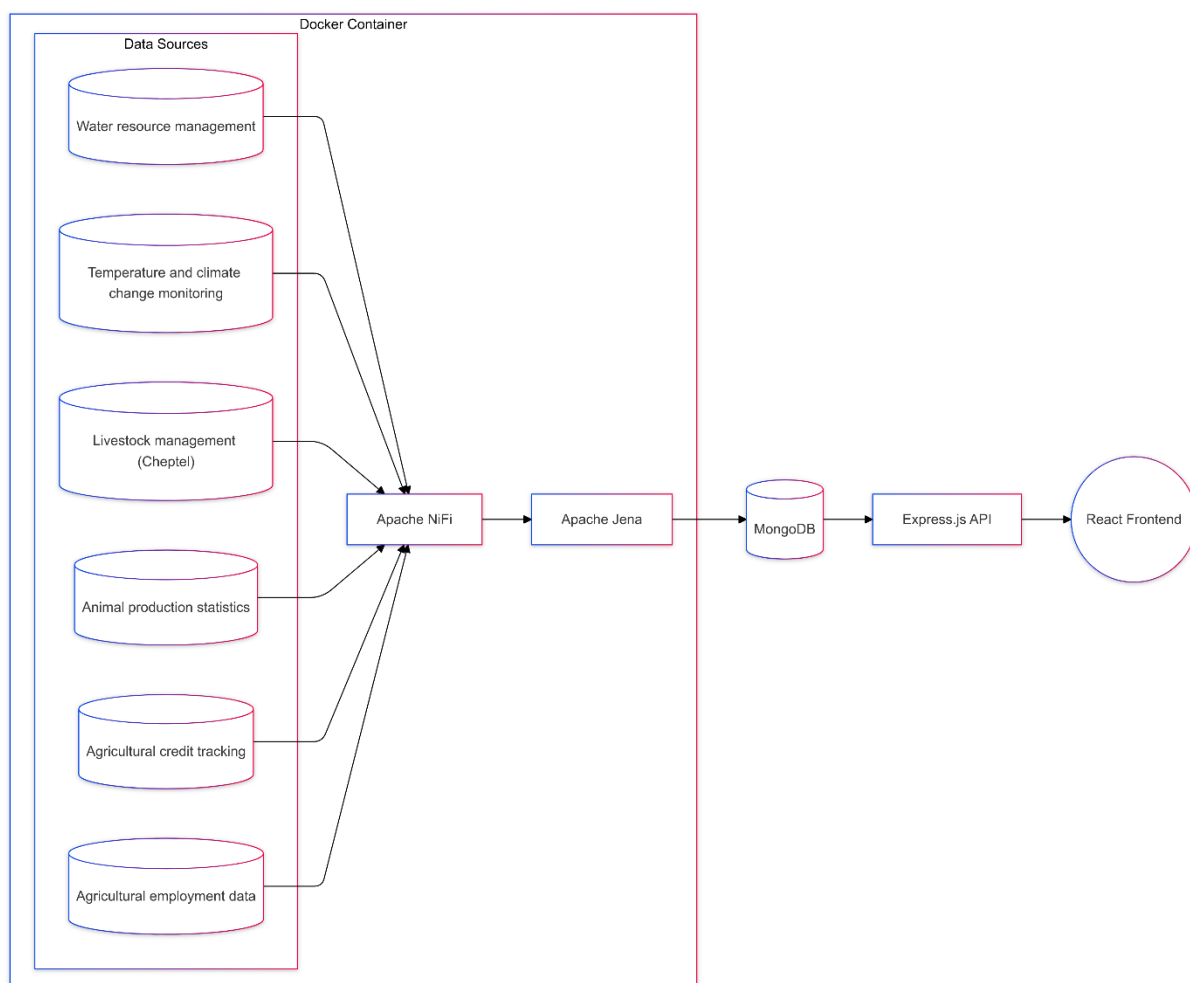
Cette architecture garantit une gestion optimale des données, favorise l'interopérabilité entre les systèmes et permet une meilleure exploitation des informations agricoles.

2.5 Architecture Globale du Système

L'architecture globale du système de gestion des données agricoles repose sur une approche modulaire et évolutive, assurant l'interopérabilité des données entre différents systèmes et technologies. Elle intègre les différentes sources de données, les processus de transformation, ainsi que les couches de stockage et d'analyse. Le système est conçu pour permettre une collecte, un traitement et une gestion efficaces des données agricoles, tout en garantissant une interopérabilité sémantique grâce à l'utilisation d'ontologies et de standards ouverts.

Diagramme de l'Architecture Globale

Voici un diagramme illustrant l'architecture globale du système, mettant en évidence les principaux composants et leur interaction :



3. Implémentation

3.1 Environnement Docker

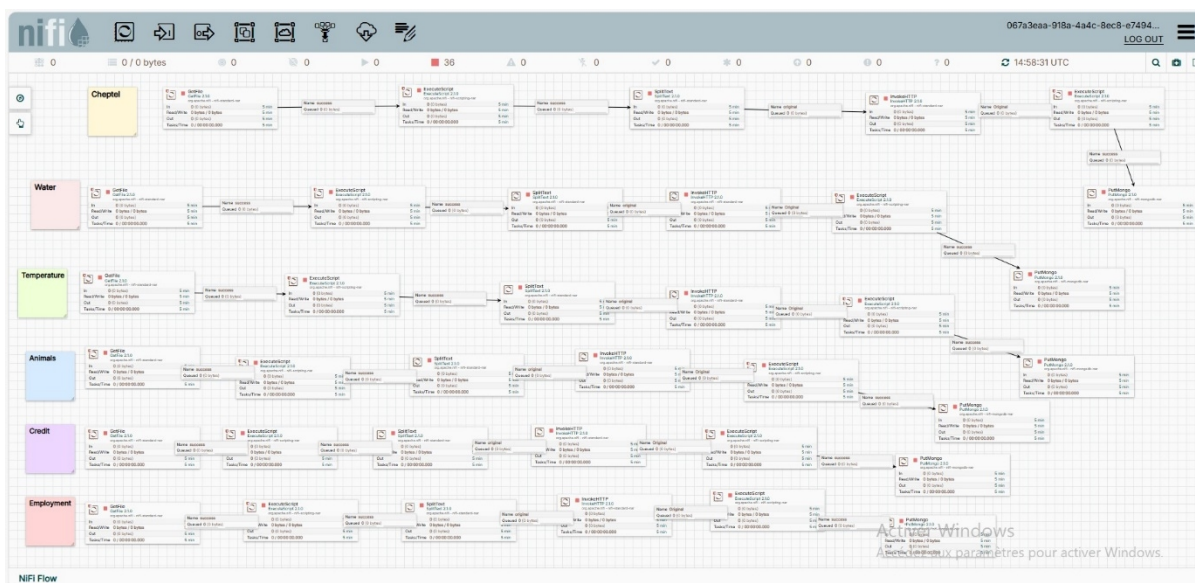
L'ensemble de l'application est déployé dans un environnement **Docker**, assurant la conteneurisation des différents composants et facilitant leur gestion et orchestration. Les sources de données, **Apache NiFi**, **Apache Jena**, **MongoDB**, et d'autres services nécessaires sont tous intégrés dans des conteneurs Docker distincts, garantissant ainsi une gestion cohérente, reproductible et évolutive des données.

<input type="checkbox"/>	<div><div></div><div></div><div></div></div> <div><div>dataspaces</div><div>Running (2/2)</div><div>3.28%</div><div>1 hour ago</div><div><div></div><div>:</div><div></div></div></div>
<input type="checkbox"/>	<div><div></div><div></div><div></div></div> <div><div><div>fuseki</div><div>8ebd3fd68f6e</div><div></div></div><div><div>stain/jena-fus</div><div>Running</div><div>3030:3030</div><div></div></div><div><div>0.54%</div><div>1 hour ago</div><div><div></div><div>:</div><div></div></div></div></div>
<input type="checkbox"/>	<div><div></div><div></div><div></div></div> <div><div><div>nifi</div><div>40cec6fd828e</div><div></div></div><div><div>apache/nifi:la</div><div>Running</div><div>8443:8443</div><div></div></div><div><div>2.74%</div><div>1 hour ago</div><div><div></div><div>:</div><div></div></div></div></div>

3.2 Traitement des Données avec Apache NiFi

Dans **Docker**, **Apache NiFi** est utilisé pour gérer les flux de données et effectuer l'intégration des différentes sources. Les étapes principales dans NiFi incluent :

1. **GetFile** : Ce processeur lit les fichiers CSV provenant des capteurs agricoles ou d'autres systèmes, les charge dans le pipeline de traitement.
2. **ExecuteScript (RDF)** : Les données sont converties en **RDF (Resource Description Framework)** via un script Groovy. Ce format sémantique facilite l'intégration et l'interopérabilité des données entre différents systèmes.
3. **SplitText** : Ce processeur divise les données en petits lots, ce qui permet un traitement parallèle et plus rapide des informations.
4. **InvokeHTTP** : Les données RDF sont ensuite envoyées vers **Apache Jena** pour être traitées et stockées sous forme de triplets RDF.
5. **ExecuteScript (JSON)** : Les données sont converties en **JSON** pour assurer leur compatibilité avec **MongoDB**.
6. **PutMongo** : Enfin, les données structurées sont stockées dans **MongoDB**, où elles peuvent être facilement accessibles et manipulées.

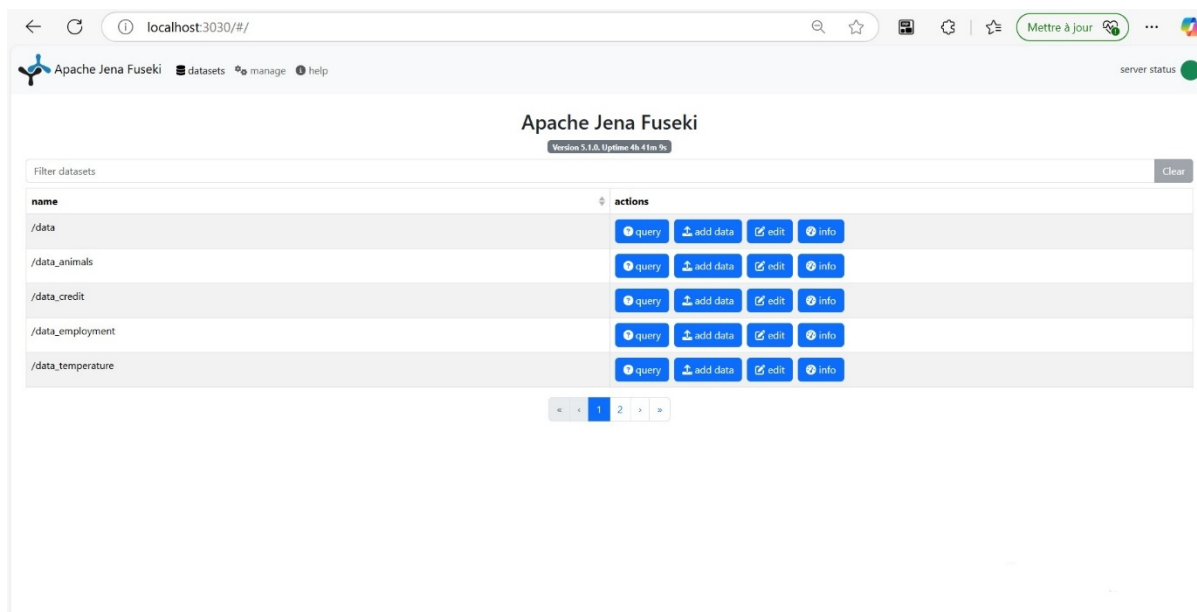


3.3 Gestion Sémantique avec Apache Jena

Une fois les données converties en **RDF**, elles sont envoyées vers **Apache Jena** pour un traitement et une gestion approfondis. **Jena** est un framework puissant pour travailler avec des données sémantiques et permet :

- **Stockage et gestion des triplets RDF**, permettant une représentation des données sous forme de relations sémantiques.
- **Exécution de requêtes SPARQL**, pour interroger et manipuler les données RDF et en extraire des informations spécifiques liées à l'agriculture.

Cela permet d'assurer une interopérabilité sémantique et une meilleure exploitation des données en fonction de leur signification et de leurs relations.

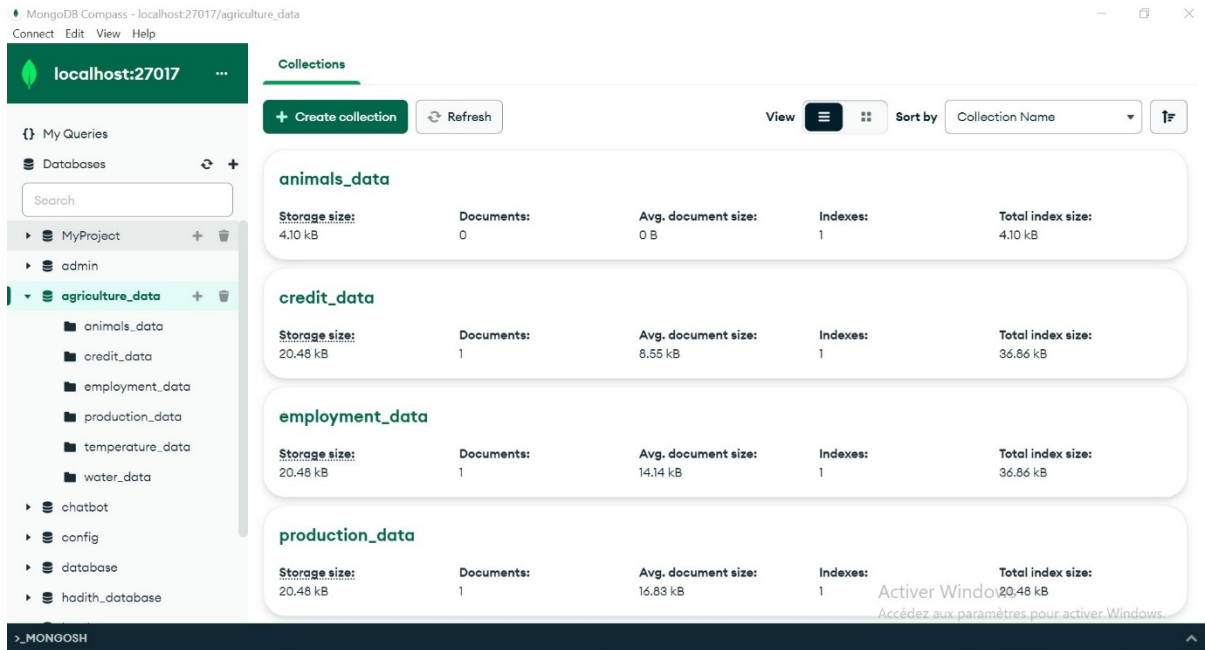


3.4 Stockage des Données avec MongoDB

Une fois traitées et enrichies, les données sont stockées dans **MongoDB**, une base de données NoSQL adaptée au stockage de données semi-structurées. La structure des données dans MongoDB comprend :

- **Données structurées** : Les informations collectées par les capteurs (telles que la température, l'humidité, etc.) sont stockées sous forme de documents JSON dans MongoDB.
- **Métadonnées** : Des informations supplémentaires, comme la source des données, la date et l'heure de collecte, sont également stockées, permettant de mieux gérer et suivre les données agricoles au fil du temps.

Ce stockage flexible permet une gestion optimale des données tout en garantissant leur évolutivité et leur accessibilité.



```

_id: ObjectId('678c2894da3c1e5bbe036b4c')
collection: "water_data"
database: "agriculture_data"
documents: Array
  0: Object
    name: "Barrage Abdelmoumen"
    type: "Abdelmoumen"
    longitude: 141748.535384
    latitude: 424137.036594
    metadata: Object
      createdAt: 1737238675817
      source: "infrastructure_integration"
      dataType: "barrage"
      status: "processed"
      region: "Souss-Massa"
      country: "Morocco"
      lastUpdated: "2025-01-18T22:17:55Z"

```

3.5 Backend avec Express.js

Le **backend** de l'application est construit avec **Express.js**, un framework Node.js léger permettant de créer une **API RESTful** pour interagir avec les données. Les principales fonctionnalités du backend incluent :

- **Gestion des requêtes SPARQL via Apache Jena** : L'API permet d'interroger les données RDF stockées dans **Apache Jena**, offrant une interface pour extraire des informations sémantiques via le langage de requêtes SPARQL.
- **Communication avec MongoDB** : L'API assure également l'interaction avec **MongoDB** pour stocker et récupérer les données structurées et les métadonnées liées aux données agricoles.

Cette API permet aux utilisateurs et aux autres services d'interagir avec les données de manière fluide et sécurisée. Voici les aspects de sécurité inclus dans le système :

🕒 Optimisation de la Sécurisation des Données :

- **Authentification JWT** : L'intégration d'un système d'authentification basé sur JSON Web Tokens (JWT) renforcera la sécurité de la plateforme en garantissant que seules les entités autorisées peuvent accéder aux données sensibles.

- **Génération des Tokens** : Les utilisateurs seront authentifiés et recevront un token JWT valide qui sera utilisé pour accéder à l'API. Ce token pourra être généré lors de la connexion ou du processus d'inscription.
- **Validation des Tokens** : L'API vérifiera la validité du token JWT pour chaque requête afin d'assurer que seules les demandes authentifiées peuvent accéder aux données.
- **Renouvellement des Tokens** : Pour améliorer l'expérience utilisateur et la sécurité, les tokens JWT peuvent être configurés pour avoir une durée de vie limitée. Lorsqu'un token expire, un nouveau peut être généré à partir d'un mécanisme de rafraîchissement (refresh token).

🕒 Gestion des Rôles et des Permissions (RBAC) :

- **Définition des Rôles** : En utilisant le contrôle d'accès basé sur les rôles (RBAC), il est possible de définir différents rôles d'utilisateur (par exemple, Administrateur, Agriculteur, Chercheur, etc.). Chaque rôle aura un ensemble spécifique de permissions permettant d'accéder à différentes fonctionnalités du système.
- **Gestion des Permissions** : Chaque rôle doit disposer d'un ensemble de permissions pour effectuer certaines actions sur la plateforme. Par exemple, un administrateur pourra ajouter ou modifier des utilisateurs, tandis qu'un agriculteur pourra uniquement consulter ses propres données.

3.6 Frontend avec Angular

Le **frontend** de l'application est développé avec **Angular**, permettant de créer une interface utilisateur dynamique et interactive. Voici les principales fonctionnalités du frontend :

- **Tableaux de bord interactifs** : Les utilisateurs peuvent visualiser les données collectées sous forme de graphiques et de tableaux dynamiques, facilitant l'analyse et la compréhension des données agricoles.
- **Visualisation des tendances agricoles** : Le système permet de suivre les évolutions des données, comme les rendements ou les conditions climatiques, offrant ainsi une vue d'ensemble des tendances agricoles.
- **Interface intuitive** : L'interface est conçue pour être simple et accessible à tous les utilisateurs, même ceux qui n'ont pas de connaissances techniques en données ou en programmation.

Cela permet aux utilisateurs de visualiser et d'analyser les données agricoles de manière efficace et intuitive, améliorant ainsi la prise de décision basée sur les données.

Application du Tableau de Bord

Connexion et Inscription

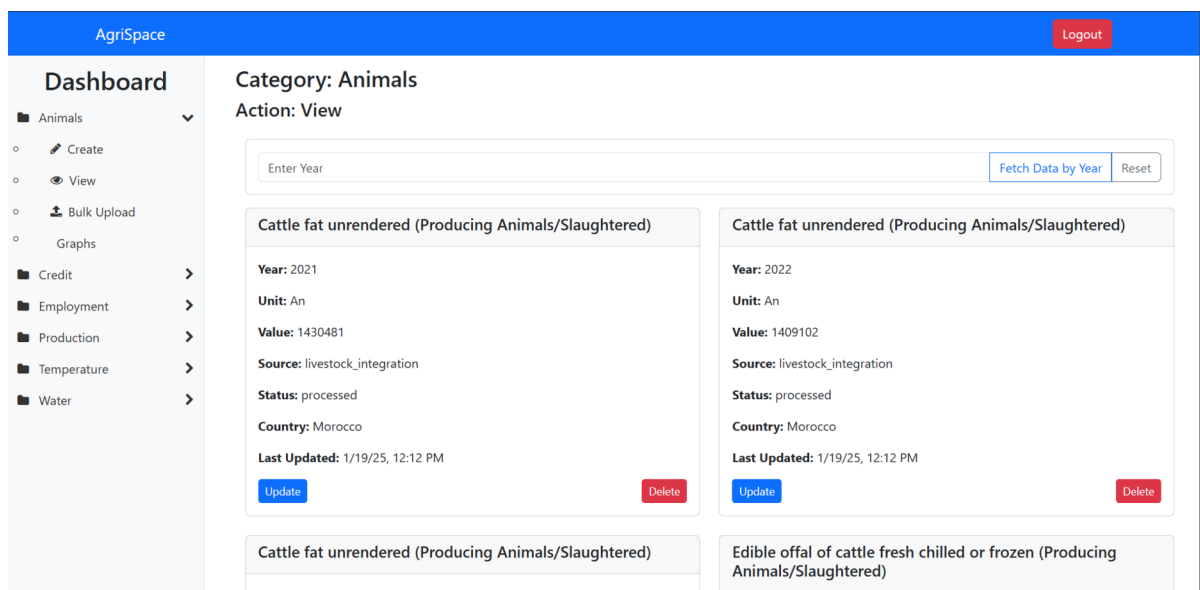
Lorsqu'un utilisateur accède à l'application, il est présenté avec une interface de connexion ou d'inscription. Cette étape permet d'assurer la sécurité et la personnalisation des données. L'utilisateur peut soit se connecter avec des identifiants déjà enregistrés, soit créer un nouveau compte pour accéder au tableau de bord.

Tableau de Bord et Création de Données

Après une connexion réussie, l'utilisateur est dirigé vers le tableau de bord principal. Celui-ci propose différentes options accessibles via la barre latérale. L'utilisateur peut créer de nouvelles entrées de données en remplissant un formulaire adapté.

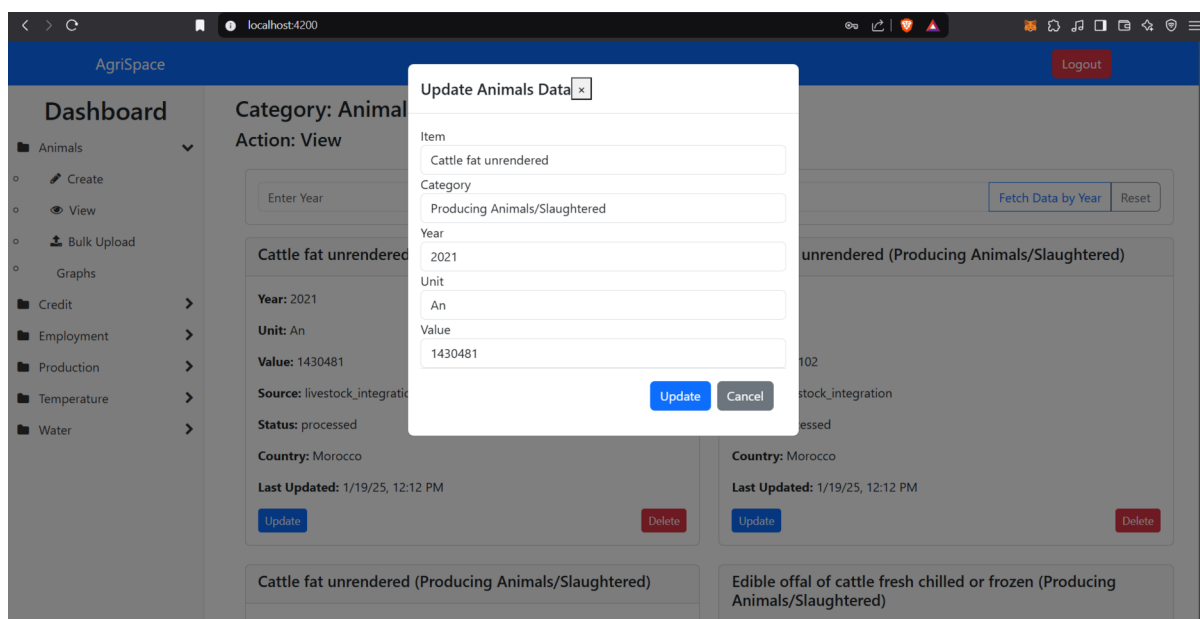
Consultation des Objets

L'interface de consultation permet d'afficher toutes les données enregistrées. L'utilisateur peut naviguer dans la liste ou utiliser un filtre basé sur l'année pour affiner sa recherche. Cette fonctionnalité assure une visualisation efficace des données.



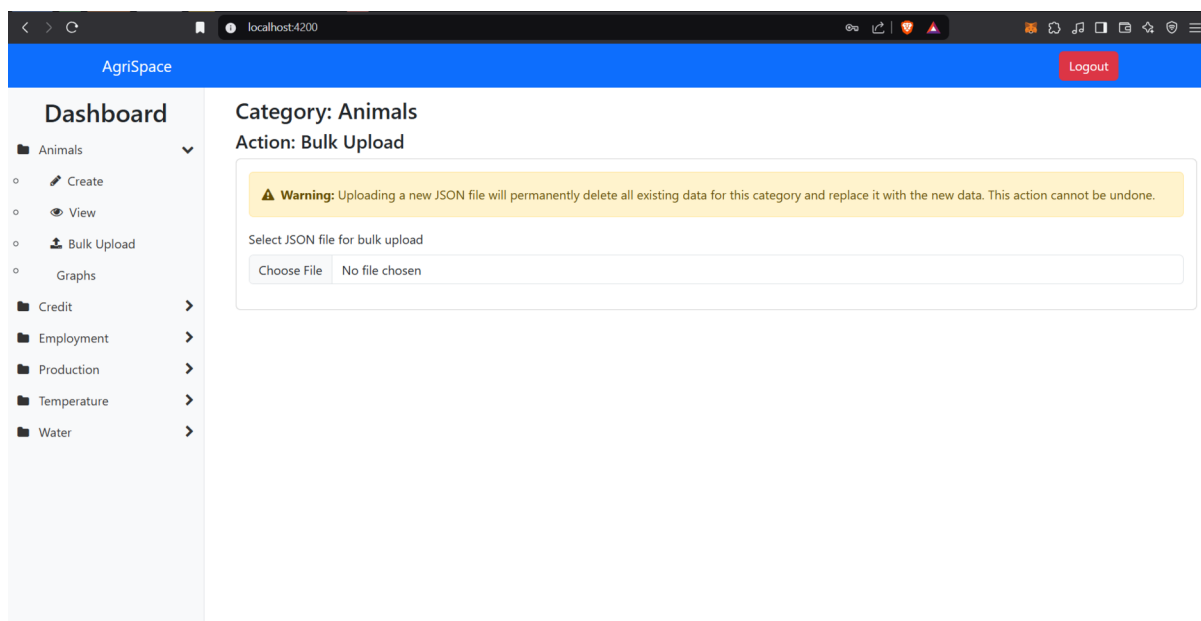
Mise à Jour et Suppression

Chaque objet listé peut être mis à jour en cliquant sur un bouton d'édition. De même, une option permet de supprimer un objet si nécessaire. Ces actions facilitent la gestion et la modification des informations enregistrées.



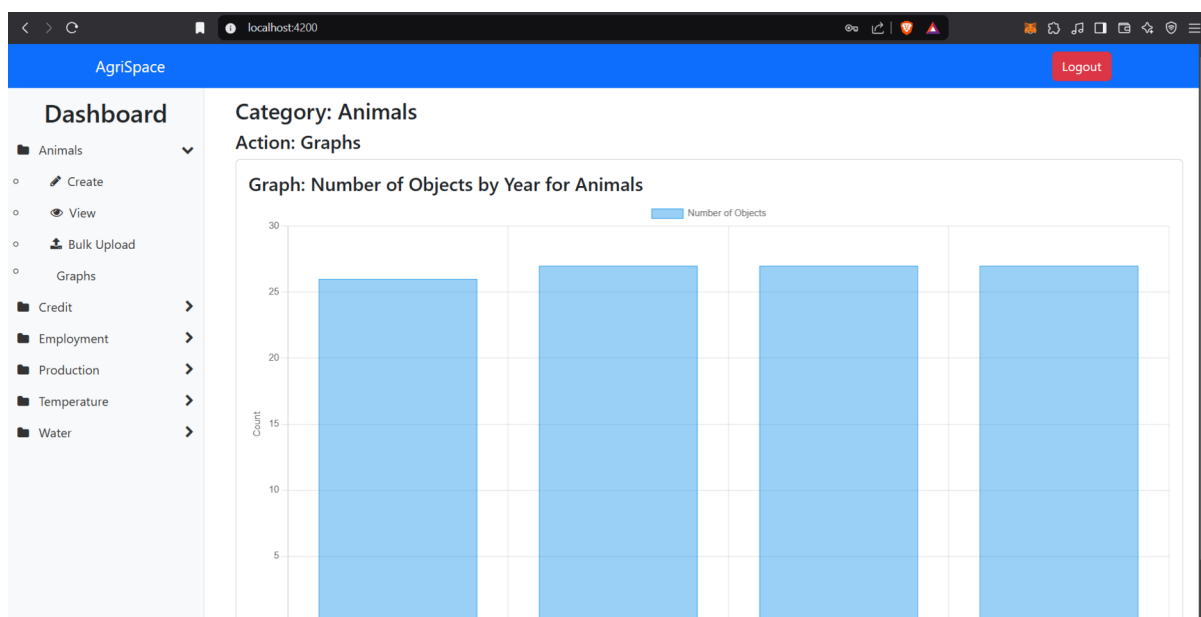
Importation en Masse

Pour simplifier la gestion de grandes quantités de données, une fonctionnalité d'importation en masse est disponible. L'utilisateur peut télécharger un fichier JSON contenant les données à ajouter, permettant ainsi une intégration rapide et efficace.



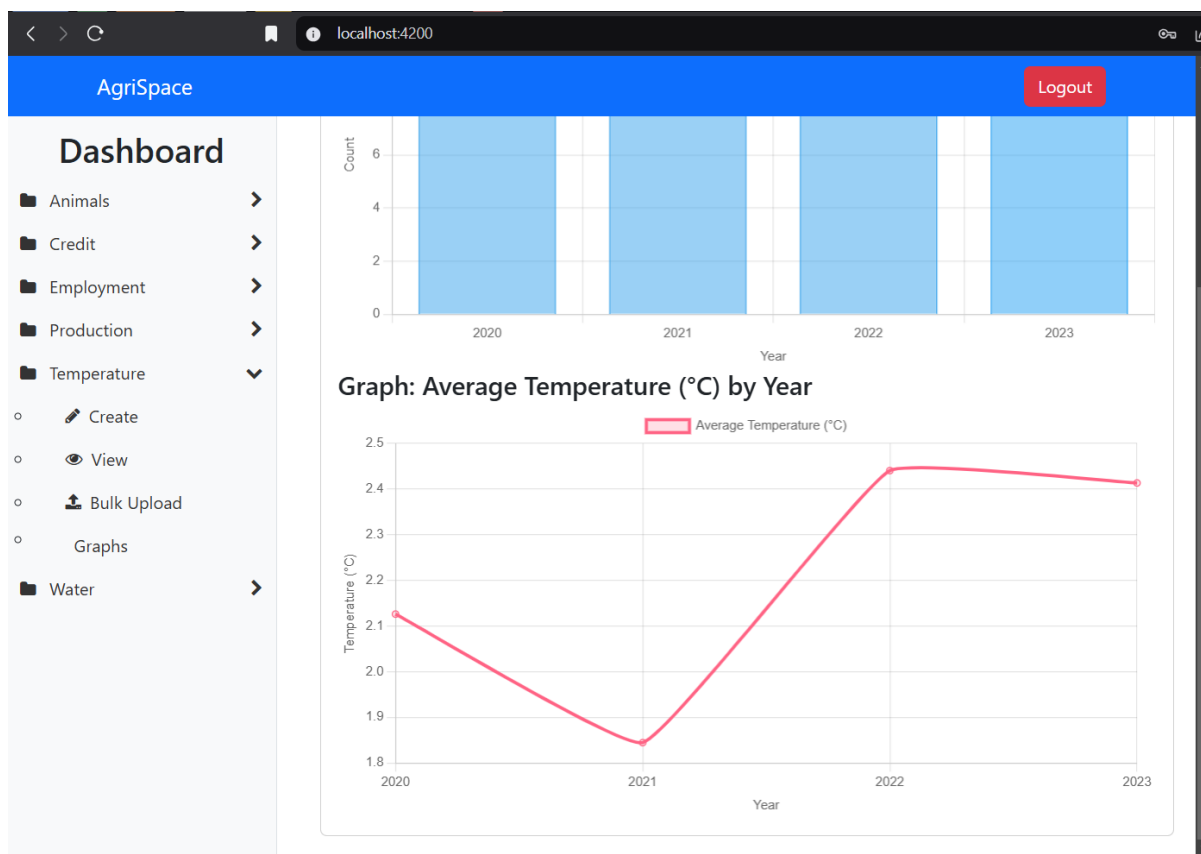
Graphique des Objets par Année

Un premier graphique est disponible pour représenter le nombre d'objets créés par année. Cet outil permet d'obtenir une vue d'ensemble des tendances de création au fil du temps.



Graphique des Températures Moyennes

Un second graphique affiche l'évolution des températures moyennes par année. Cette visualisation est utile pour analyser les tendances climatiques et interpréter les variations annuelles.



Ainsi, ce tableau de bord propose une expérience utilisateur intuitive et complète pour la gestion et l'analyse des données.

4. Conclusion et Perspectives

Le projet de **Data Space Agricole** pose les bases d'une agriculture marocaine moderne et durable en **centralisant et harmonisant** les données.

Perspectives d'évolution

- Intégration de l'intelligence artificielle pour l'analyse prédictive.
- Expansion à d'autres secteurs comme l'élevage et la gestion de l'eau.
- Collaboration avec des instituts de recherche pour enrichir les données.

Ce projet constitue un **modèle répliquable** pour la digitalisation d'autres secteurs stratégiques au Maroc.