|  |
| --- |
| Documentation technique |
| Green IA |
| 1. Architecture du système et présentation du projet |
| 1. Développement et technologies utilisées |
| 1. IA et Data 2. API, interface utilisateur et déploiement 3. Sécurité, maintenance et support 4. Tests, validation et améliorations 5. Annexes |

**Sommaire :**

[I. Architecture du système et présentation du projet 4](#_Toc171090094)

[1. Présentation du projet 4](#_Toc171090095)

[Contexte 4](#_Toc171090096)

[Objectifs 4](#_Toc171090097)

[2. Qu’est-ce que l’éco-score ? 4](#_Toc171090098)

[3. Vue d’ensemble 5](#_Toc171090099)

[4. Architecture matérielle 5](#_Toc171090100)

[5. Acteurs du système (utilisateurs, admins, autres) 5](#_Toc171090101)

[6. Interactions entre les acteurs et le système 5](#_Toc171090102)

[7. Diagrammes UML (Use Case, Sequence, Class) 5](#_Toc171090103)

[8. Liste des fonctionnalités 5](#_Toc171090104)

[Fonctionnalité 1 (code unique + description + use case + entrées et sorties + données + résultat) 5](#_Toc171090105)

[II. Développement et technologies utilisées 5](#_Toc171090106)

[9. Langages de Programmation 5](#_Toc171090107)

[10. Frameworks et bibliothèques 5](#_Toc171090108)

[11. Outils et environnements de développement 5](#_Toc171090109)

[12. Base de données (schémas, relations) 5](#_Toc171090110)

[13. Systèmes d’exploitation 5](#_Toc171090111)

[III. IA et data 5](#_Toc171090112)

[14. Collecte des données 5](#_Toc171090113)

[15. Préparation des données 5](#_Toc171090114)

[16. Prédiction des ingrédients 5](#_Toc171090115)

[17. Prédiction des mots clés des produits 5](#_Toc171090116)

[18. Prédiction de l’éco-score 5](#_Toc171090117)

[19. Entraînement des modèles 5](#_Toc171090118)

[20. Évaluation et validation des modèle 6](#_Toc171090119)

[21. Dashboard utilisateur 6](#_Toc171090120)

[22. Dashboard global 6](#_Toc171090121)

[23. Déploiement de nos solutions 6](#_Toc171090122)

[Stratégies de déploiement (blue-green, canary) 6](#_Toc171090123)

[Environnement de déploiement (dev, test, prod) 6](#_Toc171090124)

[Outils de déploiement (CI/CD pipelines, versions) 6](#_Toc171090125)

[24. Pipeline 6](#_Toc171090126)

[IV. API, interface utilisateur et déploiement 6](#_Toc171090127)

[25. Description des endpoints et protocoles utilisés 6](#_Toc171090128)

[26. Sécurité et authentification 6](#_Toc171090129)

[27. Intégration avec des services externes 6](#_Toc171090130)

[28. Présentation des pages et maquettes 6](#_Toc171090131)

[29. Charte graphique 6](#_Toc171090132)

[30. Accessibilité 6](#_Toc171090133)

[31. Environnement de déploiement 6](#_Toc171090134)

[32. Processus de déploiement 6](#_Toc171090135)

[33. Outils de déploiement (CI/CD) 6](#_Toc171090136)

[34. Stratégie de scalabilité (horizontale et verticale) 6](#_Toc171090137)

[V. Sécurité, maintenance et support 6](#_Toc171090138)

[35. Contraintes de sécurité 6](#_Toc171090139)

[36. Gestion des données sensibles 6](#_Toc171090140)

[37. Conformité et réglementations 7](#_Toc171090141)

[38. Plan de maintenance 7](#_Toc171090142)

[39. Surveillance et monitoring 7](#_Toc171090143)

[Rapport d’état de santé du système 7](#_Toc171090144)

[Rapport d’évolution des données 7](#_Toc171090145)

[Notifications aux administrateurs 7](#_Toc171090146)

[40. Plan des mises à jour 7](#_Toc171090147)

[41. Support utilisateur 7](#_Toc171090148)

[VI. Tests, validation et améliorations 7](#_Toc171090149)

[42. Critères d’acceptation pour chaque fonctionnalité 7](#_Toc171090150)

[43. Stratégie de tests 7](#_Toc171090151)

[44. Tests unitaires 7](#_Toc171090152)

[45. Tests d'intégration 7](#_Toc171090153)

[46. Tests de performance 7](#_Toc171090154)

[47. Améliorations envisagées 7](#_Toc171090155)

[VII. Annexes 7](#_Toc171090156)

[48. Glossaire 7](#_Toc171090157)

[49. Documents applicables 8](#_Toc171090158)

[50. Diffusion du document 8](#_Toc171090159)

[51. Historique des modifications 8](#_Toc171090160)

**Table des figures :**

**Aucune entrée de table d'illustration n'a été trouvée.**

# Architecture du système et présentation du projet

## Présentation du projet

### Contexte

Dans le monde d’aujourd’hui, l’empreinte écologique humaine ne cesse de croître, et la nécessité de consommer de manière plus responsable est peut-être aujourd’hui plus pressante que jamais. C’est dans ce contexte d’urgence écologique que la nécessité de Green IA est née : une application destinée à informer et guider les consommateurs vers des choix plus durables. En s’appuyant sur les dernières avancées de l’intelligence artificielle, Green IA vise à déchiffrer l’empreinte écologique des biens de consommation d’une manière simple et accessible à tous. Grâce à une application web révolutionnaire, Green IA veut rendre les pratiques écologiques alimentaires accessibles à tous en attribuant un éco-score aux produits alimentaires classiques de tous les jours, basé sur des critères environnementaux rigoureux ainsi qu’une note sur l’empreinte environnementale globale. Green IA est une application conçue dans le but de sensibiliser et d’inspirer à la protection écologique de notre planète par des choix de consommation plus écologiques.

### Objectifs

L’objectif de Green IA est de devenir une application désormais indispensable pour les consommateurs souhaitant choisir des solutions écologiques sans modifier leurs habitudes de consommation. Afin d’y parvenir, le but de Green IA est de faire en sorte que les gens s’engagent avec leurs biens à consommer d’une manière toute nouvelle en leur donnant accès aux données précises et à l’impact écologique de leurs produits. Ses objectifs principaux sont :

* Élevage de conscience : élever la conscience publique sur l’empreinte des produits à manger et rendre accessible les données pour tous.
* Informer le consommateur : fournir des données fiables et transparentes concernant les produits alimentaires et leurs empreintes carbone, consommation d’eau et plus encore.
* Pousser de bonnes options : proposer des options plus écologiques aux produits malsains et vendre des options plus respectueuses de l’écologie.
* Amélioration de la connaissance : mieux informer la bonne écologie de leurs habitudes.
* Réduction de l’empreinte : mesurer et suivre la réduction de l’empreinte hebdomadaire ou mensuelle de l’utilisateur et contribuer à des niveaux inférieurs au niveau de l’entreprise.
* Communauté positive : construire une plate-forme non seulement d’information mais de sensibilisation et d’encouragement.

## Qu’est-ce que l’éco-score ?

## Vue d’ensemble

## Architecture matérielle

## Acteurs du système (utilisateurs, admins, autres)

## Interactions entre les acteurs et le système

## Diagrammes UML (Use Case, Sequence, Class)

## Liste des fonctionnalités

### Fonctionnalité 1 (code unique + description + use case + entrées et sorties + données + résultat)

# Développement et technologies utilisées

## Langages de Programmation

## Frameworks et bibliothèques

## Outils et environnements de développement

## Base de données (schémas, relations)

## Systèmes d’exploitation

# IA et Data

## Collecte des données

## Préparation des données

## Prédiction des ingrédients

## Prédiction des mots clés des produits

## Prédiction de l’éco-score

## Entraînement des modèles

## Évaluation et validation des modèle

## Dashboard utilisateur

## Dashboard global

## Déploiement de nos solutions

### Stratégies de déploiement (blue-green, canary)

### Environnement de déploiement (dev, test, prod)

### Outils de déploiement (CI/CD pipelines, versions)

## Pipeline

# API, interface utilisateur et déploiement

## Description des endpoints et protocoles utilisés

## Sécurité et authentification

## Intégration avec des services externes

## Présentation des pages et maquettes

## Charte graphique

## Accessibilité

## Environnement de déploiement

## Processus de déploiement

## Outils de déploiement (CI/CD)

## Stratégie de scalabilité (horizontale et verticale)

# Sécurité, maintenance et support

## Contraintes de sécurité

## Gestion des données sensibles

## Conformité et réglementations

## Plan de maintenance

## Surveillance et monitoring

### Rapport d’état de santé du système

### Rapport d’évolution des données

### Notifications aux administrateurs

## Plan des mises à jour

## Support utilisateur

# Tests, validation et améliorations

## Critères d’acceptation pour chaque fonctionnalité

## Stratégie de tests

## Tests unitaires

## Tests d'intégration

## Tests de performance

## Améliorations envisagées

# Annexes

## Glossaire

|  |  |
| --- | --- |
| Abréviation | Signification |
| IA et Data Science | |
| IA | Intelligence Artificielle |
| Développement | |
| Framework | Environnement de travail facilitant le développement d’une solution technique. |
| Responsivité | Possibilité d’adapter la taille du logiciel à la taille de l’écran de l’utilisateur |
| Front | Développement des aspects visuels du logiciel |
| Back | Développement de la partie logique du logiciel (caché à l’utilisateur) |
| API REST | API Representational State Transfer Application Program Interface est un style architectural qui permet aux logiciels de communiquer entre eux sur un réseau ou sur un même appareil. Le plus souvent les développeurs utilisent des API REST pour créer des services web. Souvent appelés services web RESTful, REST utilise des méthodes HTTP pour récupérer et publier des données entre un périphérique client et un serveur. |

## Documents applicables

|  |  |
| --- | --- |
| Description | Identification |
| 23-24 Modalités Évaluations Titre EISI N7 Étudiants – AYC Pour les M2 | REF [0] |
| Dépôt Moodle filière informatique M2 | REF [1] |

## Diffusion du document

|  |  |  |  |
| --- | --- | --- | --- |
| Diffusion | Statut | Nom | Emis le |
| Edition | Charlemagne | 05/07/2024 |
|  |  |  |
|  |  |  |

## Historique des modifications

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Pages | Description de la modification - Auteur | Date |
| 0.3 |  | Architecture globale du document | 05/07/2024 |
|  |  |  |  |
|  |  |  |  |