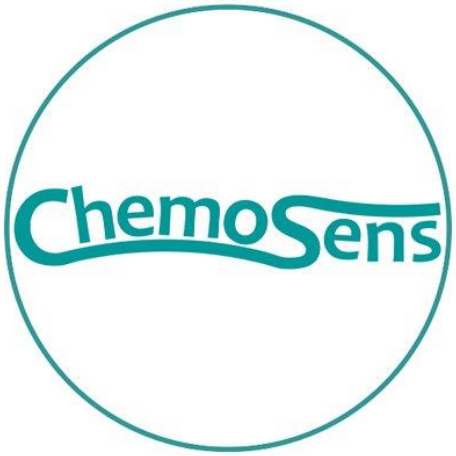
IUT Dijon-Auxerre – Département Informatique

Développement d'une solution Web d'analyse de factures

Rapport de stage BUT2 Informatique



Tuteur en entreprise : M. Michel VISALLI (Ingénieur de recherche)

Tuteur pédagogique : M. Franck MARZANI

Par Tristan DAL MOLIN
Année universitaire : 2024-25

Remerciements

Je remercie grandement M. Michel VISALLI, mon maître de stage, pour m'avoir accepté durant ces 8 semaines, avoir su me donner des consignes claires durant tout le stage et m'aider lorsque j'en avais besoin.

Ma reconnaissance ira de même aux différents employés de ChemoSens qui ont su être accueillants durant tout le long du stage, notamment M. Kipédène COULIBALY ayant grandement contribué au projet qui m'était attribué, chose que je développerai durant le reste du rapport.

Toute ma gratitude ira également aux différents acteurs de l'IUT de Dijon, dont M. Yannick BENEZETH, responsable des stages des 2ème année de BUT Informatique en cette année scolaire 2024-2025, ayant grandement aidé durant la recherche de stage, le suivi de ce dernier et contribué à la compréhension des attendus vis-à-vis de son évaluation, ainsi que mon tuteur pédagogique, M. Franck MARZANI, pour le suivi du stage et son évaluation.

TABLE DES MATIERES

I INTRODUCTION

Diplômé d'un baccalauréat scientifique (enseignements de spécialités : Mathématiques, physique-chimie, option Mathématiques Expertes) mention bien m'ayant introduit au domaine de la programmation par le biais de Python, mes compétences et ma passion pour l'informatique m'ont naturellement orienté vers un BUT Informatique dans l'objectif de travailler, à l'avenir, dans le développement de solutions numériques et/ou d'applications.

La réalisation d'un stage de huit semaines était nécessaire pour valider le diplôme, l'objectif étant de nous introduire au monde professionnel, et éventuellement identifier des possibilités futures pour une alternance ou un contrat en sortie de diplôme, tout en mettant nos compétences à contribution dans différents projets.

Durant mes recherches de stage, mon attention a fini par se focaliser sur les différents laboratoires de recherche disposant d'un département informatique dans le but de contribuer à des études en cours ou futures. Ayant de solides bases scientifiques dues à mon parcours et des compétences de développement logiciel acquises en IUT, essayer de combiner les deux dans le cadre de ce stage m'a mené à envoyer une candidature spontanée au laboratoire ChemoSens.

Accepté à la suite d'un entretien, ma mission était la suivante : développer un outil pour automatiser l'analyse de factures alimentaires.

Durant ce rapport, je présenterai en détails le contexte de ce projet (le laboratoire accueillant, les activités qui m'étaient confiées, etc.) avant d'aborder les différentes solutions proposées, les outils utilisés ou encore les difficultés rencontrées en cours de route, pour finalement revenir sur les compétences qui ont pu être approfondies au cours de ce stage.

II PRESENTATION GENERALE

II.1 PRESENTATION DE L'INRAE

L'INRAE (Institut National de Recherche pour l'Agriculture, l'alimentation et l'Environnement) est un **organisme public** de **recherche scientifique et technologique** placé sous la tutelle du **ministère de l'Enseignement Supérieur de la Recherche** (MEST)[1] et de celui de **l'Agriculture et de la Souveraineté Alimentaire**[2].

L'INRAE a pour missions principales de **produire des connaissances scientifiques sur l'agriculture, l'alimentation, et l'environnement**, en visant à accompagner la transition écologique et énergétique. Il mène des recherches sur la durabilité des systèmes agricoles, la sécurité alimentaire, et la santé publique, notamment l'impact des régimes alimentaires sur la santé humaine. L'INRAE développe également des solutions innovantes pour une agriculture plus résiliente, collabore avec des partenaires industriels et publics pour traduire ses recherches en applications concrètes, et participe à la formation et à la diffusion des connaissances scientifiques. Enfin, il répond aux enjeux mondiaux liés à l'alimentation et à la gestion des ressources naturelles dans un contexte de changement climatique.

L'organisme décompte un total de **18 centres** [1] **de recherches en France**, chacun étant spécialisé dans un ou plusieurs domaines de recherche

Le **centre de Bourgogne-Franche-Comté**, dont l'implantation principale est localisée à **Dijon**, a pour thèmes de recherches principaux **l'agroécologie**, les **territoires ruraux et périurbains** ainsi que **le goût et l'alimentation** [2].

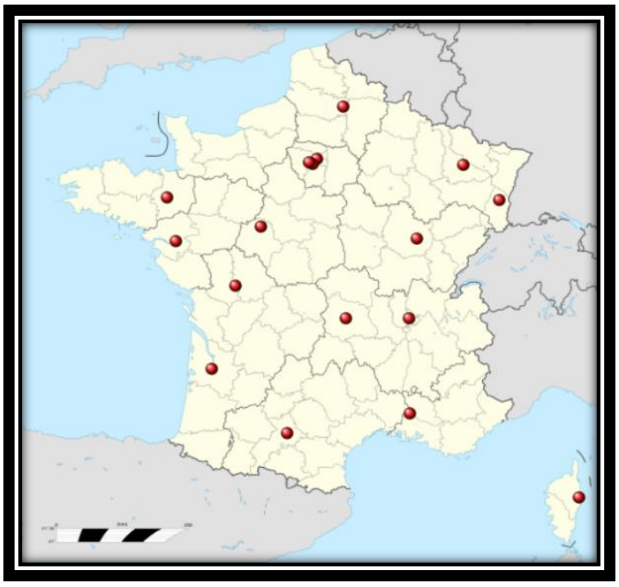


Figure 1 : Carte des centres de Recherche d'INRAE (Source)

[1] Le ministère de l'enseignement supérieur français [32] est en charge à la fois des parcours supérieurs d'étudiants et de l'attribution de budgets à différents organismes de recherche.

[2] Le ministère de l'Agriculture français [33] est responsable, non seulement du secteur agricole (alimentaire, forestier, …), mais aussi partiellement de la recherche concernant ces domaines.

II.2 PRESENTATION DU CSGA ET DE CHEMOSENS

La structure organisationnelle liant ChemoSens à INRAE est la suivante :

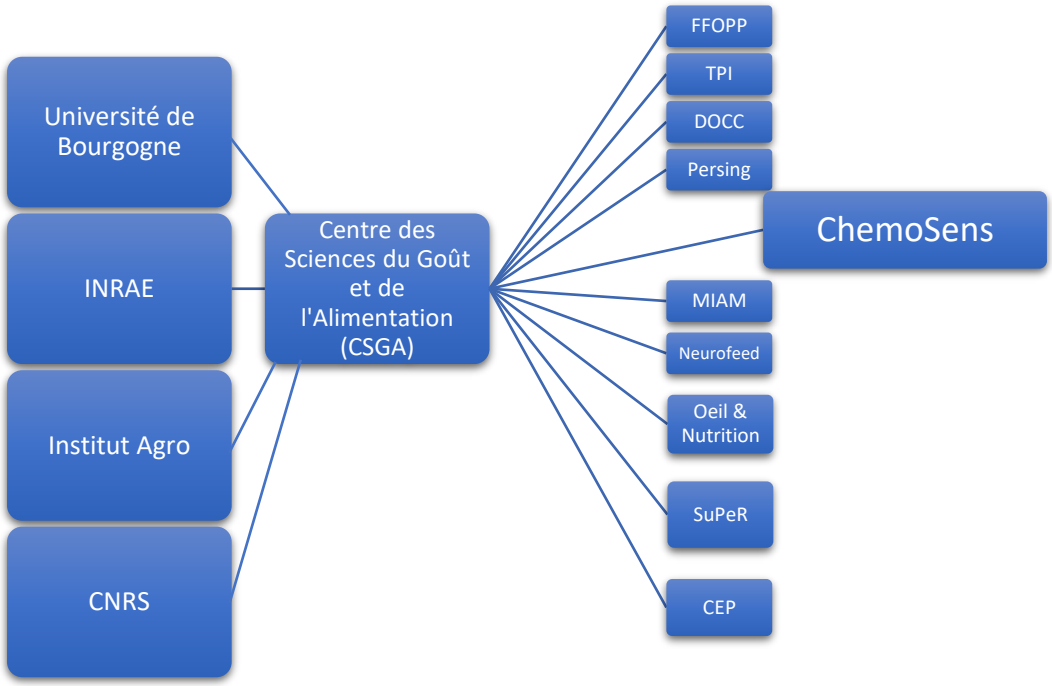


Figure 2 : Organigramme structurel de ChemoSens

Le CSGA (Centre des Sciences du Goût et de l'Alimentation) est une unité mixte de recherche sous les tutelles d'INRAE, mais également de l'Université de Bourgogne, de l'Institut Agro Dijon, et du CNRS (Centre National de la Recherche Scientifique). Le CSGA a pour principal objectif d'apporter une **meilleure compréhension des différents mécanismes** (biologiques, psychologiques, physicochimiques, etc.) **sous-tendant la perception sensorielle et le comportement alimentaire** [3].

Cette entité est organisée en différentes équipes, dont ChemoSens, au sein de laquelle j'ai réalisé mon stage. ChemoSens est une plateforme de recherche, structurée en quatre pôles techniques.

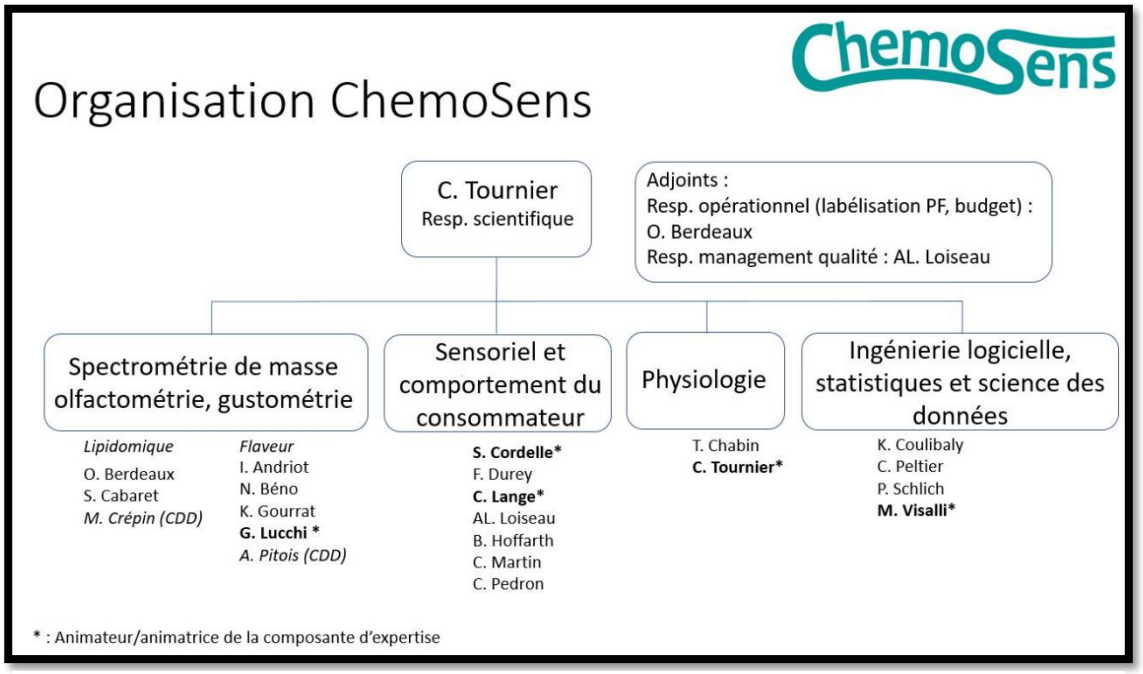


Figure 3 : Organigramme de ChemoSens (Source)

ChemoSens vient en appui aux autres équipes du CSGA, collabore avec de nombreux partenaires académiques et industriels en France, en Europe et dans le monde. ChemoSens met à disposition **son expertise technique dans** des domaines variés tels que la spectrométrie de masse (analyses physicochimiques), l'olfactométrie (mesure de l'odorat), la gustométrie, l'analyse sensorielle, le comportement du consommateur, la physiologie et la neuroimagerie, et l'ingénierie logicielle, les statistiques et la science des données.

Durant mon stage, j'ai été rattaché au pôle technique « ingénierie logicielle, statistiques et science des données », coordonné par M. Visalli (mon maître de stage). Mon lieu de travail était son bureau, partagé avec celui de M. Coulibaly, ce dernier ayant par ailleurs été responsable du déploiement de l'application développée durant ce stage. Ce pôle a notamment développé plusieurs solutions logicielles ou statistiques sont proposées telles que **TimeSens**, permettant d'acquérir des données sensorielles et temporelles [4], différents **packages R** automatisant des prétraitements de données physico-chimiques ou sensorielles [5] et différents **modèles statistiques** ou **bases de données**.

II.3 PRINCIPE DE LA MISSION

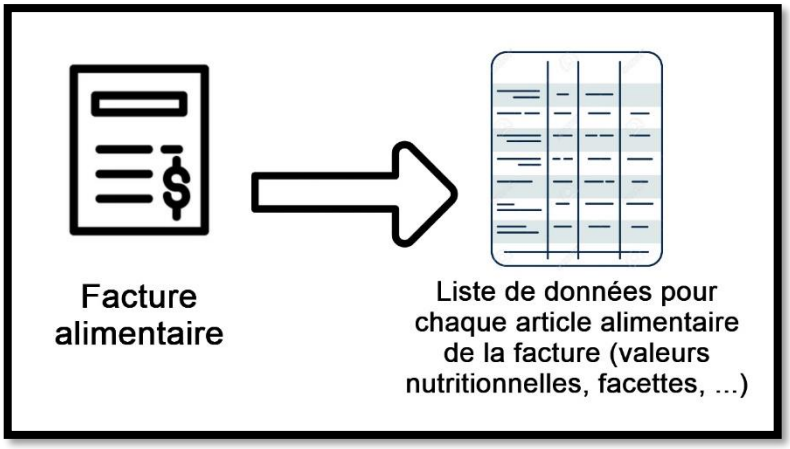


Figure 4 : Objectif du stage

Pour analyser le comportement des consommateurs, il est essentiel de collecter et d'analyser des données de consommation obtenues en vie réelle. La plupart des consommations passant par des achats, les **factures alimentaires** s'avèrent être une source d'information particulièrement intéressante. La méthode traditionnelle consiste à demander à des consommateurs de reporter tous leurs approvisionnements, par exemple en collant les factures de leurs achats alimentaires dans un carnet de suivi. Ce carnet est ensuite exploité manuellement par des opérateurs qui identifient les aliments, les associent à des référentiels de données pour les catégoriser et obtenir (par exemple) des informations sur leurs caractéristique nutritionnelles ou environnementales.

Ma mission consistait à automatiser ce process, en d**éveloppant une solution qui permette aux consommateurs eux même de téléverser leurs factures**, puis qui **extrait automatiquement l'information pertinente de ces factures** (noms des aliments, quantités, prix, etc.), classe les aliments dans des **catégories prédéfinies dans un référentiel**, et **collecte des informations supplémentaires** (telles que leurs valeurs nutritionnelles, leur taux de matière grasse, etc.) à partir de différentes sources de données externes.

Pour ce faire, j'étais totalement libre en ce qui concerne le langage de programmation et, ayant passé les derniers mois sur plusieurs projets d'applications web en PHP, j'ai naturellement décidé d'approfondir ces compétences et d'utiliser mes acquis de cours pour réaliser un tel projet.

Le schéma ci-dessous détaille chaque étape du processus :

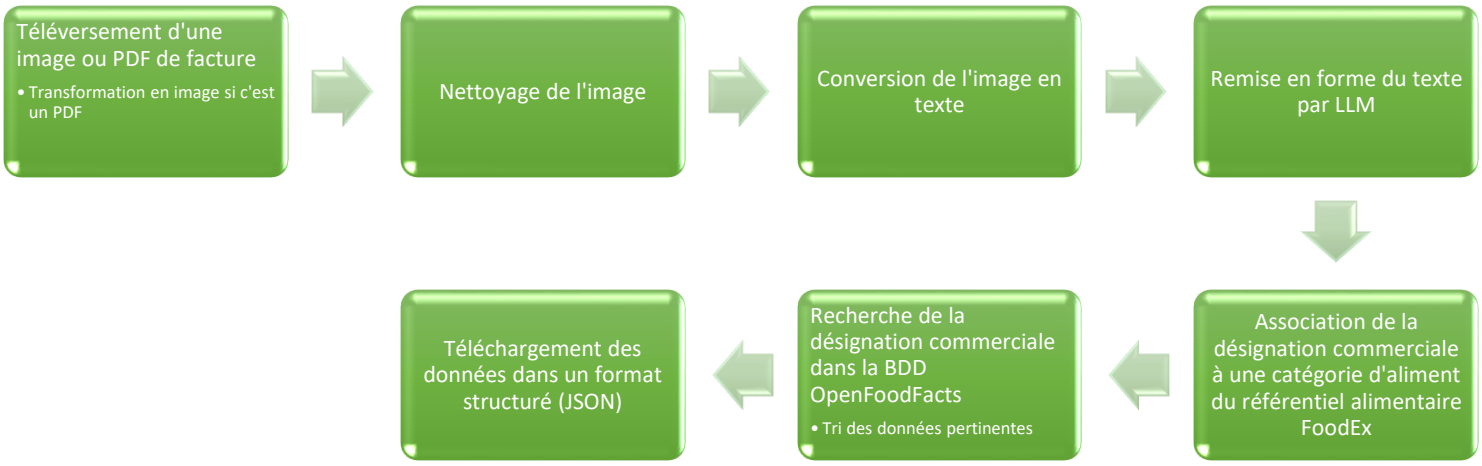


Figure 5 : Processus de l'analyse d'une facture

III LA TACHE EFFECTUEE : ANALYSE DE FACTURES

III.1 STRUCTURATION D'UNE FACTURE

Le premier aspect important du programme, et le plus difficile à implémenter, était la **restructuration d'une facture au format JSON** (étape 4 de la figure 5), le majeur problème étant que nous cherchons à **généraliser un tel processus à tous les types de factures rencontrés**, provenant de n'importe quel magasin peu importe quand, de sorte que l'on n'ait pas besoin de revenir sur ce programme pour d'autres types de factures une fois déployé.

Les **deux premières semaines de stage** ont été dédiées à la recherche et l'implémentation de la solution d'analyse de facture la plus pertinente possible qui renverrait un fichier structuré et réutilisable dans un programme. Les différentes solutions envisagées sont décrites ci-dessous.

a. API d'analyses par intelligence artificielle (IA)

La première solution envisagée était d'utiliser une solution existante en envoyant la facture à une API[3] qui renverrait tout simplement toutes les informations demandées au mieux, et nous fournirait une structure de données réutilisable pour enrichir les informations manquantes (tâches 1, 2, 3, 4 et 7 de la Figure 5).

Une journée était dédiée aux tests de différents modèles (ChatGPT, Perplexity, PDFPeer, etc.)

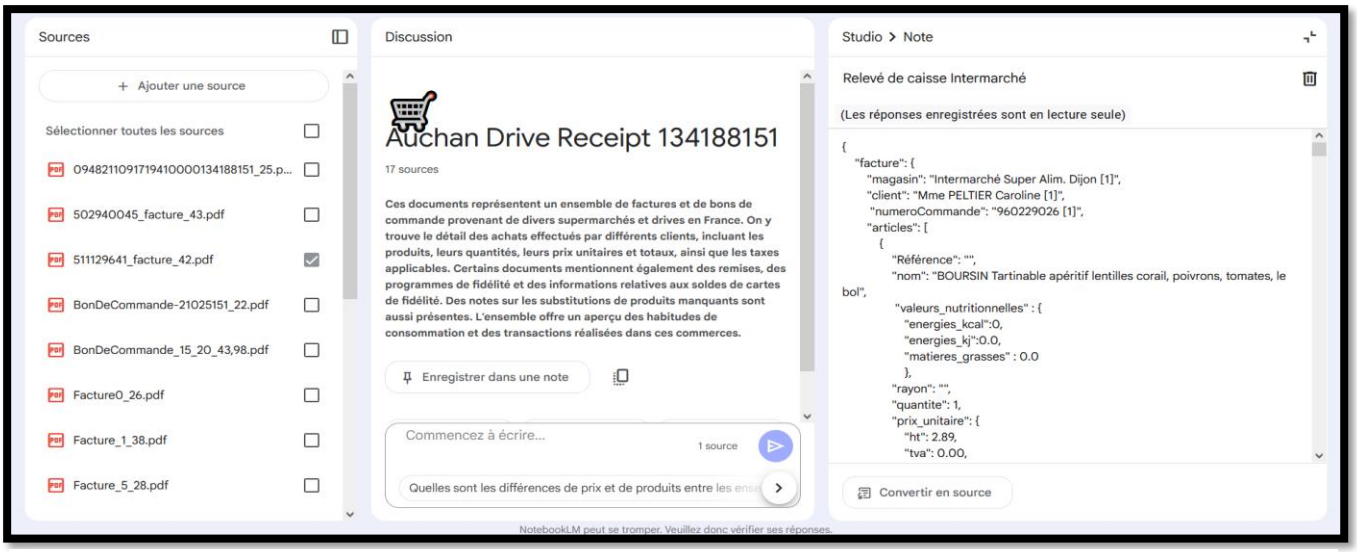


Figure 6 : Exemple de requête à NotebookLM

Si cette solution était idéale a priori, elle présentait plusieurs inconvénients dans le cadre d'une solution basée sur l'appel à des APIS. La plupart des IA ne se sont pas avérés appropriés pour traiter des images ou des gros documents (**plus la facture était longue, plus leurs réponses avaient tendance à devenir chaotiques)**. Les modèles les plus pertinents (comme NotebookLM, cf. Figure 6) **n'avaient pas d'API** (ce qui empêche donc de l'utiliser dans notre programme) ou leur **API** était **payante**, ce qui n'était pas une alternative pour ChemoSens cherchant plutôt à proposer cette solution sous forme de service gratuit.

[3] Une API (Application Programming Interface, « interface de programmation d'application ») [42] est une solution tierce qui donne un accès externe à une application. Dans notre cas, on cherche une API que du code PHP pourrait appeler pour utiliser l'application en question par exemple.

b. Transformation de la facture en texte

À la suite des premiers essais peu concluants d'analyse d'images par API, il a été décidé de d'abord **extraire le texte de la facture, avant d'envoyer ce même texte à l'IA** plutôt que les images elles-mêmes. Ainsi, peu importe la solution retenue, nous pourrions simplement **envoyer la facture par blocs de texte**, prévenant donc les problèmes liés à la taille de requêtes à une IA. Pour ce faire, la transformation se faisait en trois temps avant d'en arriver à l'étape finale :

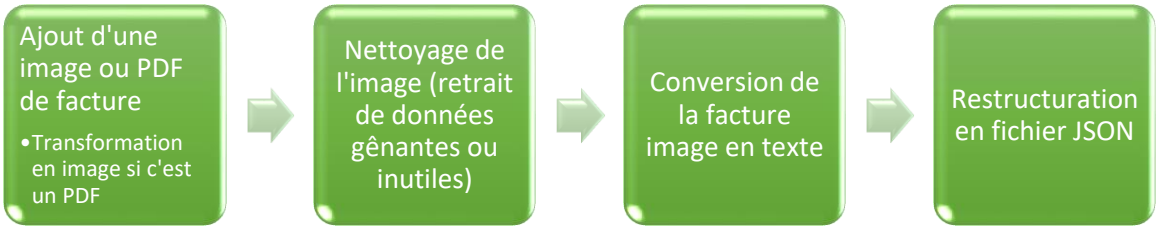


Figure 7 : Processus de prétraitement d'une facture

La première étape consistait à transformer le fichier PDF en image à l'aide de **commandes** et de **bibliothèques Shell**[4] comme **pdftoppm** [6].

La seconde nécessitait **d'éditer l'image sur le site web** directement, afin de masquer les informations personnelles des consommateurs (nom, prénom, date d'achat, etc.), ce qui se faisait à l'aide d'une bibliothèque JavaScript[5] cette fois, du nom de **Konva** [7].

C'est une fois ces informations remplies que l'on pouvait envoyer les images modifiées au programme qui **convertissait ces images en un texte** à l'aide de **Tesseract** [8].

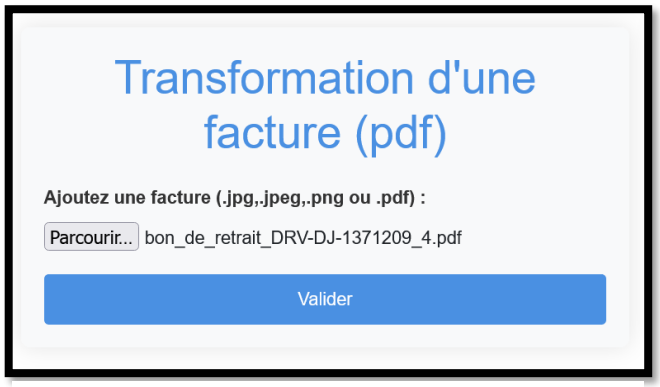


Figure 8 : Bouton de transformation d'une facture (Page d'Accueil)

Figure 9 : Édition d'images en ligne (Konva)

[4] Par Shell, on entend un interpréteur de commandes de type Unix [52], une interface beaucoup plus proche du système en lui-même. Lesdites bibliothèques sont donc des programmes exécutables dans cet environnement à l'aide d'une commande.

[5] JavaScript [53] est un langage de programmation utilisé pour des interactions directes avec une page web, notamment ici pour modifier des images à l'écran.

c. Entraînement d'un modèle

Au vu des limitations évoquées dans la section sur les IA génératives ou LLM[6], une solution était **d'entraîner notre propre modèle à la tâche**. Les factures étant désormais au format textuel, il suffisait en théorie d'envoyer ce même texte aux côtés de différentes directives à une IA pour obtenir le fichier structuré attendu.

Modèle BERT

Notre premier choix s'est porté vers un modèle BERT[7], ceci a donc demandé des compétences beaucoup plus poussées de ma part, car il fallait **héberger ce modèle** sous la forme d'un **serveur Python** avec lequel notre programme interagirait.

Étant peu familier avec Python[8] et la mise en place d'un serveur sous ce langage, cette tâche m'a demandé de plus grands efforts et beaucoup de travail de compréhension et de structuration.

Le principe de base était d'entraîner un modèle à l'aide de fichiers de **fine tuning**[9], **enrichis de conversions de factures en objet JSON** réalisés à l'aide de solutions fonctionnant précédemment (NotebookLM qui réalisait ces conversions gratuitement sur son site par exemple).

Malheureusement, cette solution présentait de nouveau des inconvénients. D'une part, les modèles BERT avaient également de **grandes difficultés à traiter des textes trop longs** : ce qui implique qu'une fois la facture envoyée, même un modèle entraîné ne renvoyait qu'un texte ne faisant pas sens par rapport à la demande. D'autre part, nous n'avions qu'une **cinquantaine de factures à disposition** ce qui ne **s'est pas révélé être assez** pour entraîner un modèle et **obtenir des performances acceptables.** Enfin, entraîner et héberger **un modèle BERT exigeait beaucoup de ressources** d'un poste de travail ou d'un serveur, impliquant un plus long temps d'exécution.

[6] Un LLM (Large Language Model, « Grand modèle de langage ») est un modèle possédant un grand nombre de paramètres, utilisés sous la forme d'un réseau neuronal n'ayant pas ou peu besoin de supervision réelle. Ils sont notamment utilisés pour déduire des suites logiques de données, dans notre cas : pour déduire une structure de données d'une facture envoyée en paramètre [55].

[7] Le langage BERT [44] permet de vectoriser un texte et de « comprendre » le contexte d'une phrase ou de mots dans une requête fournie par exemple.

[8] Python [45] est un langage de programmation connu pour sa syntaxe simple et ses outils de haut niveau. Il est notamment utilisé dans le domaine de l'IA et du Machine Learning

[9] Le fine tuning (réglage fin en français) est une méthode d'apprentissage qui permet à un LLM d'étendre ses connaissances sur des jeux de données spécifiquement liés à la question d'intérêt. Dans notre cas, nous fournissons à un modèle BERT les réponses attendues à différentes factures entrées [46]

Modèle Ollama

Après BERT, notre attention s'est portée vers les **modèles Ollama** [9] qui sont **mis à disposition gratuitement**. Cette fois-ci, les résultats se sont révélés être beaucoup plus pertinents et, avec la bonne requête, nous avons pu obtenir au moins le fichier JSON demandé.

Cependant, bien que meilleure, cette solution n'était pas idéale et présentait en partie les mêmes problèmes que beaucoup d'IA générative essayées en début de projet : les **factures trop longues** finissent par donner des **réponses de moins en moins précises**, le **format JSON exigé n'était pas totalement respecté** (empêchant donc d'automatiser son utilisation) et même dans le cas où les champs demandés seraient remplis, les informations extraites n'étaient pas suffisamment exactes.

d. Choix final

La solution d'entraîner son propre modèle aurait pu être viable si nous avions des centaines de factures déjà converties à lui fournir en entraînement, mais elle n'a pas été retenue le temps de ce stage en raison de son manque de précision et de son temps d'exécution.

Notre choix se retrouvait donc très limité : nous nous sommes donc tournés vers le seul modèle qui se trouvait être pertinent : ChatGPT. Cette IA générative réussissait à tenir le rythme même sur de longs documents, et pouvait de plus **ajouter des informations manquantes à la plupart des aliments**. Le coût de son API était également raisonnable pour de l'envoi de texte : **pour 1$US** (≈ 0.91€ à ce jour)**, c'est environ 200 factures qui pouvaient être analysés** de la sorte[10]. (Exemple de fichier JSON retourné par ChatGPT en figure 10.)

Cette solution a donc malgré tout été retenue, mais une **alternative gratuite** restera recherchée après ce stage, ce pourquoi les différentes factures analysées seront sauvegardées de sorte qu'avec suffisamment de factures, et les LLMs évoluant de plus en plus vite, un modèle puisse être entraîné à partir des résultats de ChatGPT.

Figure 10 : Conversion en JSON par ChatGPT

[10] Pour les modèles les moins chers, mais largement suffisants pour cette tâche, le coût était d'environ 1.00$ pour 1 million de « tokens » [41] (unité de mesure des caractères dans le domaine des LLM), une facture faisant en moyenne 5 000 tokens une fois mise sous format textuel, on peut donc aisément analyser 200 factures pour 1$.

III.2 ENRICHISSEMENT DE LA FACTURE

En parallèle de la recherche de solution pour la structuration de la facture, une partie importante de mon stage (de la troisième à la sixième semaine environ, pendant quatre semaines) a pu être dédiée à l'**enrichissement des désignations commerciales par des informations complémentaires issues d'autres sources de données**. Deux sources de données ont été considérées : le **référentiel FoodEx2** qui est une nomenclature standard, ainsi que la base de données collaborative **Open Food Facts**.

a. Correspondance FoodEx

Le **référentiel FoodEx2** [10] est un **système de classification développé par l'EFSA** (European Food Safety Authority) pour faciliter l'analyse et la gestion des données liées à la consommation alimentaire au niveau européen. Ce système est largement utilisé dans le cadre de recherches, d'études de consommation alimentaire, et de gestion des risques liés à l'alimentation

Principe de base

Un tableur Excel contenant toutes les correspondances FoodEx m'a été fourni. L'objectif était d'associer **une désignation commerciale donnée** (un des aliments de la facture) **à l'aliment du référentiel** qui s'en rapprocherait le plus.

Par exemple, un article d'une facture qui aurait pour désignation commerciale « Riz long grain », serait associé à l'élément de la ligne 13 du référentiel, dont le code sera « A.01.000011 » (voir Figure 11 ci-contre).

À partir d'une bibliothèque PHP permettant de lire et d'interagir avec des fichiers tableurs, j'ai pu reprendre un travail qui avait déjà été commencé plus tôt par ChemoSens en C#.

	A	C	
1	termCode	termExtendedName	Translation
2	A.01.000000	FOODEX1 terms	
3	A.01.000001	Grains and grain-based products	Céréales et produits à base de céréales
4	A.01.000002	Grains as crops	Les céréales comme cultures
5	A.01.000003	Wheat grain crop	Récolte de grains de blé
6	A.01.000004	Barley grain (Crop)	Grain d'orge (Culture)
7	A.01.000005	Corn grain (Crop)	Grain de maïs (Culture)
8	A.01.000006	Rye grain (Crop)	Grain de seigle (Culture)
9	A.01.000007	Spelt grain (Crop)	Épeautre (Culture)
10	A.01.000008	Buckwheat grain (Crop)	Grain de sarrasin (Culture)
11	A.01.000009	Millet grain (Crop)	Grain de millet (Récolte)
12	A.01.000010	Oats, grain (Crop)	Avoine, céréales (Culture)
13	A.01.000011	Rice (Crop)	Riz (Récolte)
14	A.01.000012	Other grains (Crop)	Autres céréales (Culture)
15	A.01.000013	Grains for human consumption	Céréales pour la consommation humaine
16	A.01.000014	Wheat grain	Grain de blé
17	A.01.000015	Wheat germ	Germe de blé
18	A.01.000016	Wheat grain, Durum	Grains de blé dur
19	A.01.000017	Wheat grain, soft	Grain de blé tendre

Figure 11 : Tableur du Référentiel FoodEx2

Ce dernier s'appuyait sur des comparaisons de mots-clefs entre les différentes désignations et à base d'expressions régulières[11] créées à partir de désignations et de prétraitements de texte.

[11] En informatique, une expression régulière désigne une chaîne de caractères, un ou plusieurs mots, que l'on souhaiterait utiliser à des fins de comparaisons, d'analyse textuelles ou encore de modifications d'un texte [48]. Différents opérateurs sont utilisés afin de permettre des comparaisons plus poussées entre différentes chaînes données [49]

La comparaison suivait initialement le processus suivant :

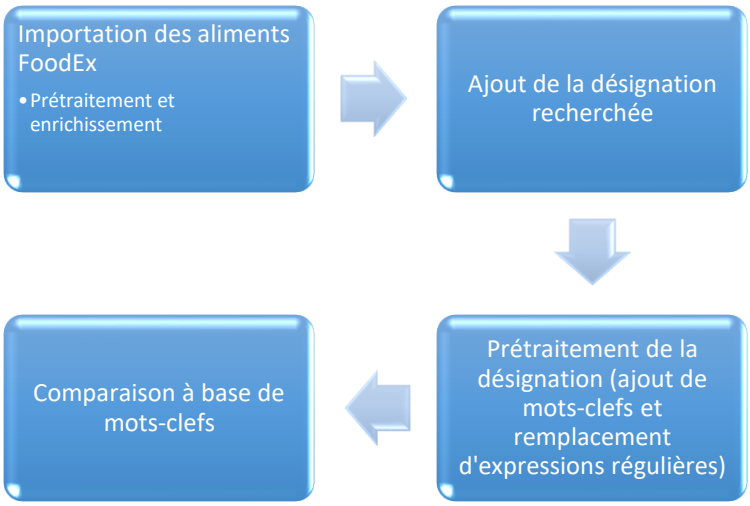


Figure 12 : Processus initial de recherche de désignation FoodEx

Mais ce dernier s'avérait inefficace en l'état : comparer simplement des mots-clefs entre eux aboutissait souvent à **retourner des erreurs ou de mauvaises correspondances** et la génération de mots-clefs (se faisant uniquement à partir de la désignation de l'article) pouvait également être hasardeuse selon le degré de détail dans la désignation commerciale (une désignation comme « Jus multifruit, melon, pommes, poires, bananes, ananas, sans sucres ajoutés, sans gluten » correspondrait bien rarement à « Jus multifruit » dans le référentiel FoodEx à cause des nombreux mots-clefs ajoutés à partir du reste de la désignation). Plusieurs décisions ont donc été prises pour trouver la meilleure correspondance possible.

Le **prétraitement d'une désignation** devait non seulement ajouter des mots-clefs mais également **modifier et enrichir cette même désignation afin de permettre des comparaisons plus pertinentes** (Notre « Jus multifruit, melon, pommes, poires, bananes, ananas, sans sucres ajoutés, sans gluten » deviendrait simplement « Jus multifruit » après prétraitement).

J'ai donc **personnellement enrichi le tableau Excel de plusieurs colonnes** dans le but d'éviter une comparaison entre un aliment et 2000 autres et plutôt limiter la sélection d'aliments à comparer à ceux de catégories, afin d'avoir des comparaisons entre désignations plus pertinentes.

termCode	masterParentCode	termExtendedName	Translatio	Désignation ajustée pour le code	CATEGORIE 1	CATEGORIE 2	PAR DEFA	PAR DEFA	MOTS-CLEFS 1	MOTS-CLEFS 2
A.01.000000	root	FOODEX1 terms								
A.01.000098	A.01.000001	Bread and rolls	Pain et petits	Pain et petits pains	PAIN				extrudés?	
A.01.000099	A.01.000098	Wheat bread and rolls	Pain et petits	Pain et petits pains de blé	PAIN	PAIN DE BLE		X		complet, pains? aux
A.01.000100	A.01.000099	Wheat bread, white	Pain de blé, b	Pain de blé, blanc	PAIN	PAIN DE BLE				
A.01.000101	A.01.000099	Wheat bread, white, gluten free	Pain de blé, b	Pain de blé, blanc, sans gluten	PAIN	PAIN DE BLE				
A.01.000102	A.01.000099	Wheat bread, white, with oil seeds	Pain de blé, b	Pain de blé blanc aux graines oléagineu	PAIN	PAIN DE BLE				
A.01.000103	A.01.000099	Wheat bread, brown	Pain de blé, b	Pain de blé, brun	PAIN	PAIN DE BLE				
A.01.000104	A.01.000099	Wheat bread, brown, gluten free	Pain de blé, b	Pain de blé, brun, sans gluten	PAIN	PAIN DE BLE				
A.01.000105	A.01.000099	Wheat bread, brown, with oil seeds	Pain de blé, b	Pain de blé, brun, aux graines oléagineu	PAIN	PAIN DE BLE				

Figure 13 : Tableur modifié du référentiel FoodEx2

J'ai ainsi ajouté des **catégories et sous-catégories** d'aliments, qui avaient chacune **un aliment par défaut** (dans le cas où un aliment correspondrait à une catégorie mais trop peu à un aliment se trouvant dedans).

De même, les désignations FoodEx2 n'étaient pas toujours les plus pertinentes à comparer à un aliment : par exemple, « Boissons au cola, caféiniques, faibles en calories » correspondrait à du « Coca Light » ou du « Cola Light », et « Lait de vache, plus de 4 % de matières grasses (y compris le lait des îles Anglo-Normandes) » qui correspond plutôt à du « Lait enrichi ». J'ai donc ajouté une colonne de manière à conserver les désignations originales mais avec des désignations **plus proches de la réalité afin de faciliter la comparaison entre désignations.**

Ce qui est précisément le dernier ajout à ce programme : pour **comparer les désignations**, j'ai utilisé **SpaCy** [11]**.** Il s'agit d'une bibliothèque Python open-source de traitement de textes qui, si le bon modèle est fourni, peut comparer deux chaînes de caractères entre elles et renvoyer un score de correspondance [12]. Fonctionnant à partir d'un réseau neuronal, cette solution permet donc des comparaisons à partir de synonymes (Comme sur l'exemple ci-dessous en Figure 14 où « carottes » partage une forte correspondance sémantique avec les « légumes racines »), ce qui est ce que nous recherchons entre un élément FoodEx2 et une désignation d'article.

```
INFO - Requête reçue: POST http://localhost:8085/SpaCy/Compare_table
INFO - Taille du tableau pour "café moulu décaféiné" : 50
INFO - Meilleur choix actuel (valeur : 499) : dolce gusto lungo décaféiné
INFO - Meilleur choix actuel (valeur : 975) : café moulu arabica décaféiné
INFO - Meilleur choix actuel (valeur : 986) : café moulu décafeine
INFO - Meilleur choix actuel (valeur : 1000) : café moulu décaféiné
INFO - Meilleur choix (valeur : 1000) : café moulu décaféiné
INFO - Réponse envoyée: 200 OK
INFO - Requête reçue: POST http://localhost:8085/SpaCy/Compare_table
INFO - Taille du tableau pour "carottes" : 133
INFO - Meilleur choix actuel (valeur : 379) : légumes et produits végétaux (y compris les champignons)
INFO - Meilleur choix actuel (valeur : 717) : légumes racines
INFO - Meilleur choix actuel (valeur : 1000) : carottes
INFO - Meilleur choix (valeur : 1000) : carottes
INFO - Réponse envoyée: 200 OK
```

Figure 14 : Exemple d'utilisation de SpaCy

La meilleure correspondance était retenue. Si le score était faible, on choisissait l'aliment par défaut de la catégorie la plus proche de notre aliment.

Au bout du compte, au processus initial (Figure 12) s'ajoutait la recherche de correspondances par mots-clefs (renvoyant donc une liste plus réduite d'éléments à comparer entre eux avec SpaCy, économisant donc un temps considérable d'exécution) puis par comparaisons entre désignations, celle renvoyant le meilleur score étant choisie par la suite en guise de correspondance FoodEx2.

Cependant, j'ai constaté que ce processus pouvait encore être optimisé, notamment en améliorant les prétraitements.

Tests unitaires et enrichissement

Les tableaux de prétraitements fonctionnaient de la façon suivante :

Expression régulière	Texte de remplacement / Mots clefs	Type
(.*) Amande (amère|douce)	\1 Amande	Enrichissement
1er age	1er âge, infantile	variété, recette, morceau, état
2e(me)? age	2e âge, infantile	variété, recette, morceau, état
3 g l	3gl	variété, recette, morceau, état
6g l	6gl	variété, recette, morceau, état
80 20	80_20	variété, recette, morceau, état
90 10	90_10	variété, recette, morceau, état
gourdes?	à boire, gourde	variété, recette, morceau, état
aux? poulets?	à la viande, au poulet	Enrichissement

Figure 15 : Tableau de prétraitement

La première colonne était dédiée à une expression régulière recherchée dans une désignation, la seconde des mots clefs à ajouter à la désignation pour enrichir le contexte pour la recherche (ou à remplacer l'expression régulière trouvée dans le cas d'un « Enrichissement » par exemple), et la troisième au type de prétraitement.

Partant de là et d'un **ensemble de tests unitaires**[12] progressivement incrémentés, mon travail consistait à ajuster les règles de prétraitements afin de maximiser le nombre de tests réussis : si « Pommes bio dorées » ne renvoyait pas la référence FoodEx liée à la désignation « Pomme » mais plutôt « Compotes de pommes bio » par exemple, alors il fallait ajouter un prétraitement qui ferait correspondre les 2. En répétant le processus et en enrichissant le tableau de prétraitements, les résultats s'affinaient de plus en plus et la façon de prétraiter le texte s'améliorait également.

En plus d'améliorer la précision des recherches, ces nombreux tests m'ont permis **d'améliorer le temps d'exécution** (notamment par l'ajout d'une **mise en mémoire cache**[13] du référentiel FoodEx prétraité par exemple, réduisant l'initialisation d'une minute par exécution à quelques secondes).

Figure 16 : Tests unitaires de correspondance FoodEx

[12] En programmation, un test unitaire correspond à une vérification du bon fonctionnement d'un programme à partir d'une valeur attendue selon une donnée à ce même programme [50]. Dans notre cas, le test consiste en ajouter une désignation et vérifier qu'on obtient le bon code en retour.

[13] La mise en cache consiste en un enregistrement temporaire de données limitant le temps d'accès futurs à ces mêmes données [51]. Ici, les objets traités sont simplement stockés sous forme de texte du côté serveur et récupérés, au lieu de refaire le traitement, si aucun fichier Excel n'a été modifié entretemps.

La limite principale de cette approche est que la qualité de la correspondance dépend de l'exhaustivité des règles définies. Afin de permettre à un utilisateur de modifier facilement ces règles et d'en ajouter de nouvelles, j'ai adapté le **programme de telle sorte que l'utilisateur n'ait pas** à retoucher le code de l'application.

En l'état, tout ce qui est à rajouter pour élever ce taux, ce sont des prétraitements dans le tableur de prétraitement, des lignes dans le référentiel FoodEx pour ajouter des désignations à différents codes ou encore modifier les mots-clefs des catégories. Tout le travail de programmation est déjà terminé, n'importe qui pourrait donc reprendre ce projet sans toucher à une ligne de code et un manuel de maintenance a été rédigé à cet effet.

Facettes, ingrédients et résultats

La correspondance entre un aliment et sa désignation étant faite, il manquait un élément complémentaire qui vient enrichir la désignation FoodEx : ses **facettes**. En effet, lorsqu'on cherche une désignation dans ce référentiel, on peut y ajouter des **facettes qui permettent de décrire les aliments de manière plus précise.**

Par exemple, de nouveau sur notre « Jus multifruit, melon, pommes, poires, bananes, ananas, sans sucres ajoutés, sans gluten », la correspondance se ferait à l'aliment qui désigne un « Jus multifruit ». Mais on peut préciser ce résultat avec des facettes, comme le fait qu'il soit sans sucres ajoutés et sans gluten.

Concrètement, le résultat d'une telle recherche serait le suivant (affiché à l'aide d'une page de recherche de code FoodEx2 créée à cet effet durant le stage) :

Figure 18 : Recherche de jus multifruit (avec facettes)

« **A.01.001452** » désigne ici le **jus multifruit** lui-même. Les facettes se composent du numéro (F04, F08, F10, etc.) correspondant au type de facette suivi d'un identifiant unique, propre à cet élément. La première facette est précédée d'un « # » et la séparation entre chacune d'entre elle sera un « $ » (en accord avec la convention de codage des aliments définie par l'EFSA)

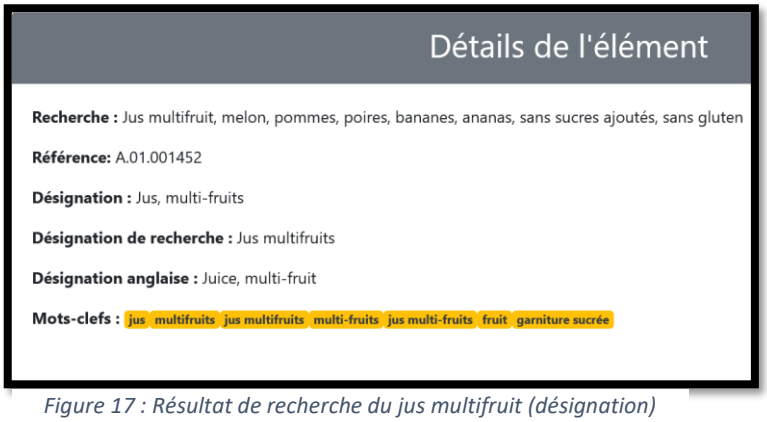


Figure 17 : Résultat de recherche du jus multifruit (désignation)

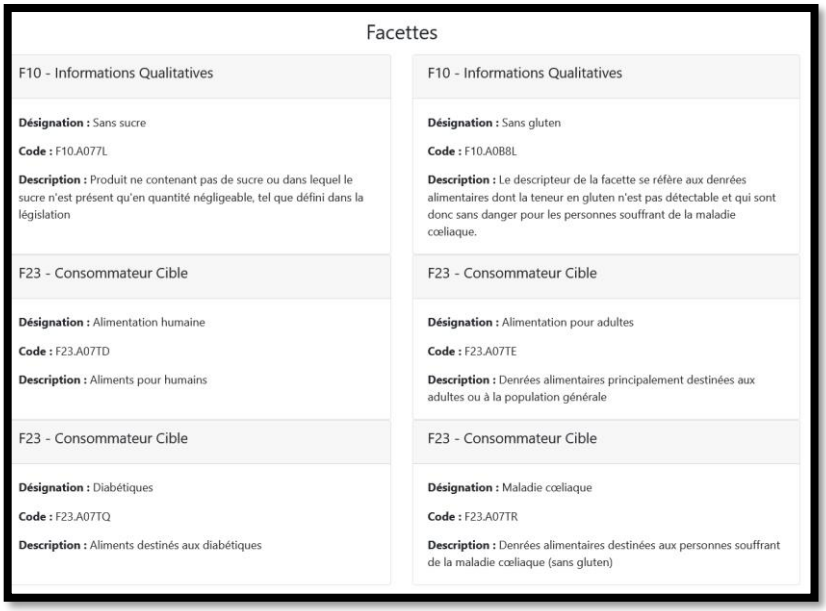


Figure 19 : Résultat de recherche du jus multifruit (Facettes)

Puis les différentes facettes (Figure 20 ci-contre) qui indiquent que cet aliment est bel et bien **sans sucre et sans gluten**.

On notera l'ajout des **facettes de consommateur cible** qui serviront à savoir quels aliments pourraient être faits pour qui, comme ce jus sans gluten et sans sucres ajoutés qui pourrait correspondre à des personnes diabétiques ou à maladies cœliaques.

Enfin, nous pourrons retrouver les facettes d'**ingrédients** (Figure 19 ci-contre) de ce jus, si indiqués dans la désignation, comme les différents fruits qui composent cet aliment multifruit dans notre cas. Chacune de ces désignations étant reprise du référentiel FoodEx2.

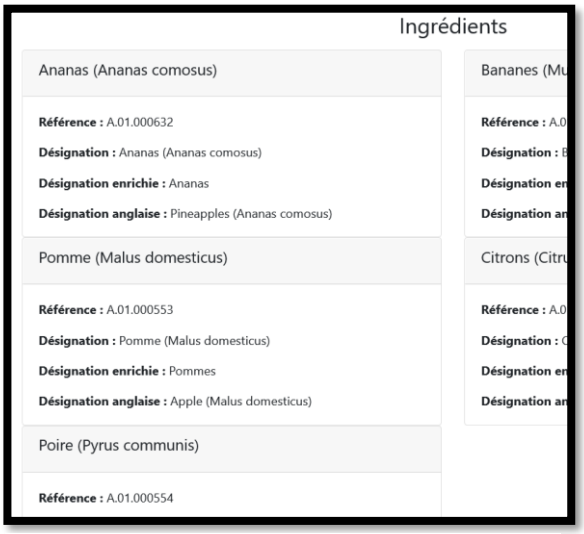


Figure 20 : Résultat de recherche du jus multifruit (ingrédients)

La recherche de facette et d'ingrédients retourne donc ce dernier résultat et le **code au complet est celui recherché initialement par ChemoSens**.

Pour ce qui est de la **recherche d'ingrédients**, elle se faisait à partir de simples séparateurs (« aux », « et », « avec », etc.) qui permettaient de découper la désignation en plusieurs parties : par exemple « Compote de poires et abricots » renvoyait une liste d'ingrédients de « poires » et d'« abricots ». À partir de là, il suffisait de rechercher les références de ces ingrédients de la même façon que vue plus tôt.

La **recherche de facettes** a nécessité plus d'approfondissements : il existe un total de 32 facettes à ce jour, chacune désignant un aspect de la désignation qui pourrait être enrichi (la facette 1 indique la source de l'aliment, la 4 ses ingrédients, la 15 la technique de préservation ou encore la 25 pour les ingrédients à risque microbiologique) [13].

Ainsi, il me fallait d'abord importer ces facettes (ou du moins, les plus pertinentes d'entre elles et celles qu'il m'était possible d'ajouter dans les temps impartis, certaines faisant plus de 2000 lignes par exemple) et enrichir à mon tour un tableau dans lequel l'application venait les rechercher.

termCode	termExtendedName	termExtendedName(FR)	Mots-clefs	termScopeNote	termScopeNote(FR	Category	Catégorie
F11.A07GD	99 % alcohol v/v	99 % alcool v/v	99%? (.*)?(alcools?|alc/vol|degré|(de)?ten	The food item has an alcohol c	La denrée alimentaire a	F11 - Alcohol Content	F11 - Taux d'alcoolémie
F11.A07GE	100 % alcohol v/v	100 % alcool v/v	100%? (.*)?(alcools?|alc/vol|degré|(de)?ter	The food item has an alcohol c	La denrée alimentaire a	F11 - Alcohol Content	F11 - Taux d'alcoolémie
F13.A07GF	Blanching	Blanchi	Blanchie?s?, blanchiements?	Process consisting of heat trea	Processus consistant en	F13 - Cooking Method	F13 - Méthode de préparation
F13.A07GG	Cooking in water	Cuisiné à l'eau	à l'eau, bouillie?s?	Boiling, Poaching of products (	Ébullition, pochage de p	F13 - Cooking Method	F13 - Méthode de préparation
F13.A07GH	Poaching	Pochage	Pochée?s?	Cooking in boiling liquid, espec	Cuisson dans un liquide	F13 - Cooking Method	F13 - Méthode de préparation
F13.A07GJ	Simmering	Mijoté	Mijotée?s?, mijotages?	Cooking gently, near or just be	Cuisson douce, proche	F13 - Cooking Method	F13 - Méthode de préparation

Figure 21 : Tableau des facettes FoodEx2

Les codes, noms et descriptions en anglais étant fourni, il m'a fallu ajouter les traductions de ces derniers, les catégories associées ainsi que des mots-clefs qui serviraient à les récupérer. C'était majoritairement un travail de traduction et de recherche de mots pertinents à associer à toutes ces facettes mais cette feuille Excel étant maintenant en place, y ajouter ou modifier des facettes reste très accessible et aisé d'utilisation.

Le processus final de cette recherche FoodEx est donc le suivant :

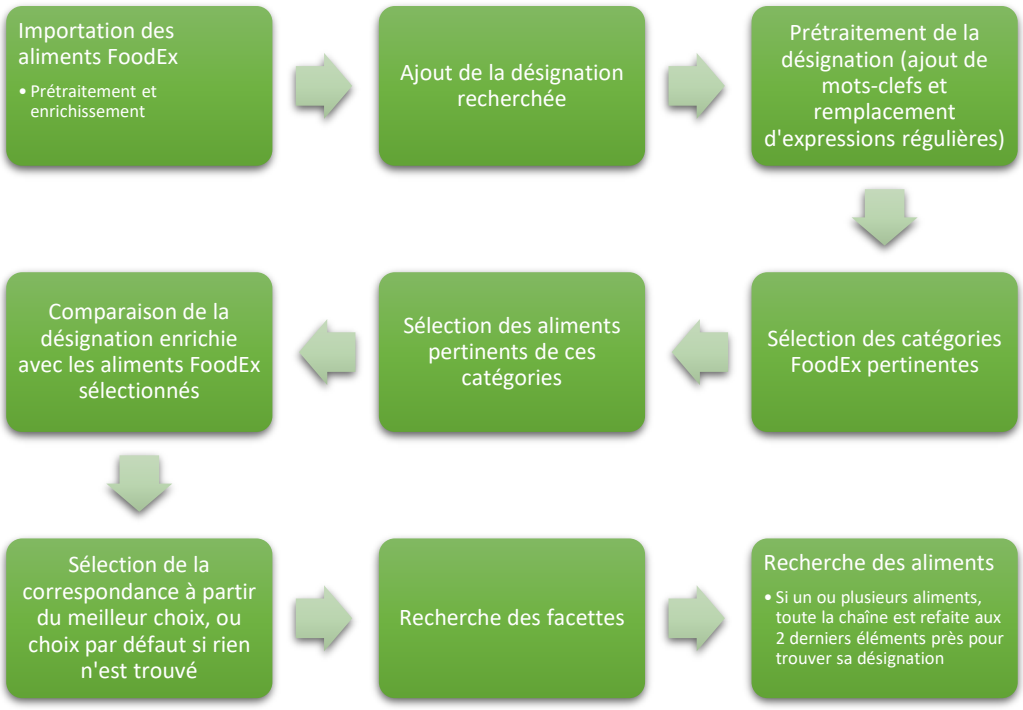


Figure 22 : Processus final de correspondance FoodEx

b. Correspondance Open Food Facts

La dernière étape pour enrichir notre désignation consistait à **récupérer des données sur Open Food Facts** [14]. Il s'agit d'une **base de données en ligne répertoriant énormément d'aliments ajoutés par des consommateurs**, incluant notamment les valeurs nutritionnelles et autres informations disponibles sur l'emballage.

Une telle ressource est précieuse au projet actuel car si FoodEx2 permet de standardiser les noms d'aliments, Open Food Facts permet d'obtenir des informations sur la qualité nutritionnelle et environnementale des aliments dans une facture.

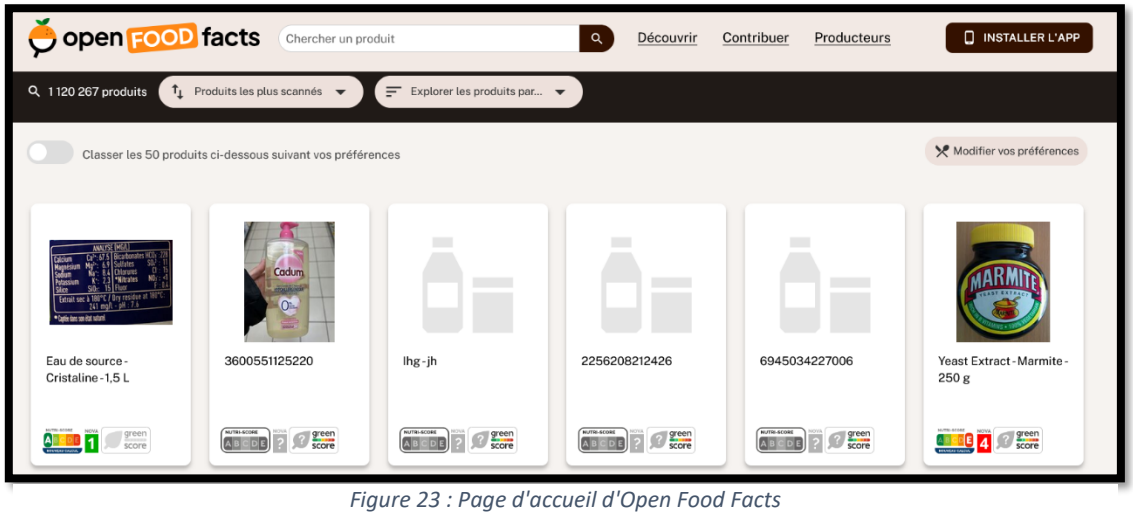


Figure 23 : Page d'accueil d'Open Food Facts

L'API gratuite d'Open Food Facts permettait de retrouver ces différentes informations en fournissant une désignation commerciale. (Exemple en Figure 24 ci-dessous).

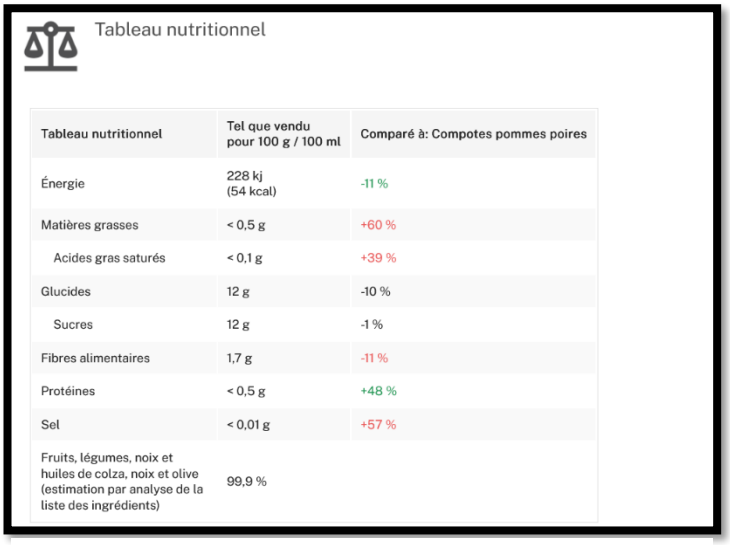


Tableau nutritionnel

Tableau nutritionnel	Tel que vendu pour 100 g / 100 ml	Comparé à: Compotes pommes poires
Énergie	228 kj (54 kcal)	-11 %
Matières grasses	< 0,5 g	+60 %
Acides gras saturés	< 0,1 g	+39 %
Glucides	12 g	-10 %
Sucres	12 g	-1 %
Fibres alimentaires	1,7 g	-11 %
Protéines	< 0,5 g	+48 %
Sel	< 0,01 g	+57 %
Fruits, légumes, noix et huiles de colza, noix et olive (estimation par analyse de la liste des ingrédients)	99,9 %	

Figure 25 : Exemple de données trouvables sur Open Food Facts

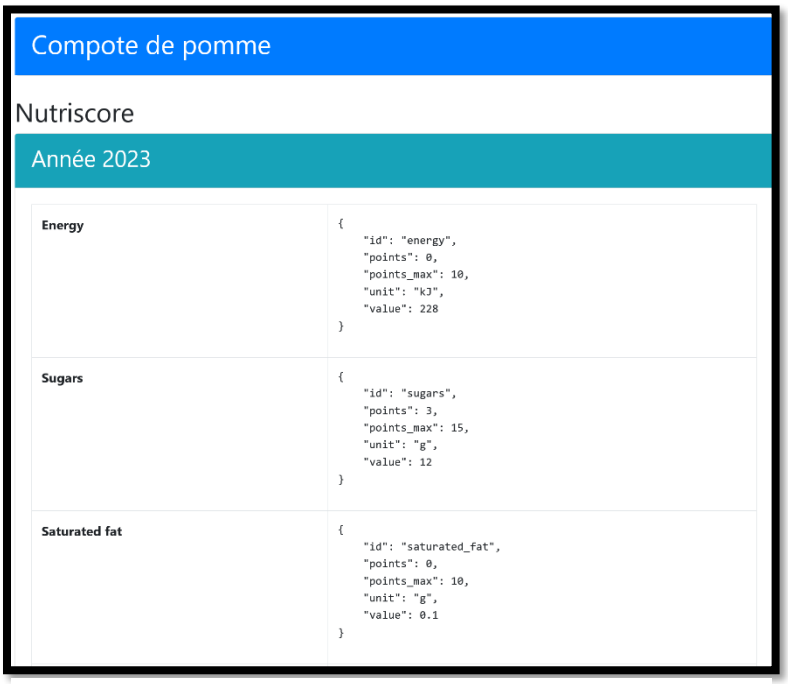


Figure 24 : Exemple de résultats retournés par l'API Open Food Facts

c. Extraction de jeux de données

Nous avions finalement besoin d'un jeu de données pertinentes incluant un grand nombre de désignations commerciales pour tester les différents outils développés, c'est pourquoi j'ai passé deux semaines de mon stage à développer un code **collectant des données relatives aux aliments vendus sur le site web Auchan.fr à l'aide de web-scraping**[14] (trois dernières cases dans la Figure 26 ci-contre).

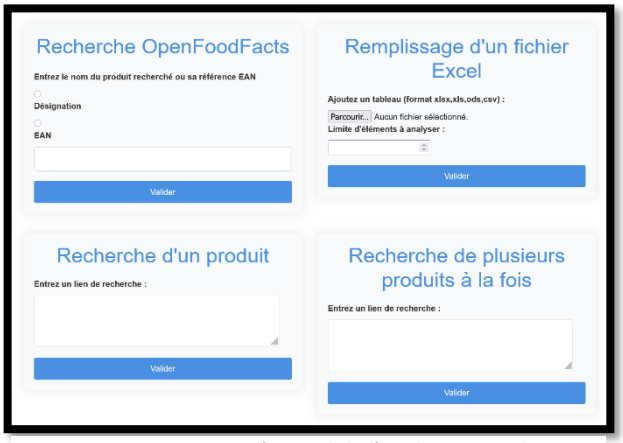


Figure 26 : Page d'accueil de l'application web

À l'aide de différents outils de web-scraping (tels que Puppeteer ou Simple HTML DOM), récupérer les données n'était pas une difficulté, il était cependant nécessaire de les récupérer judicieusement, de sorte que le site web ne bloque pas le programme à la suite d'une surcharge de requêtes d'une même adresse IP. Ainsi, il fallait mettre en place un temps d'attente entre la récupération de différentes pages

L'extraction est donc implémentée et demande un temps d'exécution considérable mais devrait, finalement, remplir environ 50 000 lignes de données alimentaires uniques.

Nom du produit	Marque	Dénomination légale de vente	Référence / EAN	Pictog	Ingrédients	Note	Valeu	Valeur é	Mat	dont	Protéines	Sel	Fibres Alimentaires
Maquereau vidé	LA MARÉE I	Maquereau vidé.	365954/203050365	Frais	MAQUEREAUX	5.0/5	807 kJ	194 kcal	14 g	3,2 g	18 g	0,16 g	
Sole vidée avec tête pièce	LA MARÉE I	Sole vidée avec tête.	324344/203050324	Frais	SOLE (Soleavul	4.6/5	328 kJ	77 kcal	0,6 g	0,1 g	18 g	0,19 g	
Sardines péché en Atlantique Nord E	LA MARÉE I	Sardines.	363345/203050363	Frais	SARDINE (Sard	5.0/5	682 kJ	163 kcal	9,5 g	2,4 g	20 g	0,22 g	
CULTIVONS LE BON Saumon Atlanti	LA MARÉE I	Saumon Atlantique Ecosse vidé av	305492/203050305	Frais	SAUMON Atlar	4.0/5	674 kJ	162 kcal	10 g	1,8 g	17 g	0,13 g	
Tacaud entier vide	LA MARÉE DU JOUR		648570/203050648	Frais		5.0/5							
Soles vidées avec tête	LA MARÉE DU JOUR		380435/203050380	Frais		5.0/5							
Sardines pêchées en Atlantiques Nor	LA MARÉE I	Sardines.	361457/203050361	Frais	SARDINE (Sardi	N	682 kJ	163 kcal	9,5 g	2,4 g	20 g	0,22 g	
Morue salée séchée 16/20	LA MARÉE DU JOUR		383269/203050383	Frais		5.0/5							
Sardines péchées en Atlantique Nor	LA MARÉE I	Sardines.	843881/203050843	Frais	SARDINE (Sardi	5.0/5	682 kJ	163 kcal	9,5 g	2,4 g	20 g	0,22 g	
Rouget barbet vidé et gratté avec tê	LA MARÉE I	Rouget-barbet vidé et gratté .	325725/203050325	Frais	ROUGET (Mull	N	659 kJ	158 kcal	9,4 g	2,7 g	18 g	0,17 g	
Merlu entier	LA MARÉE I	Merlu entier.	396656/203050396	Frais	MERLU (Merlu	N	349 kJ	83 kcal	1,4 g	0,2 g	18 g	0,22 g	
Dorade Grise entière pièce de 300 à	LA MARÉE I	Dorade grise entière.	984395/203050984	Frais	DORADE grise	N	551 kJ	131 kcal	5,3 g	1,4 g	21 g	0,15 g	1 g
Maquereaux entier	LA MARÉE I	Colis de maquereaux 3KG.	381247/203050381	Frais	MAQUEREAUX	N	807 kJ	194 kcal	14 g	3,2 g	18 g	0,16 g	

Figure 27 : Tableur de données Auchan rempli

[14] Le web-scraping [62] est une méthode de récupération, d'extraction de données d'un site web à l'aide d'un script.

IV RETROSPECTIVE DU PROJET

IV.1 LOGICIELS ET OUTILS UTILISES

a. Logiciels utilisés

Visual Studio Code

Figure 28 : Logo de Visual Studio Code

Visual Studio Code [15] est un Environnement de Développement Intégré (EDI)[15] très modulable grâce à ses nombreuses extensions proposées par différents développeurs. Dans le cadre d'une application web, donc de développement PHP, cet outil est idéal et plusieurs packs d'extensions facilitent grandement son utilisation et la détection d'erreurs.

Il en va de même pour d'autres langages de programmation utilisés en web tels que CSS, JavaScript, HTML (bien qu'ici, l'affichage des pages soit entièrement géré en PHP), ou encore les simples fichiers textuels, incluant les fichiers JSON, dont la prise en main est plus agréable sous Visual Studio Code.

Microsoft Excel

Figure 29 : Logo de Microsoft Excel

Ayant eu à réorganiser, rempli et trier de nombreux tableaux utilisés par mon programme, ce tableur [16] était essentiel et présente beaucoup d'avantages le rendant agréable d'utilisation : il est fluide, agit rapidement en cas de recherche ou de remplacements groupés, permet des tris efficaces, etc.

N'ayant pas fait de tâches plus compliquées que de remplir, trier et réorganiser, et étant déjà très familier à **Excel**, ma maîtrise du logiciel n'aura pas nécessairement beaucoup évolué au cours de ce stage. Il reste cependant bon de conserver ces habitudes et cette maîtrise.

Windows PowerShell

Figure 30 : Logo de PowerShell

Plusieurs fois au cours de ce stage, il était nécessaire d'utiliser une invite de commande et **PowerShell** [17] était un très bon choix pour tout ce qui concernait la mise en place d'un serveur Python, allant de son installation à son utilisation.

Ceci impliquait donc de créer un **environnement de travail Python séparé**, de mettre en place ses bibliothèques, gérer d'éventuelles erreurs d'installation ou incompatibilités de versions, etc. Une fois ceci fait, je devais trouver un moyen d'allumer et d'éteindre ce serveur indépendamment de mon programme ou de sorte que le processus s'exécute en arrière-plan.

Ce sont des compétences acquises durant ce stage et qui renforcent mes connaissances dans un domaine qui m'était moins familier que la programmation en elle-même.

[15] Un Environnement de Développement Intégré est un outil optimisé pour la programmation : intégrant compilateur, débogueur, éditeur de texte plus détaillé, etc. [54]

Windows Subsystem for Linux (WSL)

Figure 31 : Logo de WSL

Nombre de commandes utiles n'existent que sous système **Linux** et pour y accéder depuis Windows, une très bonne alternative est **WSL** [18]. Installable sous **invite de commande**, il permet d'exécuter des fichiers qui ne le seraient pas sous Windows en temps normal, ce qui m'a notamment permis d'effectuer la **conversion d'un fichier PDF en ou plusieurs images**, le **nettoyage** et la **lecture de ces mêmes images**, ainsi que **l'exécution locale de modèles LLM** à l'aide d'Ollama.

C'est également un outil que je maîtrisais déjà à un niveau correct avant d'arriver en stage et **continuer de me familiariser avec me permet progressivement d'atteindre un très bon contrôle** de ce dernier.

Apache

Figure 32 : Logo d'Apache

Afin d'héberger localement cette application, dans le but de pouvoir la tester en attendant de la déployer sur les serveurs de ChemoSens, **Apache** était mon premier choix. L'ayant déjà utilisé durant le projet semestriel précédant ce stage, l'utilisation et la configuration de ce serveur web m'étaient familiers et ces deux mois m'ont permis de beaucoup faire évoluer ma maîtrise de ce dernier.

En l'état, le projet ne nécessite que **très peu de changement dans la configuration** de l'application. Cependant, au cours du développement du projet, et notamment quand j'ai souhaité héberger un serveur Python en plus du serveur web, j'ai passé beaucoup de temps sur sa configuration sans que ça n'aboutisse à quelque chose de fonctionnel malheureusement. Mais une telle expérience de gestion d'erreurs et d'exceptions engendrées par ce travail m'a permis **d'en apprendre énormément de la configuration de serveurs web**.

Github

Figure 33 : Logo de Github

Pour conserver un suivi du projet et en avoir des sauvegardes, **Github** [19] aura été l'outil parfait. Cet outil d'**hébergement web et de développement de logiciels** permet la sauvegarde de différents projets et leur suivi. Une fois un projet créé déposé sur ce service et à la suite de suffisamment de modifications mises en ligne sur ce dernier, un accès aux anciennes versions du projet reste disponible et permet donc un suivi très simplifié ou des retours en arrière si besoin [20].

Dans mon cas, étant majoritairement seul à travailler sur ce projet, aucun souci de conflit (si deux personnes modifient le même fichier par exemple) n'était de mise, mais ce projet m'a permis de **continuer de me familiariser à cet outil et à ses différentes fonctionnalités**, telles que les branches, l'affichage de manuels (Readme), etc.

b. Outils PHP

Mon application étant **codée en immense partie en PHP**, j'ai également eu l'occasion de me familiariser à plusieurs de ses bibliothèques ou outils implémentés dans le langage-même dans le but de donner des solutions aux différentes demandes de mon maître de stage pour ce projet.

PHPSpreadSheet

PHPSpreadSheet [21] est une bibliothèque disponible sous **PHPOffice** [22] (ensemble de bibliothèques PHP permettant un accès à plusieurs types de documents, dont Word, PowerPoint, Vision, …) permettant un **accès par code PHP à des tableaux Excel**.

Cette bibliothèque m'a notamment permis la **lecture de tous les tableurs** de prétraitement et de référentiels lors de la recherche de correspondances FoodEx et **l'écriture de tableurs existants** lors de l'extraction de données d'Auchan. Je ne connaissais aucunement cette bibliothèque et ai su en obtenir une bonne maîtrise pour les commandes les plus simples (trouver une cellule, prendre une valeur, en attribuer une à une cellule, gérer différentes feuilles et tailles de tableaux, etc.).

Simple HTML DOM

Pour ce qui est du web-scraping effectué en début de stage, **Simple HTML DOM** [23] permettait une lecture très simple et organisée de pages HTML. Ne le connaissant pas d'origine, sa prise en main m'a pris un certain temps mais en l'utilisant j'ai acquis suffisamment d'expérience pour en obtenir une bonne maîtrise sur le long terme.

PHP JWT (Firebase)

JSON Web Token (JWT) [24] est une **méthode d'échange de jetons entre un utilisateur et un serveur** : cette sécurité permet de vérifier l'authenticité de données et d'éviter le passage multiplié de clefs d'API ou de mots de passe par exemple. Dans notre cas, cette méthode est utilisée dans le cadre d'une API : une fois connecté une première fois à l'application (que ce soit par nom d'utilisateur ou par clef API), un JWT est renvoyé à l'utilisateur et ce dernier l'utilisera afin de se reconnecter pendant un temps limité (une heure par exemple).

C'est donc PHP-JWT de Firebase [25] qui aura été utilisé pour implémenter un tel système, l'implémentant de façon efficace et optimisée, il m'a permis de facilement aborder ce milieu qui m'était totalement inconnu jusqu'alors et en avoir connaissance me sera désormais très utile dans le cadre de futures programmations d'API.

c. Outils JavaScript

Enfin, plusieurs outils en JavaScript auront été nécessaires pour des interactions plus visuelles, mais également dans le cadre de web-scraping de données utilisées en guise de test.

Konva

Figure 34 : Logo de Konva

Konva.js [7] est un Framework[16] JavaScript rendant possible **l'interaction avec des canvas[17] sur des pages web** [26]. Dans le cas de cette application, une seule fonctionnalité m'importait : **ajouter, recadrer et éventuellement supprimer des rectangles** sur chaque page des factures afin de cacher des données personnelles. Je n'ai qu'effleuré la surface du vaste catalogue proposé par cette application (filtres visuels tels que l'inversion de couleur, flou, bruit, animations, drag and drop d'images, etc.) mais je suis malgré tout sensibilisé à son utilisation à la suite de ce stage.

Node.js

Figure 35 : Logo de Node.js

Node.js [27] **est un environnement d'exécution** JavaScript, permettant le développement d'applications web en temps réel, de serveurs web ou de backend en JavaScript. Il est donc plutôt adapté aux applications nécessitant des interactions fréquentes (chat, notifications, réactions, etc.) ou le chargement de données dans une application web, comme dans le cas présent. Son utilisation s'est limitée à **l'ajout de Puppeteer à l'application**, étant donné que l'objectif n'était pas d'architecturer le projet autours de ce dernier.

Puppeteer

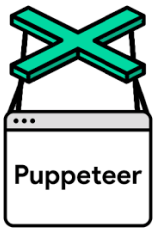


Figure 36 : Logo de Puppeteer

Puppeteer [28] **est une bibliothèque JavaScript**, installable à l'aide de Node.js, offrant une **interaction directe avec Google Chrome ou Mozilla Firefox à travers du code** (soumission de formulaires, créer des screenshots ou PDF de pages, déplacements à la molette, etc.). Avec un tel outil, il devient possible **d'automatiser et de faciliter des actions de web-scraping**.

Dans le cadre de ce projet, il était nécessaire afin de charger les pages de recherches d'aliments d'Auchan.fr en simulant le déplacement de la barre de défilement jusqu'en bas de la page afin de récupérer chacun des aliments de cette recherche. J'ai donc pu être introduit à cette bibliothèque dans un contexte très simple et, combinée à l'utilisation de la bibliothèque PHP Simple HTML DOM, **je suis désormais capable d'effectuer du web-scraping en simulant des actions de l'utilisateur** (déplacements à la molette irréguliers et espacés dans le temps, temps d'attente entre 2 ouvertures de pages, etc.)

[16] En informatique, un Framework est un ensemble de composants logiciels servant à obtenir les fondations ou grandes lignes d'un logiciel [56] . Dans le cas de Konva, les composants permettant d'interagir avec une image sont en place, il ne reste qu'à implémenter comment on les utilise dans notre application.

[17] En HTML, un canvas permet le rendu d'images à l'aide de scripts (dont JavaScript par exemple) de façon dynamique à l'aide de bitmaps (où chaque pixel est modifiable, offrant donc des interactions comme utilisées par Konva) [57]

IV.2 BILAN TECHNIQUE ET COMPETENCES

Ce stage m'a permis d'**approfondir de nombreuses compétences acquises en IUT** pour développer une application concrète et réelle, mais aussi d'en **obtenir et développer de nouvelles** que j'ai bien l'intention de remettre à contribution à l'avenir.

a. C1 : Réaliser un développement d'application

Le principal objectif de ce stage était **d'implémenter différentes fonctionnalités dans une application** (lecture de la facture par LLM, correspondance FoodEx2, web-scraping, etc.). Bien qu'il ne fût pas attendu que j'implémente toutes les fonctionnalités souhaitées par mon maître de stage pour un stage de courte durée, j'ai su **toutes les implémenter durant ces 8 semaines tout en conservant un code qualitatif,** et en le rendant **facile à améliorer, modifier ou reprendre à l'avenir**. Le **code** est **documenté** et **organisé sous une structure MVC avec Services** (voir figure 37 ci-dessous) de sorte que **chaque couche ait sa responsabilité** et puisse être modifiée si besoin.

Des **interfaces** sont également utilisées pour ce qui est des Services (servant de couche de liaison entre les Controllers et les Utils) et des Utils (couche d'interaction directe avec le serveur), implémentées par différentes **fabriques abstraites** se trouvant dans le package General/Init (en bas sur la figure 37 ci-dessous), notamment pour le choix du LLM par exemple qui ne nécessitera, en cas de changements, qu'une création de classe implémentant la même interface (I_LLMInteraction) que celle actuellement utilisée.

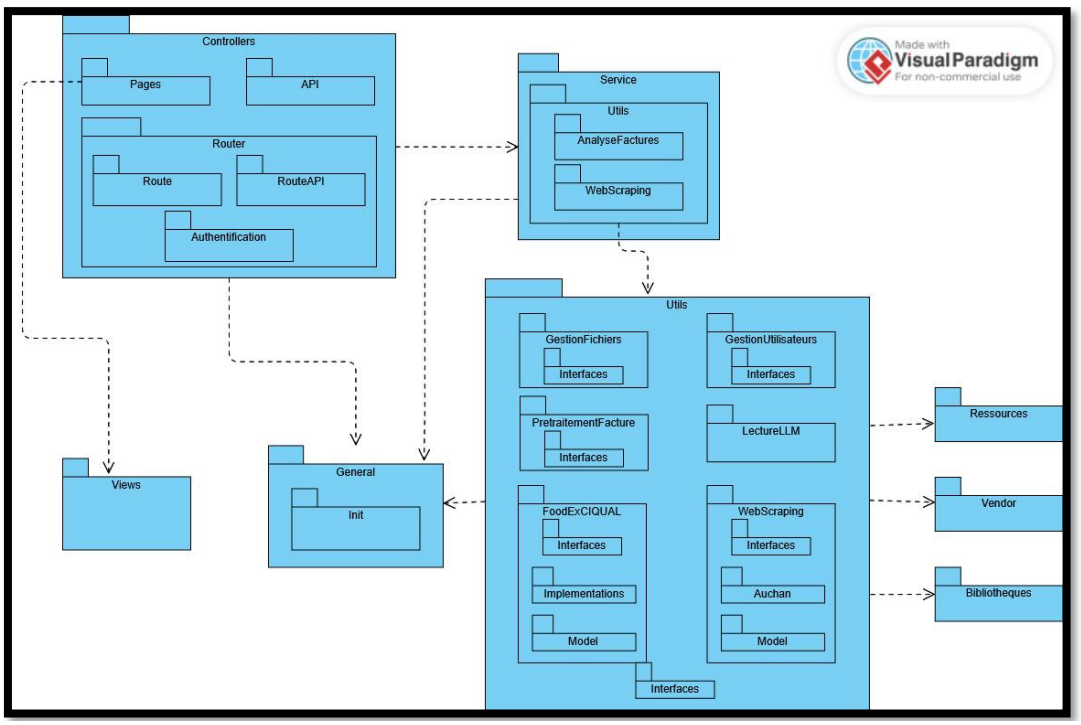


Figure 37 : Architecture du projet

Le site web, servant d'IHM avec cette application, utilise **Bootstrap**[18] pour gérer l'affichage, le rendant donc responsive et accessible à la fois (voir Annexe 2). Des **instructions sont également retranscrites à l'écran** pour aider à la prise en main de certains outils qui ne seraient pas abordables au premier abord, par exemple la page de prétraitement d'une image par un utilisateur, où différentes instructions sont données (voir figure 38 ci-dessous)

Figure 38 : Instructions de prétraitement d'images

Enfin, durant tout le projet, plusieurs **tests** ont été effectués pour **déterminer la qualité d'un LLM** à retranscrire une facture sous un format structuré, et de très **nombreux tests unitaires sont disponibles pour vérifier la qualité de la correspondance FoodEx2**, comme évoqué plus tôt. (Voir figure 39 ci-contre)

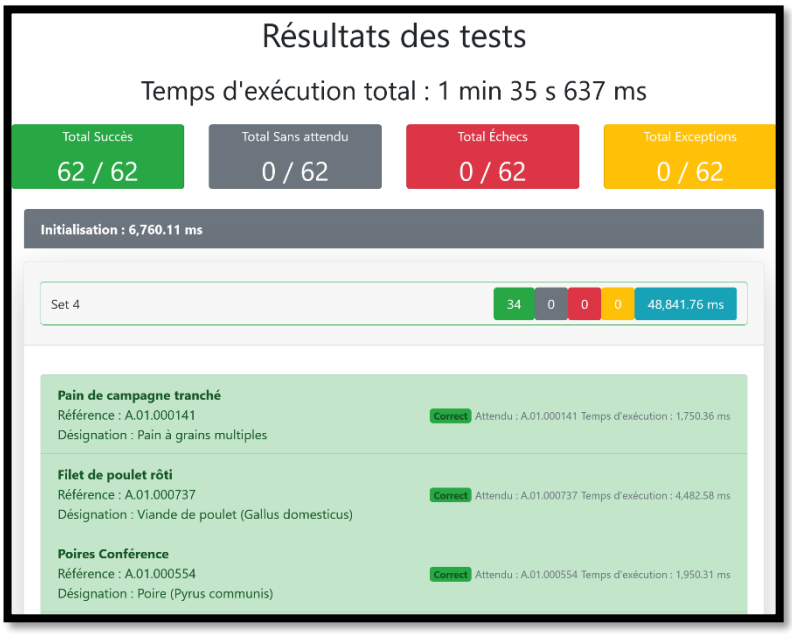


Figure 39 : Tests unitaires FoodEx2

[18] Bootstrap [58] est une collection d'outils web facilitant grandement la réalisation de l'affichage, en le gardant responsive (adapté, peu importe la taille de l'écran) et accessible.

b. C2 : Optimiser des applications

L'un des plus grands enjeux de ce programme était **d'optimiser son temps d'exécution** : si, par exemple, le choix des LLM entraînés ne fut pas retenu, c'était en partie en raison de leur temps d'exécution dépassant les 5 minutes.

De même, rechercher des correspondances FoodEx2 demandait un temps considérable d'exécution : à commencer par le **prétraitement de chaque aliment FoodEx2 à l'initialisation**, puis la **comparaison de chaque aliment à une désignation** donnée, et enfin les **appels au serveur SpaCy** qui, individuellement, prennent un temps considérable. Durant toute cette phase du projet, **j'ai su prendre plusieurs décisions visant à optimiser ce processus**. **Mettre en mémoire cache** les différents objets prétraités du référentiel FoodEx2 par exemple, servant notamment à éviter qu'ils ne soient prétraités à chaque exécution (ce qui renvoie concrètement toujours le même résultat) mais que l'on prenne simplement un objet stocké en cache de l'application à l'initialisation. Pour ce qui est des comparaisons, **la mise en place de catégories et la comparaison par mots-clefs** d'abord avec ces dernières, et ensuite avec chaque aliment se trouvant dedans, sert à limiter le plus possible le nombre d'aliments à comparer par SpaCy, de sorte que son appel soit le moins lourd possible.

Le **choix de la structure de données était imposé** par ChemoSens : il fallait travailler avec des **tableaux de données avec lesquels les chercheurs ont l'habitude de travailler**.

Ainsi, l'objectif était de **centraliser les modifications possibles dans des tableaux** en dehors du programme de sorte que quiconque reprendra le projet n'aurait pas une ligne de code à modifier. Une structure par BDD ou toute autre alternative aurait rendu cela beaucoup plus difficile pour quelqu'un qui n'aurait que très peu de connaissances en programmation SQL par exemple. La lecture et l'interprétation de tableurs Excel pourrait prendre un temps supplémentaire d'exécution, mais à l'aide de la mise en mémoire cache, ça ne sera nécessaire qu'à la suite de différentes modifications, rendant donc cette solution encore plus optimisée dans notre cas.

Sécuriser cette application était, de même, un enjeu important : plusieurs exécutions en invite de commandes sont effectuées ou différents fichiers sensibles sauvegardés sur le serveur. Ainsi, plusieurs failles de sécurité étaient possibles : une **injection de commande** ou dans le nom d'un fichier, qui pourrait **donner accès au serveur** ou bien **faire exécuter de dangereuses commandes**. Pour ce qui est de ces injections de commandes, **les paramètres seront toujours échappés et vérifiés au préalable** (à l'aide des méthodes propres au langage PHP). Les fichiers sauvegardés suivent toujours une nomenclature bien précise ne dépendant pas des noms de fichiers originels, rendant donc impossible une quelconque infraction à travers ce nom. Les paramètres de formulaires HTML seront également toujours vérifiés avant exécution du programme associé, ce qui empêche des injections par paramètres.

L'impact environnemental d'un tel programme est à prendre en considération : héberger 2 serveurs en parallèle (web et Python) est coûteux en performances et **il aurait été bien plus optimisé de tout centraliser sur le même langage de programmation**. Cependant, aucune bibliothèque aussi poussée que SpaCy n'était disponible en PHP, ayant donc forcé l'utilisation des 2 à la fois.

De plus, l'utilisation d'une IA générative est à prendre en considération : on estime **qu'une requête à ChatGPT coûte 6 à 10 fois plus d'énergie qu'une recherche internet**, ou encore qu'une courte conversation émettrait environ 30 grammes de CO2 [29]. Nous avons choisi un modèle moins coûteux en performances (gpt-4-turbo [30]) afin de réduire un tel impact et ChemoSens cherchera, à terme, à héberger son propre modèle pré-entraîné sur les résultats de ChatGPT de sorte que tout soit hébergé localement.

Bien que cette application ne vise pas une utilisation à très grande échelle, il n'empêche que **la méthode actuellement en place n'est pas la meilleure en termes d'impact climatique** et qu'un sérieux travail d'optimisation du processus pourrait donc être réalisé à l'avenir afin de trouver des alternatives.

c. C3 : Administrer des systèmes informatiques communicants complexes

L'application développée au cours de ce stage aura **plusieurs aspects communicants** à prendre en compte. Celle-ci mettant en place **à la fois un serveur web et à la fois un serveur Python**, la **communication entre les 2 et son optimisation** était l'un des principaux enjeux lors de l'optimisation du processus de correspondance FoodEx2, ce qui a majoritairement été résolu à travers **moins de requêtes pour des requêtes plus lourdes** afin d'optimiser le temps d'exécution. Différents **appels API** seront également effectués dans le processus principal de l'application tels que l'**appel à ChatGPT ou celui à Open Food Facts**, ce qui m'aura beaucoup appris des méthodes de requêtes et en partie permis de **développer ma propre API** sur cette même application. Cette dernière est également sécurisée à l'aide d'une authentification nécessaire ou de clef API demandée.

De plus, afin de **sécuriser les données de l'application**, les sauvegardes et fichiers d'utilisateurs seront toujours **restreints par un fichier .htaccess**[19], de sorte que n'importe qui ne puisse pas y accéder. Ainsi, seul le code en lui-même saurait y accéder et nul ne pourra s'y introduire simplement en changeant l'URL par exemple.

[19] Un fichier .htaccess [59] en est un de configuration définissant les règles de sécurité d'un serveur web, pouvant être propre à un dossier, limitant l'accès à différents scripts voire l'interdisant par exemple.

d. C4 : Gérer des données

Comme évoqué plus tôt, **le modèle de données choisi se présentait sous la forme de plusieurs tableaux Excel** (un pour le référentiel FoodEx2, un autre pour le prétraitement, etc.). Bien que mon travail fût de les utiliser à l'aide d'un programme, il demeurait nécessaire **d'optimiser leur agencement afin de maximiser l'efficacité** d'un tel programme. Pour cette raison, une grande partie du travail sur les correspondances FoodEx2 aura été de passer d'un référentiel ne contenant que des désignations et leur code associé à un tableau à catégories mieux structuré et aux désignations bien plus pertinentes en guise de comparaisons, et ce pour les 2000 lignes qui le composent.

termCode	masterParent	termExtendec	Translation
A.01.000000	root	FOODEX1 terms	
A.01.000001	A.01.000000	Grains and gra	Céréales et produits à base de céréales
A.01.000002	A.01.000001	Grains as crop	Les céréales comme cultures
A.01.000003	A.01.000002	Wheat grain c	Récolte de grains de blé
A.01.000004	A.01.000002	Barley grain ((	Grain d'orge (Culture)
A.01.000005	A.01.000002	Corn grain (Cr	Grain de maïs (Culture)
A.01.000006	A.01.000002	Rye grain (Cro	Grain de seigle (Culture)
A.01.000007	A.01.000002	Spelt grain (Cr	Épeautre (Culture)

Figure 41 : Tableur FoodEx2 avant modifications

termCode	masterParentCode	termExtendedName	Translatio	Désignation ajustée pour le code	CATEGORIE 1	CATEGORIE 2	PAR	PAR	MOTS-CLEFS	MOTS-CLEFS 2
A.01.000711	A.01.000701	Fruit compote, Cranberry (Vaccinium	Compote de	Compote de canneberges	PRODUITS FRUITIERS	COMPOTES				
A.01.000712	A.01.000701	Fruit compote, Pineapple (Ananas co	Compote de	Compote d'ananas	PRODUITS FRUITIERS	COMPOTES				
A.01.000713	A.01.000701	Fruit compote, Mixed fruit	Compote de	Compote mélange de fruits (multifruits	PRODUITS FRUITIERS	COMPOTES				
A.01.000714	A.01.000682	Candied fruits	Fruits confits	Fruits confits	PRODUITS FRUITIERS	FRUITS CONFITS		X		écorces? confites?,
A.01.000715	A.01.000714	Candied fruit, Cheery	Fruits confits	Joyeux Fruits confits	PRODUITS FRUITIERS	FRUITS CONFITS				
A.01.000716	A.01.000714	Candied fruit, Bananas	Fruits confits	Bananes confites	PRODUITS FRUITIERS	FRUITS CONFITS				
A.01.000717	A.01.000714	Candied fruit, Ananas	Fruits confits	Ananas confits	PRODUITS FRUITIERS	FRUITS CONFITS				

Figure 40 : Tableur FoodEx2 après modifications

De même, le **tableau de prétraitement**, originellement présenté sous la forme d'un simple tableau à 3 colonnes pour 5000 lignes, est désormais **séparé selon la catégorie de prétraitement** et agencé de sorte qu'une séparation des couches en PHP est possible (une classe pour une feuille de tableur), facilitant par la même occasion la reprise du travail de prétraitement.

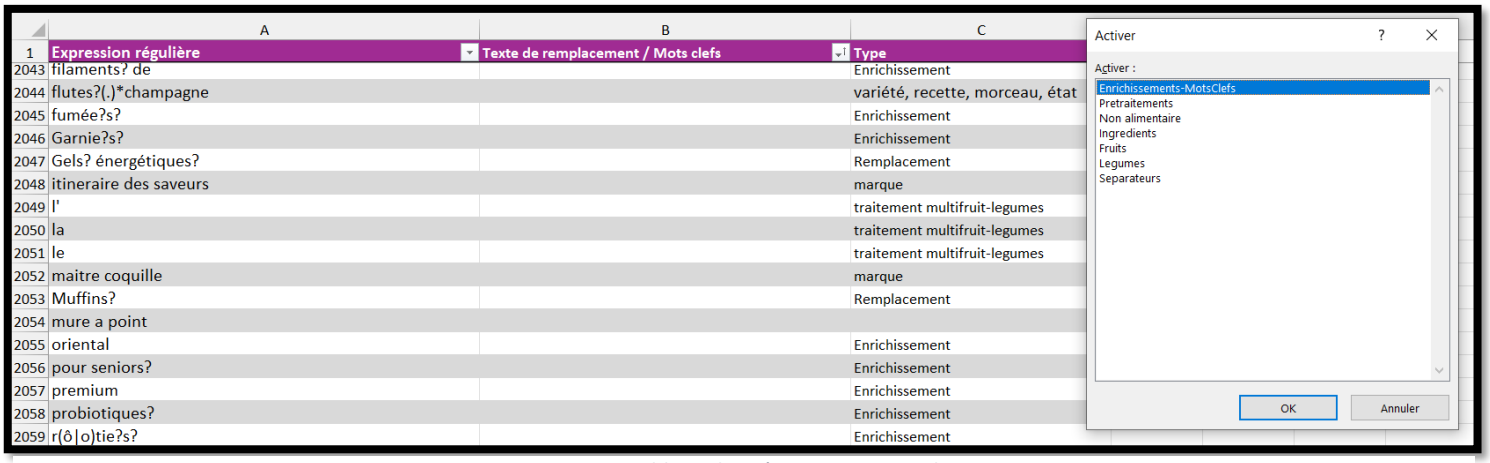


Figure 42 : Tableau de prétraitements FoodEx2

Ayant été beaucoup plus **familier au SQL[20] et à l'optimisation de requêtes** dans ce langage au cours de ma scolarité, avoir une **maîtrise d'une structure de données différente** a été très enrichissant et m'offre un point de vue plus vaste de la gestion de données de façon générale. De même que je suis habitué à utiliser Excel à des fins de lecture et d'écriture sans logique extérieure, ce projet m'a permis d'utiliser cet outil dans le but **d'optimiser des tableaux à des fins de programmation**.

Les différents modèles de données étant les plus optimisés possibles en l'état, il fallait également assurer leur sécurité, ce qui se fait à l'aide d'un simple fichier .**htaccess** permettant d'éviter que qui que ce soit n'ait accès aux référentiels ou autres données sensibles de l'application.

De même, les données de connexion d'utilisateurs seront stockées en serveur mais **le mot de passe est toujours haché**[21] avant de l'être, de sorte que même en cas de faille de sécurité, les données personnelles d'utilisateurs ne soient pas mises en danger.

e. C5 : Conduire un projet

M. Visalli, mon maître de stage, représentait la maîtrise d'ouvrage, et je demeurais **responsable de la majeure partie des fonctionnalités**. Étant déjà chef de projet depuis l'année dernière en projets semestriels de mon BUT Informatique, j'étais donc habitué à prendre en main une application sous tous ses aspects à la fois et je n'ai finalement pas eu de grands problèmes d'organisation de façon générale. Ce travail a été réalisé progressivement, une fonctionnalité à la fois, et différentes branches étaient utilisées sur Github afin d'éviter qu'une fonctionnalité non finie n'ait un impact sur la dernière version fonctionnelle de l'application.

Créer cette application impliquait également de la documenter le plus précisément possible, ce pourquoi une documentation au projet entier est disponible sur son dépôt Github, le code en lui-même ayant été documenté tout le long du projet.

Enfin, mes responsabilités durant ce stage impliquaient également de réfléchir à l'avenir de ce projet, c'est-à-dire à d'éventuelles reprises de ce dernier. Dans ce contexte, j'ai donc **rédigé un manuel de maintenance entier** qui permettrait à n'importe qui, après avoir suffisamment compris la logique derrière ces tableaux, de reprendre le projet et de l'optimiser sans avoir à toucher une ligne de code.

[20] Le SQL (Structured Query Language, « langage de requêtes structurées ») [61] est un langage d'exploitation de bases de données relationnelles, permettant donc la manipulation de données de façon optimisée

[21] En cryptographie, le hachage [60] correspond à donner une image fixe à une donnée (ici un mot de passe) sans que ça ne soit réversible (dans le cas d'un mot de passe, de sorte qu'on ne puisse pas le déduire de la fonction de hachage)

f. C6 : Collaborer au sein d'une équipe informatique

Comme évoqué en rétrospective, le **déploiement du projet a été réalisé par M. COULIBALY**. Pour rendre claire la mise en place d'un projet aussi conséquent, j'ai pu **réaliser différentes fiches de tutoriels** sous la forme de plusieurs fichiers lisibles sur Github directement.

Il aura également été **responsable de la mise en place de LLMs sur les serveurs de ChemoSens**, de sorte que des tests puissent être effectués dans de réelles conditions (mesurer les temps d'exécution tels qu'ils le seraient hébergés dans leurs serveurs par exemple), ce qui aura abouti, après de nombreux tests et modifications, à ce qu'on comprenne que ça ne serait pas une solution suffisamment efficace en l'état.

C'est finalement avec mon maître de stage que les plus grandes décisions étaient prises : savoir quels référentiels utiliser, quelle IA ou LLM serait permis, quelles limites au projet, etc. Il y aura eu de nombreuses interactions et un réel suivi du projet du début à la fin et je n'étais à aucun moment isolé durant ce stage.


README

Liens utiles

Liens en rapport au référentiel FoodEx :

- Description du référentiel (Liste et numérotation des facette pages 26-27)
- Liste des facettes et de leurs éléments

Documentation de l'application

- Architecture du projet
- Manuel de maintenance (Nécessaire à la bonne compréhension des ressources Excel)

Installation et configuration de l'application

Des (*) sont indiquées là où il est impératif de configurer l'application

- (*)Système de connexion
- (*)Packages shell et PHP
- Interactions LLM (si hébergé localement)
- Remplissage Excel (Si besoin de la partie remplissage de tableur Excel depuis Auchan)
- (*)Création espace Python + serveur SpaCy (Nécessaire pour utiliser SpaCy)
- Mise en place de modèle BERT (Non implémenté pour le moment)

Figure 43 : Page principale de tutoriel d'installation

IV.3 CONCLUSION ET RESULTAT FINAL

Le projet consistait à développer une application web qui recevrait une **facture en entrée** et renvoyait un **tableau de données structuré au format JSON et enrichi de différentes informations en sortie**. Les différentes opérations unitaires utilisés par l'application sont également disponibles sous la forme d'une **API**. (Pour plus de détails visuels sur le cheminement complet, voir annexe 1) Seul le déploiement de l'application ne relevait pas de ma responsabilité mais il sera aisément réalisable à l'aide de nombreux fichiers de documentation disponibles sur le dépôt principal du projet.

Mon objectif principal avec ce stage a été de mettre à contribution et développer les compétences développées en BUT Informatique dans un cadre professionnel, ce qui a donc été accompli et de très loin. J'ai su implémenter toutes les fonctionnalités demandées par mon maître de stage, apprendre à utiliser beaucoup d'outils que je ne connaissais pas, en particulier les LLMs et l'Intelligence Artificielles auxquels j'ai pu être concrètement introduit pour la première fois en termes de programmation.

J'ai naturellement encore une marge d'amélioration et j'ai rapidement pu prendre conscience de mes propres limites durant ce stage, ne pouvant pas nécessairement apporter de solutions idéales aux demandes de ce projet car demandant parfois l'utilisation d'outils qui prendraient bien plus de huit semaines à implémenter (les correspondances FoodEx2 par exemple qui auraient pu avoir un programme bien plus optimisé), mais c'est à force d'expérience et d'apprentissage que je finirais par obtenir les compétences qui me font défaut aujourd'hui.

Mon expérience au sein de ChemoSens a été appréciable durant ces deux mois et j'en remercie chacun des membres. Ce stage aura été une expérience enrichissante, me permettant non seulement de mettre en pratique mes connaissances scolaires mais également d'élargir mes horizons, particulièrement pour le domaine de l'Intelligence Artificielle que j'envisage désormais en guise de poursuite d'études post-BUT. Je suis convaincu que cette expérience et les compétences nouvellement acquises serviront à des opportunités futures et suis impatient de pouvoir mettre à contribution mon savoir-faire à de nouveaux projets.

V LEXIQUE

Les sources des définitions se trouvent en Bibliographie.

- **Ministère de l'enseignement supérieur français** [32] : Ministère en charge à la fois des parcours supérieurs d'étudiants et de l'attribution de budgets à différents organismes de recherche.
- **Le ministère de l'Agriculture français** [33] : Ministère responsable, non seulement du secteur agricole (alimentaire, forestier, …), mais aussi partiellement de la recherche concernant ces domaines.
- **API** [42] : Application Programming Interface, « interface de programmation d'application », il s'agit d'une solution tierce qui donne un accès externe à une application. Dans notre cas, on cherche une API que du code PHP pourrait appeler pour utiliser l'application en question par exemple.
- **Shell** [52] : interpréteur de commandes de type Unix, ayant une interface beaucoup plus proche du système en lui-même.
- **JavaScript** [53] : langage de programmation utilisé pour des interactions directes avec une page web, notamment ici pour modifier des images à l'écran.
- **LLM** [55] : Large Language Model, « Grand modèle de langage », il s'agit d'un modèle possédant un grand nombre de paramètres, utilisés sous la forme d'un réseau neuronal n'ayant pas ou peu besoin de supervision réelle. Ils sont notamment utilisés pour déduire des suites logiques de données, dans notre cas : pour déduire une structure de données d'une facture envoyée en paramètre.
- **BERT** [44] : Langage de vectorisation d'un texte permettant de « comprendre » le contexte d'une phrase ou de mots dans une requête fournie par exemple.
- **Python** [45] : langage de programmation connu pour sa syntaxe simple et ses outils de haut niveau à la fois. Il est notamment utilisé dans le domaine de l'IA et du Machine Learning
- **Fine tuning** [46] : Réglage fin en français, méthode d'apprentissage permettant à un LLM d'étendre ses connaissances sur des jeux de données spécifiquement liés à la question d'intérêt. Dans notre cas, nous fournissons à un modèle BERT les réponses attendues à différentes factures entrées
- **Expression régulière** [48] : En informatique, une expression régulière désigne une chaîne de caractères, un ou plusieurs mots, que l'on souhaiterait utiliser à des fins de comparaisons, d'analyse textuelles ou encore de modifications d'un texte. Différents opérateurs sont utilisés afin de permettre des comparaisons plus poussées entre différentes chaînes données [49]
- **Test unitaire** [50] : En programmation, correspond à une vérification du bon fonctionnement d'un programme à partir d'une valeur attendue selon une donnée à ce même programme. Dans notre cas, le test consiste en ajouter une désignation et vérifier qu'on obtient le bon code en retour.
- **Mise en mémoire cache** [51] : consiste en un enregistrement temporaire de données limitant le temps d'accès futurs à ces mêmes données.
- **Web scraping** [62] : méthode de récupération, d'extraction de données d'un site web à l'aide d'un script.
- **EDI** [54] : Environnement de Développement Intégré, un outil optimisé pour la programmation : intégrant compilateur, débogueur, éditeur de texte plus détaillé, etc.
- **Framework** [56] **:** En informatique, il s'agit d'un ensemble de composants logiciels servant à obtenir les fondations ou grandes lignes d'un logiciel.

- **Canvas** [57] : En HTML, permet le rendu d'images à l'aide de scripts (dont JavaScript par exemple) de façon dynamique à l'aide de bitmaps (où chaque pixel est modifiable, offrant donc des interactions comme utilisées par Konva)
- **Bootstrap** [58] : collection d'outils web facilitant grandement la réalisation de l'affichage, en le gardant responsive (adapté, peu importe la taille de l'écran) et accessible.
- **Htaccess** [59] : fichier de configuration définissant les règles de sécurité d'un serveur web, pouvant être propre à un dossier, limitant l'accès à différents scripts voire l'interdisant par exemple.
- **SQL** [61] : Structured Query Language, « langage de requêtes structurées », langage d'exploitation de bases de données relationnelles, permettant donc la manipulation de données de façon optimisée
- **Hachage** [60] **:** En cryptographie, correspond à donner une image fixe à une donnée (ici un mot de passe) sans que ça ne soit réversible (dans le cas d'un mot de passe, de sorte qu'on ne puisse pas le déduire de la fonction de hachage)

VI BIBLIOGRAPHIE

[1] INRAE, «Organisation INRAE,» [En ligne]. Available: https://www.inrae.fr/nous-connaitre/organigramme.

[2] INRAE, «Bourgogne-Franche-Comte,» [En ligne]. Available: https://www.inrae.fr/centres/bourgogne-franche-comte.

[3] CSGA, «Présentation du Centre,» [En ligne]. Available: https://csga.fr/presentation-du-centre.

[4] ChemoSens, «TimeSens,» [En ligne]. Available: https://www.chemosenstools.com/timesens/.

[5] ChemoSens, «Github de ChemoSens (Packages R),» [En ligne]. Available: https://github.com/ChemoSens.

[6] Ubunlog, «Pdftoppm, convertissez des fichiers PDF en images à partir d'Ubuntu,» [En ligne]. Available: ubunlog.

[7] Konva, «Site web de Konva,» [En ligne]. Available: https://konvajs.org/index.html.

[8] W. Ubuntu, «Tesseract OCR,» [En ligne]. Available: https://doc.ubuntu-fr.org/tesseract-ocr.

[9] Ollama, «Ollama.com,» [En ligne]. Available: https://ollama.com/.

[10] EFSA, «Normalisation des données (FoodEx2),» [En ligne]. Available: https://www.efsa.europa.eu/fr/data/data-standardisation.

[11] SpaCy, «Page d'accueil SpaCy,» [En ligne]. Available: https://spacy.io/.

[12] Wikipedia, «SpaCy,» [En ligne]. Available: https://en.wikipedia.org/wiki/SpaCy.

[13] EFSA, «The food classification and description system FoodEx2 (p26-27),» [En ligne]. Available: https://data.food.gov.uk/codes/foodtype/hierarchy/facet.

[14] O. F. Facts, «Page d'accueil,» [En ligne]. Available: https://fr.openfoodfacts.org/.

[15] Microsoft, «Visual Studio Code,» [En ligne]. Available: https://code.visualstudio.com/.

[16] Microsoft, «Microsoft Excel,» [En ligne]. Available: https://www.microsoft.com/fr-fr/microsoft-365/excel.

[17] Microsoft, «Qu'est-ce que PowerShell ?,» [En ligne]. Available: https://learn.microsoft.com/fr-fr/powershell/scripting/overview?view=powershell-7.5.

[18] Microsoft, «Commandes de base pour WSL,» [En ligne]. Available: https://learn.microsoft.com/fr-fr/windows/wsl/basic-commands.

[19] Github, «Page d'accueil,» [En ligne]. Available: https://github.com/.

[20] Wikipedia, «Github,» [En ligne]. Available: https://fr.wikipedia.org/wiki/GitHub.

[21] PHPOffice, «PHPSpreadSheet,» [En ligne]. Available: https://github.com/PHPOffice/PhpSpreadsheet.

[22] PHPOffice, «Dépôt principal,» [En ligne]. Available: https://github.com/PHPOffice.

[23] S. H. DOM, «Documentation Simple HTML DOM,» [En ligne]. Available: https://simplehtmldom.sourceforge.io/docs/1.9/index.html.

[24] Wikipedia, «JSON Web Token,» [En ligne]. Available: https://fr.wikipedia.org/wiki/JSON_Web_Token.

[25] G. (Firebase), «PHP-JWT,» [En ligne]. Available: https://github.com/firebase/php-jwt.

[26] Konva, «Documentation,» [En ligne]. Available: https://konvajs.org/docs/overview.html.

[27] Wikipedia, «Node.js,» [En ligne]. Available: https://fr.wikipedia.org/wiki/Node.js.

[28] Puppeteer, «QU'est-ce que Puppeteer ?,» [En ligne]. Available: https://pptr.dev/guides/what-is-puppeteer.

[29] Vert, «Électricité, eau, minéraux, CO2 : on a tenté de mesurer l'empreinte écologique de ChatGPT,» [En ligne]. Available: https://vert.eco/articles/electricite-eau-mineraux-co2-on-a-tente-de-mesurer-lempreinte-ecologique-de-chatgpt.

[30] OpenAI, «GPT-4 Turbo in the OpenAI API,» [En ligne]. Available: https://help.openai.com/en/articles/8555510-gpt-4-turbo-in-the-openai-api.

[31] Wikipedia, «Ministère de l'Enseignement supérieur (France),» [En ligne]. Available: https://fr.wikipedia.org/wiki/Minist%C3%A8re_de_l%27Enseignement_sup%C3%A9rieur_(France).

[32] Wikipedia, «Ministère de l'Agriculture (France),» [En ligne]. Available: https://fr.wikipedia.org/wiki/Minist%C3%A8re_de_l%27Agriculture_(France).

[33] Wikipedia, «Grand modèle de langage,» [En ligne]. Available: https://fr.wikipedia.org/wiki/Grand_mod%C3%A8le_de_langage.

[34] Wikipedia, «BERT (language model),» [En ligne]. Available: https://en.wikipedia.org/wiki/BERT_(language_model).

[35] Wikipedia, «Python (langage),» [En ligne]. Available: https://fr.wikipedia.org/wiki/Python_(langage).

[36] Wikipedia, «Réglage fin (Fine tuning),» [En ligne]. Available: https://fr.wikipedia.org/wiki/R%C3%A9glage_fin.

[37] Wikipedia, «Expression régulière,» [En ligne]. Available: https://fr.wikipedia.org/wiki/Expression_r%C3%A9guli%C3%A8re.

[38] Wikipedia, «Expression régulière (Opérateurs),» [En ligne]. Available: https://fr.wikipedia.org/wiki/Expression_r%C3%A9guli%C3%A8re#Op%C3%A9rateurs.

[39] Wikipedia, «Test unitaire,» [En ligne]. Available: https://fr.wikipedia.org/wiki/Test_unitaire.

[40] Wikipedia, «Mémoire Cache,» [En ligne]. Available: https://fr.wikipedia.org/wiki/M%C3%A9moire_cache.

[41] Wikipedia, «Web Scraping,» [En ligne]. Available: https://fr.wikipedia.org/wiki/Web_scraping.

[42] Wikipedia, «Environnement de développement,» [En ligne]. Available: https://fr.wikipedia.org/wiki/Environnement_de_d%C3%A9veloppement.

[43] Wikipedia, «Framework,» [En ligne]. Available: https://fr.wikipedia.org/wiki/Framework.

[44] Wikipedia, «Canvas (HTML),» [En ligne]. Available: https://fr.wikipedia.org/wiki/Canvas_(HTML).

[45] Wikipedia, «Bootstrap (framework),» [En ligne]. Available: https://fr.wikipedia.org/wiki/Bootstrap_(framework).

[46] Apache, «.htaccess,» [En ligne]. Available: https://httpd.apache.org/docs/2.4/fr/howto/htaccess.html.

[47] Wikipedia, «Structured Query Language,» [En ligne]. Available: https://fr.wikipedia.org/wiki/Structured_Query_Language.

[48] Wikipedia, «Fonction de hachage cryptographique,» [En ligne]. Available: https://fr.wikipedia.org/wiki/Fonction_de_hachage_cryptographique.

[49] OpenAI, «Pricing,» [En ligne]. Available: https://platform.openai.com/docs/pricing.

[50] Wikipedia, «Interface de programmation,» [En ligne]. Available: https://fr.wikipedia.org/wiki/Interface_de_programmation.

[51] Wikipedia, «Shell Unix,» [En ligne]. Available: https://fr.wikipedia.org/wiki/Shell_Unix.

[52] Wikipedia, «JavaScript,» [En ligne]. Available: https://fr.wikipedia.org/wiki/JavaScript.

VII ANNEXES

VII.1 CHEMINEMENT COMPLET DE L'APPLICATION

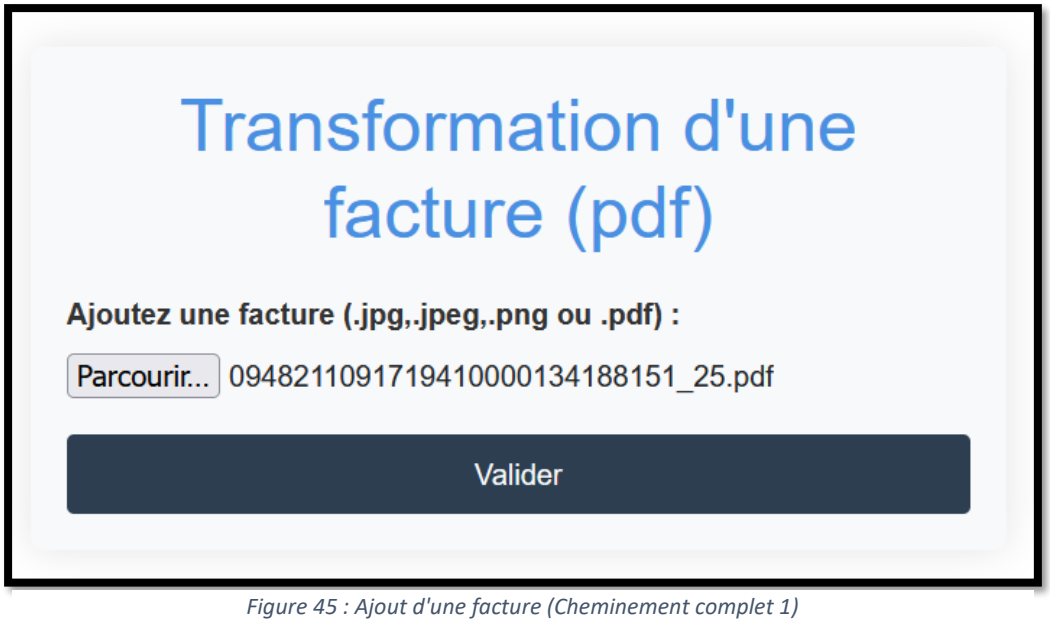


Figure 45 : Ajout d'une facture (Cheminement complet 1)

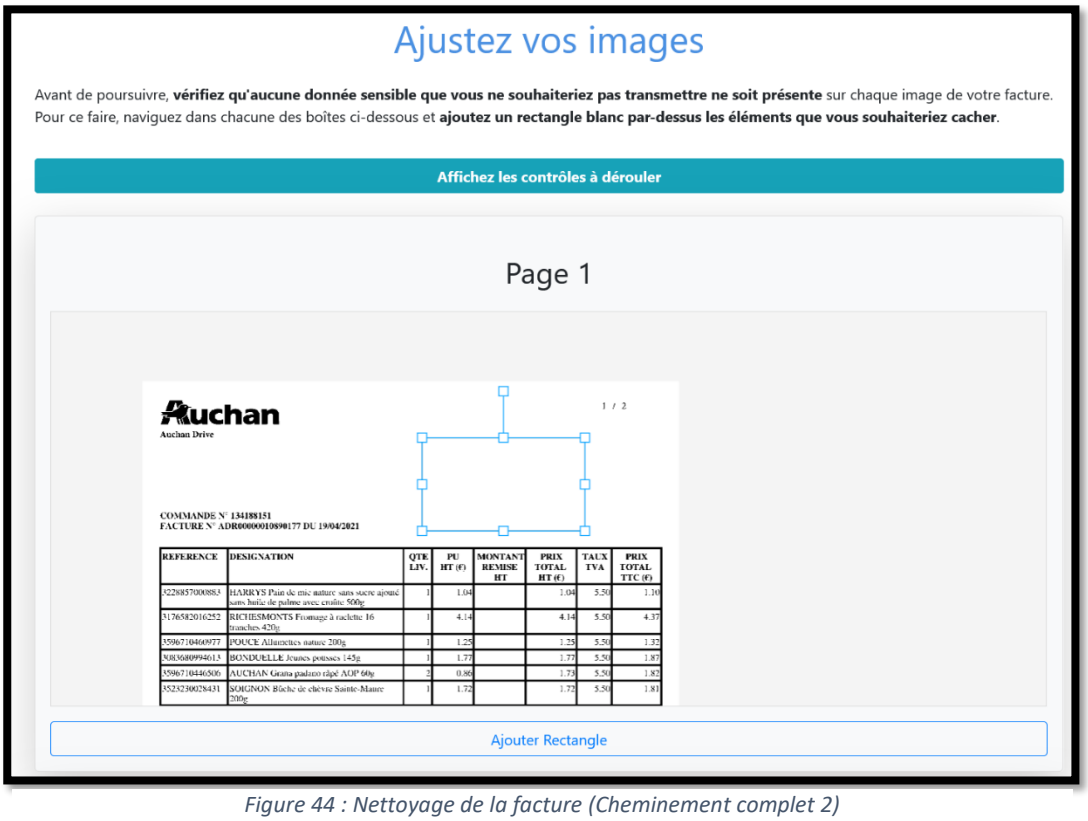
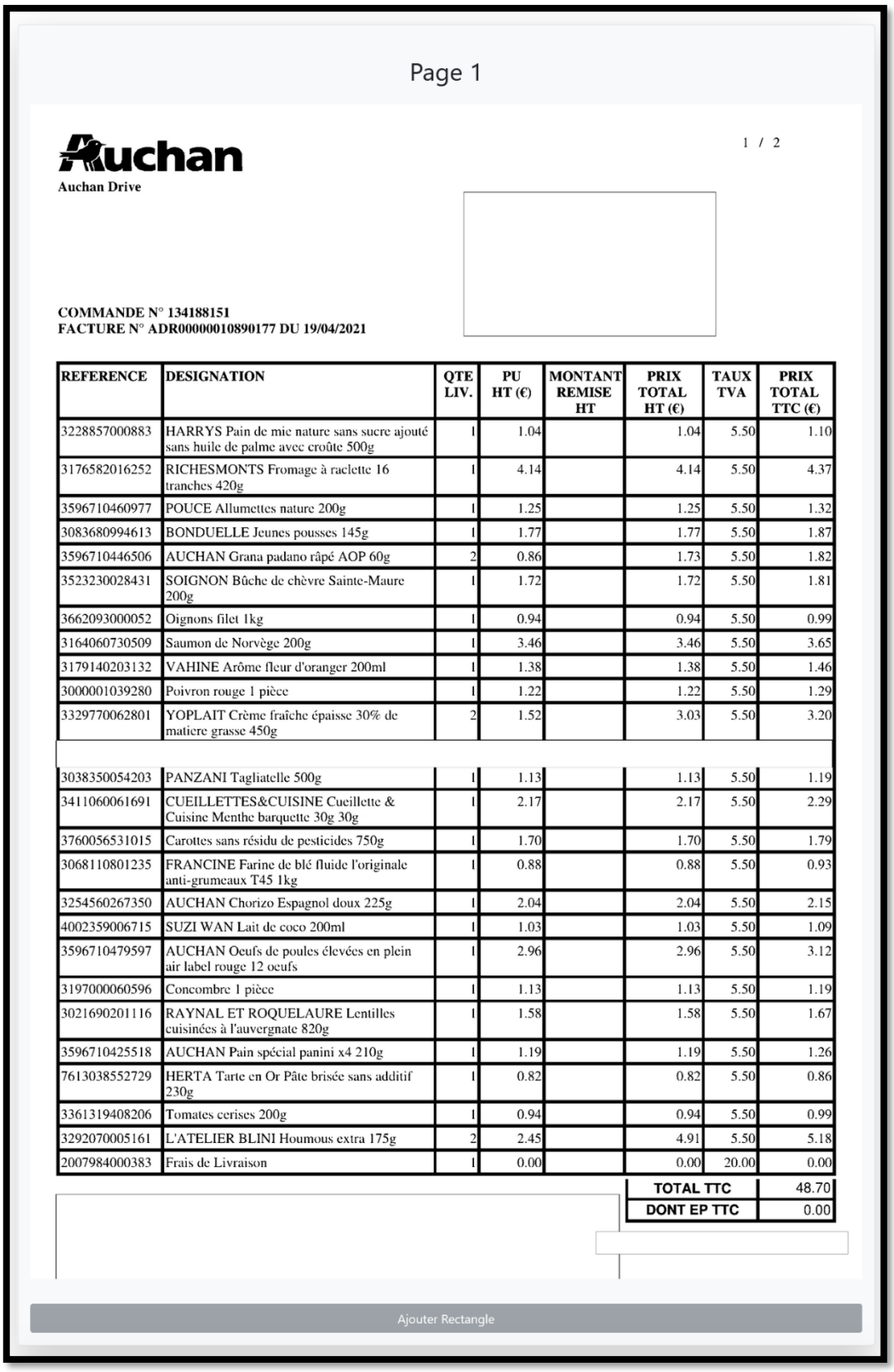


Figure 44 : Nettoyage de la facture (Cheminement complet 2)



REFERENCE	DESIGNATION	QTE LIV.	PU HT (€)	MONTANT REMISE HT	PRIX TOTAL HT (€)	TAUX TVA	PRIX TOTAL TTC (€)
3228857000883	HARRYS Pain de mie nature sans sucre ajouté sans huile de palme avec croûte 500g	1	1.04		1.04	5.50	1.10
3176582016252	RICHESMONTS Fromage à raclette 16 tranches 420g	1	4.14		4.14	5.50	4.37
3596710460977	POUCE Allumettes nature 200g	1	1.25		1.25	5.50	1.32
3083680994613	BONDUELLE Jeunes pousses 145g	1	1.77		1.77	5.50	1.87
3596710446506	AUCHAN Grana padano râpé AOP 60g	2	0.86		1.73	5.50	1.82
3523230028431	SOIGNON Bûche de chèvre Sainte-Maure 200g	1	1.72		1.72	5.50	1.81
3662093000052	Oignons filet 1kg	1	0.94		0.94	5.50	0.99
3164060730509	Saumon de Norvège 200g	1	3.46		3.46	5.50	3.65
3179140203132	VAHINE Arôme fleur d'oranger 200ml	1	1.38		1.38	5.50	1.46
3000001039280	Poivron rouge 1 pièce	1	1.22		1.22	5.50	1.29
3329770062801	YOPLAIT Crème fraîche épaisse 30% de matiere grasse 450g	2	1.52		3.03	5.50	3.20
3038350054203	PANZANI Tagliatelle 500g	1	1.13		1.13	5.50	1.19
3411060061691	CUEILLETTES&CUISINE Cueillette & Cuisine Menthe barquette 30g 30g	1	2.17		2.17	5.50	2.29
3760056531015	Carottes sans résidu de pesticides 750g	1	1.70		1.70	5.50	1.79
3068110801235	FRANCINE Farine de blé fluide l'originale anti-grumeaux T45 1kg	1	0.88		0.88	5.50	0.93
3254560267350	AUCHAN Chorizo Espagnol doux 225g	1	2.04		2.04	5.50	2.15
4002359006715	SUZI WAN Lait de coco 200ml	1	1.03		1.03	5.50	1.09
3596710479597	AUCHAN Oeufs de poules élevées en plein air label rouge 12 oeufs	1	2.96		2.96	5.50	3.12
3197000060596	Concombre 1 pièce	1	1.13		1.13	5.50	1.19
3021690201116	RAYNAL ET ROQUELAURE Lentilles cuisinées à l'auvergnate 820g	1	1.58		1.58	5.50	1.67
3596710425518	AUCHAN Pain spécial panini x4 210g	1	1.19		1.19	5.50	1.26
7613038552729	HERTA Tarte en Or Pâte brisée sans additif 230g	1	0.82		0.82	5.50	0.86
3361319408206	Tomates cerises 200g	1	0.94		0.94	5.50	0.99
3292070005161	L'ATELIER BLINI Houmous extra 175g	2	2.45		4.91	5.50	5.18
2007984000383	Frais de Livraison	1	0.00		0.00	20.00	0.00

TOTAL TTC	48.70
DONT EP TTC	0.00

Figure 46 : Prévisualisation du résultat (Cheminement complet 3)

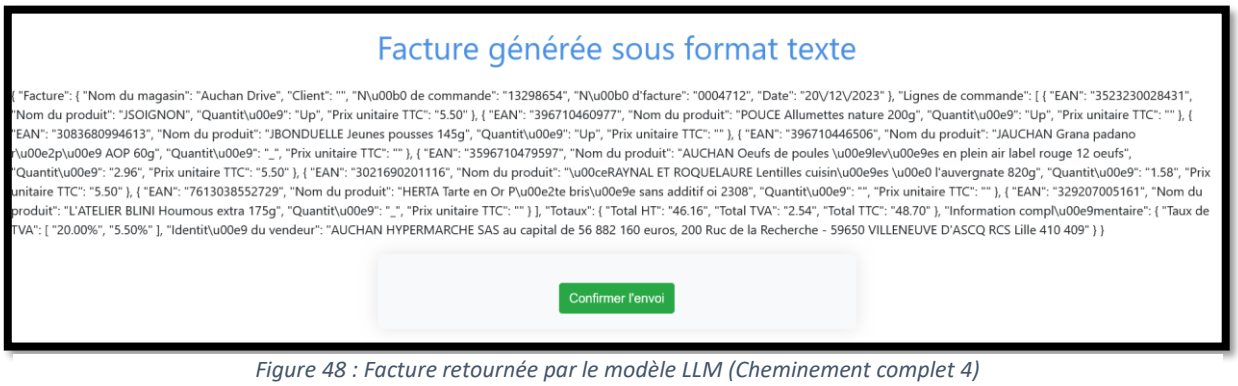


Figure 48 : Facture retournée par le modèle LLM (Cheminement complet 4)


```
  "OpenFoodFacts":"Pas de correspondance dans OpenFoodFacts trouv\u00e9e"
,
 "EAN":"396710446506",
 "name":"Grana padano r\u00e2p\u00e9",
 "Quantit\u00e9":"_",
 "Prix unitaire TTC":"",
 "Type":"Alimentaire",
 "FoodEx":{
   "Reference":"A.01.001053",
   "Designation":"Fromage",
   "DesignationEnrichie":"Fromage",
   "ReferenceFoodEx":"A.01.001053#F16.A07KZ$F23.A07TD$F23.A07TE",
   "DesignationAnglaise":"Cheese",
   "Facettes":[
     {
       "Code":"F16.A07KZ",
       "Cat\u00e9gorie":"F16 - Traitement structurel",
       "Designation":"R\u00e2page",
       "Description":"R\u00e9duire en fragments, en lambeaux ou en poudre par frottement contre une surface abrasive"
     },
     {
       "Code":"F23.A07TD",
       "Cat\u00e9gorie":"F23 - Consommateur Cible",
       "Designation":"Alimentation humaine",
       "Description":"Aliments pour humains"
     },
     {
       "Code":"F23.A07TE",
       "Cat\u00e9gorie":"F23 - Consommateur Cible",
       "Designation":"Alimentation pour adultes",
       "Description":"Denr\u00e9es alimentaires principalement destin\u00e9es aux adultes ou \u00e0 la population g\u00e9n\u00e9rale"
     }
   ],
   "Ingredients":[

   ]
 },
 "OpenFoodFacts":"Pas de correspondance dans OpenFoodFacts trouv\u00e9e"
```


Figure 47 : Fichier JSON retourné (Cheminement complet 5)

VII.2 VISUELS DU SITE WEB

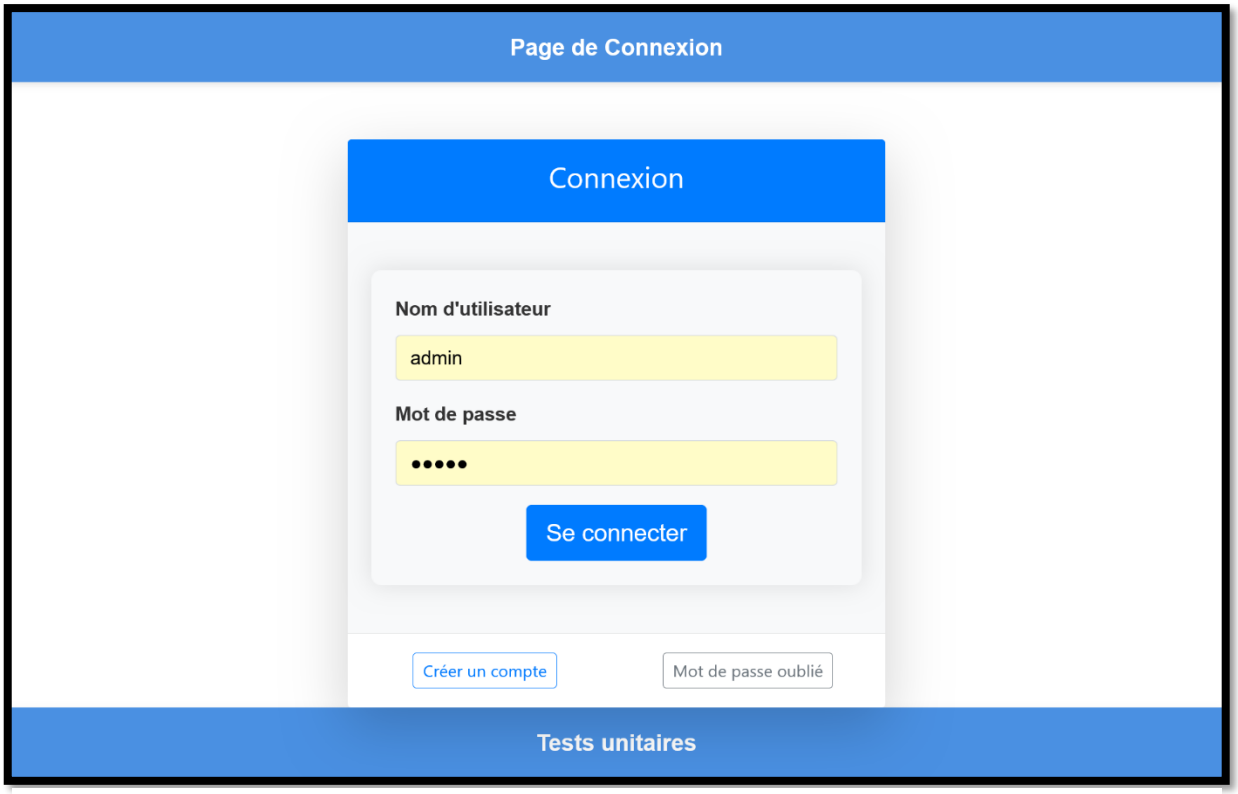


Figure 50 : Page de connexion

Figure 49 : Gestionnaire d'utilisateurs

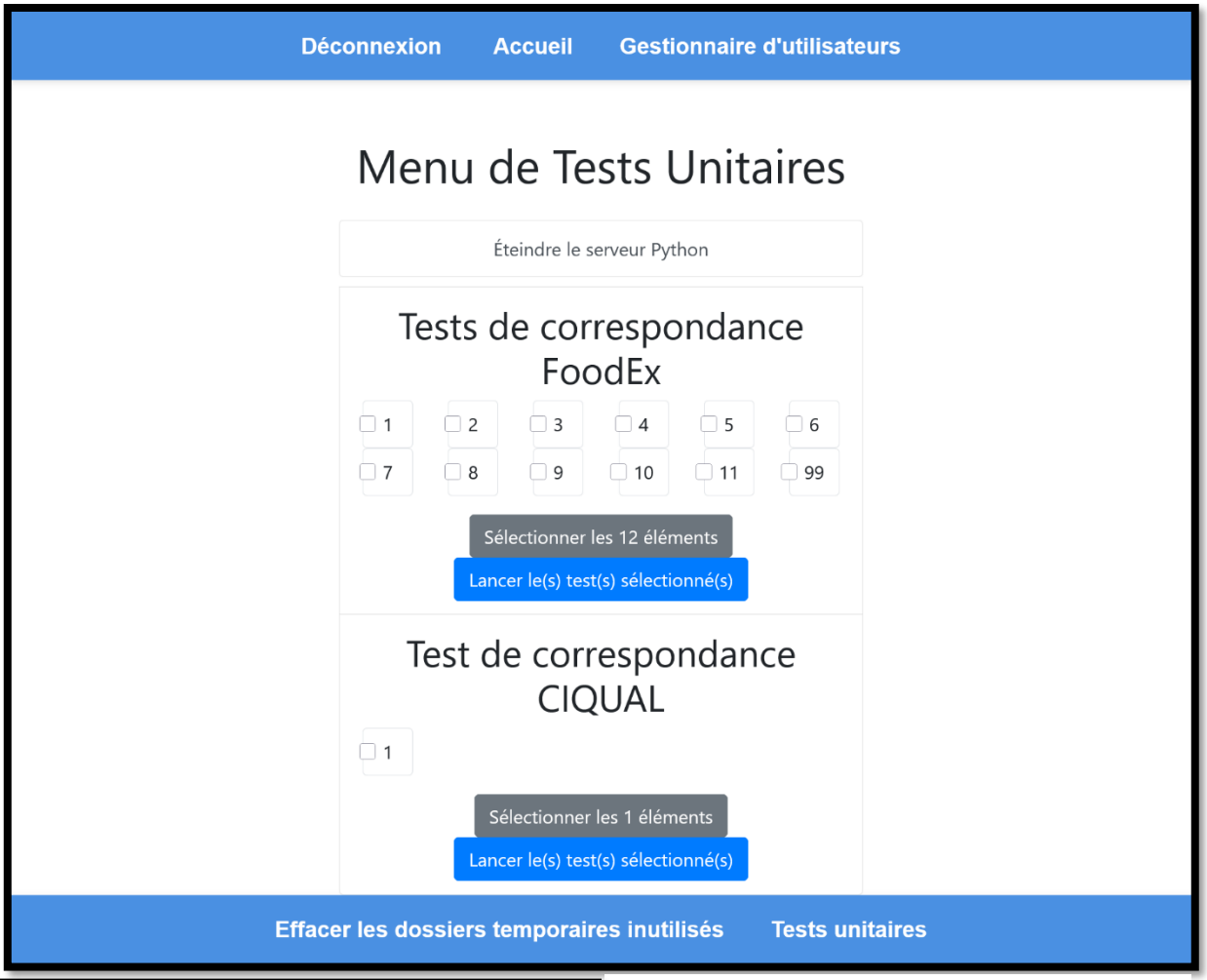


Figure 51 : Menu de tests unitaires

Figure 52 : Page d'accueil

Résultat de la recherche

Informations sur le produit n°1
Source : Auchan.fr
Nom du produit
Cranberries séchées entières d'Amérique du Nord
Marque
BROUSSE & FILS
Dénomination légale de vente
Cranberries entières
Référence
444252/3191220021078
Pictogrammes
Ingrédients
Cranberries entières
Note
N
ValEner_kJ
1510 kJ
ValEner_kcal
356 kcal
MatGrasses
0,5 g
dontAcidesSatures
< 0,1 g
glucides
84,9 g
dontSucres
68,8 g
proteines
0,5 g
sel
0,004 g
fibres

Figure 53 : Résultat d'une recherche Auchan

VIII RESUME / ABSTRACT

Ce document présente le contexte et le déroulement de mon stage de 2ème année de BUT Informatique, réalisé au sein de la plateforme de recherche ChemoSens. Ma mission était de réaliser une application web d'analyse de factures alimentaires. Dans le détail, cette application devait permettre : (i) d'extraire le texte d'une facture alimentaire (dans un format image ou PDF), (ii) d'identifier les informations pertinentes dans ce texte et les structurer sous forme de fichier JSON, (iii) de catégoriser les désignations commerciales selon le référentiel FoodEx2, et (iv) d'aller chercher des données supplémentaires sur les caractéristiques des aliments dans la base de données Open Food Facts. Utilisant une architecture sous PHP, Microsoft Excel pour gérer les données FoodEx2 ou encore Apache pour l'hébergement local de serveurs, ce projet m'a permis d'acquérir de nombreuses compétences dans le domaine de la programmation, de la rédaction de livrables (livret de maintenance, d'installation, etc.) ainsi que la gestion d'un projet informatique.

This document describes the context and progress of my second-year internship for my IT BUT carried out within the ChemoSens research platform. My mission was to create a web application for analyzing food invoices. In detail, this application had to: (i) extract the text from a food invoice (in image or PDF format), (ii) identify the relevant information in this text and structure it in the form of a JSON file, (iii) categorize commercial designations according to the FoodEx2 reference system, and (iv) fetch additional data on food characteristics from the Open Food Facts database. Using an architecture based on PHP, Microsoft Excel to manage FoodEx2 data and Apache for local server hosting, this project enabled me to acquire numerous skills in programming, writing deliverables (maintenance and installation booklets, etc.) and managing an IT project.

Mots-clefs: Factures, analyse factures, FoodEx2, OpenFoodFacts, JSON, PHP, IA, BERT, Ollama, LLM, INRAE, CSGA, ChemoSens