Thierry Koetschet

Chemin du perrey 22 | 1670 Ursy

thierry.koetschet.1998@gmail.com

SI-ca2a

Application web de fitness

rapport de tpi

C:\Users\thierry.koetschet\Documents\GitHub\TPI_Thierry_Koetschet\logo_infofit.png

Table des matières

[1 Analyse préliminaire 4](#_Toc136357507)

[1.1 Introduction 4](#_Toc136357508)

[1.2 Organisation 5](#_Toc136357509)

[1.3 Méthode de gestion de projet 6](#_Toc136357510)

[1.4 Objectifs 6](#_Toc136357511)

[1.5 Planification initiale 7](#_Toc136357512)

[1.6 Structure du dossier 8](#_Toc136357513)

[2 Analyse / Conception 9](#_Toc136357514)

[2.1 Concept 9](#_Toc136357515)

[2.1.1 Modèle conceptuel de données 9](#_Toc136357516)

[2.1.2 Modèle logique de données 10](#_Toc136357517)

[2.1.3 Base de données 11](#_Toc136357518)

[2.1.4 Moodboard 12](#_Toc136357519)

[2.1.5 Maquettes 13](#_Toc136357520)

[2.2 Stratégie de test 17](#_Toc136357521)

[2.3 Risques techniques 17](#_Toc136357522)

[2.4 Planification 18](#_Toc136357523)

[2.5 Dossier de conception 18](#_Toc136357524)

[3 Réalisation 19](#_Toc136357525)

[3.1 Dossier de réalisation 19](#_Toc136357526)

[3.2 Code source 20](#_Toc136357527)

[3.2.1 Fonction « store » de UserController.php 21](#_Toc136357528)

[3.2.2 Fonction « searchFoodstuff » de FoodController.php 22](#_Toc136357529)

[3.2.3 Fonction « lineChart » de ChartController.php 24](#_Toc136357530)

[3.3 Aperçu de l’application 25](#_Toc136357531)

[3.3.1 Page d’accueil 25](#_Toc136357532)

[3.3.2 Login 25](#_Toc136357533)

[3.3.3 Register 26](#_Toc136357534)

[3.3.4 Profil 27](#_Toc136357535)

[3.3.5 IMC 27](#_Toc136357536)

[3.3.6 Alimentation 28](#_Toc136357537)

[3.3.7 Ajouter un aliment 28](#_Toc136357538)

[3.4 Hébergement sur le serveur web 29](#_Toc136357539)

[3.5 Description des tests effectués 29](#_Toc136357540)

[3.6 Erreurs restantes 32](#_Toc136357541)

[3.7 Liste des documents fournis 32](#_Toc136357542)

[4 Conclusions 33](#_Toc136357543)

[5 Annexes 34](#_Toc136357544)

[5.1 Cahier des charges 34](#_Toc136357545)

[5.2 Horaire de travail 37](#_Toc136357546)

[5.3 Résumé du travail 38](#_Toc136357547)

[5.4 Sources – Bibliographie 39](#_Toc136357548)

[5.5 Journal de travail 40](#_Toc136357549)

[5.6 Glossaire 44](#_Toc136357550)

[5.7 Table des illustrations 45](#_Toc136357551)

[5.8 Manuel d'installation 46](#_Toc136357552)

# Analyse préliminaire

## Introduction

Ce rapport va décrire en détail la réalisation de mon projet de TPI sur une application web de fitness. Cette application permettra aux utilisateurs de s’authentifier grâce un compte et d’accéder à ses fonctionnalités. La première fonctionnalité est le calcul de l’indice de masse corporelle de l’utilisateur grâce à son poids, sa taille et son genre. La deuxième fonctionnalité est un calendrier permettant d’enregistrer les aliments consommés quotidiennement par l’utilisateur afin de calculer le total des calories et des macronutriments journaliers. Toutes ces informations sur les différents aliments absorbés seront accessibles grâce à une API publique. L’application doit être développée avec l’aide d’un framework PHP et avoir une structure de type MVC.

La raison de mon choix de partir sur un tel projet s’explique par le fait que le développement est mon domaine de prédilection en informatique, et spécifiquement le développement web, dans lequel j’aimerais idéalement poursuivre ma carrière professionnelle. De plus, lors des débuts de ma formation d’informaticien à Sainte-Croix, j’ai également développé un attrait particulier pour le sport et la musculation. C’est ainsi que m’est venu l’idée de combiner ces deux domaines pour réaliser un projet intéressant et qui pourrait également m’être utile dans ma quête du corps de mes rêves.

## Organisation

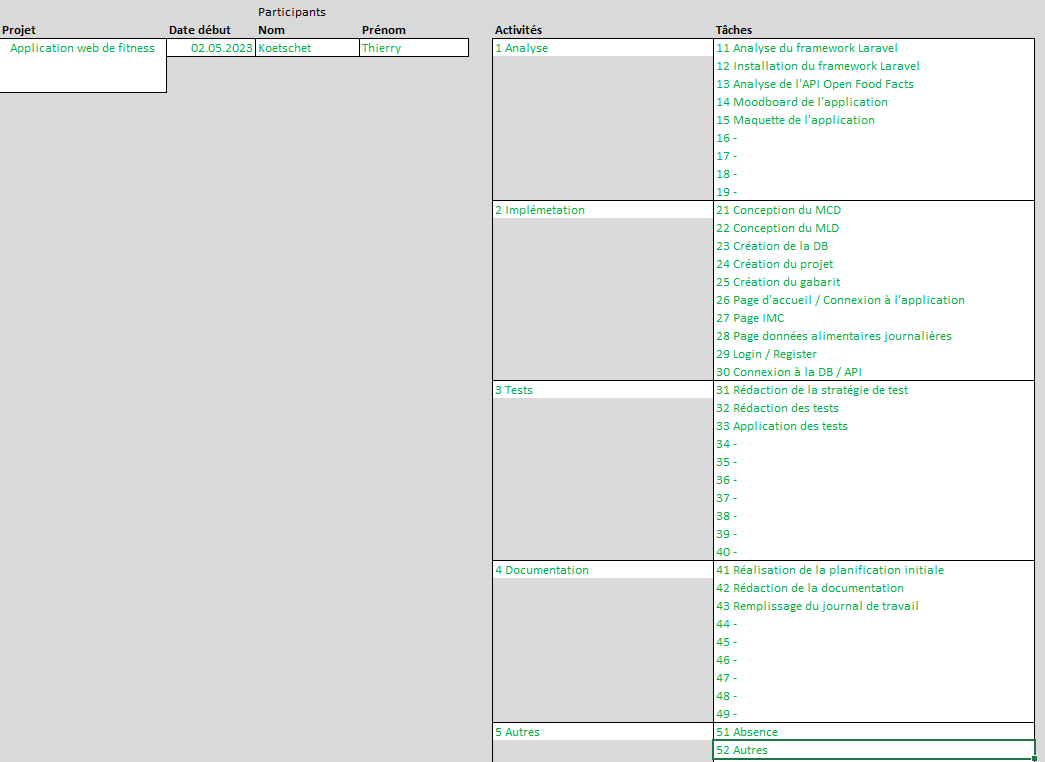


Image 1 Organisation des tâches

Dans mon organisation personnelle pour la réalisation de mon projet de TPI, j’ai décidé de séparer toutes mes tâches en quatre sprints. Le premier sprint regroupe toutes les tâches concernant la partie analytique de mon travail. Les tâches contenues dans le second sprint concernent la partie pratique du travail d’implémentation de l’application. Le troisième sprint représente la partie tests du travail et enfin le dernier sprint est un peu plus particulier car il contient le travail de rédaction de la documentation et de remplissage du journal de travail qui doit être fait du début à la fin du projet.

## Méthode de gestion de projet

J’ai choisi la méthode en cascade pour la gestion de mon projet de TPI. Je trouve que pour un projet en solo, cette méthode répond parfaitement à mes besoins en terme de gestion de projet.

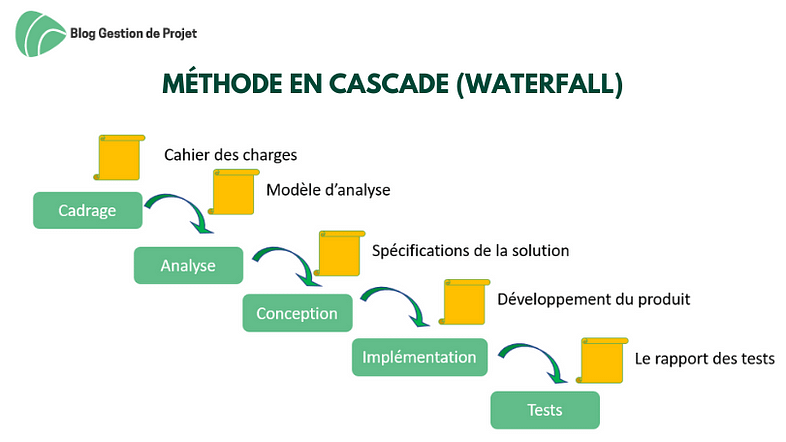


Image 2 Méthode de gestion de projet en cascade

Les cinq étapes de la méthode en cascade correspondent parfaitement à la vision que j’ai pour la réalisation d’un tel projet. En effet, le cadrage en début de projet est primordial afin de structurer notre travail et de ne pas trop s’éparpiller. La planification initiale représente parfaitement cette première étape car elle permet de poser une valeur de temps sur les différentes à réaliser dans la suite du projet. La partie analyse correspond au travail préliminaire à réaliser avant de se lancer dans la conception comme l’analyse du framework à utiliser ou encore l’API recommandée dans le cahier des charges. Après la partie analyse, j’ai réalisé la conception de mon application grâce à un moodboard, une maquette, un modèle conceptuel de données et un modèle logique de données. L’étape suivante est d’implémenter le site web, c’est-à-dire d’écrire tout le code nécessaire à son bon fonctionnement. Enfin, les tests permettent de vérifier le fonctionnement général de l’application.

## Objectifs

Les objectifs fixés dans le cahier des charges par le chef de projet et les experts sont de réaliser un site web avec l’utilisation d’un framework PHP. L’application web doit également avoir plusieurs fonctionnalités telles qu’un authentificateur sécurisé, un calculateur d’indice de masse corporelle montrant l’évolution du poids des utilisateurs grâce à un graphique et un calendrier permettant de contrôler son alimentation utilisant une API publique.

Personnellement l’objectif principal que j’aimerais atteindre à la fin de ce travail est la maîtrise d’un framework PHP car je pense que cela pourrait peser dans la balance dans la recherche d’un futur emploi dans le développement web.

## Planification initiale

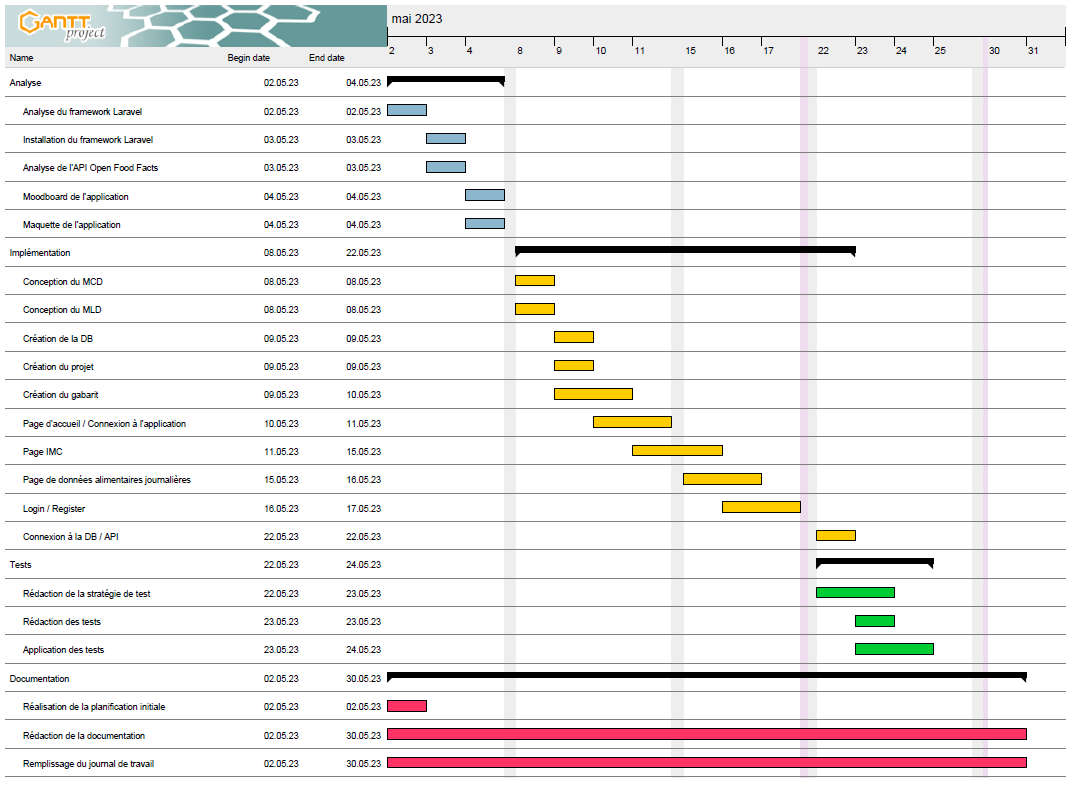


Image 3 Planification initiale du projet de TPI

## Structure du dossier

Ce dossier se divise en quatre parties principales.

La première partie se compose de l’analyse préliminaire de ce travail, c’est-à-dire d’une introduction, d’une explication de l’organisation du projet, une description des objectifs visés et puis un bref aperçu de ma planification initiale.

La deuxième englobe toute la partie analyse et conception du projet avec l’élaboration du concept, de la stratégie de test, un compte rendu des risques techniques, une révision de la planification initiale du projet et enfin la partie conception du projet.

La troisième partie représente toute la réalisation pratique du projet commençant par lister tous les fichiers du dossier de réalisation, puis une description des tests effectués et des erreurs restantes et finalement une énumération des documents fournis à la remise du projet.

La dernière décrit les conclusions auxquelles je suis arrivé à la fin de ce projet telles que les objectifs atteints ou non, mon ressenti au fil du projet, les difficultés rencontrées et les améliorations que je pourrais apporter si je devais refaire un tel projet.

# Analyse / Conception

## Concept

### Modèle conceptuel de données

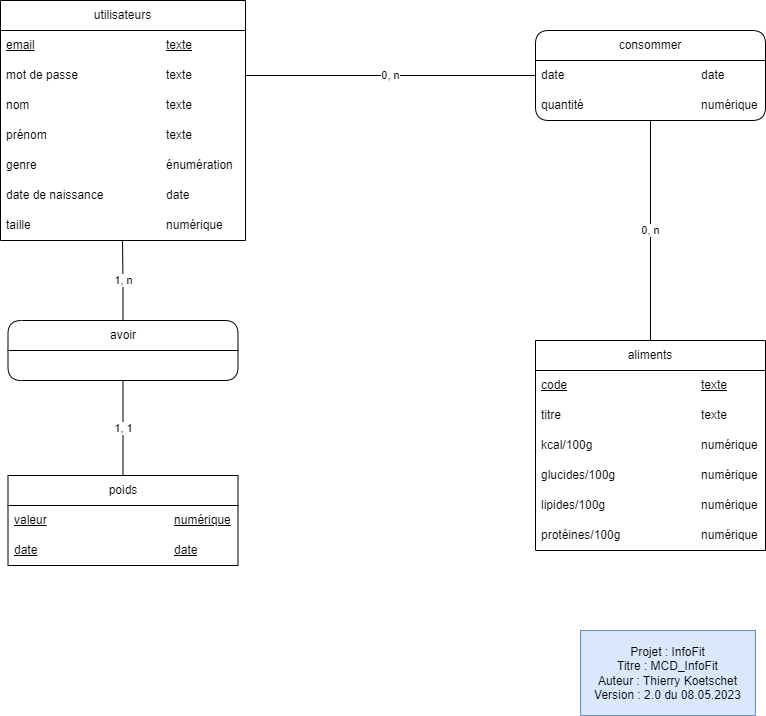


Image 4 MCD de la base de données infofit

Le MCD a été créé avec draw.io. Il s’agit d’un logiciel libre d’édition graphique de diagrammes.

Au lieu d’avoir un simple champ poids dans l’entité « utilisateurs », j’ai ajouté une entité supplémentaire « poids » avec comme attributs « valeur » et « date » afin de permettre aux utilisateurs de mon application de modifier leur poids au fur et à mesure de leur évolution et d’en avoir un historique.

### Modèle logique de données



Image 5 MLD de la base de données infofit

Le MLD a été réalisé à l’aide de MySQL Workbench afin de pouvoir générer automatiquement le script de création de la base de données.

### Base de données



Image 6 Scripte SQL de création de la base de données infofit

Voici ci-dessus le script de création de la base de données « infofit » également accessible dans mon dépôt github personnel (<https://github.com/ThierryKoetschet/TPI_Thierry_Koetschet>). Il y a quelques différences par rapport au MCD et au MLD.

Par exemple, on peut remarquer que j’ai dû ajouter à chaque table un attribut « updated\_at » et un attribut « created\_at » au format « timestamp » afin que je puisse insérer des données dans la DB grâce à mon application car Laravel l’exigeait.J’ai également modifié le type des attributs « kcal\_100g », « carbohydrates\_100g », « lipids\_100g » et « proteins\_100g » de la table « foodstuffs » en « float » car j’ai remarqué que les données retournées par l’API Open Food Facts étaient parfois des nombres à virgules et posaient des problèmes pour l’insertion des données dans la base. Enfin, le dernier changement par rapport à la conception de la base de données est l’attribut « period » de la table « users\_has\_foodstuffs » qui permet entre autre de trier les aliments enregistrer par l’utilisateur en fonction de la période dans laquelle il les a consommés.

### Moodboard



Image 7 Moodboard de l'application

Le moodboard a été réalisé à l’aide de photoshop. Il a principalement servi de croquis pour créer la palette de couleur et les logos utilisé dans le site web.

### Maquettes

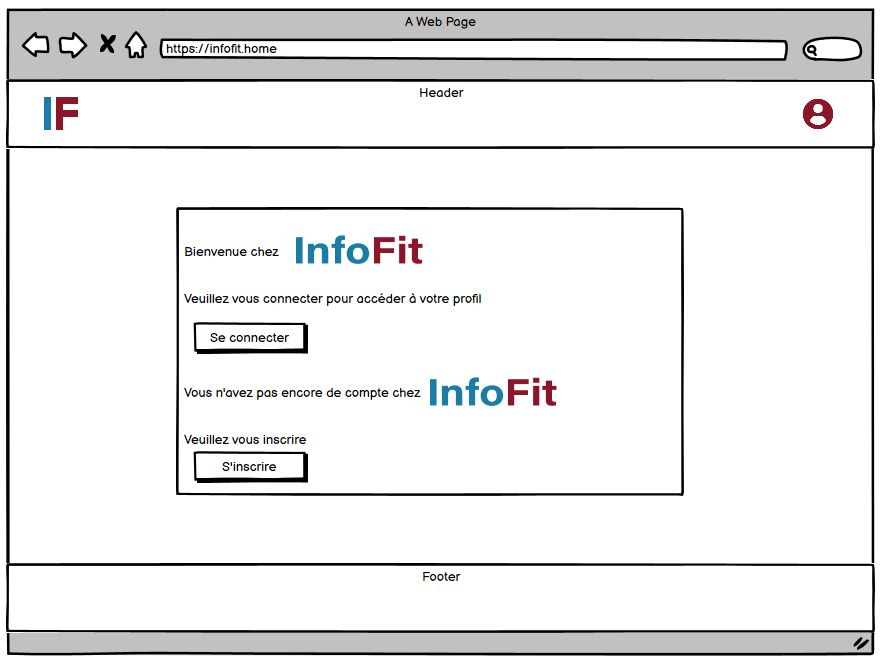


Image 8 Page d'accueil de la maquette

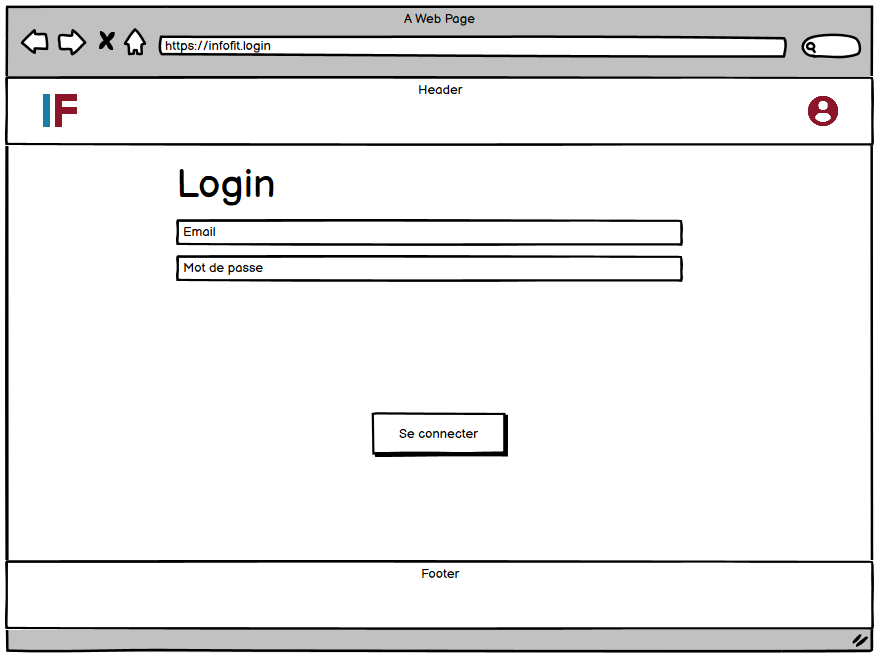


Image 9 Page de login de la maquette



Image 10 Page de register de la maquette

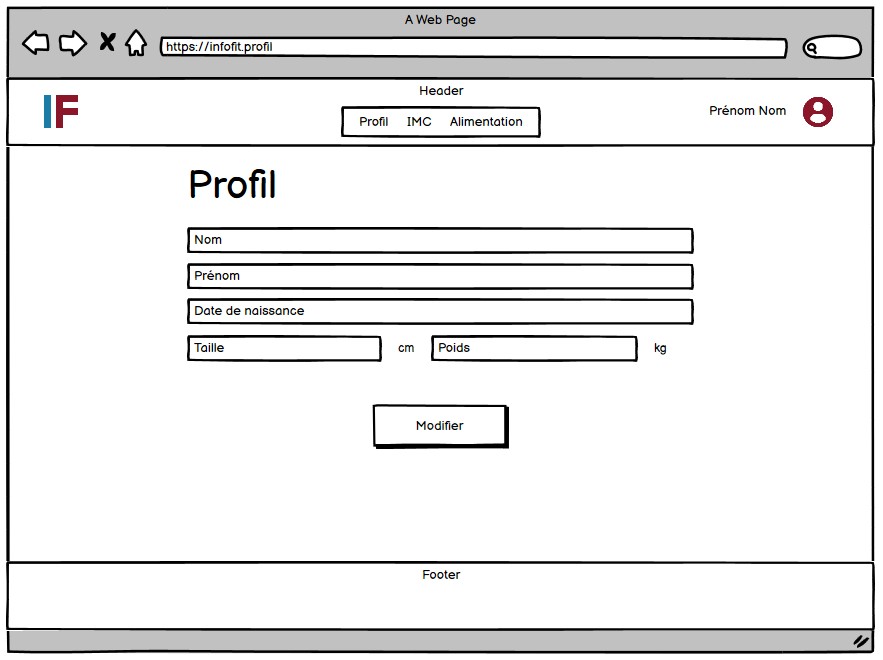


Image 11 Page profil utilisateur de la maquette



Image 12 Page de calcul de l'IMC de la maquette

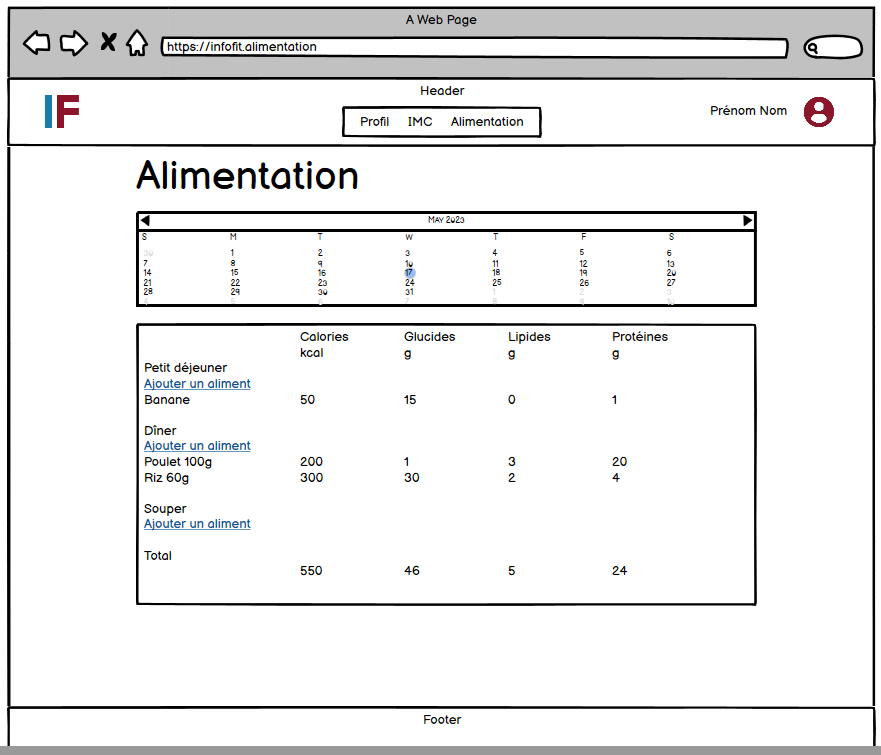


Image 13 Page alimentation de la maquette

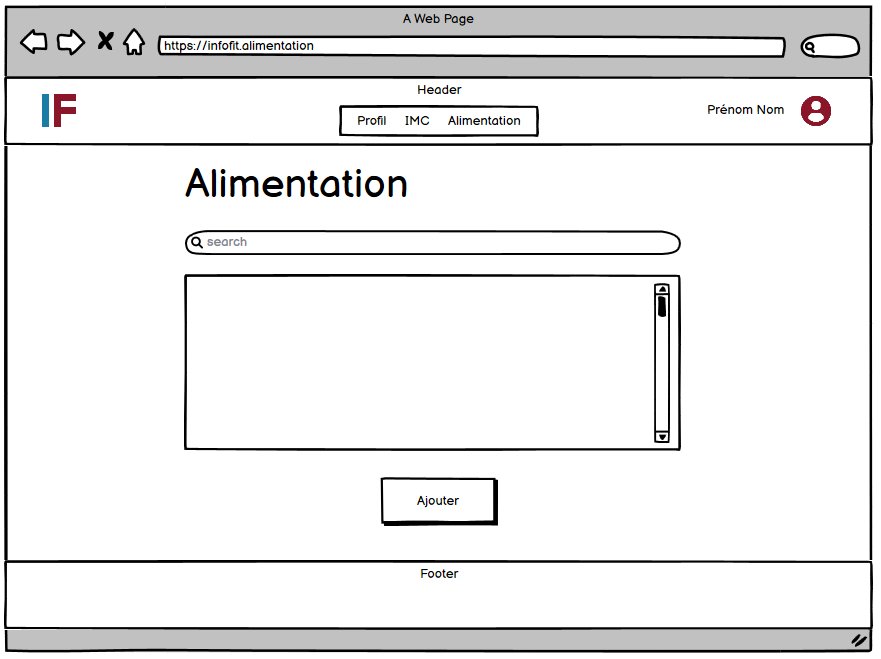


Image 14 Page d'ajout d'aliment de la maquette

Ces maquettes m’ont servi de base tout au long de l’implémentation de mon application, cependant dans la réalisation de certaines fonctionnalités, je me suis rendu compte que je ne pouvais pas copier la maquette complètement. Par exemple, la page sur l’indice de masse corporelle ne ressemble pas à la maquette car j’avais tout d’abord oublié d’y ajouter le graphique d’évolution du poids, puis j’ai également réalisé que mon envie d’implémenter un slider, qui n’était pas mentionner dans mon cahier des charges, allait me demander trop de temps.

## Stratégie de test

Dans le cadre de ce projet, je pense qu’une stratégie relativement simple est un choix judicieux. En effet, l’application est composée de trois principales fonctionnalités : un moyen d’authentification en tant qu’utilisateur et de saisie de données personnelles, un calculateur d’indice de masse corporelle et un tableau permettant d’enregistrer la consommation alimentaire journalière de l’utilisateur. Afin de vérifier le fonctionnement de l’application et de la base de données, il suffit d’effectuer des tests fonctionnels sur ces trois fonctionnalités en commençant par l’authentification car les deux autres fonctionnalités ont besoin de certaines informations enregistrées par l’utilisateur.

Je vais également tester les routes de mon application, c’est-à-dire que tous les liens fonctionnent correctement et que toutes les pages du site sont accessibles.

J’ai créé un utilisateur dont je me suis servi au cours de la réalisation de mon application contenant déjà une certaine quantité d’informations. Celui-ci sera utilisé au dans la réalisation de mes tests. Voici ces identifiants :

* Email : [thierry.koetschet@cpnv.ch](mailto:thierry.koetschet@cpnv.ch)
* Mot de passe : Pa$$w0rd

## Risques techniques

Le risque technique principal est l’apprentissage d’un nouveau framework. En effet, j’ai choisi de réalisé mon site web avec le framework PHP Laravel avec lequel je n’avais encore jamais travaillé. L’apprentissage de l’utilisation d’un tel outil est assez fastidieux et demande beaucoup de temps. Sachant que le temps est une denrée précieuse dans le TPI, il va s’en dire que je prends un paris osé en partant sur cette voie. L’une des raisons qui m’a orienté sur cette décision est que l’un de mes camarades de classe, Pablo Zubieta, avait déjà réalisé son projet de pré-TPI avec Laravel et me l’a vivement recommandé. Il m’a également fait une présentation de l’outil au début du TPI ce qui n’a fait que renforcer mon choix.

Un autre risque auquel je fais généralement face dans de tels projets est que j’ai tendance à trop vouloir en faire, c’est-à-dire à me rajouter du travail supplémentaire, pas forcément nécessaire, simplement car j’ai envie de rendre un produit fini le plus complet possible. Généralement, ce travail supplémentaire impact le temps total à disposition et fait que la fin du projet est un peu tendue.

Le fait de transcrire ces risques techniques de manière écrite me permet d’en prendre conscience et d’y faire particulièrement attention pendant la durée du travail de TPI.

## Planification

## Dossier de conception

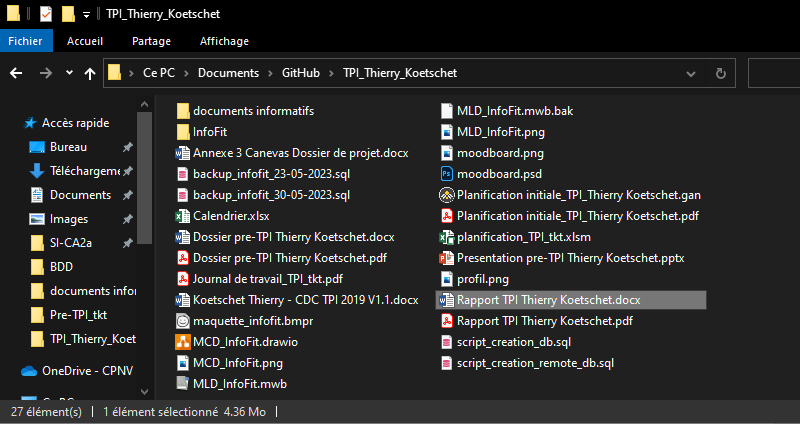


Image 15 Capture du dossier de TPI

# Réalisation

## Dossier de réalisation

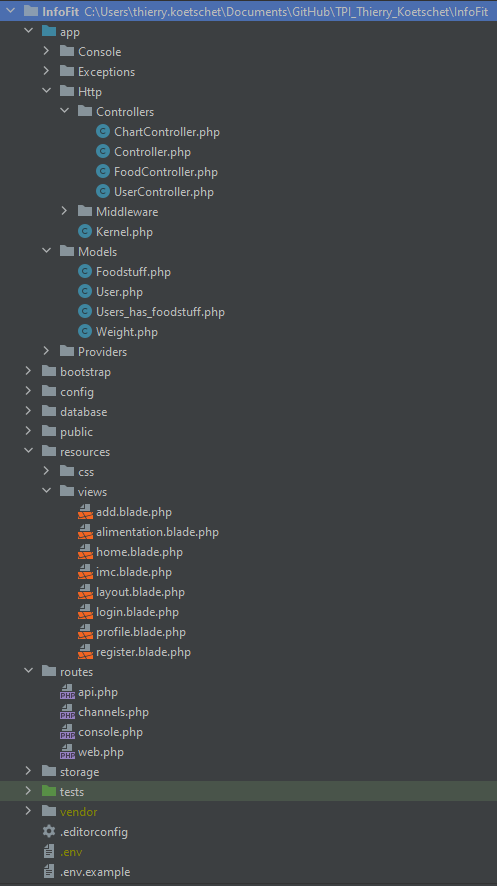


Image 16 Dossier du projet InfoFit

Vous trouverez ci-dessus une capture d’écran du dossier de mon application « InfoFit » avec les principaux dossiers et fichiers visibles.

Tout d’abord, le dossier InfoFit\resources\**views** contient tous les fichiers impactant le front end du site web. Il contient par exemple le gabarit (layout.blade.php) qui affiche le header et le footer de chaque page du site. Le reste des fichiers de ce dossier sont les différentes pages du site comme la page d’accueil (home.blade.php) ou encore la page d’enregistrement d’un utilisateur (register.blade.php). Ces fichiers sont principalement composés de code en HTML et CSS avec un peu de JavaScript et de PHP. Cependant, la majorité du CSS du site est contenu dans les fichiers **styles.css** et **charts.css** du dossier InfoFit\public\css. Les images, donc le logo et les différentes icônes, sont quant à elles contenues dans le dossier InfoFit\public\assets.

Un autre fichier important est le fichier InfoFit\routes\**web.php** car il contient toutes les routes de l’application, c’est-à-dire qu’il permet de faire le lien entre toutes ces vues et permet la transmission de données d’une page à l’autre.

Le dossier InfoFit\app\**Models** contient tous les fichiers permettant les interactions avec la base de données comme le fichier **User.php** qui permet d’insérer, modifier ou supprimer un utilisateur de l’application par exemple.

Le dossier InfoFit\app\Http\**Controllers** contient tous les fichiers servant au fonctionnement global de l’application. Le fichier **ChartController.php** contient les fonctions affichant le graphique d’évolution et calculant l’indice de masse corporelle. Le fichier **FoodController.php** permet d’afficher les aliments ajoutés par l’utilisateur par rapport à la date et la période d’ajout, de rechercher des aliments via une API et de supprimer des aliments ajoutés. Enfin, le fichier **UserController.php** contient une multitude de fonctions telles que la création, l’authentification et la suppression d’un utilisateur, la modification du profil utilisateur pour entre autre pouvoir traquer l’évolution du poids et enfin changer le mot de passe de son compte utilisateur.

## Code source

Tout le code source est disponible dans le dossier InfoFit de mon dépôt GitHub (<https://github.com/ThierryKoetschet/TPI_Thierry_Koetschet>). J’ai fait une petite sélection de fonctions qui mérite d’avoir une explication dans ce rapport.

### Fonction « store » de UserController.php

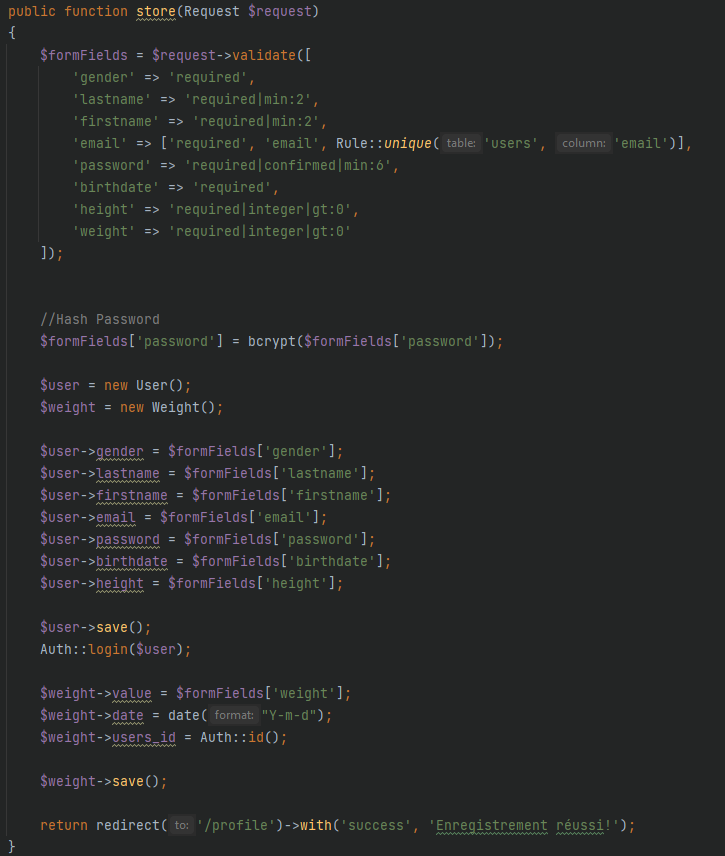


Image 17 Capture d'écran de la fonction store

Cette fonction a pour but d’enregistrer un utilisateur qui utilise l’application InfoFit pour la première fois. On remarque qu’elle possède un paramètre « Request $request » qui permet de récupérer les informations passées par l’envoi du formulaire en méthode post. Elle va alors commencer par validé tous les inputs renvoyés puis, si tous les inputs sont corrects, elle va enregistrer un nouvel utilisateur dans la base de données avec un mot de passe hashé et également un poids correspondant à l’utilisateur. Enfin, on authentifie l’utilisateur puis en retourne la page « profile.blade.php ».

### Fonction « searchFoodstuff » de FoodController.php

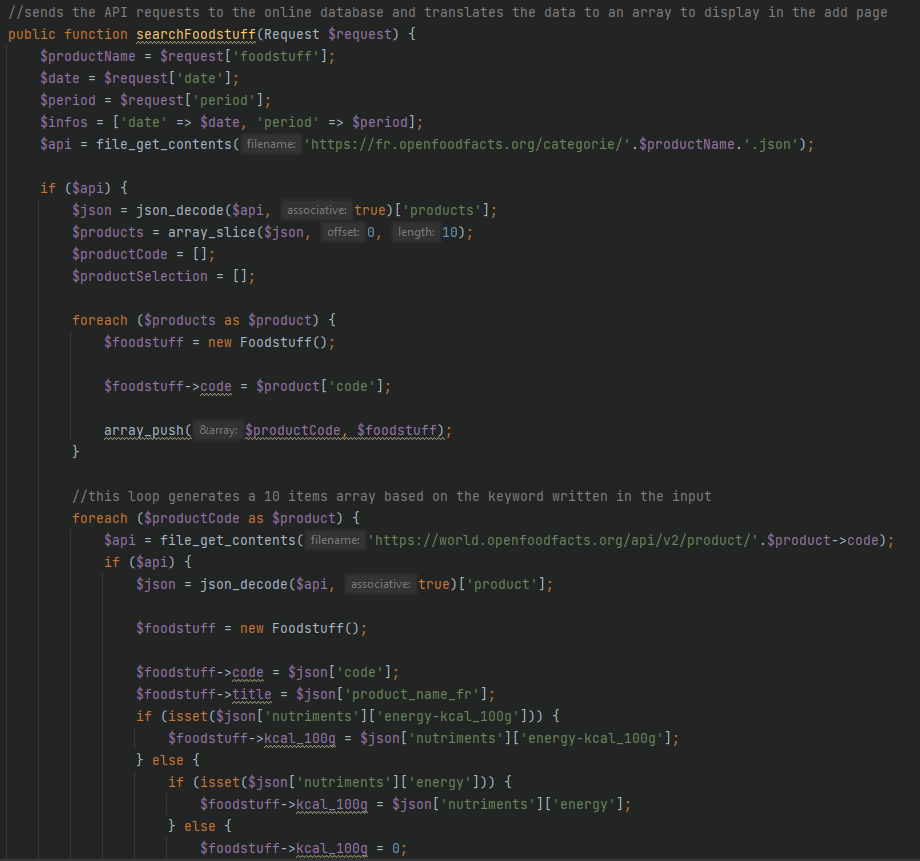


Image 18 Première partie capture de la fonction searchFoodstuff

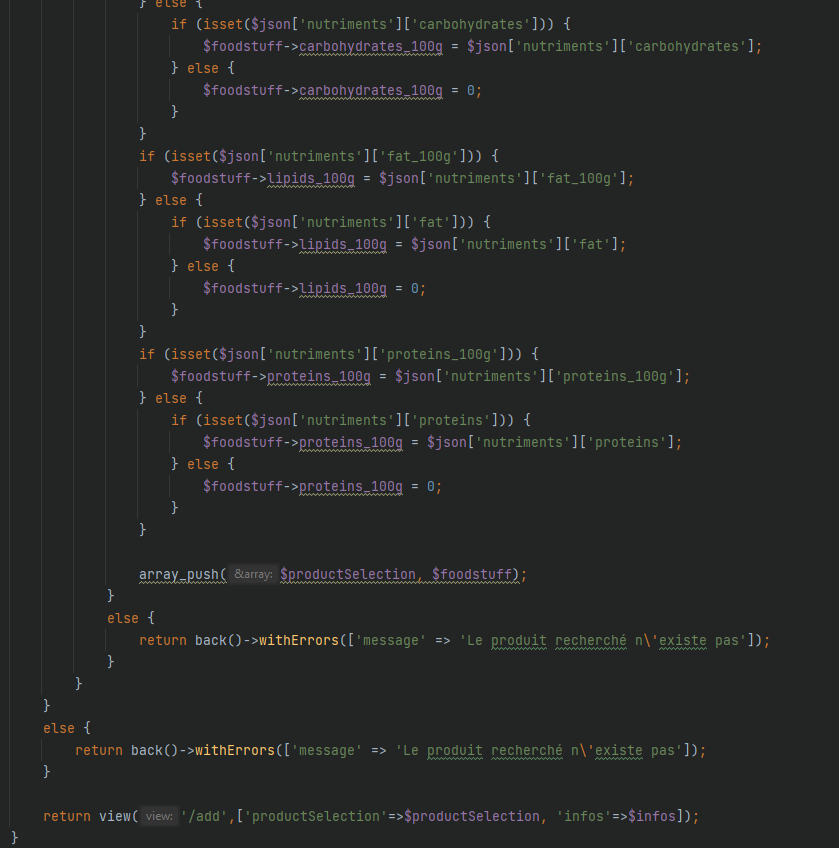


Image 19 Deuxième partie capture de la fonction searchFoodstuff

Tout comme la fonction « store », on récupère les données envoyées dans les inputs du formulaire grâce au paramètre « Request $request ». Puis elle va faire une première requête API qui va rechercher le code des produits qui ont pour catégorie le texte saisi dans l’input du formulaire. La méthode « file\_get\_contents() » permet de récupérer les informations obtenues grâce à la requête API puis la méthode « json\_encode() » permet de transformer ce grande quantité de texte au format JSON afin d’avoir un tableau associatif lisible en PHP. On réduit le nombre d’objet dans ce tableau à 10 afin de ne pas avoir une trop grande liste puis on effectue une nouvelle requête API produit par produit avec leur code qu’on a récolté précédemment. La fonction va alors créer un nouveau tableau associatif qui regroupera uniquement les informations nécessaires à afficher dans la vue, c’est-à-dire le nombre de kcal, glucides, lipides, protéines par 100 grammes ainsi que le code et nom du produit. Enfin, on renvoie ce tableau ainsi que différentes informations telles que la date sélectionnée et la période dans la vue « add.blade.php ».

### Fonction « lineChart » de ChartController.php

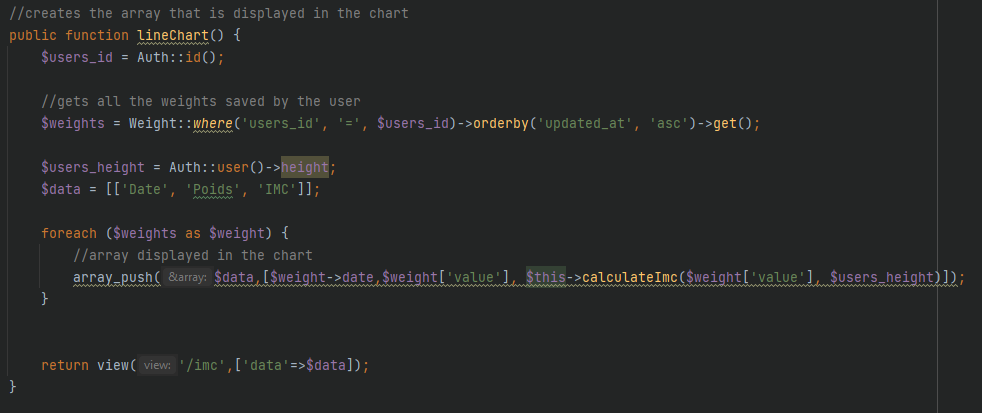


Image 20 Capture de la fonction lineChart

Dans la variable « $weights », on récupère un tableau de la table « weights » dans notre base de données grâce à une requête avec comme critère l’id de l’utilisateur et dans l’ordre ascendant. On enregistre également la taille de l’utilisateur dans la variable « $users\_height ». On effectue alors une boucle « foreach » afin de créer notre tableau associatif contenant les informations affichées dans le graphique de la vue IMC.

## Aperçu de l’application

### Page d’accueil



Image 21 Capture de la page d'accueil de InfoFit

### Login

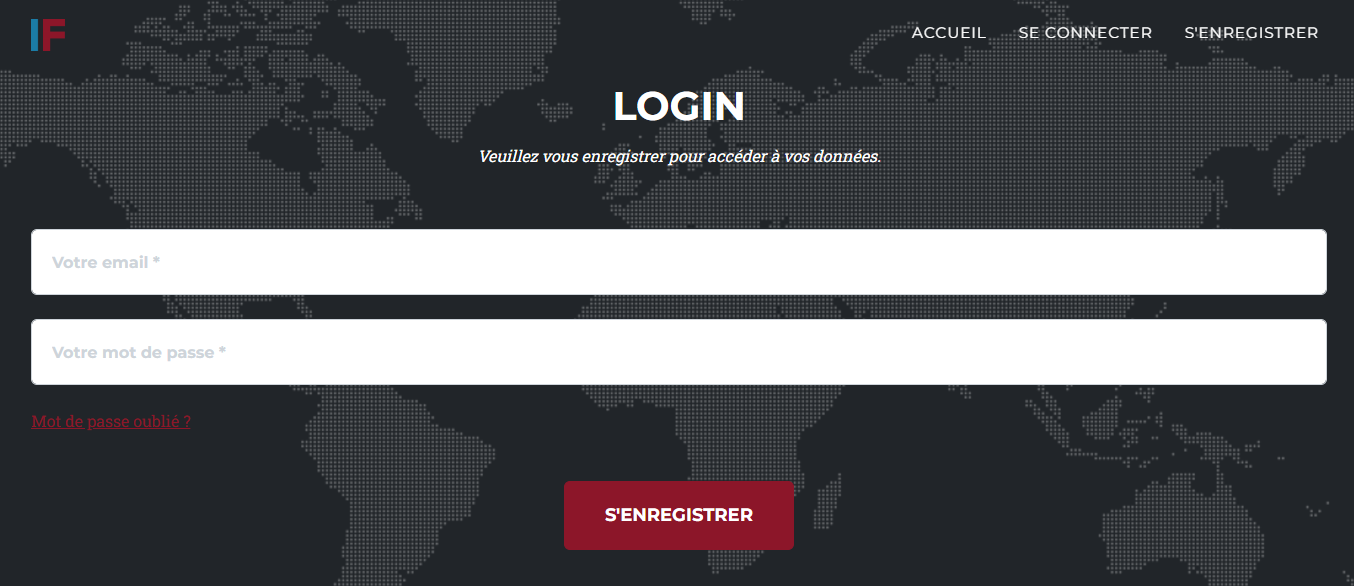


Image 22 Capture de la page de login de InfoFit

### Register

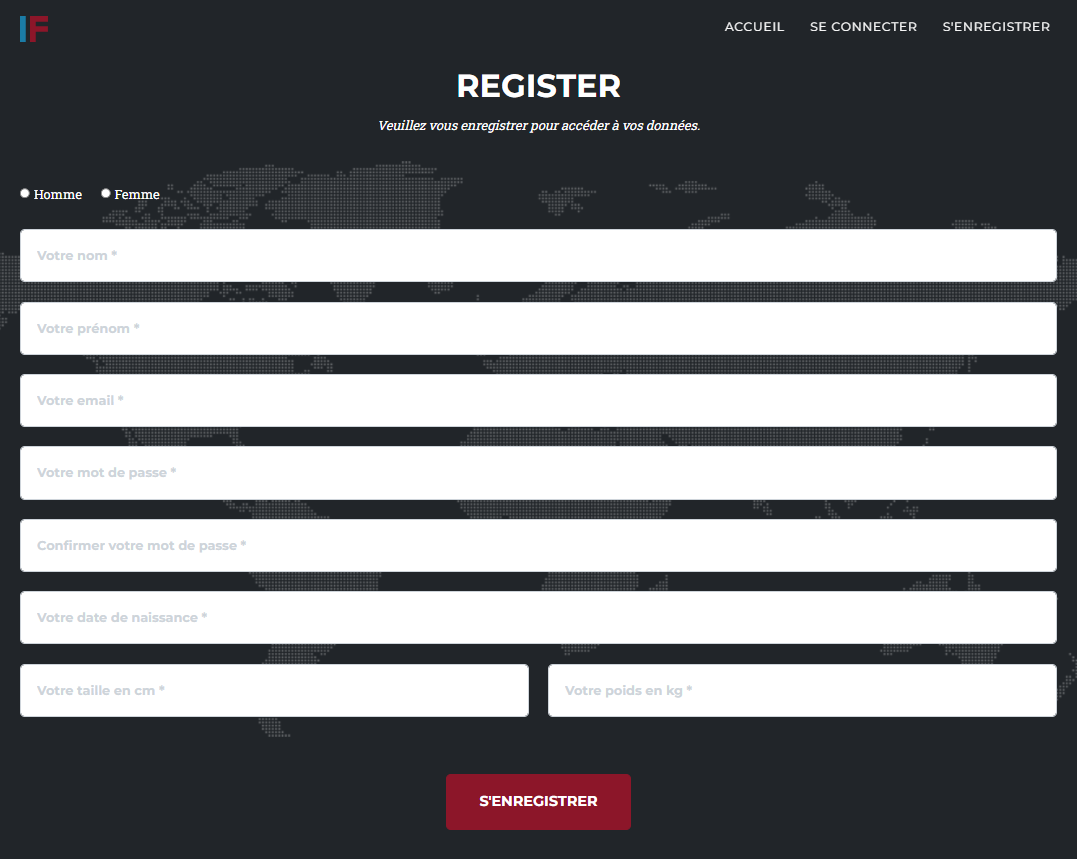


Image 23 Capture de la page de register de InfoFit

### Profil

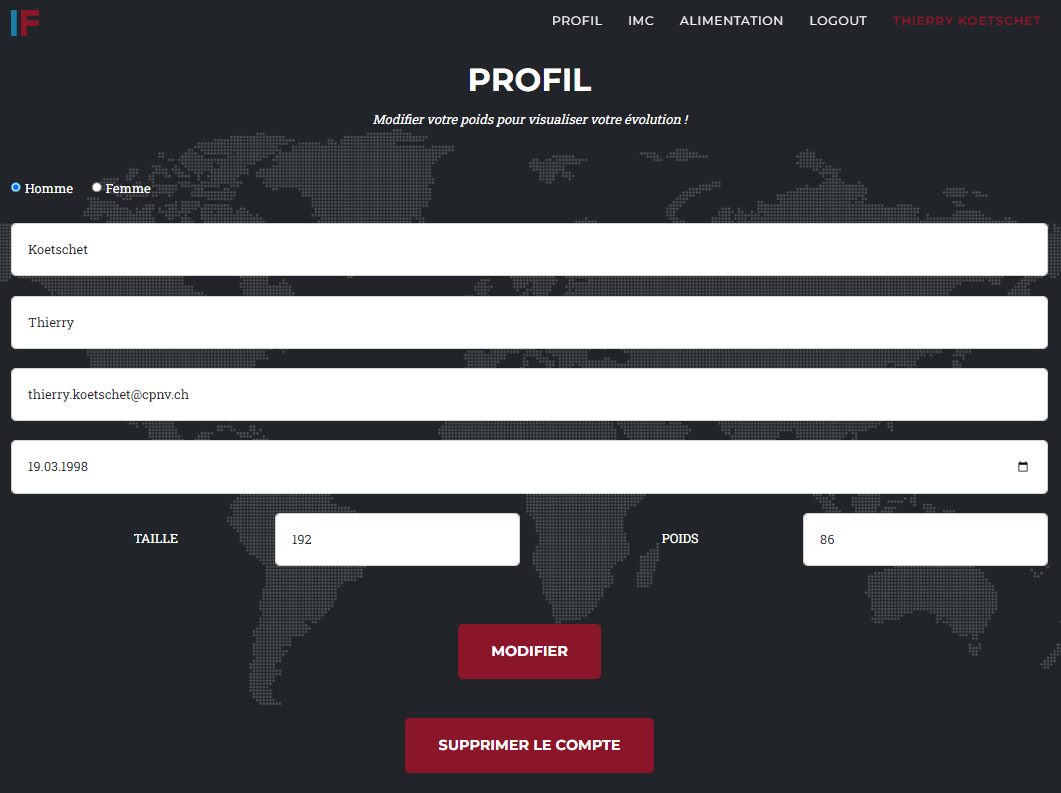


Image 24 Capture de la page de profil de InfoFit

### IMC

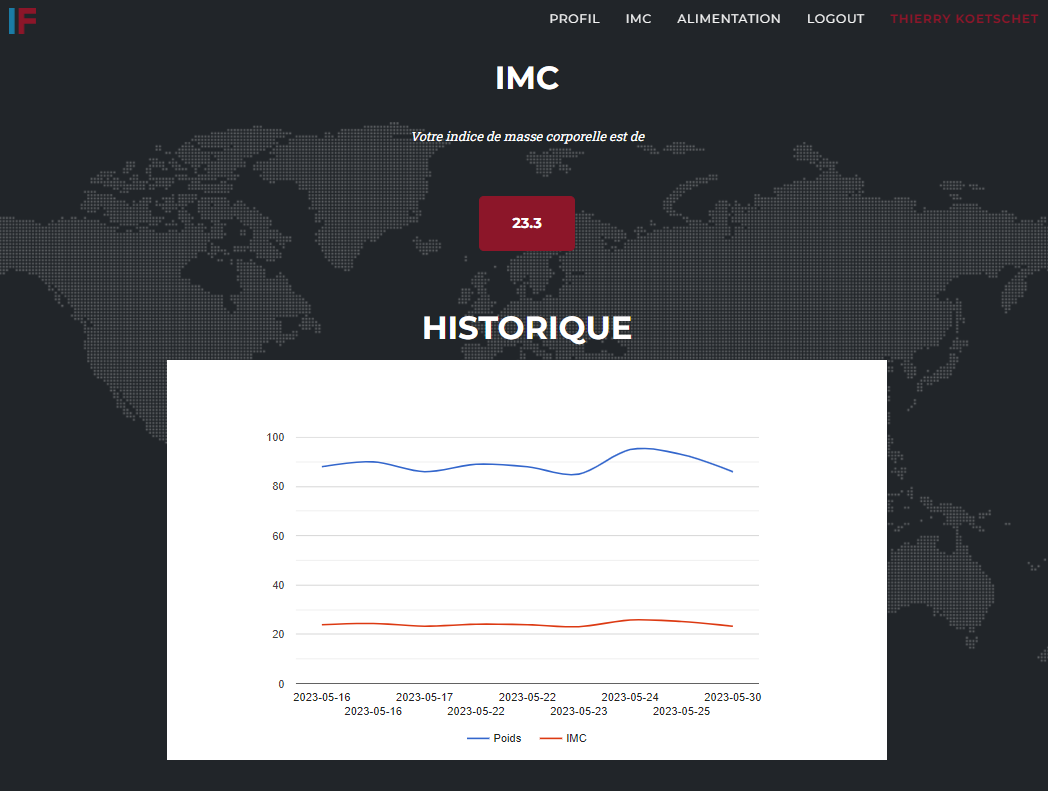


Image 25 Capture de la page d'IMC de InfoFit

### Alimentation

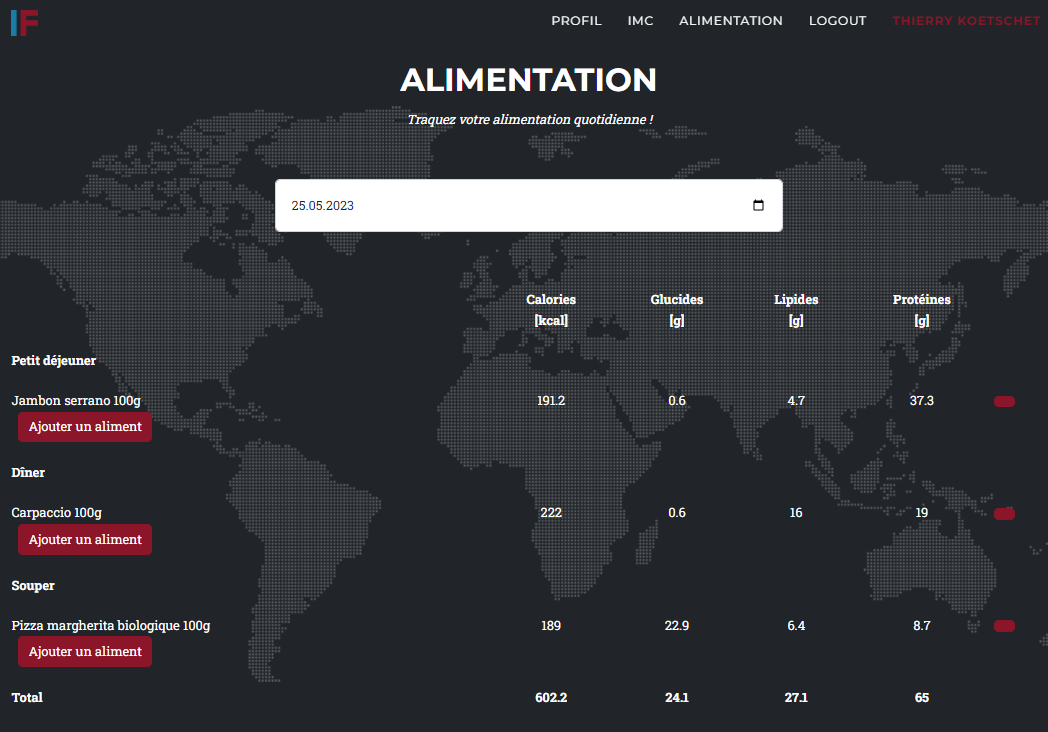


Image 26 Capture de la page d'alimentation de InfoFit

### Ajouter un aliment



Image 27 Capture de la page d'ajout d'aliment de InfoFit

## Hébergement sur le serveur web

## Description des tests effectués

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Test | Résultat | Validation |
| Tests des routes | | | |
| 30.05.2023 | Page d’accueil accessible. | La commande **php artisan serve** ouvre directement la page d’acueil.  Le lien « accueil » de la barre de navigation fonctionne. | Free symbole de coche ronde avec fond transparent 17177933 PNG with Transparent  Background |
| 30.05.2023 | Page de login accessible. | Le lien dans la barre de navigation fonctionne.  Le bouton de la page d’accueil fonctionne. | Free symbole de coche ronde avec fond transparent 17177933 PNG with Transparent  Background |
| 30.05.2023 | Page de register accessible. | Le lien dans la barre de navigation fonctionne.  Le bouton de la page d’accueil fonctionne. | Free symbole de coche ronde avec fond transparent 17177933 PNG with Transparent  Background |
| 30.05.2023 |  | Le lien fonctionne correctement. | Free symbole de coche ronde avec fond transparent 17177933 PNG with Transparent  Background |
| 30.05.2023 | Bouton « modifier » de la page profil. | Il renvoie à la page « Profil ».. | Free symbole de coche ronde avec fond transparent 17177933 PNG with Transparent  Background |
| 30.05.2023 | Bouton « supprimer le compte » de la page profil. | Il renvoie à la page d’accueil. | Free symbole de coche ronde avec fond transparent 17177933 PNG with Transparent  Background |
| 30.05.2023 | Bouton « Profil » de la barre de navigation | Il renvoie à la page « Profil ». | Free symbole de coche ronde avec fond transparent 17177933 PNG with Transparent  Background |
| 30.05.2023 | Bouton « IMC » de la barre de navigation | Il renvoie à la page « IMC ». | Free symbole de coche ronde avec fond transparent 17177933 PNG with Transparent  Background |
| 30.05.2023 | Bouton « Alimentation » de la barre de navigation | Il renvoie à la page « Alimentation ». | Free symbole de coche ronde avec fond transparent 17177933 PNG with Transparent  Background |
| 30.05.2023 | Bouton « Logout » de la barre de navigation | Il renvoie à la page d’accueil. | Free symbole de coche ronde avec fond transparent 17177933 PNG with Transparent  Background |
| 30.05.2023 | Input de date de la page alimentation | Il renvoie à la page « Alimentation » avec la date sélectionnée. | Free symbole de coche ronde avec fond transparent 17177933 PNG with Transparent  Background |
| 30.05.2023 | Boutons « Ajouter un aliment » de la page « Alimentation ». | Ils renvoient à la page « Ajouter un aliment ». | Free symbole de coche ronde avec fond transparent 17177933 PNG with Transparent  Background |
| 30.05.2023 | Bouton « chercher » de la page « Ajouter un aliment ». | Il renvoie à la page « Ajouter un aliment » s’il y a quelque chose de saisi dans l’input, sinon il renvoie une erreur. Parfois, l’API ne trouve pas non plus le texte saisi dans l’input et affiche une erreur dans l’application | Croix-Rouge Marque PNG Transparent - PNG All |
| Tests fonctionnels | | | |
| 30.05.2023 | Login |  | Free symbole de coche ronde avec fond transparent 17177933 PNG with Transparent  Background |
| 30.05.2023 | Login avec un email inexistant |  | Free symbole de coche ronde avec fond transparent 17177933 PNG with Transparent  Background |
| 30.05.2023 | Login avec un faux mot de passe |  | Free symbole de coche ronde avec fond transparent 17177933 PNG with Transparent  Background |
| 30.05.2023 | Register |  | Free symbole de coche ronde avec fond transparent 17177933 PNG with Transparent  Background |
| 30.05.2023 | Logout | Retour à la page d’accueil et nom d’utilisateur plus affiché au coin de l’écran. | Free symbole de coche ronde avec fond transparent 17177933 PNG with Transparent  Background |
| 30.05.2023 | Modification du profil | Nom, prénom, genre, email, date de naissance, taille et poids modifiable | Free symbole de coche ronde avec fond transparent 17177933 PNG with Transparent  Background |
| 30.05.2023 | Mot de passe oublié | Mot de passe réinitialisable sans la demande de l’ancien mot de passe | Free symbole de coche ronde avec fond transparent 17177933 PNG with Transparent  Background |
| 30.05.2023 | Suppression du compte en cascade | Compte supprimé ainsi que tous enregistrement faits par celui-ci dans la base de données. | Free symbole de coche ronde avec fond transparent 17177933 PNG with Transparent  Background |
| 30.05.2023 | Graphique d’évolution du poids | Affiche chaque enregistrement correctement | Free symbole de coche ronde avec fond transparent 17177933 PNG with Transparent  Background |
| 30.05.2023 | Changement de date du tableau alimentation | La page se rafraîchit et enregistre la nouvelle date | Free symbole de coche ronde avec fond transparent 17177933 PNG with Transparent  Background |
| 30.05.2023 | Recherche d’un aliment via l’API | La recherche ne fonctionne pas à chaque fois. Par exemple, le recherche avec un input avec une valeur nulle provoque une erreur. Certains aliments sont également introuvables via l’API et provoquent une erreur. | Croix-Rouge Marque PNG Transparent - PNG All |
| 30.05.2023 | Ajout d’un aliment dans le tableau alimentation | L’aliment s’ajoute dans le tableau alimentation à la bonne date et à la bonne période. | Free symbole de coche ronde avec fond transparent 17177933 PNG with Transparent  Background |
| 30.05.2023 | Suppression d’un aliment du tableau alimentation | L’aliment se supprime correctement et l’enregistrement disparaît de la base de données. | Free symbole de coche ronde avec fond transparent 17177933 PNG with Transparent  Background |

## Erreurs restantes

La page « Ajouter un aliment » de l’application contient encore plusieurs erreurs : Cliquez sur le bouton « chercher » avec un input vide va causer une erreur dans l’application. De plus, l’affichage de l’input « quantité » de liste d’aliments retournés n’est pas visible en mode étroit et certains aliments recherchés dans l’API n’existe pas et cause donc une erreur au niveau de l’application.

J’ai également remarqué que certaines pages de l’applications posent problème lorsque que l’on n’est pas logué. Par exemple, la page « profile » est problématique si l’on saisit l’URL <http://127.0.0.1:8000/alimentation/profile> dans le navigateur web sans être authentifié.

## Liste des documents fournis

Tous les documents fournis au cours ou à la fin de ce projet sont les suivants :

* Une planification initiale fournie le 2 mai 2023
* L’état d’avancée du journal de travail au format PDF chaque mardi et jeudi sur toute la durée du projet
* L’état d’avancée du rapport de TPI au format PDF chaque jeudi sur toute la durée du projet
* Le code source de l’application fourni le 31 mai 2023 sur mon dépôt GitHub (<https://github.com/ThierryKoetschet/TPI_Thierry_Koetschet>)
* Un rapport de TPI complet et un journal de travail au format PDF fourni le 31 mai 2023 à chaque expert et au chef de projet par email et une version imprimée de chaque pour l’expert numéro 1
* Une procédure d’installation et de mise en service de l’application fournie le 31 mai 2023 en annexe au rapport de TPI
* Un accès à l’application sur la plateforme mycpnv.ch fourni le 31 mai 2023

La date de rendu de certains de ces documents diffère du cahier des charges et est retardée au 31 mai 2023 pour cause de problèmes de connexion internet survenus le 30 mai 2023 pendant une période de 2 heures dus à la casse d’une fibre optique lors de travaux à proximité du bâtiment du CPNV de Sainte-Croix.

# Conclusions

Pour conclure, je pense avoir atteint les objectifs fixés autant dans mon cahier des charges comme les objectifs personnels que je m’étais fixés au début du projet. En effet, toutes les fonctionnalités décrites dans le cahier des charges semblent être respectées. Le site contient un système d’authentification sécurisé, la page de calcul d’indice de masse corporelle fonctionne correctement et enfin la page permettant aux utilisateurs de l’application d’enregistrer leurs données alimentaires journalières est opérationnelle avec des requêtes API allant chercher des informations sur l’API Open Food Facts. Je pense également avoir acquis de nombreuses connaissances sur l’utilisation du framework Laravel qui était mon objectif personnel à atteindre. J’espère que ces connaissances auront une utilité dans ma future carrière professionnelle.

Cependant, je me rends compte qu’il y a certains points de la réalisation de projet que je pourrais améliorer comme la gestion de mon temps qui pourrait être meilleure. La fin de ce projet de TPI fut quelque peu intensive car j’ai consacré un peu trop de temps à l’implémentation de l’application ce qui fait que j’ai manqué de temps pour la rédaction de la documentation et ai dû faire un peu de travail à domicile.

En somme, je suis satisfait du produit que je remets car il s’agit d’un outil je vais utiliser quotidiennement dans mon activité sportive afin de suivre mes progrès et mes informations personnelles. Et comme mon ambition future est de potentiellement lancer mon entreprise d’informatique, je pourrai m’inspirer et utiliser tous ces outils appris lors de ce projet de TPI.

# Annexes

## Cahier des charges

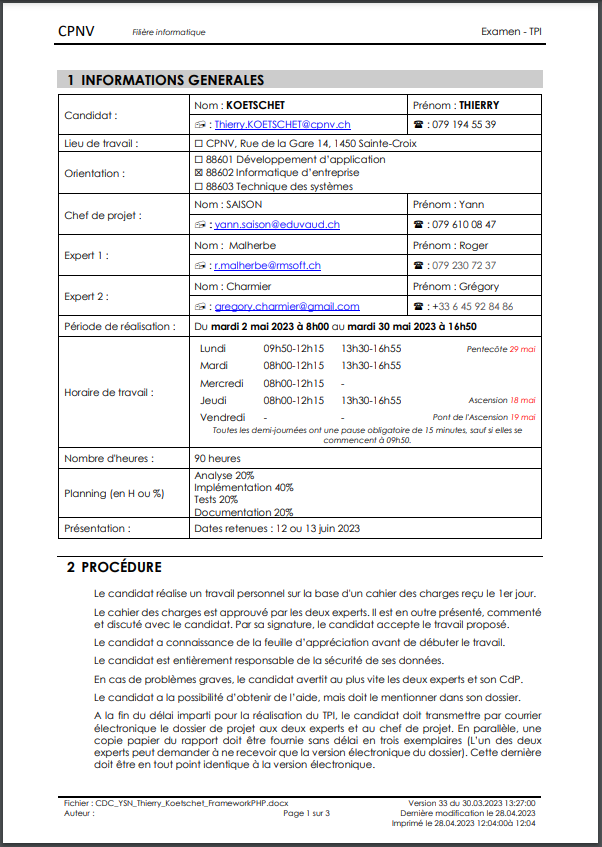


Image 28 Première page du cahier des charges

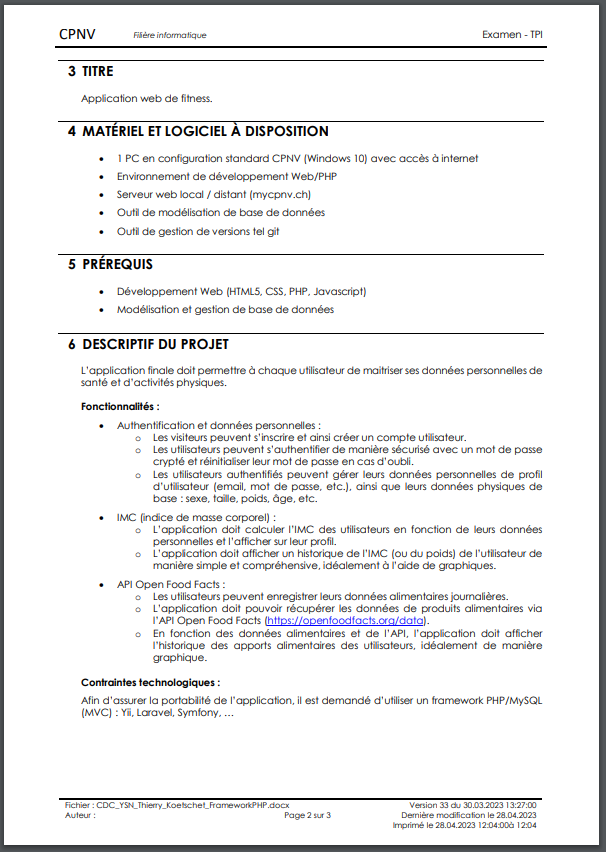


Image 29 Deuxième page du cahier des charges

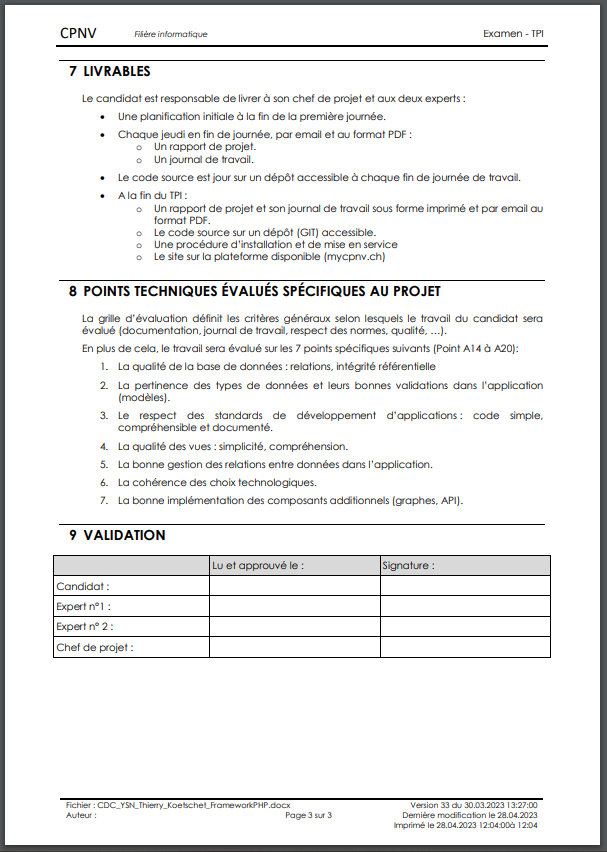


Image 30 Troisième page du cahier des charges

## Horaire de travail

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Lundi | Mardi | Mercredi | Jeudi | Vendredi |
| 08:00 – 09:35 |  |  |  |  |  |
| 09:50 – 12:15 |  |  |  |  |  |
| Pause midi | | | | | |
| 13:30 – 15:05 |  |  |  |  |  |
| 15:20 – 16:55 |  |  |  |  |  |

## 

## Résumé du travail

Ce projet de TPI est l’achèvement d’une formation de deux ans en tant qu’informaticien d’entreprise et consiste en la réalisation d’un travail pratique d’environ 90 heures. Mon cahier des charges me fixait comme objectifs de réaliser un site web avec l’utilisation d’un framework PHP et devant contenir trois principales fonctionnalités : La première fonctionnalité demande à l’utilisateur de s’authentifier afin de pouvoir accéder aux deux autres fonctionnalités. La seconde permet aux utilisateurs de calculer leur indice de masse corporelle en fonction du poids saisi dans leur profil et d’avoir une représentation graphique de leur évolution. La troisième un genre de calendrier alimentaire permettant aux utilisateurs de récupérer de nombreuses informations sur les différents aliments qu’ils ont consommé dans leur journée avec l’aide de requêtes API.

La planification initiale pour la réalisation de ce projet a été faite de la manière suivante : le travail a été divisé en quatre sprint. Le premier sprint se compose de la partie analyse et conception du travail qui représente globalement l’apprentissage de l’utilisation de nouveaux outils tels que le framework Laravel ou encore l’API Open Food Facts ainsi que tout le travail de conception de la base de données, le moodboard et la maquette de l’application. Le sprint suivant est le plus conséquent car il contient toute la partie implémentation du projet, c’est-à-dire l’écriture du code source de l’application. Celle-ci se compose de vues écrite en HTML, CSS et JavaScript, de contrôleurs contenant toutes les fonctions nécessaires au fonctionnement de l’application, de modèles permettant de faire la relation entre le site web et la base données et enfin de routes faisant le lien entre les différentes vues de l’application. Le troisième sprint concerne tout ce qui se rapporte aux tests de l’application, donc la stratégie de test, la rédaction et l’application des tests. Pour finir, le quatrième sprint contient l’écriture de la documentation et l’organisation du projet. Ce sprint s’étale sur toute la durée du projet.

Le résultat de ces quatre semaines de travail uniquement consacrée à ce projet sont un site web répondant aux critères posés dans le cahier des charges, une base de données qui a connu une certaine évolution au cours de l’implémentation de l’application mais qui fonctionne parfaitement avec celle-ci, un rapport de TPI et un journal de travail contenant chaque tâche effectuée au cours du travail. Je pense également avoir progressé sur le plan personnel au niveau du développement web et plus particulièrement sur l’utilisation de framework PHP qui je pense pourrait intéresser de futurs employeurs.

## Sources – Bibliographie

Site Yii Framework :

<https://www.yiiframework.com/> consulté le 02.05.2023

Site Laravel :

<https://laravel.com/> consulté le 02.05.2023

Site Open Food Facts :

<https://ch-fr.openfoodfacts.org/> consulté le 02.05.2023

Site MyFitnessPal :

<https://www.myfitnesspal.com/fr> consulté le 02.05.2023

Site Composer :

<https://getcomposer.org/download/> consulté le 02.05.2023

API denrées alimentaires :

<https://developer.edamam.com/food-database-api-docs> consulté le 02.05.2023

<https://world.openfoodfacts.org/data> consulté le 04.05.2023

Source d’information pour la stratégie de tests :

https://www.atlassian.com/fr/continuous-delivery/software-testing/types-of-software-testing consulté le 09.05.2023

Liens pour templates HTML/CSS :

<https://startbootstrap.com/themes> consulté le 09.05.2023

Liens pour graphiques JS :

[https://developers.google.com/chart?hl=fr](https://developers.google.com/chart/interactive/docs/gallery?hl=fr) consulté le 16.05.2023

## Journal de travail

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Date | Semaine | Activité | Heures | Description | Remarques |
| 02.05.2023 | 18 | 51 Absence | 0:30 | Présentation du cahier des charges par M. Malherbe |  |
| 02.05.2023 | 18 | 41 Réalisation de la planification initiale | 3:35 | Réalisation de la planification initiale |  |
| 02.05.2023 | 18 | 11 Analyse du framework Laravel | 1:35 | Analyse des différents frameworks proposés dans le cahier des charges et choix de Laravel |  |
| 02.05.2023 | 18 | 12 Installation du framework Laravel | 1:35 | Installation du framework et création d'un projet | Problème rencontré au niveau du fichier php.ini car il manquait une exension mais le problème a été réglé |
| 03.05.2023 | 18 | 13 Analyse de l'API Open Food Facts | 1:35 | Analyse d'API pouvant être utiliser dans mon application |  |
| 03.05.2023 | 18 | 14 Moodboard de l'application | 1:35 | Création du moodboard (logos, palette de couleurs) |  |
| 03.05.2023 | 18 | 15 Maquette de l'application | 0:50 | Début de la maquette de l'application | Fichier : maquette\_infofit.bmpr |
| 04.05.2023 | 18 | 15 Maquette de l'application | 1:35 | Fin de la maquette |  |
| 04.05.2023 | 18 | 42 Rédaction de la documentation | 2:05 | Début de mise en forme de la documentation et rédaction de l'introduction |  |
| 04.05.2023 | 18 | 21 Conception du MCD | 1:05 | Conception du MCD avec draw.io | Je ne sais pas encore comment intégrer toute la partie API à ma base de données et à mon application |
| 04.05.2023 | 18 | 22 Conception du MLD | 2:05 | Conception du MLD avec MySQL Workbench |  |
| 04.05.2023 | 18 | 43 Remplissage du journal de travail | 0:15 | Remplissage du journal de travail en fonction du travail réalisé dans la semaine |  |
| 08.05.2023 | 19 | 23 Création de la DB | 0:30 | Génération du script de création de la base de données avec MySQL Workbench et exécution du script sur HeidiSQL |  |
| 08.05.2023 | 19 | 24 Création du projet | 0:35 | Création de l'application avec Laravel |  |
| 08.05.2023 | 19 | 52 Autres | 0:45 | Choix d'un template HTML/CSS pour le site |  |
| 08.05.2023 | 19 | 26 Page d'accueil / Connexion à l'application | 0:35 | Début de la page d'accueil et prise en main du framework |  |
| 08.05.2023 | 19 | 23 Création de la DB | 1:05 | Mise à jour de la base de données | Après discussion avec M. Saison, j'ai mis à jour les MCD, MLD, et la base de données pour y ajouter une table poids afin d'avoir un historique de l'évolution du poids de chaque user. |
| 08.05.2023 | 19 | 25 Création du gabarit | 2:05 | Création du gabarit | Fichier : layout.blade.php |
| 09.05.2023 | 19 | 26 Page d'accueil / Connexion à l'application | 4:00 | Page d'accueil fonctionnelle | Fichier : home.blade.php |
| 09.05.2023 | 19 | 25 Création du gabarit | 2:20 | Gabarit fonctionnel |  |
| 09.05.2023 | 19 | 52 Autres | 0:45 | Visualisation d'une vidéo sur le fonctionnement du login et de la connexion à la base de données avec Laravel |  |
| 09.05.2023 | 19 | 43 Remplissage du journal de travail | 0:05 | Remplissage du journal de travail pour le lundi et le mardi |  |
| 10.05.2023 | 19 | 30 Connexion à la DB / API | 4:00 | Création de la migration de la base de données et connexion à la base fonctionnelle |  |
| 11.05.2023 | 19 | 29 Login / Register | 5:20 | Création d'un enregistrement d'utilisateur | Problème au niveau du modèle User |
| 11.05.2023 | 19 | 42 Rédaction de la documentation | 1:35 | Rédaction de la partie objectif et mise en page |  |
| 11.05.2023 | 19 | 43 Remplissage du journal de travail | 0:15 | Remplissage du journal |  |
| 15.05.2023 | 20 | 29 Login / Register | 5:35 | Fonctions login et register complètes |  |
| 16.05.2023 | 20 | 27 Page IMC | 4:50 | Début sur la page IMC |  |
| 16.05.2023 | 20 | 52 Autres | 0:30 | Rencontre avec M. Charmier (expert no.2) |  |
| 16.05.2023 | 20 | 27 Page IMC | 1:50 | Avance sur la page IMC | Problème pour insérer des données de la base de données dans le graphique |
| 17.05.2023 | 20 | 27 Page IMC | 1:20 | Page IMC terminée | Problème résolu pour l'insertion de données |
| 17.05.2023 | 20 | 42 Rédaction de la documentation | 2:25 | Ajout du MCD, MLD, des maquettes et rédaction de la partie organisation |  |
| 17.05.2023 | 20 | 43 Remplissage du journal de travail | 0:15 | Remplissage journal de travail |  |
| 22.05.2023 | 21 | 30 Connexion à la DB / API | 5:35 | Requêtes API fonctionnent à moitié | Problème au niveau de la taille des données récupérées |
| 23.05.2023 | 21 | 28 Page données alimentaires journalières | 6:55 | Pages de données alimentaires journalières et d'ajout d'aliments presque abouties et ajout de la suppression d'utilisateur |  |
| 23.05.2023 | 21 | 43 Remplissage du journal de travail | 0:15 | Remplissage journal de travail et envoi mail aux experts |  |
| 24.05.2023 | 21 | 30 Connexion à la DB / API | 0:50 | Requêtes API fonctionnent correctement | Correction du bug suite au retour de M. Charmier |
| 24.05.2023 | 21 | 42 Rédaction de la documentation | 3:10 | Avancement de la documentation et modification du journal de travail et du planning pour y ajouter les 5 minutes entre chaque période de 45 minutes |  |
| 25.05.2023 | 21 | 42 Rédaction de la documentation | 4:00 | Avancement de la documentation |  |
| 25.05.2023 | 21 | 28 Page données alimentaires journalières | 1:35 | Page de données alimentaires journalières complète |  |
| 25.05.2023 | 21 | 52 Autres | 1:35 | Page de profil utilisateur complète avec la modification des infos personnelles |  |
| 29.05.2023 | 22 | 52 Autres | 3:00 | Avancée de la documentation à la maison | Ecriture de la conclusion, le résumé et du manuel d'installation |
| 30.05.2023 | 22 |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Glossaire

|  |  |
| --- | --- |
| Acronyme | Explication |
| API | Application Programming Interface est un ensemble de composants logiciels facilitant le développement d’application. |
| CRUD | Acronyme pour « Create Read Update Delete » représentant la relation entre la base de données et l’application. |
| CSS | Cascading Style Sheets est language de programmation utile à la mise en page d’un site web. |
| DB | Database ou base de données |
| Framework | Ensemble de composants logiciels servant à la programmation d’application. |
| Front end | Eléments visibles d’une site web |
| HTML | Hypertext Markup Language est un langage de programmation très utilisé dans le développement web. |
| IDE | Environnement de développement |
| JSON | JavaScript Object Notation est un format d’échange de données. |
| Moodboard | Planche graphique d’ambiance ou d’inspiration. |
| MVC | Model View Controller est une manière d’organiser la structure d’un projet de développement. |
| OS | Système d’exploitation |
| PHP | Langage de programmation principalement utilisé en web. |
| SGBD(R) | Système de gestion de bases de données (relationnelles). |
| Sprint | Cycle du projet au cours duquel vont être effectuées des tâches du même style. |
| URL | Uniform Resource Locator est une adresse web unique. |
| Use case | Description écrite de la réalisation de tâches utilisée dans le développement logiciel. |

## 

## Table des illustrations

[Image 1 Organisation des tâches 5](#_Toc136357553)

[Image 2 Méthode de gestion de projet en cascade 6](#_Toc136357554)

[Image 3 Planification initiale du projet de TPI 7](#_Toc136357555)

[Image 4 MCD de la base de données infofit 9](#_Toc136357556)

[Image 5 MLD de la base de données infofit 10](#_Toc136357557)

[Image 6 Scripte SQL de création de la base de données infofit 11](#_Toc136357558)

[Image 7 Moodboard de l'application 12](#_Toc136357559)

[Image 8 Page d'accueil de la maquette 13](#_Toc136357560)

[Image 9 Page de login de la maquette 13](#_Toc136357561)

[Image 10 Page de register de la maquette 14](#_Toc136357562)

[Image 11 Page profil utilisateur de la maquette 14](#_Toc136357563)

[Image 12 Page de calcul de l'IMC de la maquette 15](#_Toc136357564)

[Image 13 Page alimentation de la maquette 15](#_Toc136357565)

[Image 14 Page d'ajout d'aliment de la maquette 16](#_Toc136357566)

[Image 15 Capture du dossier de TPI 18](#_Toc136357567)

[Image 16 Dossier du projet InfoFit 19](#_Toc136357568)

[Image 17 Capture d'écran de la fonction store 21](#_Toc136357569)

[Image 18 Première partie capture de la fonction searchFoodstuff 22](#_Toc136357570)

[Image 19 Deuxième partie capture de la fonction searchFoodstuff 23](#_Toc136357571)

[Image 20 Capture de la fonction lineChart 24](#_Toc136357572)

[Image 21 Capture de la page d'accueil de InfoFit 25](#_Toc136357573)

[Image 22 Capture de la page de login de InfoFit 25](#_Toc136357574)

[Image 23 Capture de la page de register de InfoFit 26](#_Toc136357575)

[Image 24 Capture de la page de profil de InfoFit 27](#_Toc136357576)

[Image 25 Capture de la page d'IMC de InfoFit 27](#_Toc136357577)

[Image 26 Capture de la page d'alimentation de InfoFit 28](#_Toc136357578)

[Image 27 Capture de la page d'ajout d'aliment de InfoFit 28](#_Toc136357579)

[Image 28 Première page du cahier des charges 34](#_Toc136357580)

[Image 29 Deuxième page du cahier des charges 35](#_Toc136357581)

[Image 30 Troisième page du cahier des charges 36](#_Toc136357582)

[Image 31 Capture du fichier .env de l'application 46](#_Toc136357583)

## Manuel d'installation

Tout d’abord, ce manuel d’installation explique les étapes nécessaires au lancement du site InfoFit de manière locale et dans un système d’exploitation Windows 10.

Afin de pouvoir ouvrir l’application InfoFit en local, plusieurs installations sont nécessaires. La première installation à faire est la librairie php disponible sur le site <https://www.php.net/>. La deuxième installation à effectuer est Composer (<https://getcomposer.org/>), pour ce faire, ouvrez le fichier php.ini de la librairie php que vous venez d’installer et décommentez la ligne « **extension=php\_fileinfo** », puis enregistrez le fichier. Ensuite, saisissez la commande **composer** **install** dans une invite de commande.

Une étape importante pour que l’application fonctionne correctement est de créer la base de données localement. Vous pouvez utiliser le SGBDR de votre choix ; j’ai personnellement utilisé Mariadb. Veuillez créer une nouvelle base de données nommée « infofit » et un utilisateur propre à cette base et y possédant tout droits possibles. Exécutez ensuite le scripte de création de ma base de données disponible sur mon dépôt GitHub (<https://github.com/ThierryKoetschet/TPI_Thierry_Koetschet>) sous le nom de **script\_creation\_db.sql**. Enfin, modifiez le fichier **.env.example** du projet InfoFit pour qu’il corresponde au informations que vous avez saisi lors de la création de la base de données comme l’exemple ci-dessous et renommez-le en **.env**.

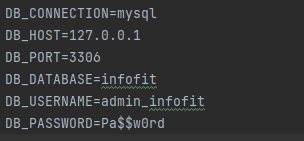


Image 31 Capture du fichier .env de l'application

Pour faciliter le lancement de l’application, je conseille l’installation d’un environnement de développement supportant le langage php. Par exemple, Visual Studio Code ou encore PhpStorm fonctionnent parfaitement et j’ai personnellement utilisé dans la réalisation de mon projet l’IDE PhpStorm. Ouvrez donc le projet InfoFit de mon dépôt GitHub dans l’IDE de votre choix ouvrez un terminal. Placez vous à la racine du projet avec un commande cd et saisissez ensuite la commande **php artisan serve** afin de lancer l’application de manière locale.