Documentation Technique

Travail Pratique Individuel

Une image contenant homme, Condition physique, muscle, Coude

Description générée automatiquement

Soares Rodrigues Flavio (I.DA-P4A)

CFPT INFORMATIQUE

17 mai 2023

Table des matières

[Introduction 5](#_Toc135168746)

[Rappel de l’énoncé 5](#_Toc135168747)

[Matériel et logiciels à disposition 5](#_Toc135168748)

[Description de l’application 6](#_Toc135168749)

[Livrables 6](#_Toc135168750)

[Acteurs 6](#_Toc135168751)

[Méthodologie 6](#_Toc135168752)

[Organisation et planification des tâches à réaliser 8](#_Toc135168753)

[Planning prévisionnel 8](#_Toc135168754)

[Planning réel 8](#_Toc135168755)

[Analyse de la planification 8](#_Toc135168756)

[Librairies et outils externes 9](#_Toc135168757)

[PHP 8.2 9](#_Toc135168758)

[Bootstrap 5.0.2 9](#_Toc135168759)

[Font Awesome 6.3.0 9](#_Toc135168760)

[MySQL Workbench 8.0.33 9](#_Toc135168761)

[PhpMyAdmin 5.1.1deb5ubuntu1 9](#_Toc135168762)

[GitHub 3.2.3 10](#_Toc135168763)

[Apache 2.4.52 10](#_Toc135168764)

[PhpMailer 6.8.0 10](#_Toc135168765)

[TinyMCE 6.4.0 10](#_Toc135168766)

[ChatGPT 3.5 10](#_Toc135168767)

[Analyse fonctionnelle 11](#_Toc135168768)

[Maquettes 11](#_Toc135168769)

[Visiteur 11](#_Toc135168770)

[Navbar visiteur 11](#_Toc135168771)

[Accueil 12](#_Toc135168772)

[Inscription 13](#_Toc135168773)

[Connexion 14](#_Toc135168774)

[Recherche lors d’un filtre 14](#_Toc135168775)

[Détail d’un article 15](#_Toc135168776)

[Utilisateur connecté 16](#_Toc135168777)

[Navbar utilisateur connecté 16](#_Toc135168778)

[Panier 17](#_Toc135168779)

[Validation commande 18](#_Toc135168780)

[Admin 19](#_Toc135168781)

[Navbar administrateur 19](#_Toc135168782)

[Créer un article 20](#_Toc135168783)

[Modifier un article 22](#_Toc135168784)

[Créer une catégorie 23](#_Toc135168785)

[Modifier une catégorie 24](#_Toc135168786)

[Détails d’un article (admin) 24](#_Toc135168787)

[Analyse organique 26](#_Toc135168788)

[Base de données 26](#_Toc135168789)

[Modèle conceptuel de données 26](#_Toc135168790)

[Colonnes 27](#_Toc135168791)

[Utilisateurs 27](#_Toc135168792)

[Article du panier 27](#_Toc135168793)

[Articles 28](#_Toc135168794)

[Catégories 28](#_Toc135168795)

[Villes 28](#_Toc135168796)

[Images 29](#_Toc135168797)

[Structure 29](#_Toc135168798)

[Classes utilisées 29](#_Toc135168799)

[Class Article 29](#_Toc135168800)

[Class User 30](#_Toc135168801)

[Class Category 30](#_Toc135168802)

[Class Image 30](#_Toc135168803)

[Class CartItem 31](#_Toc135168804)

[Class City 31](#_Toc135168805)

[Fonctions principales 31](#_Toc135168806)

[Fonctions Article 31](#_Toc135168807)

[Fonctions User 34](#_Toc135168808)

[Fonctions Category 35](#_Toc135168809)

[Fonctions Image 37](#_Toc135168810)

[Fonctions CartItem 38](#_Toc135168811)

[Fonctions City 40](#_Toc135168812)

[Arborescence de mon projet 41](#_Toc135168813)

[Mise en place de l’application 42](#_Toc135168814)

[Tests 42](#_Toc135168815)

[Environnement 43](#_Toc135168816)

[Scénarios des tests 43](#_Toc135168817)

[1.0 Accueil 43](#_Toc135168818)

[2.0 Inscription 43](#_Toc135168819)

[3.0 Connexion 44](#_Toc135168820)

[4.0 Filtres 44](#_Toc135168821)

[5.0 Détail d’un article 45](#_Toc135168822)

[6.0 Panier 46](#_Toc135168823)

[7.0 Création d’un article 47](#_Toc135168824)

[Évolution des tests 48](#_Toc135168825)

[Conclusion 49](#_Toc135168826)

[Difficultés rencontrées 49](#_Toc135168827)

[Améliorations possibles 49](#_Toc135168828)

[Bilan personnel 49](#_Toc135168829)

[Remerciements 50](#_Toc135168830)

[Glossaire 50](#_Toc135168831)

[TPI 50](#_Toc135168832)

[Librairie 50](#_Toc135168833)

[Carrousel 50](#_Toc135168834)

[Framework 50](#_Toc135168835)

[Script 50](#_Toc135168836)

[CSV 50](#_Toc135168837)

[Javascript 50](#_Toc135168838)

[AJAX 51](#_Toc135168839)

[JSON 51](#_Toc135168840)

[MVC 51](#_Toc135168841)

# Introduction

Ce projet a été réalisé dans le cadre de mon TPI (Travail Pratique Individuel) 2023. Je dois réaliser site WEB qui a pour but de valider mes compétences acquises durant ma formation au CFPT Informatique en tant qu’informaticien CFC en développement d’application.  
  
« GYM » est un site WEB permettant d’acheter du matériel de fitness/musculation. Les utilisateurs peuvent utiliser plusieurs fonctionnalités tel que : consulter les articles mis à la une, filtrer les produits selon différents critères, ajouter des articles dans un panier, modifier le panier (modification de la quantité, suppression d’article), acheter un produit.

# Rappel de l’énoncé

*Les informations suivantes sont tirées du cahier des charges.*

## Matériel et logiciels à disposition

• Ordinateur munis de deux écrans  
• Visual Studio Code + extensions : PHP Debug + PHP DocBlocker  
• Serveur Apache + module PHP + extension pdo\_mysql  
• Serveur SQL (MySQL server ou MariaDB server)  
• Client SQL (MySQL Workbench ou DBeaver)  
• Navigateurs : Edge, Chrome, Firefox.  
• Editeur de texte : Microsoft Word ou Open Office  
• Tableur : Microsoft Excel ou Open Office  
• Outil de dessin : Gimp ou Suite Adobe  
• Editeur javascript TinyMCE ou autre éditeur WIZIWIG  
• Composant de mailing PHPMailer

## Description de l’application

Le travail consiste à réaliser un site Internet de vente d’articles de fitness/musculation. Ce site Internet permettra aux :

* **Visiteurs,** utilisateurs non enregistrés, de : visualiser l’ensemble des articles, de rechercher des articles en fonction de critères.
* **Utilisateurs enregistrés** d’acheter un ou plusieurs articles.
* **Administrateurs** : d’ajouter, de modifier ou retirer des articles, ainsi que gérer les catégories d’articles.

Le site Internet doit être fonctionnel sous les navigateurs Microsoft Edge et Chrome intégrant le moteur du projet Chromium, ainsi que FireFox. Le navigateur Safari n’est pas pris en compte.

## Livrables

* Documentation du TPI (manuel technique), incluant code source
* Manuel utilisateur
* Résumé du rapport du TPI
* Journal de travail

## Acteurs

|  |  |  |
| --- | --- | --- |
| **Nom Prénom** | **Email** | **Status** |
| Flavio Soares Rodrigues | flavio.srsrd@eduge.ch | Élève |
| Dominique Aigroz | edu-aigrozd@eduge.ch | Maître d’apprentissage |
| Daniel Vanini | daniel.vanini@skyguide.ch | Expert |
| Jacques Ortola | jacques.ortola@skyguide.ch | Expert |

# Méthodologie

Pour la planification de mon projet de TPI, mon maître d’apprentissage m’a demandé d’utiliser la **méthode en cascade.**

Une image contenant texte, nature

Description générée automatiquement

1. **Exigences :** les exigences font l’objet d’une expression des besoins, cette étape a été réalisé lorsque mon maître d’apprentissage m’a demandé dans quel domaine je voulais réaliser mon TPI.
2. **Analyse :** les exigences sont analysées pour établir un cahier des charges fonctionnel, cette étape a été réalisé par mon maître d’apprentissage et validé par les experts.
3. **Conception :** le produit est conçu et spécifié de sorte à pouvoir être réalisé, j’ai réalisé cette étape lors de ma planification du projet et la création des différentes maquettes.
4. **Mise en œuvre :** le produit est réalisé sur la base des spécifications, j’ai réalisé cette étape lors du développement du site en suivant le planning réalisé au préalable.
5. **Validation :** le produit est testé et vérifié et sa conformité aux exigences est validée, cette étape sera réalisée une fois mon TPI rendu et corriger pour vérifier s’il est conforme aux exigences.
6. **Mise en service :** le produit est déployé, cette étape ne sera malheureusement pas réalisée dans le cadre de ce TPI car une fois que le projet a été présenté durant la défense oral, le projet prend fin.

# Organisation et planification des tâches à réaliser

## Planning prévisionnel

Une image contenant texte, capture d’écran, logiciel, Système d’exploitation

Description générée automatiquement

## Planning réel

## Analyse de la planification

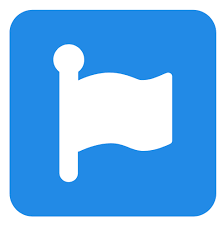
# Librairies et outils externes

## Php Icon #364275 - Free Icons LibraryPHP 8.2

Pour ce projet, j’ai utilisé le langage PHP pour les échanges avec la base de données (traitement et récupération de données). Il m’a également permis de créer des pages WEB dynamiques.

## Une image contenant logo, Graphique, symbole, Police Description générée automatiquementBootstrap 5.0.2

J'ai utilisé Bootstrap comme Framework pour styliser la majeure partie de mon application. En effet, il offre une facilité remarquable pour créer des pages responsive.



## Font Awesome 6.3.0

J’ai utilisé la librairie Font Awesome pour afficher des icônes sur mon site.

## MySQL Workbench | GPL, Oracle | Logiciel libre, LogicielMySQL Workbench 8.0.33

J’ai utilisé cet outils pour la création de mon model conceptuel de données et de ma base de données ainsi que pour importer le fichier CSV des villes.

## Une image contenant Voilier, transport, embarcation, bateau Description générée automatiquementPhpMyAdmin 5.1.1deb5ubuntu1

J’ai utilisé phpMyAdmin pour toutes les autres choses qui touches à la base de données (exportation, création de l’utilisateur de la base de données).

## GitHub Logo et symbole, sens, histoire, PNG, marqueGitHub 3.2.3

C’est une plateforme web qui permet de gérer les versions. J'ai utilisé cet outil tout au long du développement de mon projet.

Une image contenant stylos et plumes, outil d’écriture, conception

Description générée automatiquement

## Apache 2.4.52

J’ai utilisé apache2 comme serveur web local sur ma machine.

## Phpmailer png images | PNGWingPhpMailer 6.8.0

Phpmailer est une librairie écrite en php qui permet de facilité l’envois de mail. Je l’ai utilisé lors de mes envois de mail sur le site (inscription et validation de la commande).

## Full featured examples | Docs | TinyMCETinyMCE 6.4.0

TinyMCE est un éditeur de text HTML WYSIWYG (What You See Is What You Get). Je l’ai utilisé pour la description d’un article.

## ChatGPT 3.5

ChatGPT une intelligence artificiel spécialisé dans le dialogue. J’ai utilisé ChatGPT pour des tâches répétitif tel que :

* Copier/coller de fonction
* Création et correction de commentaires

Mais je l’ai également utilisé en tant qu’outil de recherche pour m’aider à résoudre des bugs en lui indiquant le message d’erreur. Car, GPT est plus efficace que la recherche sur des forums.

## Canvas

Outil de conception graphique en ligne. Je l’ai utilisé pour la création de mon logo

# Analyse fonctionnelle

## Maquettes

### Visiteur

En tant que visiteur, je peux accéder à la page d’accueil pour voir les produits à la une. Je peux également rechercher des produits grâce à différents filtres. Lors ce que je clique sur un article je peux voir les détails de celui-ci. Lors ce que je clique sur le bouton ajouter je suis redirigé vers une page d’inscription qui me permet de créer un compte sur le site.

#### Navbar visiteur

Barre de navigation disponible pour un visiteur sur le site. Le visiteur peut se connecter ou s’inscrire sur le site grâce à la partie supérieure de la navigation. Il peut également rechercher des articles sur le site grâce aux filtres dans la partie inferieur. Le filtrage fonctionne de manière indépendante. Il peut décider de rechercher sans filtre, tout comme avec un ou plusieurs voir la totalité des champs en même temps s’il le souhaite.  
Grâce à ça, il peut rechercher au sein du titre ou de la description de l’article avec la recherche par texte, il peut également rechercher par catégorie ou par le prix minimum ou maximum.



**A**

**C**

**B**

**G**

**E**

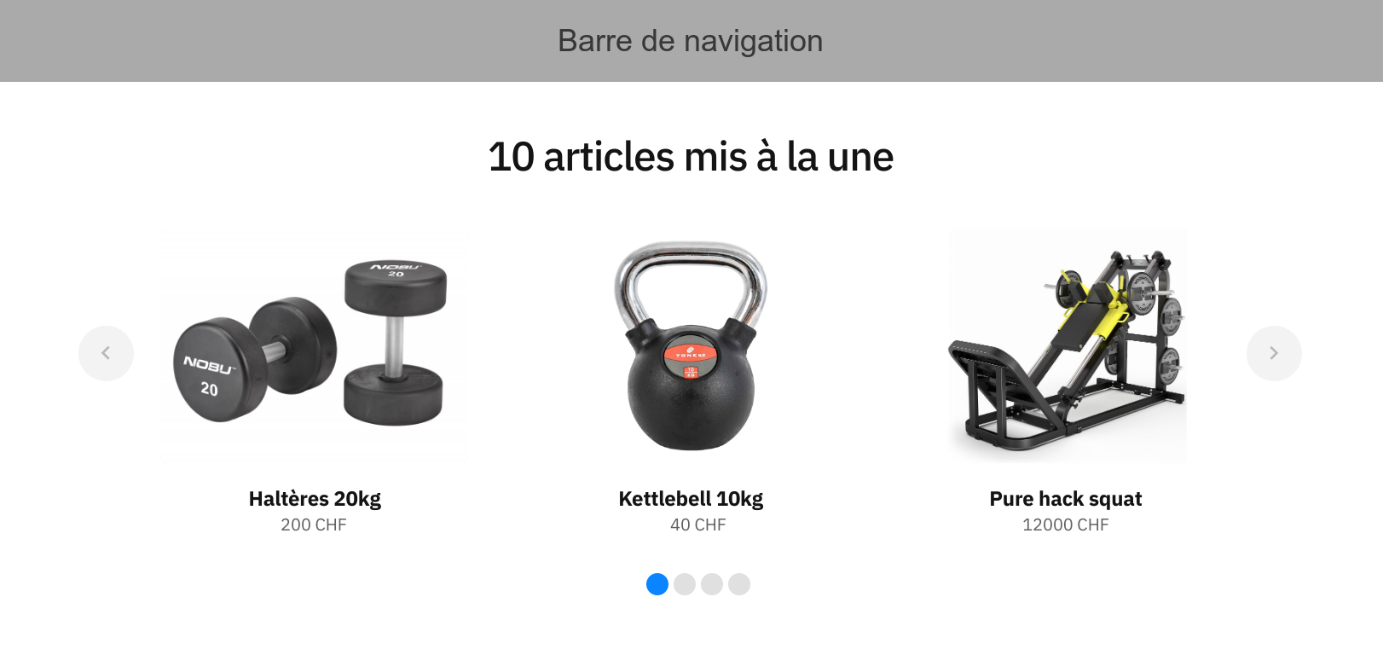
**F**

**D**

1. Logo du site qui redirige le visiteur vers la page d’accueil (*voir écran* [*Accueil*](#_Accueil))
2. Icone qui redirige le visiteur vers la page de connexion (*voir écran* [*Connexion*](#_Connexion))
3. Icone qui redirige le visiteur vers la page d’inscription (*voir écran* [*Inscription*](#_Inscription))
4. Filtre par saisie libre
5. Filtre par la catégorie d’article
6. Filtre par le prix minimum
7. Filtre par le prix maximum

#### Accueil

Page d’accueil du site. Elle permet d’afficher les 10 articles mis à la une les plus récents. En cliquant sur un article le visiteur est redirigé vers la page de détail de l’article.



**G**

**F**

**E**

**A**

**D**

**C**

**B**

1. Carrousel qui affiche les 10 articles mis à la une les plus récents
2. Photo principale de l’article mis à la une
3. Nom de l’article
4. Le prix de l’article en francs suisse
5. Bouton qui permet d’afficher les articles précédents du carrousel
6. Bouton qui permet d’afficher les articles suivant du carrousel
7. Indicateur de navigation du carrousel

#### Inscription

La page d’inscription permet à un nouvel utilisateur de s’enregistrer sur le site afin de pouvoir acheter des articles sur le site. Lors ce que le formulaire est valide et qu’il s’inscrit sur le site l’utilisateur reçoit un mail sur son adresse mail lui indiquant que son compte a été créer.

Une image contenant table

Description générée automatiquement

**I**

**H**

**J**

**G**

**F**

**E**

**A**

**B**

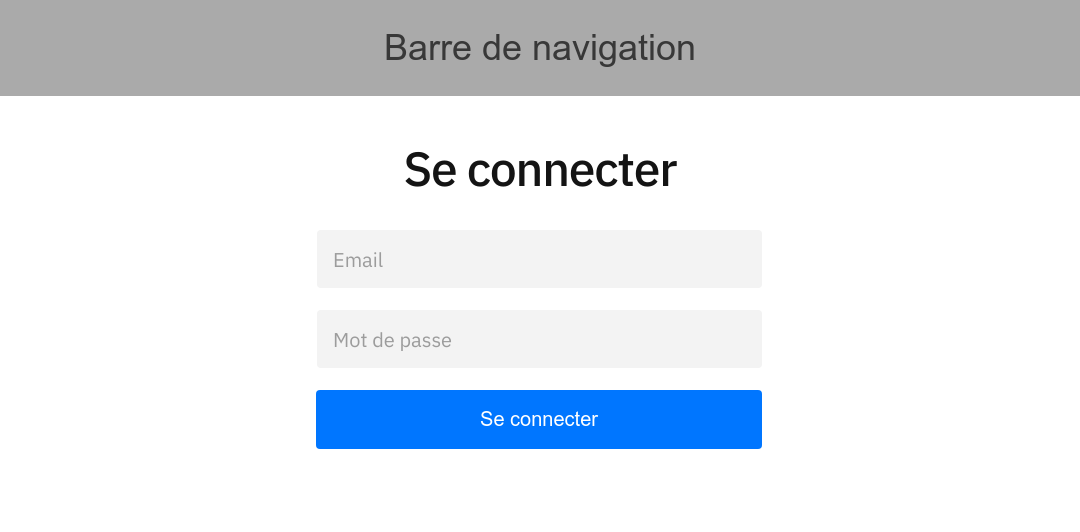
**C**

**D**

1. Email de l’utilisateur (doit être unique)
2. Nom de l’utilisateur
3. Prénom de l’utilisateur
4. Genre (homme, femme, autre)
5. Adresse 1 de l’utilisateur
6. Adresse 2 de l’utilisateur (optionnel)
7. Ville Suisse dans la quel l’utilisateur habite
8. Code postal de la ville (4 chiffres)
9. Mot de passe de l’utilisateur
10. Bouton de validation du formulaire pour inscrire le nouvel utilisateur

#### Connexion

Page de login. L’utilisateur peut se connecter au site s’il est inscrit. En insérant son email et mot de passe.



**C**

**B**

**A**

1. Email de l’utilisateur
2. Mot de passe de l’utilisateur
3. Bouton qui permet de valider le formulaire de connexion au site

#### Recherche lors d’un filtre

La page recherche lors d’un filtre permet d’afficher les articles correspondant au filtre utilisé (saisie unique, catégorie, prix min, prix max). Pour chaque article correspondant au filtre la page affiche l’image principale de l’article, le nom de l’article et le prix de l’article.

Une image contenant diagramme

Description générée automatiquement

**B**

**C**

**D**

**A**

A. Le type de filtre utilisé (Appareils dans cet exemple)  
B. La photo principale de l’article  
C. Le nom de l’article  
D. Le prix de l’article

#### Détail d’un article

Une image contenant texte, capture d’écran, appareil de gymnastique

Description générée automatiquementLa page détails d’un article affiche les informations de l’article tel que son nom, les différentes images, son prix ainsi que sa description. Et permet à l’utilisateur d’ajouter l’article dans le panier si celui-ci est connecter dans le cas contraire il est redirigé vers la page d’inscription.

**G**

**G**

**F**

**E**

**D**

**C**

**B**

**A**

1. Le nom de l’article
2. Toutes les photos de l’article
3. Le prix de l’article
4. La description de l’article
5. Bouton qui permet d’ajouter l’article au panier
6. Bouton qui permet d’afficher les articles précédents du carrousel
7. Bouton qui permet d’afficher les articles suivant du carrousel
8. Indicateur de navigation du carrousel

### Utilisateur connecté

En tant qu’utilisateur enregistré je peux me connecter au site grâce à mon email et un mot de passe. Je peux également ajouter des articles dans mon panier. J’ai la possibilité afficher mon panier pour voir les articles que j’ai ajouté dans celui-ci. Je peux également modifier la quantité ou supprimer des articles de mon panier. Je peux commander les articles de mon panier ce qui me redirige vers une page de validation de commande.

#### Navbar utilisateur connecté

Barre de navigation disponible pour un utilisateur connecté au site. L’utilisateur connecté peut accéder à son panier, il peut se déconnecter. Lui aussi a la possibilité de rechercher des articles avec les filtres.



**C**

**B**

**A**

**D**

**G**

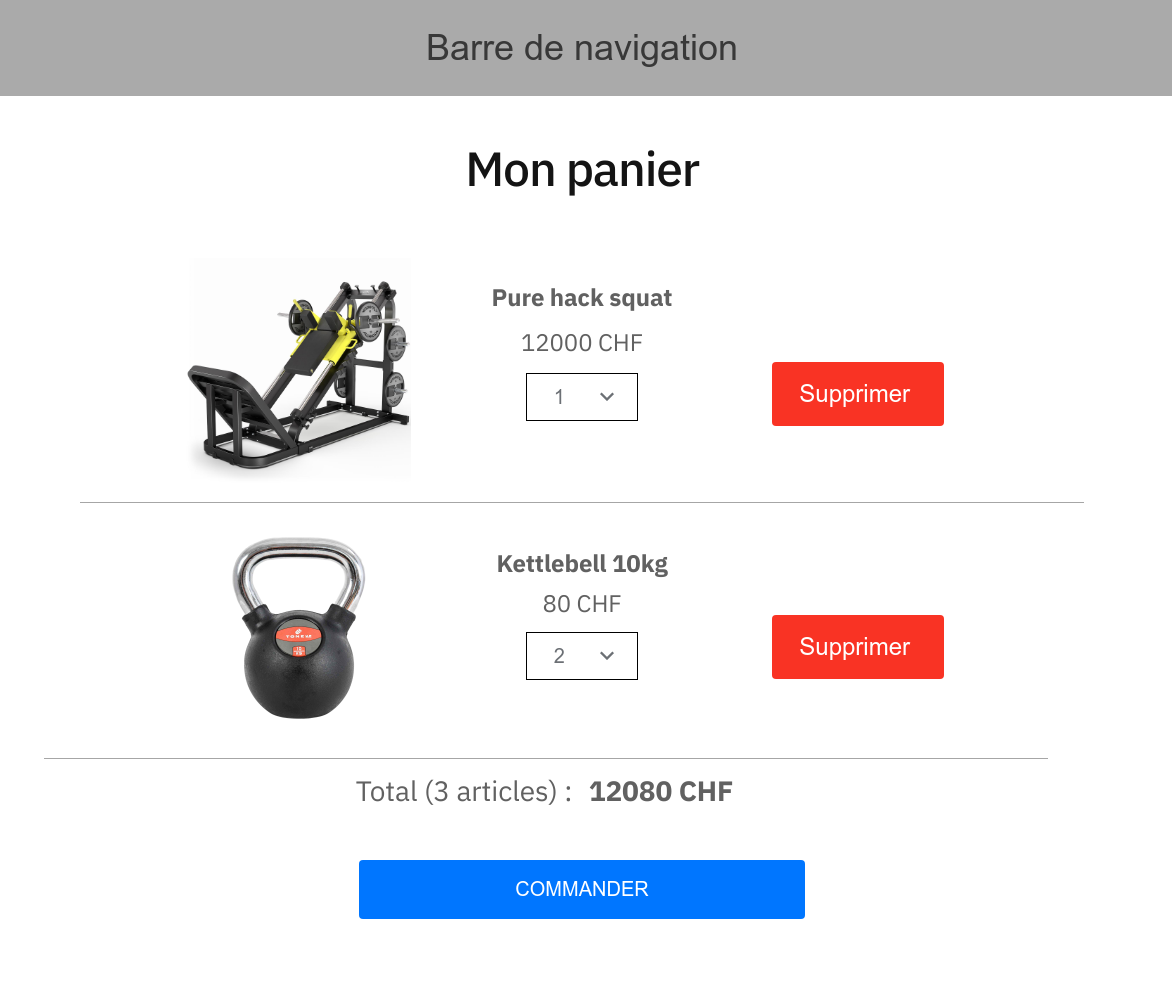
**E**

**F**

1. Logo du site qui redirige l’utilisateur connecté vers la page d’accueil (*voir écran* [*Accueil*](#_Accueil))
2. Icone qui redirige l’utilisateur connecté vers son panier (*voir écran* [*Panier*](#_Panier))
3. Icone qui déconnecte l’utilisateur
4. Filtre par saisie libre
5. Filtre par la catégorie d’article
6. Filtre par le prix minimum
7. Filtre par le prix maximum

#### Panier

Panier d’un utilisateur connecté. Dans le panier l’utilisateur peut voir la totalité des articles qu’il a ajoutés ainsi que sa quantité. Il peut modifier la quantité ce qui mettra à jour le total. Il peut supprimer un article du panier et peut commander ses articles avec le bouton commander qui va lui rediriger vers une page de confirmation.



**G**

**F**

**E**

**D**

**C**

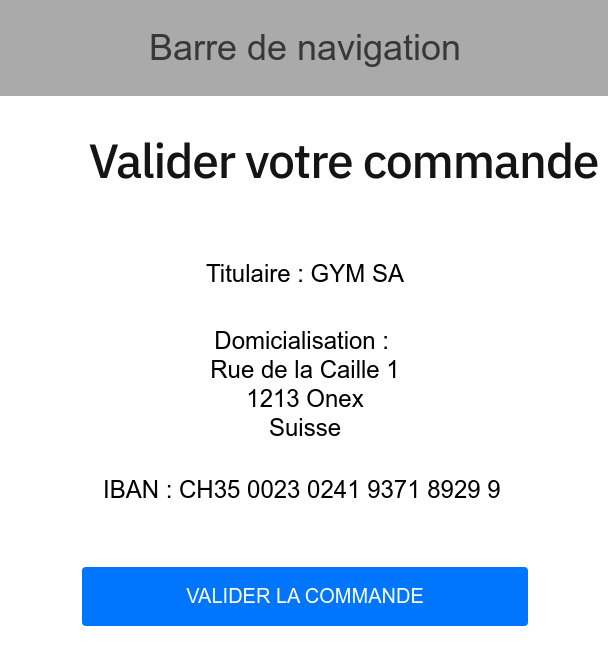
**B**

**A**

1. Image principale de l’article
2. Le nom de l’article
3. Le prix de l’article TTC multiplier par sa quantité dans le panier
4. La quantité de l’article dans le panier
5. Bouton qui permet de supprimer l’article du panier
6. Le prix total de tous les articles
7. Bouton qui redirige vers la page de validation de la commande (*voir écran* [*Validation commande*](#_Validation_commande))

#### Validation commande

Page validation de la commande. Elle affiche les données bancaires du site et permet de valider la commande grâce au bouton de validation. Une fois le bouton de validation cliqué l’utilisateur reçoit par mail la liste des articles qu’il a commandés ainsi que le prix et la quantité de chaque article et le total. Le panier est vidé une fois la commande valider et la quantité en stock est mise à jour.



**D**

**C**

**B**

**A**

1. Titulaire du compte bancaire
2. Adresse de la banque
3. IBAN du compte bancaire du site
4. Bouton qui envoie un mail qui indique que l’achat a été fait

### Admin

En tant qu’administrateur du site je peux ajouter des articles grâce à un lien dans la barre de navigation, j’ai également la possibilité de modifier un article déjà existant grâce à un bouton dans le détail de l’article. Je peux aussi ajouter une catégorie d’article à mon site grâce à un lien dans la barre de navigation et modifier une catégorie déjà existante.

#### Navbar administrateur

Barre de navigation pour le compte administrateur du site. L’administrateur peut naviguer vers les pages permettant :

* Ajouter un article au site
* Ajouter une catégorie
* Modifier une catégorie



**I**

**E**

**D**

**C**

**B**

**A**

**H**

**G**

**F**

1. Logo du site qui redirige l’admin vers la page d’accueil (*voir écran* [*Accueil*](#_Accueil))
2. Lien qui redirige l’admin vers la page ajouter un article (*voir écran* [*Créer un article*](#_Créer_un_article))
3. Lien qui redirige l’admin vers la page ajouter une catégorie (*voir écran* [*Créer une catégorie*](#_Créer_une_catégorie))
4. Lien qui redirige l’admin vers la page modifier une catégorie (*voir écran* [*Modifier catégorie*](#_Modifier_une_catégorie))
5. Icone qui déconnecte l’admin du site
6. Filtre par saisie libre
7. Filtre par la catégorie d’article
8. Filtre par le prix minimum
9. Filtre par le prix maximum

#### Créer un article

Page qui permet de créer un nouvel article. Pour crée un nouvel article l’administrateur du site doit saisir un nom d’article qui n’est pas déjà existant, écrire une description de l’article qui sera au format html, choisir au minimum une image qui sera l’image principale de l’article, définir la catégorie de l’article, sa quantité en stock, son prix et s’il est à la une ou non puis il valide la création grâce au bouton de validation.



**H**

**G**

**F**

**E**

**D**

**C**

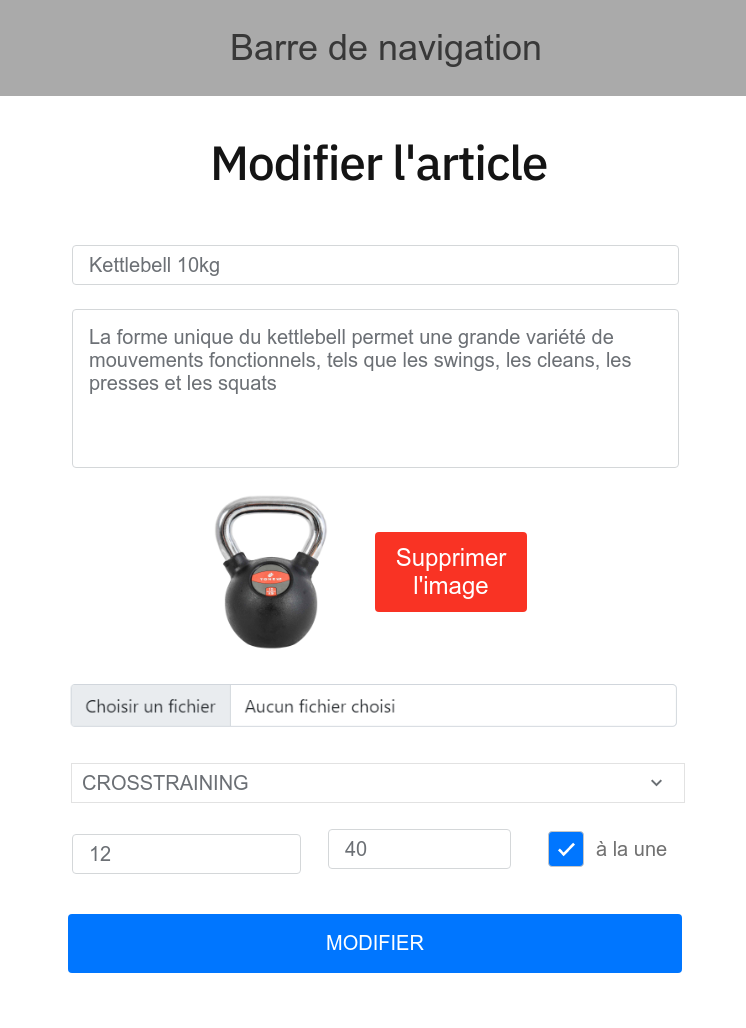
**B**

**A**

1. Nom de l’article (unique)
2. Description de l’article au format HTML
3. Un champ qui permet d’insérer au minimum une image pour l’image d’affiche de l’article
4. La catégorie de l’article
5. La quantité en stock
6. Le prix de l’article TTC
7. Check box qui permet de savoir si l’article est à la une ou non
8. Bouton de validation qui permet de créer l’article une fois tous les champs remplis et valider.

#### Modifier un article

Page qui permet de modifier un article. La page est similaire à la création mis à part l’ajout de l’affichage de chaque image correspondant à l’article avec la possibilité de pouvoir la supprimer et le bouton modifie qui met à jour l’article dans la base de données



**C**

**J**

**I**

**H**

**G**

**F**

**E**

**D**

**B**

**A**

1. Nom de l’article (unique)
2. Description de l’article au format HTML
3. Image de l’article
4. Bouton qui permet de supprimer l’image
5. Un champ qui permet d’insérer au minimum une image pour l’image d’affiche de l’article
6. La catégorie de l’article
7. La quantité en stock
8. Le prix de l’article TTC
9. Check box qui permet de savoir si l’article est à la une ou non
10. Bouton de validation qui permet de modifier l’article une fois tous les champs remplis et valider.

#### Créer une catégorie

Page qui permet d’ajouter une nouvelle catégorie d’article dans le site en insérant un nom qui n’existe pas. Le nom est automatiquement mis en majuscule pour respecter les 4 catégories qui m’ont été imposé.



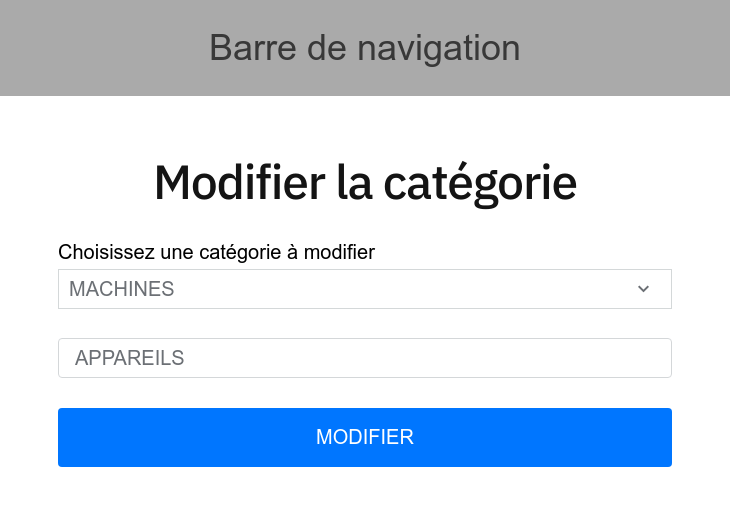
**A**

**B**

1. Nom de la catégorie (unique)
2. Bouton de validation qui permet de créer la catégorie une fois le champ valider

#### Modifier une catégorie

Page qui permet de modifier une catégorie. L’administrateur doit saisir un article à modifier dans la liste déroulante. Dans le champ en dessous l’administrateur du site doit mettre un nom unique puis il valide la modification qui va mettre à jour la catégorie avec son nouveau nom.



**C**

**B**

**A**

1. La liste de toute les catégorie (la catégorie sélectionnée sera celle qui sera modifié)
2. Le nouveau nom de la catégorie après validation du formulaire (met à jour le nom de la catégorie sélectionnée dans la liste déroulante au point A)
3. Bouton qui permet de valider la modification une fois les données saisies correctement

#### Détails d’un article (admin)

Page détails d’un article pour l’administrateur (similaire à celle d’un utilisateur). Cependant l’admin n’as pas accès au panier mais à la modification de l’article à la place.

Une image contenant texte, sport, appareil de gymnastique

Description générée automatiquement

**E**

**D**

**C**

**H**

**G**

**F**

**B**

**A**

1. Le nom de l’article
2. Toutes les photos de l’article
3. Le prix de l’article
4. La description de l’article
5. Bouton qui permet d’ajouter de modifier l’article (voir écran [*Modifier un article*](#_Modifier_un_article))
6. Bouton qui permet d’afficher les articles précédents du carrousel
7. Bouton qui permet d’afficher les articles suivant du carrousel
8. Indicateur de navigation du carrousel

# Analyse organique

## Base de données

GYM utilise une base de données MariaDB pour stocker les informations utiles pour l’application tel que : l’utilisateur, les articles, le panier, les images et autres informations utiles. J’ai utilisé la version 10.3.38 de MariaDB pour le développement de mon site.

### Modèle conceptuel de données

Voici le modèle conceptuel de données fait avec Workbench :

Une image contenant diagramme

Description générée automatiquement

Le script SQL permettant de créer toute la base de données ce situe dans le dossier :  
/docs/database/MCD sous le nom «MCD.mwb »

### Colonnes

#### Utilisateurs

Contient tous les utilisateurs du site y compris l’administrateur.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Colonne | Type | Clé | Description | Null | Défaut |
| ID | INT | PK | L’id unique de l’utilisateur enregistré | Non | / |
| NAME | VARCHAR(45) |  | Nom de l’utilisateur | Non | / |
| SURNAME | VARCHAR(45) |  | Prénom de l’utilisateur | Non | / |
| EMAIL | VARCHAR(100) |  | L’adresse mail de l’utilisateur | Non | / |
| GENDER | VARCHAR(20) |  | Le genre civil de l’utilisateur | Non | / |
| ADDRESS1 | VARCHAR(255) |  | Adresse de l’utilisateur | Non | / |
| ADDRESS2 | VARCHAR(255) |  | Adresse optionnelle | Oui | / |
| ZIP\_CODE | VARCHAR(4) |  | Code postal Suisse | Non | / |
| PASSWORD | VARCHAR(255) |  | Mot de passe haché de l’utilisateur | Non | / |
| IS\_ADMIN | TINYINT |  | Permet de définir si l’utilisateur est un admin du site | Non | 0 |
| CITIES\_ID | INT | FK | L’id unique des villes | Non | / |

#### Article du panier

Contient les articles du panier.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Colonne | Type | Clé | Description | Null | Défaut |
| USER\_ID | INT | PK | L’id unique de l’utilisateur | Non | / |
| ARTICLES\_ID | INT | PK | L’id unique de l’article | Non | / |
| QUANTITY | INT |  | Représente la quantité d’articles dans le panier (pour un même article) | Non | / |

#### Articles

Contient tous les articles du site

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Colonne | Type | Clé | Description | Null | Défaut |
| ID | INT | PK | L’id unique de l’article | Non | / |
| NAME | VARCHAR(200) |  | Nom de l’article qui est également unique | Non | / |
| DESCRIPTION | LONGTEXT |  | Description de l’article au format HTML | Non | / |
| PRICE | DOUBLE |  | Le prix de l’article TTC avec 2 nombres après la virgule | Non | / |
| STOCK | INT |  | La quantité d’articles en stock | Non | / |
| FEATURED | TINYINT |  | Permet de savoir si l’article est à la une ou non | Non | / |
| CATEGORIES\_ID | INT | FK | L’identifiant unique de la catégorie | Non | / |
| CREATION\_DATE | TIMESTAMP |  | Date de création de l’article | Non | CURRENT\_TIMESTAMP |
| UPDATE\_DATE | TIMESTAMP |  | Date de modification de l’article | Oui | / |

#### Catégories

Contient toutes les catégories d’article du site

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Colonne | Type | Clé | Description | Null | Défaut |
| ID | INT | PK | L’id unique de la catégorie | Non | / |
| NAME | VARCHAR(45) |  | Nom de la catégorie | Non | / |

#### Villes

Contient toutes les villes suisses

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Colonne | Type | Clé | Description | Null | Défaut |
| ID | INT | PK | L’id unique de la ville | Non | / |
| NAME | VARCHAR(45) |  | Nom de la ville suisse | Non | / |

#### Images

Contient toutes les images liées aux articles

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Colonne | Type | Clé | Description | Null | Défaut |
| ID | INT | PK | L’id unique de l’image | Non | / |
| CONTENT | LONGTEXT |  | Le contenu de l’image | Non | / |
| NAME | VARCHAR(255) |  | Le nom de l’image (unique) | Non | / |
| TYPE | VARCHAR(255) |  | Le type de l’image (PNG, JPG etc) | Non | / |
| MAIN\_IMAGE | TINYINT |  | Permet d’identifier si l’image est celle de présentation ou non | Non | / |
| ARTICLES\_ID | INT | FK | L’id unique de l’article | Non | / |

## Structure

### Classes utilisées (conteneur)

#### Class Article

Cette classe contient les informations pour un article

|  |  |  |
| --- | --- | --- |
| Variables | Type | Description |
| id | Int | L'identifiant unique de l'article |
| name | String | Le nom de l’article |
| description | String | La description de l'article |
| price | Double | Le prix de l'article |
| stock | Int | La quantité en stock de l'article |
| featured | Bool | Indique si l'article est mis en avant |
| creationDate | String | La date de création de l'article au format YYYY-MM-DD |
| updateDate | String | La date de dernière mise à jour de l'article au format YYYY-MM-DD |
| categoryId | Int | L'identifiant de la catégorie à laquelle l'article appartient |

#### Class User

Cette classe contient les informations pour un utilisateur

|  |  |  |
| --- | --- | --- |
| Variables | Type | Description |
| id | Int | L'identifiant unique de l'utilisateur |
| name | String | Le nom d'utilisateur |
| surname | String | Le prénom d'utilisateur |
| email | String | L'email de l'utilisateur (unique) |
| password | String | Le mot de passe haché de l'utilisateur |
| gender | String | Le genre de l'utilisateur |
| address1 | String | La première adresse de l'utilisateur |
| address2 | String | La deuxième adresse de l'utilisateur (optionnel) |
| city | String | La ville de l'utilisateur |
| zipCode | String | Le code postal de l'utilisateur |
| isAdmin | Bool | Indique si l'utilisateur est administrateur ou non |

#### Class Category

Cette classe contient les informations pour une catégorie

|  |  |  |
| --- | --- | --- |
| Variables | Type | Description |
| id | Int | L'identifiant unique de la catégorie |
| name | String | Le nom de la catégorie |

#### Class Image

Cette classe contient les informations pour une image

|  |  |  |
| --- | --- | --- |
| Variables | Type | Description |
| id | Int | L'identifiant unique de l'image |
| content | String | Le contenu de l'image (en base64) |
| name | String | Le nom de l'image (unique) |
| type | String | Le type de l'image (par exemple, "jpeg" ou "png") |
| mainImage | Bool | Indique si l'image est principale pour un article |
| articleId | Int | L'identifiant de l'article associé à l'image |

#### Class CartItem

Cette classe contient les informations pour un article du panier

|  |  |  |
| --- | --- | --- |
| Variables | Type | Description |
| usersId | Int | L'identifiant unique de l'utilisateur |
| articlesId | Int | L'identifiant unique de l'article |
| quantity | Int | La quantité d'articles |

#### Class City

Cette classe contient les informations pour une ville

|  |  |  |
| --- | --- | --- |
| Variables | Type | Description |
| id | Int | L'identifiant unique de la ville |
| name | String | Le nom de la ville |

### Fonctions principales

Les fonctions principales sont celles qui manipulant les objets conteneurs ainsi que les données de la base de données (requêtes SQL).

#### Fonctions Article

##### AddArticle

Ajoute un nouvel article dans la base de données.

AddArticle($name, $description, $price, $stock, $featured, $categoryId):bool

|  |  |  |  |
| --- | --- | --- | --- |
| Paramètres | Type | Description | Valeur retournée |
| name | String | Le nom de l'article | Retourne true si l'opération a réussi, false sinon |
| description | String | La description de l'article |
| price | Double | Le prix de l'article |
| stock | Int | Le stock disponible pour l'article |
| featured | Bool | Indique si l'article est à la une ou non |
| categoryId | Int | L'ID de la catégorie à laquelle l'article appartient |

##### GetArticle

Récupère un article grâce à son nom donné en paramètre.

GetArticle($name):mixed

|  |  |  |  |
| --- | --- | --- | --- |
| Paramètres | Type | Description | Valeur retournée |
| name | String | Le nom de l'article | Retourne l'article récupéré sous forme d'objet Article, ou false si une erreur est survenue |

##### GetArticleById

Récupère un article grâce à son ID donné en paramètre.

GetArticleById($id)mixed

|  |  |  |  |
| --- | --- | --- | --- |
| Paramètres | Type | Description | Valeur retournée |
| id | Int | L'identifiant unique de l'article | Retourne l'article récupéré sous forme d'objet Article, ou false si une erreur est survenue |

##### GetFilteredArticles

Récupère les articles en fonction des filtres spécifiés.

GetFilteredArticles($text, $categoryId, $minPrice, $maxPrice)

|  |  |  |  |
| --- | --- | --- | --- |
| Paramètres | Type | Description | Valeur retournée |
| text | String | Null | Chaîne de caractères à rechercher dans les noms et descriptions des articles. Null si aucun filtre n'est appliqué | Retourne un tableau d'objets Article filtrés ou false en cas d'erreur |
| categoryId | Int | Null | Identifiant de la catégorie à laquelle appartiennent les articles. Null si aucun filtre n'est appliqué |
| minPrice | Float | Null | Prix minimum des articles. Null si aucun filtre n'est appliqué |
| maxPrice | Float | Null | Prix maximum des articles. Null si aucun filtre n'est appliqué |

##### GetFeaturedArticles

Cette fonction récupère les 10 articles mis à la une les plus récent.

GetFeaturedArticles():mixed

|  |  |  |  |
| --- | --- | --- | --- |
| Paramètres | Type | Description | Valeur retournée |
| / | / | / | Retourne un tableau d'objets Article ou false si une erreur est survenue. |

##### ArticleExists

Vérifie si un article avec un nom donné existe dans la table `ARTICLES`.

ArticleExists($name):bool

|  |  |  |  |
| --- | --- | --- | --- |
| Paramètres | Type | Description | Valeur retournée |
| name | string | Le nom de l'article à rechercher | Retourne true si l'article existe, False sinon |

##### DecreaseStock

Diminue le stock d'un article.

DecreaseStock($id, $stock):bool

|  |  |  |  |
| --- | --- | --- | --- |
| Paramètres | Type | Description | Valeur retournée |
| id | Int | Identifiant de l'article | Retourne true si la mise à jour a été effectuée avec succès, sinon false |
| stock | Int | Quantité à déduire du stock |

##### UpdateArticle

Met l’article à jour avec les nouvelles valeurs.

UpdateArticle():bool

|  |  |  |  |
| --- | --- | --- | --- |
| Paramètres | Type | Description | Valeur retournée |
|  |  |  |  |

#### Fonctions User

##### RegisterUser

Insère un nouvel utilisateur dans la table USERS de la base de données.

RegisterUser($name, $surname, $email, $password, $gender, $address1, $address2, $city, $zipCode):bool

|  |  |  |  |
| --- | --- | --- | --- |
| Paramètres | Type | Description | Valeur retournée |
| name | String | Nom de l'utilisateur | Retourne TRUE si l'inscription réussit, sinon FALSE |
| surname | String | Prénom de l'utilisateur |
| email | String | Adresse e-mail de l'utilisateur |
| password | String | Mot de passe de l'utilisateur |
| gender | String | Genre de l'utilisateur |
| address1 | String | Adresse de l'utilisateur 1 |
| Address2 | String | Adresse de l'utilisateur 2 (optionnel) |
| cityId | Int | ID de la ville de l'utilisateur |
| zipCode | String | Code postal de l'utilisateur |

##### GetUser

Récupère un utilisateur grâce à son email donné en paramètre.

GetUser($email):mixed

|  |  |  |  |
| --- | --- | --- | --- |
| Paramètres | Type | Description | Valeur retournée |
| email | String | L'email de l'utilisateur à récupérer | Retourne l'utilisateur récupéré sous forme d'objet User, ou false si une erreur est survenue |

##### GetUserById

Récupère un utilisateur à partir de son identifiant donné en paramètre.

GetUserById($id):mixed

|  |  |  |  |
| --- | --- | --- | --- |
| Paramètres | Type | Description | Valeur retournée |
| id | Int | L'identifiant de l'utilisateur à récupérer | Retourne l'utilisateur récupéré sous forme d'objet User, ou false si une erreur est survenue |

##### EmailExists

Vérifie si l’email est déjà présent dans la base de données.

EmailExists($email):bool

|  |  |  |  |
| --- | --- | --- | --- |
| Paramètres | Type | Description | Valeur retournée |
| email | String | L'adresse email à rechercher | Retourne True si l'email existe, False sinon |

##### LoginUser

Vérifie les informations de connexion d'un utilisateur. Si les informations sont valides, enregistre l'utilisateur dans une session.

LoginUser($email, $password):bool

|  |  |  |  |
| --- | --- | --- | --- |
| Paramètres | Type | Description | Valeur retournée |
| email | String | L'adresse email de l'utilisateur | Retourne TRUE si l'utilisateur est connecté avec succès, sinon retourne FALSE |
| password | String | Le mot de passe de l'utilisateur |

#### Fonctions Category

##### GetCategories

Récupère la liste des catégories sous forme d'objet Category.

GetCategories():mixed

|  |  |  |  |
| --- | --- | --- | --- |
| Paramètres | Type | Description | Valeur retournée |
| / | / | / | Retourne le tableau d'objets Category représentant les catégories, ou false en cas d'erreur |

##### GetCategory

Récupère le nom de la catégorie correspondant à l'ID spécifié.

GetCategory($id):mixed

|  |  |  |  |
| --- | --- | --- | --- |
| Paramètres | Type | Description | Valeur retournée |
| id | Int | L'ID de la catégorie à rechercher | Retourne l'objet Category correspondant à l'ID spécifié, ou false en cas d'erreur |

##### UpdateCategory

Modifie le nom d'une catégorie dans la table `CATEGORIES`.

UpdateCategory($id, $name):bool

|  |  |  |  |
| --- | --- | --- | --- |
| Paramètres | Type | Description | Valeur retournée |
| Id | Int | L'adresse email de l'utilisateur | Retourne True si la mise à jour a été effectuée avec succès, False sinon |
| name | String | Le mot de passe de l'utilisateur |

##### CategoryExists

Vérifie si une catégorie avec un nom donné existe dans la table `CATEGORIES`.

CategoryExists($name):bool

|  |  |  |  |
| --- | --- | --- | --- |
| Paramètres | Type | Description | Valeur retournée |
| name | String | Le nom de la catégorie à rechercher | Retourne True si la catégorie existe, False sinon |

##### AddCategory

Crée une nouvelle catégorie dans la table `CATEGORIES`.

AddCategory($name):bool

|  |  |  |  |
| --- | --- | --- | --- |
| Paramètres | Type | Description | Valeur retournée |
| name | String | Le nom de la nouvelle catégorie | Retourne True si la création a été effectuée avec succès, False sinon |

#### Fonctions Image

##### AddEnc64Image

Insère une image encodée en base64 dans la base de données.

AddEnc64Image($content, $fileName, $fileType, $mainImage, $articlesId):bool

|  |  |  |  |
| --- | --- | --- | --- |
| Paramètres | Type | Description | Valeur retournée |
| content | String | Contenu de l'image encodé en base64 | Retourne true si l'insertion a réussi, sinon retourne false |
| fileName | String | Nom de fichier de l'image |
| fileType | String | Type de fichier de l'image |
| mainImage | Bool | Indique si l'image est une image principale ou non |
| articleId | Int | ID de l'article auquel l'image est associée |

##### GetImages

Récupère toutes les images associées à un article donné.

GetImages($articlesId):mixed

|  |  |  |  |
| --- | --- | --- | --- |
| Paramètres | Type | Description | Valeur retournée |
| articlesId | Int | L'identifiant de l'article | Retourne Un tableau d'objets Image représentant les images associées à l'article, ou false en cas d'erreur |

##### GetMainImage

Récupère l'image principale de l'article (l'image qui est mise en avant).

GetMainImage($articlesId):mixed

|  |  |  |  |
| --- | --- | --- | --- |
| Paramètres | Type | Description | Valeur retournée |
| articlesId | Int | L'identifiant de l'article | Retourne une Image représentant l' image principale associée à l'article, ou false en cas d'erreur |

##### DeleteImage

Supprime une image de la base de données.

DeleteImage($imageId):bool

|  |  |  |  |
| --- | --- | --- | --- |
| Paramètres | Type | Description | Valeur retournée |
| imageId | Int | L'identifiant de l'image à supprimer | Retourne true si la suppression a réussi, false sinon |

#### Fonctions CartItem

##### AddCartItem

Ajoute un article au panier d'un utilisateur ou augmente sa quantité de 1 s'il existe déjà.

AddCartItem($userId, $articleId):bool

|  |  |  |  |
| --- | --- | --- | --- |
| Paramètres | Type | Description | Valeur retournée |
| userId | Int | L'ID de l'utilisateur | Retourne true si l'article a été ajouté ou mis à jour avec succès, sinon false |
| articleId | Int | L'ID de l'article à supprimer |

##### GetCartItems

Récupère les articles présents dans le panier d'un utilisateur.

GetCartItems($userId):mixed

|  |  |  |  |
| --- | --- | --- | --- |
| Paramètres | Type | Description | Valeur retournée |
| userId | Int | L'ID de l'utilisateur | Retourne le tableau d'objets CartItem représentant les articles du panier, ou false en cas d'erreur |

##### UpdateQuantity

Met à jour la quantité d'un article dans le panier d'un utilisateur.

UpdateQuantity($userId, $articleId, $quantity):bool

|  |  |  |  |
| --- | --- | --- | --- |
| Paramètres | Type | Description | Valeur retournée |
| userId | Int | L'ID de l'utilisateur | Retourne true si la mise à jour a été effectuée avec succès, sinon false |
| articleId | Int | L'ID de l'article |
| quantity | Int | La nouvelle quantité de l'article |

##### DeleteCartItem

Supprime un article du panier en fonction de son ID.

DeleteCartItem($userId, $articleId):bool

|  |  |  |  |
| --- | --- | --- | --- |
| Paramètres | Type | Description | Valeur retournée |
| userId | Int | L'ID de l'utilisateur | Retourne true si la suppression a été effectuée avec succès, sinon false |
| articleId | Int | L'ID de l'article à supprimer |

##### CartItemExists

Vérifie si un article est déjà présent dans le panier d'un utilisateur

CartItemExists($userId, $articleId):bool

|  |  |  |  |
| --- | --- | --- | --- |
| Paramètres | Type | Description | Valeur retournée |
| userId | Int | L'ID de l'utilisateur | Retourne True si l'article est présent dans le panier, sinon False |
| articleId | Int | L'ID de l'article |

#### Fonctions City

##### GetCities

Récupère la liste des villes sous forme d'objet City

GetCities():mixed

|  |  |  |  |
| --- | --- | --- | --- |
| Paramètres | Type | Description | Valeur retournée |
| / | / | / | Retourne le tableau d'objets City représentant les villes, ou false en cas d'erreur |

##### GetCity

Récupère une ville à partir de son identifiant unique dans la base de données et retourne un objet City correspondant.

GetCity($id):mixed

|  |  |  |  |
| --- | --- | --- | --- |
| Paramètres | Type | Description | Valeur retournée |
| id | Int | L'identifiant unique de la ville à récupérer | Retourne un objet City correspondant à la ville demandée ou false si une erreur est survenue |

### Arborescence de mon projet

**/src**  
Racine de mon site web, contient tous les dossier et fichiers du site.

**/src/assets**  
Répertorie les différents sous-dossiers utilisés dans le projet (css, js, images, librairies).

**/src/assets/css**Répertorie les différents fichiers de style.

**/src/assets/images**Répertorie les différentes images utilisées dans le projet (logo, images de test).

**/src/assets/js**Répertorie les différents scripts en javascript.

**/src/assets/librairies**Répertorie les différentes librairies utilisées dans le projet (bootstrap, fontawesome, tinymce, phpmailer).

**/src/config**Répertorie les différents fichiers de configuration du site.

**/src/containers**Répertorie les différentes classes qui interagissent avec la base de données (CRUD).

**/src/db**Répertorie la classe qui permet de se connecter à la base de données.

**/src/includes**Répertorie les différents fichiers qui sont inclue dans d’autres pages du site (pas toutes les pages).

**/src/model**Répertorie les différents fichiers utilisés pour faire des requêtes dans la base de données.

**/src/session**Répertorie les différents fichiers servant pour la création d’une session.

**/src/SessionData**Dossier qui stock un fichier de session lors de la création d’une session.

**/src/tests**Répertorie les différents tests unitaires fait.

**/src/tools**Répertorie différents fichiers qui contient des fonctions d’affichage.

**/docs**Répertorie les différents documents liés à la documentation et à la planification (documentation technique, documentation utilisateur, journal de bord, planification, résumé TPI, maquettes, scripts de la base de données).

**/src/maquettes**Répertorie toutes les maquettes au format png.

**/src/database**Répertorie les scripts liées à la base de données.

**/src/CSV**Contient le fichier CSV utilisé pour récupérer les villes suisses

**/src/MCD**Répertorie les fichiers liés au Model Conceptuel de Données (MCD) de ma base de données.

### Mise en place de l’application

Lancer le script sql «GYM.sql» pour importer les données.

# Tests

Les différents tests effectués me permettent de garantir que toutes les fonctionnalités majeures du site soient présentes pour assurer la qualité du site. Je vais tester toutes les demandes du client en m’aidant des critères A15 à A20 du cahier des charges.  
Les tests que j’ai effectués sont de type fonctionnel, car ce sont les différentes fonctionnalités qui sont tester et qui vont me permettre de valider ou non son fonctionnement.

## Environnement

Les différents tests ont été reproduit grâce aux différents outils ci-dessous :

* Windows 10 Éducation
* Opera GX (noyau : 98.0.4759.22)
* Google Chrome 112.0.5615.138
* Ubuntu 20.04.01
* Apache 2
* MySQL Workbench Community 8.0
* MariaDB 10.3.38
* PHP 8.2

## Scénarios des tests

### 1.0 Accueil

|  |  |
| --- | --- |
| Nom | 1.1 Filtrés les articles avec un prix maximum |
| Situation | En tant qu’utilisateur anonyme ou connecté je souhaite pouvoir voir les 10 articles mis à la une les plus récent. |
| Résultat obtenu | les 10 articles à la une les plus récents sont affiché (Image principale, nom et prix) |
| Status | ✔OK |

### 2.0 Inscription

|  |  |
| --- | --- |
| Nom | 2.1 Création d’un nouvel utilisateur connecté |
| Situation | En tant que visiteur je veux pouvoir m’inscrire sur le site pour faire des achats. |
| Résultat obtenu | L’utilisateur est inscrit sur le site et peut désormais se connecter |
| Status | ✔OK |

|  |  |
| --- | --- |
| Nom | 2.2 Email de confirmation |
| Situation | Lors ce que je valide mon inscription et que les données saisies sont correctes j’aimerais recevoir par email une confirmation. |
| Résultat obtenu | Un mail de confirmation est envoyer |
| Status | ✔OK |

### 3.0 Connexion

|  |  |
| --- | --- |
| Nom | 3.1 Connexion au site |
| Situation | En tant qu’utilisateur enregistré j’aimerais pouvoir me connecter au site en saisissant mon email et mon mot de passe. |
| Résultat obtenu | L’utilisateur se connecte au site lors ce qu’il rentre son email et son mot de passe |
| Status | ✔OK |

### 4.0 Filtres

|  |  |
| --- | --- |
| Nom | 4.1 Filtrés les articles par saisie libre |
| Situation | En tant qu’utilisateur anonyme ou connecté je souhaite pouvoir rechercher grâce à un champs qui me permet de saisir du texte. |
| Résultat obtenu | Les articles qui correspondent à la saisie de text libre sont affiché |
| Status | ✔OK |

|  |  |
| --- | --- |
| Nom | 4.2 Filtrés les articles par catégorie |
| Situation | En tant qu’utilisateur anonyme ou connecté je souhaite pouvoir rechercher des articles selon sa catégorie via un menu déroulant. |
| Résultat obtenu | Affiche une liste déroulante avec toutes les catégories et lors ce qu’ une catégorie est sélectionné, les articles correspondant sont affiché. |
| Status | ✔OK |

|  |  |
| --- | --- |
| Nom | 4.3 Filtrés les articles avec un prix minimum |
| Situation | En tant qu’utilisateur anonyme ou connecté je souhaite pouvoir rechercher des articles selon son prix minimum en saisissant un nombre dans un champs. |
| Résultat obtenu | Les articles dont la valeur est supérieur ou égale à la saisie du prix minimum sont affichés |
| Status | ✔OK |

|  |  |
| --- | --- |
| Nom | 4.4 Filtrés les articles avec un prix maximum |
| Situation | En tant qu’utilisateur anonyme ou connecté je souhaite pouvoir rechercher des articles selon son prix maximum en saisissant un nombre dans un champs. |
| Résultat obtenu | Les articles dont la valeur est inférieur ou égale à la saisie du prix maximum sont affichés |
| Status | ✔OK |

|  |  |
| --- | --- |
| Nom | 4.5 Filtrés les articles avec plusieurs filtres |
| Situation | En tant qu’utilisateur anonyme ou connecté je souhaite pouvoir rechercher des articles selon plusieurs filtres simultané. |
| Résultat obtenu | Les articles correspondant à l’ensemble des filtres sont affichées |
| Status | ✔OK |

### 5.0 Détail d’un article

|  |  |
| --- | --- |
| Nom | 5.1 Affichage du détail d’un article |
| Situation | En tant qu’utilisateur je voudrais consulter les détails d’un article lors ce que je clique dessus. Je voudrais connaître le nom de l’article, sa description, son prix et son status pour savoir s’il est en stock. Ainsi que toutes les images de l’article. |
| Résultat obtenu | Les données concernant l’article sont affichées |
| Status | ✔OK |

|  |  |
| --- | --- |
| Nom | 5.2 Ajouter un article dans le panier |
| Situation | En tant qu’utilisateur connecté je voudrais avoir la possibilité d’ajouter l’article à mon panier si celui-ci est disponible. |
| Résultat obtenu | Ajoute l’article au panier lors ce que l’utilisateur connecté appuie sur ajouter au panier |
| Status | ✔OK |

### 6.0 Panier

|  |  |
| --- | --- |
| Nom | 6.1 Afficher le panier |
| Situation | En tant qu’utilisateur connecté j’aimerais voir les différents articles que j’ai ajouté au panier. |
| Résultat obtenu | Les articles du panier sont affichés |
| Status | ✔OK |

|  |  |
| --- | --- |
| Nom | 6.2 Modifier la quantité d’un article |
| Situation | En tant qu’utilisateur connecté j’aimerais pouvoir modifier la quantité d’un article (diminuer, augmenter). |
| Résultat obtenu | La quantité est mise à jour lors ce qu’on la change |
| Status | ✔OK |

|  |  |
| --- | --- |
| Nom | 6.3 Afficher le prix par article en fonction de la quantité |
| Situation | En tant qu’utilisateur connecté j’aimerais voir le prix par article en fonction de la quantité. |
| Résultat obtenu | Le prix par article est affiché en fonction de la quantité |
| Status | ✔OK |

|  |  |
| --- | --- |
| Nom | 6.4 Suppression d’un article du panier |
| Situation | En tant qu’utilisateur connecté j’aimerais pouvoir supprimer un article que je ne souhaite plus acheter. |
| Résultat obtenu | Lors ce que l’utilisateur connecté appuie sur le bouton supprimer l’article est supprimé du panier |
| Status | ✔OK |

|  |  |
| --- | --- |
| Nom | 6.5 Affichage du prix de l’ensemble du panier |
| Situation | En tant qu’utilisateur connecté j’aimerais voir le prix total TTC de l’ensemble de mon panier en fonction des articles et de leurs quantités. |
| Résultat obtenu | Les prix total est affiché en fonction du prix de chaque article et de la quantité |
| Status | ✔OK |

### 7.0 Création d’un article

|  |  |
| --- | --- |
| Nom | 7.1 Création d’un article |
| Situation | En tant qu’administrateur du site je veux avoir la possibilité de crée un nouvel article. |
| Résultat obtenu | Lors ce que le formulaire est valide le nouvel article est créer |
| Status | ✔OK |

|  |  |
| --- | --- |
| Nom | 7.2 Ajouter une image à l’article |
| Situation | En tant qu’administrateur du site je veux avoir la possibilité d’ajouter une ou plusieurs images à mon article. |
| Résultat obtenu | L’administrateur peut ajouter des images à l’article grâce à un input file |
| Status | ✔OK |

|  |  |
| --- | --- |
| Nom | 7.3 Supprimer une image à l’article |
| Situation | En tant qu’administrateur du site je veux avoir la possibilité de supprimer une image de mon article. |
| Résultat obtenu |  |
| Status |  |

|  |  |
| --- | --- |
| Nom | 7.4 Description en format HTML |
| Situation | En tant qu’administrateur du site je veux avoir la possibilité d’écrire la description d’un article au format HTML avec un éditeur de texte tel que TinyMCE. |
| Résultat obtenu | La description d’un article est faite avec TinyMCE et récupérer au format HTML |
| Status | ✔ OK |

## Évolution des tests

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No test | J-1 | J-2 | J-3 | J-4 | J-5 | J-6 | J-7 | J-8 | J-9 | J-10 | J-11 |
| 1.1 | ❌ | ❌ | ❌ | ❌ | ❌ | ❌ | ❌ | ✔ | ✔ | ✔ | ✔ |
| 2.1 | ❌ | ❌ | ❌ | ❌ | ❌ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| 2.2 | ❌ | ❌ | ❌ | ❌ | ❌ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| 3.1 | ❌ | ❌ | ❌ | ❌ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| 4.1 | ❌ | ❌ | ❌ | ❌ | ❌ | ❌ | ❌ | ✔ | ✔ | ✔ | ✔ |
| 4.2 | ❌ | ❌ | ❌ | ❌ | ❌ | ❌ | ❌ | ✔ | ✔ | ✔ | ✔ |
| 4.3 | ❌ | ❌ | ❌ | ❌ | ❌ | ❌ | ❌ | ✔ | ✔ | ✔ | ✔ |
| 4.4 | ❌ | ❌ | ❌ | ❌ | ❌ | ❌ | ❌ | ✔ | ✔ | ✔ | ✔ |
| 4.5 | ❌ | ❌ | ❌ | ❌ | ❌ | ❌ | ❌ | ✔ | ✔ | ✔ | ✔ |
| 5.1 | ❌ | ❌ | ❌ | ❌ | ❌ | ❌ | ❌ | ❌ | ✔ | ✔ | ✔ |
| 5.2 | ❌ | ❌ | ❌ | ❌ | ❌ | ❌ | ❌ | ❌ | ✔ | ✔ | ✔ |
| 6.1 | ❌ | ❌ | ❌ | ❌ | ❌ | ❌ | ❌ | ❌ | ✔ | ✔ | ✔ |
| 6.2 | ❌ | ❌ | ❌ | ❌ | ❌ | ❌ | ❌ | ❌ | ✔ | ✔ | ✔ |
| 6.3 | ❌ | ❌ | ❌ | ❌ | ❌ | ❌ | ❌ | ❌ | ✔ | ✔ | ✔ |
| 6.4 | ❌ | ❌ | ❌ | ❌ | ❌ | ❌ | ❌ | ❌ | ✔ | ✔ | ✔ |
| 6.5 | ❌ | ❌ | ❌ | ❌ | ❌ | ❌ | ❌ | ❌ | ✔ | ✔ | ✔ |
| 7.1 | ❌ | ❌ | ❌ | ❌ | ❌ | ❌ | ✔ | ✔ | ✔ | ✔ | ✔ |
| 7.2 | ❌ | ❌ | ❌ | ❌ | ❌ | ❌ | ✔ | ✔ | ✔ | ✔ | ✔ |
| 7.3 | ❌ | ❌ | ❌ | ❌ | ❌ | ❌ | ❌ | ❌ | ✔ | ✔ | ✔ |
| 7.4 | ❌ | ❌ | ❌ | ❌ | ❌ | ❌ | ✔ | ✔ | ✔ | ✔ | ✔ |

# Conclusion

## Difficultés rencontrées

Les difficultés que j’ai rencontrées durant mon TPI sont principalement :

* **La planification de mes tâches :** je suis quelqu’un qui a l’habitude de faire les choses sans les planifier.
* **La gestion du panier :** la gestion du panier a été très technique. Malheureusement pour avoir quelque chose de correcte j’ai été contraint à devoir faire de l’asynchrone avec ajax.
* **La page d’inscription :** j’ai rencontré des difficultés lors de la réalisation de la page d’inscription car elle contenait beaucoup de champs avec pas mal de condition pour la validation du formulaire. J’ai créé plusieurs fonctions en javascript pour gérer l’inscription. J’ai également des expressions régulières pour le filtrage des valeurs.
* **Le temps à disposition par rapport à la taille de mon projet :** Je pense que c’est de loin la chose qui a été le plus compliqué dans mon cas car j’avais beaucoup de travail à faire entre le site et la documentation

## Améliorations possibles

L’arborescence du projet pourrait être améliorer tout comme la séparation des fichiers pour éviter les fonctions d’affichage ou les scripts directement sur la page principale. Le projet pourrait également être améliorée en utilisant le design pattern MVC (model vue contrôler) qui permet d’avoir une meilleure organisation du code source. L’utilisation de plus de classes en php. Faire des requêtes ajax dans mes formulaires.

## Bilan personnel

Ce projet m’a appris à gérer un projet de grande envergure dans un temps imparti. Au début du projet j’ai très mal estimé la durée des tâches. La planification des tâches m’a pris énormément de temps ce qui a repoussé la création des premières pages de mon site.  
Certaines page m’ont pris beaucoup plus de temps que ce que j’avais planifier au départ.  
Par conséquence j’ai pris énormément de retard ce qui m’a forcé à condenser tout mon travail sur les 4 derniers jours pour finir mon projet dans les temps. Au cours de cette épreuve j’ai appris à gérer le peu de temps qu’il me restait pour avancer et terminé mon projet dans les temps. En changeant ma façon de faire les choses j’ai pu être plus efficaces et plus rapides dans mes tâches. Ce TPI m’a donné un aperçu de ce qui se passe dans le monde professionnel.

## Remerciements

Je tiens à remercier M. Aigroz pour son aide et son soutien tout au long de mon TPI.

# Glossaire

## Librairie

ensemble de codes sources précompilés et réutilisables qui fournissent des fonctionnalités spécifiques.

## Carrousel

fonctionnalité d'interface utilisateur qui permet de faire défiler ou de naviguer entre plusieurs éléments (tels que des images, des vidéos ou des contenus) de manière automatique ou interactive.

## Framework

ensemble de bibliothèques, d'outils, de conventions et de bonnes pratiques qui fournissent une structure et un environnement de travail pour faciliter le développement d'applications.

## Script

Ensemble d'instructions ou de commandes qui sont écrits dans un langage de programmation spécifique pour effectuer une tâche ou une série de tâches automatisées.

## CSV

CSV (Comma-Separated Values) est un format de fichier utilisé pour stocker des données sous forme de table.

## Javascript

C’est un langage de programmation opérant côté client.

## AJAX

AJAX est l'acronyme d'Asynchronous JavaScript And XML, autrement dit JavaScript Et XML Asynchrones. AJAX est un concept de programmation Web reposant sur plusieurs technologies comme le JavaScript. Dans mon cas je retourne des informations en JSON

## JSON

JSON (JavaScript Object Notation) est un format de données textuelles. Le JSON est généralement utilisé pour représenter des données structurées telles que des objets, des tableaux et des chaînes de caractères. Les données sont organisées en paires clé-valeur.

## MVC

MVC (Modèle-Vue-Contrôleur) est un modèle architectural utilisé dans le développement logiciel pour organiser et structurer les applications. Il divise les composants d'une application en trois parties distinctes : le modèle, la vue et le contrôleur.