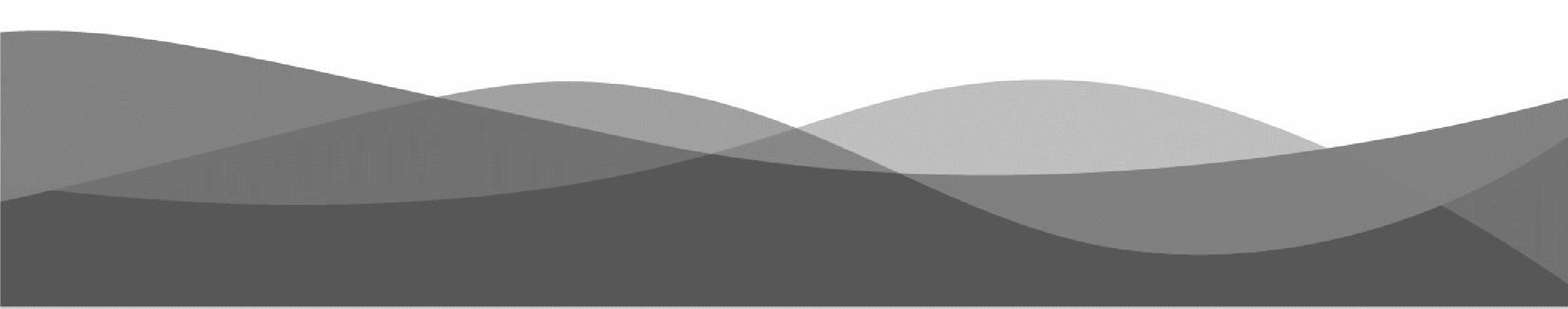
Fitness Progress

Documentation technique



Travail personnel individuel (TPI)

1. Table des matières

[2. Table des versions 4](#_Toc254936)

[3. Introduction 4](#_Toc254937)

[4. Rappel de l’énoncé 4](#_Toc254938)

[4.1 Organisation 4](#_Toc254939)

[4.2 Livrables 5](#_Toc254940)

[4.3 Matériel et logiciels à disposition 5](#_Toc254941)

[4.4 Description de l’application 5](#_Toc254942)

[5. Méthodologie 8](#_Toc254943)

[1. S’informer 8](#_Toc254944)

[2. Planifier 9](#_Toc254945)

[3. Décider 9](#_Toc254946)

[4. Réaliser 9](#_Toc254947)

[5. Contrôler 9](#_Toc254948)

[6. Évaluer 10](#_Toc254949)

[6. Planification 10](#_Toc254950)

[6.1 Product backlog 10](#_Toc254951)

[6.2 Diagramme de Gantt 15](#_Toc254952)

[7. Généralités concernant l’implémentation 17](#_Toc254953)

[7.1 Base de données 17](#_Toc254954)

[7.2 Dictionnaire de données 19](#_Toc254955)

[7.3 Structure du projet 24](#_Toc254956)

[8. Outils externes 25](#_Toc254957)

[8.1 Boostrap 25](#_Toc254959)

[8.2 Git 26](#_Toc254961)

[9. Analyse des fonctionnalités majeures 26](#_Toc254962)

[9.1 Affichage des produits 26](#_Toc254963)

[9.2 Inscription 28](#_Toc254964)

[9.3 Validation du panier 28](#_Toc254965)

[9.4 Suppression d’un compte 29](#_Toc254966)

[9.5 Pagination sur la page « produits » 29](#_Toc254967)

[10. Plan de test et tests 30](#_Toc254968)

[10.1 Périmètre des tests 30](#_Toc254969)

[10.2 Équipement de test 30](#_Toc254970)

[10.3 Scénarios de test 31](#_Toc254971)

[10.4 Évolution des tests 37](#_Toc254972)

[10.5 Échec des tests et raison 37](#_Toc254973)

[11. Conclusion 39](#_Toc254974)

[11.1 Difficultés rencontrées 39](#_Toc254975)

[11.2 Variantes de solutions et choix 40](#_Toc254976)

[11.3 Améliorations possibles 40](#_Toc254977)

[11.4 Bilan personnel 41](#_Toc254978)

[11.5 Remerciements 42](#_Toc254979)

[12 Annexes 43](#_Toc254980)

[12.1 Bibliographie 43](#_Toc254981)

[12.2 Table des illustrations 43](#_Toc254982)

# 2. Table des versions

|  |  |  |  |
| --- | --- | --- | --- |
| N° de version | Date | Auteur | Changements apportés |
| 1.0 | 21.03.2023 | Gabriel Martin | Version finale du document pour le rendu du TPI |

# 3. Introduction

Ce document représente la documentation technique du projet Ear and Write Studio. Cette documentation technique a pour but de présenter en détail les différents aspects du projet, depuis la conception jusqu'à la mise en production, en passant par le développement et les tests. Nous allons expliquer les choix technologiques que nous avons faits, les fonctionnalités que nous avons implémentées, ainsi que les solutions mises en place pour garantir la qualité et la sécurité de la plateforme.

# 4. Rappel de l’énoncé

### 4.1 Organisation

|  |
| --- |
| Élèves |
| Gabriel Martin  [gabriel-henri.mrtn@eduge.ch](mailto:gabriel-henri.mrtn@eduge.ch)  Antoine Davet  [antoine.dvt@eduge.ch](mailto:antoine.dvt@eduge.ch)  Sergio Catalan  [sergio.ctlng@eduge.ch](mailto:sergio.ctlng@eduge.ch)  Lucas Billegas  [lucas.bllgs@eduge.ch](mailto:lucas.bllgs@eduge.ch) |

### 4.3 Matériel et logiciels à disposition

* Un PC standard école, 1 écrans
* Windows 10
* Visual Studio Code
* Outil de versionnage de code (Git, avec dépôt distant sur Github)
* Navigateur web (Google Chrome)
* Outil bureautique à choix pour les documents (Google Docs,PowerPoint et Excel)

### 4.4 Description de l’application

Le site web que nous avons développé est une plateforme innovante qui permet aux utilisateurs de rechercher une instrumentale sur YouTube et de l'écouter tout en écrivant simultanément.

Une fois que l'utilisateur a trouvé la musique qu'il souhaite écouter, il peut commencer à écrire tout en laissant la musique jouer en arrière-plan. Pour sauvegarder son texte, l'utilisateur doit se connecter sur le site. Les données sauvegardées sont stockées sous forme de dossier contenant à la fois le texte et la musique dans le dossier.

Notre site web utilise l'API de YouTube pour rechercher, ainsi qu’un serveur en local pour stocker les données des utilisateurs. Les utilisateurs peuvent se connecter à leur compte et accéder à leur dossier contenant un dossier par musique et texte qu'ils ont sauvegardées.

L’application est développée à l’aide des langages web suivants : HTML/CSS, Javascript et NodeJS.

Les pages et les fonctionnalités suivantes doivent être implémentées :

* Accueil

Lorsque l'utilisateur accède à la page d'accueil, il est accueilli par un champ de recherche centré sur la page. Ce champ de recherche permet à l'utilisateur de saisir des mots clés ou des termes de recherche pour trouver les vidéos sur YouTube en rapport avec ça recherche.

* Sélection de la musique

La page de sélection de musiques est la page qui s'affiche après qu'un utilisateur a effectué une recherche sur la page d'accueil. Cette page est conçue pour afficher toutes les propositions de vidéos YouTube correspondant aux termes de recherche de l'utilisateur. Sur chaque vidéo il y a un bouton « choisir cette vidéo ». Ce bouton permet de déterminer quel vidéo l’utilisateur a sélectionné afin qu’il puisse écrire en l’écoutant.

* Inscription

La page inscription permet aux nouveaux utilisateurs de créer un compte. Pour cela, ils doivent remplir un formulaire avec un pseudo, un mot de passe et leur email.

* Connexion

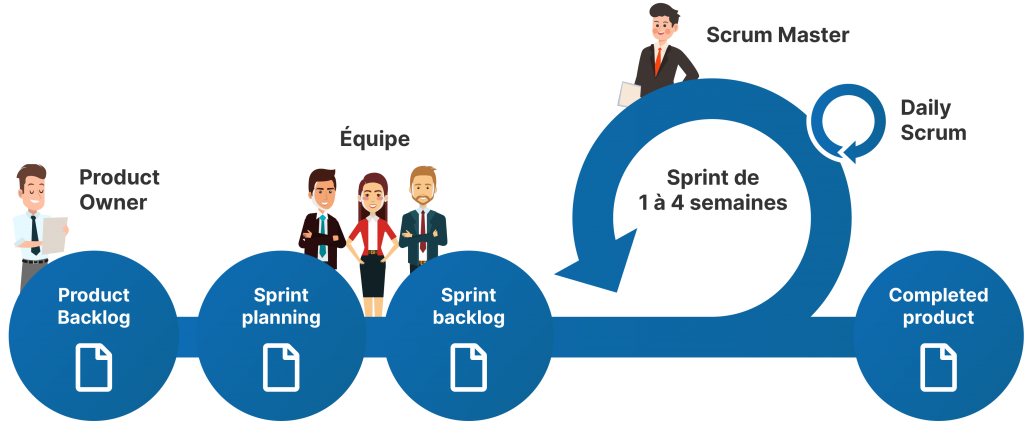
La page connexion permet aux utilisateurs non identifiés de se connecter à leur compte. Pour ce faire, ils doivent entrer leur email ainsi que leur mot de passe dans les champs affichés sur la page.

* Profil

La page « Profil » est là pour afficher les dossiers enregistrés des utilisateurs. Dans chaque dossier, on affiche le texte et la vidéo YouTube converti en mp3 sauvegarder.

# 5. Méthodologie

Pour le bon fonctionnement de mon travail j’ai dû utiliser la méthodologie qui m’a suivi pendant mes 4 ans de formation : SCRUM



# 6. Planification

### 6.1 Product backlog

|  |  |
| --- | --- |
| Nom | S4: Création du style du site |
| Description (user story) | En tant que développeur, je dois avoir un site visuel pour me concentrer seulement sur le backend de celui-ci. |
| Critère d’acceptation | Les pages mentionnées dans les consignes du TPI sont créées. |
| Priorité | P0 : Bloquant |

|  |  |
| --- | --- |
| Nom | S1 : Rechercher une vidéo YouTube |
| Description (user story) | En tant qu’utilisateur, je peux rechercher une vidéo par un mot clé via une barre de recherche. |
| Critère d’acceptation | Le test 1.1 passe |
| Priorité | P2 : Important |

|  |  |
| --- | --- |
| Nom | S2 : Affichage des vidéos sur la page sélection vidéo |
| Description (user story) | En tant qu’utilisateur, je dois voir un liste des vidéos en rapport avec les mots que j’ai recherché. |
| Critère d’acceptation | Le test 2.1 passe |
| Priorité | P1 : Critique |

|  |  |
| --- | --- |
| Nom | S3 : Page écriture |
| Description (user story) | En tant qu’utilisateur, une fois choisi sur quel vidéo je voulais écrire. J’ai une page avec la vidéo sous forme d’un audio avec un champ de texte et la possibilité de créer des boucles sur l’audio. J’ai aussi la possibilité de sauvegardé mon projet. |
| Critère d’acceptation | Le test 3.1 passe |
| Priorité | P2 : Important |

|  |  |
| --- | --- |
| Nom | S4 : Page profile |
| Description (user story) | En tant qu’utilisateur identifié, je peux accéder à mes projets sauvegardés. |
| Critère d’acceptation | Le test 6.1 passe |
| Priorité | P0 : Bloquant |

|  |  |
| --- | --- |
| Nom | S5 : Inscription |
| Description (user story) | En tant qu’utilisateur non identifié, je peux créer un compte utilisateur avec mon pseudo, mon email et mon mot de passe. |
| Critère d’acceptation | Le test 4.1 passe |
| Priorité | P0 : Bloquant |
|  |  |
| Nom | S6 : Connexion |
| Description (user story) | En tant qu’utilisateur non identifié, je peux me connecter à mon compte à l’aide de mes informations. |
| Critère d’acceptation | Le test 5.1 passe |
| Priorité | P0 : Bloquant |

|  |  |
| --- | --- |
| Nom | S7 : Utilisation d’un dépôt Git |
| Description (user story) | En tant que développeur, je dois utiliser un dépôt Git afin d’enregistrer et accéder à toutes les versions de mon projet. |
| Critère d’acceptation | Un dépôt Git est créé en privé, avec mon utilisateur ayant un droit d’accès. |
| Priorité | P0 : Bloquant |

|  |  |
| --- | --- |
| Nom | S8 : Installation de la base de données |
| Description (user story) | En tant que développeur, je dois installer une base de données qui me permettra de sauvegarder les données des utilisateurs |
| Critère d’acceptation | Il y a la base de données sur le serveur |
| Priorité | P0 : Bloquant |

|  |  |
| --- | --- |
| Nom | S9: Installation du serveur sur un pc distant |
| Description (user story) | En tant que développeur, je dois installer un serveur en local qui contiendra la base de données. |
| Critère d’acceptation | Le site et la base de données fonctionne. |
| Priorité | P0 : Bloquant |

|  |  |
| --- | --- |
| Nom | S10 : Créer une connexion à la base de données depuis une fonction JS |
| Description (user story) | En tant que développeur, je dois récupérer des données de la base de données depuis mon site. |
| Critère d’acceptation | Il est possible de se connecter à la base de données de faire des requêtes |
| Priorité | P0 : Bloquant |

### 

### 6.2 Diagramme de Gantt

Une image contenant table

Description générée automatiquement

Figure 2 Planning prévisionnel

# 7. Généralités concernant l’implémentation

### 7.1 Base de données

Mon site de vente de casquettes utilise une base de données MySQL pour stocker les utilisateurs, les produits et les commandes. La version qui a été utilisée pendant le développement de ce site est la version MySQL-5.7.24. Afin d’avoir un meilleur point de vue sur ma base de données, j’ai utilisé phpMyAdmin version 5.0.2.

Un modèle de données était fourni dans l’énoncé, mais comme nous avions le droit de l’adapter. J’ai utilisé cette option sur quelques tables avec bien sûr, l’autorisation de mon formateur.

Modèle donné avec l’énoncé :

Figure 4 Modèle fourni dans l'énoncé

Modèle final de la base de données :

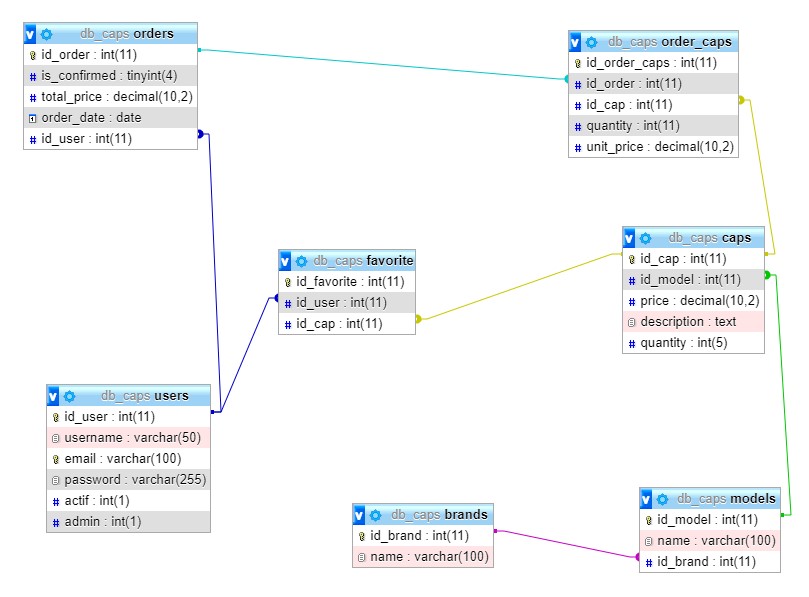


Figure 5 Modèle final

# 8. Outils externes

### 8.1 Boostrap

[Bootstrap](https://getbootstrap.com/) est une collection d'outils utiles à la création du design de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option. [source](https://fr.wikipedia.org/wiki/Bootstrap_(framework))

L’utilisation de Bootstrap était obligatoire suite à l’utilisation de Bootstrap Studio. C’est pour cela que nous pouvons retrouver dans le code de mon projet un grand nombre de class / id liés à Bootstrap.

### 8.2 Git

[Git](https://www.git-scm.com/) est un logiciel de gestion de versions décentralisé. C'est un logiciel libre créé par Linus Torvalds, auteur du noyau Linux, et distribué selon les termes de la licence publique générale GNU version 2.[Source](https://fr.wikipedia.org/wiki/Git)

Afin de pouvoir avoir des sauvegardes des différentes versions de mon application, j’ai utilisé un dépôt distant sur la plateforme [GitHub.](https://github.com/)

# 9. Analyse des fonctionnalités majeures

### 9.1 Affichage des produits

Lorsqu’un utilisateur identifié ou non arrive sur le site de vente, il peut retrouver la totalité des produits récupéré depuis la base de données. Les produits sont affichés avec un modèle, une marque, une quantité, un prix et une image.

Pour afficher les différents produits sur la page d’accueil ainsi que sur la page produit, j’ai utilisé une fonction qui fait un appelle à la base de données en demandant quels articles il souhaite récupérer.

Par exemple, pour récupérer simplement toutes les articles de la base de données, il m’a suffi d’envoyer cette simple requête :

SELECT \* FROM caps

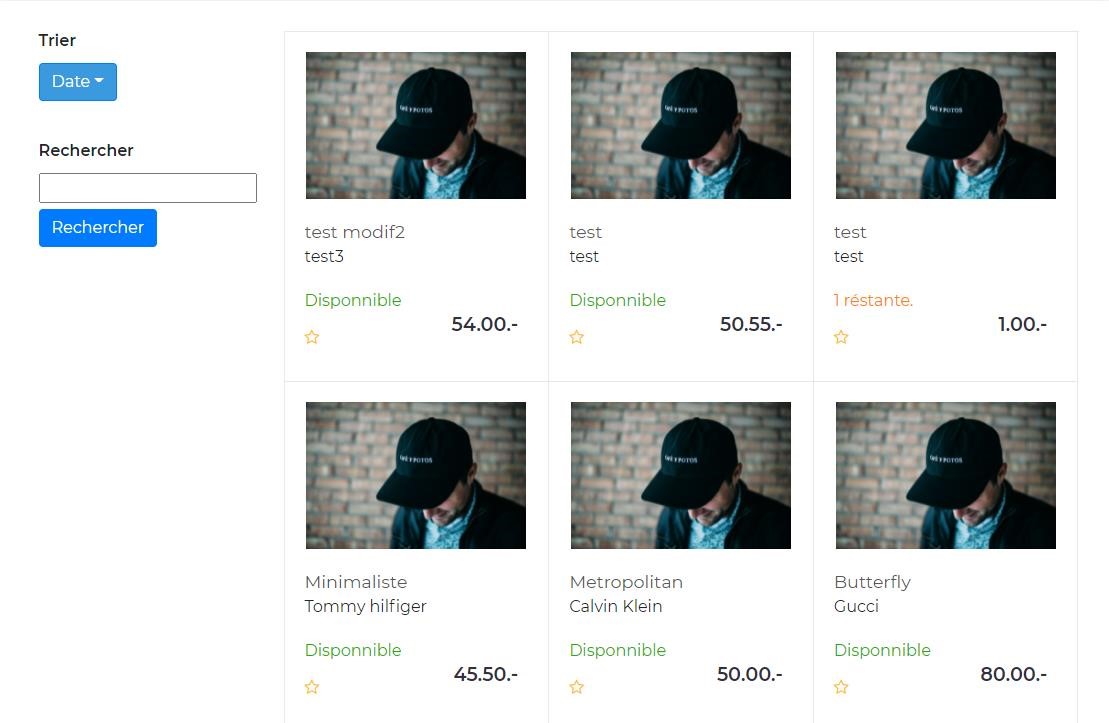


Figure 7 Résultat de la commande SQL précédente

Alors que lorsque je veux récupérer les casquettes en les triant par l’ordre alphabétique des marques, la requête envoyée est la suivante :

SELECT \* FROM caps AS c JOIN models as m ON m.id\_model = c.id\_model JOIN brand s AS b ON b.id\_brand = m.id\_brand ORDER BY b.name

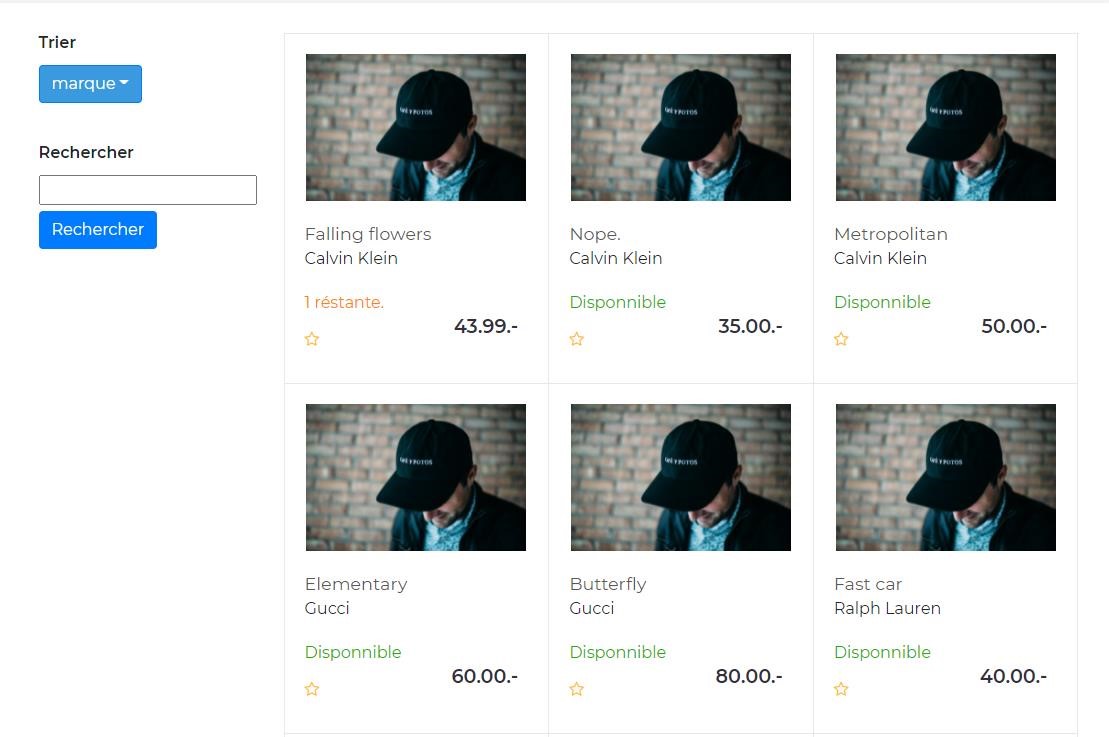
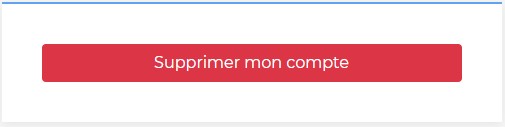


Figure 8 Résultat de la seconde commande SQL

### 9.2 Inscription

L’inscription d’un nouvel utilisateur se passe en plusieurs étapes. Une fois que l’utilisateur à rentrer ses informations, une vérification va être faite afin de voir si tous les champs ont bien été remplis. Par la suite, si le premier critère est accepté, on vérifie dans le serveur local si l’adresse email est déjà utilisé par un autre compte. Si l’adresse email n’est pas utilisé par un autre compte, le compte est alors créé.

### 9.4 Suppression d’un compte

Sur le profile, le bouton « Supprimer mon compte » est un simple bouton qui supprime l’utilisateur du dossier dans le serveur local.

### 9.5 Pagination sur la page « produits »

Contrairement à la page d’accueil qui elle, doit afficher tous les produits du catalogue, la page « produits » doit avoir un maximum de dix produits par pages. Pour des raisons de design, j’ai décidé d’en afficher seulement neuf par page. Pour cela, il a fallu que j’utilise des fonctions différentes de la page d’accueil. Pour commencer, il faut récupérer la page actuelle dans l’URL. S’il n’y a pas cette information, alors la page est par défaut définie comme la page numéro une.

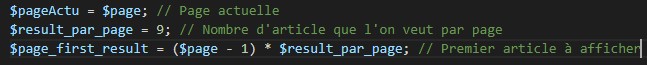


Figure 11 Variables pour la pagination

Nous pouvons voir ci-dessus les variables que nous utilisons pour la pagination avec une description à côté. Une fois arrivé à ce point-là, une fonction est appelée selon la méthode de recherche que l’on veut utiliser (classé par prix, nom, etc…). Il faut mettre alors en paramètre de ces fonctions la variable $page\_first\_result et la variable $result\_par\_page.

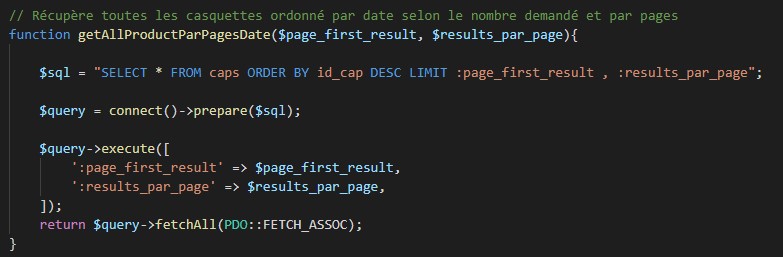


Figure 12 Function de récupération d'un certain nombre d'article

Nous pouvons voir que la fonction va alors envoyer une requête à la base de données avec comme limite, le nombre de résultat que l’on veut à partir du premier résultat que l’on a envoyé en paramètre. La requête va alors nous renvoyer les neufs résultats qu’elle aura reçus.

# 10. Plan de test et tests

### 10.1 Périmètre des tests

Mes tests ont tous étés réalisé sur le navigateur Google Chrome ainsi que sur le navigateur Mo. Afin de pouvoir tester toutes les fonctionnalités de mon projet, j’ai utilisé un compte utilisateur lambda ainsi qu’un compte administrateur. J’ai écrit une série de test que j’ai testé chaque jour, afin de pouvoir m’assurer que l’ajout d’une nouvelle fonction n’en casserai pas une autre.

### 10.2 Équipement de test

#### Environnement

Pendant le développement du projet, j’ai utilisé l’environnement de développement suivant pour effectuer mes tests:

 Google Chrome 90.0.4430.212 (64 bits) sur Windows 10 éducation, version 19041.

Une fois la partie développement terminée, j’ai effectué un nouvelle fois chaque test sous l’environnement de développement suivant afin de m’assurer que le site fonctionnait aussi sous d’autre navigateurs populaires :

 Microsoft Edge 90.0.818.62 (64 bits) sur Windows 10 éducation, version 19041.

Pour le serveur, j’ai utilisé un serveur Apache version 2.4.35 fourni via Laragon version 4.0.16 avec PHP version 7.2.19 ainsi qu’un serveur MySQL version 5.7.24.

#### Fichiers de test

Afin de pouvoir exécuter mes tests, j’ai utilisé un compte administrateur et un compte utilisateur lambda. Les produits ont aussi été créés manuellement. Toutes ces données sont disponibles dans la base de données accessible via le script SQL disponible dans le dossier /database du projet.

### 10.3 Scénarios de test

|  |  |
| --- | --- |
| Nom | 1.1 Fonction recherche |
| User story | S1 : Rechercher une vidéo YouTube |
| Situation | Lorsque nous arrivons sur la page d’accueil du site internet, nous devons voir un champ de recherche. Ce champ permettra d’effectuer la recherche. |
| Résultats | La recherche s’effectue correctement. |
| Statut | OK |

|  |  |
| --- | --- |
| Nom | 2.2 Affichage des vidéos sur la page « sélection vidéo » |
| User story | S2 : Affichage des vidéos sur la page sélection vidéo |
| Situation | Quand nous choisissons dans le menu déroulant, le triage par date, les produits s’affichent du plus récent au plus ancien ou l’inverse selon le choix que l’on fait. |
| Résultats | Les produits sont dès le début classé par date. |
| Statut | OK |

|  |  |
| --- | --- |
| Nom | 3.1 Sélection d’une vidéo |
| User story | S3 : Page écriture |
| Situation | Dès que la vidéo sera sélectionnée sa nous dirige vers une autre page. Cette page aura la vidéo en audio MP3, une fonctionnalité qui permet de faire des boucles sur l’audio, un champ un texte qui me permettra d’écrire et un bouton sauvegarde. |
| Résultats | La page est correctement affichée. |
| Statut | OK |

|  |  |
| --- | --- |
| Nom | 4.1 Affichage page profile |
| User story | S4 : Affichage page profile |
| Situation | Quand on clique sur la page profil ça doit afficher tous les projets sauvegardés de l’utilisateur |
| Résultats | Affichage correctement des projets de l’utilisateur |
| Statut | OK |

|  |  |
| --- | --- |
| Nom | 5.1 Création d’un compte |
| User story | S5 : Inscription |
| Situation | Quand on clique sur le bouton de confirmation du formulaire, le compte est créé. Si le compte est déjà existant, une erreur s’affiche. Si le compte est déjà existant mais qu’il est désactivé, le compte est réactivé. |
| Résultats | Si un compte a déjà cette adresse email, un message d’erreur s’affiche. Sinon, un nouveau compte est créé. Si l’email était déjà utilisé mais supprimé, alors les anciennes commandes sont récupérées. |
| Statut | OK |

|  |  |
| --- | --- |
| Nom | 6.1 Connexion à un compte |
| User story | S6 : Connexion |
| Situation | Quand on se connecte, un message d’erreur s’affiche si les identifiants de connexions sont incorrects ou n’existent pas. Sinon, l’utilisateur est connecté à son compte. |
| Résultats | Une erreur s’affiche si ce n’est pas bon, sinon on est connecté. |
| Statut | OK |

### 10.4 Évolution des tests

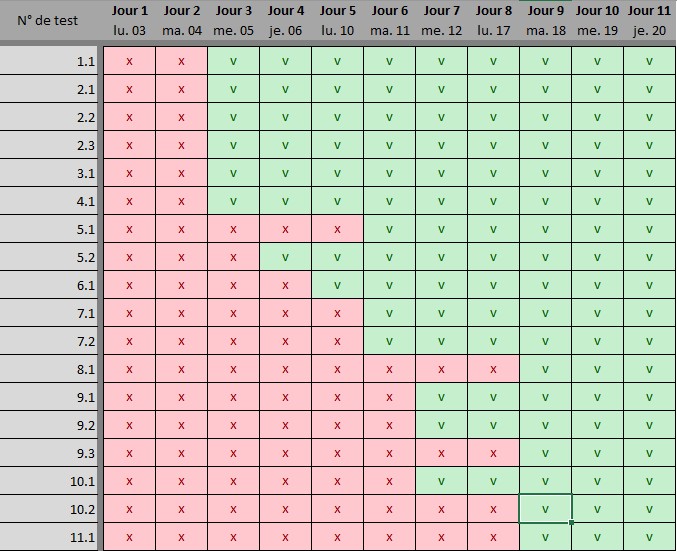


Figure 13 Résultats journalier des tests (X = KO & V = OK)

### 10.5 Échec des tests et raison

|  |  |
| --- | --- |
| Nom | 5.1 Création d’un compte |
| User story | S11 : Inscription |
| Situation | Quand on clique sur le bouton de confirmation du formulaire, le compte est créé. Si le compte est déjà existant, une erreur s’affiche. Si le compte est déjà existant mais qu’il est désactivé, le compte est réactivé. |
| Raison de l’échec | La réactivation du compte n’était pas encore implémentée. |
| Date de l’échec | 05.05.2021 & 06.05.2021 & 10.05.2021 |
| Nom | 8.1 Désinscription |
| User story | S17 : Page profile |
| Situation | Lorsqu’un utilisateur supprime son compte, un message d’avertissement s’affiche avec une demande de confirmation. |
| Raison de l’échec | Le compte est supprimé sans avertissement au préalable. |
| Date de l’échec | 11.05.2021 & 12.05.2021 & 17.05.2021 |

|  |  |
| --- | --- |
| Nom | 9.3 Confirmation des actions sur la page des produits admin |
| User story | S18 : Page produits admin |
| Situation | Pour chaque actions admin (ajouter, modifier et effacer) sur les articles, un message de confirmation s’affiche. |
| Raison de l’échec | Il n’y a pas de message de confirmation lors de la modification. |
| Date de l’échec | 12.05.2021 & 17.05.2021 |

|  |  |
| --- | --- |
| Nom | 10.2 Confirmation des actions sur les commandes  (admin) |
| User story | S19 : Page commandes admin |
| Situation | Pour chaque actions admin (ajout, modifier ou effacer) sur les commandes des utilisateurs, un message de confirmation s’affiche. |
| Raison de l’échec | Les messages de confirmation n’étaient pas implémentés. |
| Date de l’échec | 12.05.2021 & 17.05.2021 |

|  |  |
| --- | --- |
| Nom | 11.1 Confirmation des actions sur les utilisateurs  (admin) |
| User story | S20 : Page utilisateurs des admin |
| Situation | Pour chaque actions admin (ajout, modification ou suppression) des utilisateurs, un message de confirmation s’affiche. |
| Raison de l’échec | Les messages de confirmation n’étaient pas implémentés. |
| Date de l’échec | 12.05.2021 & 17.05.2021 |

# 11. Conclusion

### 11.1 Difficultés rencontrées

Durant ce projet, j’ai dû prendre plusieurs décisions notamment sur ma manière de procéder. Pour en citer un exemple, j’ai par exemple dû choisir quels environnements de travail j’allais utiliser au début de mon projet ou encore, comment j’allais créer la structure de mon site. Je ne regrette aucun de mes choix et si le projet était à refaire, je pourrais même rajouter des outils comme par exemple l’utilisation de [PHP Mess Detector.](https://phpmd.org/)

### 11.3 Améliorations possibles

# 12 Annexes

### 12.1 Bibliographie

Lors du déroulement de mon projet, j’ai utilisé les ressources suivantes pour obtenir de l’aide technique :

* La documentation officielle de PHP : <https://www.php.net/docs.php>

### 12.2 Table des illustrations

Figure 1 Méthode en 6 étapes .................................................................................... 8

Figure 2 Planning prévisionnel ................................................................................. 16

Figure 3 Planning réel ............................................................................................... 16

Figure 4 Modèle fourni dans l'énoncé ....................................................................... 17

Figure 5 Modèle final ................................................................................................ 18

Figure 6 Structure du projet ...................................................................................... 24

Figure 7 Résultat de la commande SQL précédente ................................................ 27

Figure 8 Résultat de la seconde commande SQL .................................................... 27

Figure 9 Modal .......................................................................................................... 29

Figure 10 Bouton de suppréssion ............................................................................. 29

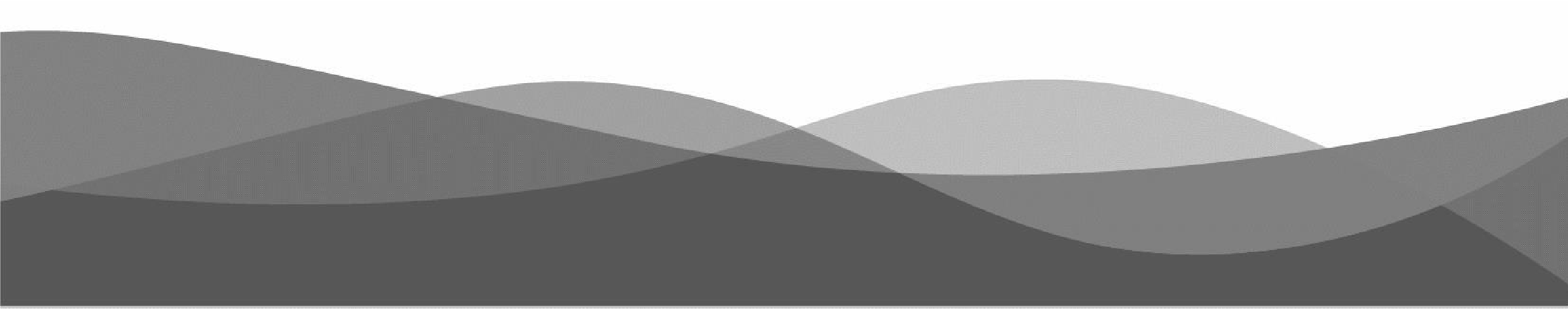
Figure 11 Variables pour la pagination ..................................................................... 29

Figure 12 Function de récupération d'un certain nombre d'article ............................ 30

Figure 13 Résultats journalier des tests (X = KO & V = OK) ..................................... 37

Code source

Projet E-commerce de casquettes



Travail personnel individuel (TPI)