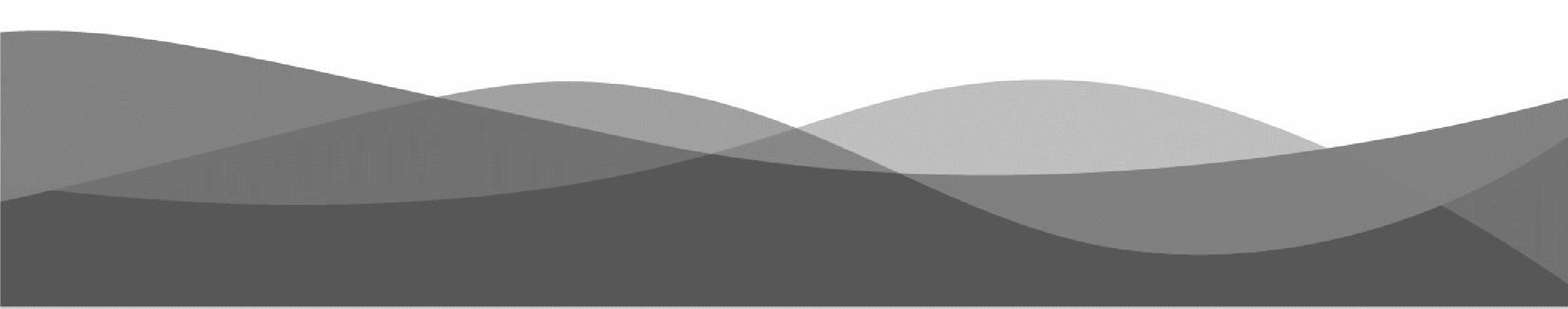
Site de E-Commerce de casquettes

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 le travail effectuer durant un atelier cherchant à simuler un TPI. Dans ce cas-là, mon TPI est un site e-commerce de vente de casquettes.

Ce site de e-commerce permet a l’utilisateur de remplir son panier de casquette, de passer des commande, se créer un compte et d’ajouter des casquettes a ses favoris.

# 4. Rappel de l’énoncé

### 4.1 Organisation

|  |  |  |
| --- | --- | --- |
| Élève |  | Maître d’apprentissage |
| Gabriel Martin |  |
| gabriel-henri.mrtn@eduge.ch |  | Fabrice Faself |

|  |  |
| --- | --- |
| Experts |  |
| ? | ? |
|  |  |

### 4.2 Livrables

24.01.2023 :

* Rien

31.01.2023 :

* Planning prévisionnel
* Back log

07.02.2023 :

* Inscription
* Accueil
* Connexion

14.02.2023 :

* Design site terminé
* Trier les casquettes

28.02.2023 :

* Page produit

07.03.2023 :

* Page panier

14.03.2023 :

* Page commande

21.03.2023 :

* Les pages admin
* Page profil
* Journal de bord
* Doc technique
* Doc utilisateurs
* Résumé TPI

### 4.3 Matériel et logiciels à disposition

* Un PC standard école, 1 écrans
* Serveur SGBD à choix (EasyPHP, Wamp, Laragon, Adminr, phpMyAdmin, autre)
* Windows 10
* Visual Studio 2017
* Logiciel de création de schémas (Visio, Gliffy, autre)
* Outil de versionnage de code (Git, avec dépôt distant sur Github / Bitbucket / GitLab)
* Navigateur web (Mozilla Firefox/Google Chrome)
* Logiciel de création de maquettes d’interfaces utilisateur (Sketch, Adobe XD, Proto.io, autre)
* Outil bureautique à choix pour les documents (Google Docs, MSOffice, OpenOffice)
* L’étudiant est autorisé à utiliser son matériel personnel au besoin

### 4.4 Description de l’application

Le projet « Site de E-Commerce de casquettes » est un site internet de vente de casquette. Les visiteurs peuvent regarder les casquettes disponibles et les ajouter à leur panier. S’ils veulent par la suite valider leur panier, ils doivent alors se connecter ou s’inscrire s’ils n’ont pas encore de compte. Une fois le panier validé, les administrateurs peuvent ensuite confirmer la commande et l’utilisateur doit alors aller chercher ses articles en magasin.

L’application sera développée à l’aide des langages web suivants : PHP, HTML et Javascript.

Les pages et les fonctionnalités suivantes doivent être implémentées :

* Accueil

La page accueil doit contenir toutes les casquettes présentes sur le site par ordre du plus récent au moins récent par défaut. Un menu déroulant permet de trier ou filtrer les casquettes par marque ou encore par prix croissant ou décroissant. Une barre de recherche est également disponible et permet de rechercher directement des casquettes par leur nom de marque ou modèle. Si une casquette n’est plus en stock, une information est affichée.

* Produits

La page produite doit elle aussi contenir la totalité des casquettes présente sur le site. Chaque casquette doit contenir un prix ainsi qu’une marque. Cette page aussi dispose d’une barre de recherche et d’un menu déroulant pour trier les casquettes.

* Item Produit

La page Item Produit est l’affichage détaillé d’un produit. Elle doit contenir le prix, la marque, le modèle et la description du produit. Un bouton doit aussi être présent afin de pouvoir ajouter l’article au panier de l’utilisateur.

* 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.

* Panier

La page panier est là pour afficher les produits enregistrés des utilisateurs. Chaque produit doit afficher sa marque, son modèle, la quantité que l’utilisateur désire acheter et le prix du produit. La page doit aussi contenir un prix total et un bouton commander.

* Une Navigation

Afin de pouvoir naviguer entre les différentes pages, le site doit contenir une barre de navigation qui affiche selon le rôle, les pages auxquelles l’utilisateur a accès.

* Un Footer

Comme la plupart des sites internet, le site doit lui aussi contenir un footer.

Pour ce projet, il sera là afin d’afficher le copyright du site.

* Profile

La page profile permet aux utilisateurs identifiés de modifier ses informations personnelles, se déconnecter ou encore supprimer son compte. Une fois le compte supprimé, si l’utilisateur recrée un compte avec la même adresse email, il doit retrouver ses anciennes commandes.

* Modification d’article

La page de modification d’article est disponible seulement pour les administrateurs. Elle leur permet de modifier un article du site.

* Ajout d’article

La page d’ajout d’article est disponible seulement pour les administrateurs.

Elle leur permet d’ajouter des articles au site.

* Commandes

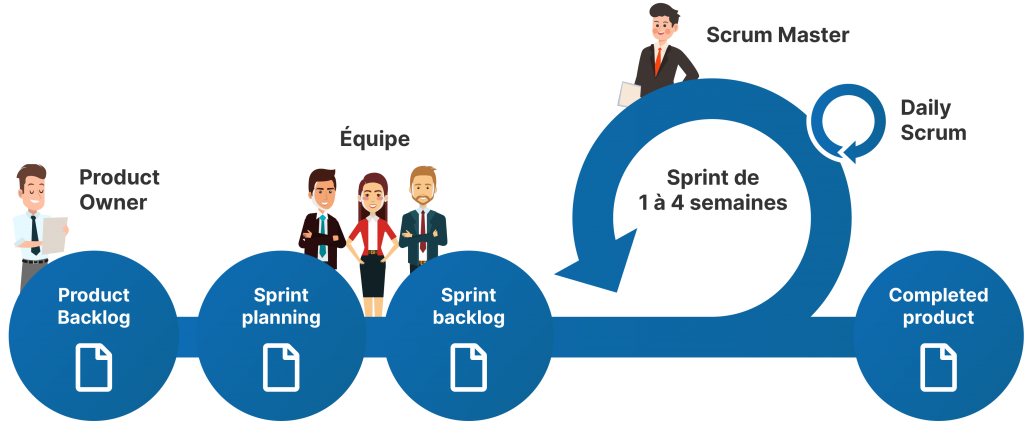
La page commandes est disponible seulement pour les administrateurs. Elle leur permet de consulter les commandes des clients, en ajouter, en éditer ou encore en supprimer.

* Utilisateurs

La page utilisateurs est disponible seulement pour les administrateurs. Elle leur permet de voir la totalité des utilisateurs du site. De pouvoir en ajouter, en modifier ou encore en supprimer.

# 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 | S1 : 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 | S2 : Installation de la base de données |
| Description (user story) | En tant que développeur, je dois installer une base de données qui respecte l’UML de l’énoncé. Pour ce faire, j’utilise le script SQL mis à disposition par mon formateur. |
| Critère d’acceptation | Les tables de l’UML sont créées à l’aide du script SQL et sont fonctionnelles. |
| Priorité | P0 : Bloquant |

|  |  |
| --- | --- |
| Nom | S3 : 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 et disposent des éléments requis (navigation, footer, contenu spécifique à la page, etc…) |
| Priorité | P0 : Bloquant |

|  |  |
| --- | --- |
| Nom | S4 : Créer une connexion à la base de données depuis une fonction PHP |
| Description (user story) | En tant que développeur, je dois récupérer des données de la base de données depuis mon site PHP. |
| Critère d’acceptation | Des données peuvent être ajoutées, éditées et supprimées depuis des fonctions PHP. |
| Priorité | P0 : Bloquant |

|  |  |
| --- | --- |
| Nom | S5 : Affichage des produits sur la page d’accueil |
| Description (user story) | En tant qu’utilisateur, je dois voir la liste des casquettes sur la page d’accueil. |
| Critère d’acceptation | Le test 1.1 passe |
| Priorité | P1 : Critique |

|  |  |
| --- | --- |
| Nom | S6 : Trier les produits |
| Description (user story) | En tant qu’utilisateur, je peux trier les produits du plus récent au moins récent, par prix croissant ou décroissant ou encore filtrer par marque. |
| Critère d’acceptation | Les tests 2.1 à 2.3 passent |
| Priorité | P3 : Secondaire |

|  |  |
| --- | --- |
| Nom | S7 : Rechercher une casquette |
| Description (user story) | En tant qu’utilisateur, je peux rechercher une casquette par sa marque ou son modèle via une barre de recherche. |
| Critère d’acceptation | Le test 3.1 passe |
| Priorité | P2 : Important |

|  |  |
| --- | --- |
| Nom | S8 : Voir un produit |
| Description (user story) | En tant qu’utilisateur, je peux, en cliquant sur un produit, voir le prix, la marque, le modèle, une photo du produit et une description de celui-ci. |
| Critère d’acceptation | Le test 4.1 passe |
| Priorité | P1 : Critique |

|  |  |
| --- | --- |
| Nom | S9 : Ajout d’un produit au panier |
| Description (user story) | En tant qu’utilisateur, je peux ajouter le produit sélectionné à mon panier. |
| Critère d’acceptation | Le produit est ajouté correctement au panier |
| Priorité | P1 : Critique |

|  |  |
| --- | --- |
| Nom | S10 : Page produit |
| Description (user story) | En tant qu’utilisateur, je peux voir les produits selon mes filtres et mes tris sur une page. |
| Critère d’acceptation | Les tests 2.1 à 2.3 passent |
| Priorité | P0 : Bloquant |

|  |  |
| --- | --- |
| Nom | S11 : 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 5.1 passe |
| Priorité | P0 : Bloquant |
| Nom | S12 : 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 5.2 passe |
| Priorité | P0 : Bloquant |

|  |  |
| --- | --- |
| Nom | S13 : Affichage du panier |
| Description (user story) | En tant qu’utilisateur, je dois voir l’intégralité de mon panier, ainsi que le total de celui-ci. |
| Critère d’acceptation | Affichage des articles ainsi que calcule du prix |
| Priorité | P0 : Bloquant |

|  |  |
| --- | --- |
| Nom | S14 : Commander |
| Description (user story) | En tant qu’utilisateur identifié, je peux valider mon panier afin de finaliser la commande. |
| Critère d’acceptation | Le test 6.1 passe |
| Priorité | P1 : Critique |

|  |  |
| --- | --- |
| Nom | S15 : Ajouter des casquettes aux favoris |
| Description (user story) | En tant qu’utilisateur identifié, je veux pouvoir ajouter une ou plusieurs casquette(s) à ma liste de favoris |
| Critère d’acceptation | Le test 7.1 passe |
| Priorité | P2 : Important |

|  |  |
| --- | --- |
| Nom | S16 : Trier les casquettes par les favoris |
| Description (user story) | En tant qu’utilisateur identifié, je peux trier les casquettes par mes favorites via un filtre |
| Critère d’acceptation | Le test 7.2 passe |
| Priorité | P2 : Important |

|  |  |
| --- | --- |
| Nom | S17 : Page profile |
| Description (user story) | En tant qu’utilisateur identifié, je peux accéder à mon profile, ou je peux voir mes commandes, changer mon mot de passe et me désinscrire. |
| Critère d’acceptation | Le test 8.1 passe |
| Priorité | P0 : Bloquant |

|  |  |
| --- | --- |
| Nom | S18 : Page produits admin |
| Description (user story) | En tant qu’administrateur, je dois avoir accès à une page qui répertorie tous les articles. Sur cette page, je peux les trier, en ajouter, en modifier, en effacer, et en rechercher. |
| Critère d’acceptation | Les tests 9.1 à 9.3 passent |
| Priorité | P0 : Bloquant |

|  |  |
| --- | --- |
| Nom | S19 : Page commandes admin |
| Description (user story) | En tant qu’administrateur, je dois voir toutes les commandes, voir les commandes d’un utilisateur particulier et voir les commandes d’un ancien utilisateur. Je peux : ajouter, modifier ou effacer une commande. |
| Critère d’acceptation | Les tests 10.1 à 10.2 passent |
| Priorité | P0 : Bloquant |

|  |  |
| --- | --- |
| Nom | S20 : Page utilisateurs des admin |
| Description (user story) | En tant qu’administrateur, je dois voir tous les utilisateurs, en voir un spécifique, en voir un ancien, en ajouter, en modifier ou encore en effacer un. |
| Critère d’acceptation | Le test 11.1 passe |
| Priorité | P0 : Bloquant |

### 

### 6.2 Diagramme de Gantt

Une image contenant table

Description générée automatiquement

Figure 2 Planning prévisionnel

Figure 3 Planning réel

# 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

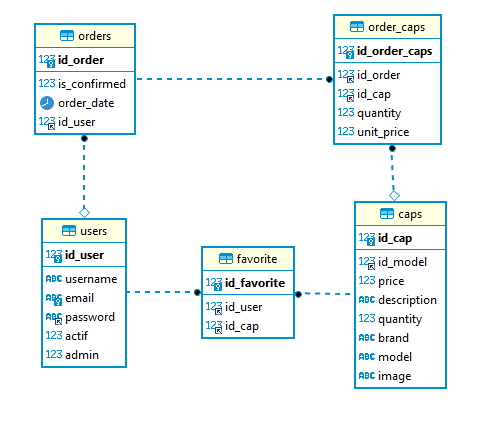
Modèle final de la base de données :

Figure 5 Modèle final

### 7.2 Dictionnaire de données

Voici le dictionnaire de données de mon application, tel que créé par phpMyAdmin :

Brands

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Colonne** | **Type** | **Null** | **Valeur par défaut** | **Commentaires** |
| id\_brand *(Primaire)* | int(11) | Non |  |  |
| name | varchar(100) | Oui | *NULL* |  |

Index

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Nom de  l’index | Type | Unique | Compressé | Colonne | Cardinalité | Interclassement | Null |
| PRIMARY | BTREE | Oui | Non | Id\_brand | 7 | A | Non |

Caps

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Colonne | Type | Null | Valeur par défaut | Commentaires |
| id\_cap (Primaire) | int(11) | Non |  |  |
| id\_model | int(11) | Oui | NULL |  |
| price | decimal(10,2) | Oui | NULL |  |
| description | text | Oui | NULL |  |
| quantity | int(5) | Non | 20 |  |

Index

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Nom de  l’index | Type | Unique | Compressé | Colonne | Cardinalité | Interclassement | Null |
| PRIMARY | BTREE | Oui | Non | Id\_cap | 13 | A | Non |
| Id\_model | BTREE | Non | Non | Id\_model | 13 | A | Oui |

Favorite

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Colonne** | **Type** | **Null** | **Valeur par défaut** | **Commentaires** |
| id\_favorite *(Primaire)* | int(11) | Non |  |  |
| id\_user | int(11) | Non |  |  |
| id\_cap | int(11) | Non |  |  |

Index

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Nom de  l’index | Type | Unique | Compressé | Colonne | Cardinalité | Interclassement | Null |
| PRIMARY | BTREE | Oui | Non | Id\_favorite | 6 | A | Non |
| Id\_user | BTREE | Non | Non | Id\_user | 1 | A | Non |
| Id\_cap | BTREE | Non | Non | Id\_cap | 6 | A | Non |

Models

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Colonne** | **Type** | **Null** | **Valeur par défaut** | **Commentaires** |
| id\_model *(Primaire)* | int(11) | Non |  |  |
| name | varchar(100) | Oui | *NULL* |  |
| id\_brand | int(11) | Oui | *NULL* |  |

Index

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Nom de  l’index | Type | Unique | Compressé | Colonne | Cardinalité | Interclassement | Null |
| PRIMARY | BTREE | Oui | Non | Id\_model | 14 | A | Non |
| Id\_brand | BTREE | Non | Non | Id\_brand | 7 | A | Oui |

Orders

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Colonne** | **Type** | **Null** | **Valeur par défaut** | **Commentaires** |
| id\_order *(Primaire)* | int(11) | Non |  |  |
| is\_confirmed | tinyint(4) | Oui | 0 |  |
| total\_price | decimal(10,2) | Oui | *NULL* |  |
| order\_date | date | Oui | *NULL* |  |
| id\_user | int(11) | Oui | *NULL* |  |

Index

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Nom de  l’index | Type | Unique | Compressé | Colonne | Cardinalité | Interclassement | Null |
| PRIMARY | BTREE | Oui | Non | Id\_order | 9 | A | Non |
| Id\_user | BTREE | Non | Non | Id\_user | 2 | A | Oui |

Order\_caps

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Colonne** | **Type** | **Null** | **Valeur par défaut** | **Commentaires** |
| id\_order\_caps *(Primaire)* | int(11) | Non |  |  |
| id\_order | int(11) | Non |  |  |
| id\_cap | int(11) | Oui | *NULL* |  |
| quantity | int(11) | Oui | *NULL* |  |
| unit\_price | decimal(10,2) | Oui | *NULL* |  |

Index

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Nom de  l’index | Type | Unique | Compressé | Colonne | Cardinalité | Interclassement | Null |
| PRIMARY | BTREE | Oui | Non | Id\_order\_caps | 20 | A | Non |
| Id\_cap | BTREE | Non | Non | Id\_cap | 7 | A | Oui |
| Id\_order | BTREE | Non | Non | Id\_order | 9 | A | Non |

Users

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Colonne** | **Type** | **Null** | **Valeur par défaut** | **Commentaires** |
| id\_user *(Primaire)* | int(11) | Non |  |  |
| username | varchar(50) | Oui | *NULL* |  |
| email | varchar(100) | Oui | *NULL* |  |
| password | varchar(255) | Oui | *NULL* |  |
| actif | int(1) | Non | 1 |  |
| admin | int(1) | Non | 0 |  |

Index

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Nom de l’index | Type | Unique | Compressé | Colonne | Cardinalité | Interclassement | Null |
| PRIMARY | BTREE | Oui | Non | Id\_user | 2 | A | Non |
| Idx\_email | BTREE | Oui | Non | Email | 2 | A | Oui |
| Idx\_password | BTREE | Non | Non | password | 2 | A | Oui |

### 7.3 Structure du projet

Pour mon projet, j’ai utilisé une structure de projet « MV ». Un dossier « Model » qui contient des éléments de style comme du CSS / Javascript ainsi que des fonctions PHP. Un dossier « Vue » qui contient les pages PHP. Voici une description détaillée de chaque fichier :

* / : Contient l’ensemble du projet, ainsi que le README pour le Git et le fichier index.php qui permet de rediriger l’utilisateur sur la page d’accueil du site.
* /database/ : Contient un script SQL qui permet d’importer la base de données sur un serveur SQL.
* /design/ : Contient le fichier source Bootstrap Studio utilisé pour créer le design du projet.
* /model/ : Contient les éléments de style CSS et Javascipt, les images ainsi que les fichiers de fonctions PHP. o /model/assets/ : Contient les éléments de style CSS et Javascript ainsi que les images du site.
  + - /model/assets/bootstrap/ : Contient les éléments du Framework Bootstrap.
    - /model/assets/css/ : Contient les feuilles de style CSS générés par Bootstrap Studio.
    - /model/assets/fonts/ : Contient les polices Figure 6 Structure du projet d’écriture du projet.
    - /model/assets/img/ : Contient les images du projet.
    - /model/assets/js/ : Contient les fichiers Javascript générés par Bootstrap Studio.
  + /model/Css-custom/ : Contient les fichiers CSS non généré par Bootstrap.
  + /model/functions/ : Contient les fichiers de fonctions PHP.
* /vue/ : Contient les pages PHP du projet.

Les dossiers en gras sont les plus importants.

# 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

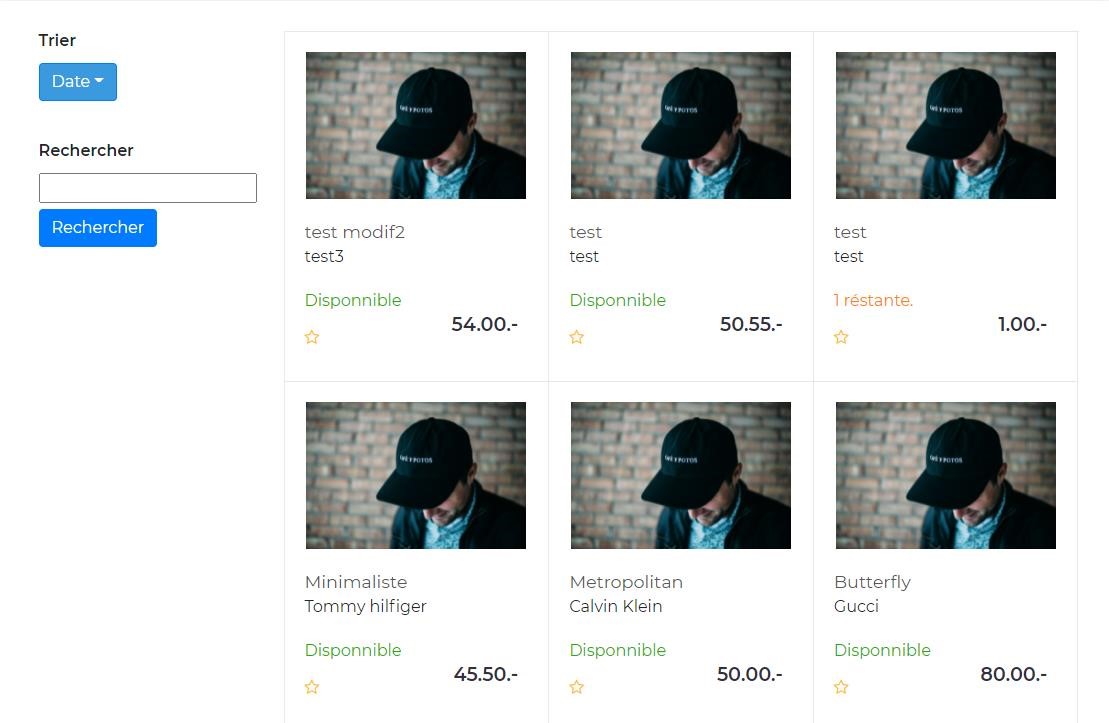


Figure 7 Résultat de la commande SQL précédente

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

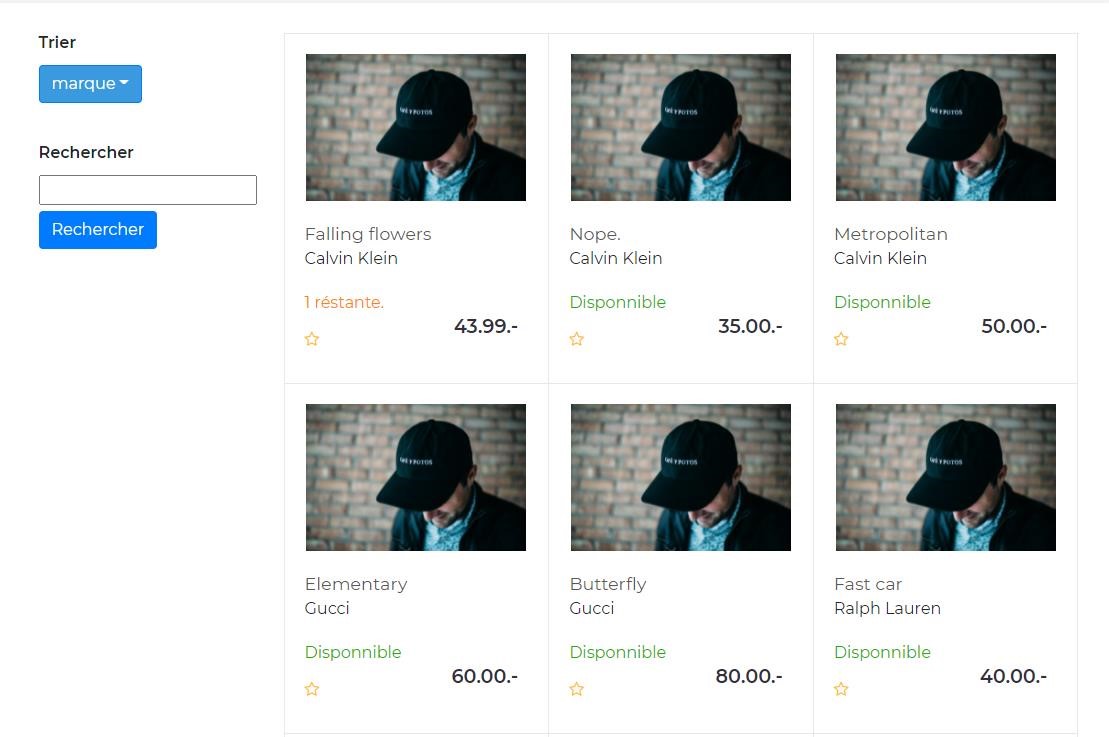


Figure 8 Résultat de la seconde commande SQL

### 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, un appel à la base de données est fait afin de pouvoir vérifier 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, sinon, une nouvelle vérification est faite afin de savoir si le compte est actif ou non. Si le compte n’est pas actif, alors le compte est réactivé et l’utilisateur peut retrouver ses anciennes commandes sur son profile.

### 9.3 Validation du panier

La page panier possède un grand de cas de figure différents. Par exemple, si l’utilisateur est identifié, si l’utilisateur n’est pas identifié, si le panier est rempli ou encore si le panier est vide. Quand un utilisateur arrive sur cette page, plusieurs vérifications sont alors faites. La première est de voir si le panier contient des articles ou non. Pour cela, une vérification de la variable de session du panier sera faite, si le panier est vide, alors un message s’affiche et le bouton ne peut être activé car il aura comme option « disabled ». Sinon, les articles sont affichés et c’est à ce moment-là qu’une nouvelle vérification va être faite. La seconde vérification est de savoir si l’utilisateur est identifié ou non. Pour cela, il faut regarder s’il existe une variable « $\_SESSION[‘role’] ». Si la variable existe, c’est que l’utilisateur est connecté et le bouton est activé. Sinon le bouton est aussi activé mais si l’utilisateur clique dessus, il est renvoyé à la page de connexion. Il reste une dernière vérification, il s’agit de la vérification du rôle de l’utilisateur connecté. Si le rôle est « user », alors rien ne change mais si le rôle est « admin », un <input type=’email’> est alors ajouté audessus du bouton de validation. Cela permet à l’administrateur de pouvoir passer une commande pour un autre compte.

### 9.4 Suppression d’un compte

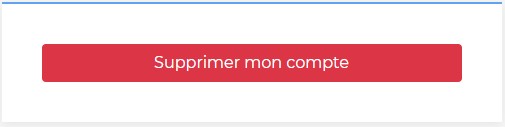
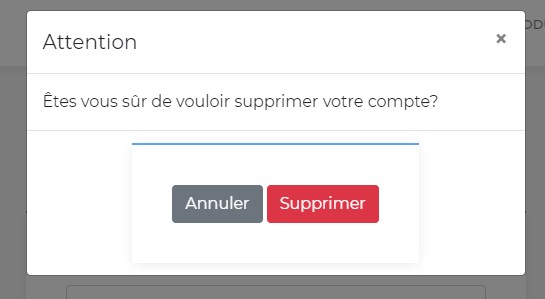
Sur le profile, le bouton « Supprimer mon compte » est un enfaite un simple bouton qui a pour rôle d’appeler une fonction Javascript de Bootstrap qui

Figure 10 Bouton de suppréssion va afficher un élément « modal ». Celui-ci contient alors un « form » qui demande une confirmation à l’utilisateur, s’il clique sur « close » alors le modal va simplement disparaitre. Sinon, la page va récupérer le formulaire et ainsi désactiver le compte dans la base de données en déconnectant l’utilisateur.



Figure

9

Modal

### 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.

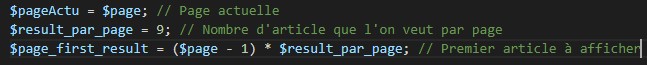


Figure 11 Variables pour la pagination

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.

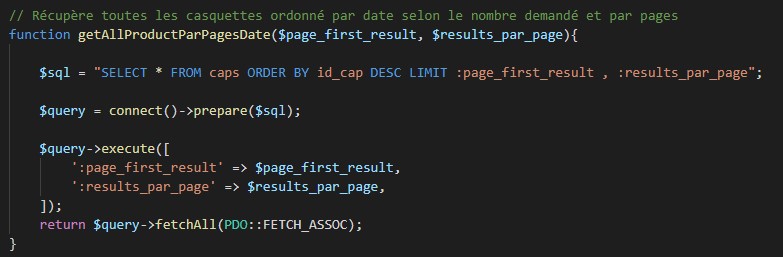


Figure 12 Function de récupération d'un certain nombre d'article

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 Edge. 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 Affichage des produits sur la page d’accueil |
| User story | S5 : Affichage des produits sur la page d’accueil |
| Situation | Lorsque nous arrivons sur la page d’accueil du site internet, nous devons voir toutes les casquettes. |
| Résultats | Les produits s’affichent correctement. |
| Statut | OK |

|  |  |
| --- | --- |
| Nom | 2.1 Triage des produits par date sur la page d’accueil |
| User story | S6 : Trier les produits |
| 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 | 2.2 Triage des produits par prix sur la page d’accueil |
| User story | S6 : Trier les produits |
| Situation | Quand nous choisissons dans le menu déroulant, le triage par prix, les produits s’affichent du plus chère au moins chère ou l’inverse selon le choix que l’on fait. |
| Résultats | Les produits sont affichés selon l’ordre que l’on choisit, par prix. |
| Statut | OK |

|  |  |
| --- | --- |
| Nom | 2.3 Filtrage des produits par la marque |
| User story | S6 : Trier les produits |
| Situation | Quand nous choisissons de filtrer les produits par une marque, seul les produits de la marque choisie s’affichent. |
| Résultats | Les produits sont classés par ordre alphabétique des marques. |
| Statut | OK |

|  |  |
| --- | --- |
| Nom | 3.1 Recherche de produit |
| User story | S7 : Rechercher une casquette |
| Situation | Quand nous cherchons un produit via une barre de recherche, les articles contenant les mots clés de la recherche s’affichent. Si aucun article est trouvé, un message d’avertissement s’affiche. |
| Résultats | Les produits s’affichent selon ce que l’on veut rechercher, Un message s’affiche s’il n’y a pas d’articles trouvé. |
| Statut | OK |

|  |  |
| --- | --- |
| Nom | 4.1 Affichage du produit |
| User story | S8 : Voir un produit |
| Situation | En cliquant sur un produit, les informations comme le prix, la marque, le modèle ainsi que la photo de celui-ci s’affiche correctement. Si le produit n’est plus en stock, un message d’avertissement s’affiche et le bouton de commande est désactivé. |
| Résultats | Les informations du produit s’affichent, si le produit n’est plus disponible, on ne peut pas le commander. |
| Statut | OK |

|  |  |
| --- | --- |
| 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. |
| 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 récupéré. |
| Statut | OK |

|  |  |
| --- | --- |
| Nom | 5.2 Connexion à un compte |
| User story | S12 : 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 |

|  |  |
| --- | --- |
| Nom | 6.1 Validation du panier |
| User story | S14 : Commander |
| Situation | Si l’utilisateur n’est pas connecté à son compte, alors il est redirigé vers la page de connexion. |
| Résultats | L’utilisateur est redirigé vers la page de connexion |
| Statut | OK |

|  |  |
| --- | --- |
| Nom | 7.1 Ajout aux favoris |
| User story | S15 : Ajouter des casquettes aux favoris |
| Situation | Si l’utilisateur n’est pas connecté à son compte, alors il est redirigé vers la page de connexion. |
| Résultats | L’utilisateur non connecté est bien redirigé |
| Statut | OK |

|  |  |
| --- | --- |
| Nom | 7.2 Triage des casquettes favorite |
| User story | S16 : Trier les casquettes par les favoris |
| Situation | Affiche les produits favoris de l’utilisateur. S’il n’y a eu pas, alors affiche un message d’avertissement. Ce mode de triage doit être disponible seulement quand l’utilisateur est connecté. |
| Résultats | Le mode ne s’affiche que quand l’utilisateur est connecté à son compte. S’il n’a pas de favoris, alors un message s’affiche |
| Statut | OK |

|  |  |
| --- | --- |
| 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. |
| Résultats | Une demande de confirmation s’affiche, si l’utilisateur clique sur « Supprimer », le compte est alors désactivé. |
| Statut | OK |

|  |  |
| --- | --- |
| Nom | 9.1 Accès à la page de produits admin |
| User story | S18 : Page produits admin |
| Situation | Si l’utilisateur n’est pas un admin, il est automatiquement renvoyé sur la page d’accueil du site. |
| Résultats | L’utilisateur est renvoyé sur la page d’accueil |
| Statut | OK |

|  |  |
| --- | --- |
| Nom | 9.2 Recherche d’article sur la page des produits admin |
| User story | S18 : Page produits admin |
| Situation | Si aucun résultat n’est trouvé, un message d’avertissement s’affiche. |
| Résultats | Un message s’affiche quand il n’y a pas de résultat |
| Statut | OK |

|  |  |
| --- | --- |
| 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. |
| Résultats | Les messages sont affichés pour toutes les actions |
| Statut | OK |
| Nom | 10.1 Affichage des commandes d’un utilisateur |
| User story | S19 : Page commandes admin |
| Situation | Affiche les commandes d’un utilisateur, s’il n’en a pas, alors affiche un avertissement. |
| Résultats | Un message s’affiche quand il n’y a pas de commandes |
| Statut | OK |

|  |  |
| --- | --- |
| 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. |
| Résultats | Les messages sont affichés pour toute les actions. |
| Statut | OK |

|  |  |
| --- | --- |
| 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. |
| Résultats | Les messages sont affichés pour toutes les actions |
| Statut | OK |

### 10.4 Évolution des tests

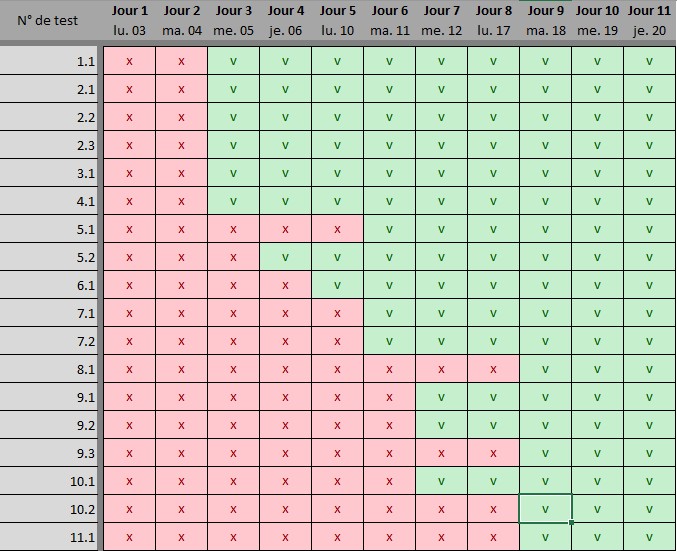


Figure 13 Résultats journalier des tests (X = KO & V = OK)

### 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

Je n’ai pas eu de grosse difficultés particulière. Cependant je peux en citer quelques une qui sont quand même survenues :

* La pagination sur la page des produits.

Je n’avais jamais fait de pagination auparavant, je suis donc allé chercher de l’aide sur internet afin de savoir comment faire cela. J’ai trouvé comment faire cette pagination sur [Javatpoint.com.](https://www.javatpoint.com/php-pagination) J’ai aussi compris le fonctionnement, il s’agit enfaite de démarrer la sélection dans la base de données à partir d’un certain nombre d’objet et de la finir à une certaine limite.

* La suppression d’articles par l’administrateur.

Tout allait bien lorsque j’ai commencé à faire la partie administrateur de la gestion des articles, mais après avoir testé la suppression, un élément m’a échappé. Si un utilisateur a dans sa liste de favoris un produit supprimé, alors il s’affichera quand même mais sans les données de celui-ci. J’ai donc dû par la suite ajouter une suppression de l’article dans les favoris des utilisateurs lors de la suppression d’un article. Un article supprimé ne s’affichera plus non plus dans les historiques de commandes, la commande sera quand même visible avec les détails des autres produits ainsi que le montant de celle-ci.

### 11.2 Variantes de solutions et choix

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

De mon point de vue, le site serait déjà fonctionnel si on corrige les problèmes mentionné plus tôt (Comme le changement de nombre de produit dans le panier). Cependant, nous pourrions ajouter quelques points pour le rendre plus performant et plus simple pour l’utilisateur.

1. Des images personnalisées pour chaque produit.

Effectivement, dans la version actuel, les produits ont une seule et même image par défaut. Si le site devait un jour être publique, il devrait selon moi y avoir une option d’ajout d’image du produit lors de la création de celui-ci.

1. Une pagination sur la page d’accueil.

Dans l’énoncé, il était spécifié que la page d’accueil devait afficher tous les produits mais si le site possède beaucoup de produits (plusieurs centaine), ce type d’affichage pourrait être problématique lors du chargement de la page.

1. Un carrousel des produits populaire sur la page d’accueil.

Il suffirait d’ajouter un champ dans la table « caps » qui contient le nombre de fois qu’un produit a été acheté. Nous n’aurions plus qu’à récupérer les 10 produits les plus acheté et à les afficher dans un carrousel sur la page d’accueil.

1. Un moyen de payement en ligne.

Pour des questions pratique, le site n’a pas cette option car il est développé dans le cadre d’un TPI. Si nous étions dans une autre situation, même sans service de livraison de la part du magasin, nous pourrions faire en sorte que la commande puisse être payer en ligne à l’avance afin de ne pas s’embêter plus tard en magasin.

1. Une confirmation de création par email ainsi qu’une demande de changement de mot de passe.

Je n’ai pas implémenté ces deux options pourtant courantes sur des sites de ventes populaires car elles ne m’étaient pas demandées. Cependant, si j’en avais eu le temps, je n’aurai pas hésité à utiliser la fonction mail() en PHP afin de pouvoir donner à l’utilisateur un liens unique et temporaire qui lui permettrai de confirmer / changer le mot de passe de son compte.

1. Ajouter des données dans des balises <meta>

Certains site populaire comme [Twitter,](https://twitter.com/) [Discord](https://discord.com/) ou encore [Facebook](https://www.facebook.com/) peuvent afficher une prévisualisation de site lorsque l’on partage une url sur ces plateforme. En utilisant un [Open Graph,](https://fr.wikipedia.org/wiki/Open_Graph_Protocol) nous pourrions afficher par exemple quelques détails de la casquettes sélectionné par l’utilisateur qui a envoyé le lien.

1. Des recherches, des changement de pages et des tries en Ajax.

Je ne connais pas encore assez l’Ajax, mais afin d’éviter un rechargement de page à chaque actions de l’utilisateur, envoyer des requêtes via ce procéder et afficher les résultats grâce à du JavaScript permettrai de remédier à ce problème.

1. Ajout d’un service de contact.

Si pour une raison, un utilisateur a un problème avec un produit ou une commande, il devrait pouvoir se rendre sur une page qui lui permettrai de contacter une personne du magasin.

### 11.4 Bilan personnel

Réaliser ce projet aura été un vrai défi de mon point de vue. J’ai commencé ce TPI sans vraiment me soucier du temps et j’ai rapidement pris de l’avance. Comme j’avais cette avance, je me suis alors dit qu’il suffisait de finir le développement de mon site et d’ensuite faire de la documentation. J’ai bien sur commencé la documentation en même temps que la création de mon site, mais je l’ai pris trop à la légère. Une fois le site fini, j’ai commencé à stresser pour la partie documentation de ce projet. Je ressors donc de cette expérience avec du savoir en plus, notamment sur l’organisation et la planification de travail de cette taille, et je suis très content d’avoir pu le réaliser.

### 11.5 Remerciements

Je tiens à remercier :

* Monsieur Garcia pour son accompagnement durant mon TPI
* Madame Terrier pour ses précieux conseils sur la documentation et l’organisation qu’il faut adopter sur des travaux comme celui-ci.
* Ma famille qui m’a aidé à garder le moral dans les moments les plus difficiles de ce TPI.

# 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>
* Le site web Stack Overflow qui permet de poser des questions à d’autres développeurs : <https://stackoverflow.com/>
* Le site de tutoriels Javatpoint : <https://www.javatpoint.com/>

### 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

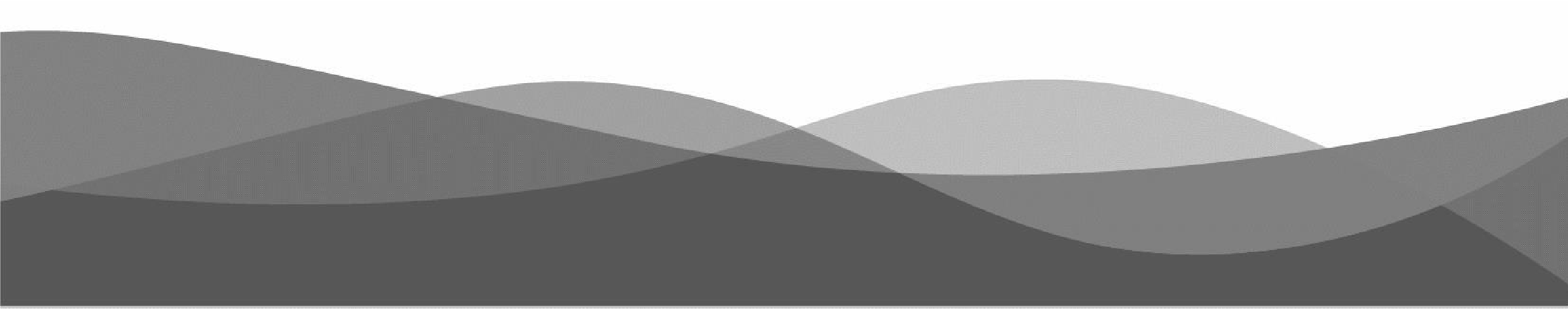
Figure 11 Variables pour la pagination ..................................................................... 29

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)