

Table des matières

[Introduction 5](#_Toc166662110)

[Versions 5](#_Toc166662111)

[Mandataire 5](#_Toc166662112)

[Mandants 5](#_Toc166662113)

[Installation 5](#_Toc166662114)

[Résumé du cahier des charges 6](#_Toc166662115)

[Matériel à disposition 6](#_Toc166662116)

[Description du projet 6](#_Toc166662117)

[Pages 6](#_Toc166662118)

[Fonctionnalités 7](#_Toc166662119)

[Points spécifiques 9](#_Toc166662120)

[Livrables 9](#_Toc166662121)

[Organisation 10](#_Toc166662122)

[Méthodologie 10](#_Toc166662123)

[S’informer 10](#_Toc166662124)

[Planifier 10](#_Toc166662125)

[Décider 10](#_Toc166662126)

[Réaliser 10](#_Toc166662127)

[Contrôler 10](#_Toc166662128)

[Évaluer 10](#_Toc166662129)

[Product Backlog 11](#_Toc166662130)

[Analyse fonctionnelle 14](#_Toc166662131)

[Use Case 14](#_Toc166662132)

[Plan de navigation 15](#_Toc166662133)

[Description des fonctionnalités majeures 15](#_Toc166662134)

[Consulter la liste des Sneakers en vente 15](#_Toc166662135)

[Filtrer les paires par marques 15](#_Toc166662136)

[Pagination 16](#_Toc166662137)

[Barre de recherche 16](#_Toc166662138)

[Réserver une paire 16](#_Toc166662139)

[Mettre une paire en favoris 16](#_Toc166662140)

[Profil 17](#_Toc166662141)

[Gérer les utilisateurs 17](#_Toc166662142)

[Maquettes 18](#_Toc166662143)

[Analyse organique 22](#_Toc166662144)

[Modèle conceptuel de données 22](#_Toc166662145)

[Modèle logique de données 23](#_Toc166662146)

[Dictionnaire de données 24](#_Toc166662147)

[USERS 24](#_Toc166662148)

[SNEAKERS 24](#_Toc166662149)

[MEETPOINT 24](#_Toc166662150)

[TRANSACTIONS 25](#_Toc166662151)

[FAVORITE 25](#_Toc166662152)

[BRANDS 25](#_Toc166662153)

[SSTATUT 25](#_Toc166662154)

[TSTATUT 25](#_Toc166662155)

[USTATUT 26](#_Toc166662156)

[MODELS 26](#_Toc166662157)

[ECOLE 26](#_Toc166662158)

[VISIBILITY 26](#_Toc166662159)

[ROLES 26](#_Toc166662160)

[DATEMEETPLACE 26](#_Toc166662161)

[Structure du projet 27](#_Toc166662162)

[Modèle 27](#_Toc166662163)

[Vue 27](#_Toc166662164)

[Contrôleur 27](#_Toc166662165)

[Description des méthodes et classes principales 28](#_Toc166662166)

[Classe Home 28](#_Toc166662167)

[Classe Gestion 28](#_Toc166662168)

[Classe Admin 29](#_Toc166662169)

[Classe Profil 29](#_Toc166662170)

[Analyse des fonctionnalités majeures 29](#_Toc166662171)

[Mettre une paire en vente 29](#_Toc166662172)

[Réserver une paire 30](#_Toc166662173)

[Ajouter une paire en favoris 30](#_Toc166662174)

[Programmes utilisés 31](#_Toc166662175)

[Visual Studio Code – 1.88 31](#_Toc166662176)

[DBeaver – 23.1.5 31](#_Toc166662177)

[Gitlab – 16.10.3 31](#_Toc166662178)

[Copilot – Free 31](#_Toc166662179)

[Pixlr – Free 31](#_Toc166662180)

[Remove.bg – Free 31](#_Toc166662181)

[Codepen - Free 31](#_Toc166662182)

[Canva – Free 31](#_Toc166662183)

[Tests 32](#_Toc166662184)

[Plan de tests 32](#_Toc166662185)

[Déroulement des tests 36](#_Toc166662186)

[Planning 37](#_Toc166662187)

[Planning prévisionnel 37](#_Toc166662188)

[Planning effectif 38](#_Toc166662189)

[Analyse de similitudes entre les deux plannings 38](#_Toc166662190)

[Conclusion 39](#_Toc166662191)

[Difficultés rencontrées 39](#_Toc166662192)

[Améliorations possibles 39](#_Toc166662193)

[Bilan personnel 40](#_Toc166662194)

[Bibliographie 40](#_Toc166662195)

[Glossaire 41](#_Toc166662196)

[Annexes 41](#_Toc166662197)

# Introduction

Ce document détaille les informations importantes qui ont permis de réaliser le projet SneakersMarket. Ce projet est un travail de diplôme, dans le cadre du TPI. Celui-ci me permet de valider les compétences et le savoir acquis lors de ces 4 années de CFC informatique section développement d’application.

SneakersMarket est un site internet d’achat-revente de chaussure. Celui-ci met en relation deux catégories d’utilisateurs, les utilisateurs qui souhaitent revendre leurs chaussures inutilisées, et les utilisateurs qui souhaitent acheter de nouvelles chaussures. L’utilisateur non connecté peut voir la liste des paires mises en vente, mais doit se connecter pour pouvoir mettre en favoris ou réserver la paire. Seules les paires marquées comme visibles et n’ayant pas encore été réservées sont disponibles à la réservation.

## Versions

|  |  |
| --- | --- |
| VERSION N° | 1.0 |
| AUTEUR | Léo Triano |
| DATE | 15.05.24 |
| DESCRIPTION | Version finale pour le rendu du TPI |

## Mandataire

|  |  |
| --- | --- |
| ÉLÈVE | Léo Triano - leo.trn@eduge.ch |
| MAITRE D’APPRENTISSAGE | M. Dido Malambu - edu-malambud@eduge.ch |

## Mandants

|  |  |
| --- | --- |
| EXPERT N°1 | M. Daniel Vanini - daniel.vanini@skyguide.ch |
| EXPERT N°2 | M. Borys Folomietow - borys@folomietow.ch |

## Installation

Pour pouvoir utiliser SneakersMarket, vous aurez besoin de :

* Visual studio code 1.88
* DBeaver 23.1.5
* Git et un accès à Gitlab 16.10.3
* Une configuration wsl avec apache2 et mysql intallé

L’installation du site se fait comme ceci :

* Clonez le projet depuis Gitlab dans votre projet /var/www/html
* Importez la dernière version du script SQL dans DBeaver
* Renommez le fichier configSample.php situé dans src/Model en config.php et modifiez les informations pour la connexion à la base de données.
* Tapez « localhost » dans votre navigateur et sélectionnez SneakersMarket

# Résumé du cahier des charges

## Matériel à disposition

* Un PC standard école
* 1 écran
* Visual studio code
* Suite Office
* MySQL
* PHP

## Description du projet

Réaliser un site web pour la vente et l'achat de sneakers entre étudiants du CFPT.

## Pages

L’application comporte :

L’application comporte :

* Une page d’accueil contient un listing des différentes chaussures ajoutés
* La possibilité de chercher une chaussure par :
  + - * Son nom
      * Par marque
* Une page avec les détails d’une paire de chaussure
* Une page de connexion
* Une page de profil pour l’étudiant
* Une page contenant les chaussures favorites de l’étudiant
* Une page de gestion contenant les chaussures achetées et celles que souhaite vendre l’étudiant

Un tableau de bord pour l’administrateur contenant les fonctionnalités de gestion :

* + - * + Une page de connexion
        + Une page de gestion des utilisateurs
        + Avec le listing des chaussures qu’ils souhaitent vendre
        + Avec le listing des chaussures qu’ils ont acheté
        + Une page de gestion des points de rencontre (cafétéria Butin, cafétéria Ternier, cafétéria Bertrand)
        + Une page de gestion des établissements (CFPT Mécanique, CFPT Construction, CFPT Informatique)

## Fonctionnalités

3.3.1 Description des actions réalisables en tant qu’utilisateur anonyme :

* L’utilisateur visualise depuis la page d’accueil les chaussures disponibles :
* Seulement 10 paires de chaussure sont visibles, les autres sont accessibles via une pagination
* Une paire de chaussures est uniquement visible si son statut est “visible”  L’utilisateur peut chercher une paire de chaussure :

 En entrant son nom

* L’utilisateur peut filtrer les chaussures par :
* Marque
* L’utilisateur visualise le détail d’une paire de chaussure

3.3.2 Description des actions réalisables en tant qu’utilisateur connecté (étudiant) :

* L’utilisateur connecté a les mêmes possibilités qu’un utilisateur anonyme

Gestion des connexions :

* L’utilisateur s’inscrit sur l’application en indiquant son nom, son prénom, un mot de passe ainsi que son adresse mail
* Le mot de passe doit avoir au minimum 8 caractères contenir un chiffre, une lettre majuscule et un caractère spécial
* Le mot de passe doit être crypté au sein de la base de donnée
* L’utilisateur ne peut s’inscrire avec une adresse mail déjà existante dans le système
* L’utilisateur se connecte en indiquant son email et son mot de passe
* L’utilisateur ne peut pas inscrire deux fois la même adresse mail
* L’utilisateur se déconnecte
* L’utilisateur peut changer ses informations

Gestion des chaussures :

* L’utilisateur peut visualiser tous ses chaussures qui sont en vente avec leurs statuts (vendu, disponible) et celles qu’il a acheté
* L’acheteur peut réserver une paire de chaussures auprès d’un vendeur
  + - * Une paire de chaussures ne peut pas être réservée plus d’une fois auprès d’un vendeur
      * Une paire de chaussures est considérée comme vendu lorsque son statut est « vendu »  L’acheteur et le vendeur peuvent supprimer la réservation d’une paire de chaussures
      * Une réservation peut être supprimée à tout moment
      * Une réservation annulée rend la paire de chaussures de nouveau disponible à la vente
      * L’acheteur peut supprimer uniquement une réservation qu’il a initiée
      * Le vendeur peut annuler les réservations de ses chaussures

3.3.3 Description des actions réalisables en tant qu’administrateur :

* L’administrateur a les mêmes possibilités qu’un utilisateur connecté  L’administrateur peut visualiser, modifier, supprimer un utilisateur
  + - * On peut bloquer un utilisateur
      * On peut débloquer un utilisateur
      * On ne peut pas supprimer un utilisateur qui a vendu des chaussures  L’administrateur peut ajouter un utilisateur
      * Un utilisateur possède un nom, un prénom, un mot de passe ainsi que son adresse mail
      * Le mot de passe doit avoir au minimum 8 caractères contenir un chiffre, une lettre majuscule et un caractère spécial
      * Le mot de passe doit être crypté au sein de la base de donnée
      * L’adresse mail ne doit pas être déjà présente dans le système
* L’administrateur peut supprimer une paire de chaussure. On peut supprimer une paire de chaussure uniquement s’il n’a pas été réservée ou vendue.

## Points spécifiques

A14 : On ne peut pas inscrire un étudiant avec une adresse mail déjà existante

A15 : On ne peut pas réserver une paire de chaussure qui est déjà réservé

A16 : Le mot de passe doit avoir au minimum 8 caractères avec un chiffre, une lettre majuscule et un caractère spécial

A17 : Un étudiant qui a un statut bloqué n’apparaît plus en tant que vendeur

A18 : Une paire de chaussure qui n’a plus le statut réservé est de nouveau disponible à la vente

A19 : Lorsqu’on filtre des chaussures par marque, uniquement les modèles de la marque apparaissent

A20 : On ne peut pas supprimer un étudiant qui a effectué des ventes

## Livrables

* Planning prévisionnel fourni le premier jour du TPI
* Manuel technique (y. compris code)
* Manuel utilisateur
* Journal de travail
* Résumé du TPI

# Organisation

## Méthodologie

Pour ce projet individuel, j’ai décidé d’opter pour la méthode en 6 étapes. Je trouve cette méthode pertinente, car elle permet de découper le travail en 6 étapes distinctes et représentatives de la partie du travail à faire.

3. Décider

2. Planifier

1. S’informer

6. Évaluer

5. Contrôler

4. Réaliser

### S’informer

J’ai commencé par soigneusement lire la grille d’évaluation qui est universelle à tous les TPI, cela me permet d’être attentif aux différents points importants dans le cadre de mon travail. J’ai également lu le cahier des charges du projet, afin de me faire une idée générale du projet à réaliser.

### Planifier

Après avoir pris connaissance des limites qui encadrent mon travail, j’ai réalisé la planification en découpant les grandes tâches en sous-sections et en y attribuant un nombre d’heures que je jugeais pertinent par rapport à la difficulté de cette dernière.

### Décider

Pendant la réalisation de ce travail, j’ai dû prendre différentes décisions afin de progresser tout en restant dans les limites d’heures que je me suis imposé dans la planification. Toutes les décisions auxquelles j’ai dû faire face sont inscrites dans mon journal de bord.

### Réaliser

Toujours en respectant les délais imposés, j’ai continuellement œuvré à la réalisation des tâches que j’avais au préalable défini. La progression, les tests réalisés, l’avancée de la documentation, les problèmes rencontrés et les solutions trouvées sont notés dans mon journal de bord.

### Contrôler

Afin de contrôler les fonctionnalités produites, j’ai réalisé un protocole de tests. Il m’a continuellement servi à contrôler les fonctionnalités déjà notées comme terminées, qu’elles soient fonctionnelles jusqu’au rendu du projet.

### Évaluer

À la fin du projet, j’ai évalué l’état final de ce dernier. Cette étape sert à recenser si le projet est complet ou non, les difficultés, les solutions et les appréciations.

# Product Backlog

Voici le Product Backlog du projet. Il comporte 7 points techniques évalués spécifiques au projet.

|  |  |
| --- | --- |
| NOM | Inscription |
| USER STORY | En tant qu’utilisateur, je peux créer un compte afin de profiter pleinement des fonctionnalités du site |
| TEST A VALIDER | 1 |
| PRIORITÉ | Prioritaire |

|  |  |
| --- | --- |
| NOM | Connexion |
| USER STORY | En tant qu’utilisateur, je peux me connecter au compte que j’ai créé afin de continuer de profiter du site |
| TEST A VALIDER | 2 |
| PRIORITÉ | Prioritaire |

|  |  |
| --- | --- |
| NOM | Profil |
| USER STORY | En tant qu’utilisateur, je peux accéder à ma page de profil afin de modifier les informations de mon compte |
| TEST A VALIDER | 5 |
| PRIORITÉ | Prioritaire |

|  |  |
| --- | --- |
| NOM | Mettre en favoris |
| USER STORY | En tant qu’utilisateur, je peux mettre une chaussure qui me plaît en favoris afin de la retrouver plus facilement plus tard |
| TEST A VALIDER | 3 |
| PRIORITÉ | Prioritaire |

|  |  |
| --- | --- |
| NOM | Gestion |
| USER STORY | En tant qu’utilisateur, je peux accéder à ma page de gestion afin de voir et de gérer les chaussures que j’ai achetées et que j’ai à vendre |
| TEST A VALIDER | 6 |
| PRIORITÉ | Prioritaire |

|  |  |
| --- | --- |
| NOM | Mettre en vente |
| USER STORY | En tant qu’utilisateur, je peux mettre en vente une paire de chaussures depuis ma page de gestion afin que les autres utilisateurs puissent la réserver |
| TEST A VALIDER | 4 |
| PRIORITÉ | Prioritaire |

|  |  |
| --- | --- |
| NOM | Page admin |
| USER STORY | En tant qu’administrateur, je peux accéder à ma page admin afin de gérer les utilisateurs, les points de rencontre et les établissements |
| TEST A VALIDER | De 7 à 9 |
| PRIORITÉ | Prioritaire |

|  |  |
| --- | --- |
| NOM | Gestion des utilisateurs |
| USER STORY | En tant qu’administrateur, je peux gérer les différents utilisateurs ainsi que les paires qu’ils ont achetées et vendues afin de garantir le bon fonctionnement du site |
| TEST A VALIDER | 10 |
| PRIORITÉ | Prioritaire |

|  |  |
| --- | --- |
| NOM | Gestion des points de rencontres |
| USER STORY | En tant qu’administrateur, je peux gérer les différents points de rencontres |
| TEST A VALIDER | 8 |
| PRIORITÉ | Prioritaire |

|  |  |
| --- | --- |
| NOM | Gestion des établissements |
| USER STORY | En tant qu’administrateur, je peux gérer les différents établissements |
| TEST A VALIDER | 9 |
| PRIORITÉ | Prioritaire |

|  |  |
| --- | --- |
| NOM | Barre de recherche |
| USER STORY | En tant qu’utilisateur, je peux rechercher une paire que je souhaite afin de la trouver plus facilement |
| TEST A VALIDER | 11 |
| PRIORITÉ | Prioritaire |

|  |  |
| --- | --- |
| NOM | Email déjà existant (A14) |
| USER STORY | En tant qu’utilisateur, je ne peux pas créer un compte avec un email déjà utilisé |
| TEST A VALIDER | 12 |
| PRIORITÉ | Prioritaire |

|  |  |
| --- | --- |
| NOM | Paire déjà réservée (A15) |
| USER STORY | En tant qu’utilisateur, je ne peux pas réserver une paire ayant déjà été réservée |
| TEST A VALIDER | 13 |
| PRIORITÉ | Prioritaire |
| NOM | Sécurité du mot passe (A16) |
| USER STORY | En tant qu’utilisateur, mon mot de passe doit contenir au moins 8 caractères, un chiffre, une lettre majuscule et un caractère spécial afin de pouvoir créer mon compte |
| TEST A VALIDER | 14 |
| PRIORITÉ | Prioritaire |

|  |  |
| --- | --- |
| NOM | Visibilité du vendeur (A17) |
| USER STORY | En tant qu’utilisateur, je ne peux plus vendre de paires si mon statut est bloqué afin de préserver la sécurité des acheteurs |
| TEST A VALIDER | 15 |
| PRIORITÉ | Prioritaire |

|  |  |
| --- | --- |
| NOM | Disponibilité de la paire (A18) |
| USER STORY | En tant qu’utilisateur, je peux de nouveau réserver une paire qui n’a plus le statut réservé |
| TEST A VALIDER | 16 |
| PRIORITÉ | Prioritaire |

|  |  |
| --- | --- |
| NOM | Filtrage des marques (A19) |
| USER STORY | En tant qu’utilisateur, je peux filtrer les paires par marque afin de mieux choisir la paire que je souhaite |
| TEST A VALIDER | 17 |
| PRIORITÉ | Prioritaire |

|  |  |
| --- | --- |
| NOM | Suppression d’un vendeur (A20) |
| USER STORY | En tant qu’administrateur, je ne peux pas supprimer un utilisateur qui a déjà effectué des ventes afin de préserver une trace des actions |
| TEST A VALIDER | 18 |
| PRIORITÉ | Prioritaire |

# Analyse fonctionnelle

## Use Case

Voici le use case prévu pour SneakersMarket. Les possibilités de l’utilisateur sont découpées en 3 :

Un utilisateur anonyme ne peut que voir la liste des paires en ventes et créer un compte ou se connecter.

Un utilisateur connecté a accès aux mêmes fonctionnalités qu’un utilisateur anonyme, mais il a également accès à toute la partie gestion, liste des favoris et réservation.

Un administrateur a accès aux mêmes fonctionnalités qu’un utilisateur connecté, et il a également accès à la partie admin.

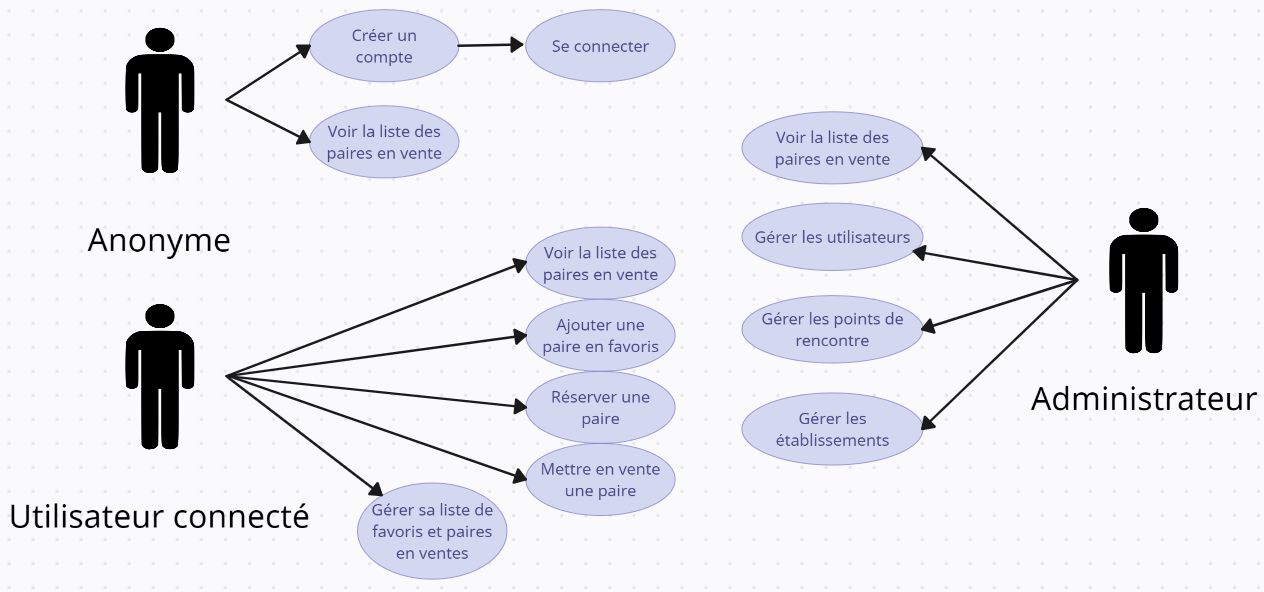


Figure 1 : Use case de SneakersMarket

## Plan de navigation

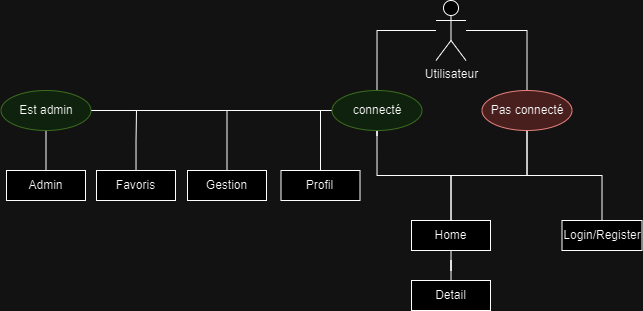


Figure 2 : Plan de navigation de SneakersMarket

## Description des fonctionnalités majeures

### Consulter la liste des Sneakers en vente

**Critères du cahier des charges :**

* L’utilisateur visualise depuis la page d’accueil les chaussures disponibles :
* Une paire de chaussures est uniquement visible si son statut est “visible”

La page d’accueil contient la liste de toutes les paires en vente :

* Qui n’ont pas été réservé
* Ou l’utilisateur n’est pas défini comme « bloqué »
* Ou la paire est définie comme « Visible »
* Ou l’utilisateur ayant posté la paire n’est pas l’utilisateur étant connecté

Les paires sont affichées avec comme informations l’image, la marque, le modèle, le prix, la taille, la description et l’utilisateur ayant posté la paire.

### Filtrer les paires par marques

**Critère du cahier des charges :**

* L’utilisateur peut filtrer les chaussures par marque

L’utilisateur peut sélectionner une marque spécifique pour afficher uniquement les paires de cette dernière. Si l’utilisateur ne sélectionne pas de marque, toutes les paires sont affichées.

### Pagination

**Critère du cahier des charges :**

* Seulement 10 paires de chaussures sont visibles, les autres sont accessibles via une pagination

En dessous de la liste des paires de chaussures, l’utilisateur dispose d’une ligne avec des numéros qui représente le nombre de pages disponibles. S’il clique sur un numéro, les paires affichées correspondront à la page sélectionnée. Pour déterminer quelles paires prendre de la base, un LIMIT a été ajouté à toutes les requêtes.

### Barre de recherche

**Critère du cahier des charges :**

* L’utilisateur peut chercher une paire de chaussures en entrant son nom

Dans le header, une barre de recherche est disponible. Si l’utilisateur recherche quelque chose, il est renvoyé sur la page « Search Result » qui affiche la liste des paires recherchées. Si aucune paire ne correspond à la recherche, un message est affiché signalant qu’aucune paire n’a été trouvée.

### Réserver une paire

**Critère du cahier des charges :**

* L’acheteur peut réserver une paire de chaussures auprès d’un vendeur

Dans la page détail, si l’utilisateur est connecté, il peut réserver une paire n’ayant pas déjà été réservée en choisissant un point de rendez-vous. Il peut annuler à tout moment sa réservation dans sa page détail. L’acheteur, de son côté, dans sa page de gestion, reçoit la demande d’achat, et peut valider ou rejeter la demande.

### Mettre une paire en favoris

**Critère du cahier des charges :**

* Liste des favoris : Les chaussures que l'étudiant a marquées comme favoris.

Dans la page détail, l’utilisateur peut mettre en favoris la paire. Celle-ci apparaît ensuite dans la page Favoris. L’utilisateur a la possibilité d’enlever la paire des favoris. L’ajout en favoris n’influence pas la visibilité de la paire, plusieurs personnes peuvent avoir la même paire en favoris. En revanche, quand une paire est validée à l’achat, la paire est enlevée des favoris de tous les utilisateurs.

### Profil

**Critère du cahier des charges :**

* L’utilisateur peut changer ses informations

L’utilisateur connecté dispose d’une page profil, accessible en cliquant sur son nom sur le header. Sur cette page, il y figure un formulaire contenant les informations de l’utilisateur. L’utilisateur a la possibilité de changer ses informations. Si l’utilisateur soumet le formulaire, les changements se font en base et dans la session.

### Gérer les utilisateurs

**Critères du cahier des charges :**

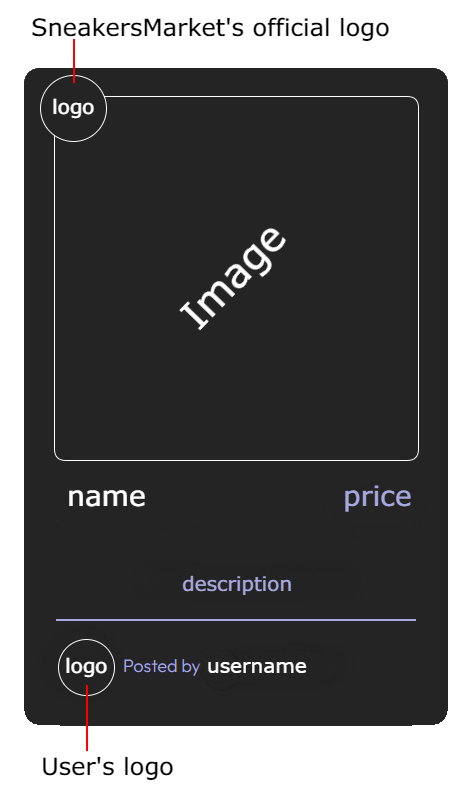
* L’administrateur peut visualiser, modifier, supprimer un utilisateur
* On peut bloquer un utilisateur
* On peut débloquer un utilisateur
* On ne peut pas supprimer un utilisateur qui a vendu des chaussures

Dans sa page Admin, une liste contenant tous les utilisateurs est affichée. Pour chaque utilisateur, l’administrateur à la possibilité de :

* Supprimer l’utilisateur s’il n’a concrétisé aucune vente
* Bloquer l’utilisateur s’il a déjà concrétisé une ou plusieurs ventes
* Modifier les informations de l’utilisateur
* Voir les paires que l’utilisateur a en vente et a vendues

Quand l’administrateur clique sur « Modifier », il est redirigé sur la page « Modify User », similaire à la page profil mais contenant les informations de l’utilisateur voulant être modifié. Une fois le formulaire soumis, les modifications se font dans la base de données.

## Maquettes



Pour l’affichage des paires de chaussures, j’ai décidé de partir sur ce design. Les cartes permettent de mieux séparer les différentes paires, ainsi que de mettre en avant les informations importantes, comme par exemple le nom ou le prix de la paire de chaussures.

Figure 3 : Maquettes des cards

Pour le design de la page détail, j’ai adoré la disposition d’une template trouvée (voir bibliographie), j’ai donc repris la disposition des éléments et adapté pour y ajouter les informations supplémentaires, comme la sélection de la taille.



Figure 4 : Maquettes de la page détail

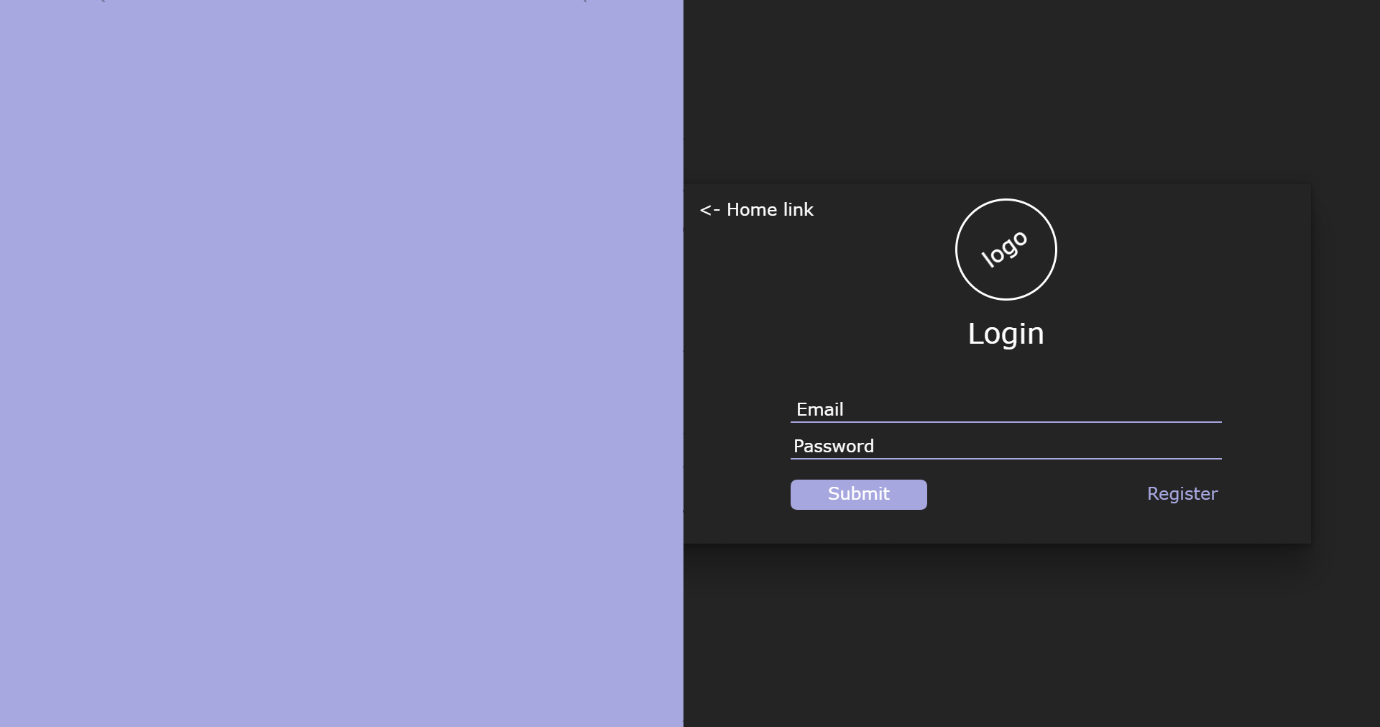
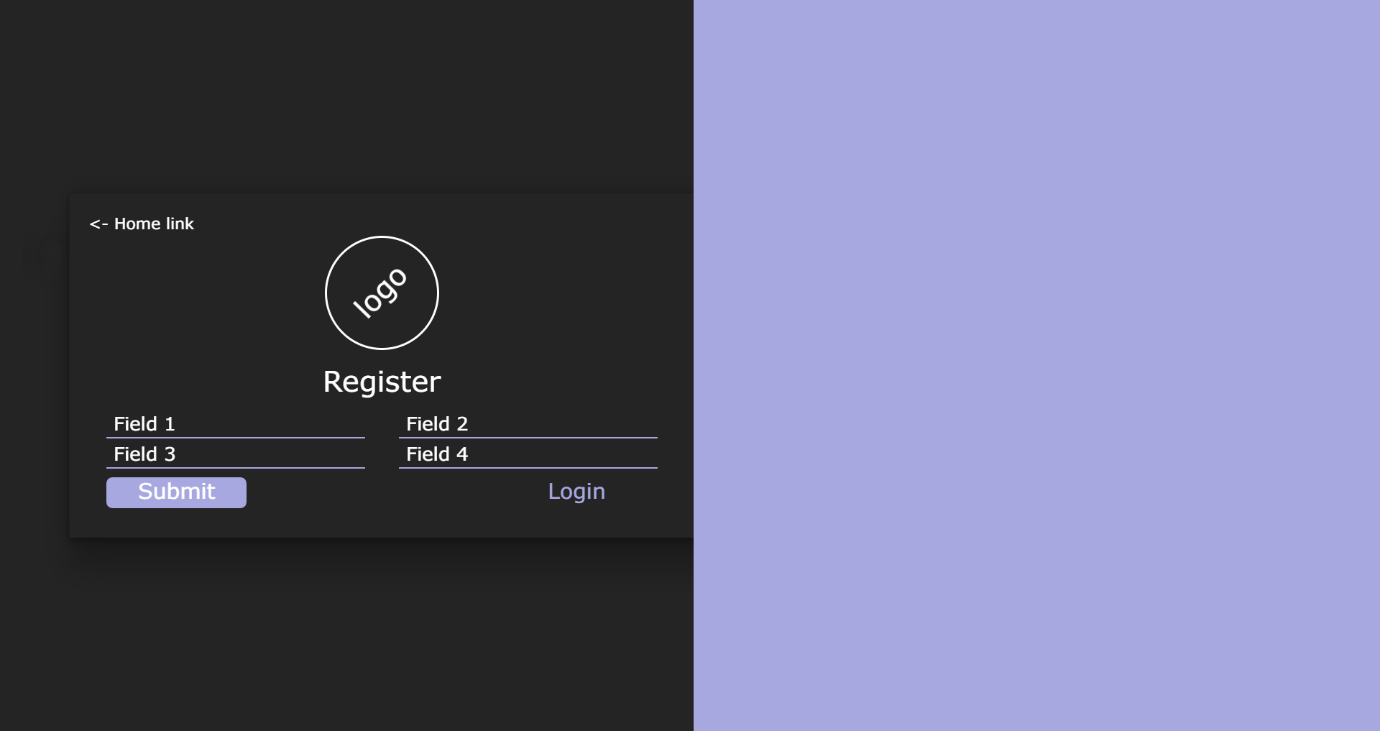
En ce qui concerne mon login et register, j’ai décidé de reprendre le design de ma page détail et de l’adapter en y ajoutant les formulaires nécessaires. Etant donné que la page telle quelle faisait vide, j’ai décidé qu’au lieu d’avoir une page login et une page register distinctes, je combinerai les deux. Ainsi, si on clique sur le lien « register », la partie violette se déplacera vers la droite pour cacher le formulaire de login et faire apparaître celui de register.

Figure 6 : Maquettes de la page auth (partie register)

Figure 5 : Maquettes de la page auth (partie login)

Pour la page home, il n’y aura qu’un simple listing des paires de chaussures. Il y aura également un select tout en haut pour filtrer par marques comme ceci :

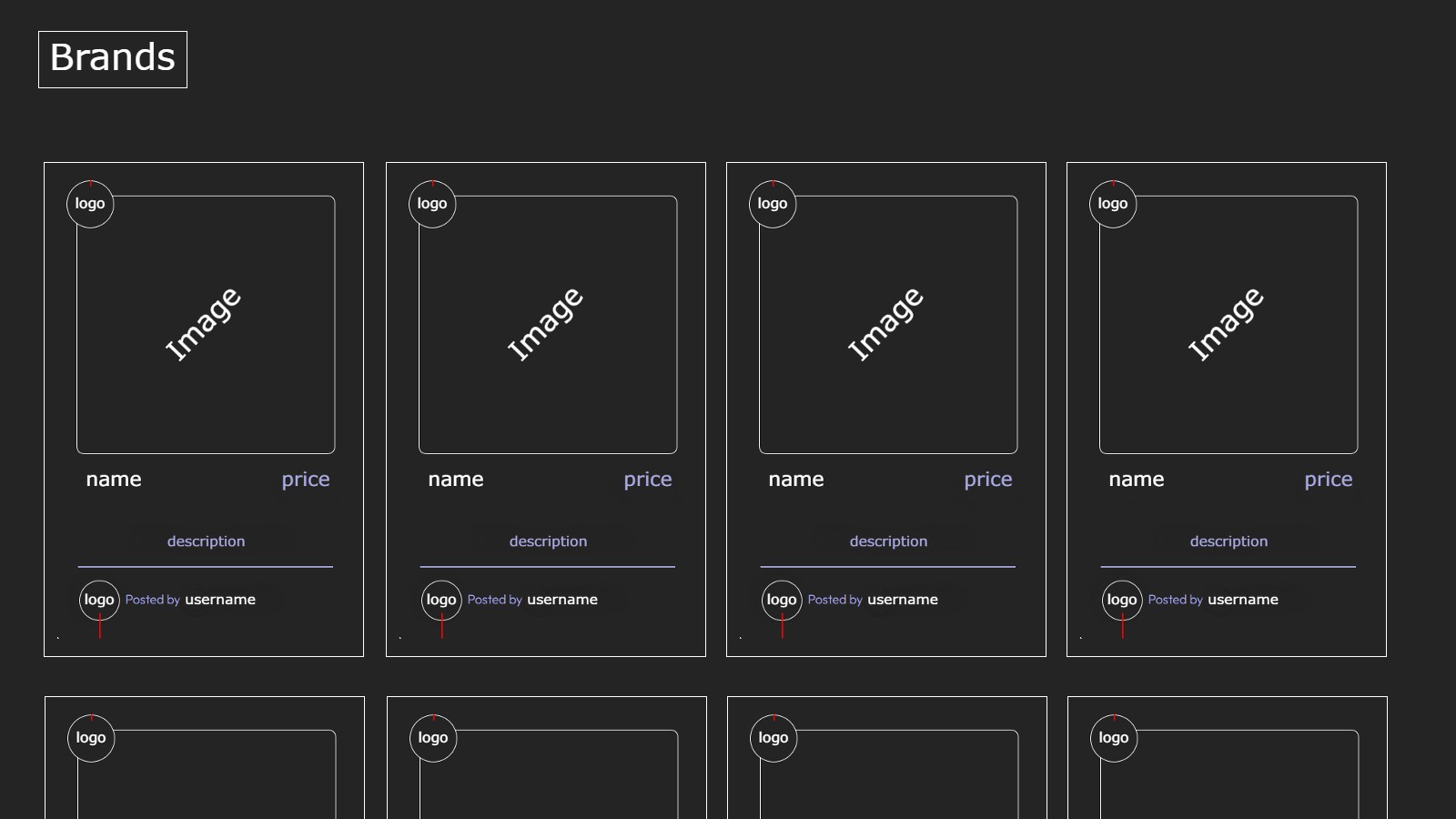


Figure 7 : Maquette de la page home

La page profile sera similaire à celle de connexion/inscription, mais en plus de cela, je rajouterai une preview pour que l’utilisateur voie l’aperçu de son profil.

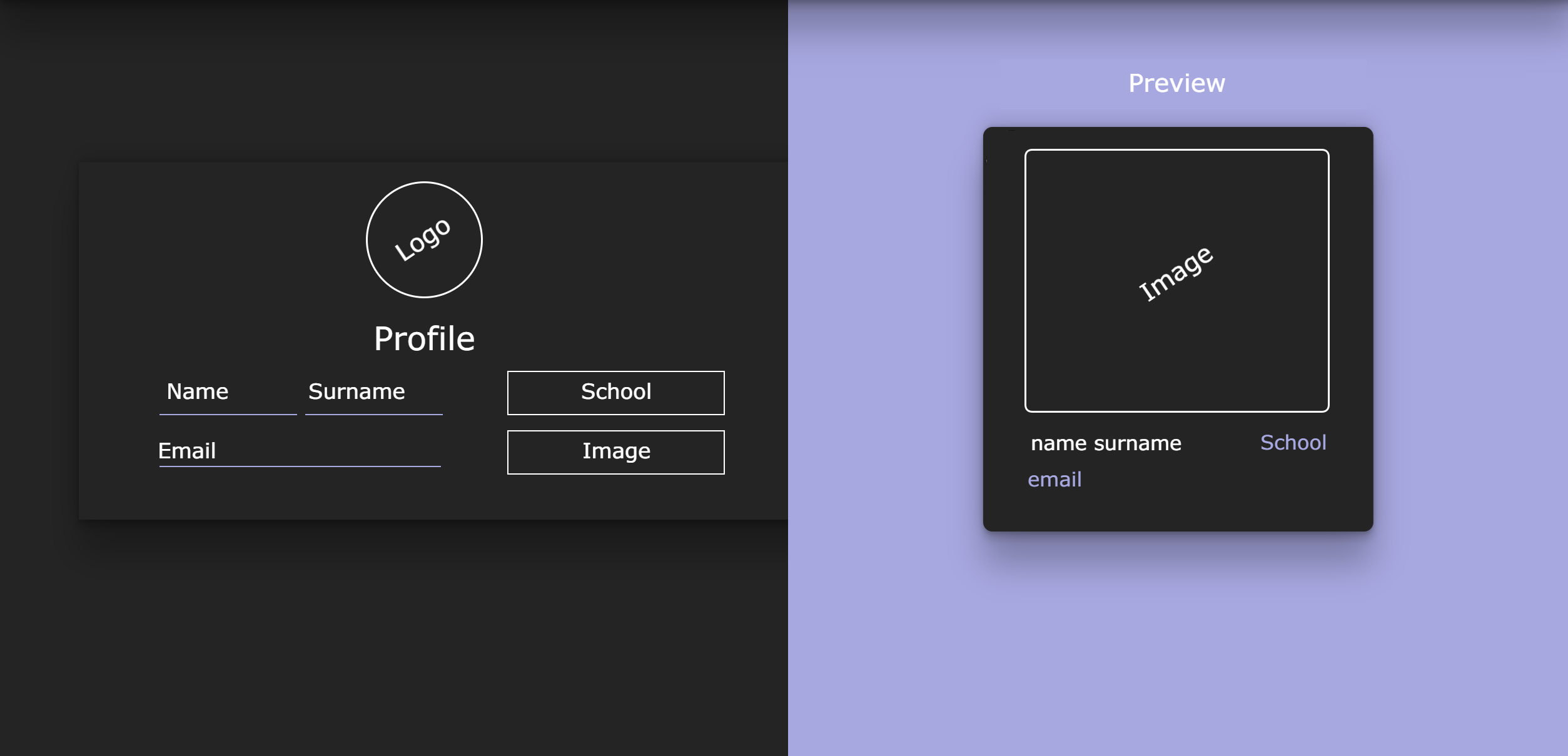
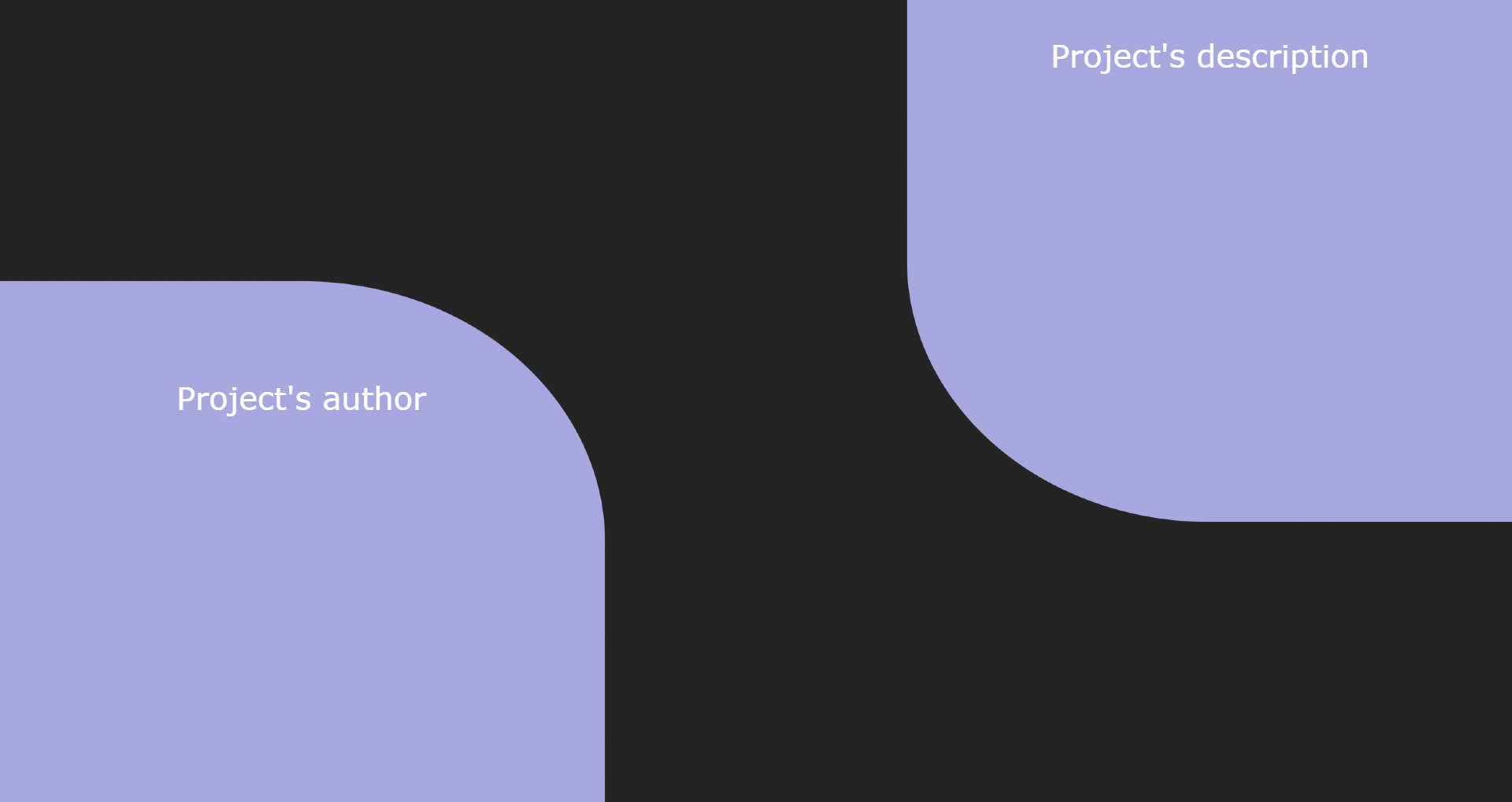


Figure 8: Maquette de la page profil

Ma page à propos se découperait en deux parties. La première étant simplement le nom du projet pour faire page de présentation, et la seconde, accessible en scrollant, contient la description du projet ainsi que l’auteur du projet. Cette page n’étant pas demandée dans le cahier des charges, je ne la ferai que si le temps me le permet. Cette page est utile pour faciliter la tâche des personnes ne connaissant pas le site, qu’ils puissent directement lire de quoi ce dernier traite.

Figure 10 : Maquettes de la page à propos (partie 2)

Figure 9 : Maquettes de la page à propos (partie 1)



# Analyse organique

## Modèle conceptuel de données

Voici le modèle conceptuel de données (MCD) de SneakersMarket. Elle a été réalisée en 3ème forme normale et comporte 14 tables, dont 2 de liaisons.

Ce dernier a beaucoup évolué au fil du temps. La première version était en français, mais afin de garder une uniformité de la langue entre le code et la base de données, je l’ai passé en anglais.

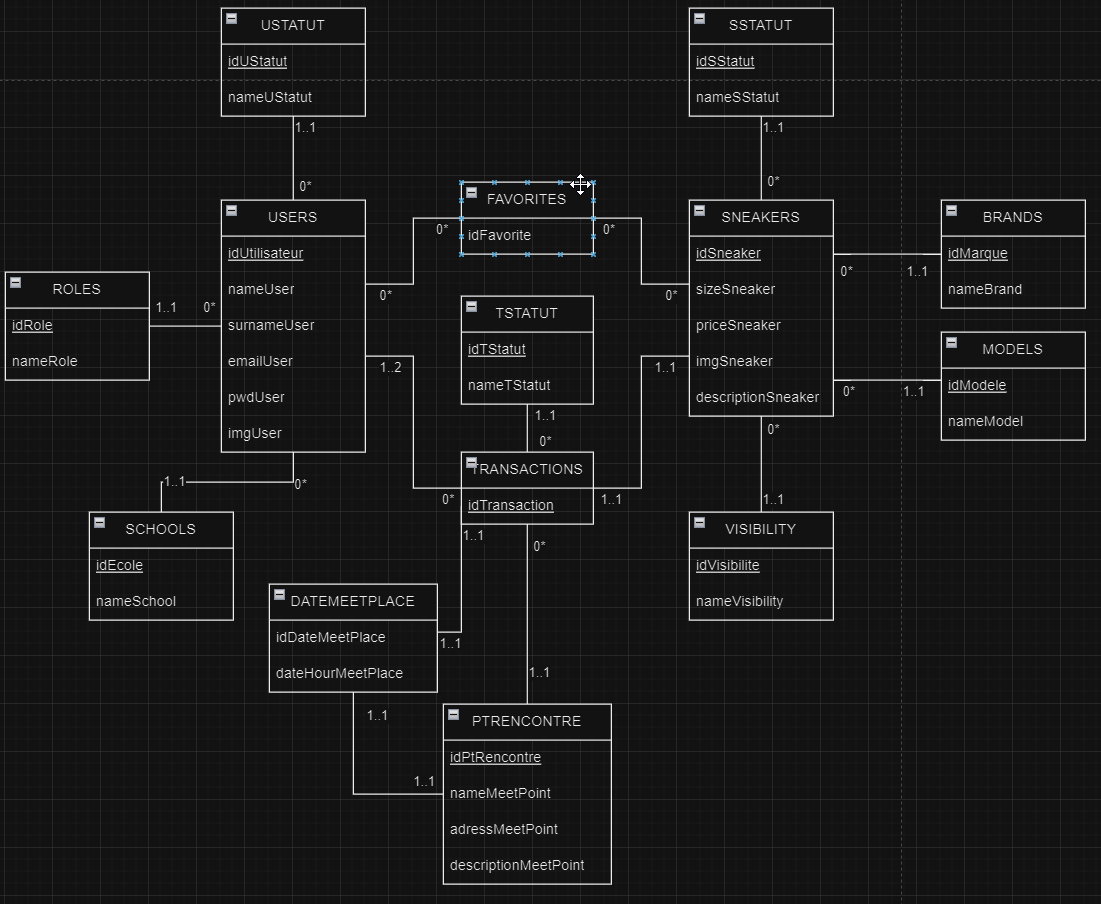


Figure 11 : Modèle conceptuel de données de SneakersMarket

## Modèle logique de données

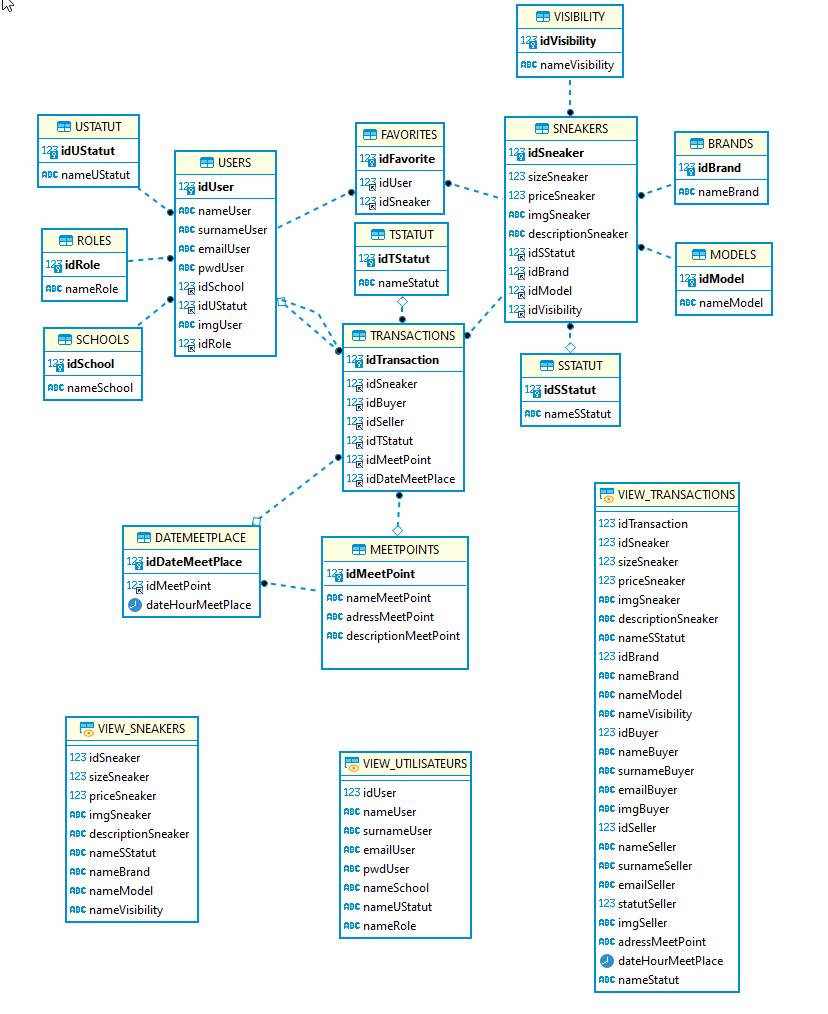


Figure 12 : Modèle logique de données de SneakersMarket

## Dictionnaire de données

### USERS

Table stockant les utilisateurs qui s’inscrivent sur le site

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nom | Type | NULL ? | Valeur par défaut | Extra |
| idUser | Int | Non | / | PRIMARY  AUTO INCREMENT |
| nameUser | Varchar | Non | / | / |
| surnameUser | Varchar | Non | / | / |
| emailUser | Varchar | Non | / | / |
| pwdUser | Varchar | Non | / | / |
| idSchool | Int | Non | / | FOREIGN KEY |
| IdUStatut | Int | Non | 1 | FOREIGN KEY |

### SNEAKERS

Table stockant les paires de chaussures du site

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nom | Type | NULL ? | Valeur par défaut | Extra |
| idSneaker | Int | Non | / | PRIMARY  AUTO INCREMENT |
| sizeSneaker | Int | Non | / | / |
| priceSneaker | Double | Non | 0.0 | / |
| imgSneaker | Longtext | Non | « notFound » | / |
| descriptionSneaker | Varchar | Non | / | / |
| idSStatut | Int | Non | / | FOREIGN KEY |
| idBrand | Int | Non | / | FOREIGN KEY |
| idModel | Int | Non | / | FOREIGN KEY |
| idVisibility | Int | Non | / | FOREIGN KEY |

### MEETPOINT

Table stockant les points de rencontres

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nom | Type | NULL ? | Valeur par défaut | Extra |
| IdMeetPoint | Int | Non | / | PRIMARY  AUTO INCREMENT |
| nameMeetPoint | Varchar | Non | / | / |
| adressMeetPoint | Varchar | Non | / | / |
| descriptionMeetPoint | Varchar | Non | / | / |

### TRANSACTIONS

Table associative stockant les transactions entre utilisateurs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nom | Type | NULL ? | Valeur par défaut | Extra |
| idTransaction | Int | Non | / | PRIMARY  AUTO INCREMENT |
| idSneaker | Int | Non | / | FOREIGN KEY |
| idBuyer | Int | Oui | NULL | FOREIGN KEY |
| idSeller | Int | Non | / | FOREIGN KEY |
| idTStatut | Int | Non | / | FOREIGN KEY |
| idMeetPoint | Int | Non | / | FOREIGN KEY |
| idDateMeetPlace | Int | Oui | NULL | FOREIGN KEY |

### FAVORITE

Table associative les sneakers favoris des utilisateurs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nom | Type | NULL ? | Valeur par défaut | Extra |
| IdFavorite | Int | Non | / | PRIMARY  AUTO INCREMENT |
| IdUser | Int | Non | / | FOREIGN KEY |
| idSneaker | Int | Non | / | FOREIGN KEY |

### BRANDS

Table stockant les marques des Sneakers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nom | Type | NULL ? | Valeur par défaut | Extra |
| IdBrand | Int | Non | / | PRIMARY  AUTO INCREMENT |
| nameBrand | Varchar | Non | / | / |

### SSTATUT

Table stockant les statuts des sneakers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nom | Type | NULL ? | Valeur par défaut | Extra |
| IdSStatut | Int | Non | / | PRIMARY  AUTO INCREMENT |
| nameSStatut | Varchar | Non | / | / |

### TSTATUT

Table stockant les statuts des transactions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nom | Type | NULL ? | Valeur par défaut | Extra |
| IdTStatut | Int | Non | / | PRIMARY  AUTO INCREMENT |
| nameTStatut | Varchar | Non | / | / |

### USTATUT

Table stockant les statuts des utilisateurs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nom | Type | NULL ? | Valeur par défaut | Extra |
| IdUStatut | Int | Non | / | PRIMARY  AUTO INCREMENT |
| nameUStatut | Varchar | Non | / | / |

### MODELS

Table stockant les modèles des sneakers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nom | Type | NULL ? | Valeur par défaut | Extra |
| IdModel | Int | Non | / | PRIMARY  AUTO INCREMENT |
| nameModel | Varchar | Non | / | / |

### ECOLE

Table stockant les écoles des utilisateurs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nom | Type | NULL ? | Valeur par défaut | Extra |
| idEcole | Int | Non | / | PRIMARY  AUTO INCREMENT |
| nomEcole | Varchar | Non | / | / |

### VISIBILITY

Table stockant la visibilité des utilisateurs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nom | Type | NULL ? | Valeur par défaut | Extra |
| idVisibility | Int | Non | / | PRIMARY  AUTO INCREMENT |
| nameVisibility | Varchar | Non | / | / |

### ROLES

Table stockant le rôle des utilisateurs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nom | Type | NULL ? | Valeur par défaut | Extra |
| idRole | Int | Non | / | PRIMARY  AUTO INCREMENT |
| nameRole | Varchar | Non | / | / |

### DATEMEETPLACE

Table stockant la date pour chaque point de rendez-vous

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nom | Type | NULL ? | Valeur par défaut | Extra |
| idDateMeetPlace | Int | Non | / | PRIMARY  AUTO INCREMENT |
| idMeetPoint | Int | Non | / | FOREIGN KEY |
| dateHourMeetPlace | Datetime | Non | / | / |

## Structure du projet

J’ai décidé d’adopter la structure MVC pour l’ensemble de mon projet.

### Modèle

La partie modèle contient la connexion à la base en PDO, les objets utilisés ainsi que toutes les requêtes, que ce soit insertion, modification, suppression et sélection. Chaque page possède sa classe avec les fonctions qui lui sont propres.

### Vue

La partie vue concerne uniquement ce qui sera visible sur le site, c’est-à-dire toute la partie HTML-CSS. Aucune logique n’est faite dans ces pages. Les seuls moments où on y met du PHP, c’est pour afficher la valeur d’une variable, faire une boucle pour afficher certains éléments ou faire une condition pour afficher ou ne pas afficher certains éléments.

### Contrôleur

Il y a un contrôleur par page. Le contrôleur sert à faire la liaison entre l’affichage et la logique, ainsi que pour récupérer les données passées en formulaires.

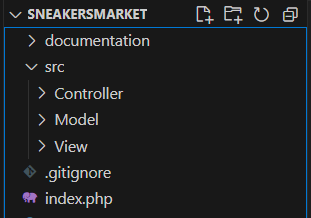


Figure 13 : Arborescence du projet

## Description des méthodes et classes principales

### Classe Home

Cette classe est dédiée à la page home. Elle gère le listing des paires ainsi que les différents filtres disponibles.

**GetAllShoesCard**

Cette méthode permet de récupérer les informations de toutes les paires afin de les afficher sous formes de cartes dans la page Home. Elle a comme paramètres :

* idSeller qui correspond à l’id de l’utilisateur ayant posté la paire. On n’affiche pas la paire si le vendeur est l’utilisateur connecté.
* statutSeller qui doit être égal à 1 pour que la paire soit sélectionnée. 1 correspond à « Actif » dans la base
* indexFirst qui correspond à l’index de la première paire sélectionnée pour la pagination
* indexLast qui correspond à l’index de la dernière paire sélectionnée pour la pagination.

**FilterByBrand**

Cette méthode sert à récupérer les Sneakers correspondant à la marque souhaitée. Elle a comme paramètres :

* idBrand qui correspond à l’id de la marque stockée dans la base
* indexFirst qui correspond à l’index de la première paire sélectionnée pour la pagination
* indexLast qui correspond à l’index de la dernière paire sélectionnée pour la pagination.

### Classe Gestion

Cette méthode est dédiée à la page gestion. Elle gère l’ajout d’une paire, les réservations et les achats.

**CreateTransaction**

Cette méthode ajoute la paire que l’utilisateur et par extension la transaction qu’il souhaite vendre en base si toutes les informations sont correctes. Dans un premier temps, elle vérifie si le modèle entré existe dans la base, si c’est le cas, il récupère son id, sinon, elle l’insère en base et récupère son id. Ensuite, elle insère la paire en base et pour finir récupère l’id de cette dernière pour créer la transaction. Elle a comme paramètres :

* size qui correspond à la taille de la paire
* price qui correspond au prix de la paire
* img qui correspond à l’image de la paire
* nameModel qui correspond au modèle entré
* idBrand qui correspond à la marque sélectionnée

### Classe Admin

Cette classe est dédiée à la page admin. la classe admin dispose de fonctions plutôt basiques pour gérer les différents CRUD des utilisateurs, paires, établissements et points de rencontre.

**BlockUser**

Cette méthode change le statut de l’utilisateur concerné. Il passe de « Active » à « Blocked ». Elle a comme paramètre l’id de l’utilisateur à changer.

### Classe Profil

**UpdateProfile**

Cette méthode met à jour les données dans la base de données de l’utilisateur en fonction de ce qu’il a modifié sur la page profil. Elle commence par vérifier les données, puis change les données dans la base. Elle a comme paramètres :

* user qui correspond à un objet User contenant toutes les informations de l’utilisateur
* img qui correspond à la potentielle image passée, s’il l’image est vide, l’image en base ne change pas

## Analyse des fonctionnalités majeures

### Mettre une paire en vente

Pour un utilisateur souhaitant devenir vendeur, mettre en vente une paire est la fonctionnalité essentielle. Disponible dans la page gestion, pour pouvoir mettre en vente une paire, l’utilisateur doit renseigner une image, la marque, le modèle, le prix, la taille et la description de cette dernière. Une fois les informations correctes et validées, la paire est disponible à la vente pour tous les autres utilisateurs.

Figure 15 : Méthode pour mettre une paire en vente

### Réserver une paire

Sur la page détail, un formulaire de réservation est disponible pour l’utilisateur. S’il y renseigne correctement le point de rendez-vous, la date et l’heure souhaitée, la réservation est transmise sur la page gestion du vendeur, et c’est à lui de décider s’il accepte ou non l’offre.



Figure 14 : Méthode pour réserver une paire

### Ajouter une paire en favoris

Également sur la page détail, une petite étoile figure à côté du bouton réserver. Si l’utilisateur clique dessus, la paire est ajoutée à sa liste de favoris. Si l’utilisateur a déjà la paire en favoris et clique à nouveau sur l’étoile, la paire est enlevée de la liste des favoris.

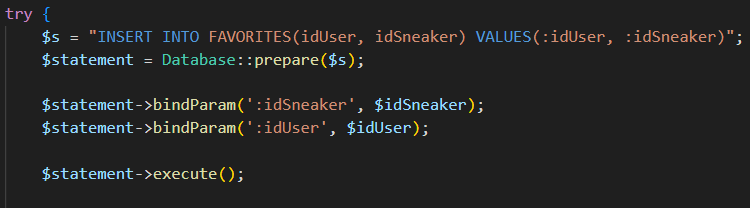


Figure 16 : Méthode pour mettre une paire en favoris

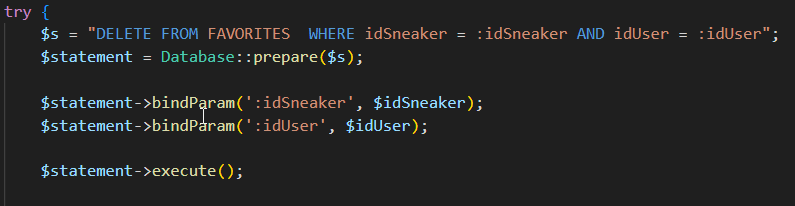


Figure 17 : Méthode pour enlever une paire des favoris

# Programmes utilisés

## Visual Studio Code icon PNG and SVG Vector Free DownloadVisual Studio Code – 1.88

Visual Studio Code est un éditeur de code développé par Microsoft. J’ai utilisé ce dernier pour écrire et corriger tout le code présent dans mon travail.

## Fichier:DBeaver logo.svg — WikipédiaDBeaver – 23.1.5

DBeaver est un logiciel qui m’a servi à gérer la base de données du projet, que ce soit l’insertion des données, la création des tables ou les requêtes.

## Gitlab Logo PNG Vector (SVG) Free DownloadGitlab – 16.10.3

Gitlab est un logiciel intégré à Git. Il m’a servi pour le suivi des versions de mon projet.

## Copilot icon in SVG, PNG formatsCopilot – Free

Copilot est un chatbot développé par Microsoft, qui permet, à l’image de ChatGPT, de converser et créer des images générées artificiellement. Je l’ai utilisé pour la création de toutes mes images sur le site.

## Pixlr Logo - Free DownloadPixlr – Free

Pixlr est un outil de retouche photo possédant une version gratuite se limitant à 3 retouches par jour. Il m’a servi à modifier mon logo comme je le souhaitais.

## Remove.bg – Free

Remove.bg est également un outil de retouche photo, mais permettant d’automatiquement retirer le fond des images, permettant ainsi de les rendre transparentes. Je l’ai utilisé afin de créer une deuxième version aux paires de chaussures, que j’affiche ensuite dans ma page détail.

## Codepen - Free

Codepen est un site internet qui regroupe les projets et exemples programmés de personnes du monde entier. Il m’a été utile pour me donner des idées de design ou pour m’aider à comprendre comment réaliser la disposition des éléments que je souhaitais.

## Canva Austin Office: Careers, Perks + Culture | Built In AustinCanva – Free

Canva est un site internet de design. Je l’ai utilisé pour créer les différents designs des pages de couvertures présentes sur ce document et le manuel utilisateur.

# Tests

## Plan de tests

Utilisateur Anonyme : A accès à la page d’accueil et la page détail, mais ne peut pas réserver.

Utilisateur Connecté : A accès à toutes les fonctionnalités sauf celles présentes sur Admin.

Administrateur : A accès à la page admin en plus.

|  |  |
| --- | --- |
| ID TEST | 1 |
| USER STORY | En tant qu’utilisateur, je peux créer une compte afin de profiter pleinement des fonctionnalités du site |
| SITUATION | L'utilisateur entre les informations correctement, et le compte se crée |
| RESULTAT OBTENU | L'inscription se fait correctement, il y a les bonnes données en session et l'utilisateur est redirigé sur home |
| STATUT | OK |

|  |  |
| --- | --- |
| ID TEST | 2 |
| USER STORY | En tant qu’utilisateur, je peux me connecter au compte que j’ai créé afin de continuer de profiter du site |
| SITUATION | L'utilisateur entre les informations correctement, et la connexion est autorisée |
| RESULTAT OBTENU | La connexion se fait correctement, les données en session sont correctes |
| STATUT | OK |

|  |  |
| --- | --- |
| ID TEST | 3 |
| USER STORY | En tant qu’utilisateur, je peux me connecter au compte que j’ai créé afin de continuer de profiter du site |
| SITUATION | L'utilisateur entre les informations correctement, et la connexion est autorisée |
| RESULTAT OBTENU | La connexion se fait correctement, les données en session sont correctes |
| STATUT | OK |

|  |  |
| --- | --- |
| ID TEST | 4 |
| USER STORY | En tant qu’utilisateur, je peux mettre en vente une paire de chaussures depuis ma page de gestion afin que les autres utilisateurs puissent la réserver |
| SITUATION | L'utilisateur rentre les bonnes informations et la paire apparaît dans la page home chez les autres utilisateurs |
| RESULTAT OBTENU | La paire est correctement ajoutée en base, la paire apparaît dans la page home des autres utilisateurs |
| STATUT | OK |

|  |  |
| --- | --- |
| ID TEST | 5 |
| USER STORY | En tant qu’utilisateur, je peux accéder à ma page de profil afin de modifier les informations de mon compte |
| SITUATION | L'utilisateur change les informations de son compte, les informations en base sont changées |
| RESULTAT OBTENU | Les informations sont correctement changées |
| STATUT | OK |

|  |  |
| --- | --- |
| ID TEST | 6 |
| USER STORY | En tant qu’utilisateur, je peux accéder à ma page de gestion afin de voir et de gérer les chaussures que j’ai achetées et que j’ai à vendre |
| SITUATION | L'utilisateur voit la liste des paires achetées et à vendre |
| RESULTAT OBTENU | Les listes sont complètes et correctes |
| STATUT | OK |

|  |  |
| --- | --- |
| ID TEST | 7 |
| USER STORY | En tant qu’administrateur, je peux accéder à ma page admin afin de gérer les utilisateurs, les points de rencontre et les établissements |
| SITUATION | L'administrateur peut bloquer et débloquer un utilisateur |
| RESULTAT OBTENU | Le blocage s'effectue correctement et l'utilisateur n'apparait plus en tant que vendeur |
| STATUT | OK |

|  |  |
| --- | --- |
| ID TEST | 8 |
| USER STORY | En tant qu’administrateur, je peux accéder à ma page admin afin de gérer les utilisateurs, les points de rencontre et les établissements |
| SITUATION | L'administrateur peut modifier et ajouter un point de rencontre |
| RESULTAT OBTENU | La modification / L'ajout s'effectue correctement |
| STATUT | OK |

|  |  |
| --- | --- |
| ID TEST | 9 |
| USER STORY | En tant qu’administrateur, je peux accéder à ma page admin afin de gérer les utilisateurs, les points de rencontre et les établissements |
| SITUATION | L'administrateur peut modifier et ajouter un établissement |
| RESULTAT OBTENU | La modification / L'ajout s'effectue correctement |
| STATUT | OK |

|  |  |
| --- | --- |
| ID TEST | 10 |
| USER STORY | En tant qu’administrateur, je peux gérer les différents utilisateurs ainsi que les paires qu’ils ont achetées et vendues afin de garantir le bon fonctionnement du site |
| SITUATION | L'administrateur peut modifier les paires à vendre d'un utilisateur |
| RESULTAT OBTENU | La modification s'effectue correctement |
| STATUT | OK |

|  |  |
| --- | --- |
| ID TEST | 11 |
| USER STORY | En tant qu’utilisateur, je peux rechercher une paire que je souhaite afin de la trouver plus facilement |
| SITUATION | L'utilisateur cherche la paire grâce à la barre de recherche |
| RESULTAT OBTENU | La/Les paire(s) cherchée(s) est/sont trouvée(s) et uniquement elle(s) est/sont affichée(s) |
| STATUT | OK |

|  |  |
| --- | --- |
| ID TEST | 12 |
| USER STORY | En tant qu’utilisateur, je ne peux pas créer un compte avec un email déjà utilisé |
| SITUATION | L'utilisateur tente de créer un compte avec un email déjà existant |
| RESULTAT OBTENU | L'inscription ne se fait pas et un message d'erreur apparait |
| STATUT | OK |

|  |  |
| --- | --- |
| ID TEST | 13 |
| USER STORY | En tant qu’utilisateur, je ne peux pas réserver une paire ayant déjà été réservée |
| SITUATION | L'utilisateur recherche une paire déjà réservée par une autre personne |
| RESULTAT OBTENU | La paire n'apparait pas |
| STATUT | OK |

|  |  |
| --- | --- |
| ID TEST | 14 |
| USER STORY | En tant qu’utilisateur, mon mot de passe doit contenir au moins 8 caractères, un chiffre, une lettre majuscule et un caractère spécial afin de pouvoir créer mon compte |
| SITUATION | L'utilisateur tente de rentrer un mot de passe qui ne correspond pas à la sécurité demandée |
| RESULTAT OBTENU | L'inscription ne se fait pas et un message d'erreur apparait |
| STATUT | OK |

|  |  |
| --- | --- |
| ID TEST | 15 |
| USER STORY | En tant qu’utilisateur, je ne peux plus vendre de paires si mon statut est bloqué afin de préserver la sécurité des acheteurs |
| SITUATION | L'utilisateur tente de vendre une paire en étant bloqué |
| RESULTAT OBTENU | La catégorie n'apparait pas dans la page de gestion |
| STATUT | OK |

|  |  |
| --- | --- |
| ID TEST | 16 |
| USER STORY | En tant qu’utilisateur, je peux de nouveau réserver une paire qui n’a plus le statut réservé |
| SITUATION | L'utilisateur ayant réservé la paire annule la réservation |
| RESULTAT OBTENU | un autre utilisateur cherche la paire et la trouve |
| STATUT | OK |

|  |  |
| --- | --- |
| ID TEST | 17 |
| USER STORY | En tant qu’utilisateur, je peux filtrer les paires par marque afin de mieux choisir la paire que je souhaite |
| SITUATION | L'utilisateur cherche la marque souhaitée |
| RESULTAT OBTENU | Les paires uniquement de cette marque apparaissent |
| STATUT | OK |

|  |  |
| --- | --- |
| ID TEST | 18 |
| USER STORY | En tant qu’administrateur, je ne peux pas supprimer un utilisateur qui a déjà effectué des ventes afin de préserver une trace des actions |
| SITUATION | L'administrateur veut supprimer un vendeur ayant déjà vendu une ou plusieurs paires |
| RESULTAT OBTENU | L'utilisateur est bloqué à la place |
| STATUT | OK |

## Déroulement des tests

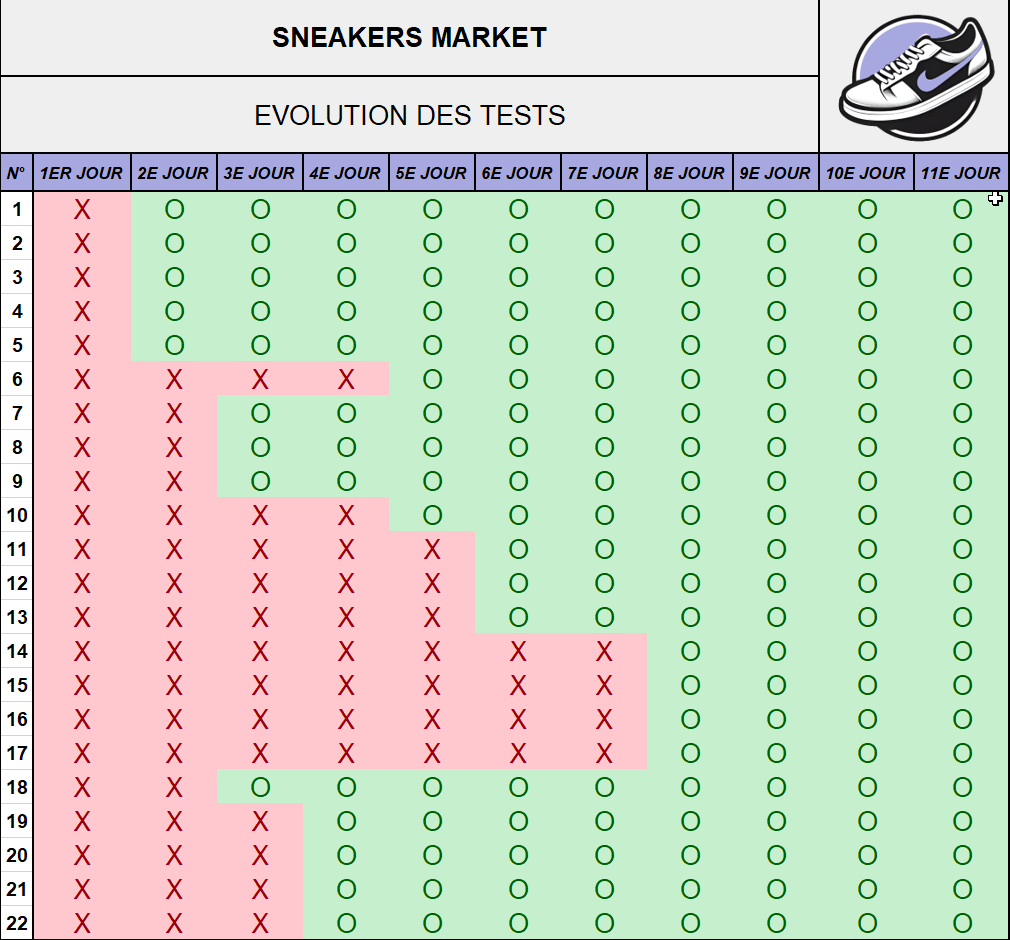
Voici le déroulement de tous mes tests. Comme on peut le voir, la majorité des tests étaient opérationnels dès le 8e jour, ce qui correspond à la fin du développement de la dernière page qui est la page administrateur.

Figure 17 : Évolution des tests de SneakersMarket

# Planning

## Planning prévisionnel

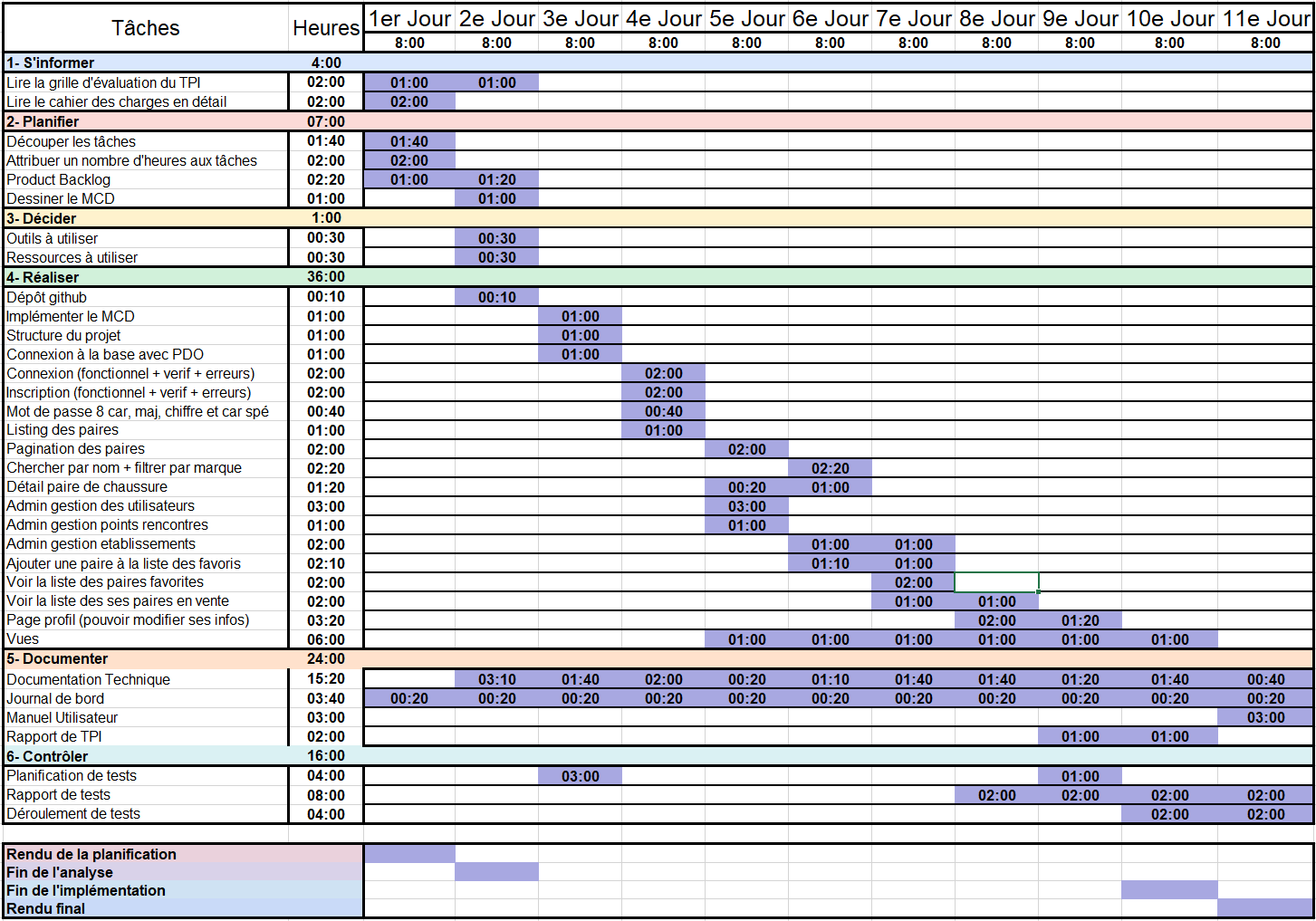


Figure 18 : Planning prévisionnel de SneakersMarket

## Planning effectif

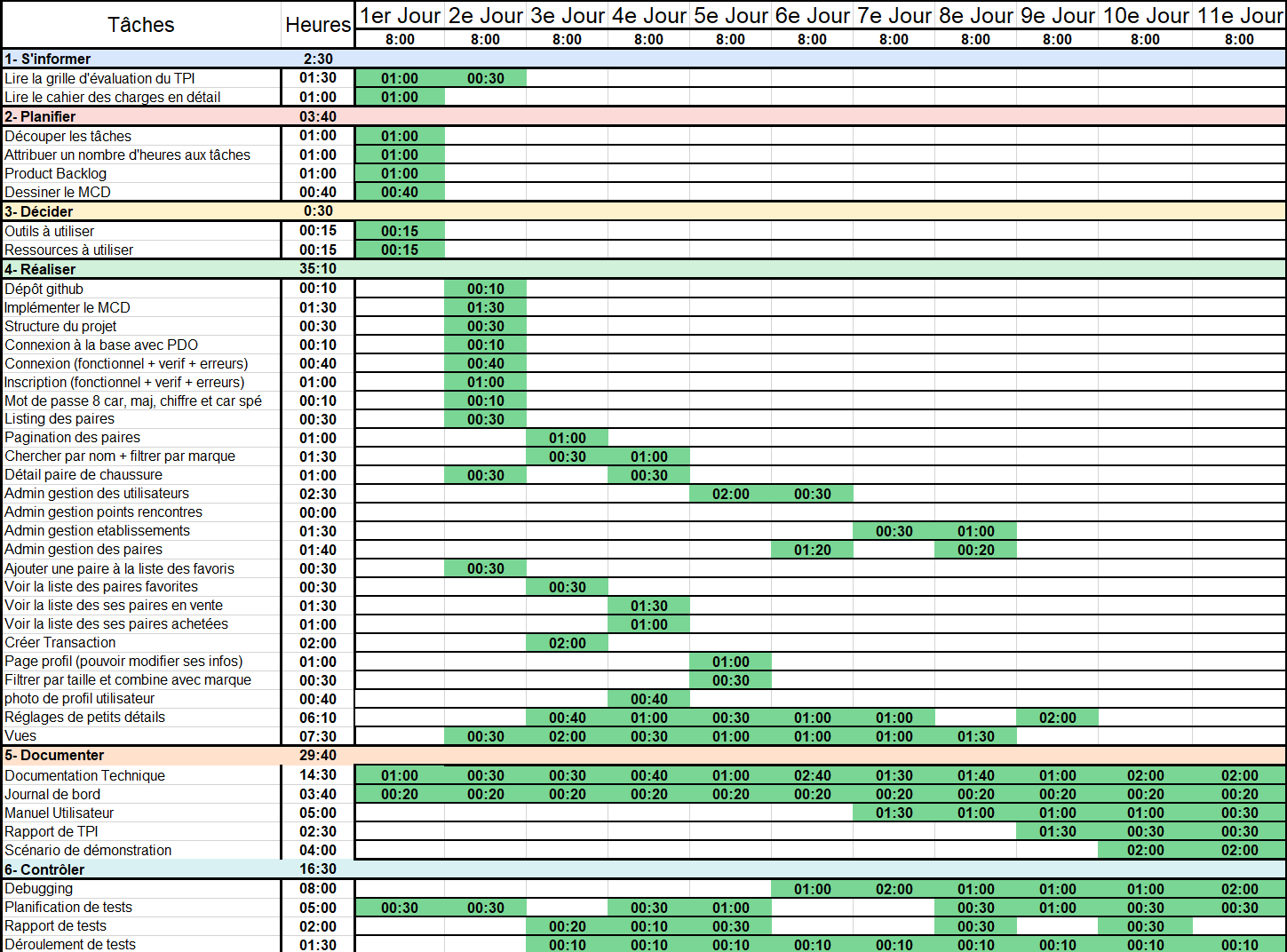
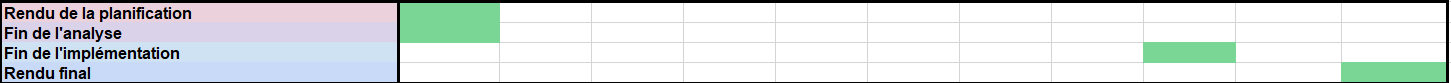


Figure 19 : Planning effectif de SneakersMarket

## Analyse de similitudes entre les deux plannings

Comme nous pouvons le voir, il y a des divergences entre le planning prévisionnel et le planning effectif au niveau du point 4 « Réaliser ». En effet, sur le planning prévisionnel, il avait été convenu que la partie admin soit faite beaucoup plus tôt que ce qui a été réalisé. En revanche, toutes les autres fonctionnalités ont été réalisées avant ces dernières.

Ce qui diverge aussi, ce sont le nombre de tâches. Pendant la réalisation du projet, je me suis rendu compte que certaines tâches manquaient comme le scénario de démonstration. Je les ai donc ajoutées dans le planning effectif.

Les jalons ont été bien respectés, toutes les tâches significatives ont été réalisées dans les temps.

# Conclusion

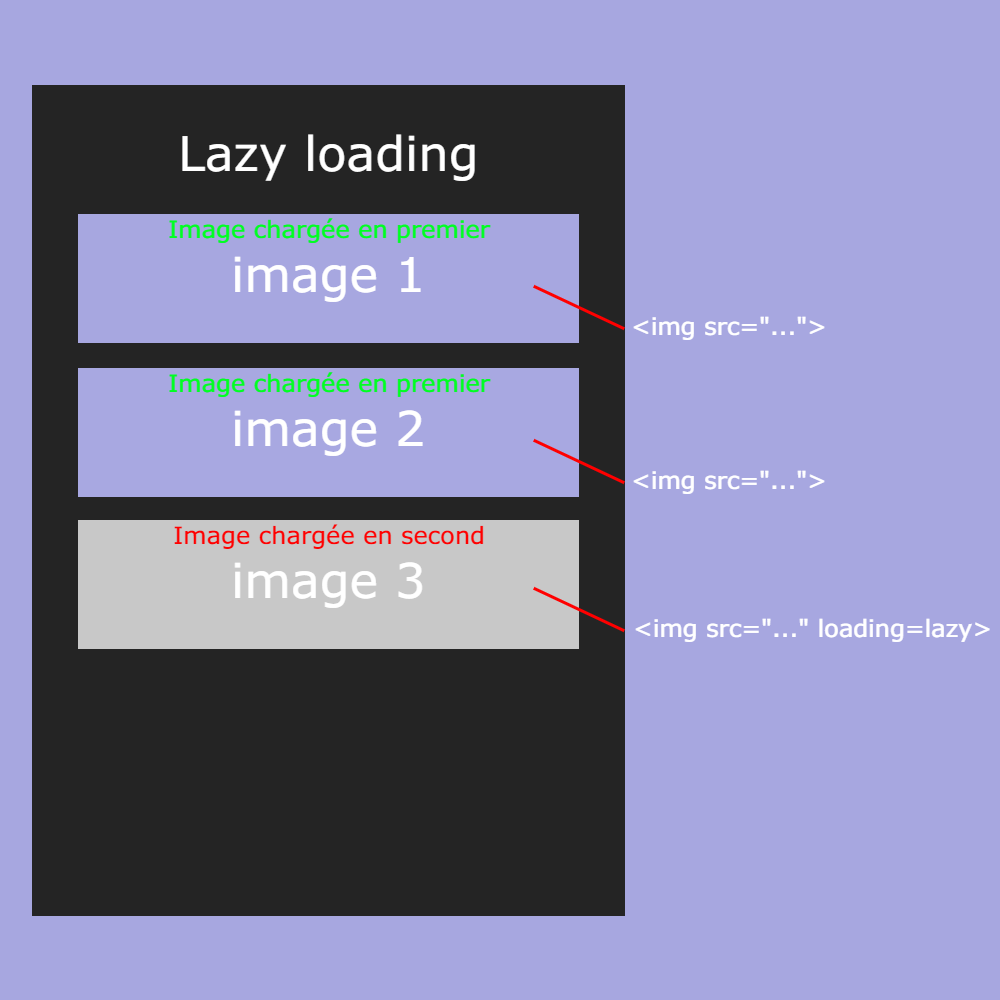
## Difficultés rencontrées

Le plus gros problème auquel j’ai été confronté lors de la réalisation de ce projet, est sans doute le fait de travailler et retravailler mon MCD et par extension ma base de données en continu. J’avais dès le départ oublié certaines tables qui étaient essentielles à quelques fonctionnalités, et le fait de devoir la modifier en continu m’obligeait à adapter le code avec les nouvelles modifications.

## Améliorations possibles

La plus grosse amélioration possible que j’ai pu penser dans la réalisation de ce projet, est sans doute l’optimisation de la page « Shop ». Ayant des images relativement lourdes, optimiser cela aurait été un bon point à ce projet, j’ai donc pensé à cette potentielle solution :

**Lazy loading**

Le lazy loading consiste à retarder volontairement le chargement de certaines ressources présentes sur la page, afin de laisser la place aux ressources plus importantes. Cette méthode fonctionnait bien, mais le problème avec cette méthode est qu’on remarquait que le carousel se chargeait bien après le reste de la page, ce qui ne rendait esthétiquement pas parfait.

Devant consacrer mon temps à la réalisation des autres fonctionnalités, j’ai décidé de passer outre cette gêne. Dans une version future, je me pencherai sur cela, et essaierai de l’améliorer.

Figure 20 : schéma explicatif du lazy loading

Une autre amélioration possible serait de pouvoir vérifier l’adresse email des utilisateurs quand ils créent leur compte, en vérifiant si l’adresse email correspond bien à l’école souhaitée.

## Bilan personnel

J'ai été ravi de travailler sur un projet aligné avec l'un de mes centres d'intérêt et d'avoir eu le maître d'apprentissage que je souhaitais. Bien que le planning prévisionnel n'ait pas été strictement suivi, j'ai finalement créé un site web répondant à toutes les exigences, tant celles du cahier des charges que les miennes. Je suis, dans l’ensemble, satisfait du travail que j’ai produit.

# Bibliographie

**Logos**

**Visual Studio Code**

<https://uxwing.com/visual-studio-code-icon/>

**DBeaver**

<https://fr.wikipedia.org/wiki/Fichier:DBeaver_logo.svg>

**Gitlab**

<https://seeklogo.com/vector-logo/273186/gitlab>

**Copilot**

<https://uxwing.com/copilot-icon/>

**Pixlr**

<https://in.pinterest.com/pin/pixlr-logo-free-download--268456827778839153/>

**Remove. bg**

<https://marjory.io/partenaires/remove-bg/>

**Codepen**

<https://seeklogo.com/vector-logo/328201/codepen>

**Canva**

<https://www.builtinaustin.com/company/canva>

**Images**

Voici l’outil que j’ai utilisé pour toutes les images figurantes sur le site. Copilot est une intelligence artificielle, j’ai utilisé sa fonctionnalité « Designer » pour générer les images dont j’avais besoin.

<https://copilot.microsoft.com>

**Détail des paires de chaussures**

Voici une autre template que j’ai utilisé pour le design. Comme pour les cartes, j’ai pris uniquement la disposition des éléments.

<https://codepen.io/JorgeAguilar/pen/poEZWKY>

**Formes qui bougent dans le login/register**

J’ai utilisé cette template pour combler la zone vide à gauche de mon login et register, j’ai uniquement utilisé le javascript qui permet de dessiner et bouger les formes géométriques

<https://codepen.io/m2creates/pen/EEvGgW>

**Pagination**

Ce tuto m’a permis de comprendre le fonctionnement d’une pagination et de la réaliser.  
<https://nouvelle-techno.fr/articles/mettre-en-place-une-pagination-en-php>

# Glossaire

|  |  |
| --- | --- |
| **MOT** | **DEFINITION** |
| TPI | Travail Pratique Individuel |
| Sneaker | Terme anglais pour les chaussures de sport ou urbaines |
| Product Backlog | Liste priorisée des tâches à réaliser pour le projet. |
| Prompt | Conversation avec une intelligence artificielle |
| CRUD | Représente les 4 actions de base pour interagir et gérer les données en base de données (créer, lire, mettre à jour et supprimer). |

# Annexes

Vous retrouverez en annexe sous format PDF :

* La Planification prévisionnelle et effective :
* Le Protocole et plan des tests :
* Les prompts générés par intelligence artificielle
* Le Code source