

Remerciements

Je tiens à remercier, mes formateurs et le pôle administratif de Philiance.

Sommaire

1. Liste des compétences du référentiel couvertes par le projet

- 1.1 Maquetter une application
- 1.2. Réaliser une interface utilisateur web statique et adaptable
- 1.3. Développer une interface utilisateur web dynamique
- 1.4. Réaliser une interface utilisateur avec une solution de gestion de contenu
- 1.5. Créer une base de données
- 1.6. Développer les composants d'accès aux données
- 1.7. Développer la partie back-end d'une application web ou web mobile
- 1.8. Elaborer et mettre en œuvre des composants dans une application de gestion de contenu

2. Résumé du projet

3. Cahier des charges

- 3.1. Présentation du projet
- 3.2. Expression du besoin
- 3.3. Étude de marché
- 3.4. Fonctions principales
- 3.5 Spécifications annexes
- 3.6 Rôles utilisateurs
- 3.7 Marché et population cible
- 3.8 Maquettage
- 3.9 Arborescence du site
- 3.10 Charte graphique
- 3.11 Planification des développements et délais

4- Spécifications techniques du projet

- 4.1. Choix technologique
- 4.2 Langages, librairies, framework et SGBD
- 4.3 Outilage
- 4.4 Architecture MV
- 4.5 Traitement des données
- 4.6 Travaux de référencement
- 4.7 Publication & déploiement

5- Réalisation et extraits de code

- 5.1. Front-end
- 5.2. Back-end
- 5.3. Fonctionnel

6-Tests de fonctionnalité réalisés et résultats

6.1. Tests utilisateurs

6.2. Tests d'intégrité

6.4. Tests d'injection

7- Description de la veille technologique sur les vulnérabilités de sécurité

8- Description d'une situation de travail ayant nécessité une recherche

9- Extrait de site en anglais

10- Evolution pour le futur

1. Liste des compétences du référentiel couvertes par le projet

1.1. Maquetter une application

J'ai commencé par faire les wireframes pour mobile et desktop avec Adobe XD. Ensuite, j'ai complété les parties manquantes du site (La maquette que j'ai utilisée n'est pas complète.), comme le menu déroulant du header pour accéder à l'espace utilisateur.

The image displays five wireframe prototypes for an audiophile website, arranged vertically. The first four prototypes represent different views of the website: a desktop homepage, a desktop product detail page, a mobile product detail page, and a desktop sidebar. The fifth prototype is a user profile page. Each wireframe includes placeholder text and buttons, such as 'NOUVEAU PRODUIT' and 'VOIR PRODUIT'. The bottom of the image features a dark background with a large headphones icon, suggesting the brand's focus on audio equipment.

1.2 Réaliser une interface utilisateur web statique et adaptable

À partir de la maquette, des wireframes et de la charte graphique, j'ai réalisé l'intégration en commençant par la structure html à l'aide du moteur de template Twig. Ensuite, je me suis occupé du style avec le css, j'ai utilisé sass pour optimiser l'écriture et l'organisation de mon code css. J'ai testé le site sur différents navigateurs et différentes tailles d'écrans pour m'assurer que l'intégration soit correcte sur l'ensemble des supports ciblés. Pour la partie responsive j'ai utilisé les requêtes média (media queries) pour adapter l'interface du site en fonction de la largeur de l'écran utilisé pour consulter le site. Et j'ai fait en sorte que le site respecte les normes d'accessibilité, de sécurités. J'ai également suis les recommandations pour le SEO.

1.3 Développer une interface utilisateur web dynamique

Pour la partie dynamique de l'interface utilisateur, j'ai utilisé le moteur de template Twig pour générer du html à la volée et j'ai installé le paquet MobileDetectBundle pour adapter l'interface en fonction du support utilisé pour consulter le site, par exemple avec MobileDetectBundle j'ai peu utilisé une version mobile et desktop des images pour certains produits.

1.4 Réaliser une interface utilisateur avec une solution de gestion de contenu

Pour permet à l'utilisateur de gérer ses informations personnelles, j'ai mis en place un espace utilisateur. Et pour la partie admin du site j'ai décidé d'utiliser le bundle EasyAdmin. Ce bundle permet de mettre en place rapidement et facilement un espace admin, il permet également de créer des CRUD pour chaque entités, et de restreindre l'accès aux actions du CRUD à l'aide d'un système de permission basé sur les rôles utilisateurs de Symfony.

1.5 Créer une base de données

Après avoir élaboré le MCD (Modèle Conceptuel des Données) et après avoir créé mes entités en me basant sur ce dernier, j'ai décidé d'utiliser Doctrine, l'ORM par défaut de Symfony pour l'accès et la gestion à la base de données du site.

Ensuite, j'ai choisi comme système de gestion de base données (SGBD) MariaDB, car il s'agit d'un fork de MySQL sous licence GPL v2 créer suite au rachat de MySQL par Oracle.

Pour la création de la base de données, j'ai exécuté la commande « doctrine:database:create » après avoir configuré au préalable la variable d'environnement « DATABASE_URL ».

Et ensuite, j'ai inséré un jeu de données pour avoir des données tests sur lesquelles travailler durant la phase de développement.

1.6 Développer les composants d'accès aux données

Pour la partie modèle du MVC j'ai utilisé Doctrine, un outil proposé nativement avec Symfony. Doctrine est un ORM, en d'autres termes une API qui permet de manipuler les données d'une base de données sous la forme d'objet. Doctrine fournit également un tas d'outils comme des commandes php disponible directement depuis la console de Symfony, comme doctrine:schema:create, doctrine:schema:drop, doctrine:schema:update. Avec Doctrine, chaque table est associée à une entité et à un

repository. Une entité s'est simplement une classe PHP utilisé par Doctrine pour représenter une table de la base de données. Un repository s'est simplement une classe php qui sert à exécuter des requêtes sur la table associé à l'entité. Doctrine permet d'ajouter une couche d'abstraction, au lieu d'écrire des requêtes SQL, avec Doctrine les requêtes sont écrite sous forme d'objets.

1.7 Développer la partie back-end d'une application web ou web mobile

Après la configuration du projet et la mise en place de la base données, j'ai commencé le développement de la partie back-end de l'application. J'ai commencé par le développement de la page d'accueil en créant un contrôleur pour la gestion des produits, puis j'ai ajouté une méthode nommée index chargé de retourner la vue correspondante à l'accueil du site.

1.8 Elaborer et mettre en œuvre des composants dans une application de gestion de contenu

Pour la gestion du contenu, j'ai encore une fois fait appel à EasyAdmin pour la créer, la lecture, la mise à jour et la suppression des produits (le contenu du site). Il permet aussi de restreindre l'accès aux actions du CRUD grâce au système de permission basé sur les rôles fourni par Symfony.

2. Résumé

2.1 Le site, la partie front-end

La partie frontale du site fait référence au code HTML, CSS et JavaScript du projet. Généralement tout ce qu'est exécuté, affiché ou stocké sur le navigateur du visiteur fait partie du front-end (du code html, css, js, les images, les cookies et les données du localStorage).

2.2 Le site, la partie back-end

La partie back-end du site concerne le code et les fichiers qui sont exécutés, affichés et stockés sur le serveur. En d'autres termes, les fichiers PHP, les fichiers JSON, les fichiers de sessions et les données stockées en base de données (qui sont le plus souvent sur le même serveur que le site).

2.3 Le site, les fonctionnalités liées à l'utilisateur

Chaque visiteur a la possibilité de créer un compte utilisateur gratuitement pour réaliser des achats et pour suivre ses commandes. Les utilisateurs ont aussi la possibilité de réinitialiser leur mot de passe et de modifier leurs informations personnelles depuis l'espace utilisateur.

2.4 Le site, la base de données

Les données de chaque utilisateur sont stockées en base de données. L'utilisation d'une base de donnée facilite le traitement des données et améliore la sécurité. Pour

assurer un minimum de sécurité, les données sensibles sont hachées avant d'être stockées et l'accès à la base de données est restreint. Le site utilise Mariadb comme SGBD un système de gestion de base de données édité sous licence GPL, un dérivé (un fork) de MySql.

3. Cahier des charges

3.1 Présentation du projet

Le site est là pour promouvoir les produits de la marque Audiophile. L'entreprise est composée d'une équipe d'amateurs et de spécialistes du son qui a pour objectif de proposer le meilleur matériel audio personnel. Audiophile possède une boutique à Paris avec des produits de démonstration accessible au public 7 jours sur 7.

3.2 Expression du besoin

Le client souhaite augmenter son membre de ventes et gagner en visibilité sur le web. Le client souhaite également s'implanter sur le marché Français de la vente en ligne de matériel audio pour le grand public et pour les clients qui recherche des produits audio de haute définition.

3.3 Étude de marché et benchmarking

Le marché du streaming musical en France en pleine croissance selon [businesscoot.com](#) : « À l'échelle mondiale, le marché du streaming musical a connu une croissance très dynamique. En effet, en 10 ans, entre 2011 et 2021, le chiffre d'affaires a été multiplié par 28, et le marché n'est toujours pas à bout de souffle, puisqu'il a connu une croissance de 24% entre 2020 et 2021. En France, le streaming suit le même élan, avec une croissance de 303% entre 2014 et 2021, sans avoir été réellement impacté par la crise sanitaire. ».

3.3.1 Les concurrents directes

Entreprise : presence audio conseil

Site internet : <https://presence-audio.com>

Avantages : propose de nombreuses marques et des produits d'occasions.

Adresse : 10, rue des Filles du Calvaire – 75003 Paris

Entreprise : digitec-derouet

Site internet : digitec-derouet.com

Avantages : propose des bandes magnétiques, amplificateurs, baladeurs, barre de son, home cinéma.

Adresse : 29, rue Custine 75018 Paris 18

3.4 Fonctions principales

3.4.1 Fonction principale 1

Permettre aux clients de commander facilement du matériel audio. Avoir à un suivi des commandes. Possibilité de se faire livrer à domicile ou en boutique.

3.4.2 Fonction principale 2

Trouver des informations utiles sur l'entreprise et les produits vendus par l'entreprise.

3.5 Spécifications annexes

Dans une future mise à jour, je compte utiliser Google Analytics pour affiner le SEO du site.

3.6 Les rôles

3.6.1 : Rôle utilisateur

Le client a le rôle ROLE_USER, ce rôle autorise le client à accéder à certaines parties du site comme l'espace utilisateur, la page d'accueil, la liste des produits listés par catégorie, au formulaire de paiement, à ses informations personnelles et à son historique de commandes.

3.6.2 : Rôle administrateur

L'administrateur a le rôle ROLE_ADMIN, ce rôle autorise l'administrateur accéder à l'ensemble du contenu du site. Il permet également à l'administrateur d'ajouter, de modifier et de supprimer des produits. Il permet aussi à l'administrateur d'avoir accès à l'espace administrateur.

3.7 Marché et population cible

3.7.1 Personnes ciblées

Médias utilisés	Personnalité	Jonathan Geeking	Infos personnelles
Smartphone	Créatif	Jonathan est étudiant en première année en art.	Jonathan Geeking
Ordinateur	Curieux	Il aime écouter de la musique dans les transports en communs.	Âge : 18 ans
	Social	Il écoute du majoritairement du rap.	Métier : étudiant en art
		Il recherche un casque audio avec une bonne autonomie.	Situation : célibataire
			Ville : Montpellier
			Revenu : 300 €

Médias utilisés	Personnalité	Philippe Jackson	Infos personnelles
Smartphone	Calme	Philippe est un ancien avocat à la retraite.	Philippe Jackson
Ordinateur	Curieux	Il aime écouter de la musique sur son temps libre poser dans son salon.	Âge : 65 ans
	Social	Il écoute beaucoup de musique classique.	Métier : avocat à la retraite
		Il recherche un casque audio de qualité hifi.	Situation : marié
			Ville : Paris
			Revenu : 3000 €

3.8 Maquettage

3.8.1 Exemple de pages sur ordinateur

The image displays three separate screenshots of a website for 'audiophile' across different sections: a product page for the XX99 Mark II headphones, a product page for the XX99 Mark I headphones, and a login/register page.

Top Screenshot (Product Page XX99 Mark II):

- Header:** audiophile, ACCUEIL, CASQUES, HAUT-PARLEURS, ÉCOUTEURS, ESPACE ADMIN, AZERTY, shopping cart icon.
- Section Title:** CASQUES
- Image:** A large image of the XX99 Mark II headphones.
- Text:** NOUVEAU PRODUIT, XX99 MARK II.
- Description:** The new XX99 Mark II headphones is the pinnacle of pristine audio. It redefines your premium headphone experience by reproducing the balanced depth and precision of studio-quality sound.
- Button:** VOIR LE PRODUIT

Middle Screenshot (Product Page XX99 Mark I):

- Header:** audiophile, ACCUEIL, CASQUES, HAUT-PARLEURS, ÉCOUTEURS, ESPACE ADMIN, AZERTY, shopping cart icon.
- Section Title:** NOUVEAU PRODUIT
- Image:** A large image of the XX99 Mark I headphones.
- Text:** NOUVEAU PRODUIT, XX99 MARK I HEADPHONES.
- Description:** As the gold standard for headphones, the classic XX99 Mark I offers detailed and accurate audio reproduction for audiophiles, mixing engineers, and music aficionados alike in studios and on the go.
- Button:** VOIR LE PRODUIT

Bottom Screenshot (Login/Register Page):

- Header:** audiophile, ACCUEIL, CASQUES, HAUT-PARLEURS, ÉCOUTEURS, SE CONNECTER, shopping cart icon.
- Section Title:** SE CONNECTER
- Form Fields:** adresse mail (azerty@live.fr), mot de passe (.....).
- Button:** SE CONNECTER / S'INSCRIRE

3.8.2 Exemple de pages sur périphérique portable

The mobile phone screen displays a product page for the ZX9 SPEAKER. At the top, there is a navigation bar with a menu icon, the brand name "audiophile", and a shopping cart icon. The main visual is a black bookshelf speaker with two white drivers, set against an orange background with a circular graphic. Below the image, the product name "ZX9 SPEAKER" is displayed in bold capital letters. A descriptive text block below the name reads: "Upgrade your sound system with the all new ZX9 active speaker. It's a bookshelf speaker system that offers truly wireless connectivity – creating new possibilities for more pleasing and practical audio setups." At the bottom of the screen is a black button with the text "VOIR LE PRODUIT" in white.

The mobile phone screen displays a category page for audiophile products. At the top, there is a navigation bar with a close icon, the brand name "audiophile", and a shopping cart icon. The page features three product categories: HEADPHONES, SPEAKERS, and EARPHONES. Each category is represented by an image of the product and a "SHOP" button. The HEADPHONES section shows a pair of over-ear headphones with gold accents. The SPEAKERS section shows a black bookshelf speaker. The EARPHONES section shows a pair of black earphones.

audiophile



nouveau produit

XX99 MARK II

The new XX99 Mark II headphones is the pinnacle of pristine audio. It redefines your premium headphone experience by reproducing the balanced depth and precision of studio-quality sound.

599,00 €

- 0 +

AJOUTER AU PANIER



audiophile

PANIER (1)

[vider](#)



XX99
Mark II
599,00
€

- 4 +

TOTAL 2396,00 €

PAYER

nouveau produit

XX99 MARK II

The new XX99 Mark II headphones is the pinnacle of pristine audio. It redefines your premium headphone experience by reproducing the balanced depth and precision of studio-quality sound.

599,00 €

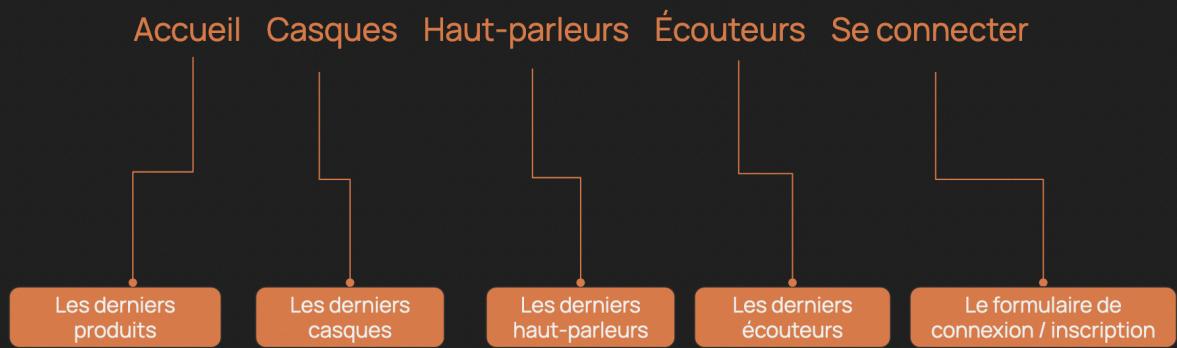
- 0 +

AJOUTER AU PANIER

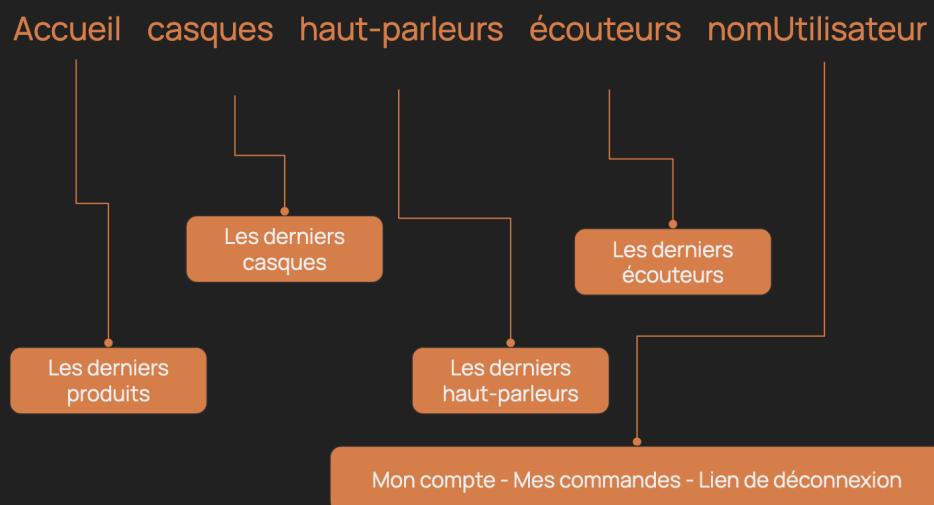


3.9 Arborescence du site

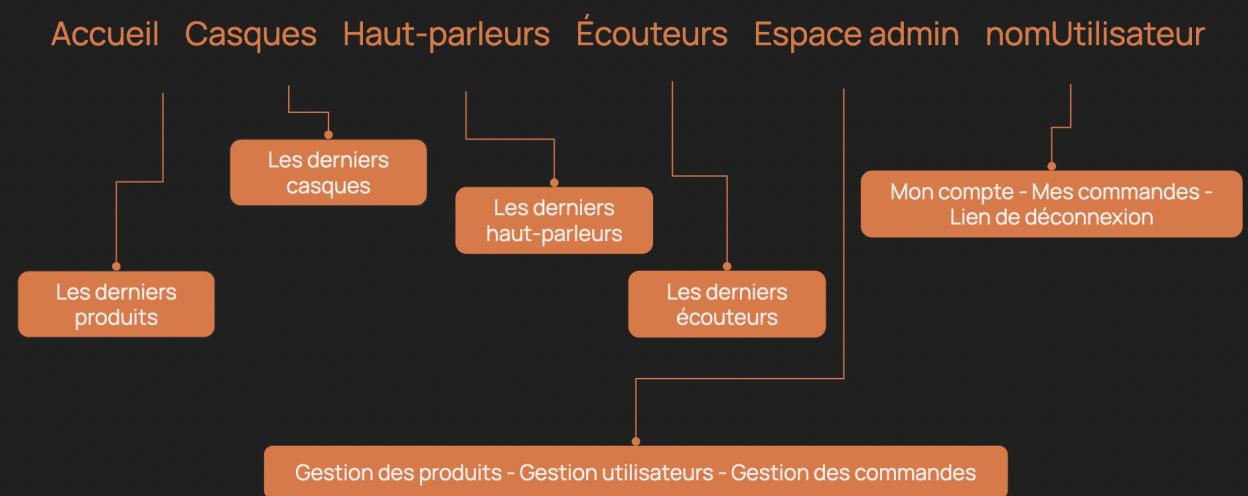
Plan du site - les utilisateurs non connectés



Plan du site - les utilisateurs connectés

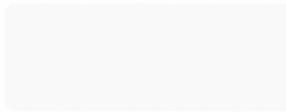
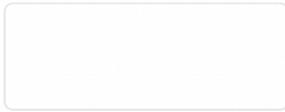


Plan du site - les administrateurs



3.10 Charte graphique : couleurs, typographie et logo

Les couleurs utilisées par l'entreprise sur ses différents supports de communications et avec ses produits sont :

			
HEX #D87D4A	HEX #101010	HEX #F1F1F1	HEX #FAFAFA
RGB 216, 125, 74	RGB 16, 16, 16	RGB 241, 241, 241	RGB 250, 250, 250
HSL 22°, 65%, 57%	HSL 0°, 0%, 6%	HSL 0°, 0%, 95%	HSL 0°, 0%, 98%
			
HEX #FBABF8	HEX #FFFFFF	HEX #000000	
RGB 251, 175, 133	RGB 255, 255, 255	RGB 0, 0, 0	
HSL 21°, 94%, 75%	HSL 0°, 0%, 100%	HSL 0°, 0%, 0%	

Les couleurs principales sont l'orange et le noir, la couleur orange est stimulante et symbolise l'énergie idéale pour des produits destinés aux sportifs. Quant au noir, il représente la sobriété et le luxe, une couleur idéale pour des produits à destination d'amateurs d'équipement audio de qualité hifi.

La typographie utilisée est la Manrope, une police de caractère moderne sans empattement open source. Conçu par Mikhail Sharanda entre 2018 et 2021.

Cette police de caractère a été choisie parce qu'elle a été conçue pour être utilisée sur le web et qu'elle prend en charge la plupart des langues latines et cyrilliques.

Pour le logo, il s'agit d'un logotype, ce type de logo permet de reconnaître l'entreprise en un coup d'œil et il est facile à mémoriser. De plus, on comprend rapidement qu'il s'agit d'une entreprise spécialisée dans la vente de matériel audio.



3.11 Planification des développements et délais

Réflexion et organisation

Jour 1

Installation et configuration - Création et configuration d'un projet Symfony

Installation et configuration - Création d'un dépôt git

Back-end - Création de la base de données

Front-end et back-end – Intégration HTML, CSS et mise en place des contrôleurs.

Jour 2

Mise en production - préparation du projet Symfony et de l'environnement pour la mise en production du site web

Dernier jour

4. Spécifications techniques du projet

4.1 Choix technologique

Pour le choix des techno, mon choix s'est orienté vers Symfony, car il s'agissait selon moi de l'outil le plus adapté au projet. Le projet demandait une personnalisation assez poussée de la partie front-end et back-end. Un CMS type wordpress n'était pas adapté pour ce genre de projet. L'avantage de Symfony, c'est qu'il repose sur l'architecture MVC, qu'il est libre, gratuit et qu'il dispose d'une bonne documentation et d'une grande communauté de développeurs sur qui compter en cas de problème.

Pour la base de données, mon choix s'est orienté vers Mariadb, car c'est un SGBD fiable, performant, facile à prendre en main, libre et open source.

Pour le css j'ai préféré utiliser le préprocesseur sass, car il offre des fonctionnalités supplémentaires et il permet de mieux organiser mon code.

4.2 Langages, librairies, framework et SGBD

Langage HTML

HTML signifie « HyperText Markup Language », il s'agit d'un langage de balisage non d'un langage de programmation, car en HTML, les structures de contrôle n'existent pas (des boucles et des blocs conditionnels). HTML est un langage de balisage, il sert à structurer des pages web.

Langage CSS

Le langage CSS est utilisé pour la mise en forme des pages web. C'est avec ce langage que les pages web sont stylisées.

Langage JS

Le JavaScript est un langage de programmation initialement développé pour dynamiser des pages. C'est un langage de programmation orienté objet, il utilise le principe du prototypage qui permet d'ajouter à la volée des attributs et des méthodes aux instances d'une classe. JavaScript peut aussi être interprété côté serveur grâce à NodeJS.

Bootstrap

Bootstrap est un framework, une collection d'outils HTML, CSS et JavaScript pour créer des pages Web. Il s'agit d'un projet gratuit et open source, créé à l'origine par Twitter.

Langage PHP

PHP : (HyperText Preprocessor) est un langage de programmation libre. Il est utilisé de nos jours pour gérer des pages web dynamiques via un serveur HTTP.

Symfony

Symfony est framework MVC libre écrit en PHP. Il fournit des fonctionnalités modulables et adaptables qui permettent de faciliter et d'accélérer le développement d'un site web.

MySQL

MySQL est un système de gestion de bases de données relationnelles (SGBDR), MySQL utilise l'architecture client serveur. Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de bases de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, PostgreSQL et Microsoft SQL Server.

4.4 Outilage

Environnement de développement intégré (IDE)

Visual Studio Code est un éditeur de code avec des fonctionnalités avancées, comme la coloration syntaxique et l'autocomplétion. Visual Code à l'avantage de disposé d'un riche catalogue d'extensions qui permettent d'étendre les fonctionnalités de bases. C'est un logiciel gratuit et développé par Microsoft pour Windows, macOS et Linux.

phpMyAdmi

PhpMyAdmin est un outil réalisé principalement en PHP met à disposition une interface web qui permet d'administrer une base de données.

4.5 Architecture MVC

Le dossier « Public » est la partie publique de mon projet, il contient tout ce qui doit être accessible par le navigateur. À l'intérieur du sous-dossier build on retrouve le dossier images pour les images de l'interface du site, un dossier fonts pour les polices de caractères et des fichiers css et JavaScript.

Le dossier « upload » sert à stocker les fichiers envoyés sur le serveur. Et pour finir le fichier index.php qui sert de point d'entrée à l'application Symfony.

Le dossier « src » contient les contrôleurs (les fichiers qui s'occupent du traitement des données), les entités (la partie modèle du MVC) le sous-dossier Security dans lequel se trouve la classe AppCustomAuthenticator et EmailVerifier.

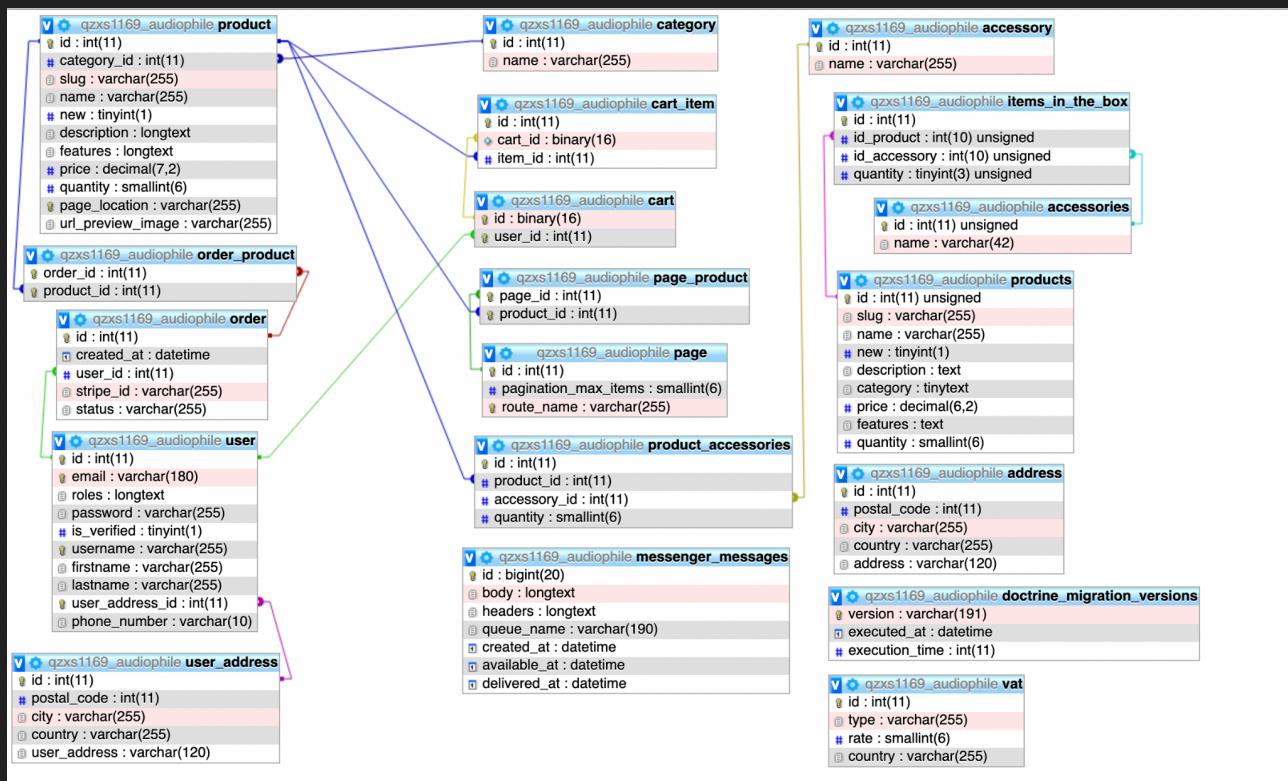
Le dossier « DataFixtures » contient la classe AppFixtures qui sert à générer un jeu de données pour les tests.

Le dossier « Templates » contient les vues de l'application, tout le html du projet. C'est la partie vue du modèle MVC.

4.6 Traitement des données

Pour manipuler les données de la base de données, j'ai fait appel à Doctrine, cet outil permet d'interagir avec une base de donnée relationnelle en utilisant des objets.

La base de données du site est composée de plusieurs tables, comme la table product pour stocker les données des produits, la table de liaison « cart_item » qui fait la relation entre la table product et cart.



4.7 Travaux de référencement

Le référencement naturel ou SEO désigne le contenu non sponsorisé et l'ensemble des techniques utilisées pour améliorer son classement dans les résultats proposés par les moteurs de recherches.

Pour la mise en place du SEO du site, j'ai commencé par une recherche de mots-clés, ensuite, j'ai utilisé les bonnes balises html pour décrire le contenu, j'ai fait en sorte que le titre et la méta description de chaque produit soit unique et j'ai ajouté un texte alternatif pour chaque image avec l'attribut alt.

J'ai également utilisé Screaming Frog, pour analyser les URL et les titres des pages du site.

4.8 Publication & déploiement

Avant de passer au déploiement du site, j'ai commencé par déterminer quel type d'hébergement sera le plus adapter au projet et ensuite, j'ai commencé à me renseigner sur les différents hébergeurs web disponible sur le marché.

Ensuite, je me suis renseigné sur la compatibilité avec les différents outils utilisés pour le projet afin d'éviter des problèmes pendent la mise en production.

Pour la mise en ligne du site, j'ai utilisé git et GitHub, car j'ai pas à chercher les fichiers modifiés un par un comme avec un client FTP pour les envoyer au serveur.

Après avoir envoyé le site sur le serveur, j'ai reconfiguré la base de données, vidé le cache, réinstaller le dossier vendor et node_modules, puis passé le projet en mode production à l'aide de la variable d'environnement APP_ENV.

4.6.1 Hébergement

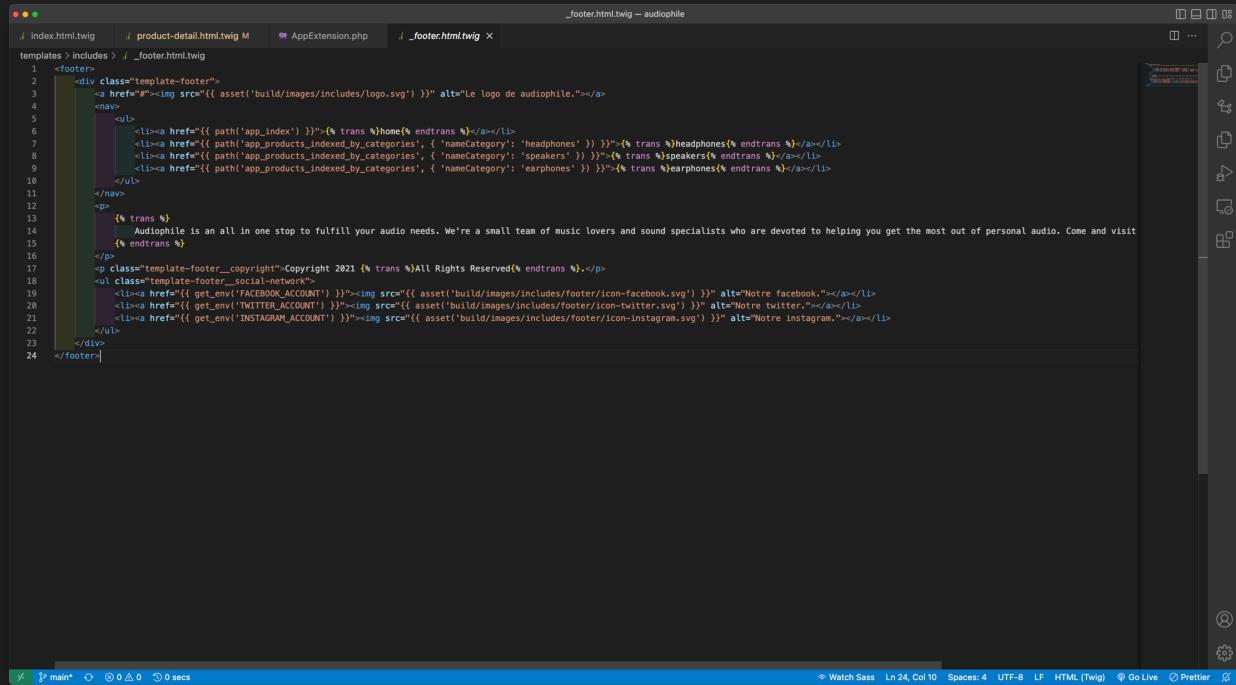
Pour l'hébergeur, j'ai choisi o2switch, un hébergeur web basé à Clermont-Ferrand. L'entreprise derrière o2switch est française, elle propose une seule offre d'hébergement web. Elle met à disposition un tas de fonctionnalités comme un accès ssh, un espace disque illimité, un nom de domaine offert, une base de données illimitées, des transferts mensuels illimités une interface cPanel et un service technique de qualité disponible 24 sur 24 et 7 jours sur 7.

4.6.2 Connexion SSL/TLS & certificat SSL pour HTTPS

Selon Wikipédia : « L'HyperText Transfer Protocol Secure est la combinaison du HTTP avec une couche de chiffrement comme SSL ou TLS. HTTPS permet au visiteur de vérifier l'identité du site web auquel il accède, grâce à un certificat d'authentification émis par une autorité tierce, réputée fiable. ». En d'autres termes, le protocole HTTPS est semblable au protocole HTTP, mais dispose d'une couche supplémentaire pour la sécurité. HTTPS permet de certifier l'identité du serveur à l'aide d'un certificat émit par un organisme officiel. Lorsque qu'une connexion HTTPS est établie, le serveur envoie son certificat SSL au navigateur. Ce certificat contient la clé publique nécessaire pour commencer la session sécurisée. Le navigateur utilise cette clé publique pour vérifier l'identité du serveur auprès d'une autorité de confiance. Google favorise les sites web utilisant le protocole HTTPS au détriment du protocole HTTP. Il est donc fortement recommandé d'utiliser le protocole HTTPS pour réduire les risques de piratage et améliorer son SEO.

5. Réalisations du candidat avec des extraits de code significatifs

5.1 Extrait Front-end



```
index.html.twig product-detail.html.twig AppExtension.php _footer.html.twig
templates > includes > _footer.html.twig
1 <div class="template-footer">
2   <a href="#"></a>
3   <nav>
4     <ul>
5       <li><a href="{{ path('app_index') }}"/>{{ trans $home % endtrans }}</a></li>
6       <li><a href="{{ path('app_products_indexed_by_categories', { 'nameCategory': 'headphones' }) }}"/>{{ trans $headphones % endtrans }}</a></li>
7       <li><a href="{{ path('app_products_indexed_by_categories', { 'nameCategory': 'speakers' }) }}"/>{{ trans $speakers % endtrans }}</a></li>
8       <li><a href="{{ path('app_products_indexed_by_categories', { 'nameCategory': 'earphones' }) }}"/>{{ trans $earphones % endtrans }}</a></li>
9     </ul>
10    </nav>
11    <p>{{ trans %}> Audiophile is an all in one stop to fulfill your audio needs. We're a small team of music lovers and sound specialists who are devoted to helping you get the most out of personal audio. Come and visit {{ endtrans }}</p>
12    <p class="template-footer_copyright">Copyright 2021 {{ trans %}All Rights Reserved{{ endtrans }}</p>
13    <ul class="template-footer_social-network">
14      <li><a href="{{ get_env('FACEBOOK_ACCOUNT') }}"/></a></li>
15      <li><a href="{{ get_env('TWITTER_ACCOUNT') }}"/></a></li>
16      <li><a href="{{ get_env('INSTAGRAM_ACCOUNT') }}"/></a></li>
17    </ul>
18  </div>
19
20
21
22
23
24</div>
</div>
```

J'ai choisi le footer comme extrait, car il utilise la fonction « `get_env` » que j'ai ajouté à Twig. Cette fonction permet d'accéder aux variables d'environnement depuis Twig, grâce à ce système, j'ai plus utilisé des variables d'environnement pour changer facilement et rapidement les liens vers les réseaux sociaux de l'entreprise.

5.2 Extrait Back-end

```
27 /**
28  * Retourne la variable d'environnement demandée.
29  *
30  * @param String $varname
31  * @return String
32  */
33 public function getEnvironmentVariable($varname)
34 {
35     return ${_ENV[$varname]};
36 }
37
38 /**
39  * Retourne le prix TTC d'un produit.
40  *
41  * @param [type] $price
42  * @param [type] $rate
43  * @return void
44  */
45 public function getPriceIncludingTax($price, $rate)
46 {
47     return $this->tavManager->getPriceIncludingTax($price, $rate);
48 }
49
50 /**
51  * Retourne le prix HT d'un produit.
52  *
53  * @param [type] $price
54  * @param [type] $rate
55  * @return void
56  */
57 public function getPriceExcl($price, $rate)
58 {
59     return $this->tavManager->getPriceIncludingTax($price, $rate);
60 }
61 }
62 }
```

Pour la partie back-end, j'ai choisi un extrait de la classe « AppExtension » qui contient la fonction `getEnvironmentVariable` vue précédemment (fonction trouvée sur Internet) et 2 autres fonctions qui sont `getPriceIncludingTax` et `getPriceExcl` que j'ai moi-même créer.

La fonction `getPriceIncludingTax` permet de récupérer le prix TTC d'un produit depuis n'importe quelle vue.

La fonction `getPriceExcl` permet de récupérer le prix hors taxe d'un produit depuis n'importe quelle vue.

Je compte refactoriser mon code et retirer les fonctions « `getPriceIncludingTax` » et « `getPriceExcl` » du fichier « `AppExtension.php` ». À la place, je pense créer un service pour chacune d'elles. Avec les services, j'aurais accès à « `getPriceIncludingTax` » et « `getPriceExcl` » directement depuis mes contrôleurs. Et pour les utiliser depuis la vue,

j'aurais qu'à faire une injection de dépendances à l'intérieur du fichier « AppExtension.php ».

5.3 Extrait fonctionnel

```
public function sendMail($params)
{
    $this->setFrom($params['from'], $params['name'] ?? null);
    $this->addAddress($params['to']);
    $this->Subject = $params['subject'];

    // Content
    $this->isHTML(true); // Set email format to HTML
    $this->msgHTML($params['body']);
    $this->send();
}
```

La méthode sendMail provient du service « PHPMailer », cette méthode sert à envoyer un mail. Elle prend un tableau en paramètre, un tableau qui contient l'adresse mail de l'expéditeur, l'adresse mail du destinataire, le sujet et le body du mail à envoyer et retourne aucune valeur.

6. Tests des fonctionnalités et résultats

6.1 Tests utilisateurs

Un test utilisateur, ou test d'utilisabilité, sert à évaluer l'ergonomie et la prise en main d'une application ou d'un logiciel. Il permet aussi de détecter des problèmes et de déterminer comment les utilisateurs utilisent le site. L'objectif d'un test utilisateur est d'améliorer l'expérience utilisateur.

6.2 Tests d'intégrité

Un test d'intégrité permet de s'assurer que les données soient valides (que les contraintes soient respectées), à jour, protégé et que le traitement sur ces derniers respects bien la réglementation relative aux données personnelles. Par exemple vérifier le type, la longueur, la provenance, si la personne est autorisée à il avoir accès, hacher les données sensibles avant leurs insertions, restreindre au maximum l'accès, n'autoriser que l'essentiel, utiliser les transactions pour éviter la corruption de données, garder à jour ses outils, faire une veille pour rester informer des nouvelles failles de sécurités et pour être au courant des nouvelles lois relatives aux traitements des données personnelles. Et surtout ne jamais faire confiance à l'utilisateur !

6.3 Tests d'injection

Pour la validation des formulaires, j'ai utilisé l'interface « ValidatorInterface », cette interface permet de poser des contraintes sur les entrées. Pour l'insertion/persistance des données en base de données, j'ai utilisé Doctrine pour préparer mes requêtes pour éviter les injections SQL.

Pour protéger au mieux le site des attaques XSS, j'ai fait appel à des fonctions php comme htmlspecialchars pour convertir les caractères spéciaux en entités HTML. Grâce à cette fonction, les caractères spéciaux sont plus interprétés par le navigateur et sont traités comme de simple caractères, ceci évite qu'une personne mal attentionnée injecte du code html ou du code JavaScript.

En ce qui concerne mes templates, j'ai rien fais de spéciale, car Twig se charge déjà d'échapper les caractères spéciaux pour éviter les attaques XSS.

7. Description de la veille technologique effectuée sur les vulnérabilités de sécurité

Pour réaliser ma veille en cybersécurité, je commence toujours par chercher l'information sur le site et les réseaux sociaux officiels des techno que j'utilise.

- Symfony : <https://symfony.com/blog/category/security-advisories>
- php : <https://www.php.net>
- Lemondeinformatique : <https://www.lemondeinformatique.fr/securite-informatique-3.html>
- developpez.com : <https://web.developpez.com>
- Apple : <https://www.apple.com/newsroom>
- Microsoft : <https://news.microsoft.com/category/press-releases>
- Visual Studio code : <https://code.visualstudio.com/blogs/2022/10/04/vscode-community-discussions>
- Visual Studio Code : <https://twitter.com/code>
- MDN : <https://developer.mozilla.org/fr>
- <https://www.ssi.gouv.fr/>
- <https://www.zataz.com/>

8. Description d'une situation de travail ayant nécessité une recherche

Lorsque j'ai voulu personnaliser la méthode index du Dashboard pour rediriger vers l'index des produits, j'ai rencontré un problème.

```
#[Route('/admin', name: 'admin')]
public function index(): Response
{
    return $this->redirect($this->adminUrlGenerator->setController(ProductCrudController::class)->generateUrl());
}
```

À chaque fois qu'un produit ou une catégorie était créé ou modifier l'utilisateur était systématiquement redirigé vers l'index des produits.

Pour résoudre ce problème, j'ai commencé dans un premier temps par chercher d'où venez le problème. Une fois le problème localisé, j'ai rafraichi à la meilleure fonction de résoudre le problème. Le problème était la méthode index du DashboardController redirigé systématiquement vers l'index des produits. Pour palier au problème, j'ai récupéré les paramètres crudAction et crudControllerFqcn pour rediriger l'utilisateur vers la bonne route et la bonne action.

```
16 class DashboardController extends AbstractDashboardController
17 {
18     private $adminUrlGenerator;
19     private $adminContextProvider;
20
21     public function __construct(AdminUrlGenerator $adminUrlGenerator, AdminContextProvider $adminContextProvider)
22     {
23         $this->adminUrlGenerator = $adminUrlGenerator;
24         $this->adminContextProvider = $adminContextProvider;
25     }
26
27     #[Route('/admin', name: 'admin')]
28     public function index(): Response
29     {
30         $uri = urldecode($this->adminContextProvider->getContext()->getRequest()->getRequestUri()); // Récupère l'uri depuis la requête.
31
32         preg_match_all('#crudControllerFqcn=(([a-z\\\\\\\\])*#i', $uri, $crudController); // Utilise une regex pour récupérer le nom du CRUD à appeler.
33         preg_match_all('#crudAction=([a-z]*#i', $uri, $crudAction); // Utilise une regex pour récupérer le nom de l'action à appeler.
34
35         /*
36             Si le paramètre avec le nom du CRUD n'est pas défini, alors assigne comme valeur le namespace du CRUD des produits.
37         */
38
39         if (empty($crudController[1])) {
40             $crudController[1] = 'Product';
41         }
42
43         $controller = $crudController[1];
44         $action = $crudAction[1];
45
46         $url = $this->adminUrlGenerator->setController($controller)->setAction($action)->generateUrl();
47
48         return $this->redirect($url);
49     }
50 }
```

9. Extrait de site en anglais utilisé dans la recherche

Pour résoudre mon problème de redirection avec le Dashboard, j'ai utilisé la fonction `php preg_match_all`.

Voici l'extrait en anglais :

« Searches subject for all matches to the regular expression given in pattern and puts them in matches in the order specified by flags.

After the first match is found, the subsequent searches are continued on from end of the last match. ».

Source : <https://www.php.net/manual/fr/function.preg-match-all.php>

Voici l'extrait en français :

Analyse le paramètre « `subject` » pour trouver une correspondance avec le `pattern` et place tous les résultats dans le tableau « `matches` », dans l'ordre spécifié par `flags`. Cette fonction retourne toutes les correspondances.

10. Evolution et travaux futurs

Dans le futur, je pense faire des corrections de bugs, refactoriser le code source et ajouter d'autres fonctionnalités.