



**MINISTÈRE
DE L'INTÉRIEUR
ET DES OUTRE-MER**

*Liberté
Égalité
Fraternité*



Dossier de synthèse du projet d'application

Chrystal BASTID

Titre professionnel : Concepteur Développeur d'Application

Année 2022/2023

Tuteur : Audric MATI

Remerciements

Je remercie les différents intervenants du Ministère de l'Intérieur (Dicom) qui m'ont permis d'effectuer une alternance au sein de leur structure dans un environnement technique enrichissant. Leur cadre de travail a joué un rôle essentiel dans le développement de mes compétences.

Je suis particulièrement reconnaissante envers Audric MATI qui m'a suivie lors de la réalisation de mon projet, ainsi qu'envers le reste de l'équipe technique (Hamza MEKHANEG) qui m'a soutenue durant toute mon année.

Je remercie l'équipe pédagogique de Doranco pour son accompagnement.

Je tiens à adresser mes sincères remerciements à mes camarades Sofiane BELHADI et Jayson MOOKEN pour leur précieuse aide tout au long de cette année. Leur soutien sans failles, leurs idées et leur collaboration ont été d'une importance capitale dans la réalisation de mon projet de conception d'application.

Enfin, je tiens à remercier ma famille et mes amis pour leur soutien inconditionnel. Leur encouragement constant a été une source de motivation et de force tout au long de ce parcours.

Je suis profondément reconnaissante envers toutes les personnes qui ont contribué de près ou de loin à la réalisation de mon projet de conception d'application. Leur soutien et leur collaboration ont été essentiels dans cette aventure enrichissante.

Encore une fois, merci à tous pour votre précieuse contribution.

Sommaire

Remerciements.....	2
Sommaire.....	3
Table des figures.....	4
Résumé.....	5
Table des abréviations.....	6
Partie I : Introduction.....	7
Avant-propos.....	7
Partie II : Cahier des charges.....	8
1. Project Presentation.....	8
Project objective.....	8
Analysis of the current state.....	8
Concept and Features of "La Voie".....	8
My role in the project.....	8
User feedback integration.....	9
Target Audience.....	9
2. Gestion du projet et conception.....	10
Contexte.....	10
Méthode agile.....	10
Cas d'utilisation.....	11
Conception et spécifications techniques.....	12
3. Graphisme & Ergonomie.....	16
La charte graphique.....	16
5. Développement de l'application.....	20
Le front.....	21
Le back.....	29
Les difficultés.....	32
Partie III : Annexes.....	35
1. Diagramme de Gantt.....	35
2. Schéma de la base de données.....	36
3. Suite des maquettes.....	36
4. Accessibilité.....	37
5. Rendu du formulaire.....	38
6. Exemple de création de table (fichier PHP).....	38

Table des figures

Figure 1: Diagramme de Gantt - La Voie sur Gantt Project.....	11
Figure 2: Cas d'utilisation - La Voie.....	11
Figure 3: Cas d'utilisation - Backoffice - La Voie.....	12
Figure 4: Couleurs de la charte graphique.....	17
Figure 5: Logo de l'application.....	18
Figure 6: Typographie de la charte graphique.....	19
Figure 7: Maquette de la page d'accueil.....	20
Figure 8: Extrait de code de base.html.twig.....	22
Figure 9: Extrait de code – personnalisation bootstrap/css.....	24
Figure 10: Rendu de la personnalisation via la css.....	24
Figure 11: Extrait de code – Formulaire contact.....	26
Figure 12: Extrait de code - page carte.....	27
Figure 13: Extrait de code - fichier js du lexique.....	28
Figure 14: Extrait du contrôleur ContactController.php.....	31
Figure 15: Tâches du diagramme de Gantt.....	35
Figure 16: Temps des tâches du diagramme de Gantt.....	35
Figure 17: Tables de la base de données.....	36
Figure 18: Maquette de la page "Carte".....	36
Figure 19: Maquette de la page "Passeport".....	37
Figure 20: Capture contraste entre texte et fond (footer).....	37
Figure 21: Capture contraste entre titre et fond (corps des pages).....	37
Figure 22: Rendu du formulaire - sur le site.....	38
Figure 23: Extrait de code - création de table en PHP.....	38

Résumé

Le dossier de synthèse présente en détail le processus de conception de l'application "La Voie". Cette application a été conçue dans le but de faciliter la localisation des spots de grimpe pour les grimpeurs et grimpeuses passionné(e)s.

L'application "La Voie" a été développée pour offrir une expérience utilisateur optimale. En utilisant une interface conviviale et intuitive, les grimpeurs et grimpeuses peuvent explorer une carte interactive présentant divers lieux de grimpe. Chaque point sur la carte est accompagné d'une fiche descriptive fournissant des informations détaillées telles que les cotations, les coordonnées GPS et d'autres détails pertinents. L'utilisation de technologies modernes, telles que JavaScript et le framework Symfony, a permis d'assurer la performance et la compatibilité de l'application.

Dans le cadre de la conception de l'application, j'ai pris en compte les retours des utilisateurs ainsi que les conseils et expertises des professionnels du domaine de l'escalade. En tant que grimpeuse confirmée, j'ai pu me mettre à la place des utilisateurs et ajuster l'application pour répondre à leurs besoins spécifiques. Cette approche basée sur mon expérience personnelle a joué un rôle clé dans la conception de "La Voie", garantissant une réponse adéquate aux attentes et aux exigences des grimpeurs et grimpeuses confirmé(e)s.

Ce dossier de synthèse met en lumière les différentes étapes du processus de conception de l'application "La Voie". À travers des maquettes et des schémas, je présenterais la conception visuelle de l'application, en mettant l'accent sur une interface ergonomique pour les utilisateurs. Je détaillerais également les outils que j'ai utilisés tout au long du développement, notamment pour la création des maquettes et la structuration des pages.

En résumé, "La Voie" est une application conçue pour simplifier la recherche et la découverte de spots de grimpe. Grâce à son interface conviviale, ses fonctionnalités avancées et son contenu riche, elle vise à faciliter l'expérience des grimpeurs et grimpeuses, en leur permettant de trouver les meilleurs spots adaptés à leurs besoins et de vivre des aventures verticales passionnantes.

Table des abréviations

API	Application Programming Interface (Interface de programmation applicative)
CSS	Cascading Style Sheets (Feuilles de style en cascade qui définissent le style des pages HTML)
HTML	Hyper Text Markup Language (Langage de balisage conçu pour représenter des pages Web.)
JS	JavaScript (Langage de programmation utilisé principalement pour le développement web.)
MCD	Modèle Conceptuel de Données (C'est une représentation abstraite des données d'un système. Il permet de décrire les entités et les relations entre ces entités de manière simplifiée.)
SQL	Structured Query Language (Langage servant à exploiter les bases de données relationnelles)
PHP	Hypertext Preprocesseur (Langage de programmation utilisé pour produire des pages web)
UML	Unified Modeling Language (Langage de modélisation unifié)
FTP	File Transfer Protocol (Protocole de communication pour le partage de fichier)
MVC	Modèle Vue Contrôleur (Architecture de projet web)
HTTP	Hypertext Transfer Protocol (« protocole de transfert hypertexte », est un protocole de communication client-serveur)

Partie I : Introduction

Avant-propos

Depuis mon plus jeune âge, ma passion pour l'informatique ne m'a jamais quittée. J'ai toujours été attirée par le fonctionnement des ordinateurs, effectuant des démontages, des recherches sur internet et m'initiant à la programmation. Cette passion m'a conduite à suivre un BAC Sciences et Technologies de l'Industrie et du Développement Durable, avec une spécialité en Innovation Technologique et Éco-conception. Ce parcours m'a permis d'acquérir une première expérience dans la réalisation de petits projets, la modélisation 3D et les premiers développements en langage C.

Suite à cela, j'ai choisi de poursuivre mes études en DUT Informatique, dans lequel j'ai eu l'opportunité de mettre en pratique mes compétences en développement web et de renforcer mes connaissances en tant que développeuse.

C'est ensuite que j'ai pris la décision de rejoindre l'Institut de Formation Doranco, car leur formation de Concepteur Développeur d'Application en alternance m'offrait de nouvelles perspectives d'évolution. Cette opportunité d'apprentissage en alternance me permet de concilier théorie et pratique, en travaillant sur des projets concrets tout en bénéficiant d'un encadrement professionnel.

Partie II : Cahier des charges

1. Project Presentation

Launched in September 2022, this project consists of the design and development of a web application dedicated to climbing.

Project objective

The main objective of "La Voie" is to simplify the search for climbing spots for climbing enthusiasts. I have noticed that there is a real need to centralize information about climbing spots in an accessible and practical way, avoiding the need to purchase multiple specialized books or rely solely on word-of-mouth.

Analysis of the current state

As part of my analysis of the current state, I conducted extensive research to understand the methods currently used to find climbing spots. I found that the references were often limited to purchasing specific books or relying on verbal recommendations.

Concept and Features of "La Voie"

Based on this analysis, I designed "La Voie" as a web application that offers an interactive mapping of climbing spots. The user-friendly and intuitive interface allows users to easily locate spots that match their preferences and desired location. Each spot is accompanied by a detailed description, including ratings, GPS coordinates, and other relevant information.

My role in the project

Throughout the duration of the project, I primarily worked independently. I took on various phases, from the design to the development of the application. However, I received guidance from my mentor in the company, whom I turned to when I needed specific advice.

User feedback integration

One of the strengths of "La Voie" lies in its constant evolution based on user feedback. I sought their feedback throughout the design and development process in order to optimize the application and adequately address their specific needs to the fullest extent possible.

Target Audience

With my application, "La Voie," my goal is to reach a wide audience, including both independent climbers and individuals at all levels of experience, whether they practice bouldering or sport climbing. The application offers a diverse range of climbing spots suitable for different levels of expertise, catering to beginners looking to get started in climbing as well as more experienced climbers seeking new challenges. Whether it's bouldering spots or climbing routes, "La Voie" provides a varied selection of climbing locations to meet the preferences and needs of everyone. By targeting a broad audience, my objective is to make climbing accessible to all and enable enthusiasts to discover new places and have enriching experiences in the vertical world.

In summary, the "La Voie" project is an innovative web application that simplifies the search for climbing spots for climbing enthusiasts. Its user-friendly interface, advanced features, and rich content make it an essential tool for all climbers.

2. Gestion du projet et conception

Contexte

Ce projet a été réalisé en autonomie, principalement pendant mon temps libre en dehors du travail, les week-ends et après les cours. J'ai utilisé l'un de mes ordinateurs de travail et bénéficié de mon environnement de travail habituel.

Cette approche indépendante m'a permis de me concentrer pleinement sur la gestion et la conception du projet. J'ai pu gérer mon emploi du temps en fonction de mes obligations professionnelles et académiques.

Malgré les défis liés à l'autonomie, j'ai maintenu une discipline et une rigueur tout au long du processus. J'ai pris des décisions, planifié, exécuté et suivi le projet de manière méthodique pour respecter les délais fixés.

La réalisation de ce projet en autonomie m'a permis de développer mes compétences en gestion de projet, en prise d'initiative et en organisation personnelle. Cela a également démontré ma capacité à travailler de manière indépendante et à mener à bien des projets complexes.

Méthode agile

Dans la gestion de mon projet, j'ai utilisé une méthode appelée Scrum, qui est une approche agile. Cela signifie que j'ai travaillé de manière itérative et collaborative tout au long du processus de développement. Même si j'étais la seule personne travaillant sur le projet, j'ai utilisé Scrum pour m'organiser.

Avec Scrum, j'ai divisé le projet en petites parties appelées "Sprints". À la fin de chaque Sprint, je me suis fixé des objectifs clairs à atteindre. Cette approche m'a permis de m'adapter facilement aux changements et de livrer régulièrement des fonctionnalités.

En plus de Scrum, j'ai utilisé un outil appelé diagramme de Gantt pour planifier mon projet. Ce diagramme m'a aidé à visualiser les différentes tâches et à fixer des dates limites pour chaque phase de travail.

En combinant Scrum avec le diagramme de Gantt, j'ai pu gérer mon projet de manière équilibrée. J'ai pu être flexible grâce à Scrum tout en gardant une vue d'ensemble et en respectant les échéances grâce au diagramme de Gantt.

Cette approche mixte a été essentielle pour la réussite de mon projet. Elle m'a permis de rester organisée, de suivre la progression et d'atteindre les objectifs et les délais que je m'étais fixés. Grâce à l'utilisation de Scrum et du diagramme de Gantt, j'ai pu gérer efficacement mon projet même en travaillant seule.

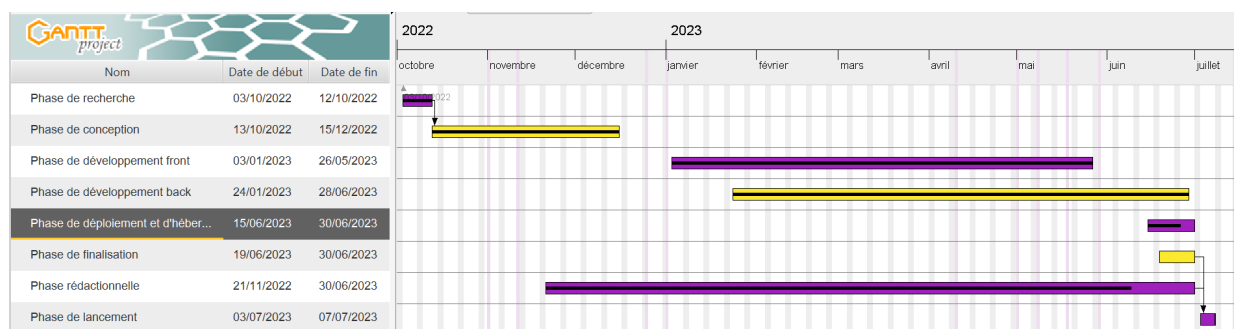


Figure 1: Diagramme de Gantt - La Voie sur Gantt Project

Cas d'utilisation

La Voie dispose d'un nom de domaine : lavoie-escalade.fr

Je me suis aidée de diagramme UML pour mieux visualiser les besoins pour les premières étape de conception.

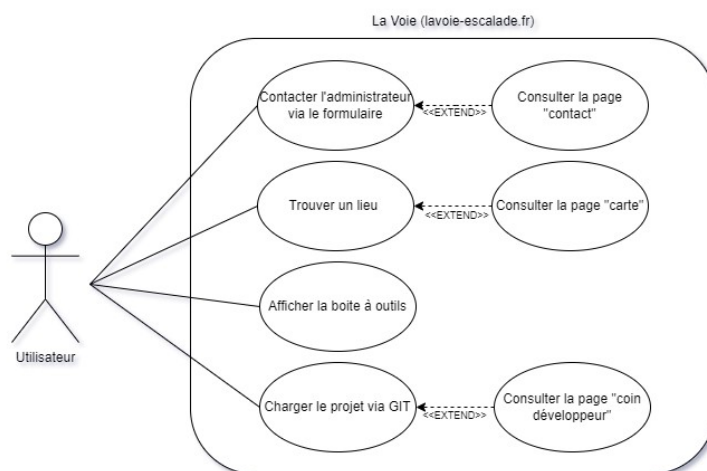


Figure 2: Cas d'utilisation - La Voie

L'acteur principal ici est l'utilisateur. Il s'agit de la personne cherchant à accéder à la plateforme (sans devoir s'inscrire ni se connecter).

Ensuite l'acteur principal du back-office est un administrateur. Il doit être en mesure de gérer les demande des utilisateurs, d'appeler l'API ou encore de mettre à jours les pages.

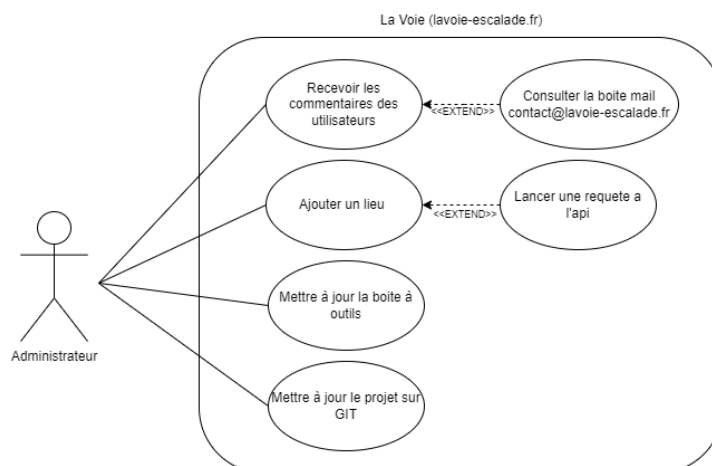


Figure 3: Cas d'utilisation – Backoffice – La Voie

Conception et spécifications techniques

1. Architecture

J'ai adopté une architecture en couches qui étend le modèle MVC et qui a permis une séparation claire des différentes responsabilités et une meilleure organisation du code. Voici l'architecture en couches que j'ai utilisée :

- 1.1. Couche de présentation : J'ai utilisé Symfony, un framework PHP, pour gérer la structure globale de mon projet. Pour la génération des vues HTML, j'ai utilisé Twig, un moteur de template qui m'a facilité la création et la gestion des fichiers HTML. De plus, j'ai employé Bootstrap, un framework CSS pour la mise en page et le style des pages. Pour apporter des fonctionnalités dynamiques et une interactivité à l'interface utilisateur, j'ai utilisé JavaScript. Cette couche m'a permis d'ajouter des comportements spécifiques aux éléments de l'interface, améliorant ainsi l'expérience utilisateur. De plus, j'ai utilisé CSS pour le style et la mise en forme des pages. Ceci correspond à la « Vue » du modèle MVC.
- 1.2. Couche de contrôle : Au niveau de la logique de gestion des requêtes HTTP, j'ai utilisé PHP pour la mise en place des contrôleurs. Ces contrôleurs sont chargés de traiter les requêtes, de récupérer les données nécessaires depuis le modèle et de les préparer pour l'affichage dans les vues. Ceci correspond aux « Contrôleurs » du MVC.

1.3. Couche de modèle : Pour la gestion de la base de données, j'ai opté pour MySQL. J'ai utilisé PHP pour créer les modèles de données et assurer les interactions nécessaires avec la base de données. Ceci correspond aux « Modèles » du MVC.

1.4. Couche de services : J'ai intégré une API externe pour récupérer des données spécifiques. Cette API a été utilisée pour enrichir les fonctionnalités de mon application. ("<https://equipements.sports.gouv.fr>").

2. API

Pour enrichir ma base de données avec des informations spécifiques à mon application, j'ai utilisé une API publique fournie par le Ministère des Sports (<https://equipements.sports.gouv.fr>). Cette API regroupe des données sur différents types d'équipements sportifs à travers le pays. Étant donné que mon application est axée sur l'escalade, j'ai dû effectuer un tri spécifique pour ne conserver que les installations sportives pertinentes.

En utilisant les fonctionnalités de l'API, j'ai pu filtrer les installations sportives en fonction de critères spécifiques à l'escalade, tels que la présence de murs d'escalade, de voies d'escalade ou d'autres équipements associés. Ce tri m'a permis de sélectionner uniquement les installations sportives qui répondent aux besoins des utilisateurs de mon application.

Une fois les données triées, j'ai intégré les informations pertinentes dans ma base de données, ce qui a permis à mon application de fournir des renseignements précis et pertinents sur les lieux de pratique de l'escalade.

L'apprentissage de l'utilisation de l'API a été essentiel pour réaliser cette opération de tri spécifique. Il m'a permis d'acquérir de nouvelles compétences techniques et de tirer parti des données disponibles pour offrir une expérience ciblée et adaptée aux utilisateurs de mon application d'escalade.

3. Base de données

Pour la gestion de ma base de données, j'ai opté pour l'utilisation de MySQL en conjonction avec phpMyAdmin. Ce choix s'est basé sur ma familiarité avec ces outils, que j'utilise depuis longtemps.

La structure de ma base de données comprend deux tables principales : 'Point' et 'Commentaire'. La table 'Point' stocke les informations relatives aux lieux d'escalade, telles que l'identifiant unique ('id'), le nom ('nom'), la latitude ('latitude'), la longitude ('longitude')

et le code postal ('code_postal'). Ces données sont extraites de l'API mentionnée précédemment et me permettent de localiser précisément les points d'intérêt pour les utilisateurs de mon application.




La deuxième table, 'Commentaire', est destinée à recueillir les commentaires laissés par les utilisateurs. Chaque ligne de cette table est identifiée par un identifiant unique ('id') et contient des informations telles que le nom ('nom'), le prénom ('prenom'), l'adresse e-mail ('mail') et le commentaire lui-même ('commentaire'). Cette structure me permet de collecter les retours des utilisateurs et de les utiliser pour éventuellement améliorer mon application.

J'ai également choisi MySQL et phpMyAdmin pour leur interface conviviale et leur flexibilité qui m'ont permis de créer, gérer et interagir facilement avec ma base de données. Je suis à l'aise avec l'écriture de requêtes SQL pour effectuer des opérations d'insertion, de mise à jour et de récupération des données nécessaires à partir de ma base de données.


Bien que ma base de données ne soit pas complexe, l'utilisation de MySQL avec phpMyAdmin m'a permis de gérer efficacement mes données et de fournir une expérience utilisateur fluide et réactive.

Vous pourrez trouver en annexe le schéma de la base de données.

4. Outils et technologies utilisées

Conception		Figma	Éditeur de graphiques vectoriels et un outil de prototypage
		Colour Contrast Analyser (CCA)	Outils qui permet de vérifier si les couleurs utilisées sont suffisamment contrastées pour les personnes malvoyantes
		GanttProject	Logiciel libre de gestion de projet écrit en Java

Développement		Visual Studio Code	Éditeur de code léger et multi-plateforme.
		Bootstrap	Collection d'outils utiles à la création du design de sites et d'applications web. C'est un framework.
		HTML / CSS	Standard pour la création de pages web, ces deux langages permettent la description et la stylisation des page web.
		JavaScript	Langage de programmation de scripts historiquement lié aux pages web mais utilisé également côté serveur.
		Symfony	Framework PHP « open source » utilisé par les développeurs pour créer des sites ou applications Web complexes, robustes, fiables, évolutifs, maintenables et performants.
		MySQL	Système de gestion de données relationnelle qui utilise le langage SQL
		Docker	Système d'exploitation (ou environnement d'exécution) pour conteneurs
Hébergement et déploiement		Ionos	Société d'hébergement Web
		FilleZilla	Client FTP utilisé pour interagit avec un serveur. Il permet d'échanger des fichiers de manières sécurisée.

Gestion de projet		Git hub	Gestionnaire de versions utilisé pour le stockage du code source des projets.
-------------------	---	---------	---

3. Graphisme & Ergonomie

La partie Graphisme & Ergonomie de mon projet se divise en deux aspects essentiels : la création d'une charte graphique attrayante et la conception de maquettes pour assurer une expérience utilisateur fluide et intuitive.

La charte graphique

1 Les objectifs

L'objectif principal de ma charte graphique est de créer une interface simple et épurée pour mon application. J'ai cherché à éviter de surcharger visuellement l'interface avec trop de couleurs et d'éléments graphiques afin de favoriser une expérience utilisateur fluide et agréable.

Pour atteindre cet objectif, j'ai choisi d'utiliser une palette de couleurs minimaliste, en me limitant à deux couleurs principales. Cela permet de maintenir une cohérence visuelle et de mettre en avant les informations essentielles.

En ce qui concerne la typographie, j'ai opté pour une police simple et lisible. L'objectif était de faciliter la lecture des informations présentées dans l'application, en évitant des polices trop stylisées ou complexes.

2 Les couleurs

Ma charte graphique repose sur l'utilisation de deux couleurs principales : le Dark Violet et l'Aureolin.

Le choix du Dark Violet est motivé par mon attrait personnel pour cette couleur. Je trouve qu'elle apporte une touche d'élégance et de mystère à l'interface de mon application. Le Dark Violet est également une couleur associée à l'univers de l'escalade, évoquant la profondeur des rochers et l'intensité/difficulté de l'activité.

Pour compléter le Dark Violet, j'ai choisi l'Aureolin comme couleur complémentaire. Le jaune vif de l'Aureolin s'associe harmonieusement avec le violet, créant un contraste visuel intéressant. Cette combinaison de couleurs apporte de la vitalité et de l'énergie à l'interface de l'application. L'Aureolin rappelle le soleil, la chaleur et la joie, évoquant ainsi l'aspect positif et passionnant de l'escalade.

En utilisant ces deux couleurs de manière équilibrée, j'ai cherché à créer un contraste visuel tout en maintenant une harmonie globale. Cela permet de mettre en avant les éléments importants de l'interface tout en conservant une esthétique cohérente et agréable.

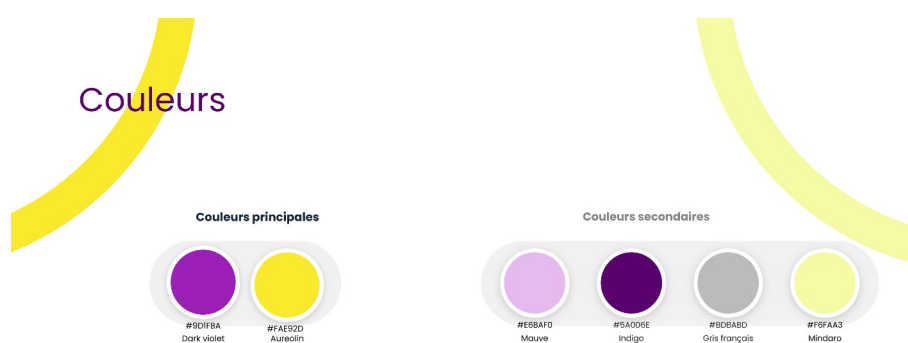


Figure 4: Couleurs de la charte graphique

3 Le logo

Le logo de mon application est une représentation visuelle puissante de l'esprit de l'escalade. Il est composé de deux éléments principaux : une montagne en haut et un mousqueton en bas.

La montagne est un symbole fort de la nature et de la liberté que chaque grimpeur recherche lorsqu'il se lance à l'assaut de sommets toujours plus hauts. Elle évoque la beauté époustouflante et l'immensité des paysages que l'on peut admirer en altitude. La montagne représente l'aventure, le défi et l'exploration de territoires inexplorés.

Le mousqueton, quant à lui, incarne la sécurité, un aspect essentiel en escalade. Cet outil permet aux grimpeurs de s'assurer et d'avancer en toute confiance. Il symbolise la confiance en soi, le contrôle et la maîtrise de l'équipement. Le mousqueton représente

également l'escalade en tant que passion qui pousse les grimpeurs à surmonter les obstacles et à repousser leurs limites.

En combinant ces deux éléments, la montagne et le mousqueton, mon logo représente l'esprit de l'escalade. Il évoque la fusion entre la beauté de la nature et la recherche de la sécurité dans cette discipline exigeante. Il symbolise l'excitation, la détermination et l'accomplissement personnel que l'on peut ressentir en pratiquant l'escalade.

Logothèque

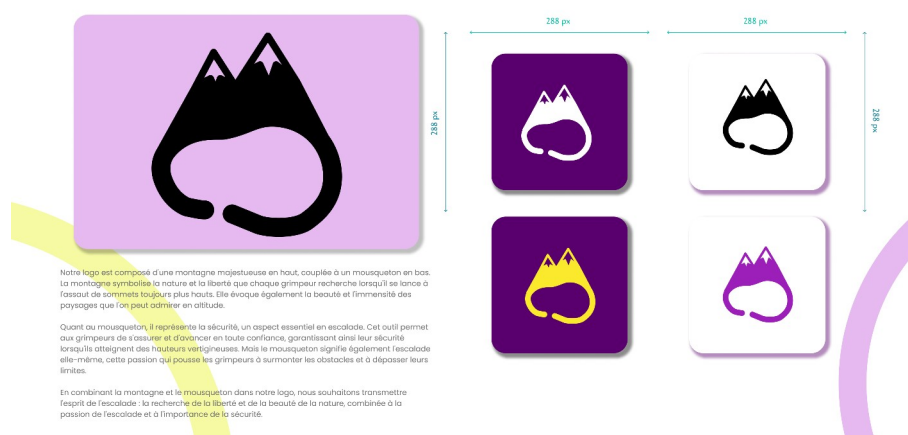


Figure 5: Logo de l'application

4 La typographie

La typographie joue un rôle essentiel dans la conception de mon application, et j'ai choisi avec soin la police Poppins pour véhiculer le bon message. Poppins est une police moderne, épurée et sans empattement qui se caractérise par sa clarté et sa lisibilité.

En optant pour la police Poppins, j'ai souhaité transmettre une sensation de simplicité et de modernité. Les lignes épurées de cette typographie apportent une esthétique contemporaine et une apparence nette à l'interface de l'application. Elle offre également une grande lisibilité, ce qui est essentiel pour garantir une expérience utilisateur agréable.

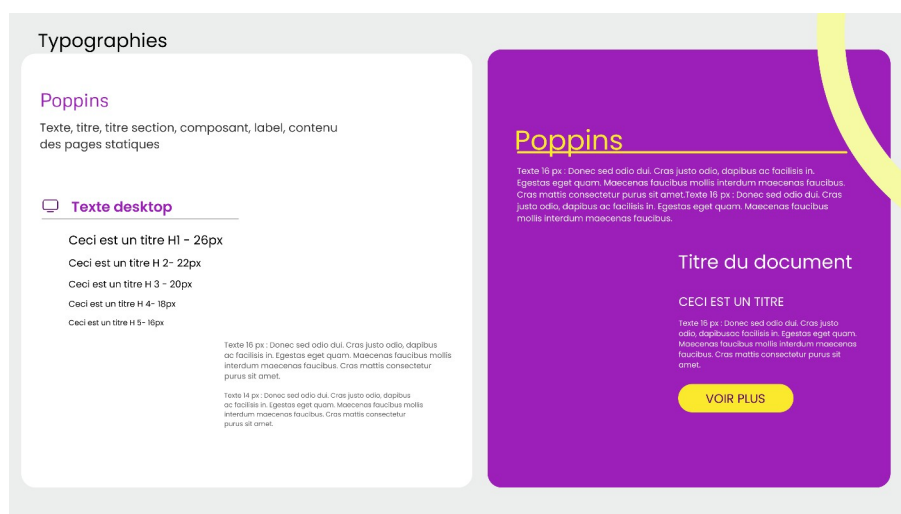


Figure 6: Typographie de la charte graphique

5 Wireframes & Maquettes

Les wireframes et maquettes sont des éléments essentiels dans le processus de conception de mon application "La Voie". Ils représentent une étape cruciale pour visualiser et structurer l'interface de manière claire et précise.

Les wireframes sont des schémas simplifiés qui décrivent la disposition des éléments sur chaque page de l'application. Ils permettent de définir l'organisation de l'information, la navigation et les interactions de base, sans se préoccuper des détails visuels. Les wireframes servent de guide pour concevoir une interface fonctionnelle et intuitive, en mettant l'accent sur l'expérience utilisateur.

Une fois les wireframes validés, les maquettes entrent en jeu. Les maquettes représentent des versions plus détaillées et esthétiques des différentes pages de l'application. Elles incluent les couleurs, les polices, les images et les éléments visuels qui donneront vie à l'interface. Les maquettes permettent de visualiser précisément l'apparence finale de l'application, en accordant une attention particulière à l'harmonie visuelle, à l'ergonomie et à l'accessibilité.

Pour réaliser les wireframes et les maquettes de mon application "La Voie", j'ai utilisé l'outil de conception Figma. Figma est un logiciel puissant et intuitif qui m'a permis de créer des interfaces visuelles de haute qualité.

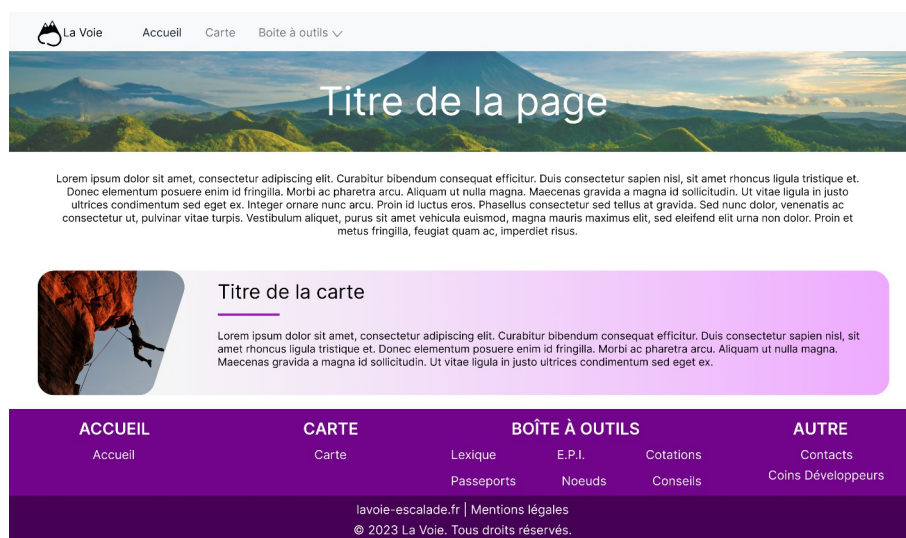


Figure 7: Maquette de la page d'accueil

5. Développement de l'application

Lors de la phase de développement de mon application web, j'ai entrepris un parcours passionnant et exigeant. J'ai commencé par créer les bases de mon application en utilisant uniquement des langages front-end tels que HTML, CSS et un peu de JavaScript.

Cependant, consciente des avantages qu'un framework de développement peut offrir en termes de rapidité, de structure et de maintenabilité du code, j'ai décidé de migrer vers Symfony. À ce stade, je n'avais aucune connaissance préalable de Symfony, mais j'ai perçu cela comme une opportunité d'apprentissage stimulante. J'ai donc entrepris une formation intensive pour acquérir rapidement et efficacement les compétences nécessaires à l'utilisation et à l'organisation de Symfony.

Une fois que j'ai maîtrisé les bases de Symfony, j'ai pu travailler de manière plus productive et me concentrer sur le développement et l'avancement de mon projet, avec l'aide précieuse de mon tuteur, Audric Mati. Son expertise dans le domaine de Symfony et sa connaissance approfondie de ce framework ont été des atouts majeurs tout au long de ma migration. Grâce à ses conseils avisés et à sa disponibilité, j'ai pu surmonter les difficultés initiales liées à l'apprentissage de Symfony et progresser rapidement dans le développement de mon application. Sa guidance m'a permis d'adopter les meilleures pratiques et d'optimiser l'organisation de mon code,

en utilisant efficacement la structure MVC de Symfony pour une séparation claire entre la logique métier et l'interface utilisateur.

Bien que le développement avec Symfony ait été une expérience enrichissante, il n'a pas été exempt de défis. J'ai rencontré des difficultés lors de l'intégration de certaines fonctionnalités spécifiques à mon application, ainsi que dans la gestion de la base de données et des requêtes complexes.

Le front

1. Header & footer

Grâce à Symfony, j'ai pu mettre en place un template de base pour mon application web. Ce template inclut le header et le footer, et il est utilisé sur toutes les autres pages de mon site. Cette approche me permet de bénéficier de plusieurs avantages.

Le header est la partie supérieure de la page. Elle facilite la navigation à travers le site. Quant au footer, c'est la partie inférieure du site qui termine chaque page. Elle contient notamment les liens vers toutes les pages de l'application ainsi que les mentions légales.

Tout d'abord, l'utilisation de blocs dans Symfony est très utile. J'ai défini des blocs pour le header et le footer dans mon template de base, et je les ai ensuite remplis avec le contenu spécifique à chaque page. Cela me permet de définir une structure cohérente pour toutes mes pages, tout en personnalisant facilement le contenu à l'intérieur de ces blocs.

Ensuite, cette méthode me permet de centraliser la gestion de l'en-tête et du pied de page. Plutôt que de devoir répéter le code correspondant sur chaque page, je n'ai besoin de les définir qu'une seule fois dans mon template de base. Ainsi, si je souhaite apporter des modifications au header ou au footer, je n'ai qu'à le faire une fois dans mon template, et ces changements seront automatiquement reflétés sur toutes les pages.

De plus, cette approche contribue à alléger mes autres pages. Au lieu d'inclure le code du header et du footer dans chaque page, je ne dois inclure que le contenu spécifique à cette page. Cela rend mon code plus clair et plus facile à maintenir.

```
base.html.twig M X
symfony-docker-template > templates > base.html.twig
1  <!DOCTYPE html>
2  <html>
3
4  > <head> ...
27 </head>
28
29 > {% block header %} ...
96 {% endblock %}
97
98 > <body> ...
100 </body>
101
102 {% block javabody %}{% endblock %}
103
104 > {% block footer %} ...
193 {% endblock %}
194
195 <!--Flèche scroll top-->
196 <a id="button-up"></a>
197 <script src="{{asset('js/scrollTop.js')}}"></script>
198 <!--Fin Flèche scroll top-->
199
200 </html>
```

Figure 8: Extrait de code de base.html.twig

Le template de base de mon application comprend différentes parties essentielles pour assurer une expérience utilisateur cohérente et conviviale.

La balise "head" (lignes 4 à 27) regroupe les éléments essentiels pour la configuration de la page, tels que le titre, les feuilles de style CSS et les scripts JavaScript. Cela permet de définir le style et les fonctionnalités de base du projet.

Ensuite, le bloc "header" (lignes 29 à 96) représentant l'en-tête de la page avec la barre de navigation et la bannière du site.

La balise "body" (ligne 98 à 100) est l'endroit où le contenu spécifique à chaque page sera inséré. Cela permet de séparer clairement la structure de la page du contenu lui-même.

Ensuite, le bloc "footer" (lignes 104 à 193) correspondant au pied de page avec les différents liens vers les sections du site.

Enfin, la flèche de défilement vers le haut (lignes 195 à 198) est un ajout pratique qui permet aux utilisateurs de revenir rapidement en haut de la page. Cela améliore l'expérience utilisateur en permettant une navigation fluide et facile.

2. Utilisation de twig

Twig est un moteur de templates utilisé dans le framework Symfony pour créer des applications web. Il permet de séparer de manière claire le code PHP et le code HTML en utilisant des fichiers de templates Twig avec une syntaxe spécifique. Les tags et les filtres Twig offrent un contrôle logique et une manipulation des données, permettant de générer des pages HTML dynamiques et adaptées aux données de l'application. L'utilisation de Twig favorise la séparation des préoccupations, facilite la réutilisabilité du code et rend le développement et la maintenance plus efficaces.

Les tags Twig sont des balises spéciales qui permettent d'implémenter des fonctionnalités logiques dans les templates. Par exemple, le tag `{% for %}` est utilisé pour créer une boucle qui itère sur une liste de données et génère du contenu dynamiquement. Les filtres Twig sont des fonctionnalités qui permettent de manipuler les données avant de les afficher dans les templates.

3. Bootstrap

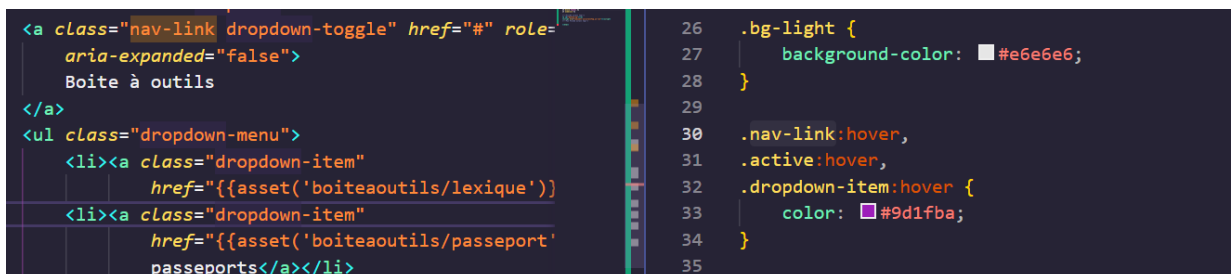
Bootstrap est un framework front-end largement utilisé pour développer des sites web et des applications web. J'ai choisi d'utiliser Bootstrap dans mon projet pour bénéficier de sa vaste collection de composants prêts à l'emploi, tels que les boutons, les formulaires, les barres de navigation, etc. Ces composants m'ont permis de gagner du temps dans le développement en me fournissant des éléments esthétiquement attrayants et fonctionnels.

Cependant, pour garantir que l'interface respecte ma charte graphique, j'ai effectué des personnalisations sur Bootstrap. J'ai ajusté les couleurs pour correspondre à l'esthétique et à l'identité visuelle de mon projet. Ces modifications m'ont permis de créer une interface cohérente et harmonieuse qui reflète l'image de marque de mon application.

4. CSS personnalisé

Pour ma personnalisation, j'ai ajusté la couleur des liens dans la barre de navigation. Par défaut, Bootstrap utilise une couleur bleue pour les liens dans la barre de navigation. Cependant, pour correspondre à ma charte graphique et ajouter une touche personnalisée, j'ai modifié la couleur des liens pour qu'ils soient violets. Cette modification a permis d'harmoniser les couleurs de la barre de navigation avec le reste de l'interface, en respectant ma palette de couleurs. Ainsi, les liens violets dans la barre de navigation se fondent parfaitement avec le style visuel global de mon projet.

Voici un extrait du code pour illustrer cette personnalisation :



```

<a class="nav-link dropdown-toggle" href="#" role="button"
  aria-expanded="false">
  Boite à outils
</a>
<ul class="dropdown-menu">
  <li><a class="dropdown-item"
    href="{asset('boiteaoutils/lexique')}">
  </li><a class="dropdown-item"
    href="{asset('boiteaoutils/passeport'
    passeports</a></li>
  
```

```

26 .bg-light {
27   background-color: #e6e6e6;
28 }
29
30 .nav-link:hover,
31 .active:hover,
32 .dropdown-item:hover {
33   color: #9d1fba;
34 }
35
  
```

Figure 9: Extrait de code – personnalisation bootstrap/css

Dans le fichier **base.html.twig**, à gauche, nous pouvons observer la présence de la classe "**nav-link**". C'est cette classe qui est utilisée pour les liens de la barre de navigation.

Ensuite, dans le fichier **lavoie.css**, à droite, j'ai ajouté une personnalisation pour la classe "**.nav-link**" afin de changer la couleur des liens en violet.

Cette modification de couleur permet d'obtenir les liens violets dans la barre de navigation, correspondant ainsi à ma charte graphique.

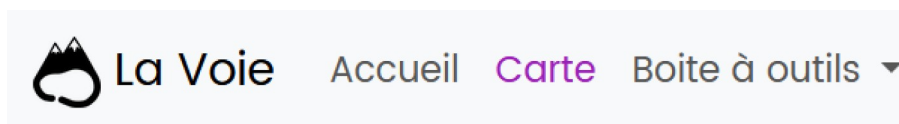


Figure 10: Rendu de la personnalisation via la css

5. Accessibilité

L'accessibilité est une considération essentielle lors du développement d'applications web. Dans le cadre de mon projet utilisant Bootstrap, j'ai accordé une attention particulière à l'accessibilité en m'assurant que le choix des couleurs respecte les normes de contraste minimum recommandées pour garantir une lisibilité optimale.

Conformément aux directives d'accessibilité, j'ai vérifié que les combinaisons de couleurs utilisées dans mon interface respectent les ratios de contraste minimum requis. Pour m'assurer du respect des normes d'accessibilité en matière de contraste des couleurs, j'ai utilisé un outil appelé "Colour Contrast Analyser (CCA)".

En utilisant le Colour Contrast Analyser, j'ai pu vérifier les ratios de contraste entre les couleurs utilisées dans mon interface. Cet outil m'a fourni des indications claires sur la conformité aux niveaux AA et AAA des normes WCAG en matière de contraste des couleurs. Ainsi, j'ai pu m'assurer que les utilisateurs, y compris ceux ayant des déficiences visuelles, peuvent lire et interagir facilement avec le contenu de mon application.

En tant que développeuse, je suis également consciente de l'importance de l'accessibilité pour les personnes atteintes de dyslexie. J'ai donc pris des mesures supplémentaires pour rendre le texte plus lisible et clair pour les personnes dyslexiques. Notamment, j'ai sélectionné une police sans empattement, qui est connue pour faciliter la lecture pour les personnes dyslexiques. Cette décision contribue à améliorer l'accessibilité et l'expérience utilisateur pour tous, y compris ceux qui ont des besoins spécifiques en matière de dyslexie.

En respectant les normes d'accessibilité, en utilisant Bootstrap comme base et en prenant en compte les besoins spécifiques des personnes atteintes de dyslexie, j'ai pu créer une interface web conviviale, accessible et inclusive pour tous les utilisateurs.

6. Formulaire

La page de formulaire que j'ai développée utilise Bootstrap pour son style et sa mise en forme. Ce formulaire a été conçu dans le but de permettre à tous les visiteurs de laisser un avis ou remarque sur mon site. Une fois soumis, les avis sont enregistrés dans ma base de données, plus précisément dans la table "Commentaire".

Voici le code de la page **contact.html.twig** qui contient le formulaire :

```
<h3>Contactez moi via le formulaire :</h3>

<form method="post">
  <div class="form-group mb-2">
    <label for="inputName">Nom</label>
    <input type="text" class="form-control" id="inputName" name="name">
  </div>
  <div class="form-group mb-2">
    <label for="inputLastName">Prénom</label>
    <input type="text" class="form-control" id="inputLastName" name="last_name">
  </div>
  <div class="form-group mb-2">
    <label for="inputEmail">E-mail</label>
    <input type="email" class="form-control" id="inputEmail" name="email" required>
  </div>
  <div class="form-group mb-2">
    <label for="inputMessage">Message</label>
    <textarea class="form-control" id="inputMessage" name="message" rows="3"></textarea>
  </div>
  <div class="d-flex justify-content-center">
    <button type="submit" class="btn btn-primary">Envoyer</button>
  </div>
</form>
```

Figure 11: Extrait de code – Formulaire contact

Ce formulaire demande aux utilisateurs de saisir leur nom, prénom, adresse mail et un message. Une fois que les utilisateurs cliquent sur le bouton "Envoyer", les informations saisies sont envoyées au serveur pour traitement. Par la suite, les données sont enregistrées dans ma base de données.

L'utilisation de Bootstrap pour la création de ce formulaire garantit une présentation cohérente, une interface responsive (adaptée aux différentes tailles d'écran) et une expérience utilisateur agréable. Les utilisateurs peuvent facilement remplir le formulaire et partager leurs avis, ce qui contribue à l'interactivité de mon site.

7. Carte

Dans ma partie "Carte", j'ai intégré une carte interactive qui utilise OpenStreetMap. Cette carte me permet de placer des points en utilisant les données récupérées depuis une API.

Une fois les points placés sur la carte, il est possible de cliquer dessus pour afficher une petite description comprenant le nom du lieu, son code postal ainsi que ses coordonnées GPS avec la latitude et la longitude.

Voici un extrait du code que j'ai utilisé pour cette fonctionnalité :

```

{% block body %}
<div id="mapid"></div>

<script>
  // Fonction pour effectuer une requête AJAX
  function getData(url, callback) {
    var xhr = new XMLHttpRequest();
    xhr.onreadystatechange = function () {
      if (xhr.readyState === 4 && xhr.status === 200) {
        callback(JSON.parse(xhr.responseText));
      }
    };
    xhr.open("GET", url, true);
    xhr.send();
  }

  // Appel de la fonction pour récupérer les données du contrôleur
  getData('{{ path('import') }}', function (data) {
    var points = data;

    console.log(points);

    // Pour créer la carte et ajouter les marqueurs
    let mymap = L.map('mapid').setView([46.603354, 1.888334], 6);

    // Ajout de la couche de tuiles OpenStreetMap
    L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
      attribution: 'Map data &copy; <a href="https://www.openstreetmap.org/">OpenStreetMap</a> contr',
      maxZoom: 18
    }).addTo(mymap);

    // Boucle pour ajouter les marqueurs à la carte
    points.forEach(function (point) {
      let name = point.name;
      let latlng = point.latlng;
      let codepostal = point.codepostal;

      console.log(latlng);

      L.marker(latlng).addTo(mymap)
        .bindPopup("<b>Nom : " + name + "</b> <br>Code postal : " + codepostal + "<br>Latitude : ");
    });
  });
</script>
{% endblock %}

```

Figure 12: Extrait de code – page carte

Grâce à ce code, j'ai pu créer une carte interactive avec OpenStreetMap et y placer des marqueurs correspondant aux points récupérés depuis l'API. Chaque marqueur est associé à une petite fenêtre contextuelle (*popup*) qui affiche le nom du lieu, son code postal ainsi que ses coordonnées GPS (*latitude et longitude*). Cette fonctionnalité est utile pour obtenir des informations détaillées sur chaque point et facilite la navigation et l'orientation des utilisateurs vers ces lieux.

8. Lexique

Dans la partie "Lexique" de mon site, j'ai utilisé JavaScript pour permettre aux personnes débutantes en escalade de comprendre les termes spécifiques de ce sport. Cette page vise à fournir un repère et à expliquer le sens des mots couramment utilisés dans l'escalade.

Voici le code JavaScript que j'ai utilisé :

```
const motsEscalade = [{ ...
  ];

// trier Les mots par ordre alphabétique
motsEscalade.sort((a, b) => {
  return a.mot.localeCompare(b.mot);
});

// créer un objet qui va stocker chaque section du Lexique
const lexique = {};

// parcourir chaque mot et Les ranger dans La bonne section du Lexique
motsEscalade.forEach((mot) => {
  const premiereLettre = mot.mot[0].toUpperCase();

  if (!lexique[premiereLettre]) {
    lexique[premiereLettre] = [];
  }

  lexique[premiereLettre].push(mot);
});

// parcourir chaque section du Lexique et trier Les mots par ordre alphabétique
for (let section in lexique) {
  lexique[section].sort((a, b) => {
    return a.mot.localeCompare(b.mot);
  });
}

// créer Le contenu HTML du Lexique trié
let contenuHtml = '';

for (let section in lexique) {
  const idSection = `${section.toLowerCase()}`; // créer un ID unique pour chaque section
  contenuHtml += `<dl><dt id="${idSection}">${section}</dt><dl>`;

  lexique[section].forEach((mot) => {
    contenuHtml += `<dd><strong>${mot.mot}</strong>: ${mot.definition}</dd>`;
  });
}

// afficher Le contenu HTML dans La page
document.getElementById('lexique').innerHTML = contenuHtml;
```

Figure 13: Extrait de code – fichier js du lexique

Le code commence par définir un tableau "**motsEscalade**" qui contient les mots de l'escalade avec leurs définitions.

Ensuite, les mots sont triés par ordre alphabétique à l'aide de la fonction **sort**. En JavaScript, la fonction **sort** est une méthode intégrée pour trier les éléments d'un tableau. Elle

peut être appliquée à un tableau pour réorganiser ses éléments selon un ordre spécifié.

Un objet **lexique** est créé pour stocker chaque section du lexique.

Chaque mot est ensuite placé dans la bonne section du lexique en fonction de sa première lettre. Les mots sont également triés par ordre alphabétique à l'intérieur de chaque section.

Enfin, le contenu HTML du lexique trié est généré en parcourant les sections du lexique. Chaque mot est affiché avec sa définition.

Le contenu HTML est ensuite affiché dans la page en utilisant la méthode **innerHTML** pour cibler l'élément avec l'ID "lexique". La propriété **innerHTML** est une propriété de l'objet **HTMLElement** en JavaScript qui permet d'accéder au contenu HTML à l'intérieur d'un élément spécifié. Elle peut être utilisée pour récupérer ou modifier le contenu HTML d'un élément.

Grâce à cette approche, les utilisateurs peuvent facilement trouver et comprendre les mots spécifiques de l'escalade, ce qui facilite leur apprentissage et leur immersion dans ce sport passionnant. Sachant que le lexique évoluera tous du long de la vie du site.

Le back

1. La base de données

La base de données constitue le fondement de mon application backend. J'ai choisi d'utiliser MariaDB avec MySQL comme moteur de base de données, et j'ai utilisé l'interface PhpMyAdmin pour gérer et administrer ma base de données.

Dans un premier temps, j'ai créé les tables de ma base de données localement en utilisant Docker. J'ai utilisé des commandes spécifiques pour faciliter ce processus. Par exemple, j'ai utilisé la commande "**php bin/console make:entity**" pour créer les entités correspondantes à mes tables. Cette commande a généré le code PHP nécessaire à la création des classes d'entités représentant mes tables.

Une fois les entités créées, j'ai utilisé la commande "**php bin/console make:migration**" pour générer les migrations. Les migrations sont des fichiers qui décrivent les changements à apporter à la structure de la base de données. Ces fichiers contiennent les instructions permettant de créer, modifier ou supprimer des tables, des colonnes, des clés, etc.

Ensuite, j'ai utilisé la commande "**php bin/console doctrine:migration:migrate**" pour appliquer les migrations et mettre à jour ma base de données. Cette commande a exécuté les migrations enregistrées, ce qui a permis de créer les tables et de mettre à jour la structure de la base de données selon les changements définis dans les migrations.

L'utilisation de ces commandes et de la migration de sauvegarde a simplifié la gestion de la structure de ma base de données. Cela m'a permis de suivre une approche de développement itératif, en pouvant facilement créer, modifier et supprimer des tables ou des colonnes sans affecter les données existantes.

Par la suite, j'ai utilisé l'interface PhpMyAdmin pour gérer et administrer ma base de données. Cette interface m'a permis de visualiser les tables, d'importer et d'exporter des données.

La combinaison de l'utilisation de Docker, des commandes Symfony et de l'interface PhpMyAdmin m'a offert un ensemble d'outils puissants pour créer et gérer efficacement ma base de données. Et cela, en suivant les meilleures pratiques de développement.

2. Les contrôleurs

Les contrôleurs jouent un rôle essentiel dans mon application backend. Ils sont responsables de la gestion des requêtes HTTP, de l'exécution des actions appropriées et de la génération des réponses correspondantes. J'ai utilisé le framework Symfony pour créer mes contrôleurs et bénéficier de ses fonctionnalités avancées.

Chaque contrôleur correspond à une partie spécifique de mon application et est responsable de la logique métier associée à cette partie. J'ai créé des contrôleurs pour gérer les fonctionnalités clés de mon application, telles que la gestion du formulaire de contact ou encore la gestion des imports des points (stockage des points de l'API vers ma base de données).

Symfony offre une structure claire et modulaire pour organiser mes contrôleurs. J'ai créé des classes de contrôleur dans mon projet, en utilisant des annotations spécifiques pour définir les routes, les méthodes et les actions correspondantes. Par exemple, j'ai utilisé l'annotation **@Route** pour définir l'URL associée à chaque action de mon contrôleur.

Voici un exemple de contrôleur que j'ai créé pour gérer la fonctionnalité de contact de mon application :

```
class ContactController extends AbstractController {
    private $entityManager;
    private $mailer;

    public function __construct(EntityManagerInterface $entityManager, MailerInterface $mailer)
    {
        $this->entityManager = $entityManager;
        $this->mailer = $mailer;
    }

    /**
     * @Route("/contact", name="contact")
     */
    public function conseil(Request $request): Response
    {
        $success = false;

        if ($request->isMethod('POST')) {
            $name = $request->request->get('name');
            $lastName = $request->request->get('last_name');
            $email = $request->request->get('email');
            $message = $request->request->get('message');

            // Création de l'entité Commentaire et enregistrement en base de données
            $commentaire = new Commentaire();
            $commentaire->setNom($name);
            $commentaire->setPrenom($lastName);
            $commentaire->setEmail($email);
            $commentaire->setCommentaire($message);

            $this->entityManager->persist($commentaire);
            $this->entityManager->flush();

            // Envoi de l'e-mail à contact@lavoie-escalade.fr
            /*$this->sendEmailNotification($name, $lastName, $email, $message);*/

            // Marquer la soumission du formulaire comme réussie
            $success = true;
        }

        return $this->render('contact/contact.html.twig', [
            'css_path' => 'css/lavoie.css',
            'success' => $success,
        ]);
    }
}
```

Figure 14: Extrait du contrôleur ContactController.php

Dans cet exemple, le contrôleur **ContactController** est responsable de la gestion de l'URL **/contact**. Lorsqu'une requête POST est reçue, le contrôleur récupère les données du formulaire de contact, les valide si nécessaire, crée une nouvelle entité **Commentaire** et l'enregistre en base de données à l'aide de l'**EntityManager**. Ensuite, il rend le template **contact/contact.html.twig** en passant une variable **success** pour indiquer si le formulaire a été soumis avec succès ou non.

Les difficultés

1. Développeuse front à développeuse fullstack

En tant que développeuse front, l'une des principales difficultés auxquelles j'ai été confrontée dans ce projet a été le fait de devoir m'impliquer davantage dans le développement backend. Habituellement spécialisée dans le développement de l'interface utilisateur et des fonctionnalités côté client, j'ai dû élargir mes compétences pour aborder des aspects plus techniques et orientés serveur.

Cette transition du front au back a nécessité un ajustement de ma perspective de développement. J'ai dû me replonger dans certains langages de programmation back, découvrir des frameworks spécifiques au backend et me familiariser avec des concepts tels que la gestion des bases de données, les API, etc.

Néanmoins, cette expérience m'a permis de repousser mes limites et de développer mes compétences de manière globale. Elle m'a également donné une vision plus complète du développement d'applications et a renforcé ma polyvalence en tant que développeuse.

2. Apprentissage de Symfony

L'une des difficultés majeures que j'ai rencontrées dans ce projet a été d'apprendre Symfony, un framework PHP. En tant que développeuse front, je n'avais pas d'expérience préalable avec Symfony, ce qui signifiait que je devais me plonger dans l'apprentissage de ce nouvel outil.

Cette phase d'apprentissage a été enrichissante mais aussi exigeante. J'ai dû me familiariser avec des concepts tels que les routes, les contrôleurs, les vues, les entités, les migrations, et comprendre comment ils s'intègrent dans l'architecture globale de Symfony.

Malgré les défis, l'apprentissage de Symfony a été une expérience précieuse. Il m'a permis de développer mes compétences en matière de développement backend et d'élargir mon champ d'expertise. Grâce à Symfony, j'ai pu bénéficier de fonctionnalités avancées et d'une structure cohérente pour développer mon application de manière efficace et robuste.

3. Nouveauté de l'utilisation de cartes

L'une des nouveautés majeures de ce projet a été l'utilisation de cartes. En tant que développeuse, je n'avais jamais eu l'occasion de travailler avec des cartes auparavant, ce qui représentait un défi stimulant.

Les cartes sont des outils puissants pour visualiser et interagir avec des données géographiques. Dans le cadre de mon projet, j'ai utilisé des cartes pour afficher les sites d'escalade.

Pour utiliser des cartes, j'ai dû explorer différentes technologies et API de cartographie. J'ai choisi une API qui correspondait le mieux à mes besoins et j'ai appris à l'utiliser efficacement. Cela impliquait de comprendre la documentation de l'API, d'acquérir les compétences nécessaires pour afficher des marqueurs sur la carte.

L'utilisation de cartes a ajouté une dimension interactive et visuelle à mon application. Les utilisateurs peuvent explorer les sites d'escalade sur une carte, zoomer, faire glisser la carte et obtenir des informations contextuelles en cliquant sur les marqueurs. Cela a permis de rendre l'expérience utilisateur plus immersive et conviviale.

4. Première utilisation d'une API

Une des grandes découvertes de ce projet a été l'utilisation d'une API. En effet c'était la première fois que j'en utilisais une. C'était une expérience nouvelle et passionnante.

Pour cette première utilisation d'une API, j'ai dû apprendre à interagir avec celle-ci en comprenant sa documentation. J'ai dû obtenir une clé d'API, qui est essentielle pour authentifier mes requêtes et accéder aux données de manière sécurisée.

Une fois que j'ai maîtrisé les bases de l'utilisation de l'API, j'ai pu récupérer les données dont j'avais besoin, telles que les informations sur les sites d'escalade, les codes postaux ou encore les coordonnées GPS. J'ai utilisé ces données pour enrichir mon application et offrir des fonctionnalités utiles aux utilisateurs.

5. Gestion d'un projet de grande envergure en solo

La gestion d'un projet de grande envergure en tant que développeuse en solo a été un défi stimulant et gratifiant pour moi. Pour la première fois, j'ai dû assumer toutes les responsabilités, de la conception à la mise en œuvre, en passant par la gestion du temps et des ressources.

L'une des principales difficultés auxquelles j'ai été confrontée était de coordonner les différentes tâches et de veiller à ce que le projet progresse de manière cohérente. J'ai dû établir un plan de travail détaillé, en identifiant les objectifs, les étapes clés et les délais pour chaque phase du projet. Cela m'a aidé à maintenir le cap et à éviter les retards.

Un autre défi majeur était de gérer les différentes compétences requises pour le projet. En tant que développeuse frontend, j'ai dû me familiariser avec des technologies et des concepts backend tels que Symfony, la base de données et la gestion des API. Cela a exigé un apprentissage intensif et une adaptation rapide pour acquérir les compétences nécessaires.

La gestion du temps a été un aspect crucial de ce projet. J'ai dû établir des priorités, organiser mon emploi du temps et respecter mes délais. La planification et l'organisation efficaces ont été essentielles pour assurer la progression régulière du projet et éviter les retards.

Malgré les défis, gérer un projet de grande envergure en solo m'a apporté une grande satisfaction. Cela m'a permis de développer mes compétences en tant que développeuse, d'acquérir de nouvelles connaissances et de repousser mes limites. Cela m'a également donné une plus grande autonomie et une meilleure compréhension de l'ensemble du processus de développement.

Partie III : Annexes

1. Diagramme de Gantt

1. Zoom sur les différentes phases du projet

Nom	Date de début	Date de fin
Phase de recherche	03/10/2022	12/10/2022
Phase de conception	13/10/2022	15/12/2022
Phase de développement front	03/01/2023	26/05/2023
Phase de développement back	24/01/2023	28/06/2023
Phase de déploiement et d'hébergement	15/06/2023	30/06/2023
Phase de finalisation	19/06/2023	30/06/2023
Phase rédactionnelle	21/11/2022	30/06/2023
Phase de lancement	03/07/2023	07/07/2023

Figure 15: Tâches du diagramme de Gantt

2. Zoom sur le temps des tâches

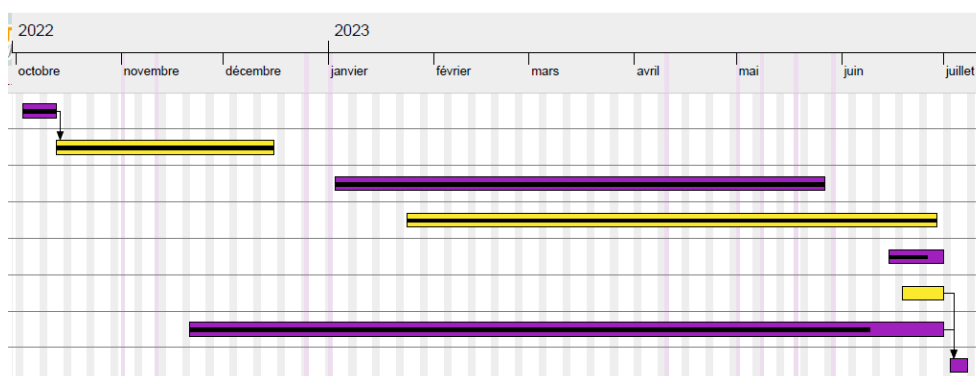


Figure 16: Temps des tâches du diagramme de Gantt

2. Schéma de la base de données

point		commentaire	
id	INT	id_commentaire	INT
nom	VARCHAR(50)	nom	VARCHAR(50)
latitude	DECIMAL(15,10)	prenom	VARCHAR(50)
longitude	DECIMAL(15,10)	mail	VARCHAR(50)
type_voie	LOGICAL	commentaire	VARCHAR(255)
observation	VARCHAR(255)		
cotation	VARCHAR(5)		
hauteur	DECIMAL(5,2)		
num_departement	SMALLINT		
ville	VARCHAR(50)		
adresse	VARCHAR(50)		

Figure 17: Tables de la base de données

3. Suite des maquettes

La page « Carte » est relativement simple. Elle est composée du même header et footer que tous le reste du site. Cette page contient une carte interactive afin de faciliter la navigation.

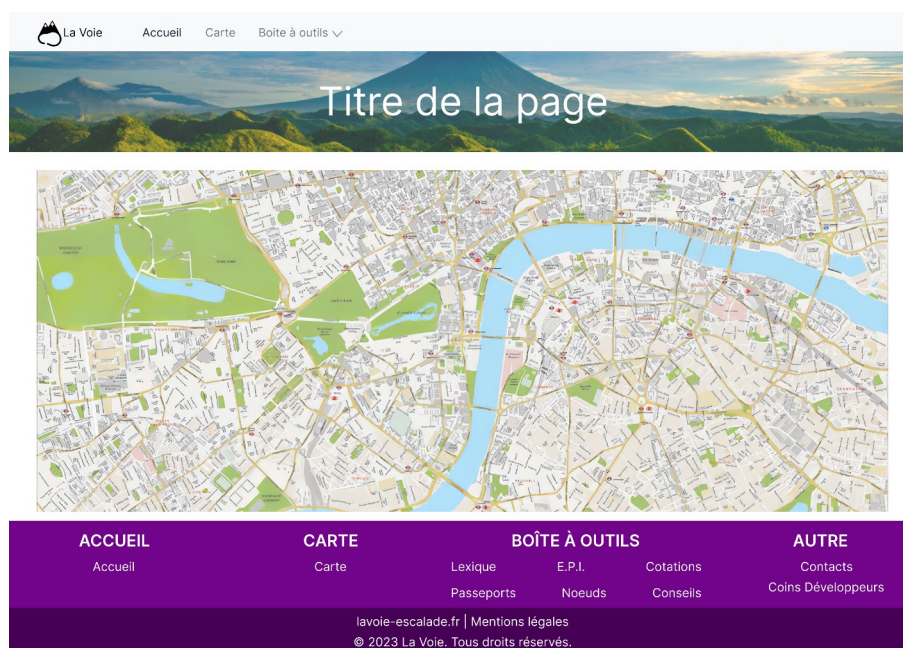


Figure 18: Maquette de la page "Carte"

La page « passeport » représente les pages qui constitue le reste du projet tel que « Noeuds » ou encore « Cotations ».

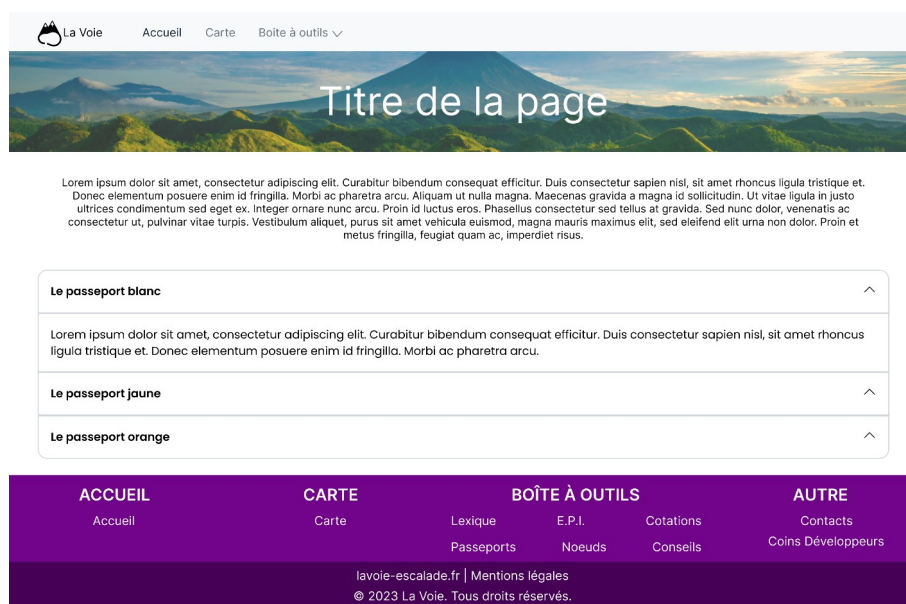


Figure 19: Maquette de la page "Passeport"

4. Accessibilité



Figure 21: Capture contraste entre titre et fond (corps des pages)

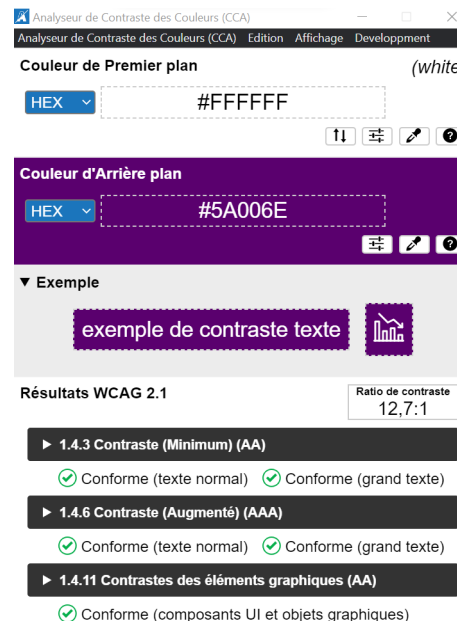


Figure 20: Capture contraste entre texte et fond (footer)

5. Rendu du formulaire

Contactez moi via le formulaire :

Nom

Prénom

E-mail

Message

Figure 22: Rendu du formulaire – sur le site

6. Exemple de création de table (fichier PHP)

```
1 <?php
2
3 namespace App\Entity;
4
5 use App\Repository\CommentaireRepository;
6 use Doctrine\ORM\Mapping as ORM;
7
8 #[ORM\Entity(repositoryClass: CommentaireRepository::class)]
9 class Commentaire
10 {
11     #[ORM\Id]
12     #[ORM\GeneratedValue]
13     #[ORM\Column]
14     private ?int $id = null;
15
16     #[ORM\Column(length: 255, nullable: true)]
17     private ?string $nom = null;
18
19     #[ORM\Column(length: 255, nullable: true)]
20     private ?string $prenom = null;
21
22     #[ORM\Column(length: 255)]
23     private ?string $email = null;
24
25     #[ORM\Column(length: 255)]
26     private ?string $commentaire = null;
27
28     public function getId(): ?int
29     {
30         return $this->id;
31     }
32
33     public function getNom(): ?string
34     {
35         return $this->nom;
36     }
37
38     public function setNom(?string $nom): self
39     {
40         $this->nom = $nom;
41
42         return $this;
43     }
44
45     public function getPrenom(): ?string
46     {
47         return $this->prenom;
48     }
49 }
```

Figure 23: Extrait de code – création de table en PHP