

# **Rapport du Projet « SPORTIFY »**

---

**Full stack Web site**

**Abdeljalil GUERROUF && LAGGOUN Khaled**

## Table des matières

<b>Présentation du Projet :</b> .....	2
<b>Analyse des besoins fonctionnels :</b> .....	2
<b>Analyse des besoins techniques</b> .....	2
<b>UI/UX DESIGN :</b> .....	4
<b>Objectif du Design :</b> .....	4
<b>Police principale :</b> .....	4
<b>Palette des couleurs :</b> .....	4
<b>Design général :</b> .....	5
<b>Architecture du projet et arborescence des fichiers</b> .....	6
<b>Schéma global de l'architecture</b> .....	6
<b>Arborescence et analyse des fichiers :</b> .....	6
<b>Fonctionnalité clé du site</b> .....	8
<b>Inscription des utilisateurs :</b> .....	8
<b>Connexion et déconnexion :</b> .....	8
<b>Réservation de cours sportifs :</b> .....	8
<b>Consultation des réservation :</b> .....	8
<b>Annulation de réservation :</b> .....	9
<b>Formulaire de contact :</b> .....	9
<b>Demande de devis personnalisé :</b> .....	9
<b>Architecture de base de données</b> .....	10
<b>Structure de la base de données :</b> .....	10
<b>Relations entre tables :</b> .....	12
<b>Principes d'architecture :</b> .....	13
<b>Sécurité de l'application :</b> .....	14
<b>Protection contre SQL Injection :</b> .....	14
<b>Sécurité des mots de passe :</b> .....	14
<b>Cahier des charges pour le projet Sportify</b> .....	15
<b>Répartition des taches :</b> .....	15
<b>Guide d'installation et de configuration du site sur votre machine</b> .....	16

# Introduction de projet :

## Présentation du Projet :

« **SPORITFY** » est un site web dédié à une académie d'arts martiaux. Il présente nos créneaux d'entraînement, nos tarifs, nos offres ainsi que nos objectifs. S'inscrire sur notre plateforme constitue déjà une première étape vers la découverte et la pratique des arts martiaux. C'est l'occasion idéale pour explorer nos formules et commencer ton parcours dans cet univers. Le site est disponible à l'adresse suivante : <http://sportify-lggr.great-site.net/>

## Analyse des besoins fonctionnels :

- **Système d'authentification :**
  - Permettre aux utilisateurs de s'inscrire et de se connecter de manière sécurisée.
- **Réservation de créneaux :**
  - Offrir la possibilité de réserver des créneaux d'entraînement en ligne et d'annuler une réservation.
- **Calcul des tarifs**
  - Intégrer un système automatique de calcul des prix en fonction des offres choisies.
- **Présentation de l'Académie :**
  - Mettre en avant les objectifs, les réalisations ainsi que les informations pratiques comme l'adresse.
- **Facilitation du contact :**
  - Permettre aux visiteurs de nous contacter facilement via le site.

## Analyse des besoins techniques

- **UI/UX Design :**
  - **Outil utilisé :** FIGMA
  - FIGMA est utilisé pour concevoir et analyser l'interface du site web. Il facilite la création de maquettes intuitives et contribue à offrir une expérience utilisateur (UX) optimale.
- **Frontend :**
  - **Technologies utilisées :** HTML, CSS, JavaScript, Bootstrap

- Ces technologies permettent d'intégrer les maquettes issues de FIGMA, de structurer les pages web, d'ajouter de l'interactivité et de dynamiser les éléments de l'interface. Elles permettent également de gérer la logique côté client afin de faciliter l'envoi et la récupération des données depuis ou vers le serveur.
- **Backend :**
  - **Technologies utilisées :** PHP MySQL
  - PHP est utilisé pour la gestion des traitements côté serveur, tandis que MySQL permet la création et la gestion de la base de données.
  - **PHPMailer :** Une bibliothèque open source disponible sur GitHub, utilisée pour l'envoi des courriels via un serveur SMTP en PHP
- **Outil du développement :**
  - **IDE :** Visual Studio Code (**VS Code**), un éditeur de code performant et adapté au développement en HTML, CSS, JavaScript et PHP.
  - **Plateforme de collaboration :** Git et GitHub, utilisés pour le travail collaboratif et le suivi de version du projet. (<https://github.com/Lgnkhaed/Sportify>).
  - **XAMPP :** Un environnement de développement local permettant de simuler un serveur web, d'exécuter du code PHP et de gérer les bases de données via phpMyAdmin

# UI/UX DESIGN :

## Objectif du Design :

Ce design vise à offrir une interface intuitive, moderne et responsive, adaptée à tous types d'utilisateurs, qu'ils soient débutants ou expérimentés dans les arts martiaux. L'accent a été mis sur la simplicité de navigation, l'accessibilité des informations et une expérience utilisateur fluide. L'objectif est également de permettre à des personnes peu familières avec l'informatique de pouvoir utiliser le site sans difficulté.

## Police principale :

**BEBAS Neue** – récupérée sur **Google Fonts**, cette police a été choisie pour son caractère imposant et sa forte présence visuelle, reflétant le sérieux et la rigueur des arts martiaux. Elle s'intègre parfaitement à la palette de couleurs du site, principalement noir et blanc, pour renforcer l'atmosphère stricte et professionnelle du site.

## Palette des couleurs :

- **Couleurs principales :**
  - **Noir : #000000**  
Utilisé pour le fond principal du site, afin de transmettre une atmosphère sérieuse, disciplinée et épurée
  - **Blanc : #FFFFFF**  
Utilisé pour les textes et certaines sections de contenu afin de créer un contraste clair avec le fond noir et garantir une lisibilité optimale.
- **Couleur du Footer :**
  - **Gris foncé : DARK du Bootstrap #6c757d**
  - Cette couleur a été choisie pour le footer, afin de maintenir une tonalité sobre et renforcer l'aspect professionnel et sérieux du site.
- **Couleur d'accents :**
  - **Rouge : #ff0000**  
Utilisé pour les messages d'erreur, les alertes et autres éléments nécessitant l'attention de l'utilisateur. Le rouge est une couleur

conventionnelle pour signaler les erreurs et avertir l'utilisateur de manière directe.

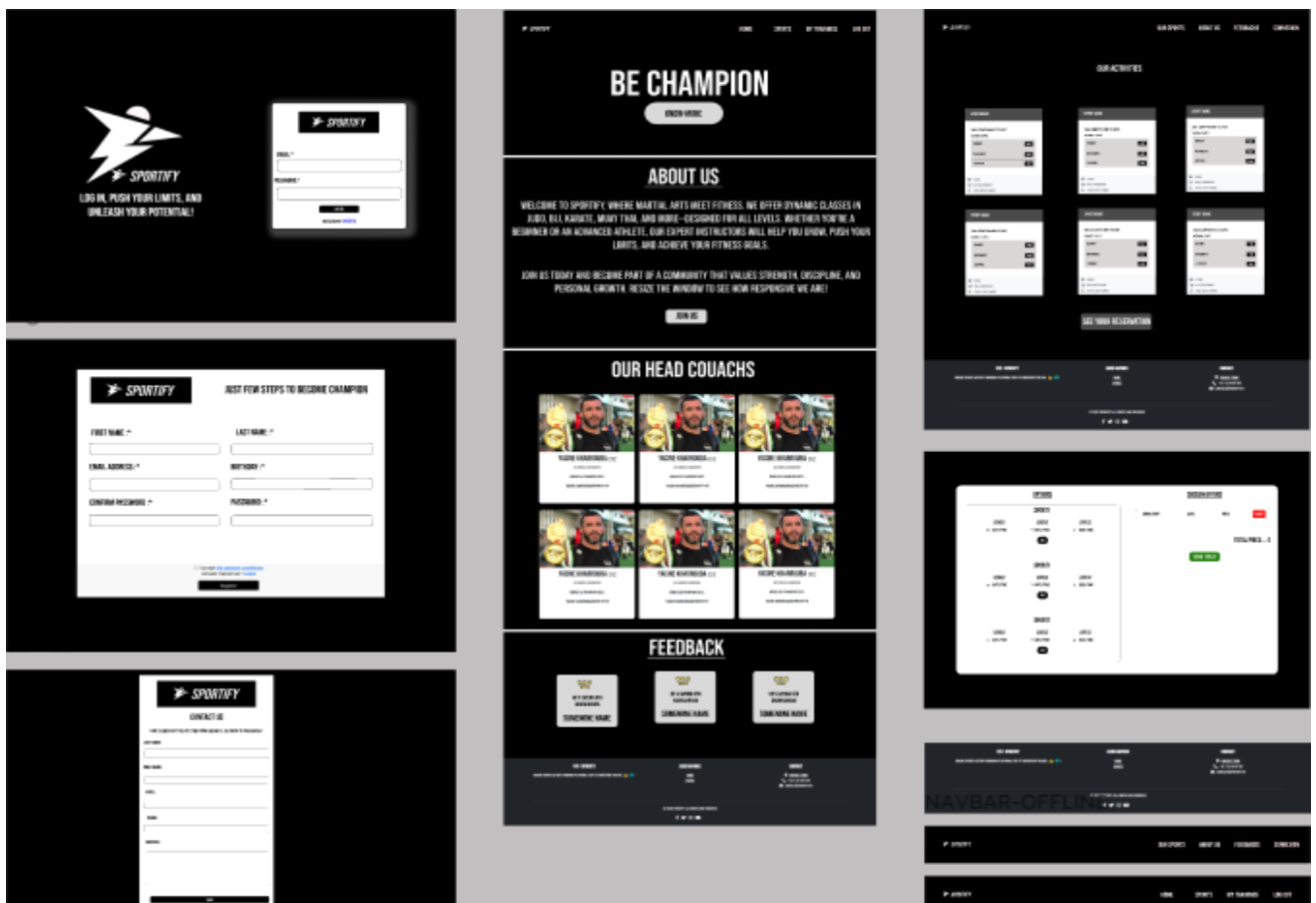
- **Vert : #008000**

Utilisé pour les messages de succès, indiquant clairement les actions réussies. Le vert est un choix standard pour les confirmations et les opérations validées.

- **Blue : #0000FF**

Utilisé pour les messages indiquant qu'une opération est en cours, afin de signaler à l'utilisateur que l'action est en traitement. Le bleu est une couleur généralement associée à l'attente et au progrès.

## Design général :



# Architecture du projet et arborescence des fichiers

## Schéma global de l'architecture

Le site **SPORTIFY** est conçu en suivant une architecture structurée et modulaire qui sépare clairement la partie frontend (client) et la partie backend (serveur). Cette séparation facilite le développement, la maintenance et la scalabilité de l'application. Voici un aperçu global de l'architecture :

### 1. Frontend (côté client) :

L'interface utilisateur est réalisée avec HTML pour la structure, CSS (et Bootstrap pour le responsive) pour le style, et JavaScript pour dynamiser l'expérience utilisateur. Tous ces éléments se retrouvent dans le dossier assets.

### 2. Backend (côté serveur) :

La logique métier est gérée en PHP, tandis que MySQL assure le stockage et la gestion des données. La communication entre le serveur et la base de données se fait via un fichier de configuration dédié, et des scripts SQL permettent de créer et de gérer la base de données.

### 3. Envoi d'emails :

L'intégration de PHPMailer, téléchargé depuis GitHub, permet de gérer l'envoi des mails (confirmation d'inscription, récapitulatifs des réservations, devis, etc.), assurant ainsi une communication automatisée avec les utilisateurs.

## Arborescence et analyse des fichiers :

### 1. SPORTIFY :

#### a. assets :

i. **css** : Les fichiers CSS regroupent les styles personnalisés et les éventuelles surcharges de Bootstrap, assurant une présentation cohérente et responsive du site.

ii. **js** : Les scripts JavaScript sont responsables de la manipulation du DOM, de la validation des formulaires et de l'animation pour améliorer l'interactivité et l'ergonomie.

**iii. images :** Ce dossier contient toutes les ressources graphiques (logos, illustrations, bannières, etc.) qui agrémentent les différentes pages du site.

**iv. video :** Les fichiers vidéo, souvent utilisés pour des présentations ou des promotions, se trouvent dans ce répertoire.

**b. config :**

**i. database\_connexion.php :** Ce script centralise la configuration de la base de données en regroupant les identifiants (serveur, base de données, utilisateur, mot de passe) et permet d'établir la connexion nécessaire pour l'accès aux données dans les autres scripts PHP.

**c. database :**

**i. creation\_bdd.sql :** Ce fichier SQL contient toutes les instructions nécessaires à la création et la configuration de la base de données, incluant la création des tables, des index, et la définition des relations entre les différentes entités.

**d. includes :**

**i. login.php :** Gère l'authentification des utilisateurs en vérifiant leurs identifiants lors de la connexion.

**ii. logout.php :** Permet de déconnecter les utilisateurs de manière sécurisée.

**iii. register.php :** Facilite la création de nouveaux comptes en traitant et en stockant les informations saisies lors de l'inscription.

**iv. reservation.php :** Traite la réservation des cours, en mettant à jour la base de données avec les inscriptions des utilisateurs.

**v. cancel\_reservation.php :** Permet l'annulation d'une réservation, modifiant ainsi les informations stockées dans la base de données.

**vi. contact\_form.php :** Gère les soumissions du formulaire de contact, en procédant à la validation des données et à la gestion des demandes des visiteurs.

**vii. devis\_mail.php :** S'occupe de la demande de devis personnalisé en traitant les informations saisies par l'utilisateur et en envoyant un récapitulatif par mail via l'outil PHPMailer.

**e. pages :**

**i. home.php :** La page d'accueil du site présente l'ensemble des fonctionnalités, les services proposés et éventuellement des témoignages clients.

**ii. page\_login.php :** Interface dédiée à la connexion des utilisateurs au site.

**iii. page\_register.php :** Interface permettant aux nouveaux utilisateurs de s'inscrire sur la plateforme.

**iv. sports.php :** Liste et présente les différentes activités et cours sportifs disponibles, avec la possibilité d'y souscrire pour les utilisateurs connectés.



**v. contact.php** : Contient le formulaire de contact et les coordonnées pour joindre le service client.

**vi. page\_devis.php** : Propose un formulaire permettant aux utilisateurs de demander un devis personnalisé pour des services spécifiques.

**vii. my\_reservations.php** : Affiche un récapitulatif des réservations effectuées par l'utilisateur connecté, avec tous les détails nécessaires.

**f. PHPMailer :**

Ce dossier contient la librairie PHPMailer, qui est utilisée pour l'envoi automatisé des emails (confirmations, rappels, devis, etc.) et améliore la gestion de la communication par email sur le site.

## Fonctionnalité clé du site

### Inscription des utilisateurs :

Les visiteurs peuvent créer un compte en remplissant un formulaire. Le système vérifie que l'adresse e-mail n'est pas déjà enregistrée dans la base de données. Les informations sont ensuite stockées pour permettre une authentification sécurisée.

### Connexion et déconnexion :

Les utilisateurs existants peuvent se connecter avec leur adresse e-mail et mot de passe. Une fois connectés, ils accèdent à des fonctionnalités réservées. Un bouton "Déconnexion" est disponible dans la barre de navigation pour terminer la session.

### Réservation de cours sportifs :

Si l'utilisateur est connecté, il peut directement réserver un cours sportif parmi ceux proposés. Dans ce cas, après avoir sélectionné le cours souhaité, le système enregistre la réservation dans la base de données et envoie un email de confirmation avec le récapitulatif de la réservation. En revanche, si l'utilisateur n'est pas connecté, il pourra uniquement consulter les détails des cours. Dès qu'il choisit un cours, il est automatiquement redirigé vers la page de connexion afin de lui permettre de s'authentifier avant de finaliser sa réservation.

### Consultation des réservation :

L'utilisateur connecté peut accéder à la page "Mes réservations" pour consulter l'ensemble des cours sportifs qu'il a réservés. Cette page présente sous forme de liste les détails de chaque réservation (nom du cours, date, heure, coach, etc.), offrant une vue récapitulative de son planning.

## Annulation de réservation :

L'utilisateur a la possibilité d'annuler une réservation qu'il ne souhaite plus honorer. En cliquant sur le bouton "Cancel" associé à une réservation, le système supprime l'enregistrement correspondant dans la base de données, libérant ainsi la place pour d'autres utilisateurs.

## Formulaire de contact :

Une page dédiée permet aux visiteurs, qu'ils soient connectés ou non, de **contacter facilement le service client de SPORIFY**. Grâce à un formulaire simple et clair, l'utilisateur peut renseigner son nom, son adresse email et son message (question, remarque, demande d'information, etc.).

Après la soumission, les données saisies sont **vérifiées** (au niveau client et serveur) pour garantir leur validité. Une fois validées, ces informations sont automatiquement **transmises par email à l'adresse fictive** du service client :

*sportify.contact.87@mail.com*, en utilisant la bibliothèque **PHPMailer**.

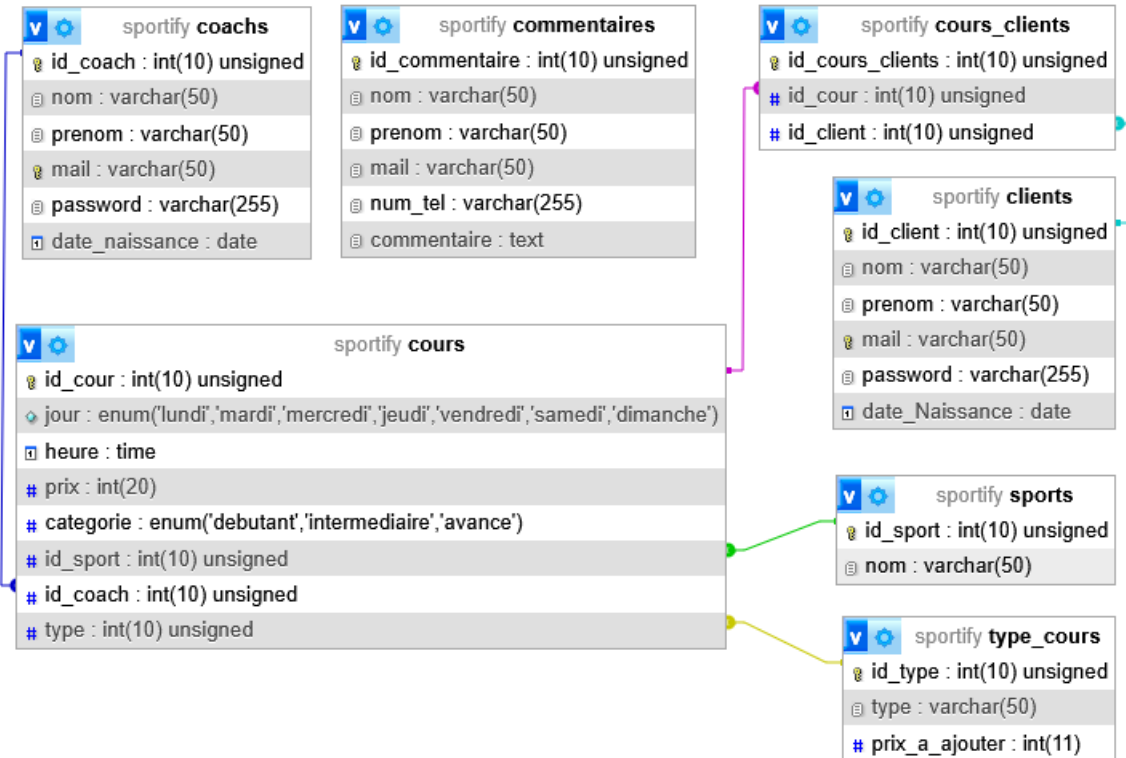
Cette fonctionnalité assure une **communication rapide, fluide et accessible** entre les utilisateurs et l'équipe de la plateforme, sans avoir besoin de passer par des outils externes. Elle améliore également le **service après-vente** en permettant aux utilisateurs d'obtenir rapidement des réponses à leurs préoccupations.

## Demande de devis personnalisé :

L'utilisateur peut demander un devis en remplissant un formulaire avec ses besoins spécifiques . Une estimation est générée et envoyée par e-mail grâce à PHPMailer.

# Architecture de base de données

## Structure de la base de données :



La base de données est appelée **SPORTIFY** et elle contient plusieurs tables qui interagissent entre elles. Ces tables sont organisées de manière logique pour gérer les différents aspects du site : utilisateurs, coaches, sports, types de cours, et plus encore.

### a. Table clients

- **Objectif** : Cette table enregistre les informations des utilisateurs (clients) inscrits sur le site.
- **Colonnes** :
  - **id\_client** : Identifiant unique pour chaque client.
  - **nom et prenom** : Informations personnelles de l'utilisateur.
  - **mail** : L'email de l'utilisateur, qui est unique.
  - **password** : Mot de passe pour l'authentification.
  - **date\_Naissance** : Date de naissance de l'utilisateur pour vérifier son âge et personnaliser les recommandations.

#### **b. Table coaches**

- **Objectif** : Cette table contient les informations des coaches, responsables des cours.
- **Colonnes** :
  - **id\_coach** : Identifiant unique pour chaque coach.
  - **nom et prenom** : Informations personnelles du coach.
  - **mail** : Email unique pour chaque coach.
  - **password** : Mot de passe pour l'authentification du coach.
  - **date\_naissance** : Date de naissance du coach.

#### **c. Table commentaires**

- **Objectif** : Permet aux clients de laisser des commentaires sur les cours, les coaches, ou les services en général.
- **Colonnes** :
  - **id\_commentaire** : Identifiant unique pour chaque commentaire.
  - **nom, prenom, mail, num\_tel** : Informations personnelles pour identifier l'auteur du commentaire.
  - **commentaire** : Le contenu du commentaire.

#### **d. Table sports**

- **Objectif** : Liste des sports proposés par le site.
- **Colonnes** :
  - **id\_sport** : Identifiant unique pour chaque sport.
  - **nom** : Nom du sport (par exemple, Judo, Boxe, Karaté, etc.).

#### **e. Table type\_cours**

- **Objectif** : Liste des types de cours (par exemple, collectif, privé, etc.) et les prix associés.
- **Colonnes** :
  - **id\_type** : Identifiant unique pour chaque type de cours.
  - **type** : Le type de cours (par exemple, "collectif").
  - **prix\_a\_ajouter** : Prix additionnel pour ce type de cours.

#### **f. Table cours**

- **Objectif** : Liste de tous les cours disponibles, avec des détails tels que le jour, l'heure, le prix, la catégorie (débutant, intermédiaire, avancé), le coach et le sport associé.

- **Colonnes :**

- **id\_cour** : Identifiant unique pour chaque cours.
- **jour** : Le jour de la semaine où le cours est proposé.
- **heure** : L'heure à laquelle le cours commence.
- **prix** : Prix du cours.
- **categorie** : Niveau du cours (débutant, intermédiaire, avancé).
- **id\_sport** : Identifiant du sport associé au cours (clé étrangère).
- **id\_coach** : Identifiant du coach responsable de ce cours (clé étrangère).
- **type** : Type de cours (clé étrangère vers la table type\_cours).

**g. Table cours\_clients**

- **Objectif** : Table de liaison entre les clients et les cours auxquels ils sont inscrits.

- **Colonnes :**

- **id\_cours\_clients** : Identifiant unique pour chaque enregistrement.
- **id\_cour** : Identifiant du cours (clé étrangère).
- **id\_client** : Identifiant du client (clé étrangère).

## Relations entre tables :

La structure de la base de données repose sur des relations entre les tables, ce qui permet une organisation cohérente des données.

- **Clients et Cours** : Les clients peuvent s'inscrire à plusieurs cours via la table `cours_clients`, ce qui crée une relation *plusieurs à plusieurs* entre clients et cours.
- **Coachs et Cours** : Chaque cours est dirigé par un coach, donc une relation *un à plusieurs* existe entre coachs et cours.
- **Sports et Cours** : Chaque cours est associé à un sport spécifique, ce qui crée une relation *un à plusieurs* entre sports et cours.
- **Types de Cours et Cours** : Les cours peuvent avoir différents types (par exemple, collectif ou privé), ce qui crée une relation *un à plusieurs* entre `type_cours` et cours.

## Principes d'architecture :

### a. Normalisation :

Les données sont bien normalisées, avec des tables dédiées pour les entités principales (clients, coaches, sports, etc.) et des tables de liaison pour gérer les relations. Cela permet de minimiser la duplication des données et d'assurer une gestion efficace des informations.

**b. Sécurité :** Les mots de passe sont stockés dans un format crypté (VARCHAR(255)), ce qui est important pour la sécurité des utilisateurs. Il serait cependant recommandé d'utiliser un algorithme de hachage sécurisé plutôt que de stocker les mots de passe en texte brut.

### c. Scalabilité :

L'architecture est suffisamment flexible pour gérer une augmentation du nombre d'utilisateurs, de coaches, de sports, et de cours. En ajoutant simplement de nouvelles entrées dans les tables appropriées, la base peut évoluer facilement.

# Sécurité de l'application :

## Protection contre SQL Injection :

Pour protéger le site contre les attaques de type **SQL Injection**, une stratégie de sécurisation a été mise en place au niveau du fichier **database\_connexion.php** . Ce fichier contient une fonction nommée **executeQuery()** , conçue pour exécuter les requêtes SQL de manière sécurisée.

- **Fonction executeQuery() :**

Cette fonction prend deux paramètres :

- Une **chaîne de caractères** représentant la requête SQL avec des **variables** ( :XXX ) à la place des valeurs dynamiques.
- Une **liste de paramètres** qui seront liés à ces variables au moment de l'exécution

- **Avantages de cette méthode :**

- Cette méthode protège mieux le site contre les attaques SQL grâce à l'utilisation de requêtes préparées, qui empêchent un pirate d'injecter du code malveillant.
- Le code est plus clair et plus facile à comprendre ou à modifier plus tard.
- On peut réutiliser les requêtes facilement dans plusieurs endroits du site, sans devoir tout réécrire.

## Sécurité des mots de passe :

Pour protéger les comptes utilisateurs, nous utilisons les fonctions natives de PHP :

- **password\_hash()** : pour chiffrer les mots de passe au moment de l'inscription,
- **password\_verify()** : pour vérifier les mots de passe lors de la connexion, tout en gardant les données sécurisées.

En plus de cela, nous avons mis en place des **règles de création de mot de passe sécurisé** :

- Minimum de 8 caractères
- Au moins une lettre majuscule
- Au moins un chiffre
- Au moins un caractère spécial
- Au moins une lettre minuscule

# Cahier des charges pour le projet Sportify

## Répartition des taches :

### Tâches LAGGOUN Khaled :

- UI / UX Design.
- Page d'accueil.
- Système d'authentification pour les pages connexion et inscription.
- Le formulaire de demande un devis.

### Tâches de GUERROUF Abdeldjalil :

- Conception de la base de données (BDD).
- Page sports avec le système de réservation.
- Page mes réservations
- Formulaire de contact.
- Système de messagerie avec **PHPMailer**



# Guide d'installation et de configuration du site sur votre machine

## Téléchargement du projet depuis GitHub :

- Rendez-vous sur le dépôt GitHub du projet <https://github.com/lgnkhaed/Sportify> et téléchargez l'archive ZIP contenant tous les fichiers du site.
- Une fois le fichier téléchargé, décompressez-le dans un dossier de votre choix.

## Installation de XAMPP :

- Téléchargez et installez XAMPP (compatible avec Windows, macOS ou Linux) depuis le site officiel.
- Ouvrez le panneau de contrôle de XAMPP et démarrez les modules **Apache** et **MySQL**.

## Configuration de la base de données :

- Ouvrez votre navigateur et accédez à **phpMyAdmin** (généralement via <http://localhost/phpmyadmin>).
- Dans l'onglet **Import**, sélectionnez le fichier SQL situé dans le dossier database (/SPORTIFY/database/sportify.sql ) et importez-le pour créer la structure de la base de données.

## Installation du site dans XAMPP :

- Copiez l'intégralité du dossier du projet (contenant les dossiers assets, config, database, includes, pages, etc.) dans le dossier htdocs de XAMPP.  
Par exemple, sur Windows, le chemin sera généralement :  
C:\xampp\htdocs\Sportify.

## Lancement du site :

- Dans votre navigateur, accédez à l'URL :  
<http://localhost/Sportify/pages/home.php>