



# Cahier des charges "..."

▼ Version	V0
☰ Type	Exercice
📅 Date de création	@18 mars 2024 10:08
📅 Dernière modification	@18 mars 2024 13:32

## Spécifications fonctionnelles

Vous êtes libre de choisir le thème de votre site, cela peut être celui d'une agence e-sport, d'un restaurant, ou encore d'une salle de sport, peu importe ! **Vous devez néanmoins respecter le cahier des charges suivant.**

### Liste des fonctionnalités attendues :

- Système d'authentification de l'utilisateur
- Système de réservation, quel qu'ils soient (sous-entendu, avoir toutes les méthodes du CRUD pour l'un de vos modèles).

💡 En manque d'idées ? Pourquoi ne pas créer le site de votre salle de sport, offrant la possibilité de réserver une séance avec un coach ? Un cinéma présentant ses différents films et la possibilité de réserver une place ? Ou encore un restaurant exposant ses plats et offrant la possibilité de réserver une table pour dîner ?

Vous êtes libres d'imaginer l'application que vous désirez, appuyez-vous sur une bonne conception de base de données ainsi qu'une maquette efficace.

## Authentification

Un système d'authentification complet est mis à la disposition de l'utilisateur pour que celui-ci puisse créer une liste de tâches privée, accessible uniquement avec ses identifiants.

Nom du champ	Type du champ	Commentaire
Nom	Texte	requis, 3 min - 50 max.
Prénom	Texte	non requis, 3 min - 50 max.
Mot de passe	Mot de passe	requis, 7 min.
Email	Email	requis, 3 min - 80 max.

### 1. Inscription

Une vérification des champs est effectuée à la soumission du formulaire. Chaque champ de saisie affiche une erreur adaptée à l'erreur rencontrée. Les messages d'erreurs doivent disparaître lorsque le formulaire est à

nouveau soumis avec les champs valides.

Lorsque le formulaire est soumis, une requête **POST** contenant les données du futur utilisateur est envoyée au backend. Ce dernier vérifie et traite les données. Une fois les données traitées, l'utilisateur est redirigé en tant qu'utilisateur authentifié si l'inscription s'est correctement déroulée.

- ☐ En tant qu'utilisateur, je peux créer un compte.
- ☐ En tant qu'utilisateur, je suis redirigé après une inscription réussie.

## 2. Connexion

---

Une vérification des champs est effectuée à la soumission du formulaire. Chaque champ de saisie affiche une erreur adaptée à l'erreur rencontrée. Les messages d'erreurs doivent disparaître lorsque le formulaire est à nouveau soumis avec les champs valides.

Lorsque le formulaire est soumis, une requête **POST** contenant les données personnelles de l'utilisateur est envoyée au backend. Ce dernier vérifie et traite les données. Une fois les données traitées, l'utilisateur est redirigé tant qu'utilisateur authentifié si la connexion s'est correctement déroulée.

- ☐ En tant qu'utilisateur, je peux me connecter à l'aide de mon duo email / mot de passe.
- ☐ En tant qu'utilisateur, je suis redirigé après une connexion réussie.

## 3. Modification du profil

---

Une vérification des champs est effectuée à la soumission du formulaire. Chaque champ de saisie affiche une erreur adaptée à l'erreur rencontrée. Les messages d'erreurs doivent disparaître lorsque le formulaire est à nouveau soumis avec les champs valides.

Lorsque le formulaire est soumis, une requête **POST** contenant les données personnelles mises à jour est envoyée au backend. Ce dernier vérifie et traite les données. À l'issue du traitement, l'utilisateur est informé si la mise à jour s'est correctement déroulée.

Si l'utilisateur souhaite modifier son mot de passe, il doit le saisir deux fois, comme lors de l'inscription, afin de s'assurer qu'il n'a pas fait d'erreur lors de la première saisie.

- ☐ En tant qu'utilisateur, j'ai accès à une page de profil.
- ☐ En tant qu'utilisateur, je peux modifier mes informations personnelles.
- ☐ En tant qu'utilisateur, je peux modifier mon mot de passe.

## 4. Suppression du compte

---

Si l'utilisateur souhaite supprimer son compte, une requête **POST** est envoyée au backend, lequel traite la requête et, si la suppression s'est correctement déroulée, redirige l'utilisateur sur la page d'inscription où un message l'informera du bon déroulement de son action et du regret que nous avons de le voir partir.

- ☐ En tant qu'utilisateur, je peux supprimer mon compte.
- ☐ En tant qu'utilisateur, je suis redirigé vers la page d'inscription après une suppression réussie.

# Spécifications techniques

- HTML
- Bootstrap
- Javascript
- PHP + PDO
- MySQL

Vous utiliserez le framework CSS **Bootstrap** pour accélérer l'intégration de l'application grâce à son API de classes CSS ainsi que les composants mis à disposition.

Pour éviter d'avoir à installer **Bootstrap** sur votre projet, vous devrez l'intégrer à celui-ci par le biais du CDN mis à votre disposition.

**Liste des composants Bootstrap intéressants (non exhaustive) :**

- Navbar - barre de navigation.
- List group - liste d'item.
- Form - mise en page des formulaires.
- Badge - petit label de couleur.
- Modal - fenêtre popup en position absolue.
- Toast - petite fenêtre de notification.
- Tooltips - information au survol de l'élément.

## Base de données

La base de données doit contenir au minimum :

- La table des utilisateurs
- Une table représentant une réservation

La table représentant une réservation doit avoir minimum deux relations autres que celle avec l'utilisateur :

- une Many to One de votre choix
- une Many to Many de votre choix

💡 Si vous n'avez pas d'idées : un réservation de séance de sport peut avoir une relation avec la salle dans laquelle la séance se déroulera ainsi qu'avec un type de séance (réservation pour un cours de Yoga dans la salle n°6)