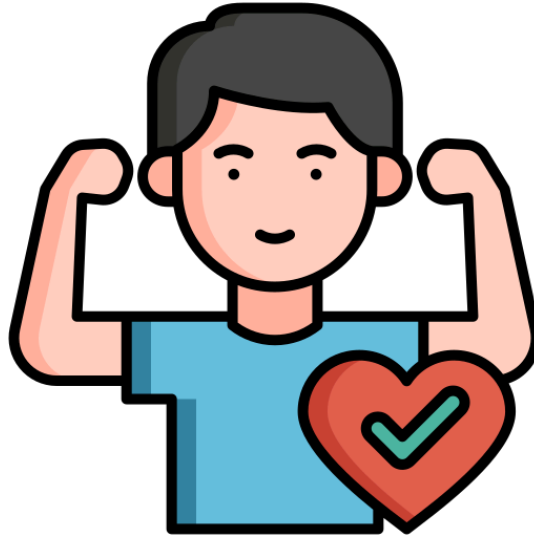


Projet Health

Interfaces utilisateurs et composants



Sujet.....	2
API REST.....	3
Création de compte Professionnel de santé.....	4
Création de compte Patient.....	5
Spécifications fonctionnelles.....	6
Fonctionnalités optionnelles.....	7
Consignes et modalités de rendu.....	8
Critères d'évaluation.....	8

Sujet

Votre projet consiste à réaliser une application web de type **SPA** (*Single Page Application*) avec la bibliothèque **React.js** (<https://react.dev/>) et le langage **TypeScript**, en lien avec les thématiques de l'**IoT**, du **Big Data** et de l'**E-Santé**.

- **IoT (Internet of Things)** : recueil et affichage de données physiologiques, géographiques, d'activité physique...
- **E-Santé** : recommandations, alertes, suivi... communiqués à partir de données obtenues via un ou plusieurs objets connectés (ex: montre connectée, balance connectée...) puis valorisées (statistiques, prévisions, tendances...).
- **Innovation** : votre application devra proposer une innovation conceptuelle, technique, ergonomique, thématique, graphique ou logique... (au moins l'un des critères).

L'application s'adressera à un prescripteur (professionnel de la santé : médecin, nutritionniste... ou à un coach sportif) dont l'objectif est de suivre et conseiller des patients dans leur objectif de perte ou de gain de poids.

Les acteurs (patients, professionnels de santé, coach sportif...) et les données (physiques, psychiques, sportives) du système sont **fictifs**.

API REST

L'application proposera des fonctionnalités de **suivi d'activités physiques** et d'**évolution physiologique et psychique des patients** à partir de données fournies au format *JSON* par une **API REST** (<https://health.shrp.dev>).

Par défaut, les routes de l'API sont publiques.

Seules les routes mises en évidence sont privées et nécessitent la communication d'un jeton d'authentification *access_token* (header HTTP *authorization* en mode *bearer*).

- **Utilisateur** (coach sportif, médecin...) : **/user**
- **Patients** (données personnelles telles que le sexe, date de naissance, taille, poids de départ, objectif de poids...) : **/items/people**
- **Données physiologiques** (évolution du poids...) : **/items/physiologicalData**
- **données d'activités physiques** (nombre de pas, durée d'activité, calories consommées...) : **/items/physicalActivities**
- **données psychiques** (état psychologique du patient) : **/items/psychicData ***

(*) Endpoint nécessitant un *access_token* pour être consulté.

Création de compte Professionnel de santé

Pour créer un compte utilisateur de type "professionnel de santé" permettant d'accéder aux routes privées de l'API (ex: /items/psychicData), effectuez une requête HTTP **POST** sur le *endpoint* **/user** avec les données suivantes au format **JSON** :

```
{
  "role": "16317dcf-1e2f-4fba-969f-6f6b15ba1062",
  "email": "john@doe.com",
  "password": "azerty"
}
```

Pour un professionnel de santé, la valeur de l'attribut **role** doit toujours être **"16317dcf-1e2f-4fba-969f-6f6b15ba1062"**. Les valeurs des attributs **email** et **password** doivent être adaptées.

Création de compte Patient

Vous pouvez également créer un compte utilisateur de type *"patient"* en modifiant la valeur de l'attribut **role** : **"0777b505-b6cc-4308-9f17-c47fda65da29"**.

Attention : les données sont générées de façon aléatoire et peuvent contenir des incohérences.

Sur cette base commune, chaque groupe personnalisera son application en apportant une touche d'innovation d'ordre technique et/ou conceptuel et/ou ergonomique...

Vous pourrez utiliser des données complémentaires, réelles ou fictives, provenant d'autres *APIs* (existantes ou réalisées par vos soins) ou saisies par l'utilisateur.

Vous vous inspirerez des applications telles que **Apple Health, Google Fit, Nike Sport...**

Spécifications fonctionnelles

- L'application sera affichée dans le **navigateur web, en format mobile** (orientation portrait). Vous opterez pour une approche "*Mobile First*".
- **Composition de l'application à l'aide de plusieurs composants React**, conçus et implémentés par vos soins,
- Emploi du pattern ergonomique **Master / Details** (Master = liste d'items, Details = écran dédié à la présentation d'un item sélectionné),
- Navigation entre écrans à l'aide du module **react-router** (<https://reactrouter.com/en/main>),
- Emploi des **props** pour transmettre des données d'un composant React "*parent*" à un composant "*enfant*",
- Interactions avec l'*API REST* (avec **fetch** ou avec le module **Axios**, <https://axios-http.com/>),
 - **Authentification** d'un utilisateur,
 - Récupération de la **liste des patients** (écran Master),
 - Récupération des **détails d'un patient** à partir de son id (écran Details),
 - Récupération des **activités physiques** d'un patient,
 - Récupération de **l'évolution physiologique** d'un patient,
 - En option, récupération de **l'évolution psychique** d'un patient.
- Affichage de l'évolution physiologique ou des activités physiques d'un patient au moyen d'un diagramme à l'aide d'une bibliothèque spécialisée telle que **Recharts** (<https://recharts.org/en-US/>) ou toute autre représentation graphique de votre choix.
- Mise en place d'au moins un **formulaire avec le module react-hook-form** (<https://react-hook-form.com/>) permettant à l'utilisateur de se connecter (*Sign In*), saisir de données, effectuer une recherche dans les données...
- Emploi des **hooks** React **useState**, **useEffect**, **useContext**,
 - Gestion d'état avec **useState**,

- Partage de données avec ***useContext***,
- Initialisation des requêtes auprès de l'API avec ***useEffect***.
- Formulaire de connexion : ***Sign In*** (récupération de l'***access_token JWT***),
- Affichage de l'état de connexion de l'utilisateur courant dans l'UI (ex : "[john@doe.com](#) est connecté") ,
- **Bouton de déconnexion (*Sign Out*)** de l'utilisateur courant.

Fonctionnalités optionnelles

- Conservation de l'***access_token*** dans le ***localStorage*** dans la mémoire vive de l'application,
- Emploi du ***refresh_token*** pour obtenir un nouvel ***access_token*** et éviter la déconnexion de l'utilisateur courant,
- Interactions avec des ***routes privées de l'API*** en communiquant l'***access_token*** dans le header HTTP ***authorization*** en mode ***Bearer*** (ex : endpoint `/items/psychicData`).
- Intégration d'1 ou plusieurs autres ***API tierce(s)*** au projet.

Consignes et modalités de rendu

- Au sein de votre "groupe de niveau" (groupe 1 ou groupe 2), vous formerez un groupe de projet, de 3 à 4 personnes aux compétences complémentaires (analyse / conception, ergonomie, développement web...).
- **Délai de rendu** : au plus tard, dimanche 18/5/2025 (23h59). Un seul rendu par groupe.
- **Fournir un accès au dépôt git distant de votre projet** (*Github, Gitlab, Bitbucket...*), en veillant à ajouter alex@shrp.dev / shrp777 en tant que collaborateur, si le dépôt est privé.
- **Renseigner un fichier README.md** à la racine du projet où vous indiquerez le prénom et le nom de chaque membre du groupe.
- Fournir une **capture vidéo** (Commentaires audio non nécessaires) de l'application en fonctionnement (à intégrer au dépôt git) + un document synthétique de présentation du projet (concepts, fonctionnalités clé, points d'innovation...).

Critères d'évaluation

- Par défaut, la note sera identique pour l'ensemble du groupe pourra être individualisée en fonction de l'investissement et du travail continu.
- Travail continu (TD d'initiation à TypeScript, React...),
- Respect des consignes,
- État fonctionnel et qualité du code,
- Innovation,
- UI / UX,
- Comportement en cours (autonomie, intérêt, participation, assiduité, ponctualité, respect...).