

# SPORT MAP

Marseille sport

Accueil A propos Sports Carte

**Marseille Sport**

Trouvez votre terrain idéal en quelques clics !

[Carte des équipements](#)



**Les sports**

Découvrez tous les sports disponibles à Marseille

[Sports](#)

**Le projet**

Localisation, maintenance préventive collaborative, communauté

[A propos](#)

## **Sommaire**

- 1. Introduction personnelle**
- 2. Présentation du projet Sport Map**
- 3. Contexte et besoins identifiés**
- 4. Objectifs détaillés du projet**
- 5. Étapes de conception et de réalisation**
- 6. Compétences mobilisées (Référentiel CDA)**
- 7. Outils, technologies, architecture technique**
- 8. Problèmes rencontrés et solutions apportées**
- 9. Évaluation du projet et tests**
- 10. Perspectives et évolutions futures**
- 11. Conclusion et bilan professionnel**

## 1. Introduction personnelle

Depuis mon plus jeune âge, j'ai toujours été attiré par deux mondes qui, en apparence, n'ont rien à voir : **le numérique et le sport**. Né et ayant grandi à Marseille, j'ai grandi dans un environnement où la mer, les terrains de sport et la diversité culturelle forment un cadre de vie vivant, parfois difficile, mais toujours inspirant. Très tôt, j'ai appris à m'exprimer à travers la pratique sportive et les outils numériques. L'un m'offrait un exutoire physique, l'autre un espace de création et de compréhension du monde.

Après avoir expérimenté différents métiers et domaines, je me suis orienté naturellement vers le **développement web et applicatif**. J'y ai trouvé une discipline à la fois logique et créative, rigoureuse et ouverte. J'ai rapidement compris que le code n'était pas une simple succession d'instructions, mais un langage permettant de **traduire des idées en solutions concrètes**. C'est ce qui m'a poussé à intégrer la formation **Concepteur Développeur d'Applications (CDA)** au sein de **La Plateforme**, une école du numérique basée à Marseille, résolument tournée vers l'insertion professionnelle et l'apprentissage par projet.

Cette formation a constitué un véritable tournant. Non seulement elle m'a permis de consolider mes acquis en développement (HTML, CSS, JavaScript, React, PHP, SQL...), mais elle m'a surtout appris à structurer un projet, à analyser des besoins, à concevoir une architecture logicielle et à produire une application viable. Le format intensif, orienté vers l'autonomie et la professionnalisation, m'a confronté à la réalité du métier de développeur : travailler en équipe, apprendre continuellement, livrer dans les délais, documenter son travail, et s'adapter.

C'est dans ce contexte qu'est née l'idée de mon projet de fin de formation : **SportMap**. Ce projet n'est pas un simple exercice technique. Il est le fruit de mes **valeurs personnelles**, de mon **parcours de vie**, et d'un **constat social** : l'accès à la pratique sportive est encore trop inégalitaire. Beaucoup de personnes, en particulier dans les quartiers populaires ou les zones rurales, ignorent les infrastructures sportives disponibles autour d'eux. D'autres ne savent pas si ces équipements sont accessibles,

fonctionnels ou sûrs. Et les collectivités, de leur côté, manquent souvent de remontées terrain fiables pour les entretenir ou les valoriser.

Avec SportMap, j'ai voulu proposer **une solution numérique simple, intuitive et utile** pour toutes et tous. Une carte interactive, collaborative, qui permet à chacun de **localiser rapidement les équipements sportifs publics** autour de soi, de **connaître leur état**, et de **contribuer à l'amélioration de ces espaces communs**. Ce projet m'a permis de mettre en pratique les compétences techniques acquises pendant la formation, mais aussi de développer une vision plus large de ce que peut être le rôle d'un développeur : un **acteur du changement**, capable de créer des outils à impact.

Au-delà du code, j'ai appris à réfléchir comme un concepteur, à écouter les besoins des utilisateurs, à prendre en compte les contraintes techniques, juridiques et sociales. J'ai travaillé avec des outils professionnels (Git, Jira, Confluence, API REST, Mapbox, React, Tailwind, Django...), structuré mes tâches en sprints, rédigé de la documentation, mis en place des tests, corrigé des bugs, et surtout, j'ai été **confronté à mes propres limites, que j'ai appris à dépasser**.

Ce dossier est donc le reflet de mon **parcours de transformation professionnelle**. Il témoigne de mon engagement, de ma progression, et de ma volonté de construire des solutions durables, accessibles et pertinentes. Il marque aussi une étape vers un futur que j'espère riche en collaborations, en apprentissages, et en projets porteurs de sens.

## 2. Projet SportMap

### Contexte et historique du projet

Le projet SportMap a été initié dans le cadre de la formation Concepteur Développeur d'Applications (CDA) dispensée par l'organisme LaPlateforme. Durant cette formation, mon camarade de promotion et moi-même avons constaté qu'il existait un manque de visibilité concernant les infrastructures sportives publiques accessibles en libre-service dans de nombreuses communes françaises. Cette observation nous a motivés à proposer une solution concrète. Ainsi, SportMap est né de l'idée de répertorier et de cartographier de manière intelligente l'ensemble des équipements sportifs publics en France, afin d'en faciliter la découverte par les citoyens. Ce travail initial s'inscrit dans une démarche citoyenne et participative, visant à améliorer l'accès à l'information sportive pour tous.

### Objectif général du projet

L'objectif principal du projet SportMap est de fournir aux citoyens une plateforme centralisée pour trouver facilement les équipements sportifs publics autour d'eux. Pour atteindre cet objectif, nous avons développé une **application Web responsive** complétée par une **application mobile** dédiée. SportMap permet de visualiser sur une carte interactive l'ensemble des infrastructures sportives publiques d'une zone donnée. Il s'agit de cartographier intelligemment ces équipements afin de rendre accessibles les informations (localisation, type d'équipement, etc.) de manière claire et actualisée. Cette solution vise à promouvoir la pratique du sport en facilitant l'accès à l'information, en particulier pour les personnes souhaitant découvrir les installations proches de chez elles.

### Description de la solution

La solution SportMap se présente sous la forme d'une application Web responsive, accompagnée d'une version mobile. L'interface se veut simple et intuitive, afin d'assurer une prise en main rapide par tout utilisateur. Elle intègre plusieurs fonctionnalités clés :

- **Carte interactive** : permet de visualiser les équipements sportifs sur une carte numérique. L'utilisateur peut zoomer et naviguer pour explorer les installations selon la zone géographique choisie.

- **Géolocalisation** : localise l'utilisateur en temps réel pour afficher les équipements les plus proches, simplifiant ainsi la recherche.
- **Filtres par type d'équipement** : offre la possibilité de filtrer les résultats par catégorie (stades, gymnases, terrains multisports, piscines, etc.), pour affiner la recherche en fonction de la pratique sportive souhaitée.
- **Signalement collaboratif** : donne aux citoyens les moyens de contribuer au projet. Ils peuvent proposer l'ajout de nouveaux équipements ou signaler des modifications (changement d'adresse, fermeture, nouvelle installation, etc.), ce qui enrichit continuellement la base de données.
- **Interface simple et épurée** : conçue pour être intuitive, l'interface s'adapte à tous les supports (ordinateurs, tablettes, smartphones) et se focalise sur l'essentiel pour faciliter la navigation (menus clairs, icônes explicites, temps de réponse rapide).

## État actuel du projet

La version 1 (V1) de SportMap est aujourd'hui opérationnelle et accessible en ligne. Elle inclut les éléments suivants :

- **Version 1 opérationnelle** : l'application Web est en production et consultable par les utilisateurs. Une version mobile basique est également disponible en mode progressive Web App.
- **Base de données initiale** : environ **5 800 équipements sportifs** publics sont référencés sur la carte à ce jour. Ces données ont été collectées à partir de sources publiques et de contributions citoyennes.
- **Interface réactive** : l'application affiche rapidement les résultats et s'adapte aux différents écrans (ordinateurs et mobiles), garantissant ainsi une bonne expérience utilisateur.
- **Évolutions en cours** : des fonctionnalités complémentaires sont en cours de développement pour les prochaines versions. Par exemple, l'amélioration du système de filtrage avancé et l'optimisation du module de signalement collaboratif sont en cours d'implémentation. Nous travaillons également à l'enrichissement permanent de la base de données (ajout de nouveaux équipements, vérification des informations existantes).

## Objectifs techniques et sociaux

SportMap poursuit des objectifs à la fois techniques et sociaux :

- **Sur le plan technique**, l'ambition est de développer une application robuste et évolutive. Nous privilégions une architecture modulaire et l'exploitation de technologies Web modernes (frameworks JavaScript, API de cartographie, etc.) pour assurer la maintenabilité et la performance du système. L'utilisation de **données ouvertes (Open Data)** publiques garantit par

ailleurs la qualité et la mise à jour régulière des informations.

- **Sur le plan social**, le projet vise à **rendre le sport plus accessible au plus grand nombre**. En centralisant les informations sur les équipements disponibles, SportMap simplifie la découverte des installations de sport. De plus, le projet cherche à **favoriser la participation citoyenne** : chaque utilisateur peut contribuer directement à l'amélioration de la carte en ajoutant ou corrigeant des données. Enfin, SportMap souhaite **soutenir les collectivités locales** en leur offrant un outil de visualisation et de suivi des infrastructures sportives existantes. Les communes peuvent ainsi mieux connaître leur patrimoine sportif et valoriser leurs équipements auprès des habitants.

## Spécificités du projet

SportMap se distingue par plusieurs spécificités notables :

- **Dimension citoyenne** : l'application est conçue comme une plateforme participative. Les utilisateurs jouent un rôle actif en signalant de nouveaux équipements ou en validant les informations existantes. Cette approche collaborative renforce l'engagement des citoyens dans le développement du projet et assure une mise à jour dynamique des données.
- **Approche Open Data** : le projet tire parti des données ouvertes disponibles (bases municipales, plateformes nationales, etc.) pour initialiser et compléter le référentiel des équipements sportifs. Cette utilisation de l'Open Data permet d'assurer la transparence des informations et de faciliter l'actualisation régulière de la carte.
- **Potentiel d'extension nationale** : bien que le projet ait débuté à une échelle locale (zones pilotes ou secteurs limités), son architecture modulaire et la nature des données permettent une montée en charge progressive. SportMap est conçu pour pouvoir être déployé à l'échelle nationale, couvrant à terme l'ensemble du territoire français.

À ce stade, SportMap reste un projet indépendant : **aucun partenariat institutionnel n'a encore été formalisé**. Cette autonomie garantit la neutralité des données et fait de SportMap une initiative citoyenne libre d'accès. Néanmoins, la structure open data du projet laisse la porte ouverte à de futurs partenariats avec des collectivités locales ou des organismes sportifs, qui pourraient renforcer la diffusion de l'outil et la qualité des données.

## 3. Contexte et besoins identifiés

### Une pratique sportive libre en plein essor, mais peu structurée

En France, on observe depuis plusieurs années une montée en puissance de la **pratique sportive en autonomie**, notamment dans les espaces publics : city-stades, terrains multisports, skateparks, parcours

santé, terrains de football ou de basket en accès libre, etc. Cette tendance est particulièrement forte chez :

- les jeunes,
- les personnes précaires ou éloignées des clubs traditionnels,
- les pratiquants amateurs,
- les sportifs occasionnels ou en reprise.

La crise sanitaire liée au Covid-19 a renforcé cette dynamique : nombreux sont ceux qui ont adopté une pratique sportive plus libre, sans inscription dans une structure, favorisant les lieux en extérieur.

Cependant, malgré cette évolution des usages, **aucun outil numérique centralisé ne permet aujourd'hui d'accéder facilement à une carte exhaustive et fiable des équipements sportifs en libre accès.**

---

### Un manque de visibilité sur les infrastructures disponibles

Les collectivités locales mettent souvent à disposition des équipements, mais ces derniers sont :

- **mal référencés**,
- **dispersés sur différents sites**, ou noyés dans des documents PDF peu lisibles,
- **inconnus du grand public**.

Les usagers doivent alors compter sur le bouche-à-oreille ou explorer leur environnement de manière aléatoire. Cela crée une **inégalité d'accès à l'information**, qui peut renforcer les inégalités sociales et territoriales.

Par exemple, un jeune souhaitant jouer au basket près de chez lui doit souvent passer par des groupes Facebook locaux, poser des questions sur des forums, ou se déplacer à l'aveugle. Cette perte de temps et d'énergie est un frein direct à la pratique du sport libre.

---

### Une absence d'outil cartographique collaboratif dédié au sport

Il existe bien des outils cartographiques spécialisés (comme Google Maps, OpenStreetMap ou Géoportail), mais **aucun n'est dédié exclusivement à la pratique sportive libre et collaborative**. Par ailleurs :

- Google Maps n'affiche que partiellement les équipements et sans distinction sport/sport.
- OpenStreetMap est très technique et non grand public.
- Les applications sportives comme Decathlon Outdoor ou Strava ciblent surtout le sport individuel ou de performance (randonnée, running, vélo), pas les équipements collectifs.

Il existe donc un **véritable vide entre les besoins des usagers et les solutions numériques actuelles**.

---

### Besoin exprimé d'un outil simple, communautaire et fiable

Dans le cadre de l'idéation du projet, plusieurs échanges informels avec des pratiquants réguliers de sport libre (notamment à Marseille) ont mis en évidence des besoins récurrents :

- “*Je ne sais pas où sont les terrains de foot en bon état dans mon quartier.*”
- “*Y'a trop de monde à tel endroit, mais je n'ai aucun moyen de le savoir avant d'y aller.*”
- “*Je suis nouveau dans la ville, je ne sais pas où aller m'entraîner.*”
- “*On pourrait tous gagner du temps si on avait une appli qui les référence bien.*”

Ces témoignages montrent un **besoin fort de centralisation, de clarté, et de participation communautaire**. Les usagers veulent pouvoir contribuer à l'amélioration de l'information (ajout d'un terrain, signalement de dégradations), sans dépendre uniquement de l'action des institutions.

---

### Besoins techniques identifiés au démarrage du projet

Pour répondre à ces enjeux, le projet devait donc intégrer plusieurs caractéristiques techniques dès le départ :

- Une **carte interactive rapide et intuitive**, avec navigation fluide.
- Une **base de données fiable**, avec possibilité d'évolution vers l'open data.
- Une **authentification légère** mais sécurisée, pour les contributions citoyennes.
- Un **design mobile-first**, étant donné que la majorité des utilisateurs utiliseront l'appli depuis leur téléphone.
- Une **modularité** permettant d'ajouter d'autres couches fonctionnelles dans le futur (ex. affluence, notation, évènements...).

---

### Côté institutionnel : des communes démunies face à l'entretien et au suivi

Enfin, du côté des collectivités locales, le besoin est également réel. Nombre de communes ne disposent pas :

- d'une **vision claire** de l'état et de la fréquentation de leurs équipements sportifs,
- d'un **outil interactif à partager aux habitants**,
- d'un moyen de **faire remonter facilement les besoins de maintenance ou d'amélioration**.

SportMap pourrait donc à terme devenir un **outil d'aide à la décision**, un support d'analyse et de pilotage pour les mairies et directions des sports.

---

### En résumé

Le projet SportMap répond à des besoins identifiés sur plusieurs plans :

- **citoyen** : accès simplifié à l'information, meilleure autonomie sportive,
- **communautaire** : participation active des usagers à l'amélioration des données,
- **institutionnel** : outil potentiel de gestion et de valorisation des équipements publics,
- **technique** : solution légère, moderne et scalable.

## 4. Objectifs détaillés du projet

### Objectif principal

Le projet **SportMap** a pour objectif principal de **développer une solution numérique (web et mobile) permettant à tous les citoyens de localiser facilement les équipements sportifs publics accessibles en libre-service.**

Ce projet s'inscrit dans une logique de **service utile, accessible, évolutif et participatif**, dans le but de favoriser la pratique du sport pour tous, sans contraintes ni barrières sociales, géographiques ou technologiques.

---

### Objectifs généraux

#### 1. Cartographier intelligemment les équipements sportifs publics

- Référencer de manière centralisée les équipements accessibles en libre-service (terrains de basket, city-stades, tables de ping-pong, etc.).
- Utiliser les données ouvertes existantes (OpenData) pour garantir l'exactitude de la base initiale.

#### 2. Offrir une interface claire, intuitive et mobile-first

- Concevoir une interface simple d'utilisation, rapide et adaptée à tous les types d'utilisateurs (jeunes, familles, touristes...).
- Permettre une navigation fluide via une carte interactive.

### 3. Favoriser l'implication des citoyens dans l'entretien des données

- Permettre aux utilisateurs de signaler des équipements non référencés, fermés ou dégradés.
- Mettre en place un système de validation simple des contributions, garantissant la fiabilité des informations.

### 4. Développer un outil utile aux collectivités locales

- Offrir aux communes un moyen de visualiser leur patrimoine sportif.
- Proposer à terme une version back-office pour les agents municipaux (suivi, mises à jour, statistiques).



## Objectifs techniques

### 1. Concevoir une application web performante et évolutive

- Utilisation de technologies modernes (React, TailwindCSS, Mapbox, Django, etc.).
- Respect des bonnes pratiques de développement (responsive design, accessibilité, modularité).

### 2. Créer une base de données robuste et maintenable

- Structuration des données sportives en catégories (type, localisation, état, disponibilité...).
- Optimisation des requêtes cartographiques pour une expérience utilisateur fluide.

### 3. Implémenter une architecture sécurisée

- Mise en place d'un système d'authentification simple pour les contributeurs.
- Sécurisation des accès aux bases de données et aux points de contribution.

### 4. Développer une API interne pour l'évolutivité

- Centraliser les requêtes via une API REST facilitant les échanges entre les différentes couches de l'application (front, back, base de données).
  - Préparer le projet à une éventuelle extension vers une application mobile native.
- 

## Objectifs sociaux et d'impact

### 1. Réduire les inégalités d'accès à la pratique sportive

- Faciliter l'accès à l'information pour les populations éloignées du sport encadré (quartiers prioritaires, zones rurales...).
- Valoriser les équipements existants auprès des habitants.

### 2. Encourager la pratique libre et la santé par le sport

- Promouvoir l'activité physique régulière via une approche communautaire.
- Contribuer indirectement à la lutte contre la sédentarité.

### 3. Sensibiliser à la gestion participative des biens publics

- Renforcer la responsabilisation citoyenne autour des équipements publics.
  - Créer un pont entre usagers et collectivités via un outil numérique transparent.
- 

## Objectifs d'évolution à moyen/long terme

- **Déploiement à l'échelle nationale**, avec possibilité d'internationalisation à terme.
  - **Intégration de données dynamiques** : taux de fréquentation estimé, horaires d'ouverture, météo, etc.
  - **Fonctionnalités sociales optionnelles** : événements, défis sportifs, valorisation d'initiatives locales.
  - **Ouverture d'une API publique** pour favoriser la réutilisation des données par d'autres services ou acteurs du sport.
-

## En résumé

Les objectifs du projet SportMap sont multiples, complémentaires et alignés avec les valeurs de service public et d'innovation numérique. Ce projet vise autant à **offrir une solution technique fiable** qu'à **apporter une réponse sociale concrète** à une problématique réelle et actuelle : le manque de visibilité sur les infrastructures sportives de proximité.

## 5. Étapes de conception et de réalisation

### 5.1. Phase de réflexion et définition du périmètre

Avant même de commencer à coder, une phase importante d'idéation a été menée. Nous avons listé les problèmes concrets rencontrés par les pratiquants de sport libre et identifié les fonctionnalités à forte valeur ajoutée. En tant qu'usagers nous-mêmes, cette phase s'est faite de manière intuitive, mais avec

une volonté de structuration professionnel

**La Plateforme**

**Atelier 29/04**

**Fiche idée**

**Nom du projet**

Problématique identifiée : à quel problème particulier s'adresse votre idée ? Si besoin, revoir la cartographie réalisée

Description de votre idée : quels sont les services, les fonctionnalités de votre idée ?

Proposition de valeur : ce projet dans tel contexte va permettre de...

Veille : avez-vous vu d'autres projets similaires ?

Storyboard : illustrez votre idée via des dessins, croquis, images réelles ou générées par IA. Illustrez la manière dont il interagit avec l'utilisateur

Equipe : présentez-vous :)

## Conseils pour la sélection des idées

- A. Quelles fonctionnalités propose l'appli ?
- Identification terrain sur le secteur
  - Type de terrains (foot, basket...)
  - signalisation de dégradation sur le terrain
  - Nombre de joueurs sur terrain(pour le futur de l'app)

### Impact utilisateur:

- A. Considérer l'impact potentiel de l'idée sur les utilisateurs.
- Meilleur prévoyance de son entraînement
  - Meilleur qualité des infrastructures sportive
  - Fun
- B. Est-ce que cette idée résout un problème important ou améliore significativement l'expérience utilisateur?
- Améliore l'expérience joueur dans le sens où elle permet de mieux isoler son terrain sur lequel pratique ainsi que de plus pratiquer son sport en permettant de gagner du temps
  - Permet de connaître le nombre de joueurs (pour le futur de l'app)
- C. Quel serait le bénéfice de cette idée ?
- gain de temps
  - meilleure expérience utilisateur

### Originalité:

- A. - Évaluer à quel point l'idée est innovante?
- B. - Est-ce qu'elle apporte une nouvelle perspective ou une approche unique par rapport à ce qui existe déjà sur le marché?

## 2. Utiliser une matrice d'évaluation:

Créer une matrice d'évaluation avec les critères définis. Chaque idée sera notée pour chaque critère sur une échelle de 1 à 5 (ou une autre échelle appropriée). Par exemple:

Idée	Faisabilité Technique	Impact Utilisateur	Originalité	Alignement avec les Objectifs	Score total
Yassine					
Sofiane					

Cette phase a permis de poser les bases du **MVP** (Minimum Viable Product) : une première version simple, mais fonctionnelle, répondant à un besoin précis : *trouver facilement un terrain de sport public proche de chez soi*.

Les objectifs fixés à ce stade :

- Identifier la nature exacte des données à exploiter.
- Déterminer les fonctionnalités essentielles de l'interface (filtrage, carte, recherche).

- Évaluer les technologies à utiliser.
  - Répartir les tâches entre les deux membres de l'équipe.
- 

## 5.2. Analyse de l'existant et veille technique

Avant de concevoir l'application, nous avons réalisé une **veille concurrentielle** et une étude technique des solutions existantes :

- **Google Maps** et **OpenStreetMap** pour la cartographie.
- Des initiatives locales comme certaines cartes associatives ou municipales.
- Des portails Open Data (data.gouv.fr, collectivités territoriales, etc.).

Nous avons identifié une absence de solutions généralistes, intuitives et orientées sport en accès libre. La veille nous a aussi permis de choisir les outils techniques les plus adaptés à notre besoin.

 **Tableau comparatif des solutions existantes**

Solution	Forces	Faiblesses	Choix retenu ?	Justification du choix ou du rejet
<b>Google Maps</b>	- Interface intuitive- Données à jour- Couverture mondiale	- Peu d'infos sur les types d'équipements- Pas d'aspect collaboratif	 Non	Ne permet pas de différencier les équipements sportifs
<b>OpenStreetMap</b>	- Base de données ouverte- Données complètes pour certains usages	- Interface technique- Peu lisible pour le grand public	 Non	Trop complexe pour une appli destinée à tous les citoyens
<b>Géoportail (IGN)</b>	- Données très précises- Sources officielles fiables	- Interface peu mobile-friendly- Pas de signalement collaboratif	 Non	Pas adapté au grand public ni à une appli web intuitive

<b>Strava / Decathlon Outdoor</b>	- Données communautaires riches- UX bien pensée pour sportifs	- Ciblé sur le sport de performance- Peu d'intérêt pour les lieux fixes	✗ Non	Vise un public différent de celui de SportMap
<b>Solution maison avec Mapbox + OpenData</b>	- Totalement personnalisable- Intégration du collaboratif	- Nécessite plus de développement	✓ Oui	Permet de répondre précisément au besoin de cartographie participative

### 🔧 Comparatif de frameworks JavaScript (React vs Vue vs Angular)

Framework	Forces	Faiblesses	Choix retenu ?	Justification du choix ou du rejet
<b>React.js</b>	- Large communauté- Flexible et modulaire- Composants réutilisables- Vite à prendre en main	- Nécessite d'ajouter d'autres libs (routing, state, etc.)	✓ Oui	Convient parfaitement pour une app scalable comme SportMap
<b>Vue.js</b>	- Facile à apprendre- Syntaxe intuitive- Excellent pour des petits projets	- Moins adopté en entreprise- Moins d'outils "industriels"	✗ Non	Trop limité pour un projet évolutif à long terme
<b>Angular</b>	- Très complet- Intégration native d'outils (routing, tests, etc.)	- Courbe d'apprentissage raide- Structure imposée	✗ Non	Trop complexe pour un petit projet en autonomie

### 🗃 Comparatif des bases de données (PostgreSQL vs MySQL vs Firebase vs MongoDB)

Base de données	Forces	Faiblesses	Choix retenu ?	Justification du choix ou du rejet
PostgreSQL	- Base relationnelle <b>robuste</b> et open source - Supporte <b>JSON</b> et <b>GeoJSON</b> via PostGIS - Idéal pour les <b>données géolocalisées</b>	- Configuration initiale plus complexe - Courbe d'apprentissage pour PostGIS	<input checked="" type="checkbox"/> Oui	Idéal pour gérer les <b>équipements géolocalisés</b> , tout en conservant un modèle relationnel solide
MySQL	- Structurée- Maîtrisée par de nombreux devs- Bonne documentation	- Moins flexible- Moins adapté aux données non relationnelles	<input checked="" type="checkbox"/> Non	Structure trop rigide pour les besoins de SportMap
Firebase	- En temps réel- Facile à intégrer avec front- Auth intégrée	- Fermé (Google)- Moins personnalisable- Coûteux à long terme	<input checked="" type="checkbox"/> Non	Trop dépendant de Google, moins de contrôle
MongoDB	- Base <b>NoSQL</b> flexible - Stockage natif en <b>JSON</b> - Parfait pour des structures de données dynamiques - Hébergement cloud (Atlas)	- Non relationnelle - Nécessite une gestion manuelle de la <b>cohérence entre données</b>	<input checked="" type="checkbox"/> Non	Moins adapté ici car les relations entre entités sont importantes ; besoin de structure plus rigide

### 🧠 Comparatif des solutions backend envisagées

Base de données	Forces	Faiblesses	Choix retenu ?	Justification du choix ou du rejet
Django	- Framework complet (ORM, admin, sécurité, auth, routing...) - Rapide à développer - Excellent pour des APIs structurées	- Peut être trop <b>opinioné</b> - Moins flexible pour du microservice ou API minimalistique	<input checked="" type="checkbox"/> Oui	Permet de développer rapidement une API REST robuste, avec une gestion native des utilisateurs et du modèle

<b>Express.js</b>	<ul style="list-style-type: none"> <li>- Léger, minimaliste, <b>flexible</b></li> <li>- Large écosystème JavaScript (Node.js)</li> <li>- Facile à déployer</li> </ul>	<ul style="list-style-type: none"> <li>- Nécessite beaucoup de <b>boilerplate</b> pour une API complète</li> <li>- Moins sécurisé par défaut</li> </ul>	Non	Trop bas niveau pour le projet, nécessite beaucoup d'implémentation manuelle (auth, ORM, etc.)
<b>Laravel</b>	<ul style="list-style-type: none"> <li>- Framework <b>PHP très complet</b></li> <li>- Excellente gestion de la base de données et du routing</li> </ul>	<ul style="list-style-type: none"> <li>- Syntaxe plus lourde</li> <li>- PHP moins apprécié pour les API modernes</li> </ul>	Non	Environnement plus complexe à maintenir, moins adapté au projet basé sur JSON et PostgreSQL
<b>FastAPI</b>	<ul style="list-style-type: none"> <li>- <b>Très rapide</b> et moderne (Python)</li> <li>- Parfait pour des <b>APIs REST</b> ou microservices</li> <li>- Typage fort avec Pydantic</li> </ul>	<ul style="list-style-type: none"> <li>- <b>Moins de fonctionnalités intégrées (pas d'admin, moins d'outils intégrés que Django)</b></li> </ul>	Non	Très bon candidat, mais Django offre plus de productivité grâce à ses outils intégrés pour ce projet précis

Sur le plan des données, nous avons sélectionné plusieurs jeux de données Open Data, en priorisant :

- Les équipements sportifs géolocalisés (INSEP, Ministère des Sports).
  - Les données municipales accessibles (Marseille, Lyon, Paris, Bordeaux...).
- 

### 5.3. Choix de l'architecture technique

Dès cette étape, nous avons fait le choix d'une **architecture en React pour le front-end**, associée à une API REST (Django) et une base de données PostgreSql.

Ce choix s'explique par :

- La réactivité de React (cartes dynamiques, état local/global).
- La compatibilité entre l'écosystème Django et React.
- La facilité d'extension via des composants.

- L'utilisation de **Mapbox** pour l'intégration cartographique personnalisée.

**Structure envisagée :**

- **Front-end** : React + TailwindCSS
- **Back-end** : Django
- **Database** : PostgreSQL
- **Cartographie** : Mapbox GL JS
- **Déploiement** : Railway

[ Front React (Mapbox, Vite) ]



[ Django REST Framework ]



[ PostgreSQL + données GeoJSON ]

 Le tout est conteneurisé via Docker, versionné avec Git, et déployé automatiquement via GitHub Actions + Railway.

---

---

## 5.4. Organisation du travail et méthode de développement

Dès le lancement opérationnel du projet, nous avons adopté une **méthodologie agile simplifiée**, basée sur des sprints courts et des réunions régulières de suivi.

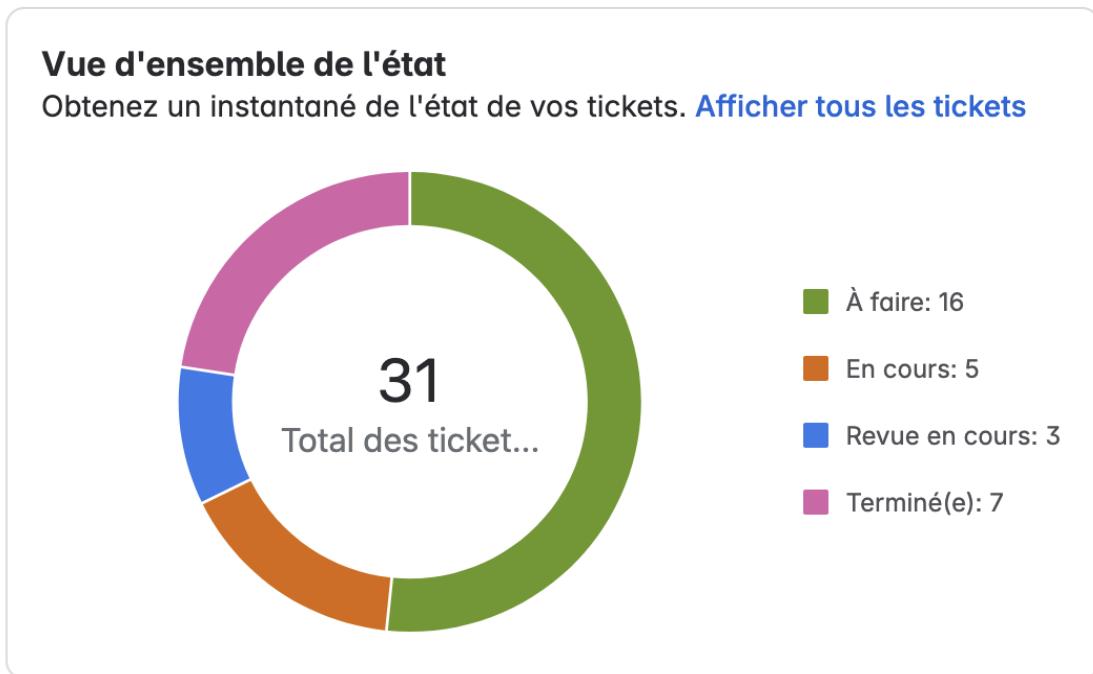
Chaque sprint durait entre **1 et 2 semaines**, avec des objectifs clairs :

- Définition des fonctionnalités à développer.
- Planification des tâches sur Trello (ou Jira).

- Suivi de l'avancement.
- Bilan et ajustements.

Nous avons découpé le projet en **trois grandes phases** :

1. Mise en place de l'architecture (React, API, base de données).
2. Développement des fonctionnalités de base (affichage carte, filtres, signalement).
3. Optimisation, mise en production et documentation.



	Type	Clé	Résumé	État	Commentaires	Personne	+
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SPORT-1	TICKET #001 - Configuration environnement Docker	TERMINÉ(E)	Ajoutez un commentaire		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SPORT-2	TICKET #002 - Structure projet Django	TERMINÉ(E)	Ajoutez un commentaire		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SPORT-3	TICKET #003 - Structure projet React	EN COURS	1 commentaire		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SPORT-29	TICKET #004.5 Conception MCD, MLD, MPD	TERMINÉ(E)	Ajoutez un commentaire		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SPORT-4	TICKET #004 - Base de données PostgreSQL	TERMINÉ(E)	Ajoutez un commentaire		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SPORT-5	TICKET #005 - Système authentification	TERMINÉ(E)	Ajoutez un commentaire		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SPORT-6	TICKET #006 - Configuration email	TERMINÉ(E)	Ajoutez un commentaire		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SPORT-7	TICKET #007 - JWT Authentication	TERMINÉ(E)	Ajoutez un commentaire		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SPORT-8	TICKET #008 - Frontend authentification	EN COURS	Ajoutez un commentaire		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SPORT-9	TICKET #009 - API installations sportives	REVUE EN COURS	Ajoutez un commentaire		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SPORT-10	TICKET #010 - Interface liste installations	REVUE EN COURS	Ajoutez un commentaire		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SPORT-11	TICKET #011 - Détail installation	À FAIRE	Ajoutez un commentaire		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SPORT-12	TICKET #012 - Géolocalisation	À FAIRE	Ajoutez un commentaire		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SPORT-13	TICKET #013 - Recherche et filtres	REVUE EN COURS	Ajoutez un commentaire		
<a href="#">+ Créer</a>							

## 5.5. Outils de versioning et collaboration

Nous avons utilisé **Git** comme système de gestion de version, avec **GitHub** pour l'hébergement du code source. Cela nous a permis de :

- Travailler simultanément sans conflit,
- Proposer des **pull requests** pour valider les contributions,
- Documenter l'évolution du projet grâce aux commits et issues.

Voici une organisation type de notre dépôt GitHub côté front :

```
Sport_Map_frontend/
├── public/      # Fichiers statiques et index.html
├── src/        # Code source principal React
│   ├── assets/    # Images, logos, styles globaux
│   └── components/ # Composants réutilisables
```

```

|   └── pages/      # Pages principales de l'application
|   └── services/    # Fonctions d'appel API, gestion auth, etc.
└── node_modules/  # Dépendances Node.js (auto-généré)

```

```

CDA_Backend/
├── config/        # Configuration Django
├── installations/ # App équipements sportifs
├── authentication/ # App authentification
├── signalements/ # App signalements
└── data/          # Données

```

**Branches**

Branch	Updated	Check status	Behind	Ahead	Pull request
main	2 days ago	Default			...

**Your branches**

Branch	Updated	Check status	Behind	Ahead	Pull request
feature/documentation-design	2 weeks ago		4	0	...
feature/report-system	2 weeks ago		5	1	...
feature/ui-improvements	2 weeks ago		4	0	...

**Branches**

Branch	Updated	Check status	Behind	Ahead	Pull request
main	11 hours ago	1 / 2	Default	...	

**Your branches**

Branch	Updated	Check status	Behind	Ahead	Pull request
signalement	now	1 / 2	0	0	...
auth	2 days ago	0 / 1	4	5	...

**Active branches**

Branch	Updated	Check status	Behind	Ahead	Pull request
signalement	now	1 / 2	0	0	...
auth	2 days ago	0 / 1	4	5	...



## 5.6. Développement de l'interface front-end

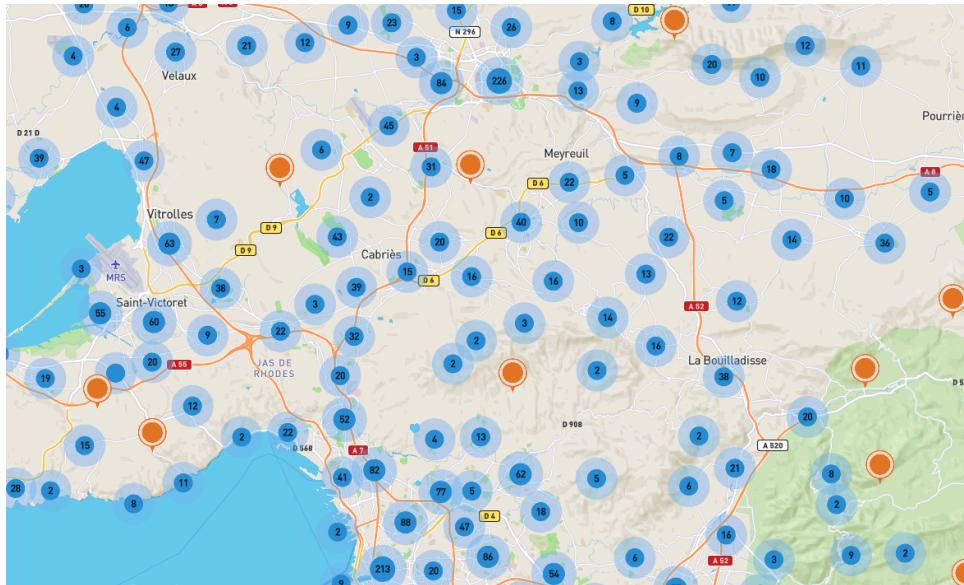
Le front-end de SportMap a été développé en **React**, avec une attention particulière portée à :

- la **simplicité d'utilisation**,
- la **rapidité d'affichage**,
- l'**adaptabilité mobile**.

Nous avons utilisé **TailwindCSS** pour accélérer la mise en page responsive et construire une interface cohérente et moderne.

### Fonctionnalités front-end principales :

- Affichage dynamique de la carte via **Mapbox GL JS**.
- Intégration de la **géolocalisation** du navigateur.
- Système de **filtres** (par type de terrain, sport pratiqué).
- Affichage de **fiches d'équipements** au clic sur un point de la carte.



Marseille sport



Marseille sport

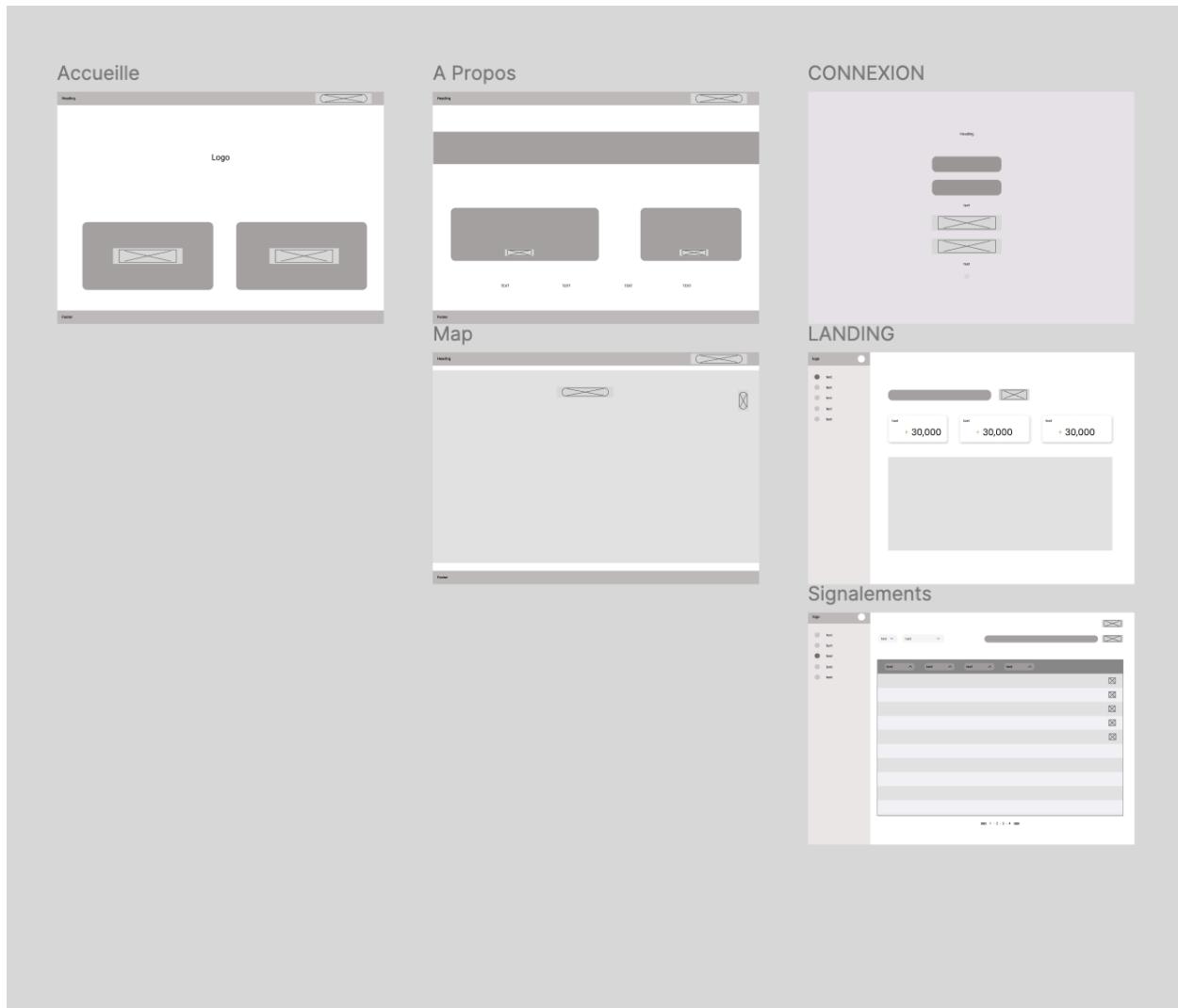


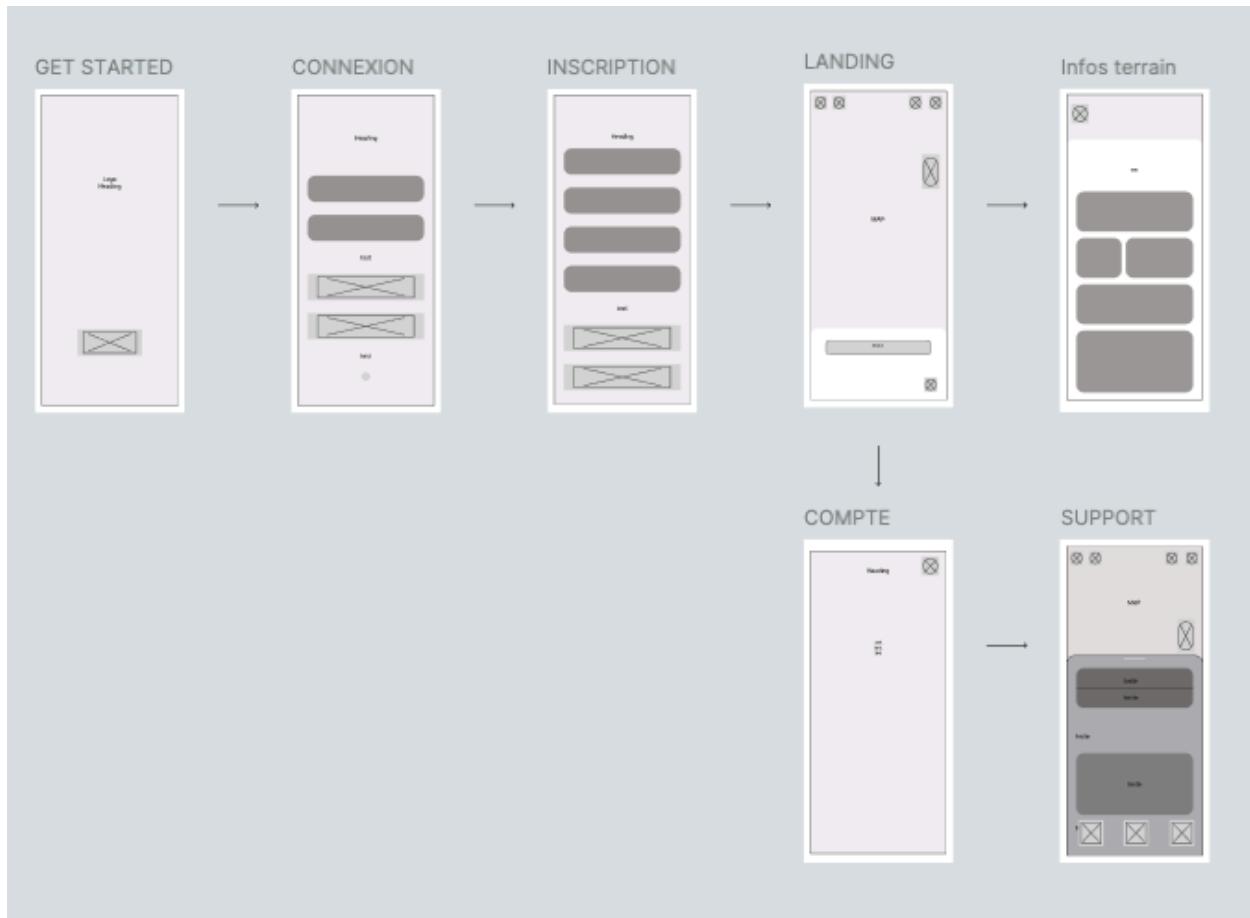
Accueil

A propos

Sports

Carte





## 5.7. Développement du back-end et des API

Le back-end de SportMap a été conçu avec Django, dans une logique d'API REST structurée et sécurisée.

L'objectif était de fournir un service efficace, clair et maintenable, pour interagir avec la base de données PostgreSQL

### Fonctionnalités back-end principales :

- **Connexion sécurisée à la base PostgreSQL.**
- **Récupération d'équipements sportifs** via des routes filtrées.
- **Système de validation des contributions** (pour éviter les doublons ou données erronées).
- **Gestion d'utilisateurs simples** avec jeton (authentification basique avec JWT pour la partie contribution).

## ## 🌎 \*\*Routes Installations Sportives\*\* (Public)

...

📍 BASE: /api/v1/installations/

```
  └── GET /equipments/      # Liste équipements avec filtres
      ├── Query: ?bounds=sw_lng,sw_lat,ne_lng,ne_lat
      ├── Query: ?types=Terrain de football,Court de tennis
      └── Response: JSON Array[Installation]

  └── GET /geojson/        # Format GeoJSON pour cartes
      └── Response: GeoJSON FeatureCollection

  └── GET /sports/         # Catalogue des sports
      ├── Query: ?detailed=true
      └── Response: {sports: [...], count: N}

  └── GET /installations/   # Liste paginée
      ├── Query: ?limit=20
      └── Response: {installations: [...], count: N}
...
```

## ## 🔒 \*\*Routes Authentification\*\* (Public)

...

📍 BASE: /auth/

```
  └── POST /register/      # Incription avec code email
      ├── Body: {email, password}
      └── Response: {message: "Code envoyé"}

  └── POST /verify/        # Validation code inscription
      ├── Body: {email, code}
      └── Response: {message: "Compte validé"}

  └── POST /login/         # Connexion utilisateur
      ├── Body: {email, password}
      └── Response: {token: "jwt...", message: "Connexion réussie"}

  └── POST /request-password-reset/ # Demande reset password
      ├── Body: {email}
      └── Response: {message: "Lien envoyé"}

  └── POST /reset-password/    # Reset avec token
      ├── Body: {email, token, new_password}
      └── Response: {message: "Mot de passe réinitialisé"}
...
```

## ## 📝 \*\*Routes Signalements\*\* (Authentifié - JWT requis)

```

📍 BASE: /api/v1/signalements/  
🔒 Header: Authorization: Bearer <jwt\_token>

```
  └─ POST /create/          # Créer un signalement
      └─ Body: {
          installation_id: 123,
          message: "Description du problème",
          images_url: "https://...",
          type: "Dégradation"
        }
      └─ Response: {message: "Signalement créé", id: 456}

  └─ GET /                # Mes signalements (utilisateur)
      └─ Response: [{id, message, type, etat, date}, ...]

  └─ GET /all/             # Tous signalements (admin)
      └─ Query: ?status=Nouveau,En cours,Résolu
      └─ Response: [...signalement, utilisateur, installation], ...]

  └─ PUT /{id}/            # Modifier un signalement (admin)
      └─ Body: {etat: "En cours", admin_comment: "..."}
      └─ Response: {message: "Signalement mis à jour"}

  └─ DELETE /{id}/         # Supprimer signalement (admin)
      └─ Response: {message: "Signalement supprimé"}
```

...

## ## 📊 \*\*Codes de Réponse HTTP\*\*

| Code    | Signification                                        | Utilisation |
|---------|------------------------------------------------------|-------------|
| **200** | ✓ Succès   GET, PUT réussi                           |             |
| **201** | ✓ Crée   POST signalement créé                       |             |
| **400** | ✗ Requête invalide   Paramètres manquants/incorrects |             |
| **401** | ✗ Non authentifié   JWT manquant/invalidé            |             |
| **403** | ✗ Interdit   Pas les permissions (admin requis)      |             |
| **404** | ✗ Non trouvé   Ressource inexistante                 |             |
| **409** | ✗ Conflit   Email déjà utilisé                       |             |
| **500** | ✗ Erreur serveur   Erreur interne                    |             |

## ## 🔒 \*\*Authentification JWT\*\*

```bash

```
# 1. Obtenir le token
POST /auth/login/
Body: {"email": "user@example.com", "password": "password"}
```

Response: {"token": "eyJ0eXAiOiJKV1QiLCJ..."}

## # 2. Utiliser le token

GET /api/v1/signalements/

Headers: Authorization: Bearer eyJ0eXAiOiJKV1QiLCJ...

...

Extrait de code d'une route Django (/api/equipements),

```
urlpatterns = [
    path('equipments/', views.get_equipments, name='get_equipments'),
    path('geojson/', views.get_geojson, name='get_geojson'),
    path('sports/', views.get_sports, name='get_sports'),
    path('installations/', views.installations_list, name='installations_list'),
]
```

Extrait de test api (/api/sports), (/api/geojson),

The screenshot shows the Postman application interface with two separate requests:

- Request 1: GET https://cdabackend-production-3c8a.up.railway.app/geojson**
  - Method: GET
  - URL: https://cdabackend-production-3c8a.up.railway.app/geojson
  - Body: JSON (Preview shows a GeoJSON feature collection with a single point geometry)
  - Headers: (11)
  - Test Results: 200 OK (451 ms, 2.55 MB)
- Request 2: GET https://cdabackend-production-3c8a.up.railway.app/sports**
  - Method: GET
  - URL: https://cdabackend-production-3c8a.up.railway.app/sports
  - Body: JSON (Preview shows a list of sports activities)
  - Headers: (11)
  - Test Results: 200 OK (404 ms, 5.3 KB)

---

## 5.8. Système de signalement citoyen

L'un des piliers de SportMap est sa dimension **participative**, permettant aux citoyens de **signaler directement les problèmes constatés** sur les équipements sportifs de leur commune.

Chaque utilisateur peut soumettre un **signalement** concernant :

-  Un équipement cassé, mal entretenu ou dangereux
  -  Un souci de localisation ou d'accessibilité
  -  Un commentaire ou une remarque utile pour la collectivité
- 

Le système de signalement repose sur une **chaîne front–API–base solide** :

### Frontend (React) :

- Formulaire de signalement avec :
  -  Auto-remplissage de l'équipement concerné (via recherche ou clic sur la carte)
  -  Localisation GPS automatique du signalement
  -  Validation des champs avant envoi : message obligatoire, coordonnées valides

### Backend (Django) :

- Création sécurisée du signalement (authentification requise via JWT)
  - Vérification des données reçues
  - Stockage dans PostgreSQL avec gestion des statuts ( en attente,  validé,  refusé)
  - Notifications possibles à l'équipe de modération
- 



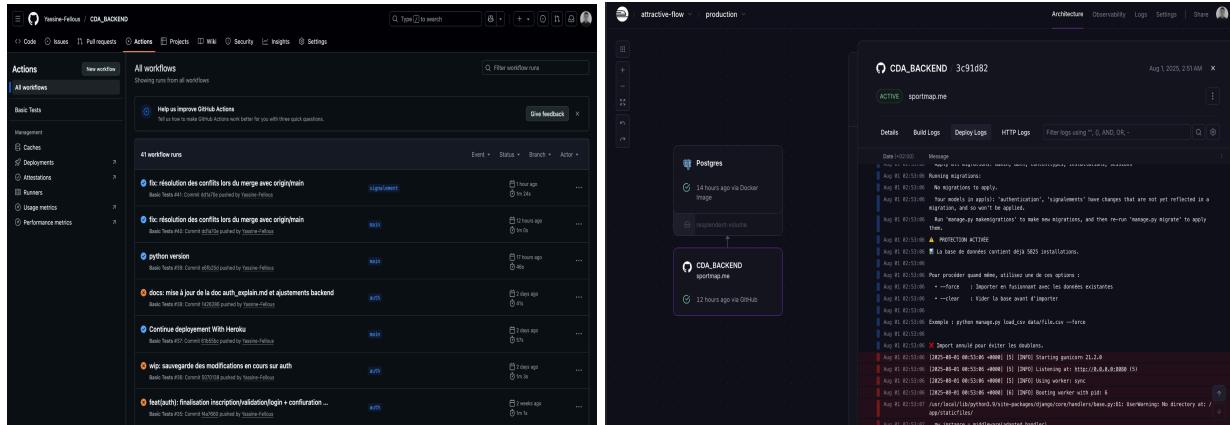
## 5.9. Déploiement et mise en ligne

Le projet a été déployé sur les plateformes suivantes :

- **Front-end (React)** : hébergé sur **Railway**, pour un déploiement rapide, continu et sans configuration complexe.
- **Back-end (API Django)** : déployé via **Railway**, avec mise en place des variables d'environnement pour sécuriser les accès à la base de données.
- **Base de données (PostgreSQL)** : hébergée sur railway pour une disponibilité 24/7, et sécurisée par clé privée.

## Déploiement CI/CD :

Grâce aux hooks GitHub, chaque **push sur la branche main** déclenche automatiquement un déploiement à condition que notre Intégration continue est validé :



```

    class Installation(models.Model):
        # TOUT EN TextField = ILLIMITÉ !
        inst_numero = models.TextField(blank=True, null=True)           # == ILLIMITÉ
        coordonnees = models.JSONField()
        inst_nom = models.TextField(blank=True, null=True)           # == ILLIMITÉ
        equip_type_name = models.TextField(blank=True, null=True)       # == ILLIMITÉ
        equip_type_famille = models.TextField(blank=True, null=True)     # == ILLIMITÉ
        equip_aps_nom = models.TextField(blank=True, null=True)         # == ILLIMITÉ
        equip_acc_libre = models.BooleanField(default=False)
        equip_url = models.TextField(blank=True, null=True)           # == ILLIMITÉ
        inst_adresse = models.TextField(blank=True, null=True)         # == ILLIMITÉ
        inst_cp = models.TextField(blank=True, null=True)           # == ILLIMITÉ
        equip_prop_nom = models.TextField(blank=True, null=True)       # == ILLIMITÉ
        equip_gest_type = models.TextField(blank=True, null=True)       # == ILLIMITÉ
        inst_acc_handi_bool = models.BooleanField(default=False)

    class Meta:
        db_table = 'installations'
        verbose_name = 'Installation Sportive'
        verbose_name_plural = 'Installations Sportives'

    def __str__(self):
        return f'{self.inst_nom} - {self.equip_type_name}'

```



## 5.10. Documentation technique

Dès le départ, nous avons documenté l'ensemble des fonctionnalités et de l'architecture du projet.

Cela comprend :

- Un **README** complet (présentation, installation, déploiement, stack),
- Des **commentaires dans le code** (front et back),
- Un **schéma d'architecture technique**, pour les futurs contributeurs.

Cette documentation assure la **pérennité du projet** et facilite sa reprise ou extension.

👉 **À insérer ici :**

- Capture du README affiché sur GitHub,
- Capture de commentaires dans un fichier Map.jsx ou server.js,
- Schéma d'architecture commentée.



## 5.11. Résultats obtenus

À ce stade du projet, la **version 1 (V1)** de SportMap est pleinement fonctionnelle.

Elle remplit l'objectif initial fixé au début de la formation : permettre à tout utilisateur de **trouver rapidement les équipements sportifs en accès libre autour de lui**, grâce à une **interface claire, rapide et accessible**.

### Fonctionnalités livrées :

- Carte interactive connectée à une base de données (5800 équipements référencés).
- Filtres dynamiques par type d'équipement.
- Géolocalisation automatique.
- Formulaire de contribution.
- Interface responsive Web.
- Mise en ligne front + back + base de données.

Ces livrables constituent un socle solide pour des **versions futures** avec davantage de services (statistiques d'affluence, interactions sociales, outils pour les collectivités...).

The screenshot displays the "Marseille sport" website interface. At the top, there's a navigation bar with links for Accueil, A propos, Sports, Authentification, Carte, and a search icon. Below the navigation is a main header with the text "Marseille Sport" and a sub-instruction "Trouvez votre terrain idéal en quelques clics !". A blue button labeled "Carte des équipements" is visible. To the right, there's a large image of a hand holding a smartphone displaying a map application. Below this, two smaller images are shown: one of a map titled "Les sports" with several blue dots indicating locations, and another of a basketball court with the text "Le projet" overlaid.

**Découvrez les Sports à Marseille**

Search bar: Rechercher un sport...

|  |  |  |
|--|--|--|
| <b>Activités de forme et de santé</b><br>Voir sur la carte → | <b>Aikido / Aikibudo / Budo</b><br>Voir sur la carte →                                 | <b>Américain)</b><br>Voir sur la carte →   |
| <b>Anglais</b><br>Voir sur la carte →                        | <b>Aquagym</b><br>Voir sur la carte →  | <b>Arbalète / Armes anciennes / Bench rest / Silhouettes métalliques / Tir sportif de vitesse</b><br>Voir sur la carte → |
| <b>Attelage</b><br>Voir sur la carte →                       | <b>Autres Danses</b><br>Voir sur la carte →  | <b>Aviron</b><br>Voir sur la carte →   |
| <b>Aéromodélisme</b><br>Voir sur la carte →                  | <b>Badminton</b><br>Voir sur la carte →  | <b>Baignade loisirs</b><br>Voir sur la carte →   |
| <b>Balle au tambourin</b><br>Voir sur la carte →             | <b>Ballet sur glace / Danse sur glace / Patinage artistique</b><br>Voir sur la carte → | <b>Ballon au poing</b><br>Voir sur la carte →  |

Marseille sports

localhost:5173/map

Applications film permis music tripp instagreen CIEG/PL idée cam projet papa Ukulele Tabs formation/cour Nouvelle version de Chrome disponible

Accueil A propos Sports Authentification Carte

Search a sport...

Marseille sport

localhost:5173/auth

Connexion

Bienvenue !  
Pas encore de compte ?

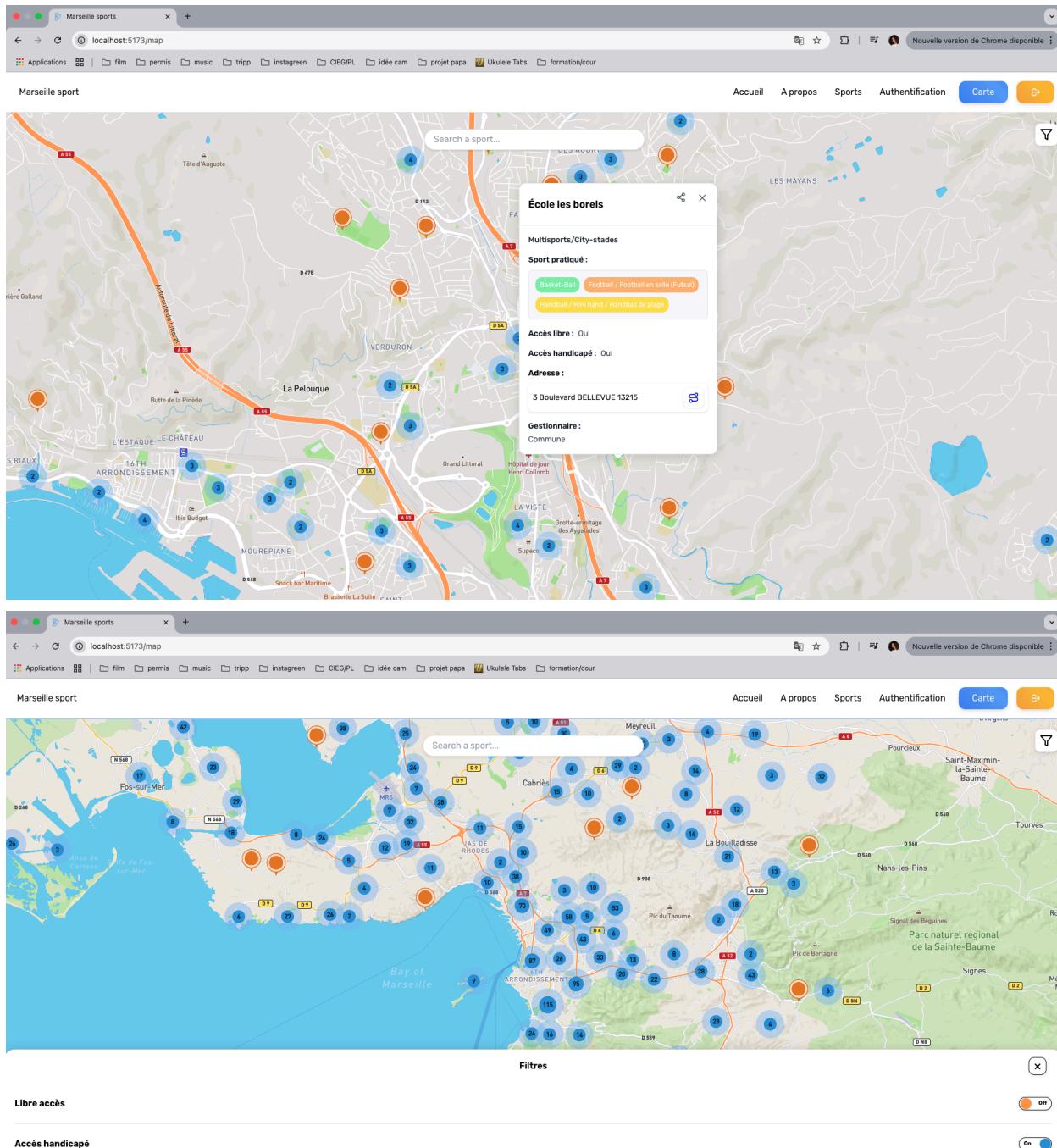
Inscription

Nom d'utilisateur ou Email

Mot de passe

Mot de passe oublié ?

Connexion



## 🧠 5.12. Compétences mobilisées et apprentissages techniques

Durant les différentes phases du projet, j'ai pu mobiliser l'ensemble des compétences acquises en formation CDA, et en développer de nouvelles, notamment :

### **Compétences techniques :**

- Conception d'architecture (choix technologiques, structuration).
- Programmation front-end avec **React** et gestion d'état.
- Intégration cartographique avec **Mapbox GL JS**.
- Mise en place d'API REST sécurisée avec Django.
- Connexion à une base de données **PostgreSql**.
- Hébergement Railway, gestion de domaine et de variables d'environnement.

### **Compétences transversales :**

- Gestion de projet en méthode agile.
- Collaboration efficace à distance avec Git/GitHub.
- Documentation technique.
- Répartition des rôles et gestion du temps.



## **5.13. Difficultés rencontrées**

Le projet n'a pas été exempt d'obstacles, parmi lesquels :

- L'importation de données open data avec des formats hétérogènes (CSV, GeoJSON, API non standardisées).
- Les limites de la géolocalisation sur mobile (précision, autorisations).
- Les conflits de dépendances lors des builds sur Railway.
- L'équilibrage entre ergonomie visuelle et performances sur carte.

Chaque difficulté a été surmontée progressivement, grâce à la veille technique, aux tests utilisateurs, et à l'entraide entre développeurs.

---



## 5.14. Transition vers la suite

Cette phase de réalisation m'a permis de consolider ma posture de **concepteur développeur d'applications**, capable de piloter un projet de bout en bout.

La suite du dossier présentera plus en détail **les compétences techniques et transversales validées** dans le cadre du référentiel RNCP, à travers le projet SportMap.

## 6. Compétences mises en œuvre dans le cadre du projet

Le projet **SportMap** m'a permis de mettre en pratique, en autonomie et en situation réelle, l'ensemble des **compétences professionnelles** visées par le titre RNCP "**Concepteur Développeur d'Applications**" délivré par **LaPlateforme**.

Ce projet constitue un cas concret de développement logiciel complet, de la phase d'analyse à la mise en production. Ci-dessous, un tableau croisé entre les **blocs de compétences RNCP** et les **réalisations associées** dans le cadre de SportMap :

---



### Bloc 1 : Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité

| Compétence RNCP  | Mise en œuvre dans SportMap                                  |
|--|--|
| Maquetter une application                                    | Maquettage de l'interface Web et mobile avec Figma et Trello |
| Réaliser une interface utilisateur Web statique et dynamique | Développement complet en React avec composants fonctionnels  |

Développer une interface utilisateur avec une solution mobile

Application responsive, pensée en mobile-first  
(tailwindCSS, PWA)

Mettre en œuvre des éléments de sécurité dans l'interface

Validation des champs, protection contre les injections côté client

MapPopup.jsx

```
import React from 'react';
import { Popup } from 'react-map-gl';
import { X, Share2, AlertTriangle } from 'lucide-react';
import { useNavigate } from 'react-router-dom';
import { backgroundStyleSport } from '@utils/backgroundStyleSport';
import PopupHeader from './PopupHeader';
import AddressSection from './AddressSection';
import './MapPopup.css';

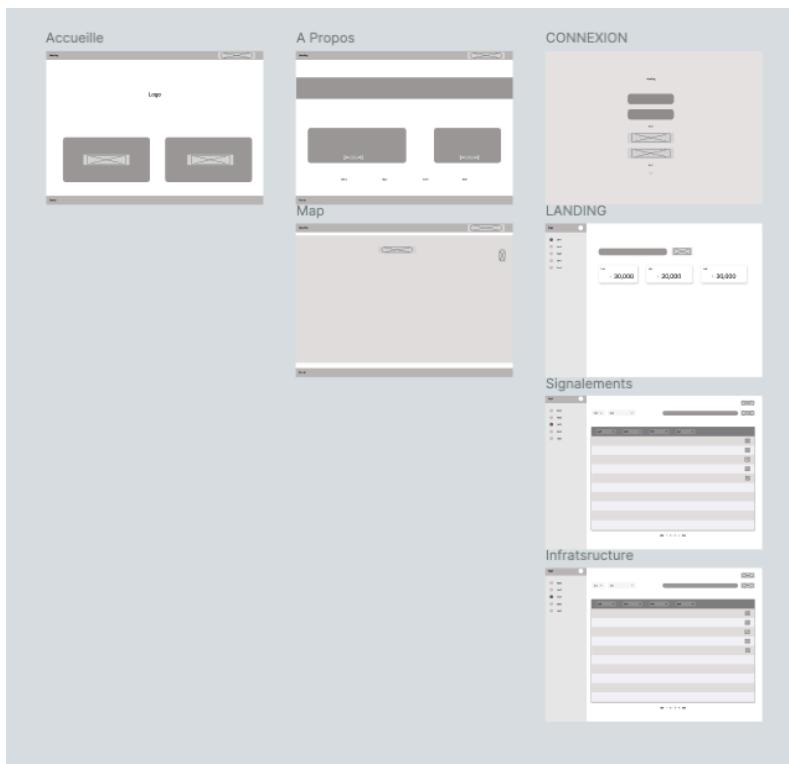
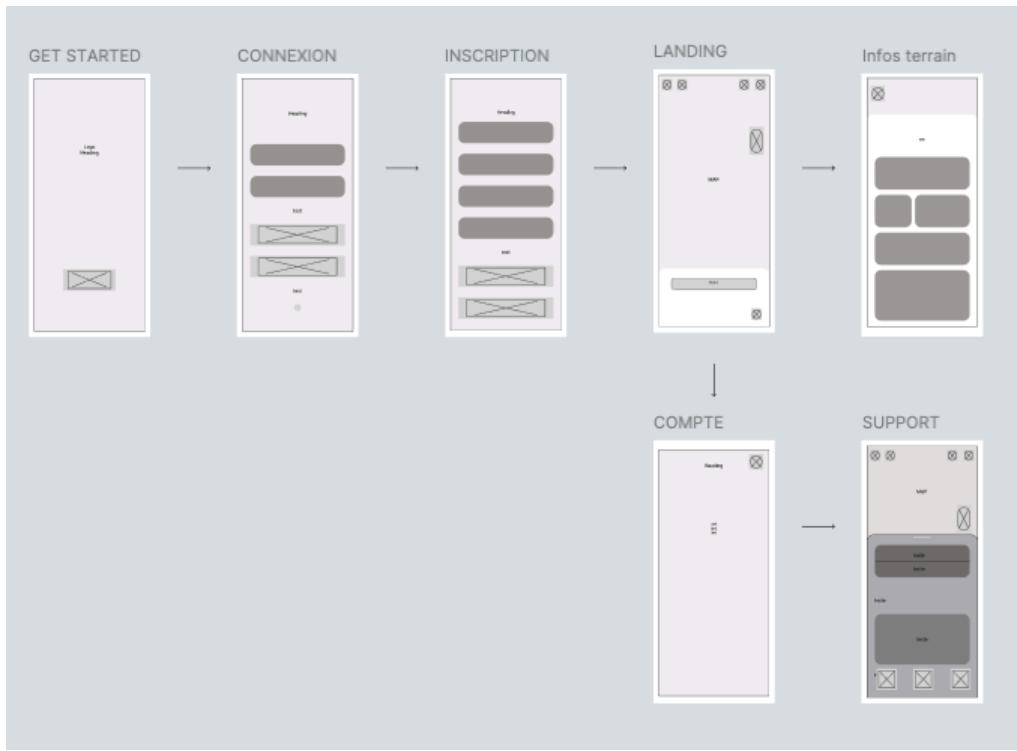
const MapPopup = ({ popupInfo, onClose }) => {
  const navigate = useNavigate();

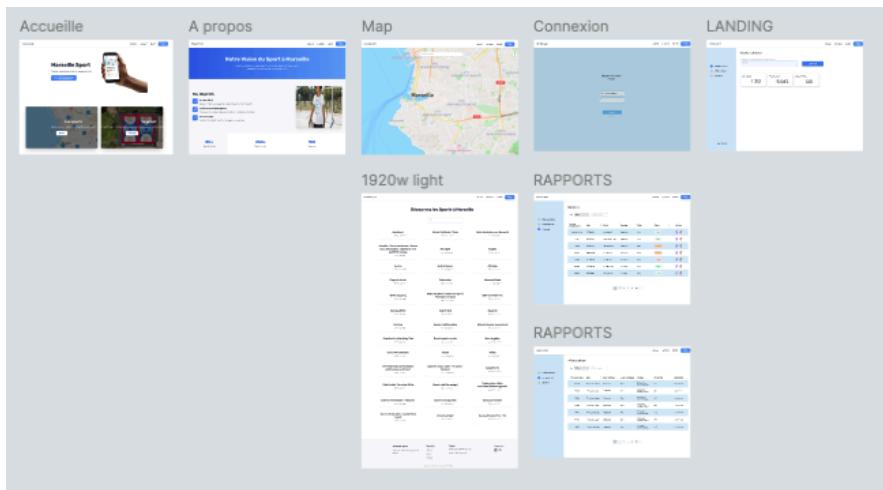
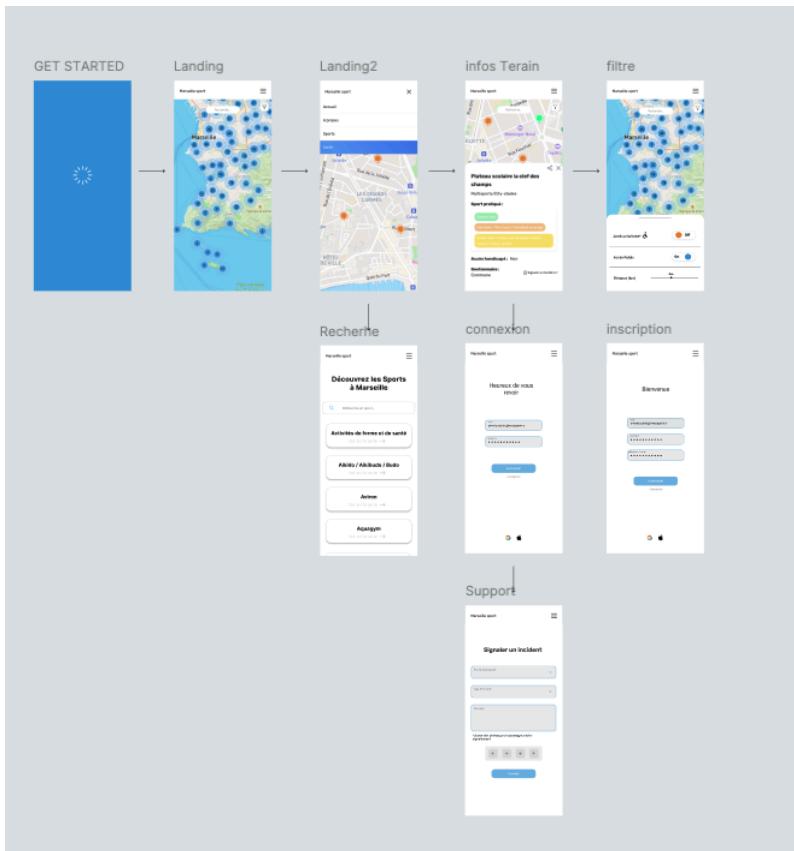
  if (!popupInfo) return null;

  const equipmentData = {
    nom: popupInfo.properties.name,
    adresse: popupInfo.properties.address,
    ville: popupInfo.properties.city,
    coordonnees: {
      latitude: popupInfo.latitude,
      longitude: popupInfo.longitude
    },
    type: popupInfo.properties.type,
    famille: popupInfo.properties.family
  };

  const handleReportClick = () => {
    navigate('/report', {
      state: { equipmentData }
    });
  };
}

VITE v6.0.7 ready in 393 ms
+ Local: http://localhost:5172/
+ Network: use --host to expose
+ press h + enter to show help
BrowsersList: browsers data (canuse-lite) is 8 months old. Please run:
npx update-browserslist-db@latest
May you should do it manually: https://github.com/browserslist/update-db#readme
2:58:11 AM [vite] env changed: restarting server...
2:58:11 AM [vite] server restarted.
```





## Bloc 2 : Concevoir et développer la persistance des données en intégrant les recommandations de sécurité

Compétence RNCP

Mise en œuvre dans SportMap

|   |   |
|---|---|
| Concevoir une base de données                                     | Modélisation relationnelle : tables utilisateurs, équipements, signalements, etc.                     |
| Mettre en place une base de données relationnelle ou NoSQL        | Mise en œuvre de <b>PostgreSQL</b> (relationnelle) avec extension <b>PostGIS</b> pour les données géo |
| Développer des composants d'accès aux données                     | Développement d'API REST en <b>Django</b> utilisant l'ORM pour interagir avec PostgreSQL              |
| Mettre en œuvre des éléments de sécurité dans l'accès aux données | Utilisation de <b>variables d'environnement</b> (clé secrète, DB) et <b>permissions Django</b>        |

```

from django.db import models
from django.utils import timezone

# Modèle pour l'authentification des utilisateurs
class UserAuth(models.Model):
    email = models.EmailField(unique=True)
    password = models.CharField(max_length=128)
    is_verified = models.BooleanField(default=False)
    verification_code = models.CharField(max_length=6, blank=True, null=True)
    reset_token = models.CharField(max_length=64, blank=True, null=True)
    reset_token_created = models.DateTimeField(blank=True, null=True)

    class Meta:
        db_table = 'authentication_userauth'
        verbose_name = 'Utilisateur'
        verbose_name_plural = 'Utilisateurs'

    def __str__(self):
        return self.email

```



## Bloc 3 : Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité

| <b>Compétence RNCP</b>                                   | <b>Mise en œuvre dans SportMap</b>   |
|--|--|
| Concevoir une application organisée en couches           | Architecture MVT séparant Front (React), API (Django), BDD (PostgreSql)  |
| Développer des composants métier                         | Gestion des signalements utilisateurs, tri des filtres, affichage carte  |
| Développer des composants de présentation                | Affichage dynamique des équipements avec React + Mapbox  |
| Développer des composants d'accès aux données            | Requêtes optimisées en PostgreSQL via Django ORM, avec filtrage, jointures et support des données spatiales (GeoJSON). |
| Mettre en œuvre des éléments de sécurité dans le système | Authentification basique pour les contributions (JWT)  |

👉 **À insérer ici :** Capture d'un fichier de route API ou schéma d'architecture logique.

#### **Bloc 4 : Préparer et exécuter les plans de tests d'une application**

| <b>Compétence RNCP</b>                                  | <b>Mise en œuvre dans SportMap</b>                                 |
|---|--|
| Élaborer des plans de test                              | Élaboration de jeux de test unitaires pour les filtres et la carte |
| Élaborer et exécuter des scénarios de test fonctionnels | Tests manuels sur les contributions, filtres, géolocalisation      |
| Automatiser les tests (optionnel)                       | Non automatisés mais tests reproductibles manuellement             |

Rédiger la documentation technique et utilisateur

Rédaction d'un README complet, documentation API et schéma projet



À insérer ici : Capture de tests dans Postman ou console, extrait de README.



## Bloc 5 : Préparer et exécuter le déploiement d'une application

### Compétence RNCP

Préparer le déploiement

### Mise en œuvre dans SportMap

Configuration des environnements Railway

Mettre en œuvre le déploiement

Mise en ligne CI/CD via GitHub – push déclenche build automatique

Réaliser les évolutions et la maintenance de l'application

Mise à jour des filtres, correctifs après feedback utilisateur

Assurer la traçabilité et la maintenance du code

Suivi Git, branches nommées, commits explicites, pull requests



## Bilan global des compétences mobilisées

Au terme du projet, je suis en mesure de :

- Concevoir une application Web complète, modulaire et documentée,
- Développer des interfaces utilisateurs modernes et responsives,

- Créer et interroger des bases de données sécurisées,
  - Gérer un projet en autonomie ou en binôme (agile),
  - Déployer une application en production,
  - Rédiger une documentation technique claire.
- 

## Transition – Synthèse des compétences mobilisées dans SportMap

Afin de préparer l'analyse de l'impact global du projet, cette page fait la synthèse des **compétences techniques et transversales** mises en œuvre durant l'ensemble du développement de l'application SportMap.

### Tableau récapitulatif des blocs RNCP activés

| Bloc de compétences RNCP                 | Compétences mises en œuvre dans SportMap  |
|--|---|
| <b>Bloc 1 – UI et sécurité interface</b> | Maquettage, responsive design, React, Tailwind, sécurisation des formulaires  |
| <b>Bloc 2 – Persistance des données</b>  | Conception de la base en PostgreSQL, développement d'une API REST sécurisée avec Django, gestion des accès via JWT, et déploiement sur le cloud (Railway).                                  |
| <b>Bloc 3 – Architecture multicouche</b> | Structuration en couches frontend (React) ↔ API REST (Django) ↔ base de données PostgreSQL, avec une architecture modulaire, routage React côté client, et services personnalisés côté API. |
| <b>Bloc 4 – Tests et documentation</b>   | Tests fonctionnels, plan de test manuel, documentation technique (README, API, schémas)   |

|  |  |
|--|--|
| <b>Bloc 5 – Déploiement et maintenance</b> | CI/CD avec Railway, GitHub actions, suivi de version, hébergement sécurisé |
|--|--|

## Compétences transversales renforcées

| Compétence transversale                    | Illustration dans SportMap  |
|--|---|
| <b>Méthodologie agile / gestion projet</b> | Organisation en sprints, Trello/Jira, communication efficace avec le binôme   |
| <b>Communication professionnelle</b>       | Présentation orale du projet, documentation claire, échanges avec utilisateurs test   |
| <b>Veille technologique</b>                | Comparaison de frameworks, choix d'une stack technique adaptée, intégration de Mapbox avec données spatiales PostgreSQL (GeoJSON), et hébergement avec Railway. |
| <b>Travail collaboratif à distance</b>     | Git, branches dédiées, pull requests, gestion de conflits   |

## Conclusion intermédiaire

Le projet SportMap a été un véritable **levier de montée en compétences**, en me plaçant dans une situation proche de celle rencontrée en entreprise ou en start-up : porteur de projet, développeur full-stack, gestionnaire de données et coordinateur technique.

## 7. Outils, technologies et architecture technique

Le développement du projet SportMap repose sur une architecture professionnelle, modulaire et évolutive, utilisant une stack technologique moderne parfaitement adaptée aux projets web full-stack. Le découplage complet entre le front-end et le back-end assure une meilleure scalabilité, testabilité et organisation du code.

---

## 7.1. Environnement de développement

Le projet est développé sur un environnement **Dockerisé**, avec deux dépôts GitHub distincts pour garantir une séparation claire entre les responsabilités frontend et backend.

| Util                           | Utilisation   |
|--------------------------------|---|
| <b>Visual Studio Code</b>      | Éditeur principal de développement                              |
| <b>Git + GitHub</b>            | Suivi de version, branches séparées, gestion collaborative      |
| <b>Docker / Docker Compose</b> | Conteneurisation du front, du back et de la base de données     |
| <b>Postman</b>                 | Tests des routes de l'API REST Django                           |
| <b>pgAdmin / psql CLI</b>      | Administration de la base PostgreSQL                            |
| <b>Notion / Trello</b>         | Organisation du projet, suivi des tâches et planification agile |



À insérer ici : captures de Docker Desktop, pgAdmin, structure des deux dépôts GitHub.

---

## 7.2. Technologies utilisées

### Front-end (React + Vite + TailwindCSS + Mapbox GL JS)

- **React.js** : composantisation poussée de l'interface utilisateur.
- **Vite** : serveur de développement rapide avec hot reload, gestion moderne des modules.
- **TailwindCSS** : framework CSS utilitaire, permettant un design responsive propre et cohérent.

- **Mapbox GL JS** : affichage des cartes interactives, ajout de couches dynamiques, gestion des popups.
- **fetch()** : utilisé pour toutes les requêtes API vers le back-end Django (GET / POST / PUT), sans dépendance tierce.

## Back-end (Django + Django REST Framework + ORM)

- **Django** : framework Python structuré en **MVT (Model – View – Template)**.
- **Django REST Framework** : mise en place d'une API REST robuste, documentation automatique avec Swagger/Redoc.
- **Django ORM** : manipulation sécurisée et typée des données PostgreSQL.
- **PostgreSQL** : base relationnelle optimisée pour la recherche d'équipements, avec extensions futures possibles (PostGIS).

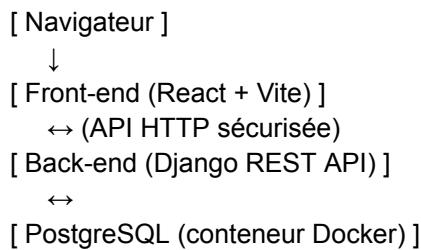
 À insérer ici : extrait de modèle Django, route DRF, ou structure de services (/auth/, /reporting/, etc.)

---

## 7.3. Architecture technique

Le projet adopte une **architecture MVT** côté back-end et **SPA** côté front-end.

### Vue d'ensemble de l'architecture



- **Architecture MVT Django** : séparation claire entre modèles, vues logiques (views.py), et les endpoints exposés via DRF.
- **Front-end React** : communique via appels RESTful, isolé dans un dépôt indépendant.
- **Back-end Django** : organisé en **modules fonctionnels** (authentification, signalements, équipements), chacun dans son propre dossier avec modèles, vues et routes associées.
- **Docker Compose** permet de lancer toute l'architecture localement en un seul docker-compose up.



## 7.4. Sécurité et gestion des accès

- Utilisation de **fichiers .env** pour stocker les variables sensibles (clés API, identifiants BDD).
  - Contrôle des permissions dans Django REST Framework (authentification, droits d'accès aux endpoints).
  - Vérification des données côté back avec des **serializers Django** (validation typée et personnalisée).
  - Protection de la base PostgreSQL via un mot de passe fort, conteneur isolé, et connexions restreintes.
- 



## 7.5. Déploiement et hébergement

- **Docker Compose** permet un déploiement local ou distant en un clic.
  - Le front-end peut être hébergé sur **Railway** ou **Netlify** (build Vite), connecté à une API publique sécurisée.
  - Le back-end peut être déployé sur **Railway**, ou un serveur VPS, avec variables d'environnement et base PostgreSQL cloud (ex : Supabase ou ElephantSQL).
- 

# 8. Problèmes rencontrés et solutions apportées



## Problème d'affichage des équipements sportifs sur la carte

L'un des premiers problèmes rencontrés lors du développement de la carte interactive a été lié à **l'affichage de plusieurs milliers de points** représentant les équipements sportifs sur l'interface Mapbox. En effet, charger directement l'ensemble des données dans une couche GeoJSON causait :

- un **ralentissement important de l'affichage**,
- une **surcharge visuelle** sur les zones densément équipées,

- et parfois même un blocage partiel du rendu, en particulier sur mobile.



## Solution mise en place : clustering des points

Pour résoudre ce problème, j'ai implémenté le **clustering des points via Mapbox GL JS**, en combinant la configuration suivante :

- Activation de l'option cluster: true dans la balise <Source> ;
- Définition d'un rayon de cluster (clusterRadius: 35) et d'un zoom maximum de regroupement (clusterMaxZoom: 14) ;
- Ajout de trois couches spécifiques :
  - clusterLayer : affichage des bulles de regroupement (points multiples),
  - clusterCountLayer : indication du nombre de points dans chaque cluster,
  - unclusteredPointLayer : affichage des points individuels non regroupés.

Un exemple de structure :

```
<Source
  id="equipments"
  type="geojson"
  data={filteredEquipments || getFilteredFeatures()}
  cluster={true}
  clusterMaxZoom={14}
  clusterRadius={35}
>
  <Layer {...clusterLayer} />
  <Layer {...clusterCountLayer} />
  <Layer {...getUnclusteredPointLayer()} />
</Source>
```



## Résultats obtenus

- Le rendu de la carte est devenu **fluide et lisible**, même avec plusieurs milliers de points.
- L'interface est désormais **adaptée aux différents niveaux de zoom**.

- L'expérience utilisateur est améliorée, grâce à une **visualisation progressive** : clusters pour vue large, points individuels pour vue rapprochée.



### Problème : Recherche sportive et filtrage des équipements

Une des fonctionnalités clés de l'application SportMap est de permettre à l'utilisateur de **filtrer dynamiquement** les équipements en fonction de plusieurs critères :

- type de sport,
- accès libre ou non,
- accessibilité aux personnes en situation de handicap.

Cependant, en combinant ces filtres, plusieurs problèmes techniques ont été identifiés lors du développement :

1. ! Les résultats n'étaient pas pertinents si les noms de sports étaient mal orthographiés ou accentués ;

- 
2. ⚠ Certains filtres (ex : “Accès handicapé”) s’appliquaient mal si des propriétés n’étaient pas correctement typées dans les objets GeoJSON ;
  3. 🧠 La combinaison de plusieurs filtres ralentissait l’affichage car tous les équipements étaient rechargés à chaque modification.
- 



## Solutions mises en œuvre



### 1. Normalisation des chaînes de caractères

Pour garantir une correspondance entre la recherche utilisateur et les données de l’API, j’ai mis en place une **fonction de normalisation** des chaînes, en supprimant la casse et les accents :

```
const normalizedSearchTerm = searchTerm
    .toLowerCase()
    .normalize("NFD")
    .replace(/[\u0300-\u036f]/g, "");
```

Cela permet de faire correspondre une recherche comme “foot” avec des équipements enregistrés sous la forme “Football”, “football”, ou même “fóotbâll”.

---



### 2. Fonction de filtrage unifiée et évolutive

Toutes les règles de filtrage sont regroupées dans une **fonction dédiée `getFilteredFeatures()`**. Elle s’exécute **localement**, sans nouvelle requête API, en filtrant directement les objets de la FeatureCollection via les propriétés suivantes :

- `feature.properties.sports` (sport concerné),
- `feature.properties.free_access` (accès libre),
- `feature.properties.inst_acc_handi_bool` (accès handicapé).

Les filtres actifs sont stockés dans `activeFilters`, et mis à jour via des boutons ou des paramètres d’URL.

---



### 3. Optimisation des performances

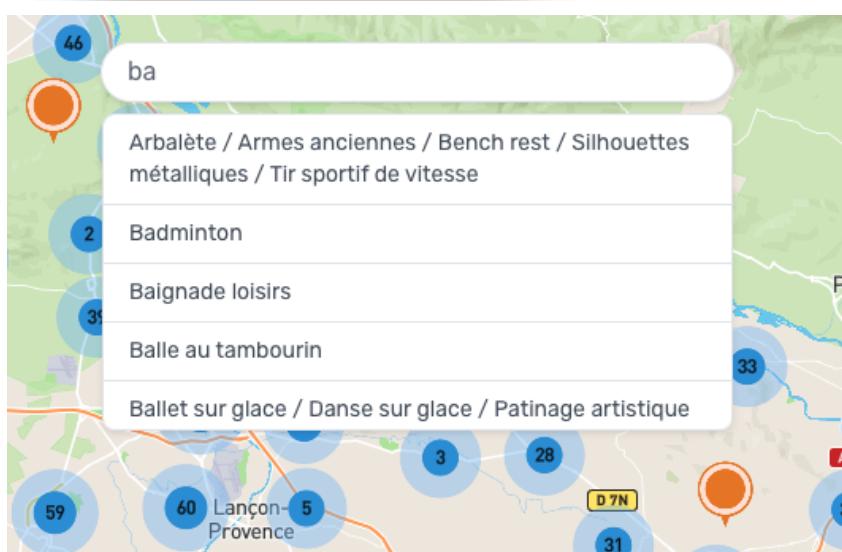
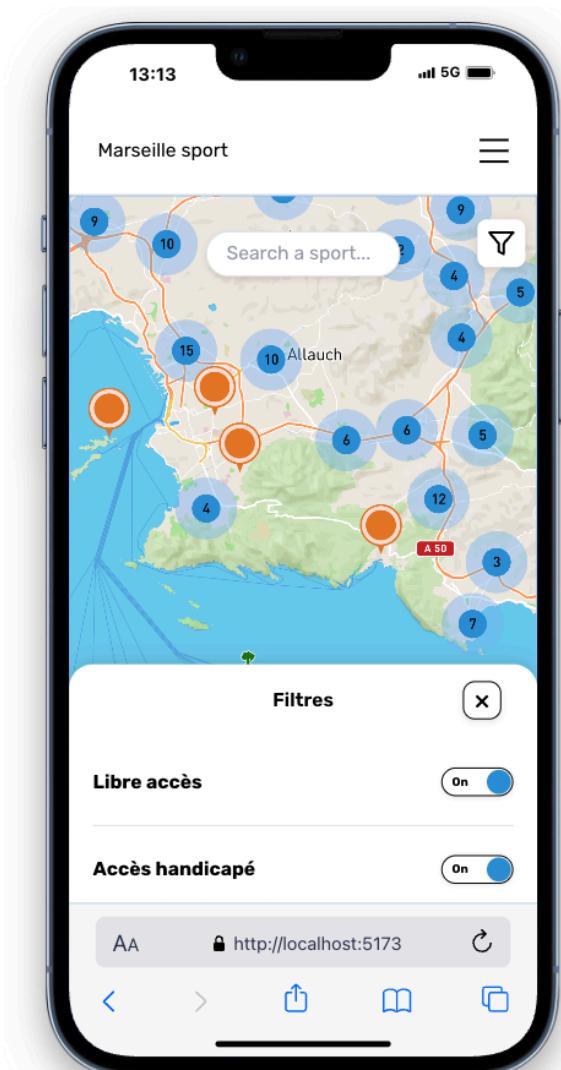
En cas d'absence de filtre, les équipements complets sont affichés. En revanche, dès qu'un filtre est actif :

- le nombre de points est réduit,
  - la carte reste fluide,
  - les clusters s'adaptent automatiquement.
- 



### Résultats

- La recherche est désormais **tolérante aux fautes de frappe**, fluide et instantanée.
- Les filtres sont **combinables sans conflit**.
- L'utilisateur voit **immédiatement les changements** sur la carte sans rechargement.



## 9. Évaluation du projet et tests



### Démarche qualité mise en place

Afin de garantir la robustesse et la fiabilité du projet **SportMap**, une phase de tests fonctionnels et d'évaluation continue a été mise en œuvre tout au long du développement. Elle s'est articulée autour de trois grands axes :

1. **Tests manuels fonctionnels** par scénarios utilisateurs ;
2. **Vérifications de la qualité visuelle et UX sur plusieurs supports** (mobile / desktop) ;
3. **Tests API côté back-end avec les outils fournis par Django** (shell, admin, console).



### Tests réalisés (exemples concrets)

Fonctionnalité testée	Méthode de test	Résultat attendu	Résultat obtenu
Chargement de la carte	Manuel (navigateur)	Carte s'affiche au bon centre / bon zoom	<input checked="" type="checkbox"/> Conforme
Filtrage par sport	Manuel (recherche + clic)	Affichage uniquement des points du sport choisi	<input checked="" type="checkbox"/> Conforme
Affichage du bon icône sur chaque point	Visuel / manuel	Icône selon l'état actif ou non	<input checked="" type="checkbox"/> Conforme

Partage d'équipement par URL	Paramètres d'URL	Zoom automatique et popup ouverte	<input checked="" type="checkbox"/> Conforme
Clustering dynamique des points	Interaction / zoom	Clusters apparaissent/déclenchés à bon niveau	<input checked="" type="checkbox"/> Conforme
Comportement responsive mobile	DevTools navigateur	Mise à l'échelle et accessibilité optimales	<input checked="" type="checkbox"/> Conforme
Requête API avec fetch()	Console navigateur	Réponse au bon format JSON, traitement correct	<input checked="" type="checkbox"/> Conforme



## Retours utilisateurs simulés

Bien qu'aucun test utilisateur officiel n'ait encore été organisé, plusieurs **collègues développeurs** de la formation ont testé différentes fonctionnalités et formulé des retours pertinents, notamment sur :

- l'**ergonomie des filtres**,
- la **lisibilité des clusters**,
- le **temps de chargement de la carte**.

Ces retours ont été intégrés à chaque sprint de développement, dans une logique d'amélioration continue.

---

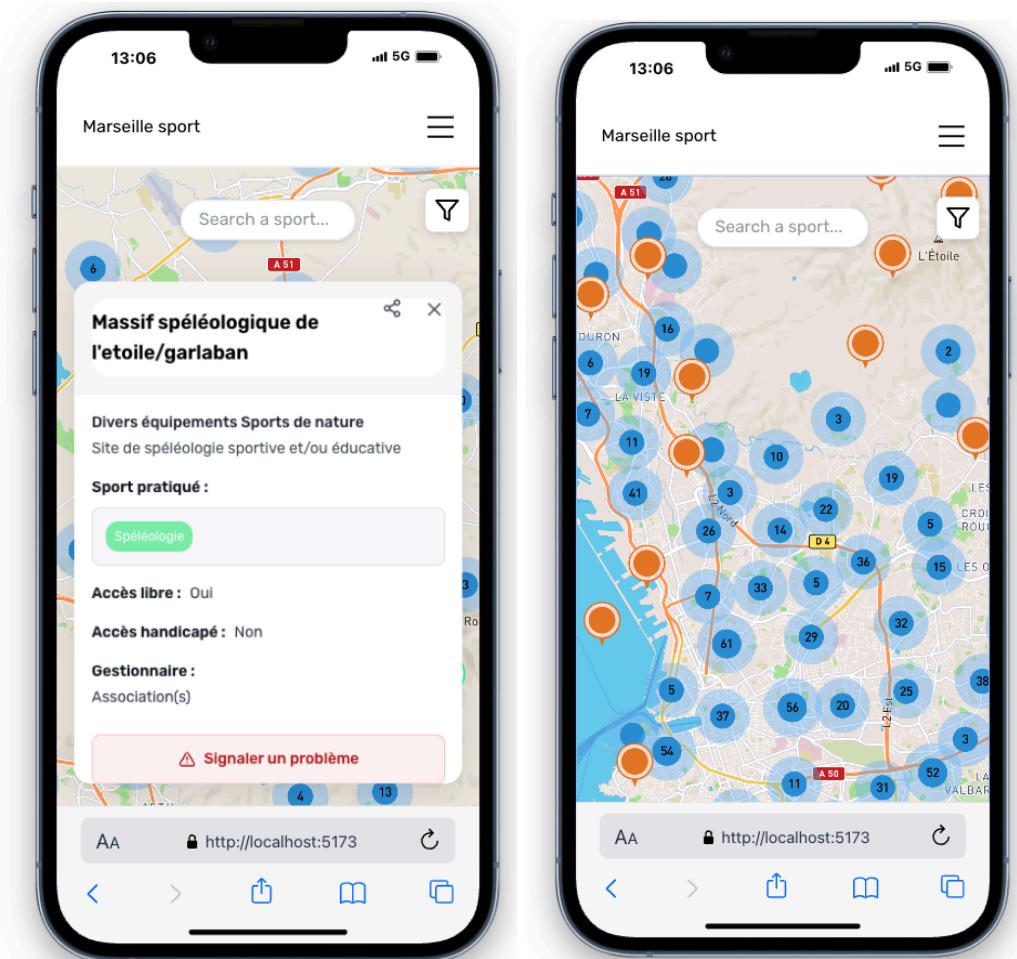


## Bilan technique des tests

Les tests ont permis de **valider la stabilité** de l'architecture MVT adoptée, la séparation des responsabilités entre le front-end (React, Vite) et le back-end (Django REST + PostgreSQL), ainsi que la fiabilité du système de filtrage dynamique et de rendu cartographique.

👉 *Captures recommandées ici :*

- un exemple de console fetch avec retour JSON ;



```

    useEffect(() => {
      const apiUrl = import.meta.env.VITE_API_URL;
      fetch(`${apiUrl}/geojson/`)
        .then(res => {
          if (!res.ok) {
            throw new Error('Network response was not ok');
          }
          return res.json();
        })
        .then(data => {
          console.log('Equipments:', data);
          setEquipments(data);
          setLoading(false);
        })
        .catch(error => {
          setError(error);
          setLoading(false);
        });
    }, []);
  }

  return { equipments, errorEquipments, loadingEquipments };
}

```

## 10. Perspectives et évolutions futures

Dans une optique d'amélioration continue, SportMap envisage de nombreuses évolutions pour enrichir l'application et l'adapter aux besoins de sa communauté. Les prochaines versions incluront de nouvelles fonctionnalités sociales et collaboratives destinées à dynamiser l'usage, ainsi qu'un élargissement géographique et partenarial visant à étendre la plateforme à l'échelle nationale. Par ailleurs, des avancées techniques sont planifiées pour renforcer l'efficacité, l'ergonomie et l'accessibilité de SportMap.

### 10.1 Nouvelles fonctionnalités prévues

Plusieurs modules seront développés pour favoriser l'interaction entre les utilisateurs et l'enrichissement participatif des données :

- **Partage social entre utilisateurs** : mise en place d'un fil d'actualité et de fonctionnalités de partage (commentaires, likes, recommandations) pour permettre aux membres de signaler leurs terrains favoris, de laisser des avis sur les équipements visités ou d'échanger des conseils. Cette

dimension sociale créera un véritable réseau collaboratif autour des infrastructures sportives.

- **Statistiques d'utilisation** : intégration d'un tableau de bord d'administration fournissant des indicateurs d'usage (nombre de visites, équipements les plus recherchés, fréquentation des pages, etc.). Ces statistiques permettront d'orienter les décisions de développement et d'évaluer l'impact de SportMap.
- **Ajout de terrains via une interface dédiée** : élaboration d'un module intuitif permettant aux utilisateurs authentifiés (après validation) de proposer de nouveaux équipements, de compléter ou de corriger les informations existantes. Chaque saisie sera soumise à une vérification avant publication, assurant la fiabilité des données.
- **Système de modération et signalement collaboratif** : implémentation d'un mécanisme participatif de remontée d'anomalies (fiches inexactes, équipement fermé, photo obsolète, etc.). À l'instar des applications municipales comme Imagina qui proposent un « service de signalement de problèmes », SportMap permettra à la communauté de signaler rapidement toute incohérence, garantissant ainsi l'actualité et la qualité des informations. Les rapports des utilisateurs seront ensuite validés ou modérés par les administrateurs.

Ces évolutions fonctionnelles s'inscrivent dans une démarche agile, où chaque nouvelle version sera enrichie par les retours d'usage et testée auprès d'un groupe pilote avant déploiement global.

## 10.2 Extension nationale et partenariats stratégiques

Au-delà de la feuille de route fonctionnelle, SportMap ambitionne de passer d'un périmètre local à un déploiement national. Après une phase pilote réussie sur une ville ou une agglomération, le projet s'étendra progressivement à d'autres communes et régions. Cette montée en charge sera planifiée en plusieurs étapes pour assurer la scalabilité de l'architecture et l'adaptation aux spécificités locales (données géographiques, règlementations, langues si nécessaire, etc.). L'objectif est qu'à terme, SportMap référence l'ensemble des équipements sportifs français, en commençant par couvrir les grandes villes puis en élargissant aux territoires plus ruraux.

La réussite de cette extension reposera sur des partenariats étroits avec les acteurs publics et associatifs locaux :

- **Collectivités territoriales (communes, intercommunalités, départements, régions)** : collaboration pour accéder aux données officielles d'équipement (plans d'urbanisme, budgets d'entretien, etc.), pour relayer SportMap auprès des citoyens (newsletter municipale, affichages, réseaux sociaux institutionnels) et pour garantir la mise à jour régulière des informations. Les collectivités pourraient également valoriser SportMap en tant que service aux administrés, encourageant les clubs et le grand public à contribuer au projet.
- **Associations et fédérations sportives** : implication des structures sportives locales (clubs, fédérations, ligues scolaires, etc.) pour promouvoir l'application auprès de leurs membres et intégrer l'inventaire de leurs installations. Ces partenaires pourront enrichir la base de données (par exemple, en signalant de nouveaux terrains annexes, des salles municipales ou des partenariats avec des gymnases privés) et utiliser les statistiques de fréquentation pour mieux

comprendre les besoins de la pratique sportive sur leur territoire.

- **Acteurs privés et fournisseurs de services** : ouverture de discussions avec des entreprises locales (clubs de fitness, événements sportifs, équipementiers) pour des services connexes, ainsi qu'avec des organismes publics (offices de tourisme, académies, services d'éducation) afin d'assurer une visibilité accrue. Ces partenariats permettront d'établir SportMap comme un outil de référence pour l'accès aux infrastructures sportives, tout en créant un écosystème vertueux de partage d'information et de promotion de l'activité physique.

Grâce à cette stratégie de déploiement territorial couplée à un réseau de partenariats, SportMap vise à devenir un service largement adopté au niveau national, facilitant l'orientation et la pratique sportive pour tous les citoyens français.

### 10.3 Évolutions techniques et innovations

Sur le plan technique, plusieurs axes d'amélioration et d'innovation sont envisagés pour pérenniser et enrichir SportMap :

- **Intégration d'API et services externes** : connexion avec des API partenaires pour enrichir l'expérience utilisateur. Par exemple, l'intégration d'un service de météo locale pourra aider les sportifs à planifier leurs activités, tandis que l'ajout d'API de transport en commun (Vélo'v, tramway, bus) facilitera l'accès aux sites. SportMap pourra aussi exploiter des API d'annonces locales pour promouvoir des événements sportifs ou des créneaux de réservation sur certains terrains.
- **Exploitation des données ouvertes (Open Data)** : recueil automatique des jeux de données publics existants. Notamment, l'application pourra tirer parti de la base « Data ES » du Ministère des Sports, qui recense plus de 330 000 équipements sportifs en France. L'accès à cette source officielle, régulièrement mise à jour, garantira une couverture initiale exhaustive des infrastructures. Comme le souligne une étude récente, la mise à disposition des données sportives ouvertes « va continuer à autonomiser les citoyens et tous les acteurs concernés grâce à des applications modernes dans ce domaine ». SportMap s'appuiera sur cette logique d'open data pour sécuriser et enrichir son contenu.
- **Amélioration de l'interface (UX/UI) et accessibilité** : refonte progressive de l'ergonomie et du design pour rendre l'application toujours plus intuitive et adaptée aux usages mobiles et web. Ceci inclut la mise en œuvre des bonnes pratiques (expérience utilisateur, design d'interaction) et la conformité aux normes d'accessibilité (pour malvoyants ou handicapés moteurs, selon les référentiels RGAA). L'objectif est de garantir une navigation fluide, un temps de chargement optimisé et une accessibilité élargie (choix de contrastes, compatibilité écran sombre, prise en charge de lecteurs d'écran, etc.).
- **Analyse des données et fonctionnalités prédictives** : au-delà des simples statistiques d'utilisation, SportMap explorera l'utilisation de l'analyse de données avancée (big data, machine learning) pour anticiper les besoins. Par exemple, en étudiant les tendances de fréquentation des lieux (périodes de pic d'affluence, sports émergents), il sera possible d'émettre des recommandations sur l'ouverture de nouveaux terrains ou la maintenance préventive des installations. Des outils d'analyse prédictive pourront ainsi aider les gestionnaires d'équipements

à planifier la modernisation et l'entretien en fonction de l'évolution des usages.

Ces développements techniques seront menés dans un cadre agile et modulaire, afin de réagir rapidement aux retours utilisateurs et aux évolutions du marché. La feuille de route privilégiera la qualité logicielle (tests automatisés, intégration continue) et la sécurité (protection des données personnelles, conformité RGPD).

En synthèse, SportMap entend consolider sa base fonctionnelle tout en ouvrant son horizon géographique et partenarial. Le projet s'inscrit dans une vision à long terme : devenir l'outil référent de cartographie des infrastructures sportives en France, fruit d'une coopération étroite entre développeurs, usagers, collectivités et associations. Les évolutions futures décrites ci-dessus illustrent cette ambition de croissance durable, appuyée sur l'innovation technologique et l'engagement communautaire.

## 11. Conclusion et bilan professionnel

La réalisation du projet *SportMap* s'est révélée être une expérience particulièrement formatrice, à la fois sur le plan technique, méthodologique et personnel. Elle m'a permis de mettre en pratique l'ensemble des compétences acquises au cours de ma formation à *La Plateforme*, tout en me confrontant aux exigences concrètes d'un projet de développement en conditions réelles.

### Bilan technique

Tout au long de ce projet, j'ai pu consolider et approfondir mes connaissances sur un socle technologique moderne et complet :

- **Côté backend**, l'usage de *Django* et de son ORM, couplé à *PostgreSQL* et conteneurisé via *Docker*, m'a permis de structurer une API robuste, évolutive et sécurisée.
- **Côté frontend**, j'ai pu mettre en œuvre *React* avec *Vite*, associé à *TailwindCSS* pour la partie design. L'intégration de *Mapbox GL JS* a apporté à l'application une cartographie interactive puissante et réactive.
- La mise en place d'une architecture MVT claire et la séparation des responsabilités (deux dépôts GitHub distincts, services organisés par fonctionnalité) ont contribué à un développement plus lisible, maintenable et scalable.

Ces choix technologiques, ainsi que les contraintes rencontrées (optimisation du rendu des points, clustering, filtrage dynamique...), m'ont amené à développer des réflexes d'ingénierie solides : découpage logique du code, gestion des dépendances, tests manuels, documentation, gestion des états et des appels asynchrones avec *fetch*, etc.

### Bilan méthodologique

Sur le plan méthodologique, *SportMap* a renforcé ma capacité à gérer un projet de bout en bout. En travaillant en binôme(également développeur), j'ai appris à organiser efficacement les tâches, à prioriser les besoins fonctionnels, et à maintenir une communication régulière pour coordonner nos actions.

La rédaction du cahier des charges, l'analyse des utilisateurs, la définition des MVP et le suivi d'un backlog ont été des jalons essentiels pour structurer notre avancement. J'ai également compris l'importance d'itérer, de recueillir des retours dès les premières étapes et de maintenir une documentation technique et fonctionnelle claire.

## Bilan personnel

D'un point de vue personnel, ce projet a renforcé ma confiance en mes capacités à concevoir, développer et finaliser une application concrète à fort impact social. Il m'a permis de m'approprier pleinement le rôle de concepteur développeur d'applications, en intégrant les dimensions techniques, organisationnelles et éthiques du métier.

Ce travail m'a également conforté dans mon souhait de contribuer à des projets à vocation utile : *SportMap* ne se limite pas à une application technique, il s'agit d'un outil au service de l'inclusion, de l'accessibilité et de la promotion de l'activité physique pour tous.

## Vers l'avenir

Cette expérience m'a ouvert de nombreuses perspectives : je me sens aujourd'hui prêt à intégrer une équipe technique, à poursuivre mes apprentissages autour de technologies avancées (DevOps, cybersécurité, IA...), ou à développer de nouveaux projets entrepreneuriaux ou citoyens.

Ce projet de fin d'année constitue ainsi un véritable tremplin vers le monde professionnel. Il marque l'aboutissement d'un parcours de formation exigeant, mais surtout le début d'un chemin de développeur conscient de l'impact que peuvent avoir les outils numériques sur la société.