



Accueil

Sommaire

Présentation

Conclusion

Page 01

RP2 Projet 2025

Projet TCB

Kit d'arbitrage et de suivi des performances des joueurs

Application web – statistiques : Site Web



Sommaire



Introduction

- Présentation du projet et de son contexte
- Rôle de l'étudiant 1 dans l'équipe

Objectifs de la partie

- Missions confiées
- Fonctions attendues dans l'application

Travail réalisé

- Backend : serveur Node.js, récupération des données
- Frontend : affichage, graphiques, comparaison
- Authentification et sécurité
- Collaboration avec l'étudiant 2

Difficultés rencontrées

- Problèmes techniques
- Solutions apportées

Travail en cours et prochaines étapes

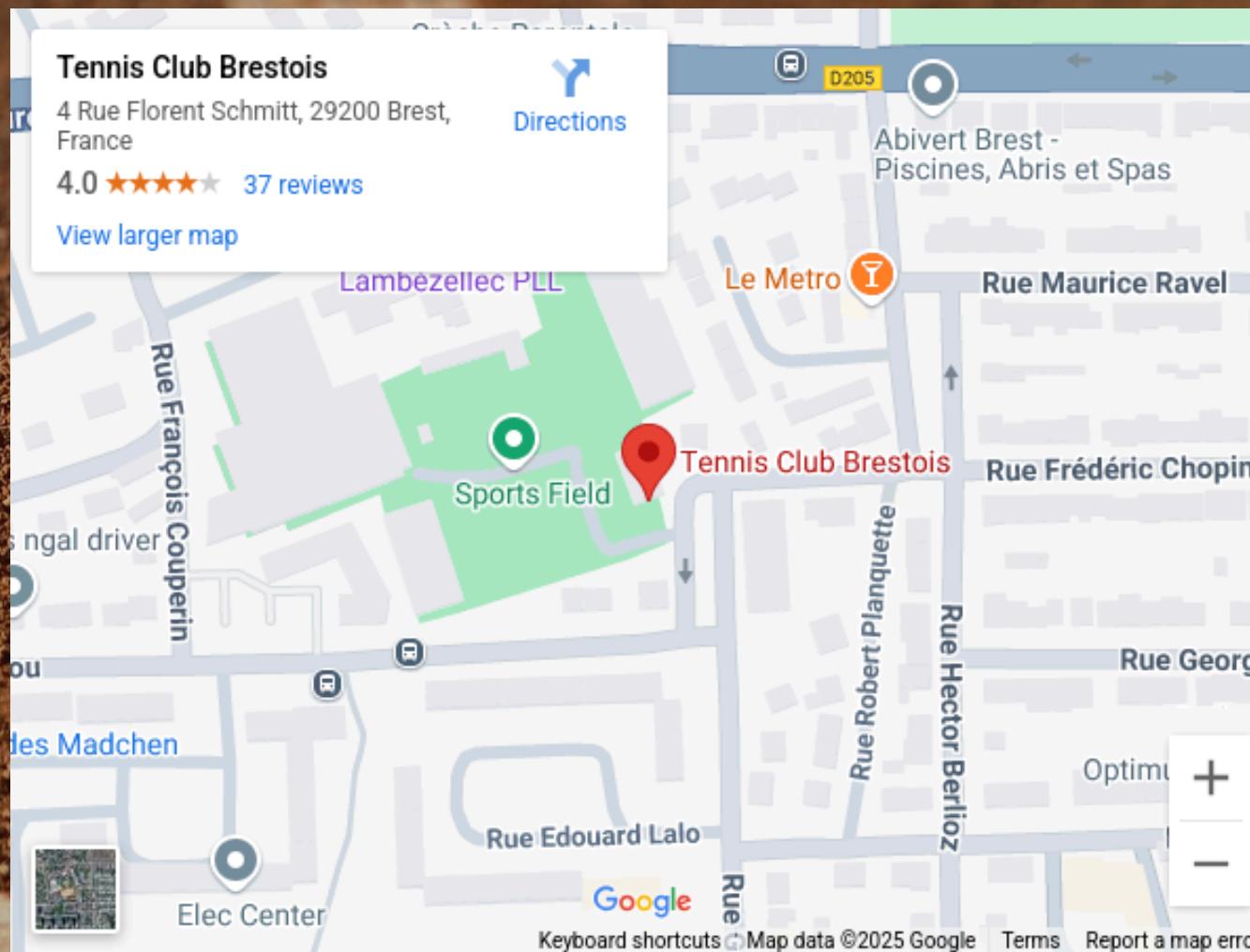
- Sécurité, graphique global, recette
- Tâches planifiées

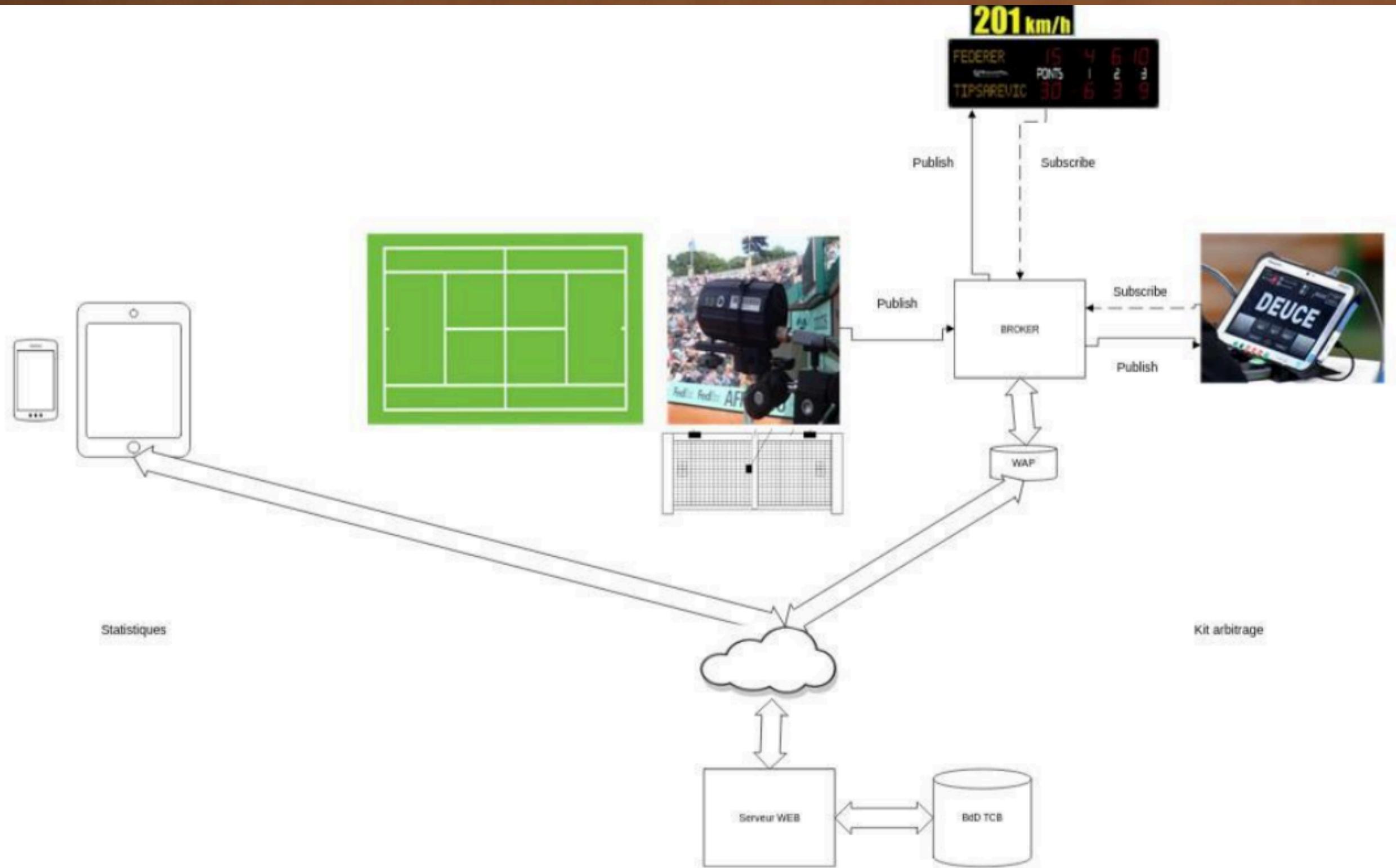
Conclusion

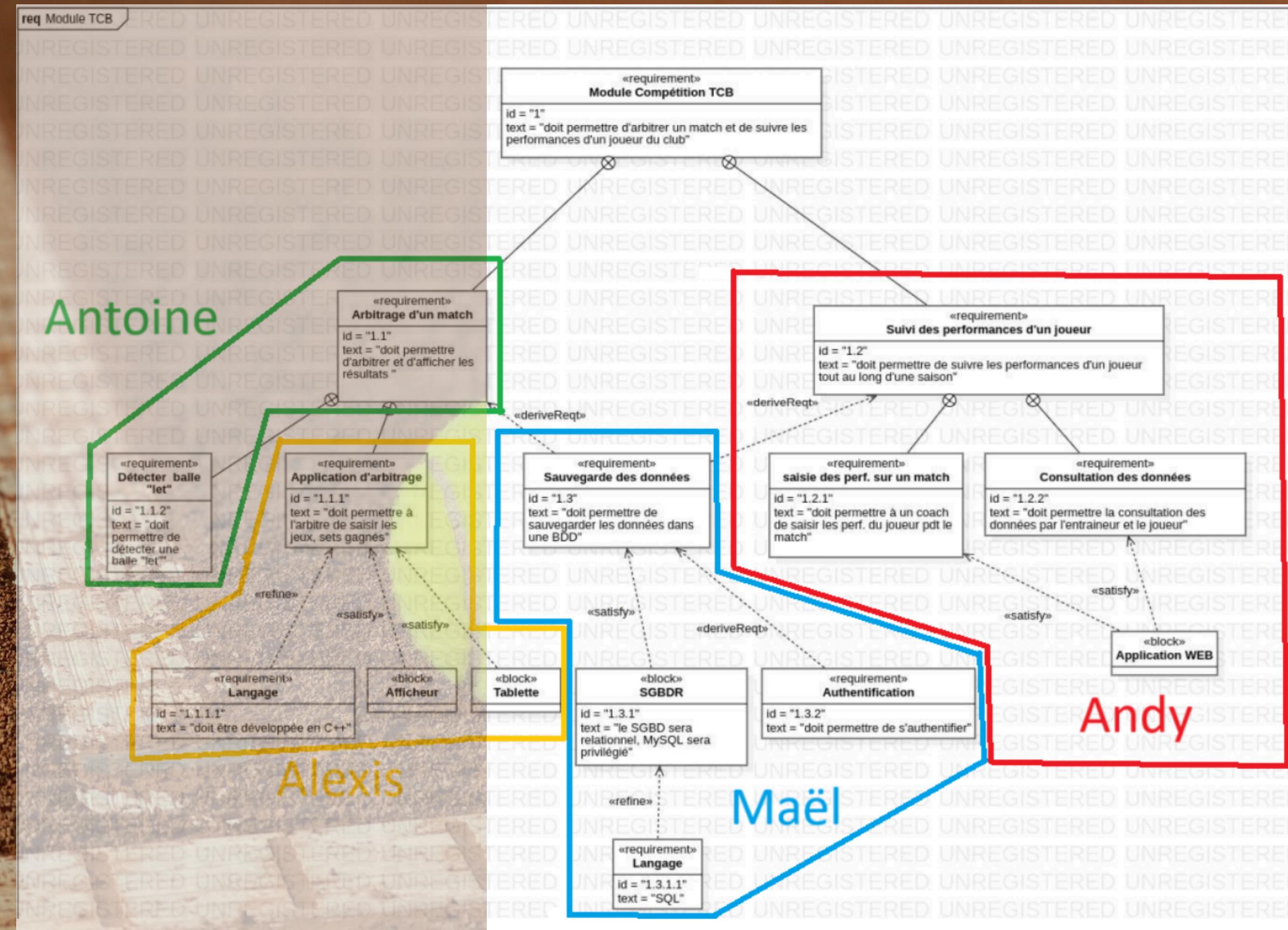
Introduction

- Projet commandité par le Tennis Club Brestois (TCB)

Objectif : développer une solution pour suivre et analyser les performances des joueurs

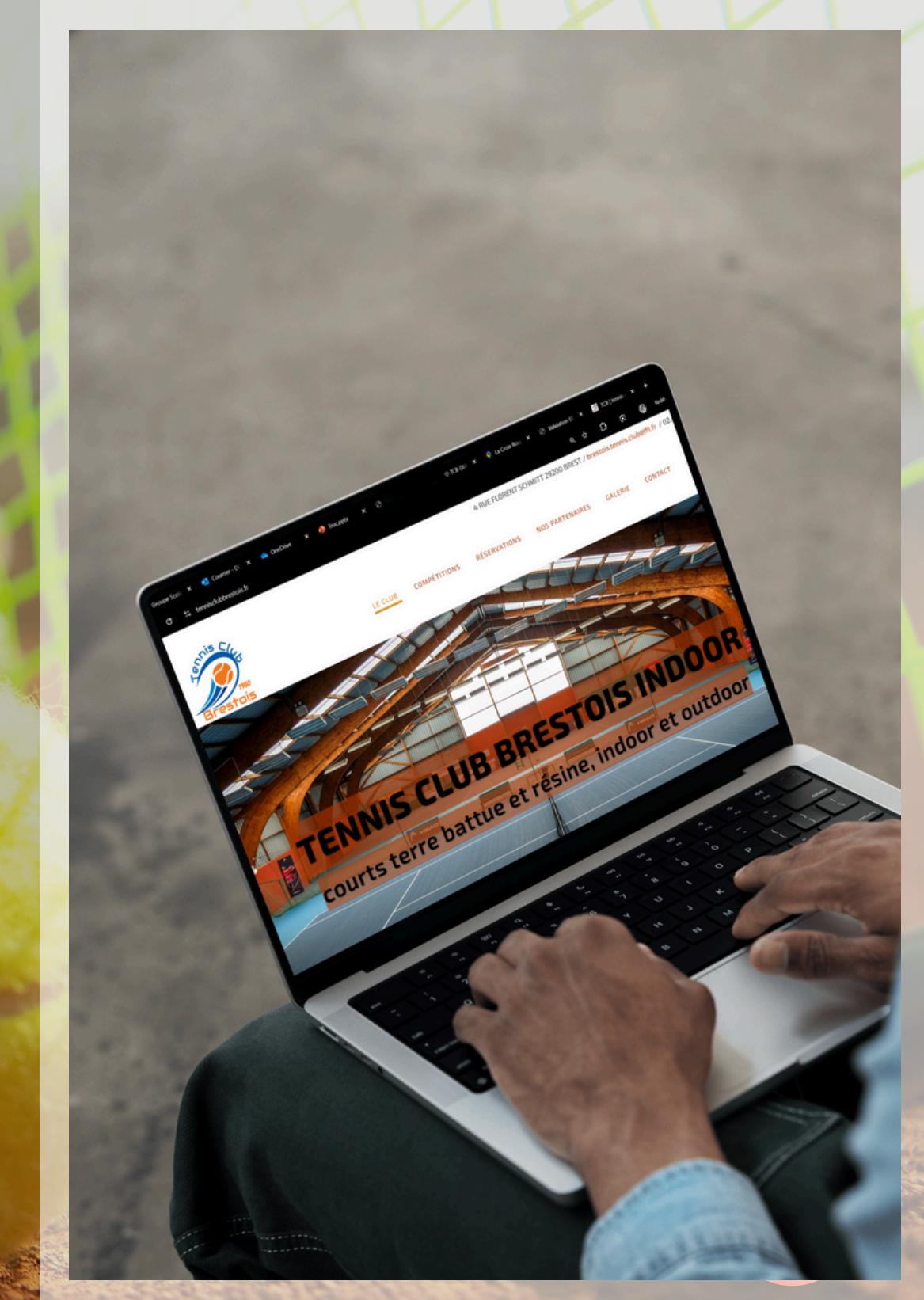


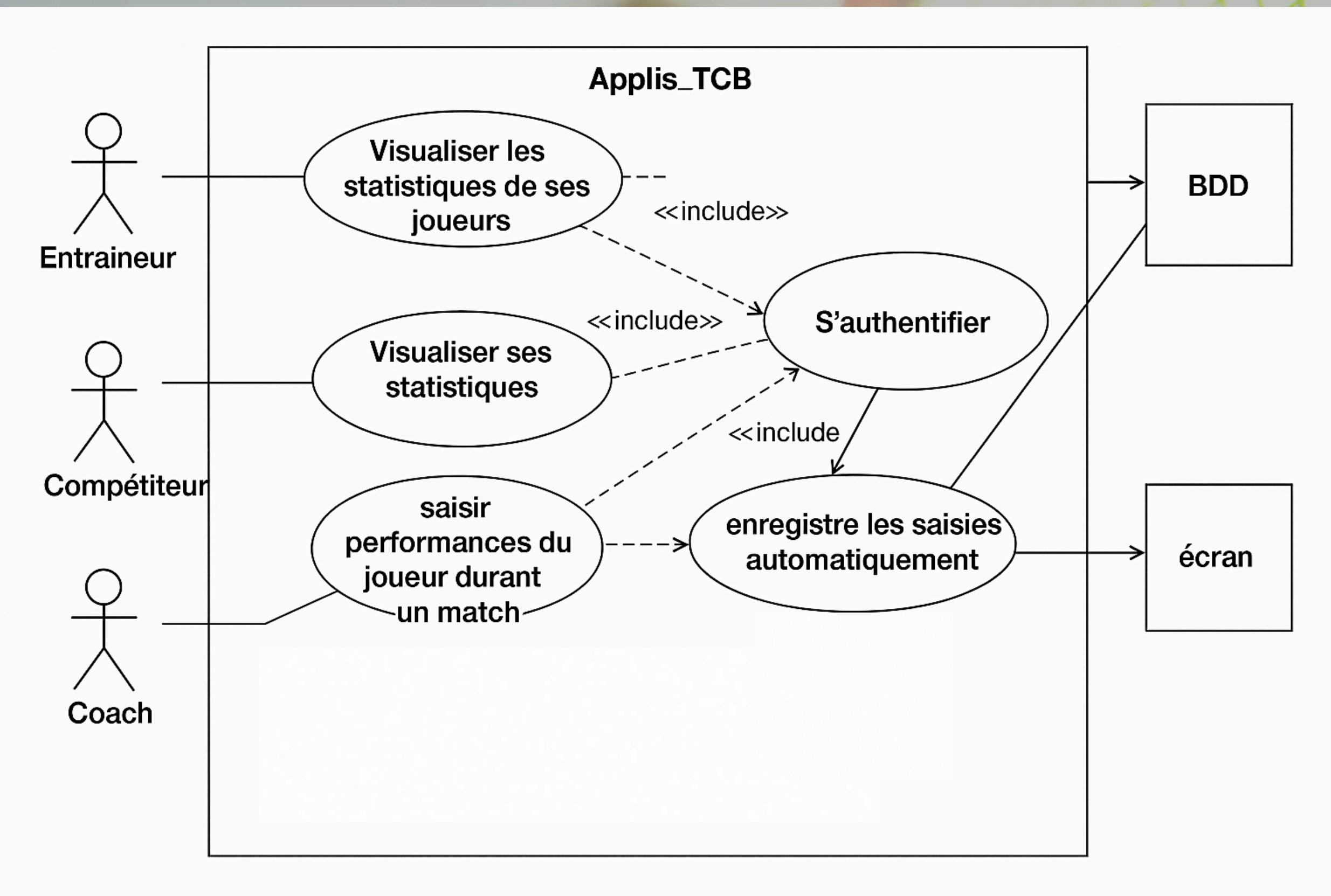




Objectifs de ma partie

- Créer une application web responsive (design + ergonomie)
- Afficher les statistiques des joueurs par match
- Générer des graphiques (match donné ou global)
- Permettre la comparaison entre deux joueurs
- Mettre en place une authentification (joueur / entraîneur / accompagnateur)





Travail réalisé : Backend

Structure du serveur Express (Node.js)

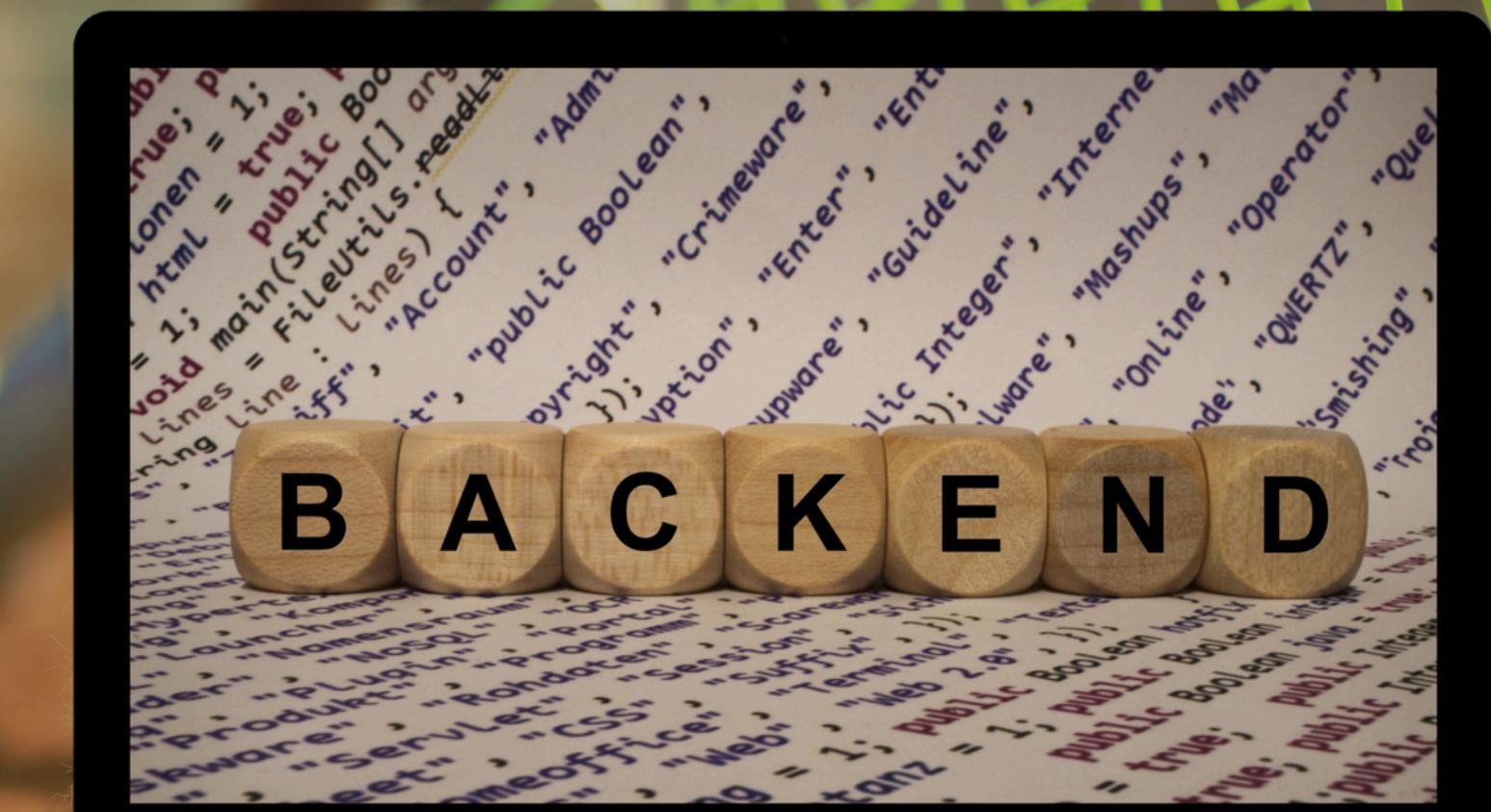
Connexion à la base de données MySQL via MQTT

Gestion des requêtes SQL

Traitement des statistiques

Gestion des erreurs

Authentification et sessions



Structure du site

```

Site_TCB_v1
> .vscode
> node_modules
public
> images
★ favicon.ico
# style.css
views
└ accueil-joueur.html.twig
└ accueil.html.twig
└ comparaison.html.twig
└ consultation-joueur.html.twig
└ consultation.html.twig
└ index.html.twig
└ login.html.twig
└ saisie-donnees.html.twig
JS script.js
└ statistiques_coach.html.twig
└ statistiques_joueur.html.twig
└ statistiques.html.twig
└ test.html.twig
└ 404.html.twig
{} package-lock.json
{} package.json
JS server.js

```

Accueil

Sommaire

Présentation

Conclusion

Page 08

Solutions utilisées

Composant	<input checked="" type="checkbox"/> Choix utilisé	<input type="checkbox"/> Autre solution possible	<input type="triangle-left"/> Pourquoi moins adaptée dans TON projet
Serveur web	Express.js	Fastify	Plus rapide mais moins de plugins et plus compliqué
Templating	Twig	EJS	Moins structuré que Twig pour la séparation HTML/JS
Authentification	express-session	JWT	Complexé à gérer pour les rôles + requiert plus de sécurité côté client
Sécurité mdp	bcrypt	argon2	Plus sécurisé, mais installation plus complexe et surdimensionnée ici
MQTT	mqtt (npm)	paho-mqtt en Python	Aurait impliqué un backend Python à part
Routing	Express natif	Koa.js	Plus compliqué
Requête SQL	SQL via MQTT	Appels directs REST sur la base	Dépendance forte entre frontend/backend
Sessions	express-session	Cookies HTTP + localStorage	Plus fragile, moins sécurisé, pas adapté à un accès multi-utilisateur

Tests unitaires

🔒 Authentification et gestion de session

- ✓ Connexion avec identifiants valides ---> Vérifie qu'un utilisateur accède bien à son espace
- ✓ Connexion avec identifiants invalides ---> Vérifie le refus et le message d'erreur
- ✓ Session correctement créée ---> Vérifie que req.session contient les bons champs
- ✓ Déconnexion ---> Vérifie que la session est détruite proprement
- ✓ Accès sans session ---> L'utilisateur non connecté est redirigé/refusé

🧠 Middleware de contrôle d'accès

- ✓ Accès interdit à une route protégée (joueur sans session) ---> Doit renvoyer 401 ou redirection
- ✓ Accès entraîneur à ses joueurs ---> L'entraîneur ne peut voir que ses joueurs
- ✓ Joueur ne peut pas accéder aux données d'un autre joueur ---> Vérifie l'isolation des données par ID/session

🔒 Sécurité

- ✓ Hachage d'un mot de passe avec bcrypt ---> Données salées dans la BDD
- ✓ Injection SQL simulée ---> La chaîne n'est jamais insérée dans la requête brute (si applicable)
- ✓ Bouton de déconnexion ---> Ne doit pas permettre de se reconnecter sans re saisir les identifiants

Travail réalisé : Frontend

- Développement de l'ensemble des pages de l'interface utilisateur (joueur, entraîneur, statistiques, comparaison...)
- Mise en place d'une interface fluide, claire et responsive, adaptée à tous types d'écrans
- Création de la page de comparaison entre deux joueurs
- Affichage dynamique d'un tableau comparatif clair et intuitif
- Génération d'un graphique radar interactif avec Chart.js
- Pop-up de visualisation des statistiques détaillées par match

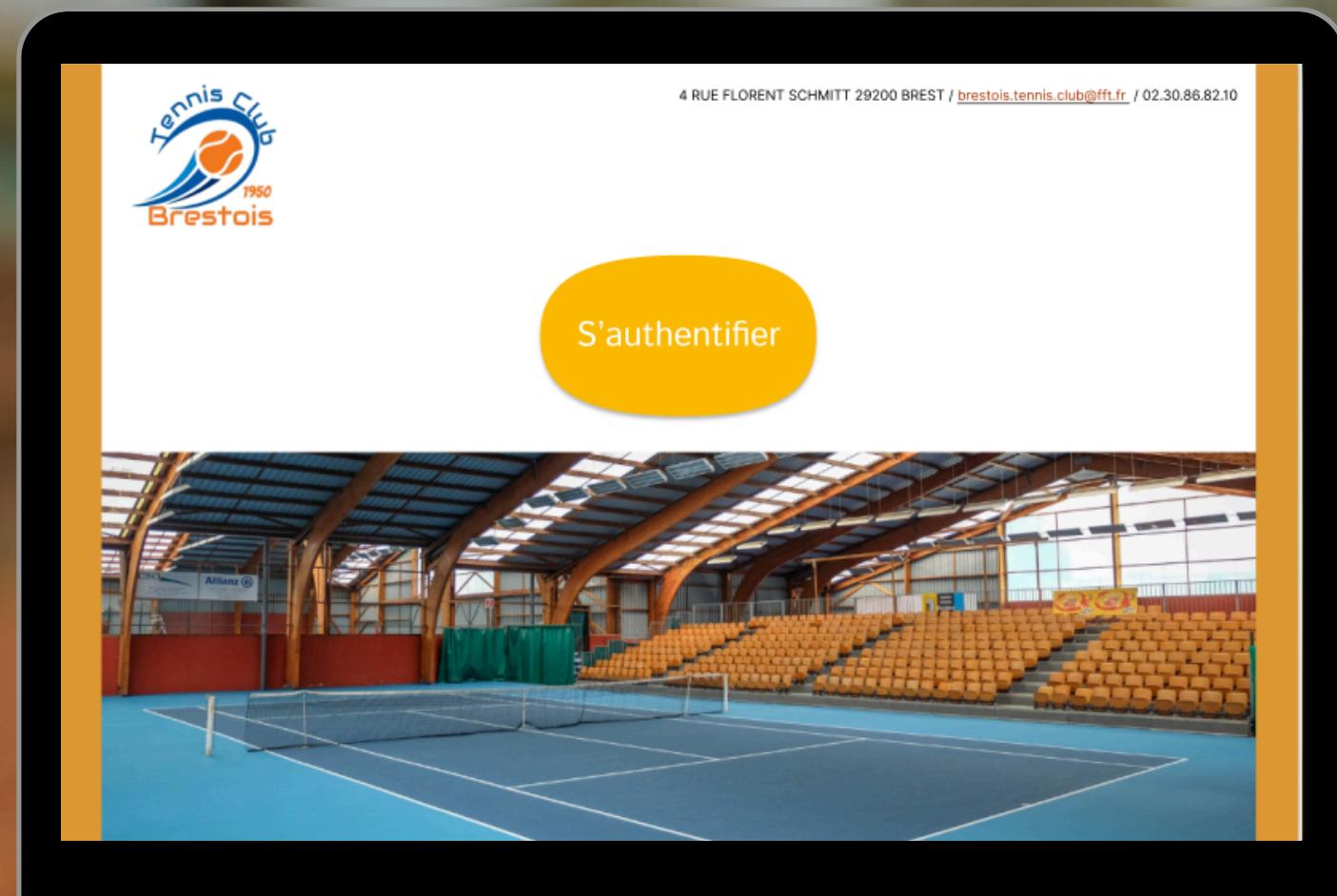


Solutions utilisées

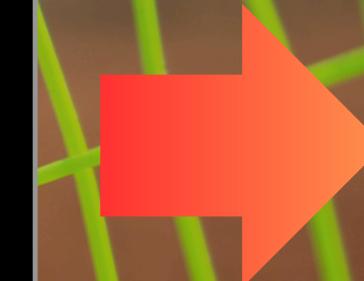


Composant	✓ Solution utilisée	✗ Alternative moins adaptée	⚠ Pourquoi moins adaptée
Affichage des pages HTML	Twig	HTML statique pur	Pas dynamique, duplication de code, aucun rendu conditionnel
Affichage graphique	Chart.js	Tableaux HTML simples	Pas visuel ni interactif, difficile à lire
Comportements dynamiques	JavaScript natif	Pas de JS ou JS inline	Pas d'interaction sans rechargement complet de la page
Design responsive	CSS media queries	Taille fixe en pixels	Inadapté aux écrans mobiles ou tablettes
Structure des fichiers	Twig + arborescence views	Tout dans un seul fichier HTML	Code illisible, non maintenable
Affichage contextuel (détail)	Pop-up JavaScript / modale HTML	Nouvelle page pour chaque vue	Lenteur, manque de fluidité dans l'interface

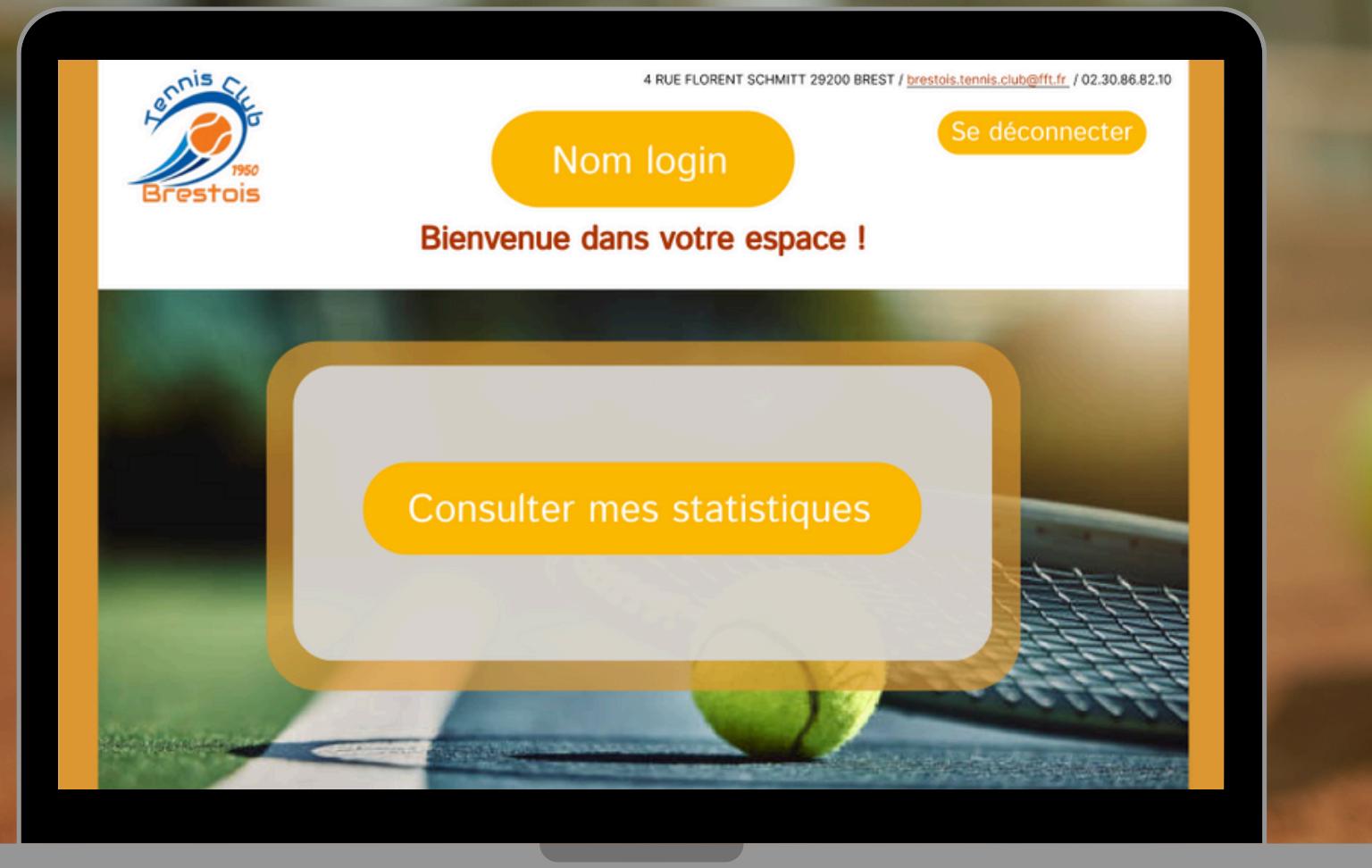
Mock-up:



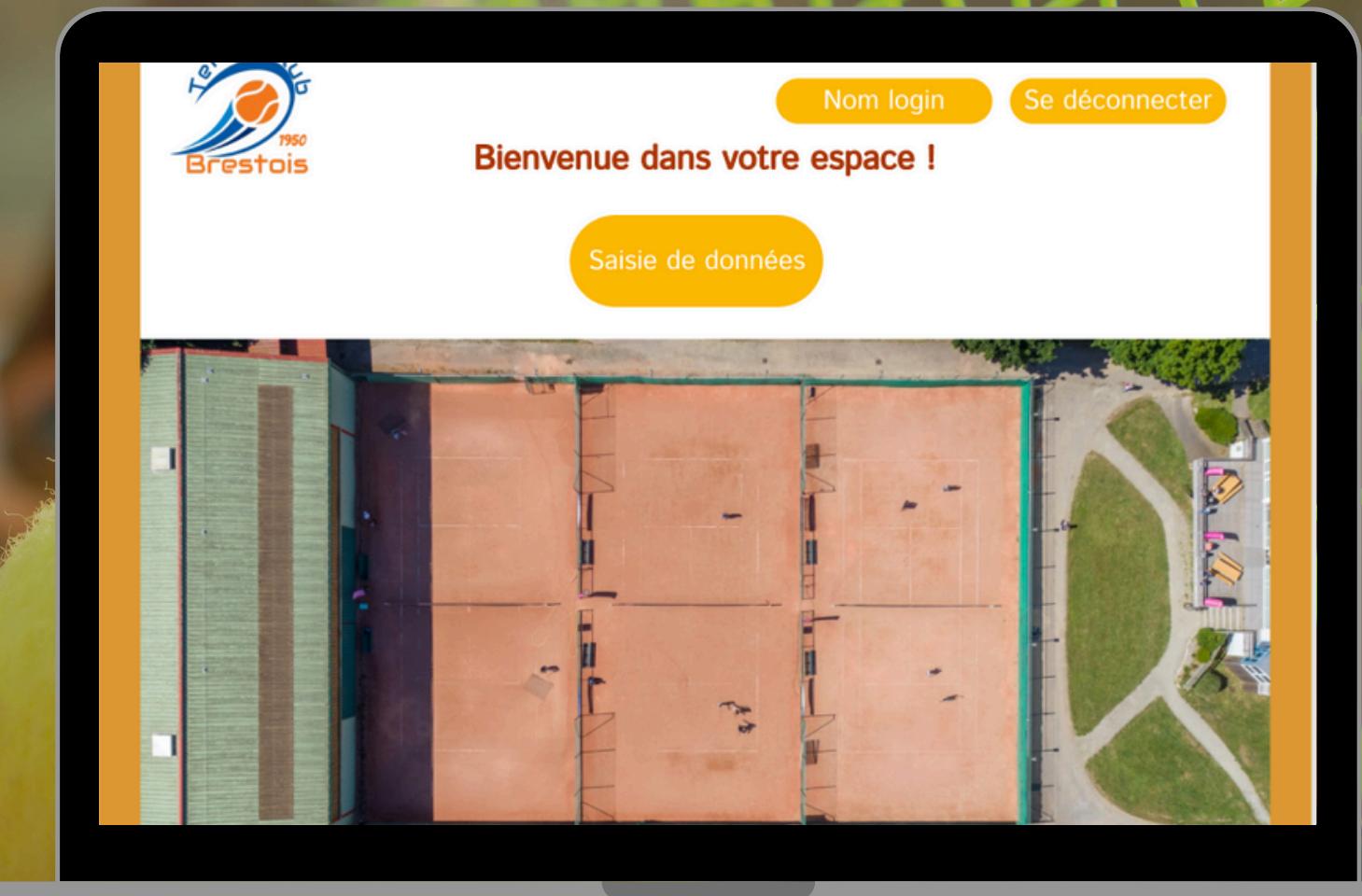
Figma



Mock-up:



Mock-up:





Accueil

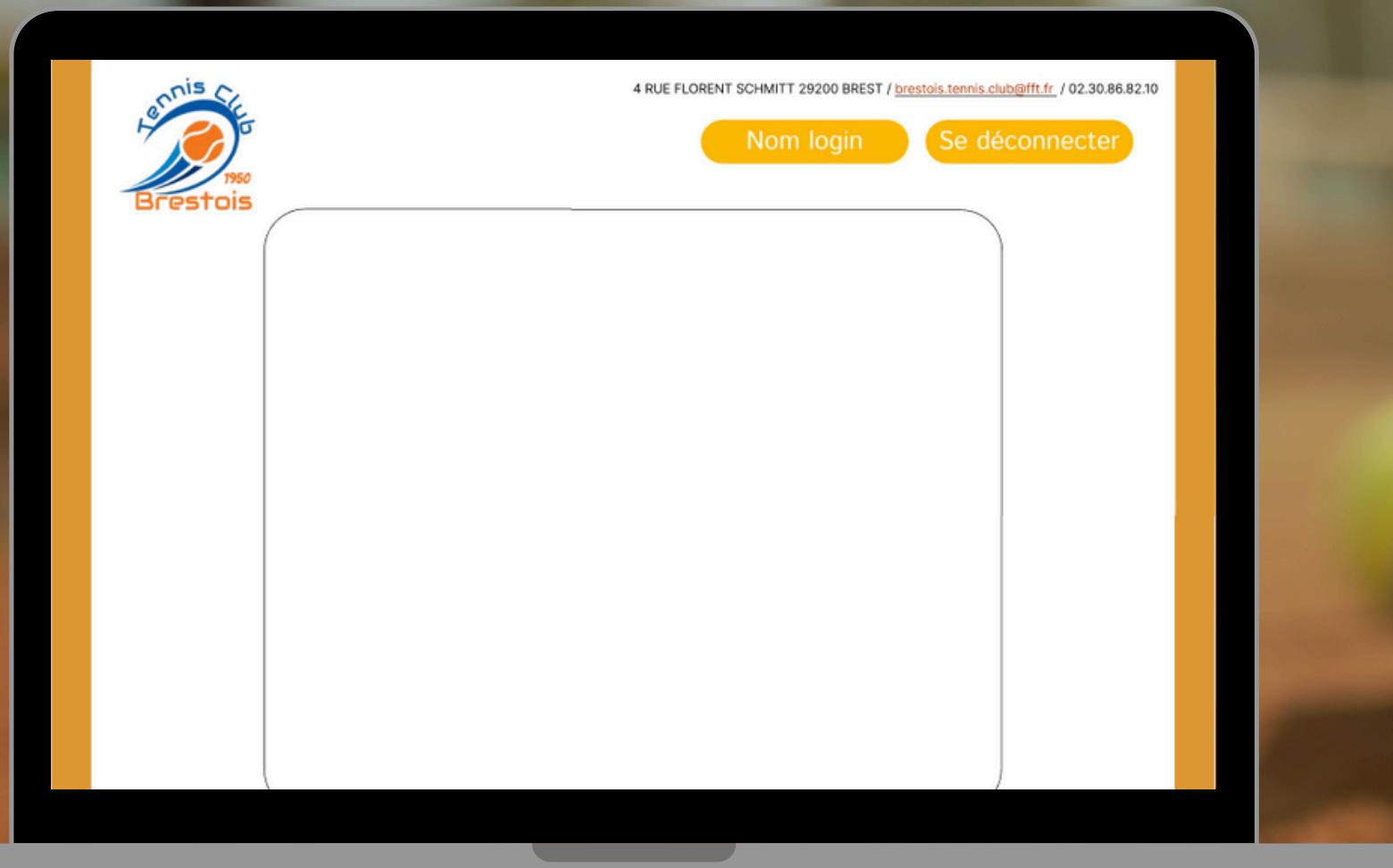
Sommaire

Présentation

Conclusion

Page 15

Mock-up:





Accueil

Somaire

Présentation

Conclusion

Page 16

Mock-up:



4 RUE FLORENT SCHMITT 29200 BREST / brestois.tennis.club@fft.fr / 02.30.86.82.10

[Nom login](#)
[Se déconnecter](#)

Choisissez le joueur pour consulter ses données

Joueur X

[Comparaison](#)
[Retour](#)

The screenshot shows a mobile application interface for the Tennis Club Brestois 1950. At the top left is the club's logo, featuring a blue tennis ball with orange stripes and a blue swoosh above it, with the text "Tennis Club Brestois 1950". At the top right are links for "Nom login" and "Se déconnecter". The main title "Statistiques du joueur" is centered above a table of player statistics. The background of the app is a blurred image of a tennis court with a yellow ball. A large orange arrow graphic is visible on the left side of the screen.

Statistique	Valeur
Aces	19
Winners	20
Fautes	3
Double fautes	3
Amortis	16
Smashs	28
Volées	18
Abandons	0

[Afficher les graphiques](#)



Accueil

Sommaire

Présentation

Conclusion

Page 17

Mock-up:



4 RUE FLORENT SCHMITT 29200 BREST / brestois.tennis.club@fft.fr / 02.30.86.82.10

Nom login [Se déconnecter](#)

Comparaison des joueurs

Statistique	Valeur
Aces	19
Winners	20
Fautes	3
Double fautes	3
Amortis	16
Smashs	28
Volées	18
Abandons	0

Statistique	Valeur
Aces	19
Winners	20
Fautes	3
Double fautes	3
Amortis	16
Smashs	28
Volées	18
Abandons	0

[Retour](#)

Connexion à la BDD

 alwaysdata

```
// Connexion à MySQL
const db_config = {
  host: 'mysql-lcr-mm.alwaysdata.net',
  user: 'lcr-mm',
  password: 'Azerty.1@',
  database: 'lcr-mm_tcb2'
};

let db;

function handleDisconnect() {
  db = mysql.createConnection(db_config);

  db.connect(function(err) {
    if (err) {
      console.error('Erreur de connexion à MySQL:', err);
      setTimeout(handleDisconnect, 2000); // attend 2s avant retry
    } else {
      console.log("Connexion à MySQL réussie !");
    }
  });

  db.on('error', function(err) {
    console.error('Erreur MySQL :', err);
    if (err.code === 'PROTOCOL_CONNECTION_LOST' || err.code === 'ETIMEDOUT') {
      handleDisconnect(); // reconnecte automatiquement
    } else {
      throw err;
    }
  });
}

handleDisconnect(); // Lancer la première connexion

db.connect((err) => {
  if (err) {
    console.error('Erreur de connexion à MySQL:', err);
    return;
  }
  console.log('Connexion à MySQL réussie');
});
```

Gestion de l'authentification

```
// Configuration des sessions
app.use(session({
  secret: 'secret_key',
  resave: false,
  saveUninitialized: true,
  cookie: { secure: false, maxAge: 3600000 } // 1h de session
}));
```

```
// Redirection en fonction du statut
let redirectUrl = "/";
if (user.statut === 1) redirectUrl = "/accueil-joueur";
if (user.statut === 2) redirectUrl = "/accueil";
if (user.statut === 3) redirectUrl = "/accueil";
```

```
// Route de connexion
app.post("/login", async (req, res) => {
  const { email, password } = req.body;
  console.log(`Tentative de connexion : ${email}`);

  if (!email || !password) {
    return res.json({ success: false, error: "Veuillez remplir tous les champs." });
  }

  const sql = "SELECT id_personne, email, mot_de_passe, statut FROM Personnes WHERE email = ?";
  // ... exécution SQL + vérification mot de passe ...
});
```

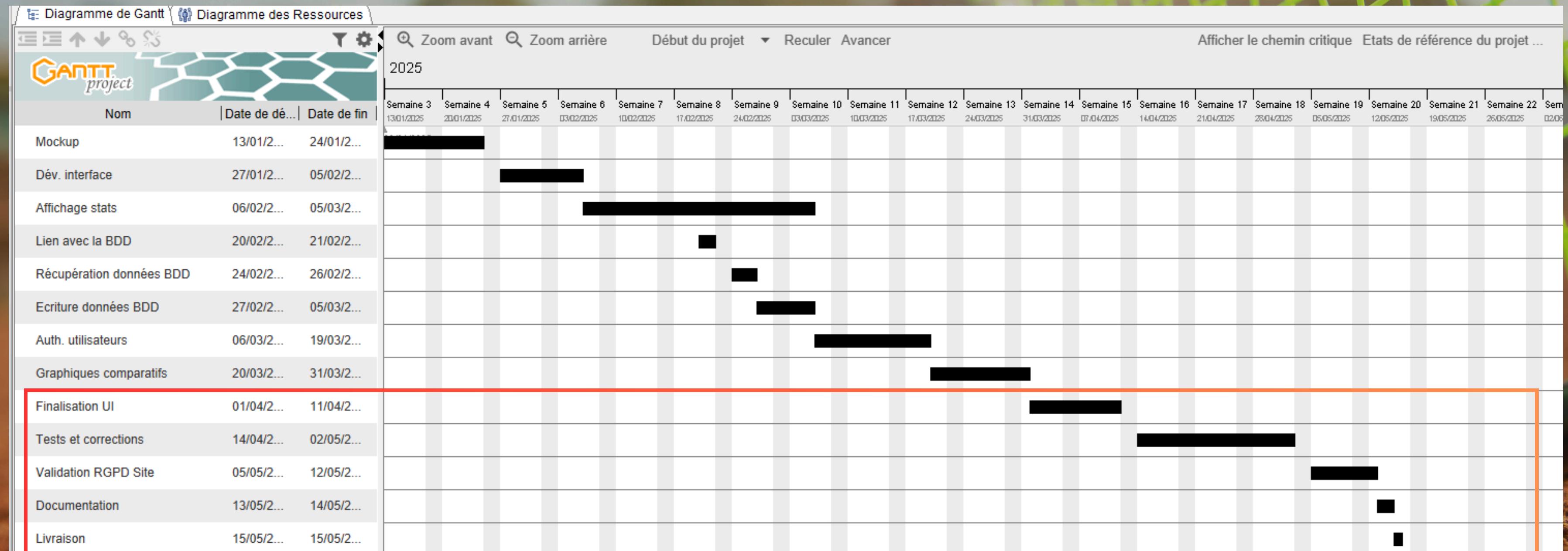
```
app.get('/accueil-joueur', checkAuth, (req, res) => {
  if (req.session.statut !== 1) {
    return res.status(403).send("Accès interdit");
  }

  const id = req.session.userId;
```

Difficultés rencontrées

 Incompatibilité entre la version Node.js et certains modules	Version trop récente en local vs version plus ancienne sur la machine cible	Retour à Node.js LTS + remplacement de mysql2/promise par mysql classique
 Erreurs undefined dans la page de comparaison	Changement de structure des objets retournés par le backend	Reprise de l'ancienne structure stats[0] pour compatibilité
 Sélection indépendante des matchs sur la comparaison	Complexité de gestion côté frontend et backend pour chaque joueur	Condition dynamique dans les routes + filtrage JS côté client
 Accès non autorisé à certaines pages	Mauvaise gestion initiale des rôles utilisateurs dans les sessions	Ajout d'un middleware de contrôle avec req.session.statut
 Lenteur au premier appel de données (MQTT)	Temps de connexion au broker ou SQL long	Relance automatique
 Affichage non responsive sur certains écrans	CSS en px fixe ou éléments qui débordaient	Réécriture avec media queries pour adapter le rendu mobile
 Graphiques Chart.js qui s'affichent vides	Aucune donnée récupérée de la base pour certains matchs	Vérification JS avant de générer le graphique

Suivi du planning



Conclusion

- L'ensemble des fonctionnalités principales sont développées et fonctionnelles (responsive, authentification, statistiques, comparaison, affichage graphique)
- Le site est responsive, fluide et adapté à différents profils d'utilisateurs (joueur / entraîneur)
- Plusieurs problèmes techniques ont été rencontrés (compatibilité Node.js, traitement des données, MQTT...) et résolus
- La communication avec l'équipe a permis d'intégrer les contraintes liées à la base de données et à la structure du projet

