Updated saved memory

🔥 **Bravo Nayer** pour l’idée ! Un site web pour salle de sport avec BI et Machine Learning, ça peut être un projet de stage puissant et très valorisant 💼💪. Voici une **proposition complète de projet**, bien structurée, avec stack MERN, Machine Learning et Power BI.

**🧠 Nom du projet : GYM360 – Smart Gym Management & Analytics**

**🎯 Objectif du projet**

Créer une plateforme web intelligente de gestion de salle de sport, intégrant :

* Portail client (inscription, abonnement, suivi sportif)
* Espace admin (gestion utilisateurs, planning coachs, produits, paiements)
* Tableau de bord BI via **Power BI**
* Module ML pour :
  + Recommandations d'entraînement personnalisées
  + Analyse de churn (perte d’abonnés)
  + Prévision de revenus et fréquentation

**🏗️ Architecture professionnelle**

**⚙️ Frontend – React.js (Next.js en option)**

* Authentification (JWT)
* Interface client (accès aux séances, statistiques, progression)
* Interface admin (gestion, visualisation, export)
* Dashboard avec graphiques dynamiques (recharts, apexcharts...)

**🧩 Backend – Node.js + Express**

* API RESTful pour utilisateurs, abonnements, paiements, statistiques
* Auth middleware (JWT)
* Connexion à MongoDB
* Intégration avec Power BI (export CSV/JSON + automatisation)

**🗄️ Base de données – MongoDB**

* Modèle User, Coach, Session, Subscription, Product, Stats

**🤖 Machine Learning – Python (service à part ou microservice Flask/FastAPI)**

* Algorithmes :
  + KMeans pour segmentation clients
  + Random Forest pour prédire churn
  + Recommender system pour séances
* Export des résultats vers le backend via REST API

**📊 Business Intelligence – Power BI**

* Connexion à la base ou fichiers générés (abonnements, revenus, churn)
* Rapports dynamiques (fréquentation, satisfaction, prévision)

**🔁 Fonctionnalités principales**

**👤 Côté Client**

* Inscription / Paiement abonnement
* Réservation séances
* Statistiques d’évolution
* Suggestions d’entraînements (via ML)
* Formulaire de feedback

**🛠️ Côté Admin / Manager**

* CRUD utilisateurs, coachs, abonnements
* Dashboard en temps réel (revenus, fréquentation)
* Export vers Power BI
* Alertes sur les abonnés inactifs
* Prévision de la fréquentation / charge coachs

**🚀 DevOps & Pro Workflow**

* 🐳 **Docker** pour backend + ML
* 📦 GitHub + CI/CD avec GitHub Actions
* 🔒 Auth sécurisée (hash, JWT, roles)
* 📁 Dossier pro : /frontend, /backend, /ml-service, /powerbi-data

**📌 Livrables finaux**

* Application MERN complète déployée (Render, Vercel, Railway…)
* Dashboards Power BI professionnels
* Rapport PDF + présentation pour stage
* Code source GitHub bien documenté

**✅ Valeur ajoutée pour CV et entreprise**

* Compétences MERN full-stack
* Machine Learning appliqué à la vraie donnée
* Usage BI stratégique avec Power BI
* DevOps, microservices, sécurité
* Projet facilement adaptable pour toute salle de sport réelle !

\*

**🧱 PHASE 1 – Conception & Préparation (1–2 jours)**

✅ Objectif : Valider la logique, organiser le projet

**✅ Étapes :**

1. **Valider le périmètre du projet** (ce qu’on fait, ce qu’on ne fait pas)
2. **Finaliser les diagrammes** (use case, classe, séquence, base de données)
3. **Choisir les outils** :
   * Stack : MERN + ML microservice (Python/Flask) + Power BI
   * VS Code, GitHub, Docker, Postman
4. **Créer structure du projet** :
   * /frontend (React)
   * /backend (Express)
   * /ml-service (Python Flask)
   * /bi (Power BI data/export)
5. **Initialiser Git + README + Trello ou Notion (roadmap)**

**💻 PHASE 2 – Développement Backend (4–6 jours)**

✅ Objectif : Créer toutes les API REST et la base de données MongoDB

**✅ Étapes :**

1. **Setup Express.js + MongoDB + Auth JWT**
2. **Modèles MongoDB (User, Client, Coach, Session, Feedback, Payment, etc.)**
3. **Routes REST sécurisées :**
   * Auth: /register, /login
   * Client : /profile, /sessions, /feedback
   * Admin : /users, /sessions, /payments
4. **Middleware rôles : Admin, Coach, Client**
5. **Connexion à MongoDB Atlas ou local**
6. **Tests Postman complets**

**🎨 PHASE 3 – Frontend React (5–7 jours)**

✅ Objectif : Créer UI client + admin, dashboard interactif

**✅ Étapes :**

1. **Setup React + Tailwind + Router + Axios**
2. **Auth pages : Login, Register**
3. **Client Area :**
   * Home, Séances, Statistiques, Suggestions ML
   * Composant feedback
4. **Admin Area :**
   * Dashboard (recharts, tables, etc.)
   * Gestion abonnés / coachs / paiements
   * Export CSV/JSON vers BI
5. **Gestion état (Zustand ou Redux si nécessaire)**

**🧠 PHASE 4 – Module Machine Learning (4–6 jours)**

✅ Objectif : Recommandations, analyse churn, prévision fréquentation

**✅ Étapes :**

1. **Créer microservice Flask avec endpoints :**
   * /predict-churn
   * /recommend-session
2. **Modèles :**
   * KNN / KMeans pour recommandation
   * Random Forest / Logistic Regression pour churn
3. **Connexion ML <-> backend Express (via Axios)**
4. **Exporter CSV pour entraînement**
5. **Tests avec données mockées**
6. **Exposer résultats dans React UI**

**📊 PHASE 5 – BI avec Power BI (3–4 jours)**

✅ Objectif : Dashboard Power BI dynamique pour l’admin

**✅ Étapes :**

1. **Exporter données du backend (abonnements, feedback, paiements)**
2. **Créer dashboards sur Power BI Desktop**
3. **Rapports :**
   * Abonnés actifs/inactifs
   * Fréquentation hebdo/mensuelle
   * Revenu total prédit
   * Satisfaction par coach
4. **Documenter usage et images des dashboards**
5. **Possibilité de publier sur Power BI Cloud**

**🚀 PHASE 6 – Test, Sécurité, Déploiement (3–5 jours)**

✅ Objectif : Avoir une version propre, stable et présentable

**✅ Étapes :**

1. **Tests unitaires & manuels**
2. **Sécurisation API (limitation, validation, headers)**
3. **Déploiement :**
   * React (Vercel/Netlify)
   * Backend + ML (Render / Railway / Docker VPS)
   * Power BI en ligne (si possible)
4. **Version finale GitHub + Rapport + Présentation PowerPoint**

**📁 Livrables finaux**

* Site full-stack MERN + ML opérationnel
* API documentée
* Dashboards Power BI
* Rapport de stage complet (PDF)
* Démo live + GitHub link + poster (optionnel)

**🧩 Résumé des Tables MongoDB (GYM360)**

| **Nom du Modèle** | **Description / Utilité** |
| --- | --- |
| **User** | Base pour tous les utilisateurs : clients, coachs, admin (contient nom, email, mot de passe, rôle, etc.) |
| **Client** | Étend le modèle User pour les clients. Contient leurs objectifs fitness, et lien vers leur abonnement. |
| **Coach** | Étend User aussi. Contient la spécialité du coach et les clients qu’il suit. |
| **Session** | Représente une séance de sport entre coach et un ou plusieurs clients. (date, type, statut, etc.) |
| **Subscription** | Abonnement d’un client (mensuel/annuel) : dates, type, prix, statut actif/expiré |
| **Payment** | Paiement lié à un abonnement. Montant, date, statut payé ou en attente. |
| **Feedback** | Feedback du client sur un coach après une séance : note (1 à 5) et commentaire. |
| **MLRecommendation** | Résultat d’analyse ML : liste de séances recommandées + score de risque de churn (perte client). |
| **BIExportData** | Fichiers générés pour Power BI : sessions, abonnements, paiements — stockés avec date et chemin. |

**🔄 Relations principales (entre les tables)**

* Client.userId → 🔗 User.\_id
* Coach.userId → 🔗 User.\_id
* Client.subscriptionId → 🔗 Subscription.\_id
* Session.coachId → 🔗 Coach.\_id
* Session.clientIds[] → 🔗 Client.\_id
* Payment.clientId → 🔗 Client.\_id
* Payment.subscriptionId → 🔗 Subscription.\_id
* Feedback.clientId → 🔗 Client.\_id
* Feedback.coachId → 🔗 Coach.\_id
* Feedback.sessionId → 🔗 Session.\_id
* MLRecommendation.clientId → 🔗 Client.\_id