PROJET DE FIN D’ÉTUDES

Cycle d’Ingénieur – ~~Spécialité~~ Ingénierie des Systèmes Informatiques

Titre du Projet :

# MagicFit : Plateforme intelligente de coaching sportif et nutritionnel

Réalisé par : Yassine CHAHBOUNIA, Sara HJIRA

Encadrant académique : Pr. Mohammed LAZAAR

Encadrant en entreprise : [À remplir]

# Table des matières

1. Résumé

2. Introduction

3. Problématique

4. Objectifs

5. Étude de l’existant

6. Architecture du système

7. Modélisation des données

8. Développement Backend

9. Développement Frontend

10. Fonctionnalités clés

11. Sécurité

12. Tests et Validation

13. Résultats obtenus

14. Difficultés rencontrées

15. Perspectives

16. Conclusion

17. Annexes

# Résumé

MagicFit est une plateforme web de coaching sportif et nutritionnel. Elle propose des services intelligents tels que des programmes personnalisés, un suivi visuel des performances, un coach virtuel basé sur l’IA et une gestion de réservation. Ce projet vise à répondre aux nouveaux besoins des utilisateurs connectés en leur offrant une expérience riche, intuitive et motivante. Le rapport décrit les étapes de conception, développement, tests et déploiement du projet MagicFit, tout en exposant les choix technologiques justifiés.

MagicFit est une plateforme web de coaching sportif et nutritionnel. Elle propose des services intelligents tels que des programmes personnalisés, un suivi visuel des performances, un coach virtuel basé sur l’IA et une gestion de réservation. Ce projet vise à répondre aux nouveaux besoins des utilisateurs connectés en leur offrant une expérience riche, intuitive et motivante. Le rapport décrit les étapes de conception, développement, tests et déploiement du projet MagicFit, tout en exposant les choix technologiques justifiés.

# Introduction

La transformation numérique touche tous les secteurs, y compris celui du sport. Le coaching physique traditionnel est limité dans sa portée et son accessibilité. MagicFit ambitionne de digitaliser cette relation avec des outils modernes, interactifs et autonomes. Grâce aux frameworks Laravel et Angular, combinés à l'intelligence artificielle (OpenAI), nous avons conçu une plateforme complète, sécurisée et évolutive.

La transformation numérique touche tous les secteurs, y compris celui du sport. Le coaching physique traditionnel est limité dans sa portée et son accessibilité. MagicFit ambitionne de digitaliser cette relation avec des outils modernes, interactifs et autonomes. Grâce aux frameworks Laravel et Angular, combinés à l'intelligence artificielle (OpenAI), nous avons conçu une plateforme complète, sécurisée et évolutive.

# Problématique

L’objectif principal est de concevoir un système centralisé qui gère utilisateurs, programmes, exercices, suivis et réservations tout en restant simple d’utilisation. Les défis incluent :   
- l’authentification sécurisée,  
- la cohérence des données entre le frontend et backend,  
- l’intégration fluide d’un chatbot IA,  
- la limitation d’accès selon les abonnements.

L’objectif principal est de concevoir un système centralisé qui gère utilisateurs, programmes, exercices, suivis et réservations tout en restant simple d’utilisation. Les défis incluent :   
- l’authentification sécurisée,  
- la cohérence des données entre le frontend et backend,  
- l’intégration fluide d’un chatbot IA,  
- la limitation d’accès selon les abonnements.

# Objectifs

Les objectifs opérationnels sont :  
- Concevoir une architecture RESTful scalable  
- Gérer des rôles utilisateurs (coach/client)  
- Implémenter des dashboards personnalisés  
- Afficher dynamiquement les progrès (poids, date)  
- Développer une interface de réservation (groupée/privée)  
- Intégrer une IA pour guider l’utilisateur  
- Limiter les fonctionnalités selon les abonnements actifs

Les objectifs opérationnels sont :  
- Concevoir une architecture RESTful scalable  
- Gérer des rôles utilisateurs (coach/client ou admin)  
- Implémenter des dashboards personnalisés  
- Afficher dynamiquement les progrès (poids, date)  
- Développer une interface de réservation (groupée/privée)  
- Intégrer une IA pour guider l’utilisateur  
- Limiter les fonctionnalités selon les abonnements actifs

# Étude de l’existant

Des applications comme Freeletics ou FizzUp proposent du coaching en ligne, mais elles manquent souvent de personnalisation, de retour IA et de modularité. MagicFit surpasse ces limites grâce à une architecture découpée, une logique métier personnalisable et une IA prompte à s'adapter aux besoins des utilisateurs.

Des applications comme Freeletics ou FizzUp proposent du coaching en ligne, mais elles manquent souvent de personnalisation, de retour IA et de modularité. MagicFit surpasse ces limites grâce à une architecture découpée, une logique métier personnalisable et une IA prompte à s'adapter aux besoins des utilisateurs.

# Architecture du système

Le système est organisé en 3 couches :  
1. Frontend : Angular, composant SPA, services, modules, routing sécurisé  
2. Backend : Laravel API REST, Middleware JWT, contrôleurs et Eloquent ORM  
3. Services externes : OpenAI, Gmail SMTP, hébergement local/Docker

L’authentification utilise un token JWT transmis depuis Angular dans chaque requête API. Les communications IA sont gérées par OpenRouter pour garantir rapidité et flexibilité.

Le système est organisé en 3 couches :  
1. Frontend : Angular, composant SPA, services, modules, routing sécurisé  
2. Backend : Laravel API REST, Middleware JWT, contrôleurs et Eloquent ORM  
3. Services externes : OpenAI, Gmail SMTP, hébergement local/Docker

L’authentification utilise un token JWT transmis depuis Angular dans chaque requête API. Les communications IA sont gérées par OpenRouter pour garantir rapidité et flexibilité.

Le système est organisé en 3 couches :  
1. Frontend : Angular, composant SPA, services, modules, routing sécurisé  
2. Backend : Laravel API REST, Middleware JWT, contrôleurs et Eloquent ORM  
3. Services externes : OpenAI, Gmail SMTP, hébergement local/Docker

L’authentification utilise un token JWT transmis depuis Angular dans chaque requête API. Les communications IA sont gérées par OpenRouter pour garantir rapidité et flexibilité.

# Modélisation des données

Les entités :  
- Utilisateur : rôles, email, mot de passe haché  
- Programme : nom, description, durée, type (gratuit/premium)  
- Exercice : temps, difficulté, muscles ciblés  
- Suivi : date, poids, commentaire  
- Réservation : type, statut, email, confirmation  
- Abonnement : date de début, validité, type

Les relations sont en one-to-many et many-to-many selon les entités. Chaque modèle Laravel inclut les fonctions de validation, accès, et relation.

Les entités :  
- Utilisateur : rôles, email, mot de passe haché  
- Programme : nom, description, durée, type (gratuit/premium)  
- Exercice : temps, difficulté, muscles ciblés  
- Suivi : date, poids, commentaire  
- Réservation : type, statut, email, confirmation  
- Abonnement : date de début, validité, type

Les relations sont en one-to-many et many-to-many selon les entités. Chaque modèle Laravel inclut les fonctions de validation, accès, et relation.

# Développement Backend

Le backend Laravel repose sur des routes API REST :  
- CRUD complet (Programmes, Exercices, Réservations)  
- Authentification (register, login) avec JWT  
- Middleware pour restreindre l’accès à certaines routes  
- Notification par email après réservation  
- Intégration OpenRouter (proxy OpenAI)

Le backend Laravel repose sur des routes API REST :  
- CRUD complet (Programmes, Exercices, Réservations)  
- Authentification (register, login) avec JWT  
- Middleware pour restreindre l’accès à certaines routes  
- Notification par email après réservation  
- Intégration OpenRouter (proxy OpenAI)

Le backend Laravel repose sur des routes API REST :  
- CRUD complet (Programmes, Exercices, Réservations)  
- Authentification (register, login) avec JWT  
- Middleware pour restreindre l’accès à certaines routes  
- Notification par email après réservation  
- Intégration OpenRouter (proxy OpenAI)

# Développement Frontend

Le frontend Angular offre :  
- Composants dynamiques (listes, formulaires, dashboard)  
- Appels API sécurisés avec tokens JWT  
- Visualisation via Chart.js des suivis  
- Interface coach avec vue sur utilisateurs et programmes  
- Intégration chatbot IA avec formulaire + animation de chargement  
- Responsive design avec Bootstrap

Le frontend Angular offre :  
- Composants dynamiques (listes, formulaires, dashboard)  
- Appels API sécurisés avec tokens JWT  
- Visualisation via Chart.js des suivis  
- Interface coach avec vue sur utilisateurs et programmes  
- Intégration chatbot IA avec formulaire + animation de chargement  
- Responsive design avec Bootstrap

Le frontend Angular offre :  
- Composants dynamiques (listes, formulaires, dashboard)  
- Appels API sécurisés avec tokens JWT  
- Visualisation via Chart.js des suivis  
- Interface coach avec vue sur utilisateurs et programmes  
- Intégration chatbot IA avec formulaire + animation de chargement  
- Responsive design avec Bootstrap

# Fonctionnalités clés

- Authentification sécurisée avec JWT  
- Formulaires de suivi, filtre par date et poids  
- Réservations privées ou collectives avec statut  
- Assistant IA intelligent intégré via formulaire  
- Accès limité selon abonnement (gratuit/premium)  
- Tableaux de bord coach/admin avec filtres

- Authentification sécurisée avec JWT  
- Formulaires de suivi, filtre par date et poids  
- Réservations privées ou collectives avec statut  
- Assistant IA intelligent intégré via formulaire  
- Accès limité selon abonnement (gratuit/premium)  
- Tableaux de bord coach/admin avec filtres

# Sécurité

Le backend utilise :  
- Hachage des mots de passe (bcrypt)  
- Tokens JWT pour authentification  
- Middleware pour accès restreint  
- Validation côté serveur + client  
- Sécurisation des routes API avec Auth:api

Le backend utilise :  
- Hachage des mots de passe (bcrypt)  
- Tokens JWT pour authentification  
- Middleware pour accès restreint  
- Validation côté serveur + client  
- Sécurisation des routes API avec Auth:api

# Tests et Validation

Tests effectués :  
- Tests unitaires avec PHPUnit (controllers, models)  
- Tests manuels via Postman (CRUD, login)  
- Vérification de la conformité frontend (Angular console)  
- Tests sur les erreurs 422, formats de requêtes, tokens

Tests effectués :  
- Tests unitaires avec PHPUnit (controllers, models)  
- Tests manuels via Postman (CRUD, login)  
- Vérification de la conformité frontend (Angular console)  
- Tests sur les erreurs 422, formats de requêtes, tokens

# Résultats obtenus

- Application fonctionnelle avec tous les modules connectés  
- Réservation automatisée avec email  
- IA coach répondant aux demandes avec personnalisation  
- Statistiques disponibles dans le tableau de bord  
- Système de filtrage dynamique par date et poids

- Application fonctionnelle avec tous les modules connectés  
- Réservation automatisée avec email  
- IA coach répondant aux demandes avec personnalisation  
- Statistiques disponibles dans le tableau de bord  
- Système de filtrage dynamique par date et poids

# Difficultés rencontrées

- Intégration des tokens JWT dans Angular  
- Résolution des erreurs Laravel 422  
- Problèmes de migration des données avec perte évitée  
- Chargement dynamique des réponses IA en frontend

- Intégration des tokens JWT dans Angular  
- Résolution des erreurs Laravel 422  
- Problèmes de migration des données avec perte évitée  
- Chargement dynamique des réponses IA en frontend

# Perspectives

- Paiement en ligne pour débloquer les contenus premium  
- Application mobile (Flutter ou React Native)  
- Enregistrement vocal pour les suivis  
- Export PDF ou Excel des suivis  
- Interface Smart Mirror pour les exercices

- Paiement en ligne pour débloquer les contenus premium  
- Application mobile (Flutter ou React Native)  
- Enregistrement vocal pour les suivis  
- Export PDF ou Excel des suivis  
- Interface Smart Mirror pour les exercices

# Conclusion

MagicFit est un projet complet, structuré et aligné sur les nouvelles tendances technologiques. Il a permis d’explorer plusieurs axes : développement fullstack, sécurité, intégration API, UX/UI et productivité logicielle. Il constitue une base solide pour un produit commercialisable ou une plateforme en SaaS.

MagicFit est un projet complet, structuré et aligné sur les nouvelles tendances technologiques. Il a permis d’explorer plusieurs axes : développement fullstack, sécurité, intégration API, UX/UI et productivité logicielle. Il constitue une base solide pour un produit commercialisable ou une plateforme en SaaS.

# Annexes

- Diagramme de cas d'utilisation (UML)  
- Diagramme de classes  
- Captures d’écran (dashboard, réservation, IA)  
- Requêtes API types (GET, POST, PUT, DELETE)  
- Modèle relationnel de la base de données  
- Lien GitHub : https://github.com/ton-compte/MagicFit\_App

- Diagramme de cas d'utilisation (UML)  
- Diagramme de classes  
- Captures d’écran (dashboard, réservation, IA)  
- Requêtes API types (GET, POST, PUT, DELETE)  
- Modèle relationnel de la base de données  
- Lien GitHub : https://github.com/ton-compte/MagicFit\_App