Report submitted to the

UNIVERSITY OF MOHAMED BOUDIAF - M'SILA, ALGERIA

FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

in partial fulfillment of the requirements for the degree of

Master's in Computer Science

Specialization: Computer Science

Option: Networks and Information and Communication Technologies

Presented by

Benhennia Ziyad
Zougheleche Abderraouf

Entitled

PsyFit: Intelligent Platform for Personalized Training and Nutrition Using Artificial Intelligence and the Psycho-Fitness Model.

Under the supervision of

Dr. Lakehel Meftah

Jury Members

Dr. Chatra Mohamed	University of M'sila	President
Dr. Bahache Mohamed	University of M'sila	Examiner

June, 2026

الملخص

تقدم هذه المذكرة تصميم وتطوير وتنفيذ منصة **PsyFit**، وهي منصة ويب ذكية ومخصصة للياقة البدنية والتغذية تجمع بين التدريب البدني والرفاهية النفسية. على عكس المنصات التقليدية التي تركز حصرياً على المقاييس البدنية، تقدم PsyFit إطاراً للتقييم المزدوج يقيم الملف البدني للمستخدم (بما في ذلك مؤشر كتلة الجسم ومعدلات الأيض باستخدام قاعدة بيانات وزارة الزراعة الأمريكية (USDA) إلى جانب حالته النفسية (قياس مستويات التحفيز، الطاقة، القلق، والعادات السلوكية). تم تطوير الواجهة الأمامية باستخدام React.js وVite وTypeScript، بينما تم تشغيل الواجهة الخلفية باستخدام Node.js وExpress.js وقاعدة بيانات MySQL عبر Prisma ORM. بالإضافة إلى ذلك، تدمج المنصة نموذج الذكاء الاصطناعي Gemini 3.5 Flash من Google عبر Vercel AI SDK لتقديم دعم تفاعلي على مدار الساعة لإدارة التوتر والنوم والتوجيه الرياضي والنفسي. ورغم أن النموذج الأولي يثبت جدواه الفورية، فإن بناء نموذج للياقة النفسية والبدنية المتكامل بالكامل يمثل اتجاهاً للعمل المستقبلي نظراً لضيق وقت المشروع. في النهاية، تضع PsyFit أساساً قوياً لجيل جديد من منصات العافية الشاملة.

الكلمات المفتاحية: اللياقة البدنية الشخصية، تخطيط التغذية، التنميط النفسي، النماذج اللغوية الكبيرة، بيئة تطوير الويب.

Abstract

This Master's thesis presents the design, development, and implementation of **PsyFit**, an intelligent, personalized fitness and nutrition web platform that bridges physical training and psychological well-being. Conventional platforms focus exclusively on physical metrics, whereas PsyFit introduces a dual-assessment framework evaluating physical profiles (including Body Mass Index and metabolic parameters using the USDA Food Data Central database) alongside psychological states (motivation, energy, stress, and behavioral habits). The system utilizes a modern development stack consisting of React.js, Vite, and TypeScript on the frontend, and Node.js, Express.js, and MySQL (via Prisma ORM) on the backend. Furthermore, the platform integrates Google's Gemini 3.5 Flash LLM via the Vercel AI SDK to provide continuous, empathetic wellness chatbot support. While the prototype demonstrates immediate feasibility, constructing a fully automated psycho-fitness model remains a direction for future work due to project time constraints. Ultimately, PsyFit establishes a robust foundation for holistic, next-generation wellness platforms.

Keywords: Personalized Fitness, Nutrition Planning, Psychological Profiling, Large Language Models, Web Development Stack.

Dedication

إلى أبي العزيز زوغلاش سليم، مصدر دعمي وقوتي،

إلى أمي الغالية عليوة حسينة، نبع الحنان والدعاء،

إلى إخوتي وأخواتي، سندي ورفقاء دربي،

إلى أستاذي المشرف الأستاذ الدكتور لكحل مفتاح، على دعمه وتوجيهاته القيمة،

إلى كل من مدّ لي يد العون والمساعدة،

وإلى كل من عرفني من قريب أو بعيد، وإن لم تتسع صفحات هذا العمل لذكرهم، فإن مكانهم محفوظ في قلبي،

أهديكم هذا العمل المتواضع، عرفانًا وامتنانًا.

Zougheleche Abderraouf

Dedication

إلى من كانا نعم السند والعون، من تعبت يداهما لترتفع أيامي: والدي الغالي بن هنية محمد، وإلى من لا تكفيني الدنيا لأوفّيها حقها: والدتي الحبيبة ولد تومي الربح، إلى من علّماني أن النجاح يبدأ بالإخلاص وينتهي بالدعاء.

إلى إخوتي وأخواتي، وقبلهم إلى من كانت أكثر من أخت، سندًا وحضنًا: أختي العزيزة بن هنية خديجة.

إلى من علمني كيف تكون الكلمة مفتاحًا، والمعرفة نورًا: أستاذي المشرف الفاضل الدكتور لكحل مفتاح، أتقدم له بجزيل الشكر والتقدير على ما بذله من جهد وتوجيه.

وإلى رفاق الدرب، أصدقاء السنين والأيام، وإلى كل من عرفني فأحبني، أو رآني من بعيد فدعا لي بالخير.

أهديكم هذا العمل المتواضع، ثمرة جهد وسهر، وقطفًا لسنوات من الكفاح. لولا أنتم، ما كانت خطواتي يومًا بهذا الثبات.

شكرًا لكم، فهذا النجاح لكم قبل أن يكون لي.

Benhennia Ziyad

Acknowledgements

First and foremost, we express our profound gratitude to Allah for His mercy and guidance throughout this journey.

We would like to extend our heartfelt appreciation to our supervisor, Dr. Lakehel Meftah, for his invaluable guidance, support, and patience. His expertise and encouragement have been instrumental in shaping this research and our academic growth.

We would also like to thank all the jury's members for accepting to review and judge this project, namely, Dr. Chatra Mohamed and Dr. Bahache Mohamed. The final version of this report has benefited from their careful reading and valuable comments.

We are also deeply thankful to the teachers from our faculty of mathematics and computer science for providing us with the necessary knowledge and skills to succeed in our studies.

Lastly, we offer our regards to everyone who has stood by us and supported us during this journey.

Contents

List of Tables

List of Figures

General Introduction

The integration of artificial intelligence into health and fitness applications has revolutionized personalized training and nutrition. Modern users expect tailored fitness solutions that account for both their physical capabilities and their psychological states. Consequently, contemporary platforms must evaluate not only physiological metrics but also psychological determinants, such as motivation, stress levels, and social preferences (e.g., individual versus group training). Applications that ignore these psychological and social factors often suffer from high user attrition, as individuals are significantly more likely to adhere to exercise regimens that align with their mental well-being and personal preferences. In this context, artificial intelligence serves as a critical enabler of holistic personalization, delivering responsive interventions that respect and adapt to the user's comprehensive needs.

PsyFit represents an innovative platform designed to address this gap by creating customized fitness programs through a comprehensive assessment of users' psychological and physical profiles. Developed using React for the frontend and Express.js with MySQL for the backend [1], PsyFit collects user data during onboarding, calculates Body Mass Index (BMI), and generates personalized workout programs based on training frequency and user level (beginner, intermediate, or advanced).

Motivation and Problem Statement

Contemporary fitness applications suffer from several critical limitations. First, most platforms rely exclusively on physical metrics such as age, weight, and height, while neglecting psychological determinants of exercise adherence. Second, nutritional recommendations are typically derived from population-average formulas rather than individual metabolic profiles. Third, existing systems lack real-time mood monitoring and adaptive support mechanisms, resulting in retention rates below 20% after 30 days. These limitations underscore the need for an integrated platform that bridges the gap between physical training and psychological well-being.

Objectives

The primary objectives of this work are as follows:

1. Enable user assessment for psycho-physical profiling and automated BMI computation.

2. Generate rule-based personalized workout programs and nutrition plans with detailed macronutrient breakdowns (calories, proteins, carbohydrates, and fats).

3. Provide a mood tracker for continuous psychological state monitoring, weekly reports, profile management, and an external API-powered psychology chatbot.

Scope and Limitations

The current implementation includes a home page, program viewer, nutrition module, mood tracker, user profile management, and an AI-powered chatbot. The weekly report feature remains under development. The platform is designed as a web-based proof-of-concept; mobile applications and payment integration are not yet implemented.

Methodology

The development follows an incremental approach:

- User data collection through structured onboarding forms
- Automated BMI calculation based on CDC guidelines
- Rule-based generation of personalized fitness and nutrition programs
- Persistent storage of user profiles and progress in MySQL database.
- Real-time mood updates with interactive visualizations

Report Structure

This report is organized into three chapters:

- **Chapter 1** presents the theoretical framework, examining intelligent fitness systems, AI-driven personalization, and the psycho-fitness model underlying PsyFit.
- **Chapter 2** details the system design, including the C4 architectural model, Unified Modeling Language (UML) analysis, and the integration of artificial intelligence within the platform.
- **Chapter 3** discusses implementation and deployment, presenting all PsyFit pages and the technologies utilized.

Finally, the report concludes with an analysis of results and directions for future work.

Chapter 1: Intelligent Platforms for Personalized Training and Nutrition

1 Introduction

The convergence of artificial intelligence, wearable sensors, and mobile health technologies has catalyzed a paradigm shift in how individuals manage their fitness and nutrition. Contemporary platforms no longer serve merely as digital logs; they function as intelligent coaches capable of delivering personalized recommendations, tracking physiological and psychological states, and adapting interventions in real time. This chapter establishes the theoretical foundations underlying PsyFit, beginning with a taxonomy of personalized fitness and nutrition platforms, followed by an examination of their evolutionary trajectory. Subsequently, the conceptual framework of PsyFit is introduced, emphasizing the dual importance of physiological and psychological assessments. The chapter then explores the role of artificial intelligence in enabling genuine personalization, with particular attention to large language models, retrieval-augmented generation, and authoritative nutritional knowledge bases. Finally, key challenges in existing systems are synthesized, and a comparative overview of established platforms such as Noom and Simple is provided.

2 Personalized Nutrition and Fitness Platforms

2.1 Definition of a Personalized Fitness and Nutrition Platform

A Personalized Fitness and Nutrition Coach is an AI-driven platform designed to empower users in achieving their health and wellness goals through tailored fitness plans and dietary recommendations. The system collects user data such as age, weight, height, activity level, workout preferences, and dietary requirements to create a structured and customized fitness routine.[2]

2.2 Types of Personalized Fitness and Nutrition Platforms

Personalized fitness and nutrition platforms can be classified into three primary domain focuses: fitness-only, nutrition-only, or integrated.

Fitness-only platforms

are exemplified by applications such as Peloton[3] and Strava [4]. These platforms concentrate exclusively on exercise programming, progress tracking, and performance analytics, offering features such as workout logging, route mapping, and comparative metrics against peer groups.

Figure 1.1: Peloton Platform [3]

Figure 1.2: Strava Platform [4]

Nutrition-only platforms

such as Cronometer[5] and Nutritionix[6] focus primarily on dietary intake tracking, macronutrient analysis, and food logging. These systems typically maintain extensive food databases and provide detailed micronutrient breakdowns.

Figure 1.3: Cronometer Platform [5]

Figure 1.4: Nutritionix Platform [6]

Integrated platforms

such as MyFitnessPal[7] and Apple Health[8] attempt to span both domains within a unified system, though with varying levels of sophistication in each domain. These platforms aggregate data from multiple sources, including manual user entry and synchronized wearable devices.

Figure 1.5: MyFitnessPal Platform [7]

Figure 1.6: Apple Health Platform [8]

2.3 Evolution of Personalized Fitness and Nutrition Systems

The history of personalized fitness and nutrition systems spans roughly four generations, each characterized by distinct technological capabilities and limitations:

First Generation (1980s - early 2000s) - Paper-Based Era

At this stage, programs were entirely offline, relying on printed workout plans and consultations with trainers or dietitians.

Characteristics:

- Static, non-adaptive programs (no update over time).
- Personalization depended only on the expert (trainer/dietitian).
- No data tracking or automation.
- Limited scalability and accessibility.

Second Generation (early 2000s) — Early Digital Platforms:

With the rise of the internet, fitness tools became digital, allowing users to log workouts and meals online.

Characteristics:

- Digital tracking of fitness and nutrition.
- Based on general guidelines (not truly personalized).
- Predefined programs with little flexibility.
- Improved convenience but limited intelligence.

Third Generation (2010s) — Mobile & Smart Tracking Era:

The spread of smartphones enabled mobile apps with better tracking and basic personalization features.

Characteristics:

- Mobile apps with real-time tracking (activity, calories).
- Use of simple algorithms and rule-based recommendations.
- Integration of social features and gamification.
- Large-scale data collection from users.
- Personalization still limited (not fully intelligent).

Fourth Generation (2018 – present) — AI-Driven Intelligent Systems:

Modern platforms use advanced AI and data analytics to deliver highly personalized and adaptive recommendations.

Characteristics:

- Use of machine learning, deep learning, and AI models.
- Real-time, dynamic personalization based on user data.
- Natural language interaction (chatbots, assistants).
- Integration of psychology and behavior science.
- Highly adaptive and data-driven decision making.

3 What is PsyFit?

3.1 Definition and Conceptual Framework

PsyFit is an intelligent web platform that personalizes fitness training and nutrition plans for each user by combining two assessments at registration: a physical (physiological) assessment measuring weight, height, fitness level, and goals; and a psychological assessment measuring stress, mood, motivation, and anxiety. The platform employs rule-based artificial intelligence to generate a weekly workout program and a nutrition plan adapted to both the body and the mental state of the user. Additionally, PsyFit includes a mood tracker, an AI chatbot for fitness and psychological support, and weekly well-being reports.

3.2 Physiological Assessment and BMI

Personalized fitness systems require an initial physiological assessment to classify users and estimate appropriate levels of effort[9] . One of the most widely used indicators for this purpose is Body Mass Index (BMI), which compares body weight to height and provides standardized categories for adult weight status[10]. According to the Centers for Disease Control and Prevention (CDC), adult BMI categories are defined as underweight for values below 18.5, healthy weight from 18.5 to less than 25, overweight from 25 to less than 30, and obesity for values of 30 or higher[10] .

BMI is useful because it is simple, fast, and easy to calculate in digital environments[10]. In fitness applications, it can be used during onboarding to provide an initial estimate of physical profile and to orient basic recommendations related to training intensity or weight-related goals. However, BMI has recognized limitations. The CDC explicitly states that BMI is a screening tool rather than a direct diagnostic measure of body fatness or overall health. Two individuals can have identical BMI values while having substantially different body composition, physical capacity, or health status. For this reason, BMI should not be used alone. In a well-designed fitness platform, it should be interpreted alongside age, physical activity level, training history, health constraints, and behavioral factors.

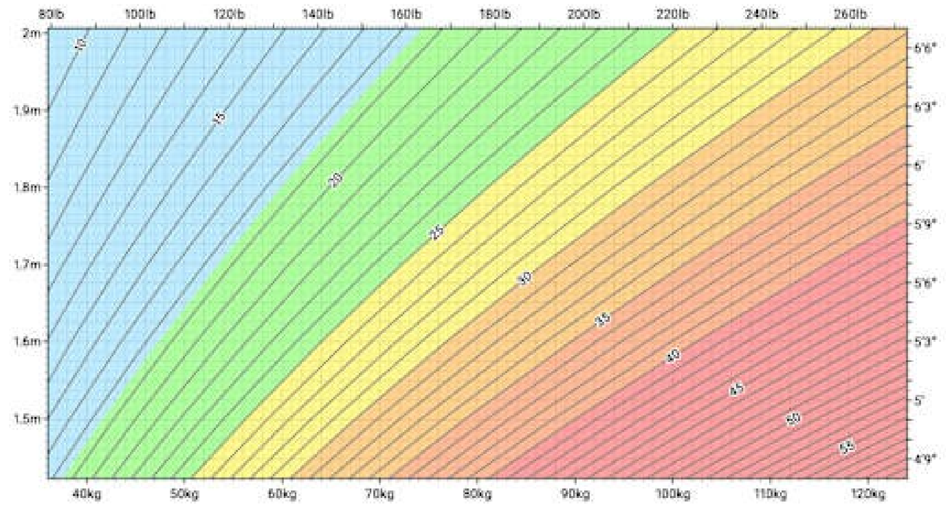

Figure 1.7: BMI Categories Chart [9]

However, BMI has limits that must be recognized. The CDC clearly states that BMI is a screening tool rather than a direct diagnostic measure of body fatness or overall health[11]. Two people can have the same BMI while having very different body composition, physical capacity, or health status.

For that reason, BMI should not be used alone. In a well-designed fitness platform, it should be interpreted alongside age, physical activity level, training history, health constraints, and possibly behavioral factors[9]. This broader interpretation is more consistent with modern approaches to personalized digital fitness

3.3 Psychological Factors and Exercise Adherence

Long-term exercise participation depends heavily on psychological and social conditions[12]. A systematic review of adherence to exercise referral schemes found that psychosocial factors such as motivation, self-efficacy, and social support are strongly related to whether users continue participating over time[12] . This finding implies that an exercise system cannot be considered truly personalized if it measures only the body while ignoring the mind.

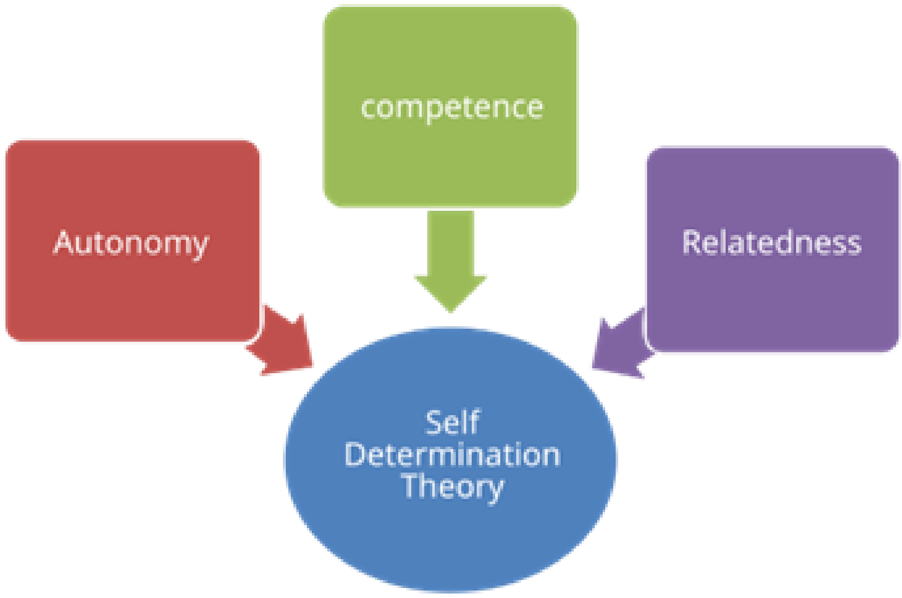

Figure 1.8: Self-Determination Theory Framework [10]

One of the most influential theories in this area is Self-Determination Theory (SDT)[13] Research applying SDT to physical exercise adherence found that more self-determined forms of motivation are positively associated with persistence[13], while less self-determined forms are negatively associated with persistence. In practical terms, users are more likely to continue exercising when they perceive the activity as personally meaningful and internally supported rather than externally imposed.

The same study demonstrated that autonomy support influences the satisfaction of basic psychological needs, which in turn are related to autonomous regulation, helping explain exercise adherence[13]. These findings are especially important for digital platforms because they suggest that the user experience itself can either support or weaken motivation.

For PsyFit, this implies that personalization must include psychological profiling alongside physical classification.

4 Artificial Intelligence in Fitness and Nutrition Systems

4.1 The Role of AI in Enabling Genuine Personalization

Traditional fitness and nutrition applications typically offer generalized advice based on static rules (e.g., BMI category or age group). However, such approaches fail to account for individual differences in physiology, behavior, lifestyle, and real-time contextual factors. Artificial intelligence (AI) addresses this limitation by enabling genuine personalization—dynamic adaptation of recommendations based on continuous learning from user data.

AI-driven personalization operates through three core mechanisms[14][15].

1. **Multi-modal data fusion:** Integrating heterogeneous data sources, including demographic information, dietary logs, physical activity records, sleep patterns, heart rate variability, and genetic markers (where available). Wearable devices (e.g., smartwatches, fitness trackers) provide continuous physiological signals that enrich user profiles.

2. **Predictive modeling:** Machine learning models, such as deep neural networks and random forests, predict user-specific outcomes (e.g., weight change, muscle gain, nutritional deficiencies). For instance, Chua [14] demonstrated that a U-Net-based convolutional neural network achieved 90.36% accuracy in BMI classification, outperforming simpler architectures. Ponte-Isminio et al.[15] reported meal recommendation accuracy of 92.38% and exercise model AUC-ROC of 94.90%.

3. **Closed-loop adaptation:** The system observes user adherence and physiological responses, then updates its recommendations iteratively. This feedback loop ensures that personalization improves over time, accommodating weight loss, strength gains, injury recovery, or changing preferences.

Retrieval-Augmented Generation (RAG) has emerged as a key architecture for personalization in language-based health interfaces. RAG combines a large language model with a vector database of user-specific information, retrieving relevant context (past goals, allergies, exercise history) at inference time without requiring model fine-tuning [16]. This approach balances personalization with computational efficiency and data privacy.

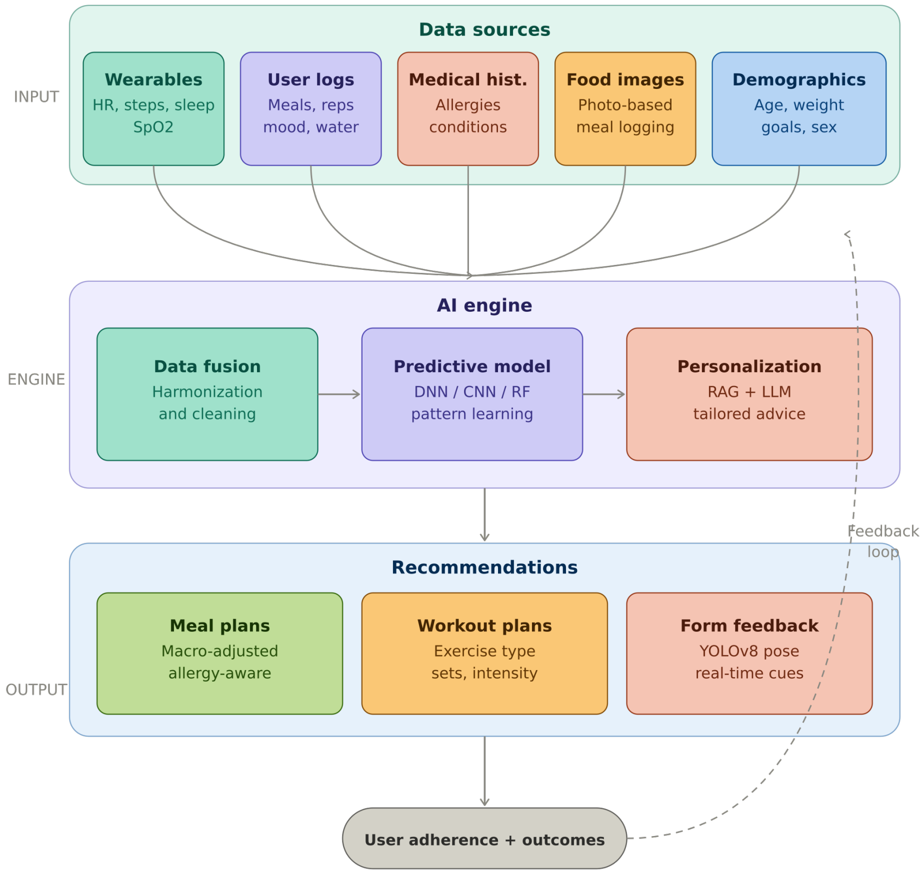

Figure 1.9: AI-Enabled Health System Architecture

4.2 Large Language Models and AI Chatbots in Health Platforms

While Section 4.1 focused on algorithmic personalization, this section addresses the interaction layer—how users engage with AI-driven health systems. Large language models (LLMs) and AI chatbots have transformed health platforms from static interfaces into conversational, intuitive, and accessible health companions[17].

Evolution from Rule-Based to Generative Chatbots

Early health chatbots relied on finite-state machines or pattern-matching (e.g., AIML-based systems), which could only respond to predefined queries with scripted answers. These systems lacked contextual understanding and could not handle open-ended user inputs. Modern generative chatbots, powered by transformer-based LLMs (e.g., GPT-4, Llama, Biomistral), engage in natural, multi-turn dialogue and generate contextually appropriate responses in real time[18].

Key Technical Architectures

Retrieval-Augmented Generation (RAG) is the predominant architecture for health chatbots. A RAG system first retrieves relevant information from a curated knowledge base (clinical guidelines, nutrition databases, medical literature) and then conditions the LLM's response on this retrieved context, significantly reducing hallucination[19].

Fine-tuned domain-specific models offer an alternative. MedBot, an LLM-powered assistant for heart disease management, employs Biomistral—a Mistral variant fine-tuned on biomedical literature—achieving high accuracy with 2–3 second latency[17].

Persistent memory layers enable chatbots to retain long-term context across sessions. Healthify's Ria assistant maintains user preferences, health changes, and progress over time, functioning as a longitudinal health partner rather than a stateless query processor[16].

Multimodal Interaction for Reduced Friction

User engagement is inversely correlated with logging friction. AI chatbots reduce friction through:

- **Voice-based interaction:** Users describe meals or workouts conversationally. Healthify's Ria supports live voice in over 50 languages, including mixed-language inputs[16].
- **Vision-based food logging:** Computer vision models (e.g., YOLOv8) estimate calories and macronutrients from meal photographs[19]. Meal Roaster enables food logging via WhatsApp images with sub-second response times[20].
- **Platform integration:** Chatbots operating within existing messaging platforms (WhatsApp, Telegram) eliminate app download barriers. Meal Roaster, for example, transforms WhatsApp into a complete nutrition assistant [20].

Safety and Clinical Considerations

Deploying LLMs in health contexts requires rigorous safety measures. Grounding in authoritative sources ensures responses are traceable to verified clinical data, as demonstrated by the MS chatbot using domain-specific medical RAG[18]. Scope limitation is essential—AI complements rather than replaces medical professionals; MedBot explicitly avoids diagnosis or treatment recommendations[17]. Finally, explicit disclaimers inform users of AI limitations, a practice adopted by all reviewed systems.

4.3 The USDA Food Data Central as a Nutritional Knowledge Base

A foundational requirement for any AI-driven nutrition system is access to a comprehensive, authoritative, and machine-readable repository of food composition data. The United States

Department of Agriculture (USDA) Food Data Central (FDC) serves as the premier open-access knowledge base in this domain[21]. Unlike aggregated online sources that may contain inconsistent or unverified information, FDC provides standardized, laboratory-validated nutritional profiles for over 450,000 food items, including macronutrients, micronutrients, vitamins, minerals, and portion-specific measurements across multiple serving sizes. The database is programmatically accessible via a Representational State Transfer (REST) Application Programming Interface (API), enabling real-time queries for AI fitness and nutrition systems. For example, a chatbot processing a user's meal description ("I ate 200g of grilled chicken breast with steamed broccoli") can retrieve precise nutritional values from FDC, calculate total caloric and macronutrient intake, and compare it against daily personalized targets. Furthermore, FDC supports complex queries such as filtering by food category, nutrient range, or data source (e.g., Standard Reference, Experimental, or Branded Foods). By grounding recommendations in USDA-verified data, AI systems mitigate the risk of hallucination or inaccurate nutritional counsel, thereby enhancing both trust and clinical safety.

5 Key Challenges in Existing Systems and Proposed Solutions

Challenge	**Impact on Users and Outcomes**	**Solution Implemented in PsyFit**
Absence of psychological assessment in recommendation engines	Recommendations are physically generic, fail to account for stress, mood, or motivation; high dropout rates among psychologically vulnerable users	Full dual-assessment pipeline: structured psychological and physical intake, both integrated into recommendation engine
Generic, population-average nutritional recommendations	Caloric and macronutrient targets derived from static formulas, not individual metabolic profiles; poor dietary adherence	Harris-Benedict BMR engine + USDA FoodData Central nutritional data; individual caloric surplus/deficit/maintenance
No real-time mood or psychological state monitoring	Platforms cannot detect deteriorating user well-being; no mechanism to prevent motivationally-driven dropout	Continuous Mood Tracker module with trend analysis; AI chat interface provides real-time responsive support
Low long-term user retention and engagement	Average fitness app retention drops below 20% after 30 days; revenue and health impact severely limited	Weekly AI-generated holistic reports, mood-integrated progress tracking, gamified goal setting, and AI chat support
Privacy concerns around personal health data	Users reluctant to share sensitive health and psychological data; mistrust reduces depth of engagement	Secure authentication, isolated user data architecture, explicit consent flows, and transparent data use policy
Lack of AI-powered psychological and fitness support chat	Users facing motivational blocks, emotional difficulties, or specific questions have no immediate responsive support channel	Integrated LLM-based AI chat module providing 24/7 fitness, nutritional, and psychological support
Nutrition databases of uncertain quality	Caloric estimates based on crowd-sourced or manufacturer-reported data; nutritional inaccuracies undermine dietary plan integrity	USDA FoodData Central Foundation Foods dataset: peer-reviewed, laboratory-validated food composition data
Static workout plans that do not adapt to psychological state	A high-intensity program assigned to a severely stressed user is counter-productive and risks injury or burnout	Fitness program recommendation engine integrates psychological assessment scores to modulate exercise type, intensity, and volume

Table 1.1: Challenges, Impacts, and Proposed Solutions

6 Existing Intelligent Fitness and Nutrition Platforms: A Comparative Overview

6.1 Noom

Noom is a digital health platform that integrates cognitive-behavioral therapy (CBT), acceptance and commitment therapy (ACT), and dialectical behavior therapy (DBT) to promote sustainable weight loss and lifestyle modification. Rather than prescribing dietary restrictions, Noom addresses the psychological determinants of eating behavior identifying cognitive patterns, emotional triggers, and motivational drivers through daily psychoeducational lessons, personalized coaching, and self-monitoring tools. The platform operationalizes behavior change theory by targeting long-term habit formation rather than short-term caloric restriction, positioning it within the field of digital behavioral medicine[22].

Main advantages

- Based on proven psychological methods that help users understand why they eat the way they do.
- Focuses on building lasting healthy habits instead of quick weight loss.
- Daily lessons are personalized to each user's goals and lifestyle.
- Provides access to real human coaches with proper health credentials.
- Officially recognized as a certified diabetes prevention program by the CDC.
- Uses a simple green-yellow-red food labeling system that makes healthy choices easy.

Possible limitations

- Daily logging and lessons take time, so users who lack motivation may quit early.
- The food color system might make some users feel that foods are "good" or "bad," which can be unhealthy for people with a history of eating disorders.
- Does not track vitamins or minerals, only calories.
- Expensive subscription (over $199 billed upfront), which is not affordable for everyone.
- Requires constant use of the app, so it does not work well for people who prefer in-person support.

- Focusing too much on weight numbers may reduce feelings of body positivity and self-compassion.

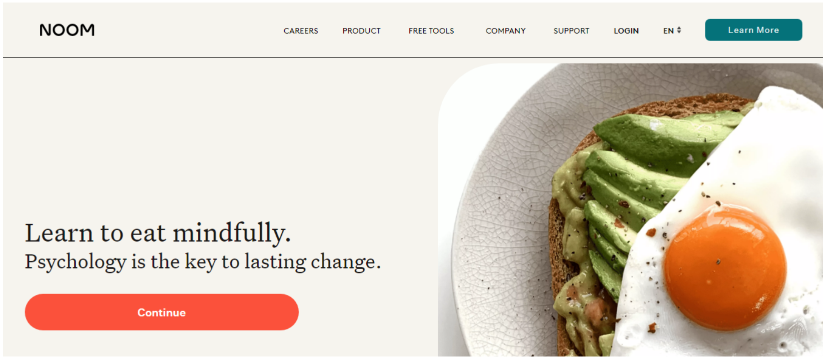

Figure 1.10: Noom Platform [22]

6.2 Simple

Simple is a science-based mobile wellness platform designed to support weight management through intermittent fasting, nutritional awareness, and behavioral habit formation. Launched in 2019 as a fasting tracker and expanded in 2023 into a full nutrition application, Simple employs a proprietary Nutrition Score system and an AI-powered coaching agent to guide users toward sustainable dietary routines without the cognitive burden of calorie counting. Positioned within the domain of mHealth (mobile health) interventions, Simple operationalizes time-restricted eating protocols a chrono-nutritional approach alongside food logging, hydration tracking, and movement monitoring to foster holistic lifestyle regulation[23].

Main advantages

- No calorie counting required, users simply follow eating and fasting time windows.
- AI coach gives instant, personalized tips on meals, exercise, and fasting schedules.
- The Nutrition Score system teaches users about food quality without making them feel restricted.
- The platform tracks everything together, including fasting, water intake, movement, and sleep.
- It is easy for beginners with flexible fasting plans that fit different daily routines.
- A large one-year study proved that using this app for intermittent fasting leads to real, sustainable weight loss.

Possible limitations

- The platform focuses mainly on when to eat rather than what to eat, so users might overeat or miss nutrients during eating windows.
- It does not track vitamins or minerals in detail, making it less than ideal for users with specific medical or dietary needs.
- No real human coach is available—only AI support—which may not be sufficient for users needing emotional or medical guidance.
- Intermittent fasting is not safe for pregnant women, individuals with diabetes, or those with a history of eating disorders.
- The best features require a paid subscription, so the free version is very limited.
- The amount of information in the app can feel overwhelming for beginners who have not yet established a routine.

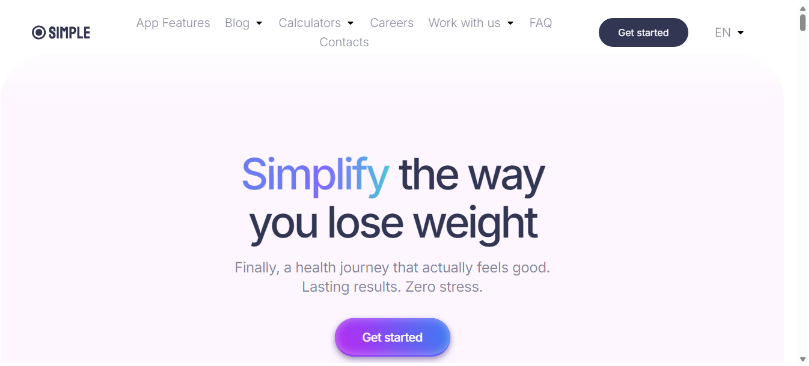

Figure 1.11: Simple Platform [23]

7 Conclusion

This chapter has established the theoretical foundations underlying intelligent platforms for personalized training and nutrition. The evolution from paper-based programs to AI-driven systems has been traced, revealing a consistent trajectory toward greater personalization, real-time adaptation, and psychological integration. The conceptual framework of PsyFit was introduced, emphasizing the dual necessity of physiological and psychological assessments. The role of artificial intelligence in enabling genuine personalization was examined, with particular attention to predictive modeling, retrieval-augmented generation, and authoritative nutritional knowledge bases such as the USDA Food Data Central. Key challenges confronting existing systems were synthesized, and platform-specific solutions were proposed. Finally, a comparative overview of established platforms—Noom and Simple—provided practical context for the design decisions underlying PsyFit. The next chapter builds upon these foundations by presenting the system architecture, including C4 models, and Unified Modeling Language analyses.

Chapter 2: Platform Design

1 Introduction

In this chapter, we present a comprehensive overview of the design and architecture of the PsyFit platform. We begin by introducing the C4 model, which provides a structured way to visualize the system's architecture at different levels of abstraction. We then delve into the specific components of the backend server, detailing how they interact with each other and with the database. Following this, we analyze the system's behavior using Use Case diagrams to illustrate how different types of users interact with the platform. We also utilize Sequence diagrams to show the flow of data and interactions between components during key operations and we present Class diagrams to depict the static structure of the system, including its main entities and their relationships. Finally, we discuss the integration of artificial intelligence into PsyFit through the use of the Gemini API, which allows us to provide personalized recommendations and support to users based on their psychological and physical assessments. This chapter serves as a detailed blueprint for the implementation of the PsyFit platform, ensuring that all components are well-defined and that their interactions are clearly understood.

2 C4 model overview

To communicate the architecture of PsyFit in a clear and structured manner, we adopted the C4 model proposed by Brown[24]. The C4 model stands for Context, Containers, Components, and Code. It provides a hierarchical set of diagrams that allow different stakeholders to understand the system from their respective perspectives. For this Master's thesis, we focus on the first two levels: System Context and Containers. These levels are sufficient to describe the high-level structure of the platform and the major technologies involved without diving into implementation details that might change during development. The decision to use the C4 model stems from its practical advantages over traditional UML diagrams for architectural documentation. Unlike UML component diagrams, which can become cluttered and overly detailed, the C4 model maintains a clean separation between different levels of granularity. Each level adds detail progressively, making the architecture easier to understand for both technical and non-technical audiences.

2.1 System Context Diagram :

A context diagram represents the highest level of system abstraction within the C4 model. This diagram illustrates the relationships between the central PsyFit system and the external users or systems that interact with it. The primary purpose is to provide a big-picture understanding of the system's environment without diving into technical details.

- **PsyFit System:** The central software system that provides personalized psycho-fitness services. It integrates psychological assessment with physical training and nutrition planning to deliver holistic wellness recommendations.
- **Visitor:** An individual who has not yet created an account or logged into the platform. Visitors can browse the landing page to learn about the platform's features, read about the psycho-fitness approach, and access the authentication page. They cannot access any personalized features, submit assessments, or view plans until they register. The visitor's journey typically ends either by leaving the site or by deciding to create an account.
- **User (Authenticated):** A registered individual who has successfully logged into the platform. Once authenticated, the user can complete psychological and physical assessments, receive a personalized plan generated by the system, view daily workouts and meal plans, track their mood over time, interact with the AI wellness assistant through chat, generate weekly progress reports, and update their profile information. The user represents the primary beneficiary of the PsyFit platform.

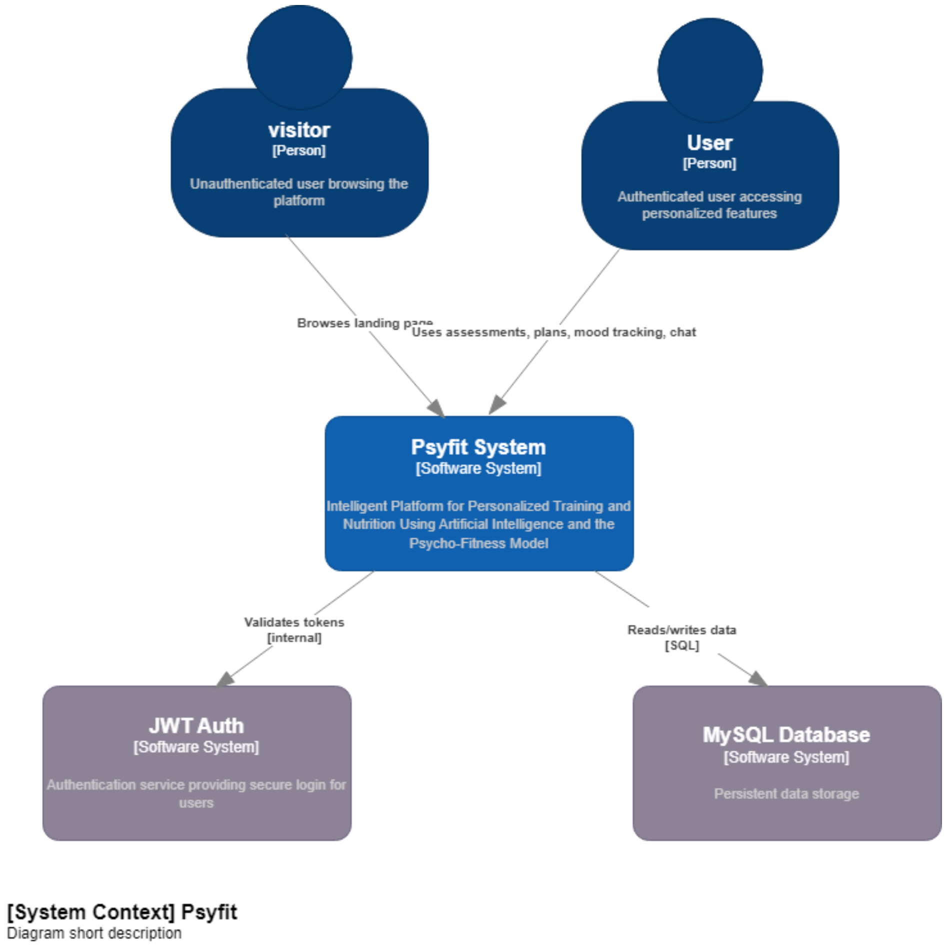

Figure 2.1: C4 Model - System Context Diagram

2.2 Container Diagram :

After defining the overall context of the system, we moved on to designing the Container Diagram, which shows how to divide the system into independent logical units, such as the backend server, user interface, and database. Through this scheme, we aimed to clarify the distribution of technical responsibilities between these containers, as well as to define the methods of communication and interaction between them using REST APIs.

Basic Components

Landing Page Application

- Developed with React and TypeScript.
- Represents the public-facing marketing interface with which visitors interact.
- Includes hero section, features, how-it-works steps, and contact form.
- Connects to the backend server via REST API to fetch public information only.

Dashboard Application

- Developed with React, TypeScript, and Tailwind CSS.
- Represents the protected user interface with which authenticated users interact.
- Provides access to assessments, personalized plans, mood tracking, chatbot, and reports.
- Connects to the backend server via REST API using JWT bearer tokens for authentication.

REST API Server

- Developed with Express.js and Node.js.
- Processes requests, implements business logic (BMI, BMR, TDEE calculations), verifies permissions, and generates JWT tokens.
- Manages database interactions using Prisma Client ORM.
- Automatically generates personalized plans when physical assessment is submitted.

Database (MySQL)

- A single relational database managed through Prisma Client ORM.
- Contains all persistent data including user profiles, psychological and physical assessments, generated plans, workout programs, exercises, nutrition plans, meals, mood entries, chat sessions and messages, and weekly reports.

JWT Authentication

- Implemented internally using jsonwebtoken and bcrypt.
- Handles user login, token generation, and token validation.
- Secures all protected API endpoints.

Interaction Components

- The frontend applications send requests to the server via HTTP.
- The server interacts with the database using SQL through Prisma Client ORM.

- JWT tokens are used to secure interaction between the dashboard application and the API server.

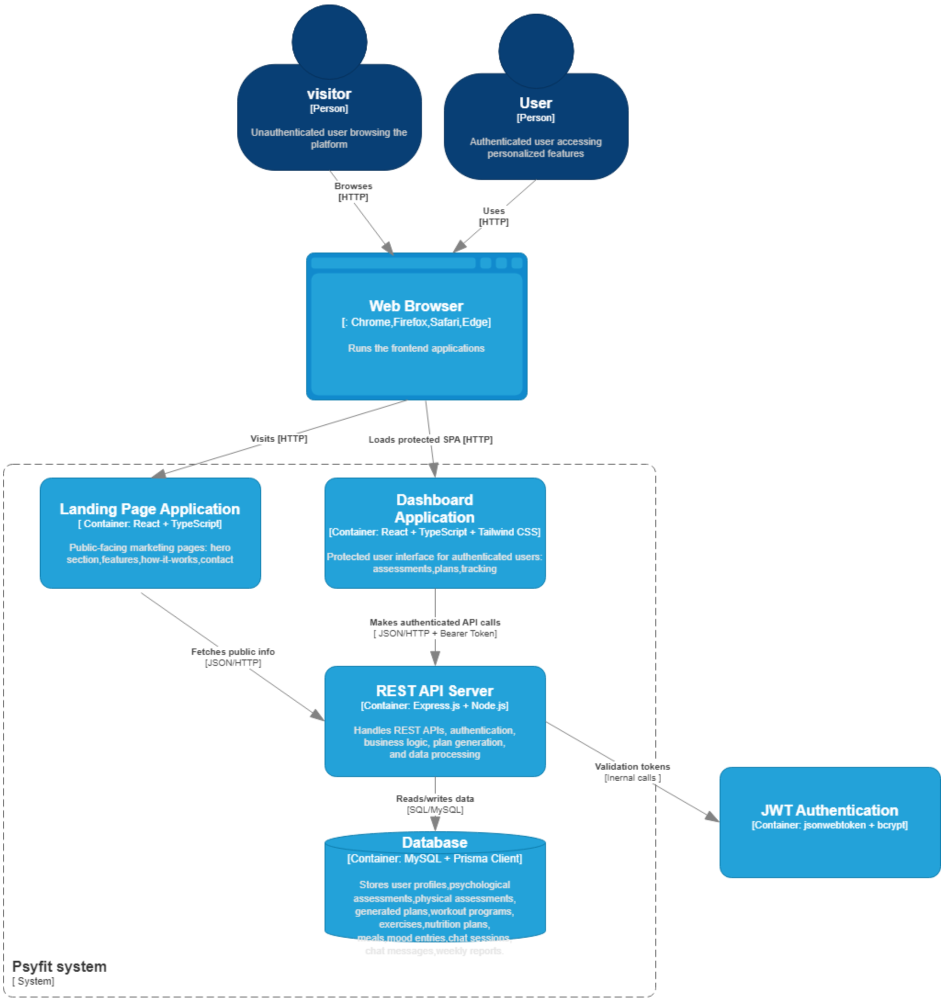

Figure 2.2: C4 Model - Container Diagram

3 Back-end Server

Through the C4 model analysis, we now examine the internal architecture of the PsyFit back-end server, PsyFit uses a single-database approach with role-based access separation. The following breakdown shows how responsibilities are distributed across the different technical layers and explains how these components interact to ensure efficient performance and easy maintainability.

3.1 Main components of the backend

Routing Layer (Express.js Router)

Within the backend architecture of PsyFit, we designed the routing layer using Express.js built-in routing system. This layer is responsible for receiving incoming HTTP requests and directing them to the appropriate handler functions. It selects the correct endpoint for each request based on the URL path and the HTTP method (GET, POST, PUT, DELETE). Through this layer, we can organize the API structure cleanly and facilitate access to the various functions provided by the system, such as authentication, assessment submission, and plan retrieval.

Middleware Layer (Express.js Middleware)

The backend architecture also includes a middleware layer implemented through Express.js middleware functions. This layer acts as a bridge between the routing layer and the business logic. We designed this layer to pre-process requests before they reach the endpoint handlers. The main tasks performed at this level include validating incoming data, authenticating users using JWT tokens, and verifying user permissions before allowing access to sensitive operations. For example, the authentication middleware checks the token and retrieves the user object before any protected endpoint executes. This layer is essential for maintaining security and proper request validation throughout the system.[25].

Controller Layer (Route Handlers)

Controllers represent one of the most important parts of the backend architecture because they contain the core business logic. We created a set of route handler functions to process different types of requests coming from the frontend applications. These handlers manage user registration and login, profile updates, psychological and physical assessment submissions, automatic plan generation, workout tracking, nutrition logging, mood recording, chat interactions, and report generation. Each controller coordinates between other layers such as the database access layer and the authentication system, ensuring that requests are processed correctly and responses are returned in the expected format.

3.2 Database Layer

Main Database (MySQL)

PsyFit uses a single MySQL database that contains all the data for the entire platform. Unlike multi-tenant systems that maintain separate databases per tenant, our approach stores user profiles, psychological assessments, physical assessments, generated plans, workout programs, exercises, nutrition plans, meals, mood entries, chat sessions, messages, and weekly reports all within one unified schema. This simplifies deployment and reduces the complexity of managing multiple database connections. User separation is achieved through foreign key relationships and query filtering rather than physical database isolation.

Prisma Client ORM

To facilitate interaction with the database, we adopted the Prisma Client ORM library. This tool allows database queries to be written in a structured and secure way using JavaScript objects instead of raw SQL strings. It also enables the automatic transformation of database rows into JavaScript objects that can be easily manipulated within the backend code. Using an ORM reduces the risk of SQL injection attacks, improves development productivity by eliminating repetitive query writing, and makes the code more readable and maintainable. Prisma also handles relationship loading, transaction management, and connection pooling automatically. Additionally, Prisma provides type safety through generated TypeScript types, which helps catch errors during development rather than at runtime.

4 Analyze the system using a Use Case diagram

After providing a comprehensive overview of the system architecture through the C4 diagrams and detailing the database structure using the database layer and ORM, we now move on to analyze the system behavior from the user's perspective. The Use Case Diagram is used to identify the main interactions between users (actors) and the system, and shows the basic functions provided by the application. This diagram helps to understand how the system is used by each actor (such as visitor, authenticated user during assessment, and authenticated user accessing the dashboard), and contributes to accurately determining functional requirements.

For PsyFit, we have identified three distinct actors based on their authentication status and stage in the user journey. The Visitor represents unauthenticated users who are exploring the platform. The User (Assessment) represents authenticated individuals who are completing the psychological and physical evaluations. The User (Dashboard) represents authenticated individuals who have finished the assessments and are now using the personalized features of the platform. Each actor has a different set of use cases that reflect their goals when interacting with PsyFit.

4.1 Visitor Use Case Diagram

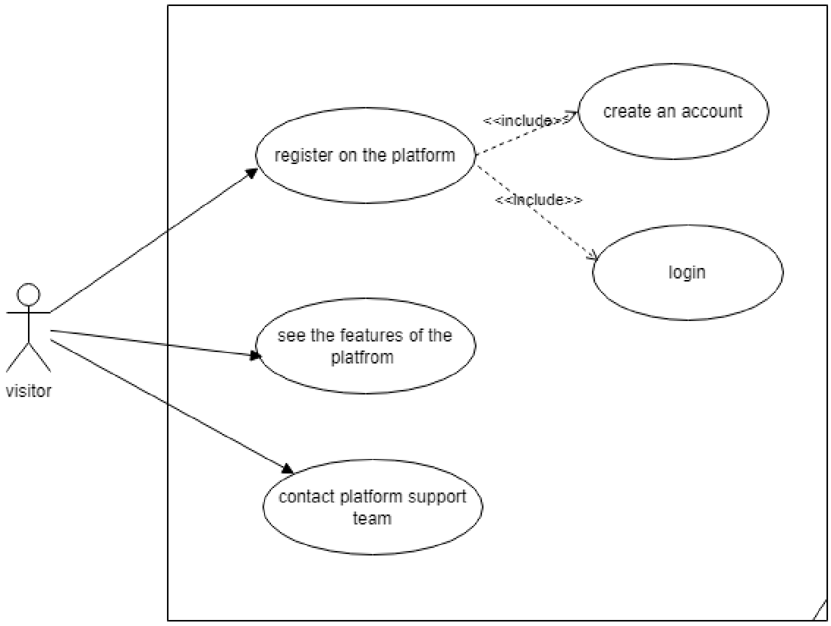

Figure 2.3: Visitor Use Case Diagram

Actors:

- **Visitor**: An individual who has not yet created an account or logged into the platform. Visitors can only access public information and cannot perform any personalized actions.

Use Cases:

- **Register on the platform**: The visitor can create a new account by providing the required information. This use case includes creating an account and logging in.
- **See the features of the platform:** The visitor can browse the landing page to learn about what PsyFit offers, including psychological assessments, personalized plans, mood tracking, and chatbot assistance.
- **Contact platform support team:** The visitor can reach out to the support team through the contact form for any inquiries or assistance.

4.2 User Assessment Use Case Diagram

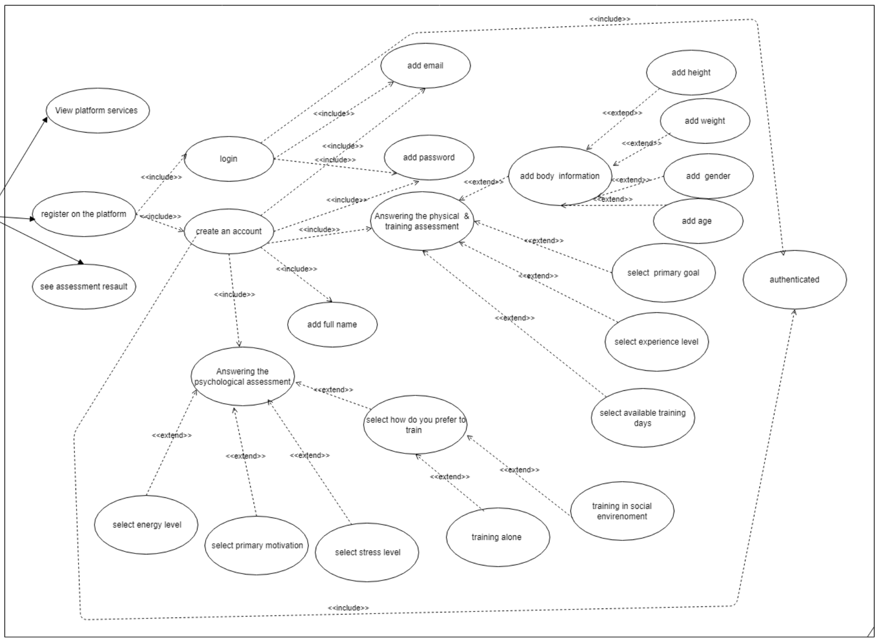

Figure 2.4: User Assessment Use Case Diagram

Actors:

- **User (Authenticated)**: A registered individual who has successfully logged into the platform and is completing the assessment process to receive personalized recommendations.

Use Cases:

- **Register on the platform**: The user creates an account by providing email, password, and full name. This includes adding email, adding password, and adding full name.
- **Login**: The authenticated user accesses their account using email and password.
- **Answer the psychological assessment**: The user completes the psychological evaluation by providing:
  - Energy level (low, moderate, or high).
  - Primary motivation (low, medium, or high).
  - Stress level (on a scale from very low to very high).
  - Training preference (training alone or training in social environment).

- **Answer the physical and training assessment**: The user completes the physical and training evaluation by providing:
  - Body information (age, gender, weight, height).
  - Primary goal (improve general fitness, build muscle, or lose weight).
  - Experience level (beginner, intermediate, or advanced).
  - Available training days (2-3, 4, or 5-6 days per week).
- **See assessment result**: After completing both assessments, the user can view their personalized results including BMI, daily calories, macros, and recommended program.

4.3 User Dashboard Use Case Diagram

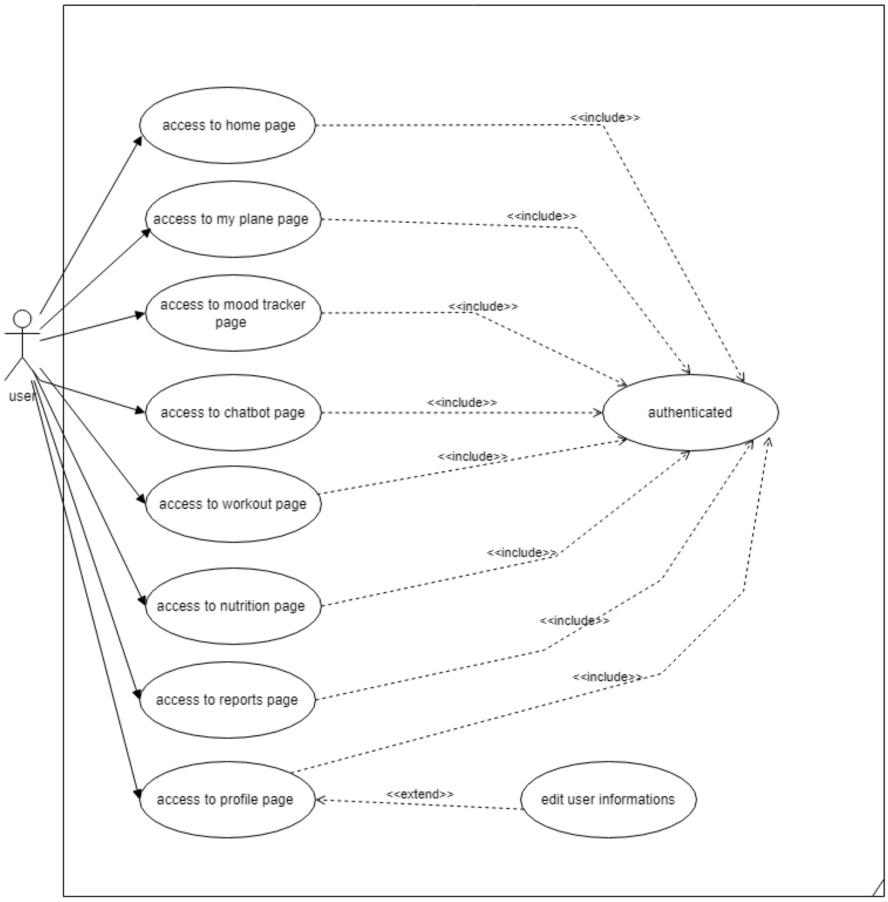

Figure 2.5: User Dashboard Use Case Diagram

Actors:

- **User (Authenticated)**: A registered individual who has completed the assessment and now accesses the dashboard to utilize the personalized features of the platform.

Use Cases:

- **Access to home page**: The user can view the main dashboard displaying today's progress, quick stats, and recommended tasks.
- **Access to my plan page**: The user can view their personalized daily plan including workouts, meals, and mental wellness tasks.
- **Access to mood tracker page**: The user can log daily mood entries and view historical mood trends.
- **Access to chatbot page**: The user can interact with the AI wellness assistant for support with stress, motivation, sleep, and other concerns.
- **Access to workout page**: The user can view their workout program, see weekly schedules, and mark exercises as completed.
- **Access to nutrition page**: The user can view their meal plans, track consumed meals, and see remaining calories.
- **Access to reports page**: The user can generate weekly reports showing mood averages, workout completion, streaks, and insights.
- **Access to profile page**: The user can view and edit their personal information. This use case extends to edit user information including updating profile details.

5 System Analysis Using Class Diagrams

After creating the main structure of the system with the C4 model, and depicting user interactions via Use Case Diagrams, we now use Class Diagrams to represent the system's static structure. These diagrams provide information about:

- Main entities like User, PsychologicalAssessment, PhysicalAssessment, GeneratedPlan, WorkoutProgram, NutritionPlan, MoodEntry, ChatSession, and WeeklyReport, and their details (data fields).
- Relationships between classes, such as associations and compositions.
- How data moves between parts of the system, like linking a GeneratedPlan to a WorkoutProgram and a NutritionPlan.

Using these diagrams, we change real-world ideas, like a user completing an assessment and receiving a personalized plan, into a clear software design that developers and stakeholders can understand and use easily.

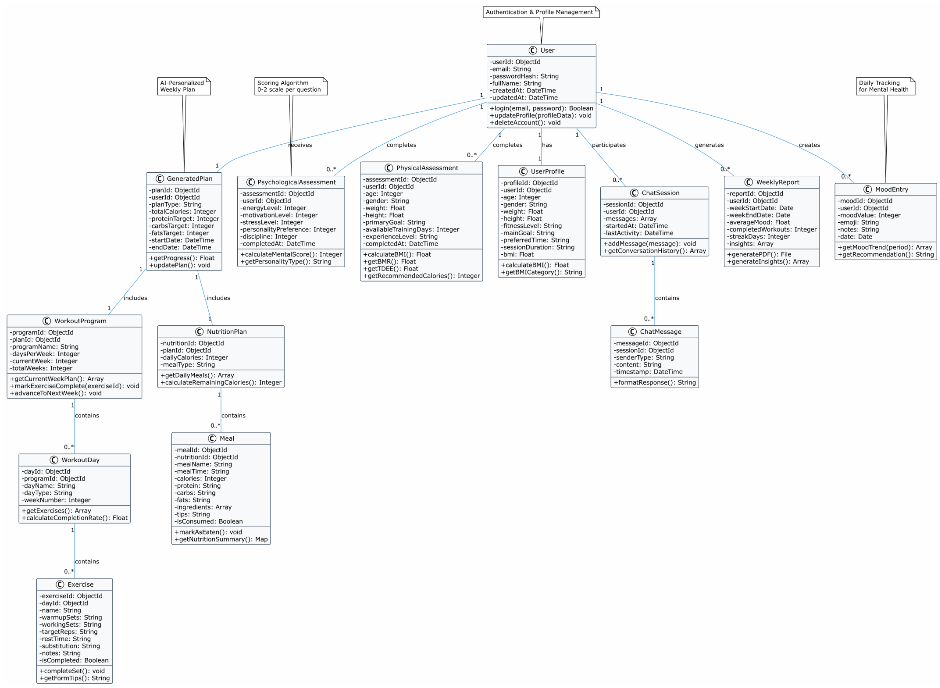

Figure 2.6: Class Diagram

1. User

The base class that represents any person interacting with the PsyFit platform. It contains general information such as:

- userId
- email
- passwordHash
- fullName
- createdAt
- updatedAt

2. UserProfile

A class that extends User with physical attributes and fitness preferences. Contains:

- age

- gender
- weight
- height
- fitnessLevel
- mainGoal
- preferredTime
- sessionDuration
- bmi

3. PsychologicalAssessment

Represents the psychological evaluation completed by the user. Includes:

- energyLevel
- motivationLevel
- stressLevel
- personalityPreference
- discipline
- completedAt

4. PhysicalAssessment

Represents the physical evaluation completed by the user. Includes:

- age
- gender
- weight
- height
- primaryGoal
- availableTrainingDays
- experienceLevel
- completedAt

5. GeneratedPlan

The central plan entity linked to a user. Contains:

- planType
- totalCalories
- proteinTarget
- carbsTarget
- fatsTarget
- startDate
- endDate

6. WorkoutProgram

Represents the exercise regimen within a plan. Includes:

- programName
- daysPerWeek
- currentWeek
- totalWeeks

7. WorkoutDay

Represents a single training day within a program. Includes:

- dayName
- dayType
- weekNumber

8. Exercise

Represents an individual exercise within a workout day. Includes:

- name
- warmupSets
- workingSets

- targetReps
- restTime
- substitution
- notes
- isCompleted

9. NutritionPlan

Represents the dietary component within a plan. Includes:

- dailyCalories
- mealType

10. Meal

Represents an individual meal within a nutrition plan. Includes:

- mealName
- mealTime
- calories
- protein
- carbs
- fats
- ingredients
- tips
- isConsumed

11. MoodEntry

Tracks the user's daily emotional state. Includes:

- moodValue (1 to 5)
- emoji
- notes
- date

12. ChatSession

Represents a conversation between a user and the AI assistant. Includes:

- startedAt
- lastActivity

13. ChatMessage

Represents an individual message within a chat session. Includes:

- senderType (user or ai)
- content
- timestamp

14. WeeklyReport

Stores aggregated summaries of user activity. Includes:

- weekStartDate
- weekEndDate
- averageMood
- completedWorkouts
- streakDays
- insights

6 Analyze the System Using Sequence Diagrams

After presenting the system architecture using the C4 model, examining the backend server components, and analyzing user interactions via Use Case Diagrams, we now utilize Sequence Diagrams to explore how the system operates step by step.

Each diagram illustrates how the system components — such as the frontend, backend (API layer), and the MySQL database — interact in real time during key operations. These diagrams help visualize the flow of requests and responses, clarifying the internal logic of user actions like registering, completing assessments, viewing the dashboard, logging mood, and generating reports.

6.1 User Registration and Assessment Flow – Sequence Diagram

1. The user accesses the platform's landing page and clicks the "Start Free" button. The frontend displays the signup form where the user enters their email, password, and full name.

2. The frontend sends a POST request to the authentication API endpoint (/api/v1/auth/register) with the provided credentials.

3. The backend receives the request and performs several operations. It hashes the password using bcrypt, creates a new user record in the users table, and generates a JWT access token containing the user's ID as the subject claim. The token is signed using the HS256 algorithm.

4. The backend returns the JWT token to the frontend. The frontend stores this token in localStorage for use in subsequent authenticated requests.

5. The user is redirected to the psychological assessment page. For each question, the frontend temporarily stores the responses in localStorage since the assessment is completed in two stages.

6. After finishing the psychological questions, the user proceeds to the physical assessment form and submits their physical data including age, gender, weight, height, primary goal, experience level, and available training days. The frontend combines both sets of assessment data and sends a POST request to the physical assessment endpoint (/api/v1/assessment/physical) with the JWT token in the Authorization header.

7. The backend verifies the JWT token by checking its signature and expiration. If valid, it extracts the user ID and retrieves the user record. The backend then calculates the BMI using the formula weight / $(\text{height}/100)^2$, calculates the BMR using the Mifflin-St Jeor equation based on gender, and calculates the TDEE by multiplying the BMR by an activity multiplier determined by the available training days.

8. Based on the user's primary goal, the backend determines the daily calorie target (TDEE - 500 for fat loss, TDEE + 300 for muscle gain, or TDEE for maintenance). It then calculates the macronutrient targets using fixed percentage splits appropriate for each goal.

9. The backend automatically generates a complete 8-week plan by inserting records into the generated_plans, workout_programs, workout_days, exercises, nutrition_plans, and meals tables.

10. Finally, the backend returns the complete plan data to the frontend, which displays the assessment results page. When the user clicks "Go to Dashboard", the frontend redirects them to the main dashboard.

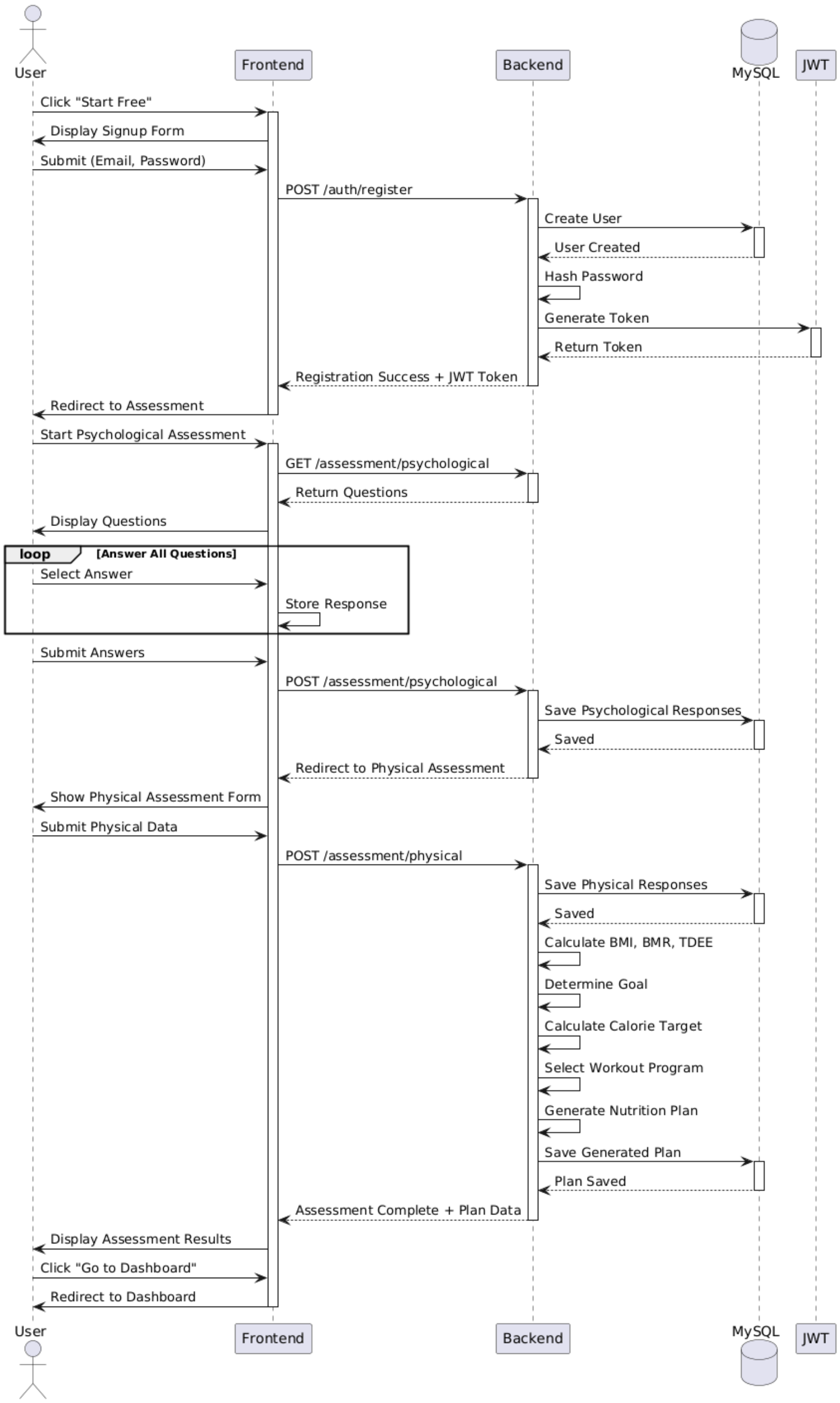

Figure 2.7: User Registration & Assessment Flow - Sequence Diagram

6.2 Dashboard and Daily Plan – Sequence Diagram

This sequence diagram explains how the dashboard retrieves and displays personalized data for the user, and how the user interacts with tasks:

1. The user navigates to the /dashboard page. The frontend sends multiple authenticated GET requests to fetch the user profile, today's workout plan, today's meals, and recent mood entries. Each request includes the JWT token in the Authorization header as a Bearer token.

2. For each request, the backend verifies the JWT token by checking its signature using the secret key and validating the expiration time. If the token is valid, the backend extracts the user ID and queries the appropriate tables. The user_profiles table is queried for profile data. The workout_days and exercises tables are queried for today's workouts. The meals table is queried for today's nutrition plan.

3. The backend returns all the requested data to the frontend. The frontend calculates the progress percentage by counting completed tasks against total tasks and renders the dashboard with the progress bar, task list, and quick stats.

4. When the user clicks on a workout task to start exercising, the frontend sends a GET request to the /api/v1/workouts/today endpoint to fetch the detailed exercises for the current day.

5. The user performs the exercise and then clicks the complete button. The frontend sends a PUT request to the /api/v1/workouts/complete/{exercise_id} endpoint. The backend verifies the token, updates the is_completed field of the corresponding exercise record in the database to TRUE, and returns the updated status.

6. The frontend receives the completion confirmation and refreshes the progress display, showing the user their updated completion status in real time.

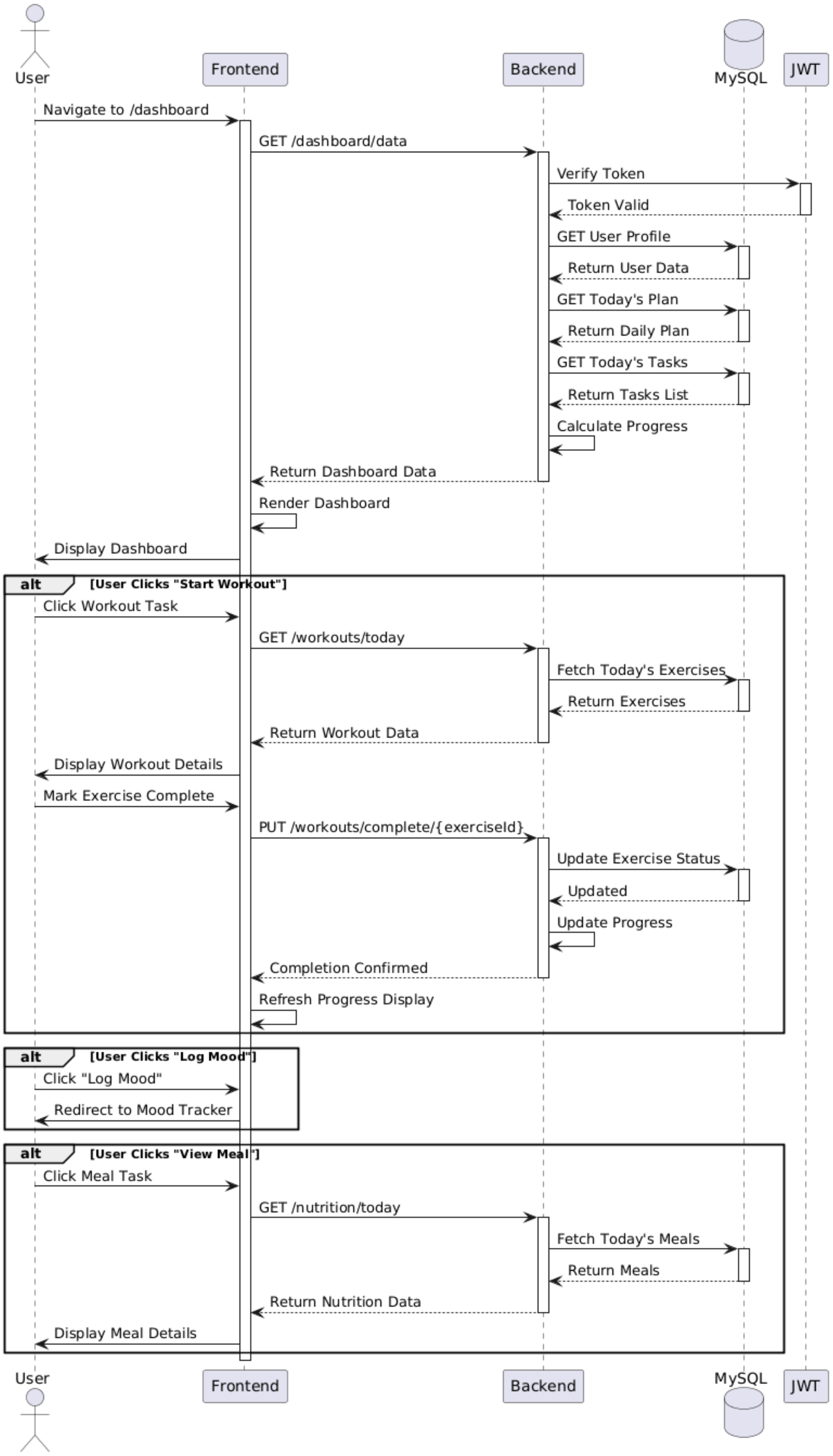

Figure 2.8: Dashboard and Daily Plan - Sequence Diagram

6.3 Weekly Report Generation – Sequence Diagram

This diagram shows how a user generates and exports a weekly progress report:

1. The user navigates to the /dashboard/reports page. The frontend sends a GET request to the /api/v1/reports/weekly endpoint with the JWT token in the Authorization header.

2. The Auth Middleware checks if the token is valid by verifying its signature using the secret key and checking the expiration claim. If valid, it extracts the user ID from the token's payload and allows the request to proceed to the report handler.

3. The API queries the mood_entries table for mood data from the last 7 days, filtering by the user ID and date range. It also queries the exercises table joined with workout_days to retrieve workout completion data for the same period. Additionally, it queries the user_profiles table for weight history if available.

4. Using this retrieved data, the API calculates the average mood score by summing all mood values and dividing by the number of entries. It determines the workout streak by checking consecutive days where at least one exercise was marked as completed. It also calculates the completion rate for workouts and meals.

5. The API then generates personalized recommendations based on detected patterns. For example, if the average mood is below 3 out of 5, the insight might suggest light exercise or meditation. If the workout streak is high, the insight acknowledges the user's consistency.

6. The API compiles all this information into a complete report object and returns it to the frontend in JSON format. The frontend displays the report with charts generated using Recharts and shows the insights in a card component.

7. If the user clicks the "Export PDF" button, the frontend sends a request to generate a PDF file. The backend generates the PDF containing the same report data and returns the file as a downloadable attachment. The frontend then triggers the browser's download mechanism.

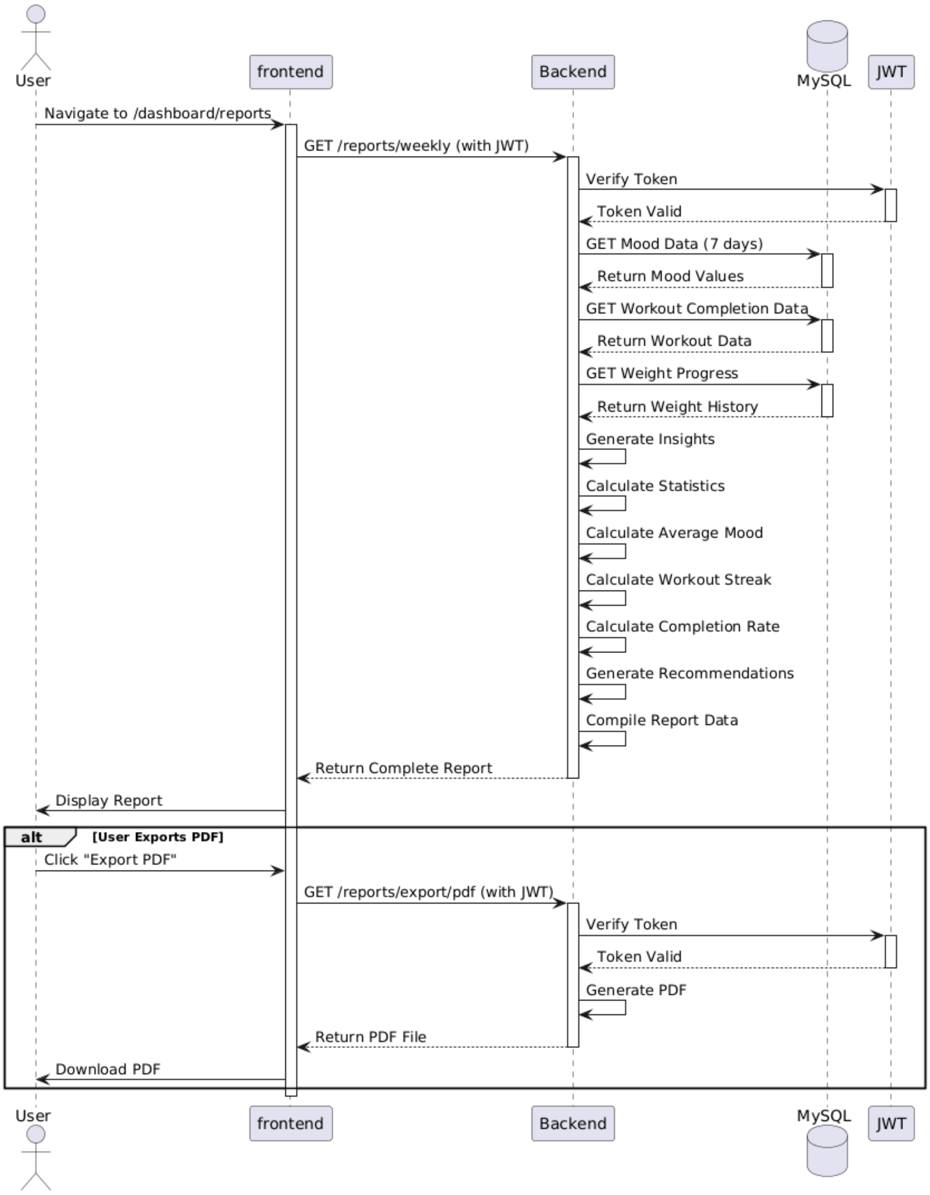

Figure 2.9: Weekly Report Generation - Sequence Diagram

6.4 Mood Tracking – Sequence Diagram

This sequence diagram demonstrates how a user logs their daily mood into the system:

1. The user navigates to the /dashboard/mood page. The frontend sends a GET request to the mood history endpoint (/api/v1/mood/history) with the JWT token included in the Authorization header as a Bearer token.

2. The backend then queries the mood_entries table in the MySQL database to retrieve mood records from the last 7 days. The query filters by the user ID and the date range, returning the mood values and their corresponding dates.

3. The database returns the mood history data to the backend. The backend processes this data by organizing it into a format suitable for chart visualization, typically an array of objects containing date and mood value pairs.

4. The backend returns both the raw history data and the prepared chart data to the frontend. The frontend renders the mood selection interface, displaying a 5-point emoji scale ranging from very sad (1) to excellent (5).

5. The user selects a mood value by clicking on the corresponding emoji button. The frontend stores this selection temporarily in its local state. The user may also add optional notes describing how they are feeling.

6. When the user clicks the "Record Mood" button, the frontend sends a POST request to the mood endpoint (/api/v1/mood) containing the mood value and any notes provided. The request includes the JWT token for authentication.

7. The backend again verifies the JWT token. Once confirmed, it inserts a new record into the mood_entries table with the user ID, mood value, notes, and the current date as the timestamp.

8. After successfully saving the mood entry, the backend returns a success response to the frontend.

9. The frontend updates its local state by appending the new mood entry to the existing history and refreshing the chart display. The user sees a success message confirming that the mood has been recorded, and the updated mood chart appears with the new entry reflected.

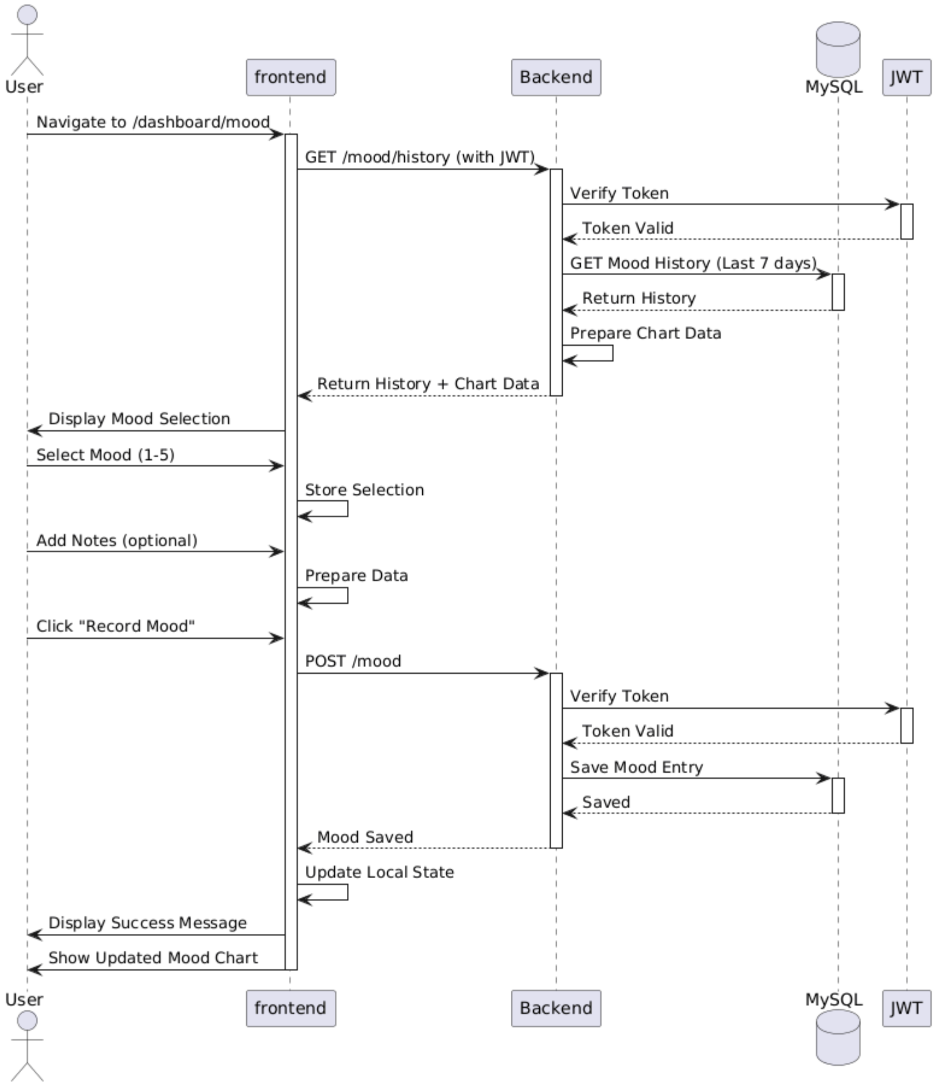

Figure 2.10: Mood Tracking - Sequence Diagram

7 Integrating Artificial Intelligence into PsyFit via the Gemini API

When we first designed the PsyFit platform, the original plan was to build a custom psychological-fitness profiling model, an AI component that analyzes behavioral patterns and provides tailored fitness and nutrition recommendations based on a user's mental state. However, during the development phase we realized that the scope of training and fine-tuning such a model from scratch exceeded the available time and computational resources for this project. Therefore, we elected to incorporate an existing Large Language Model via an external API, namely Google's Gemini, to provide intelligent, conversational support to users directly within the platform.

7.1 Large Language Models

Definition:

Large Language Models (LLMs) are deep learning-based artificial intelligence models that were previously trained on massive volumes of text data with the goal of understanding natural language and producing responses that resemble those of humans.

Methods of Using LLMs in Software Systems:

- **Direct Use**: Without the requirement for programming, the user interacts with the model using a pre-made interface like the Gemini web platform or ChatGPT.
- **API Integration**: Through the use of an API key, the model is integrated into a software application to enable intelligent interaction, automated content generation, and decision-making capabilities.

7.2 Using the Gemini API in PsyFit with Vercel AI SDK

What is the Gemini API?

Gemini is a multimodal AI model from Google DeepMind that can comprehend code, pictures, and text all at once. It is appropriate for PsyFit since it can translate casual user inquiries into precise and useful answers and supports several languages, including Arabic[26].

What is Vercel AI SDK?

The Vercel AI SDK is a modern TypeScript toolkit designed to simplify the integration of AI models into JavaScript and TypeScript applications. It provides a unified interface for working with various AI providers including Google Gemini, OpenAI, and Anthropic. The SDK handles streaming responses, manages conversation state, and provides React hooks for easy frontend integration[27].

Method of Integration Within the System

```
src > modules > chatbot > chatbot.schemas.js > ...
import { z } from "zod";

export const sendMessageSchema = z.object({
  body: z.object({
    sessionId: z.string().uuid().optional(),
    message: z.string().min(1, "Message cannot be empty").max(2000),
    stream: z.boolean().default(false),
  }),
});

export const createSessionSchema = z.object({
  body: z.object({
    title: z.string().min(1).max(100).optional(),
  }),
});

export const WELLNESS_SYSTEM_PROMPT = `You are a compassionate and knowledgeable wellness assistan

Your role:
- Provide empathetic, evidence-based support for mental health, physical wellness, stress, sleep,
- Listen actively and respond thoughtfully to the user's emotional state
- Offer practical, actionable tips and techniques (e.g. breathing exercises, sleep hygiene, workou
- Ask follow-up questions to better understand the user's needs
- Encourage healthy habits without being preachy or judgmental

Boundaries:
- You are NOT a licensed therapist or doctor. For serious mental health crises or medical issues,
- Keep responses warm, concise, and supportive — typically 2–4 short paragraphs
- Never diagnose conditions or prescribe medications
- If a user expresses intent to harm themselves or others, immediately direct them to emergency se

Always respond in the same language the user writes in.`;
```

Figure 2.11: system prompt configuration for the wellness chatbot

```
import { google } from "@ai-sdk/google";
import { generateText, streamText } from "ai";
import { v4 as uuidv4 } from "uuid";
import db from "../../config/database.js";
import { config } from "../../config/index.js";
import { NotFoundError, ForbiddenError } from "../../utils/errors.js";
import { WELLNESS_SYSTEM_PROMPT } from "./chatbot.schemas.js";

// ── Gemini client setup ──────────────────────────────────

const GEMINI_MODEL = google("gemini-3.5-flash");
```

Figure 2.12: Gemini Model Setup and AI SDK Configuration

As shown in Figures 2.11 and 2.12, the integration follows a structured approach. Figure 2.11 illustrates the schema validation and system prompt configuration for the wellness chatbot, while Figure 2.12 shows the Gemini model setup and AI SDK configuration. An API key was created and connected to the project using Google AI Studio to obtain access to Gemini.

The Vercel AI SDK was installed along with the Google provider package. The following steps were implemented:

1. **Gemini Model Configuration**: As seen in Figure 2.12, the system imports the Google provider from `@ai-sdk/google` and the `streamText` function from the `ai` package. The Gemini model is configured as `google("gemini-3.5-flash")` and stored in the `GEMINI_MODEL` constant. The `uuid` library is used to generate unique session identifiers.

2. **Wellness System Prompt**: Figure 2.11 shows the wellness system prompt which defines the assistant's role as a compassionate wellness assistant. The prompt establishes clear boundaries: the assistant is not a licensed therapist or doctor, never diagnoses conditions or prescribes medications, and responds in the same language as the user. For serious mental health crises, the assistant directs users to emergency services.

3. **Request Validation using Zod**: As illustrated in Figure 2.11, all incoming chat requests are validated using Zod schemas. The sendMessageSchema validates the session ID as an optional UUID, ensures the message is not empty, and limits the message length to 2000 characters. The createSessionSchema validates the optional session title. This validation layer prevents invalid or malicious inputs from reaching the AI model.

7.3 Areas where Artificial Intelligence was Used in the Platform

Based on the chatbot prompt we configured, here are the main areas where AI helps users in PsyFit:

- **Mental Health and Emotional Support**: The AI listens to users' feelings and responds with empathy. It helps with anxiety, low mood, and emotional distress while clearly stating it is not a replacement for professional therapy.

- **Stress Management**: The AI provides practical stress relief techniques such as breathing exercises and relaxation strategies when users report feeling overwhelmed.

- **Sleep Improvement**: The AI shares simple sleep hygiene tips like maintaining a consistent bedtime, reducing screen time before sleep, and practicing relaxation techniques.

- **Nutrition Guidance**: The AI offers general healthy eating tips and meal suggestions. It encourages balanced eating habits without prescribing specific diets or medical advice.

- **Workout Ideas**: The AI suggests simple workout ideas that fit the user's energy level and fitness goals. It encourages physical activity in an approachable, non-judgmental way.
- **Motivation and Habit Building**: The AI encourages users to stay consistent with their fitness and wellness goals by offering small achievable steps and celebrating progress.
- **Multi-Language Support**: The AI responds in the same language the user writes in, supporting both Arabic and English to serve Algerian users effectively.

8 Conclusion

In this chapter, we provided a comprehensive analysis of the PsyFit system architecture and design. We utilized the C4 model to break down the system into its main components and their interactions, giving a clear overview of how the frontend, backend, and database layers are structured. We then analyzed user interactions through Use Case Diagrams, illustrating the different functionalities available to visitors, authenticated users during assessment, and authenticated users accessing the dashboard. Sequence Diagrams were used to detail the step-by-step flow of key operations such as user registration, assessment completion, dashboard interaction, mood tracking, and report generation. In addition, we presented a Class Diagram to show the main entities in the system and their relationships, providing a clear blueprint for the software design. Finally, we discussed the integration of artificial intelligence into PsyFit using the Gemini API, explaining how we configured the AI model and the areas where it provides support to users. This chapter lays the foundation for understanding the technical implementation of PsyFit and sets the stage for further development and enhancement of the platform. The next chapter will focus on the implementation details, including the technologies used, code structure, and key algorithms that power the system.

Chapter 3: Development and Implementation

1 Introduction

In this chapter, we will discuss the technologies used to build the platform, explaining how they contributed to creating a secure, fast, and user-friendly application. Then, we will present the results of our work and discuss the different aspects of the platform we developed. We will start by showcasing the various pages of the platform, providing a detailed description of their features and functionalities.

2 Used Technologies

This section describes the tools and technologies used to build the platform. It covers our development environment, programming tools, and the solutions used to make the application secure, fast, and reliable.

2.1 Development Environment

To build the system efficiently, we used a set of essential tools that helped organize our workflow and speed up development:

- **Visual Studio Code (VS Code)**: A free and powerful code editor made by Microsoft. It helps us write code faster by using features like color-coded text (syntax highlighting), smart code suggestions (auto-completion), and instant error detection.
- **Thunder Client**: A fast and lightweight API testing extension built directly into VS Code. We used it to test our backend routes and API requests (like GET and POST) safely and quickly without needing to leave our code editor.
- **Google Chrome**: A fast and popular web browser used to test the platform. We specifically used its built-in "Developer Tools" to test, inspect, and debug the website's front-end in real time.
- **Git**: A system used to track changes in our code. It helps save different versions of the project, prevents losing work, and makes it easy to go back to older versions if a mistake happens.

- **GitHub**: A cloud platform that hosts our Git code online. It is used to back up the project safely, track tasks, and manage the development process in an organized way.

2.2 Development Stack

Back-end Stack

The back-end of our system was developed using the following technologies:

- **Node.js**:A JavaScript runtime environment that allows us to run JavaScript on the server side, outside of a web browser. It is fast, efficient, and handles a large number of user requests smoothly[28].
- **Express.js**: A lightweight framework built on top of Node.js. It makes it much easier to build the server, create URL routes, and handle requests and responses when building our REST APIs[29].
- **MySQL**: A relational database management system used to safely store and organize all the platform's data, including information about users, profiles, assessments, generated plans, workouts, nutrition plans, meals, mood logs, and chat sessions[1].
- **JWT with Refresh Token**: JSON Web Tokens (JWT) are used to securely log users in and protect communication between the front-end and back-end. We included "Refresh Tokens" so users stay logged in securely without having to type their password over and over[25].
- **Prisma ORM**: is an integral part of the Prisma ORM ecosystem. It serves as a generated, type-safe query builder that replaces traditional Object-Relational Mapping (ORM) methods by letting you interact with your database using intuitive JavaScript and TypeScript functions instead of raw SQL.

Front-end Stack

The front-end of our system was developed using the following technologies:

- **HTML**: The standard markup language used to create the basic structure of our web pages, such as text, forms, and buttons.
- **CSS**: The styling language used to design the look of the website, including colors, fonts, spacing, and layouts.
- **JavaScript**: The main programming language used to make the website interactive, handling things like checking form inputs and talking to the server in real time.

- **React.js**: A powerful JavaScript library used to build the user interface using "components" (small, reusable pieces of UI). This makes the code highly organized and easy to maintain[30].

- **TailwindCSS and Shadcn/ui**: TailwindCSS is a styling framework that helps us design beautiful interfaces quickly by writing styles directly inside the HTML. Shadcn/ui is a library built on Tailwind that provides clean, professional, and ready-to-use design components like buttons and dropdowns[31].

- **Vite**: A modern development tool that makes building the website much faster. It helps load the project instantly in our browser while we are writing code.

- **TypeScript**: A programming language that builds on JavaScript by adding static types. It helps catch errors early and makes the code easier to understand and maintain[32].

- **React Router DOM**: A library that allows us to create multiple pages in our React application and navigate between them without reloading the page, providing a smooth user experience.

3 Presentation of the Platform

3.1 Landing page

The landing page of the platform is designed to provide users with an overview of the available features and functionalities.

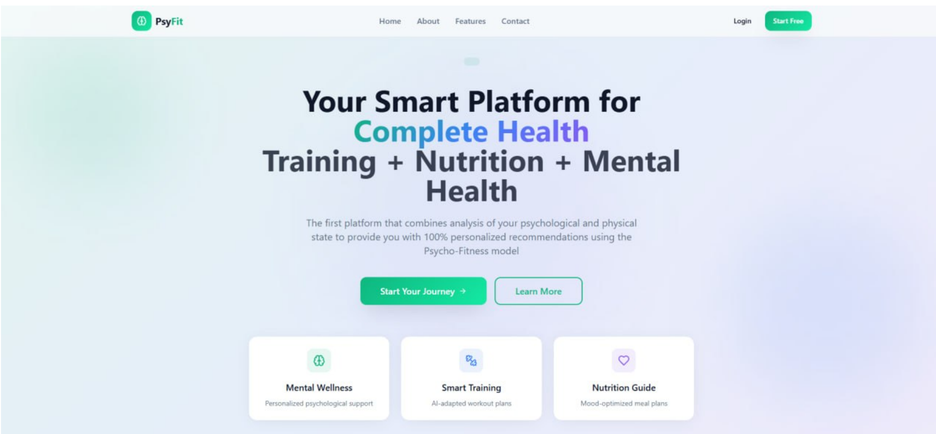

Figure 3.1: Landing page

3.2 Login page

The login page allows users to access their accounts after verifying their email and password.

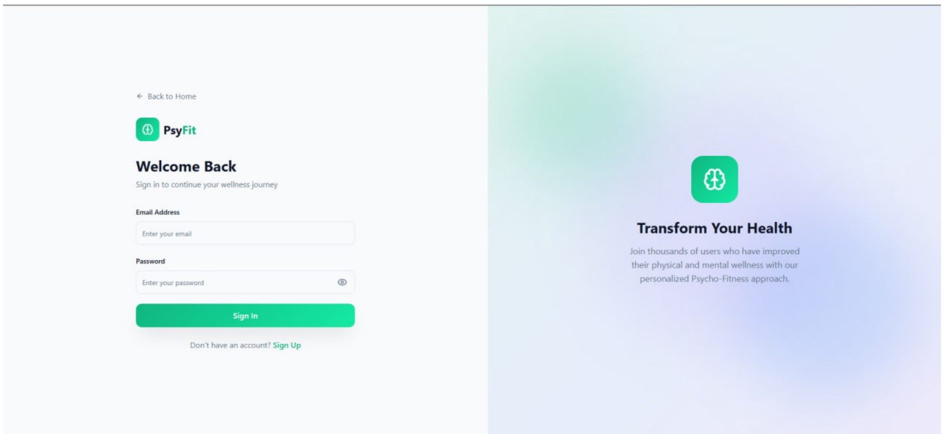

Figure 3.2: Login page

3.3 Sign Up page

The sign-up page enables new users to create an account by providing necessary information such as fullname, email, and password.

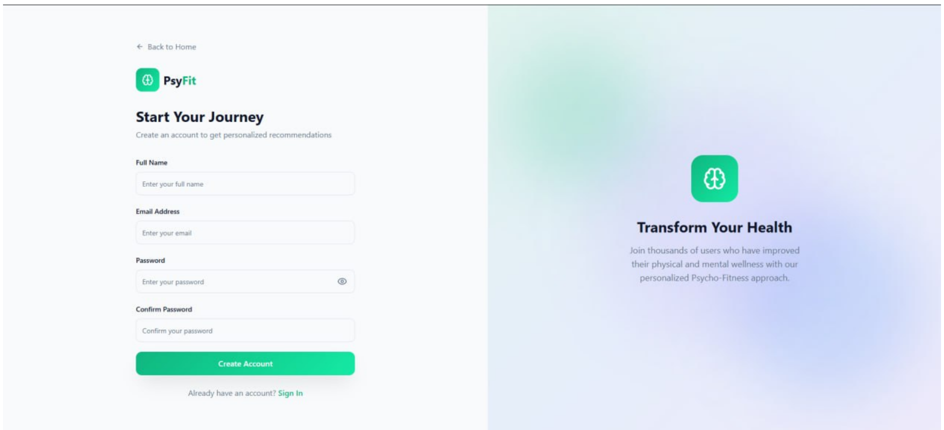

Figure 3.3: Sign Up page

3.4 Psychological Assessment page

The psychological assessment or (features) page allows users to complete a series of questions divided into two sections to evaluate their mental health status.

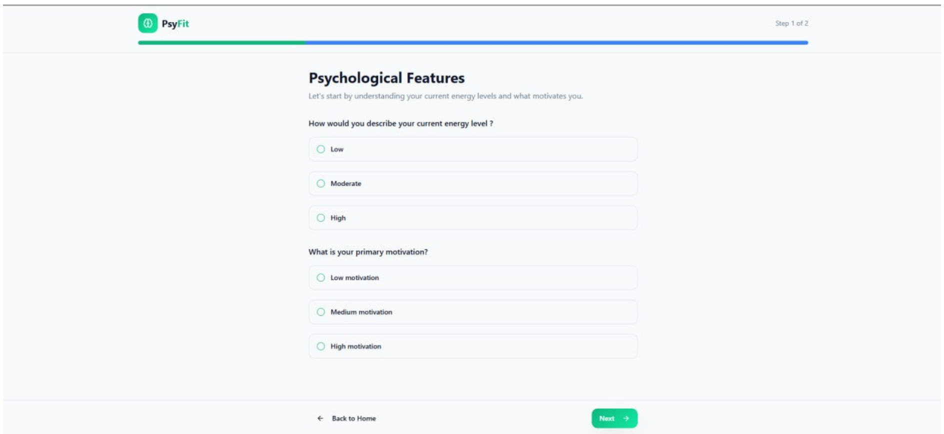

Figure 3.4: Psychological Assessment section 1 page

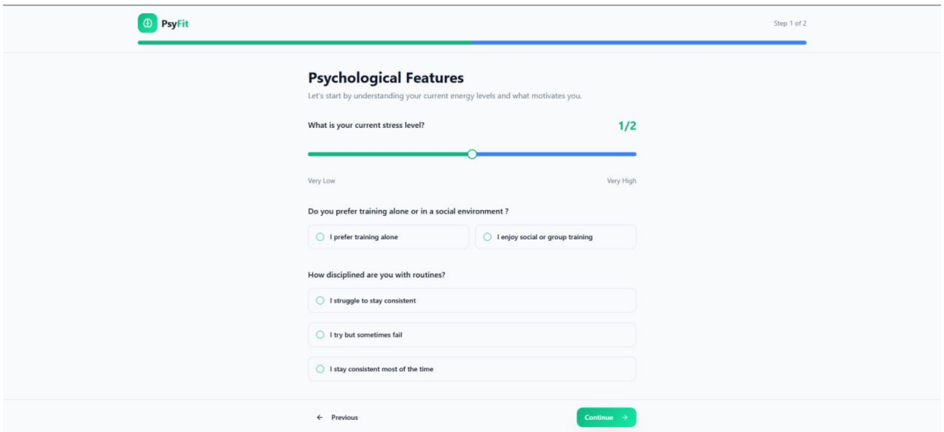

Figure 3.5: Psychological Assessment section 2 page

3.5 Physical and Training Assessment page

The physical and training assessment page allows users to complete a series of questions divided into two sections (BMI calculation, training features) to evaluate their physical health status and training habits.

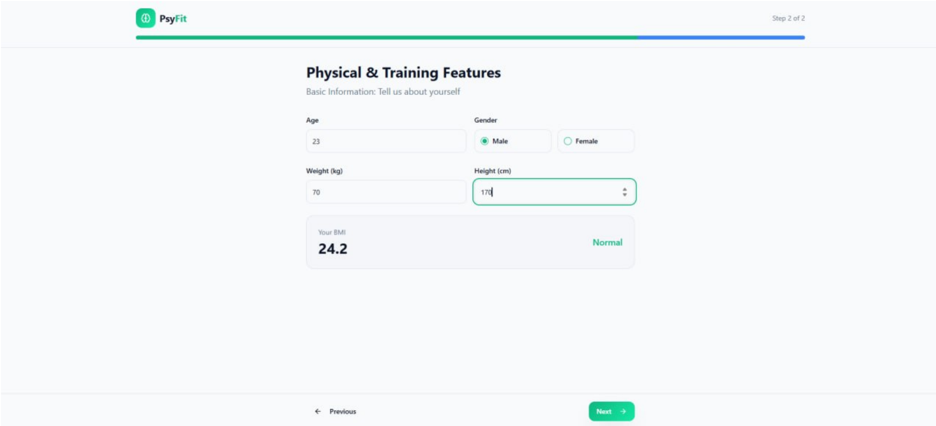

Figure 3.6: user BMI calculation page

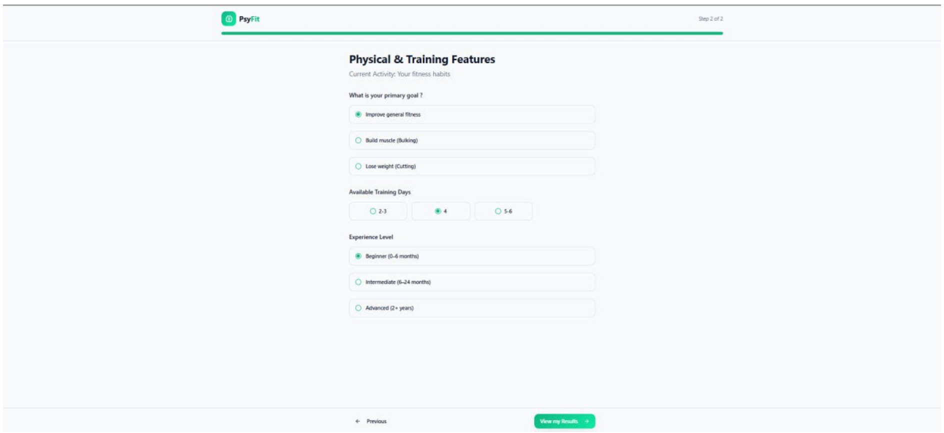

Figure 3.7: user physical and training assessment page

3.6 User assessment results

The user assessment results page displays daily caloric intake, BMI, mental health status, and the recommended workout program based on the user's responses to the psychological and physical assessment questions.

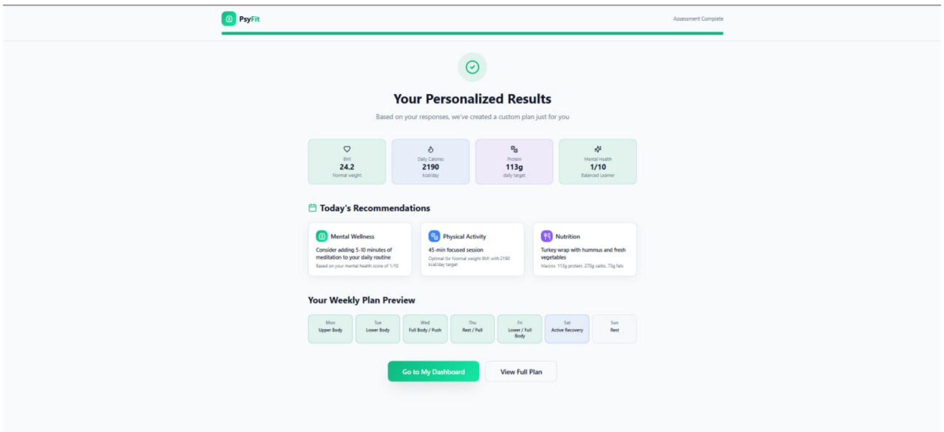

Figure 3.8: user assessment results page

3.7 User Dashboard

Home page

The user dashboard home page provides users with an overview of their today's plan.

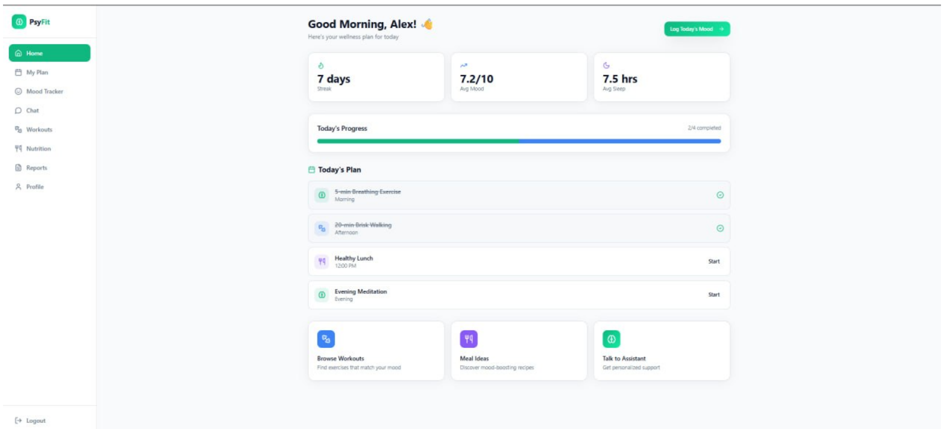

Figure 3.9: home page

My plan page

The my plan page allows users to view their today's workout and nutrition plans, and see their progress.

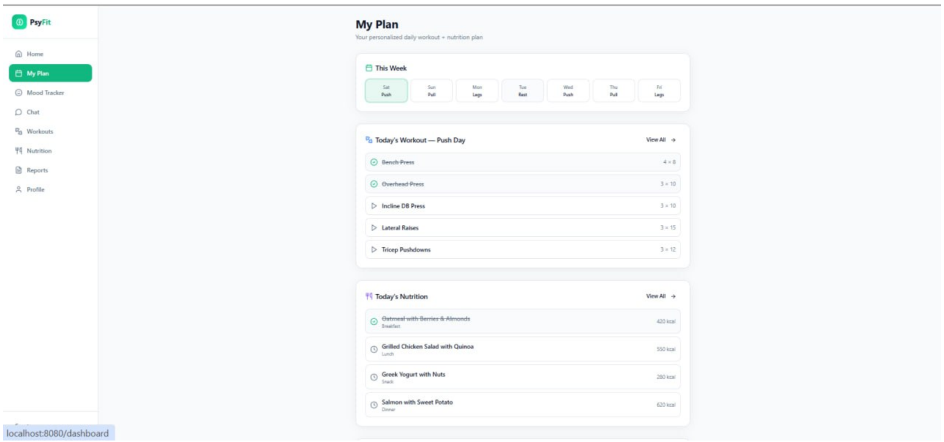

Figure 3.10: my plan page

Mood tracker page

The mood tracker page allows users to track their mood and mental health status over time.

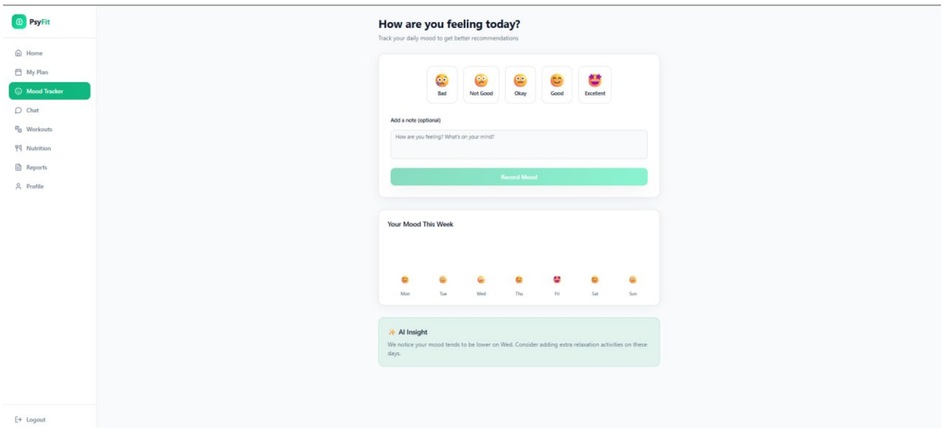

Figure 3.11: mood tracker page

chatbot page

The chatbot page allows users to interact with a virtual assistant that provides support and guidance related to their fitness and mental health goals.

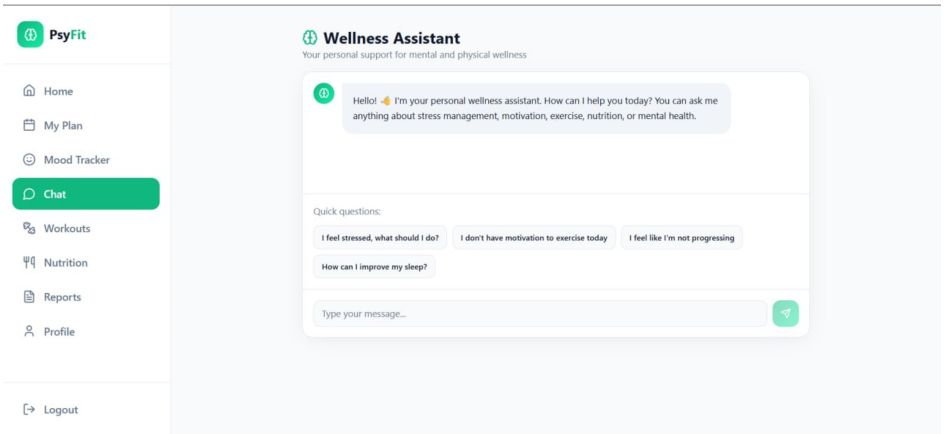

Figure 3.12: chatbot page

Workout page

The workout page shows the user's workout plan for this week, including exercises, warm-up sets, sets, reps in each set and rest periods.

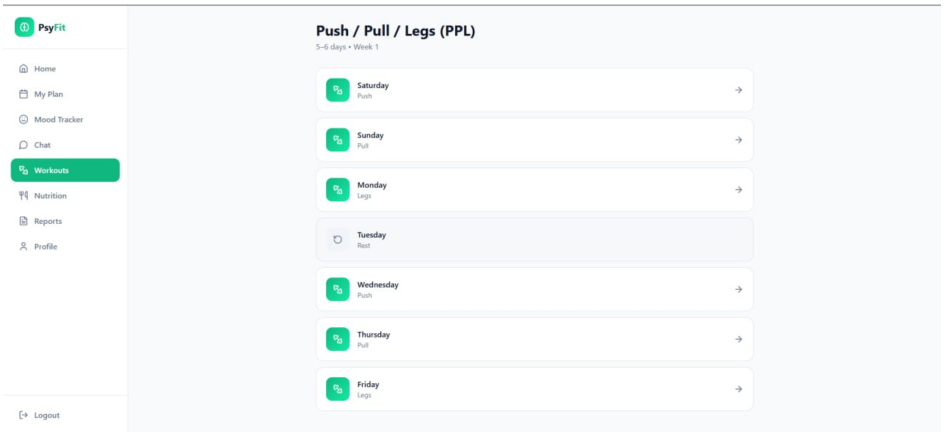

Figure 3.13: workout page

Nutrition page

The nutrition page shows the user's daily meal plan, including breakfast, lunch, dinner, and snacks, along with the calories intake for each meal and the total daily calories intake.

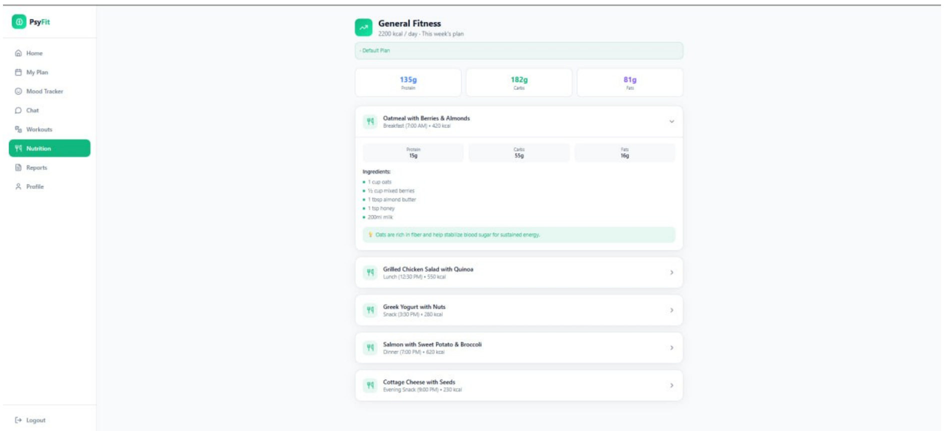

Figure 3.14: nutrition page

Report page

The report page allows users to view their progress and mood status over time, and export their reports in PDF format.

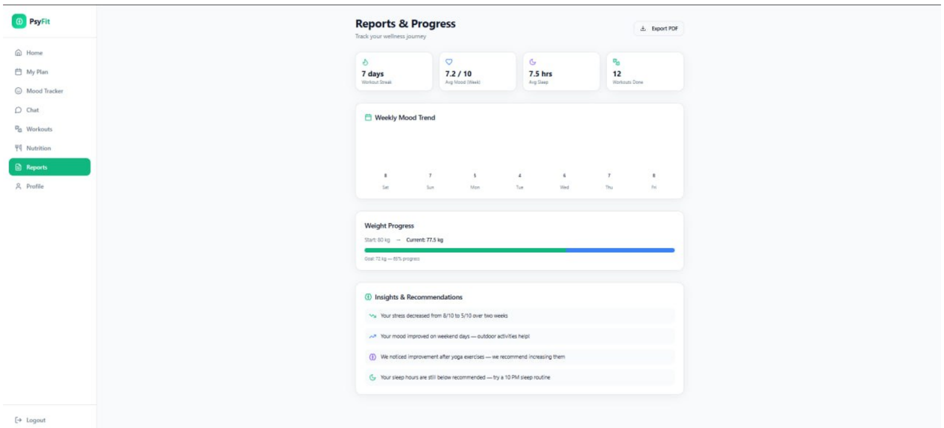

Figure 3.15: report page

Profile page

The profile page allows users to view and edit their personal information, such as fullname, email, and password.

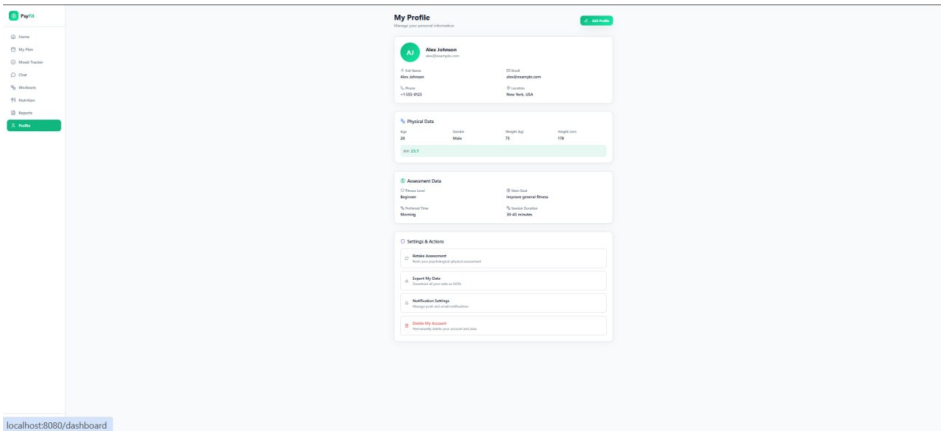

Figure 3.16: profile page

4 Conclusion

In this chapter, we presented the results of our work by showcasing the different pages of the platform and describing the technologies used to build it. We provided a detailed overview of the user interface and the development stack, highlighting how each technology contributed to creating a secure, fast, and user-friendly application.

General Conclusion

In this work, we presented **PsyFit**, an innovative health and fitness platform that bridges the gap between physical conditioning and psychological well-being. By addressing the limitations of contemporary fitness applications, which predominantly focus on physical attributes while ignoring psychological factors, this project has demonstrated the feasibility of a holistic, personalized training companion.

Summary of Achievements

The development of PsyFit has successfully achieved several milestones:

- Designed and implemented a responsive web-based prototype utilizing a modern stack composed of React for the frontend, Express.js for the backend, and MySQL for database management.
- Enabled automated user profiling incorporating physical measurements (such as CDC-compliant BMI calculation) alongside rule-based fitness and macronutrient nutrition plans tailored to different difficulty levels.
- Developed and integrated a mood tracking interface along with an external API-powered psychology chatbot, offering users immediate psychological support and mental health monitoring.

Key Contributions

The main contributions of this project include:

- Establishing an integrated architectural framework that treats physical exercise and psychological states as interconnected domains.
- Providing a functional baseline that showcases how rule-based algorithms and conversational AI agents can enhance personalization in health applications.
- Offering a scalable web application structure capable of supporting real-time feedback loops and continuous profile adaptation.

Future Work and Perspectives

While the current prototype achieves its primary objective of integrating physical training with psychological tracking, several areas remain open for future enhancement:

- **Development of the Psycho-Fitness Model**: Due to time constraints within the project schedule, a formal, fully realized psycho-fitness model could not be constructed and integrated. In future iterations, we plan to research and build this comprehensive model to dynamically adapt fitness and nutrition plans based on deep cognitive, motivational, and behavioral datasets rather than static rules.
- **Feature Completion**: Finalize the weekly reporting module to provide users with visual, long-term analytics regarding their mood variations and workout consistency.
- **Platform Expansion**: Transition the web-based proof-of-concept to native mobile applications (iOS and Android) to improve accessibility and allow on-the-go tracking.
- **Monetization and Deployment**: Integrate secure payment processing gateways and prepare the platform for public cloud deployment.

Ultimately, PsyFit lays the groundwork for next-generation wellness systems that place human psychology at the center of physical development.

References

[1] MySQL. *MySQL Getting Started Guide*. Accessed: May 09, 2026. 2026. URL: https://dev.mysql.com/doc/mysql-getting-started/en/.

[2] ``The Personalized Fitness and Nutrition Coach.'' In: *International Journal of Research Publication and Reviews* 6.4 (2026). Accessed: May 09, 2026. URL: https://ijrpr.com/uploads/V6ISSUE4/IJRPR42392.pdf.

[3] Inc. Peloton Interactive. *Peloton*. Accessed: May 03, 2026. 2026. URL: https://www.onepeloton.com/.

[4] Strava. *Strava*. Accessed: May 03, 2026. 2026. URL: https://www.strava.com/.

[5] Cronometer. *Cronometer*. Accessed: May 03, 2026. 2026. URL: https://cronometer.com/index.html.

[6] Nutritionix. *Nutritionix*. Accessed: May 03, 2026. 2026. URL: https://www.nutritionix.com/.

[7] MyFitnessPal. *MyFitnessPal*. Accessed: May 03, 2026. 2026. URL: https://www.myfitnesspal.com/.

[8] Apple Inc. *Apple Health*. App Store. Accessed: May 03, 2026. 2026. URL: https://apps.apple.com/us/app/apple-health/id1242545199.

[9] Centers for Disease Control and Prevention. *About Adult BMI*. CDC. Accessed: May 09, 2026. 2026. URL: https://www.cdc.gov/healthyweight/assessing/bmi/adult_bmi/index.html.

[10] C. B. da Silva et al. ``Self-Determination and Physical Exercise Adherence in the Contexts of Fitness Academies and Personal Training.'' In: (2015). Accessed: May 09, 2026. URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC4519215/.

[11] ``Motivational variations in fitness: a population study of exercise motives and barriers.'' In: *Frontiers in Psychology* (2024). Accessed: May 09, 2026. URL: https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2024.1377947/full.

[12] H. M. Eynon, J. F. Fallowfield, and C. M. Hillsdon. ``Assessing the psychosocial factors associated with adherence to exercise referral schemes: A systematic review.'' In: *Scandinavian Journal of Medicine & Science in Sports* (2019). Accessed: May 09, 2026. URL: https://pubmed.ncbi.nlm.nih.gov/30742334/.

[13] GeeksforGeeks. *Social Fitness App using MERN Stack*. Accessed: May 09, 2026. 2024. URL: https://www.geeksforgeeks.org/mern/social.

[14] A. X. W. Chua. *Predictive personalized workout and dietary guidance system*. Final Year Project, UTAR. Accessed: May 09, 2026. 2025.

[15] J. K. Ponte-Isminio et al. ``Web application based on artificial intelligence for the control of healthy habits in people with unbalanced diet in Lima." In: *International Journal of Online and Biomedical Engineering* 21.12 (Oct. 2025). Accessed: May 09, 2026, pp. 121–141.

[16] TechCrunch. *Healthify upgrades its AI assistant Ria with real-time conversation capabilities*. TechCrunch. Accessed: May 09, 2026. Dec. 2025.

[17] ``MedBot: Intelligent health care assistant for heart disease powered by large language models." In: *Proceedings of IEEE Conference*. Accessed: May 09, 2026. Vellore, India, July 2025.

[18] ``Development of a chatbot for supporting patients with multiple sclerosis." In: *Proceedings of IEEE Conference*. Accessed: May 09, 2026. Cacak, Serbia, June 2025.

[19] ``AI-driven personalized fitness workout assistance and tracker using YOLOv8 algorithm." In: *Proceedings of IEEE Conference*. Accessed: May 09, 2026. Bali, Indonesia, Dec. 2024.

[20] TechCabal. *London tech founder Ademola Balogun builds conversational AI infrastructure for nutrition through WhatsApp*. TechCabal. Accessed: May 09, 2026. Mar. 2026.

[21] Agricultural Research Service U.S. Department of Agriculture. *FoodData Central: USDA's comprehensive nutrient database*. Accessed: May 09, 2026. 2025. URL: https://fdc.nal.usda.gov/.

[22] Noom Inc. *Noom: The Last Weight-Loss Program You'll Need*. https://www.noom.com/. Accessed: May 09, 2026. 2024.

[23] K. E. Koopman et al. ``The effectiveness of mobile app usage in facilitating weight loss via intermittent fasting: An observational study." In: *Obesity Science & Practice* 10.3 (June 2024). Accessed: May 09, 2026. DOI: 10.1002/osp4.755. URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC11091450/.

[24] Simon Brown. *The C4 Model for Visualizing Software Architecture*. Accessed: May 10, 2026. 2023. URL: https://c4model.com/.

[25] JWT. *Introduction to JWT (JSON Web Tokens)*. Accessed: May 16, 2026. 2024. URL: https://www.jwt.io/introduction.

[26] Google AI. *Gemini API Documentation*. Accessed: May 16, 2026. 2024. URL: https://ai.google/research.

[27] Vercel. *Vercel AI SDK Documentation*. Accessed: May 16, 2026. 2024. URL: https://ai-sdk.dev/docs/introduction.

[28] Node.js Foundation. *Node.js Documentation*. Accessed: May 16, 2026. 2024. URL: https://nodejs.org/docs/latest/api/.

[29] Express.js. *Express.js Documentation*. Accessed: May 16, 2026. 2024. URL: https://expressjs.com/.

[30] React. *React Documentation*. Accessed: May 16, 2026. 2024. URL: https://react.dev/learn.

[31] Tailwind CSS. *Tailwind CSS Documentation*. Accessed: May 16, 2026. 2024. URL: https://tailwindcss.com/docs/installation/using-vite.

[32] TypeScript. *TypeScript Documentation*. Accessed: May 16, 2026. 2024. URL: https://www.typescriptlang.org/docs/.