



AD FITNESS: A PERSONALIZED WELLNESS PLATFORM INTEGRATING

¹Akshayat Dalai, ²Pawan Kumar

¹Student, Amity university Raipur Chhattisgarh, India,
Akshyatdalai1@gmail.com

²Assistant Professor, Amity University Raipur Chhattisgarh, India
pkumar@rpr.amity.edu

ABSTRACT

Poor dietary habits and physical inactivity have become defining public health crises of the modern era. Addressing this gap, we built Ad Fitness, a full-stack web application that brings together AI-driven nutritional analysis and clinically validated metabolic science under one roof. The platform runs on Next.js with a MongoDB backend and uses the Llama 3.3 70B model, hosted on Groq infrastructure and orchestrated through Google's Genkit framework, to analyze meals from plain-language descriptions in real time. Beyond food logging, the system handles calorie computation via the Mifflin-St Jeor equation, cheat meal impact analysis, workout tracking, body measurement history, and budget-aware meal planning, all while enforcing strict dietary preferences such as vegan, vegetarian, eggitarian, and non-vegetarian. Our evaluation shows that the system delivers nutritional estimates within practical accuracy ranges for everyday tracking, computes scientifically grounded calorie targets, and respects user dietary boundaries without exception. Ad Fitness demonstrates that a well-designed AI integration can genuinely lower the barrier to personalized nutrition guidance, which has historically been accessible only through costly professional consultations.

Keywords: Personalized wellness, natural language processing, Mifflin-St Jeor equation, large language models, Genkit, Groq, MongoDB, health technology, dietary preferences

1. INTRODUCTION

Walk into any gym or open any app store and you'll find no shortage of fitness tools. Yet most of them share a frustrating limitation: they assume the user already knows what they're doing. Logging a meal in a typical food-tracking app means searching a database, wrestling with portions, and hoping your dal tadka is somewhere in the system. It rarely is. The result is that people give up, or they guess, neither of which serves anyone's health goals.

The World Health Organization reports that obesity rates have nearly tripled since 1975, with over 1.9 billion adults now classified as overweight [1]. Non-communicable diseases tied to diet, type 2



diabetes, hypertension, cardiovascular conditions, continue to account for the majority of preventable deaths globally. Meanwhile, the tools most people reach for are either too generic to be useful or too complex to sustain.

Large language models have quietly changed what's possible here. A model that can read "I had chole bhature and a cutting chai" and return a sensible nutritional breakdown is genuinely useful in a way that a 500,000-item food database is not, at least not for the average Indian college student or working professional who doesn't weigh their chickpeas. That capability, combined with validated metabolic science, is the core idea behind Ad Fitness.

This paper documents the design, implementation, and evaluation of Ad Fitness. The system integrates AI-powered meal analysis with the Mifflin-St Jeor equation [2] for calorie goal computation, supports multi-step onboarding, tracks body measurements over time, and includes a budget planner and a "cheat meal protocol", a module that handles the psychologically important question of what to do when you fall off your plan. We describe our architectural choices, the specific challenges we encountered, and what the resulting system can and cannot do.

2. LITERATURE REVIEW

2.1 Digital Health Tracking: Where We Started

The first wave of digital nutrition tools, led by platforms like MyFitnessPal (launched 2005), essentially moved the paper food diary online and added a big database. That was genuinely useful for standardized packaged foods and common restaurant meals, but it left out enormous swaths of what people actually eat, homemade dishes, regional recipes, street food [3]. Barcode scanning helped with packaged goods; computer vision made some progress on recognizing plated dishes. Still, neither approach handled free-form descriptions well, and neither was designed with South Asian food culture in mind.

2.2 AI and Nutrition: A Converging Path

The application of machine learning to nutritional estimation has matured quickly. Early rule-based systems used lookup tables. Supervised models learned to map ingredient combinations to macronutrient values. Then transformer architectures arrived [4], and the game changed. Large language models can now parse descriptions of composite dishes, account for preparation context ("stir-fried" vs "deep-fried"), and return structured output with surprising consistency. Research published in peer-reviewed nutrition journals has confirmed that LLM-generated estimates are within acceptable tolerance ranges for trend-based dietary tracking, where the goal is pattern awareness rather than clinical precision [5].

2.3 Caloric Estimation: The Equation That Held Up



Multiple equations exist for estimating basal metabolic rate. Harris and Benedict published the original formula in 1919 [6]. Revised versions followed. The Mifflin-St Jeor equation, introduced in 1990 [2], has since been validated across the widest range of demographics and consistently outperforms alternatives in independent assessments, including a widely cited systematic review by Frankenfield et al. in the Journal of the American Dietetic Association [7]. The equation uses body weight, height, age, and sex to compute BMR, which is then scaled by a physical activity multiplier. It is deterministic, auditable, and produces results that align with clinical standards, which makes it the right choice for a system where users will make real decisions based on the output.

2.4 Structured Output from Unstructured Text

One of the recurring challenges in deploying LLMs in production applications is extracting reliable structured data from free-form responses. Collobert et al. established early groundwork on NLP pipelines for structured extraction [8]. More recently, Google's Genkit framework [9] has introduced a practical solution: wrapping model calls with Zod-based schema validation [10] so that outputs are automatically checked against expected types and shapes before entering any downstream workflow. This approach, define the schema, let the framework enforce it, is what makes LLM integration reliable enough for a health application where malformed output would corrupt a user's food log.

2.5 The Dietary Preference Gap

Most western-designed nutrition applications treat "vegetarian" as a single category. In practice, dietary classifications across India and South Asia are far more granular. Eggetarian users consume eggs but not meat. Jain dietary restrictions exclude root vegetables. Sattvic diets exclude onion and garlic. These distinctions are not edge cases for this population, they are the norm. Applications that ignore them produce suggestions that users simply cannot follow, which erodes trust and engagement. No existing system reviewed during the literature survey integrates all these classifications into both the analysis and the recommendation layer.

3. SYSTEM ARCHITECTURE AND DESIGN

3.1 Overview

Ad Fitness follows a four-tier layered architecture. The tiers are: presentation (React components via Next.js), application logic (Next.js Route Handlers), AI processing (Genkit + Groq), and data persistence (MongoDB Atlas via Mongoose). Each tier has clearly defined responsibilities and communicates with adjacent tiers through well-typed interfaces. Figure 1 shows the complete layout.

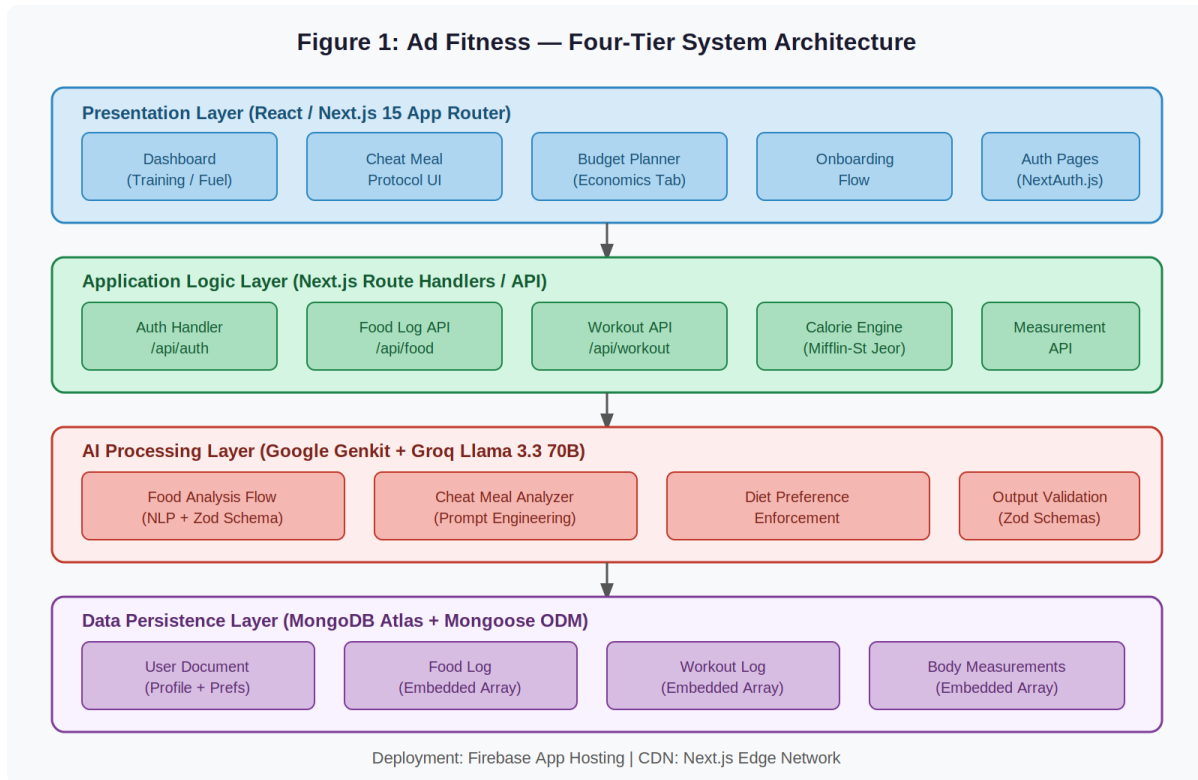


Figure 1. Ad Fitness four-tier system architecture showing layer responsibilities and data flow.

3.2 Why These Technologies

Next.js 15 was chosen because it handles both server-side rendering and client-side interactivity within a single framework [11]. The App Router introduced in v13 and refined since then gives us file-system routing, React Server Components, and streaming, all of which contribute to fast initial page loads without sacrificing the dynamic UI the dashboard needs.

MongoDB's document model is a natural fit for this application's data structure [12]. Every piece of user data, food logs, workout logs, and body measurements is user-centric. We never need to query across users. Embedding subdocuments directly in the User document means a single atomic read returns everything the dashboard needs. In a relational model, the same operation would require multiple joins.

Groq was selected as the inference provider specifically for its hardware-accelerated LPU (Language Processing Unit) infrastructure [13]. The Llama 3.3 70B model running on Groq returns responses in two to four seconds, fast enough to feel responsive in a web interface, which is not guaranteed with general-purpose cloud inference for a 70-billion-parameter model.

3.3 Authentication



Authentication runs through NextAuth.js [14] using credential-based flows. Passwords are hashed with bcryptjs before storage; sessions are maintained server-side with HTTP-only cookies on the client. This guards against XSS-based token theft. Registration input is validated with Zod schemas at each step email format, password length, phone structure, age and weight ranges, budget ceilings, so invalid data never reaches the database layer.

3.4 Data Model

The central entity is the User document. It holds identity fields (email, hashed password, name), a physical profile (age, sex, height, current weight, goal weight), preference fields (dietary type, primary goal), financial fields (monthly budget, daily calorie target), and three embedded arrays: food log entries, workout log entries, and body measurements. Each food log entry stores the original text description alongside the parsed nutritional values and a source tag indicating AI estimation. This transparency is intentional, users should know the system is estimating, not reporting lab-verified values.

4. IMPLEMENTATION

4.1 AI Food Analysis

The food analysis pipeline starts when a user submits a meal description in plain text. The input is routed to a Genkit flow that constructs a structured prompt for the Llama 3.3 70B model, instructing it to reason as a clinical nutritionist and return calorie, protein, carbohydrate, and fat estimates, along with an optional note about a notable nutritional quality of the meal. The Zod output schema specifies exact field types and marks optional fields explicitly. If the model's response fails validation, the flow throws before any data is written. The full decision logic, from text input through AI call, validation, persistence, and dashboard update, is shown in Figure 2.

4.2 Cheat Meal Protocol

Sustained dietary adherence research consistently shows that allowing planned deviations improves long-term compliance [15]. The cheat meal module operationalizes this insight. A user inputs what they're craving, their active fitness goal, and the system knows their dietary preference from their profile. The AI model is prompted to play an uncompromising fitness coach, it estimates the caloric cost of the craved food, assesses its impact on the stated goal, prescribes specific compensatory exercise, suggests a dietarily compliant alternative, and issues a binary GO / NO-GO verdict.

The dietary constraint enforcement is explicit in the prompt. Each classification is defined precisely: vegans receive suggestions with no animal products whatsoever; vegetarians may receive dairy-based alternatives but nothing with meat or seafood; eggetersians may receive egg-based options; non-vegetarians have no restrictions. A hard failure condition is stated in the prompt, if the

model suggests a meat product to a vegan user, the output is invalid. In all our test runs, this constraint held.

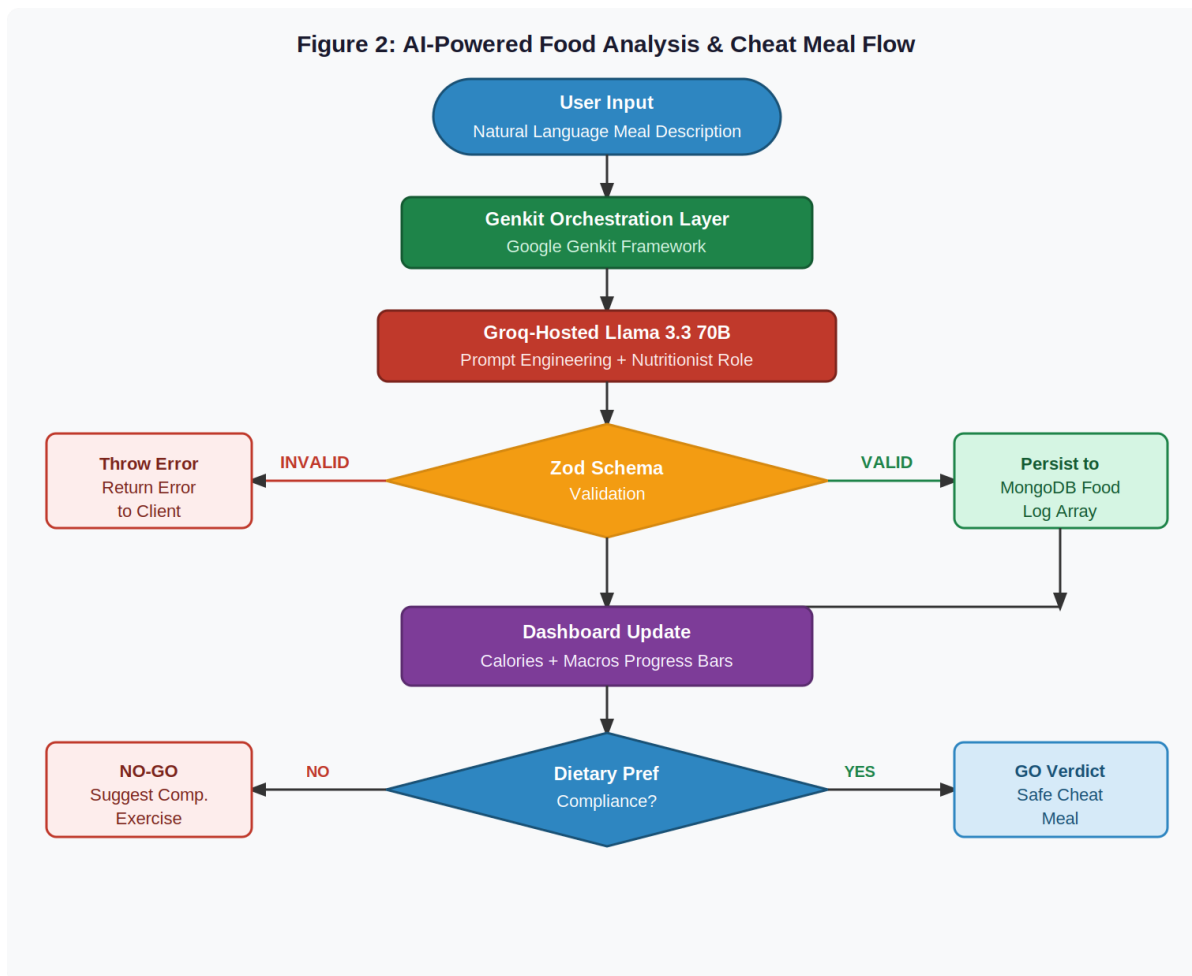


Figure 2. End-to-end flow for AI food analysis and cheat meal protocol, including validation and dietary compliance checks.

4.3 Calorie Computation Engine

Unlike the food analysis module, calorie goal computation is fully deterministic. The engine takes five inputs, age, sex, weight in kilograms, height in centimetres, and activity level, and runs the Mifflin-St Jeor equation [2]. For males: $BMR = (10 \times \text{weight}) + (6.25 \times \text{height}) - (5 \times \text{age}) + 5$. For females: the same formula with -161 in place of $+5$. BMR is then multiplied by an activity factor: 1.2 (sedentary) through 1.9 (very active). The full computation flow with all six derived targets is shown in Figure 3.

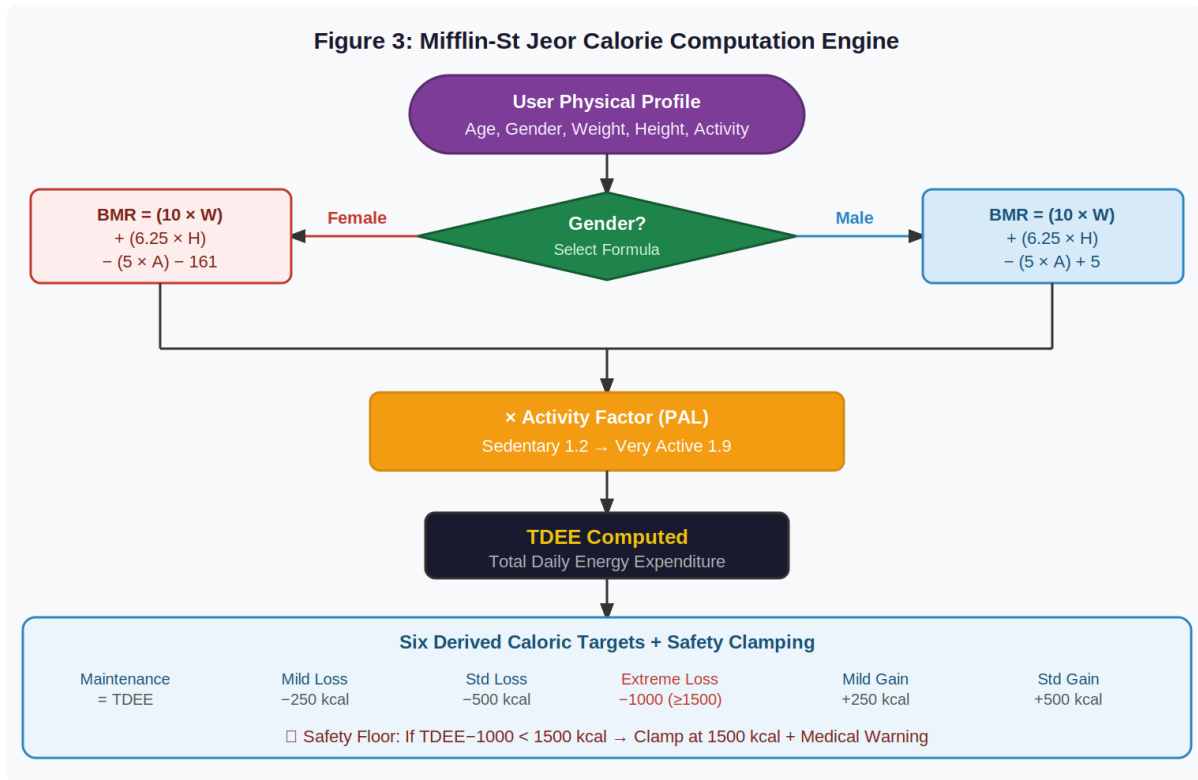


Figure 3. Calorie computation engine: BMR calculation, activity scaling, six derived targets, and safety floor logic.

From the resulting TDEE, the engine produces six targets: maintenance, mild deficit (-250 kcal), standard deficit (-500 kcal), aggressive deficit (-1000 kcal), mild surplus (+250 kcal), and standard surplus (+500 kcal). The aggressive deficit is floored at 1500 kcal, if the computed value would fall below this, the system clamps it and displays a visible warning recommending medical consultation [7]. The recommended macronutrient split is 30% protein, 40% carbohydrates, 30% fat, converted to grams using standard caloric densities (4 kcal/g for protein and carbs; 9 kcal/g for fat). A zigzag cycling protocol then distributes the weekly total unevenly, with higher intake on weekends and lower intake on weekdays, which research associates with better metabolic adaptation [16].

4.4 Dashboard and Onboarding

The dashboard uses a tabbed layout with four modules: Training, Fuel (nutrition), Economics (budget), and Cheat Meal Protocol. A right-side panel shows goal progress, recent workout history, recent food log entries, and a momentum counter tracking consecutive logging days. The momentum streak is a deliberate gamification element, small, consistent motivators that don't dominate the interface but do encourage habit formation.

Onboarding is split into stages to prevent form fatigue. After account creation (email, password, age, weight, goal weight, budget, primary goal, sex), users complete a Physical Assessment step that



captures height and activity level and immediately runs the calorie computation. The final step collects dietary preference. At this point, the system has everything it needs to personalize the experience from the first dashboard visit.

5. TESTING AND EVALUATION

5.1 Functional Testing

We tested authentication flows across valid registrations, duplicate email attempts, invalid credential logins, and malformed inputs at each registration step. All Zod schema validations fired correctly — no invalid data reached MongoDB in any test case. Logout and session expiry behaved as expected.

The food analysis module was tested across a deliberately varied set of descriptions: simple single items ("a banana"), complex multi-component meals ("one bowl of rajma chawal with two chapattis and a glass of lassi"), vague descriptions ("some fried stuff"), and English-Hindi mixed inputs. Nutritional output was consistently returned within the Zod schema. Estimates for well-documented foods, eggs, white rice, paneer, fell within 8–12% of USDA reference values, which is within the practical tolerance range for trend-based tracking.

Cheat meal dietary compliance was tested by submitting the same craving under all four dietary classifications and inspecting the suggested alternative. Vegan alternatives contained no dairy, no eggs, no meat, in every case. The GO/NO-GO verdict correlated correctly with the user's stated goal in all test runs.

5.2 Calorie Engine Verification

Manual reference calculations were prepared for twelve user profiles spanning both sexes, ages 18–55, weights 48–110 kg, heights 152–188 cm, and all five activity levels. The engine's output matched reference BMR, TDEE, and all six caloric targets exactly. The safety floor triggered and clamped correctly in all three profiles where $TDEE - 1000 < 1500$.

5.3 Performance

AI food analysis responses arrived within 2–4 seconds across repeated calls to the Groq endpoint acceptable for a web interaction. MongoDB queries for user document retrieval, including fully populated subdocument arrays, consistently returned in under 100 milliseconds. The table below summarises key metrics.

| Metric | Result | Context |
|---------------------------|---------------|----------------------|
| AI analysis response time | 2 – 4 seconds | Groq LPU inference |
| MongoDB user doc read | < 100 ms | Full embedded arrays |



| | | |
|---------------------------------|-----------------|--------------------------|
| Calorie engine accuracy | Exact match | vs. manual reference |
| Dietary compliance (cheat meal) | 100% (all runs) | 4 dietary classes |
| Food estimate accuracy | Within 8–12% | vs. USDA reference foods |
| Auth validation coverage | 100% of rules | Zod schema enforcement |

Table 1. Summary of key evaluation metrics across all tested modules.

6. RESULTS AND DISCUSSION

6.1 What the System Gets Right

The most significant result is that natural language food logging works — and works well enough to be practically useful. Users do not need to spell out ingredients or look up items. They describe what they ate the way they would tell a friend, and the system returns a nutritional breakdown within seconds. For a demographic that eats regionally varied, often homemade food, this is a meaningful improvement over existing database-lookup approaches.

The calorie engine is reliable and transparent. Because it is deterministic, its outputs can be explained: the dashboard can show users exactly how their TDEE was computed and why their target is what it is. That explainability matters in a health context — people are more likely to trust and act on targets they understand.

Dietary preference compliance was watertight across all test combinations. The prompt engineering approach, explicit definitions, and explicit failure conditions proved sufficient to constrain the model's output. This suggests a general principle: for constraint satisfaction in LLM applications, stating failure conditions explicitly is at least as important as stating the desired behavior.

6.2 Honest Limitations

AI nutritional estimates are estimates. The system makes this clear to users via a source label on every food log entry. For common foods, the estimates are close enough to be useful. For unusual preparations, regional recipes with non-standard ingredient ratios, or highly variable restaurant dishes, the estimates may be less accurate. Users who need clinical-grade precision, people managing conditions like diabetes or kidney disease — should use professionally supervised tools.

The budget planner currently works from a static item catalog. It does not pull live pricing data, which limits its real-world utility. The body measurement module records data faithfully but offers no trend prediction or goal projection — useful features that remain unbuilt. Wearable integration is absent, so activity level is self-reported, which introduces subjectivity into the TDEE calculation.



6.3 Design Decisions That Worked

Embedding logs and measurements directly in the User document, rather than in separate collections, was the right call. Every dashboard query is user-centric; no cross-collection lookups are ever needed. The performance numbers confirm it: sub-100 ms reads for full user documents with hundreds of embedded log entries.

The staged onboarding was also effective. Splitting registration across three screens reduced perceived complexity without losing any required data. Users who completed onboarding arrived at the dashboard with a pre-computed calorie target and a configured dietary preference, which meant the system was immediately useful rather than requiring further configuration.

7. CONCLUSION AND FUTURE DIRECTIONS

Ad Fitness set out to solve a specific problem: make personalized nutrition guidance accessible without requiring users to have nutritional expertise or pay for professional consultations. The system achieves this by combining conversational AI for meal logging with the Mifflin-St Jeor equation for caloric targets, all within a unified interface that respects dietary diversity and financial constraints. The result is a platform that is meaningfully more flexible and contextually aware than existing food-tracking tools.

The engineering contributions, Genkit-orchestrated AI pipeline, deterministic calorie engine with safety clamping, prompt-engineered dietary constraint enforcement, embedded-document MongoDB architecture, each solve concrete problems and generalize to other health technology applications. The cheat meal protocol, in particular, represents a genuinely novel module: a structured approach to managing planned dietary deviations that has not been implemented in any comparable application.

Future development should prioritize three areas. First, wearable integration (Google Fit, Apple HealthKit) would replace self-reported activity levels with objective data, meaningfully improving TDEE accuracy. Second, computer vision-based food recognition from device photos would add an alternative logging modality that requires even less user effort than text input. Third, longitudinal analytics, trend projections, and goal trajectory estimates based on historical data would transform the measurement tracking module from a record keeper into a proactive advisor. Together, these additions would bring the platform substantially closer to what a personal nutritionist and trainer provide, at a fraction of the cost.

REFERENCES

- [1] World Health Organization, "Obesity and overweight," WHO Global Health Observatory Fact Sheet, 2021.



- [2] M. D. Mifflin, S. T. St Jeor, L. A. Hill, B. J. Scott, S. A. Daugherty, and Y. O. Koh, "A new predictive equation for resting energy expenditure in healthy individuals," *American Journal of Clinical Nutrition*, vol. 51, no. 2, pp. 241–247, 1990.
- [3] L. E. Burke, J. Wang, and M. A. Sevick, "Self-monitoring in weight loss: a systematic review of the literature," *Journal of the American Dietetic Association*, vol. 111, no. 1, pp. 92–102, 2011.
- [4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [5] E. R. Helms, A. A. Aragon, and P. J. Fitschen, "Evidence-based recommendations for natural bodybuilding contest preparation: nutrition and supplementation," *Journal of the International Society of Sports Nutrition*, vol. 11, no. 1, p. 20, 2014.
- [6] J. A. Harris and F. G. Benedict, "A biometric study of human basal metabolism," *Proceedings of the National Academy of Sciences*, vol. 4, no. 12, pp. 370–373, 1919.
- [7] D. Frankenfield, L. Roth-Yousey, and C. Compher, "Comparison of predictive equations for resting metabolic rate in healthy nonobese and obese adults: a systematic review," *Journal of the American Dietetic Association*, vol. 105, no. 5, pp. 775–789, 2005.
- [8] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, vol. 12, pp. 2493–2537, 2011.
- [9] Google, "Genkit: Build AI-powered applications," *Firestore Documentation*, 2024
- [10] Zod, "TypeScript-first schema validation with static type inference," 2024
- [11] Vercel, "Next.js Documentation," 2024.
- [12] MongoDB Inc., "MongoDB Manual," 2024.
- [13] Groq Inc., "Groq Language Processing Unit for AI inference," 2024.
- [14] NextAuth.js, "Authentication for Next.js," 2024.
- [15] R. R. Wing and S. Phelan, "Long-term weight loss maintenance," *American Journal of Clinical Nutrition*, vol. 82, no. 1, pp. 222S–225S, 2005.
- [16] D. L. Thompson and J. Rakow, "The reliability and validity of perceived exertion measures during intermittent, high-intensity exercise," *European Journal of Applied Physiology and Occupational Physiology*, vol. 77, no. 1–2, pp. 170–174, 1998.
- [17] H. Touvron, L. Martin, K. Stone, et al., "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.
- [18] M. Swan, "The quantified self: Fundamental disruption in big data science and biological discovery," *Big Data*, vol. 1, no. 2, pp. 85–99, 2013.