IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151126

FITGUARD AI: A RULE-BASED FITNESS AND NUTRITION DECISION SUPPORT SYSTEM

Khatik Mohammad Sabique Sikandar[1], Mohit Manilal Patel[2], Bhavesh Dinesh Patil[3], Pranav Kishor Sutar[4],Prof. Kaminee Subhash Patil[5]

Student, Artificial Intelligence and Machine Learning, RCPIT, Shirpur, India[1]

Student, Artificial Intelligence and Machine Learning, RCPIT, Shirpur, India[2]

Student, Artificial Intelligence and Machine Learning, RCPIT, Shirpur, India[3]

Student, Artificial Intelligence and Machine Learning, RCPIT, Shirpur, India[4]

Assistant Professor, SES's R. C. Patel Institute of Technology, Shirpur 425405, India[5]

Abstract: Generic diet and exercise advice ignores differences in body size, daily activity, goals and food preferences, while hand-made plans are slow to prepare. This paper describes FitGuard AI, a web-based decision support prototype that converts a short user profile into a personalized calorie target, a seven-day meal plan, a weekly workout split and a screening-level risk indicator. The back end is written in Python with Flask and keeps profiles and audit entries in SQLite; the front end uses HTML, CSS and JavaScript. Energy needs are estimated with the Mifflin–St Jeor equation, scaled by an activity factor and shifted by a goal-specific adjustment, and every result is accompanied by an explanation trace that lists the formulas and rules that produced it. In a recorded example (male, 20 years, 180 cm, 54 kg, active, muscle-gain goal) the system produced a BMI of 16.7, a BMR of 1570 kcal/day, an energy expenditure of 2708 kcal/day and a target of about 2960 kcal/day. All five modules passed the functional tests of the first project stage. The recommendation path is deterministic; no trained learning model or image analysis is implemented, and no clinical validation has been done. The paper therefore presents the system as an inspectable engineering baseline, discusses its safety and privacy limits, and lists the evaluation needed before real-world use.

Keywords: Decision support, explainable recommendation, BMI, BMR, TDEE, nutrition planning, Flask, SQLite.

I. INTRODUCTION

Rising rates of obesity, diabetes and heart disease have made everyday diet and exercise planning an important public concern. Most people lack the time or access to a nutritionist, and many fitness applications return the same template to everyone. A useful tool should respond to the person's age, sex, body size, activity and goal, and it should also tell the person why a particular plan was proposed.

Recommender research for health spans knowledge graphs, collaborative filtering, adaptive nutrition frameworks and learned exercise models [3]–[7], but such approaches usually need large datasets and are hard to audit. For a first-stage academic prototype, an alternative is to build on physiological equations that are already well established and to encode the remaining choices as readable rules. This keeps behavior predictable and lets a reviewer trace each number to a formula.

FitGuard AI follows this route. It was developed as the Stage-I project of the Department of Artificial Intelligence and Machine Learning at R. C. Patel Institute of Technology, Shirpur. The contributions of this paper are: (i) a layered Flask/SQLite design that separates request handling, decision logic and storage; (ii) a documented calculation chain from validated input to calorie target; (iii) an explanation trace that exposes the inputs, formulas and rules behind every plan; (iv) a persistence model with audit records and lightweight retrieval endpoints; and (v) a candid statement of what the prototype does not do, so that its "AI" label is not read as a claim of machine learning, image analysis or clinical accuracy.

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151126

II. RELATED WORK

Body mass index (BMI) is the weight-to-squared-height ratio used by the World Health Organization to screen for underweight, overweight and obesity; it indicates a weight category, not body composition or health status [1]. For resting energy expenditure, Mifflin et al. fitted a regression on healthy adults that remains a common default because it needs only weight, height, age and sex [2]. Multiplying the result by an activity factor gives total daily energy expenditure (TDEE), and a signed calorie offset turns this into a goal-oriented target.

Research on intelligent health advice is broad. Renner et al. outline how adaptive systems could personalize nutrition advice over time [4]. Yuan et al. combine K-means clustering with collaborative filtering for diet recommendation [5], and Wang et al. add contextual information to healthy-food recommenders [8]. In fitness, Dharia et al. present the PRO-Fit assistant [7], Ekambaram and Kumar describe a machine-learning exercise recommender [6], and Liu et al. use a fitness knowledge graph with attention in R-CKGAT [9]. A review by Sadeghi et al. stresses that explanations are essential for trust when AI supports health decisions [3]. Table I summarizes the families of approach and where the present work sits.

TABLE I Families of Health Recommendation Approaches

Approach	Strength	Weakness	Relevance here
BMI screening [1]	Simple, standard categories	Ignores muscle and fat distribution	Used for categories and risk flags
Predictive equations [2]	Needs few inputs	Population average, not individual	Basis of energy estimate
Rule-based systems	Transparent, easy to test	Fixed thresholds, not adaptive	Core of plan and risk logic
Learned recommenders [5]–[9]	Adapt to user behavior	Need data, harder to audit	Future extension
Explainable AI [3]	Builds trust	Adds design effort	Explanation trace implemented

III. PROBLEM STATEMENT AND OBJECTIVES

The problem addressed is how to turn a handful of user-entered measurements and preferences into a safe, understandable fitness and nutrition plan without a specialist, while keeping an inspectable record of what was asked and what was advised.

The objectives are to accept and clean a user profile; reject impossible values early; compute BMI, BMR and TDEE; adjust the energy target for the user's goal; derive meal and workout plans matched to diet preference, goal and BMI category; raise screening flags for body-weight, blood-pressure and diabetes inputs; show the reasoning behind each result; and store profiles and audit summaries for later review.

IV. PROPOSED SYSTEM

A. Architecture

FitGuard AI is organized in four layers (Fig. 1). The browser interface collects the profile and renders the dashboard, recommendations and explanation pages. A Flask application in *app.py* receives requests and returns HTML or JSON. All calculations, rules and plan templates live in *core.py*, which has no web dependencies and can therefore be unit-tested directly. Storage functions in *db.py* manage a local SQLite file. Three read or write endpoints are exposed: POST /api/recommend, GET /api/logs and GET /api/profile-summary.

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151126

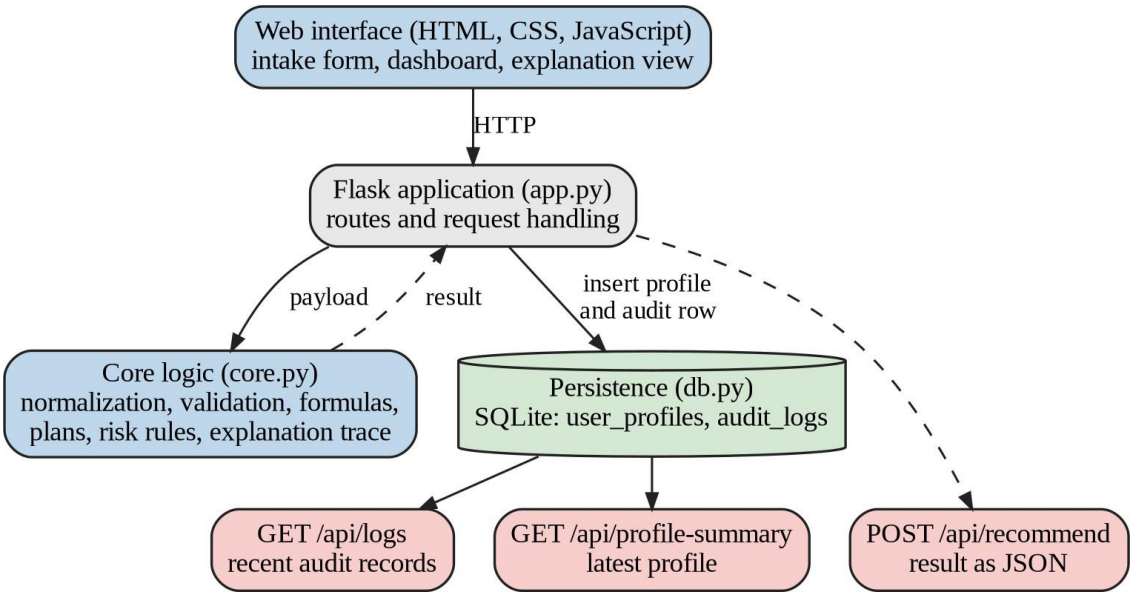


Fig. 1. Layered architecture and API endpoints of FitGuard AI

B. Request Workflow

Fig. 2 shows what happens when a profile is submitted. The payload is normalized and then validated; any violation ends the request with an error list and HTTP status 400. A valid payload passes through the energy estimation, the risk and diet rules and the plan generators. An explanation trace is assembled, the profile and a compact audit summary are written in one database transaction, and the combined result is returned to the interface as JSON.

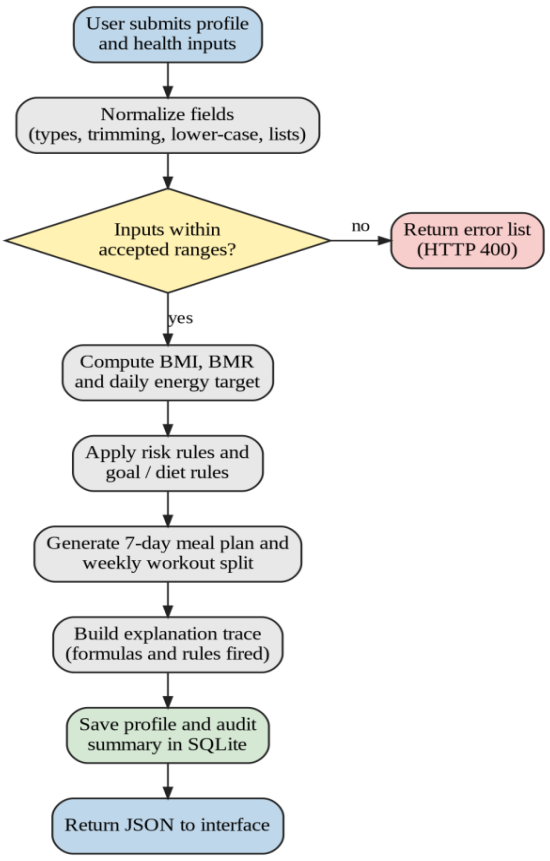


Fig. 2. Flowchart of the recommendation request

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151126

C. *Input Normalization and Validation*

Normalization converts numeric fields to numbers, trims text, lower-cases categorical values such as sex, activity level, goal and diet preference, and splits comma-separated entries for meal-photo, body-photo and report labels into lists. Validation then checks that age lies between 10 and 100 years, height between 100 and 230 cm and weight between 25 and 250 kg, and that sex, activity level, goal and diet preference are among the accepted options (Table II). These checks stop malformed requests, but they cannot confirm that a value is truthful or medically plausible for a given person.

TABLE II Accepted Input Ranges and Options

Field	Accepted values	Note
Age	10–100 years	Numeric
Height	100–230 cm	Numeric
Weight	25–250 kg	Numeric
Activity level	Sedentary, light, moderate, active, very active	Selects activity factor
Goal	Fat loss, maintenance, muscle gain	Selects calorie offset
Diet preference	Vegetarian, vegan, eggs, meat/omnivore	Selects meal templates

D. *Energy Estimation*

The calculation chain is drawn in Fig. 3. BMI is computed from the normalized weight W (kg) and height H (cm):

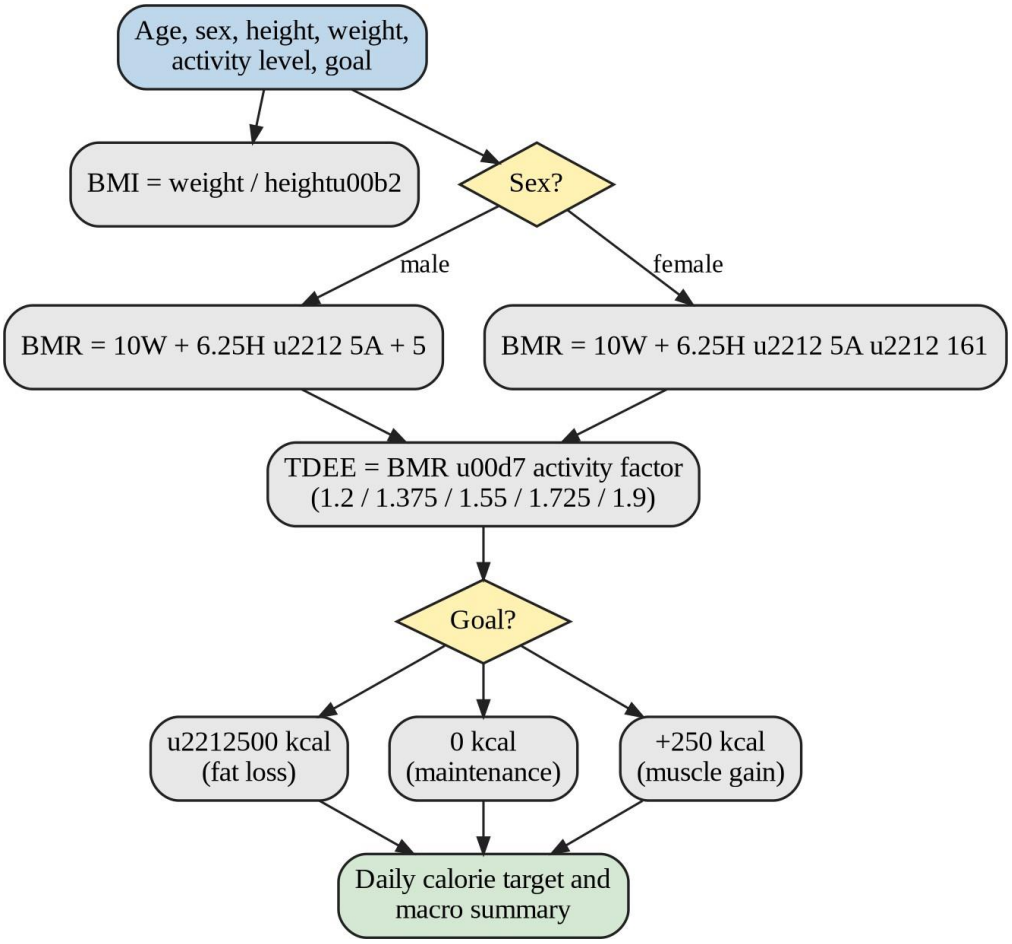


Fig. 3. Flowchart of BMI, BMR, TDEE and goal-adjusted calorie target $\text{BMI} = W / (H / 100)^2$ (1)

Resting energy expenditure uses the Mifflin–St Jeor equation [2], where A is age in years:

$$\text{BMR}_m = 10W + 6.25H - 5A + 5, \qquad \text{BMR}^f = 10W + 6.25H - 5A - 161 \quad (2)$$

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151126

where the subscripts denote male and female. TDEE is the product of BMR and the activity factor f, and the final target adds the goal offset Δg:

$$\text{TDEE} = \text{BMR} \times f, \qquad \text{Target} = \text{TDEE} + \Delta g \quad (3)$$

The factors are 1.2 (sedentary), 1.375 (light), 1.55 (moderate), 1.725 (active) and 1.9 (very active); the offsets are −500 kcal for fat loss, 0 for maintenance and +250 kcal for muscle gain (Table III). A macronutrient summary (protein, fat and carbohydrate in grams) is displayed with the target.

TABLE III Activity Factors and Goal Offsets

Activity level	Factor f	Goal	Offset Δg (kcal)
Sedentary	1.2	Fat loss	−500
Light	1.375	Maintenance	0
Moderate	1.55	Muscle gain	+250
Active	1.725		
Very active	1.9		

E. Meal and Workout Planning

Meal plans are built from template meals chosen by diet preference (vegetarian, vegan, vegetarian with eggs, or meat/omnivore) and then worded for the goal, for example a calorie-reduction or protein-emphasis note. The interface presents a seven-day table with breakfast, lunch, dinner and snack. Workout plans branch on BMI and goal: a lighter routine is returned when BMI is below 18.5, and different weekly structures are used for fat loss and muscle gain. Plans are templates and not individual prescriptions.

F. Risk Screening

A deterministic risk engine (Fig. 4) maps selected inputs to a risk level. The rules consider a very low or underweight BMI, a BMI of 30 or above, an elevated blood-pressure entry and a diabetes entry. The resulting level, such as "Moderate", is shown on the dashboard, influences the safety wording of the plan and is stored in the audit table. The rules are screening flags only; they are not a validated risk prediction.

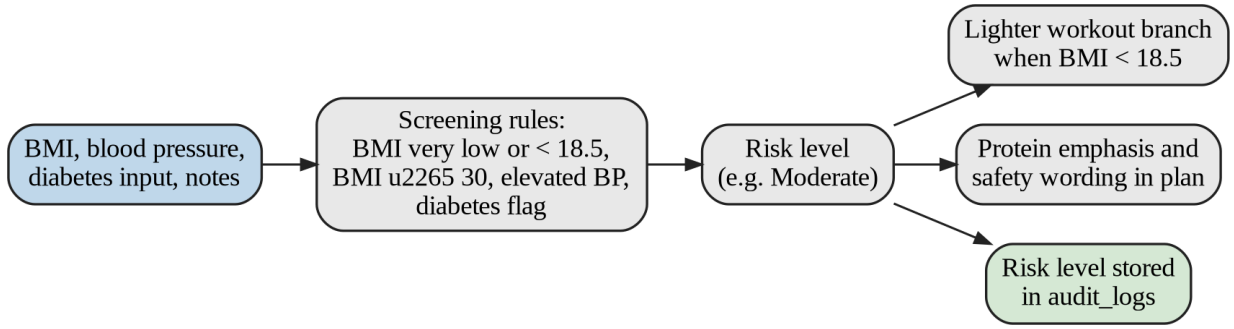


Fig. 4. Risk screening and its influence on the plan

G. Explanation Trace

To support transparency, each response carries a trace that lists the BMI computation, the BMR equation, the TDEE multiplication, the goal adjustment and the final target, together with the rules that fired (goal rule, risk rule, diet rule and any protein-emphasis rule). The explainability page displays this trace and a history log of past decisions, so a user or examiner can recompute every figure by hand.

H. Data Model and Persistence

Two SQLite tables are used (Table IV). *user_profiles* stores the timestamp, name, age, sex, height, weight, activity level, goal, diet preference, allergies, blood-pressure and diabetes entries, other conditions, symptom notes and the serialized photo and report label lists. *audit_logs* stores the timestamp, user name, risk level, a short summary and the JSON payload. Foreign-key enforcement is switched on for each connection. The profile and audit rows of one request are committed together, so the two tables stay consistent.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151126

TABLE IV Persistence and API Summary

Item	Content / behavior
user_profiles	Demographics, measurements, goal, diet, health entries, label lists
audit_logs	Timestamp, user, risk level, summary, JSON payload
POST /api/recommend	Validate, compute, store, return profile and result (400 on error)
GET /api/logs	Recent audit records, newest first, limit parameter
GET /api/profile-summary	Latest saved profile, or null when none exists

V. IMPLEMENTATION

The prototype is written in Python with Flask for routing and templates, SQLite for storage, and HTML, CSS and JavaScript for the dark-themed dashboard, which has pages for intake, recommendations, explainability and history. The reference machine in the Stage-I report was a laptop or desktop with an Intel Core i3/i5 or better processor, 4 GB RAM and 50 GB of free disk running Windows 10/11. The source consists of three modules (*app.py*, *core.py*, *db.py*), a SQLite database file and the *templates* and *static* folders. The intake form also accepts comma-separated labels for meal photos, body photos and medical reports, but these are stored as text only. Table V lists the stack.

TABLE V Software Stack

Component	Technology	Role
Back end	Python, Flask	Routing, request handling, JSON/HTML responses
Decision logic	Python (core.py)	Normalization, validation, formulas, rules, plans
Database	SQLite (db.py)	Profiles and audit records
Front end	HTML, CSS, JavaScript	Forms, dashboard, plan and explanation views
Environment	Windows 10/11, VS Code	Development and execution

VI. RESULTS AND DISCUSSION

A. Worked Example

The prototype was run with the profile of a 20-year-old active male, 180 cm tall and weighing 54 kg, with a muscle-gain goal and a meat/omnivore diet. Table VI reproduces the values displayed by the explainability page and **recomputes** them by hand. BMI is $54/1.8^2 = 16.7$, which falls in the underweight band. BMR is $10(54) + 6.25(180) - 5(20) + 5 = 1570$ kcal/day, TDEE is $1570 \times 1.725 \approx 2708$ kcal/day, and adding 250 kcal gives about 2958 kcal/day, shown as 2960 kcal. The macro summary of 103 g protein, 82 g fat and 452 g carbohydrate corresponds to roughly 2958 kcal when using 4, 9 and 4 kcal per gram, so it is consistent with the target. The risk level was Moderate, and the weekly split contained light full-body sessions, a mobility day and a recovery day, in line with the lighter branch used below a BMI of 18.5.'

TABLE VI Worked Example from the Prototype

Quantity	Value displayed	Hand check
BMI	16.7	$54 / 1.80^2 = 16.7$
BMR (kcal/day)	1570	$540 + 1125 - 100 + 5$
TDEE (kcal/day)	2708	1570×1.725
Goal offset	+250 kcal	Muscle gain
Calorie target	2960 kcal	≈ 2958, rounded
Macros P / F / C (g)	103 / 82 / 452	$412 + 738 + 1808 \approx 2958$ kcal
Risk level	Moderate	Rule-based

B. Module Testing

Each module was tested against an expected behavior and then the integrated application was run end to end. All five test cases passed (Table VII). These are functional checks: they show that the intended path works, not that the advice is clinically correct.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151126

TABLE VII Functional Test Results

Module	Expected behavior	Status
User interface	Data entry works	Passed
Input validation	Invalid data rejected	Passed
Recommendation engine	Personalized plan generated	Passed
Explainability	Reasoning displayed clearly	Passed
Database	Data stored and retrieved	Passed

C. Discussion

Because every number can be reproduced from three equations and two lookup tables, reviewers can verify the output quickly, which is the main benefit of the rule-based design. The worked example also shows why transparency matters: a very low BMI together with a muscle-gain goal yields a high calorie target that a professional might want to review, and the trace makes this visible instead of hiding it. At the same time, the example illustrates the weakness of fixed thresholds, since the risk level alone does not capture the person's full situation.

VII. LIMITATIONS, SAFETY AND ETHICS

The Stage-I report describes the system as multimodal and as containing a lightweight machine-learning component. The inspected implementation does not support either claim: photo and report fields are stored as text labels, no vision or document model is present, and recommendations come from formulas and rules. This paper accordingly describes the system as a rule-based decision support prototype, and any "multimodal" or "learning" wording should be used only after such components are built and evaluated.

Other limits follow. Equations and thresholds give population-level estimates and are not suitable for minors, pregnancy, chronic illness or eating-disorder risk without professional review. No benchmark, user study or expert review has been run, so no accuracy or outcome figures are reported. The database holds sensitive health entries in a local file without authentication, consent handling or retention rules. The application entry point runs Flask in debug mode, which must be disabled for any deployment. Outputs must be presented with a clear non-diagnostic disclaimer.

VIII. CONCLUSION AND FUTURE SCOPE

FitGuard AI demonstrates a compact, layered way to turn a user profile into an energy target, meal and workout plans, a screening-level risk indicator and a human-readable explanation, with all records kept in SQLite. The worked example and the module tests show that the pipeline functions as designed, and its transparency makes it a suitable baseline for further work.

Future work should add automated unit and boundary tests for every formula, goal branch and validation range; user authentication, consent and secure deployment; review of meal and exercise templates by qualified professionals; and conservative escalation messages for symptoms and known conditions. If image-based meal or posture analysis is intended, a defined vision pipeline should be built and tested on consented data, and any learned component should be compared against the present rule-based baseline using appropriate metrics and user studies.

ACKNOWLEDGMENT

The authors thank their guide, Prof. Kaminee Patil, the Head of the Department, Prof. Dr. U. M. Patil, the Director, Prof. Dr. J. B. Patil, and the Project Stage-I Coordinator, Prof. P. D. Lanjewar, of R. C. Patel Institute of Technology, Shirpur, for their guidance and support.

REFERENCES

[1] World Health Organization, "Body mass index (BMI)." Available: https://www.who.int/

[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," *Am. J. Clin. Nutr.*, vol. 51, no. 2, pp. 241–247, 1990.

[3] Z. Sadeghi et al., "A review of explainable artificial intelligence in healthcare," *Computers and Electrical Engineering*, 2024.



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.151126

[4] B. Renner et al., "Perspective: A conceptual framework for adaptive personalized nutrition advice systems," *Advances in Nutrition*, 2023.
[5] Z. Yuan et al., "Personalized diet recommendation based on K-means and collaborative filtering," *J. Physics: Conference Series*, 2019.
[6] L. Ekambaram and S. Nithish Kumar, "A machine learning-based intelligent fitness exercise recommendation system," 2024.
[7] S. Dharia et al., "PRO-Fit: A personalized fitness assistant framework," 2016.
[8] J. Wang et al., "Improving healthy food recommender systems through contextual information," 2024.
[9] Z. Liu et al., "R-CKGAT: A recommendation algorithm based on scientific fitness knowledge graph," *Scientific Reports*, 2025.
[10] Pallets Projects, "Flask documentation." Available: https://flask.palletsprojects.com/
[11] SQLite, "SQLite documentation." Available: https://www.sqlite.org/docs.html
[12] FitGuard AI source code repository. Available: https://github.com/Mohammad-sabique/FitGuard-Ai

 This work is licensed under a Creative Commons Attribution 4.0 International License