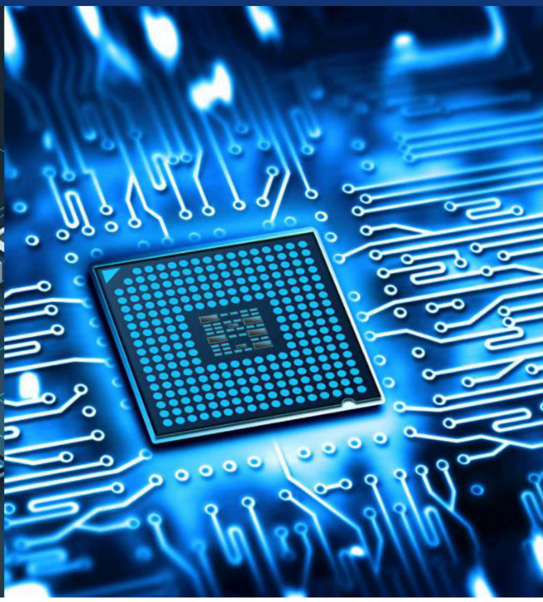
IJIRCCE

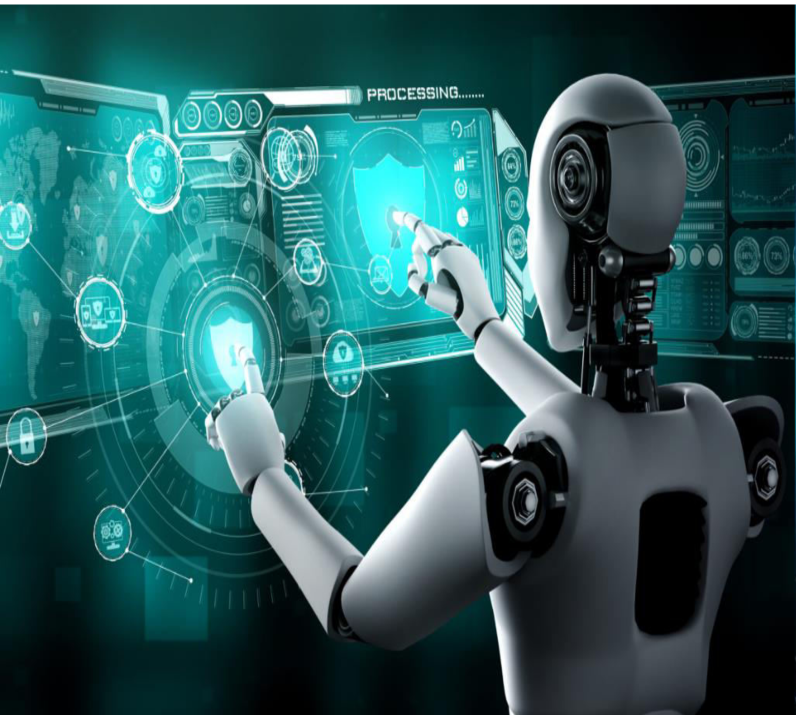
ISSN(O): 2320-9801

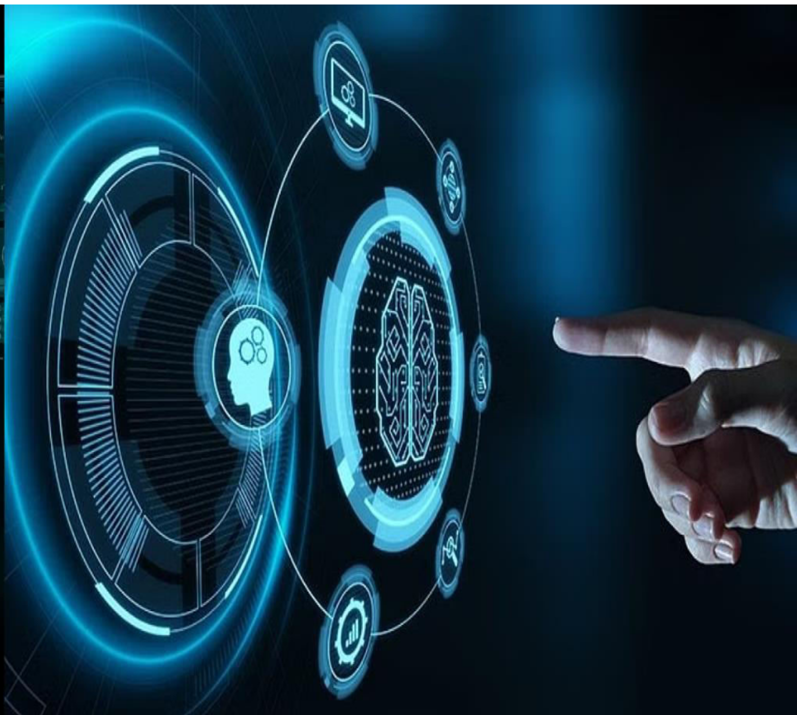
ISSN(P): 2320-9798



International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)





Impact Factor: 8.771

Volume 14, Issue 3, March 2026

© 2026 IJIRCCE | Volume 14, Issue 3, March 2026 DOI: 10.15680/IJIRCCE.2026.1403055

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

AI-Powered Smart Dietary Management and Personalized Recipe Generation System

K. Mokshith Rao[1], K. Avinash[2], G. Ashish[3], E. Satish Babu[4]

UG Students, Dept. of Computer Science & Engineering, Jyothishmathi Institute of Technology & Science (Autonomous), Karimnagar, Telangana, India[1,2,3]

Assistant Professor (Guide), Dept. of Computer Science & Engineering, Jyothishmathi Institute of Technology & Science (Autonomous), Karimnagar, Telangana, India[4]

ABSTRACT: This paper presents an AI-Powered Smart Dietary Management and Personalized Recipe Generation System a comprehensive cross-platform application that unifies meal planning, nutrition tracking, grocery management, and AI recipe generation within a single interface. The system integrates Google Gemini 2.5 Flash for real-time personalized recipe generation conditioned on user-supplied ingredients, dietary restrictions, allergies, cuisine preferences, and nutritional goals computed via the Mifflin-St Jeor equation. Core modules include a calendar-based weekly meal planner with AI auto-fill, an atomic grocery list synchronizer, a real-time health dashboard providing BMI tracking and macronutrient analysis, native push notifications, and Free/Pro/Annual subscription tiers. Built on React 19 with TypeScript, Node.js with Express, Supabase PostgreSQL with row-level security, and Capacitor for Android and iOS, the platform delivers a native-quality cross-platform experience. User acceptance testing with 25 participants over two weeks yielded an overall satisfaction rating of 4.6/5.0, with 95 percent of recipe generations completing within the 5-minute target window and average API response times of 180 ms under normal load.

KEYWORDS: Artificial Intelligence; dietary management; large language model; meal planning; personalized nutrition; Google Gemini; cross-platform; recipe generation; Supabase; React

I. INTRODUCTION

Dietary management and meal planning are critical components of a healthy lifestyle yet remain persistently time-consuming and cognitively demanding. The World Health Organization estimates that unhealthy diets contribute to approximately 11 million preventable deaths annually. Busy professionals, students, and families routinely struggle to balance macronutrient targets, decide what to eat, and coordinate grocery procurement — all under severe time constraints. Poor decisions stemming from decision fatigue and limited nutritional literacy drive obesity, type 2 diabetes, and cardiovascular disease at scale. In most cases the barrier is not intention but the absence of integrated, intelligent tools.

Existing solutions address these problems only partially. Recipe platforms such as Yummly and Tasty offer static databases requiring manual browsing. Nutrition trackers such as MyFitnessPal enable meal logging but produce no recipes and do not synchronize grocery lists. Meal planning tools typically lack AI-driven generation and real-time nutritional analysis. The resulting fragmentation forces users to maintain multiple disconnected applications, substantially increasing cognitive overhead and the likelihood of abandoning healthy eating goals. Commercial platforms such as FitGenie and Spoonacular address portions of the problem but none provides a unified end-to-end solution covering recipe generation, meal scheduling, grocery management, health analytics, and native mobile delivery.

This paper presents the AI-Powered Smart Dietary Management and Personalized Recipe Generation System (the System), integrating all these functions within a single cross-platform application. Key contributions are: (i) a structured LLM prompting pipeline for ingredient-conditioned recipe generation using Google Gemini 2.5 Flash; (ii) an atomic grocery synchronization mechanism preserving user customizations; (iii) a scientifically grounded BMR/TDEE nutrition goal computation module; (iv) a subscription-gated multi-tier architecture; and (v) empirical validation through user acceptance and performance testing conducted in 2026.

© 2026 IJIRCCE | Volume 14, Issue 3, March 2026 DOI: 10.15680/IJIRCCE.2026.1403055

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

II. LITERATURE REVIEW

RecipeGPT [1] pioneered generative pre-training transformer models for cooking recipe generation. RecipeNLG [2] established a large-scale semi-structured dataset of 2.2 million recipes and quality benchmarks. Most directly relevant, RecipeBuild [3] demonstrated superior recipe completion over general-purpose LLMs including Phi-2, LLaMA2, LLaMA3, and Mistral using a fine-tuned BERT-based Masked Language Model with heterogeneous self-attention — directly informing the structured JSON prompting strategy used in the proposed System. Recipe1M+ [4] extended cross-modal recipe understanding by pairing over one million recipes with 13 million food images for ingredient-level embedding learning. Reciptor [9] and RecipeBowl [10] further contributed Set Transformer-based hierarchical recipe encoding and ingredient recommendation approaches.

A randomized controlled trial by Celis-Morales et al. [5] demonstrated that personalized dietary recommendations based on individual phenotypic profiles achieve significantly better adherence than generic guidelines. The UNEP Food Waste Index [6] identifies poor meal planning as a primary contributor to household food waste of 30–40 percent. Research on mobile health [7] consistently identifies mobile-first design, push notifications, and personalized progress feedback as the strongest predictors of sustained user engagement. Chain-of-thought prompting [8] substantially improves LLM output quality and coherence, directly informing the System's prompting strategy. Table I summarizes the comparative landscape of existing dietary management platforms.

Table I. Comparative Analysis of Dietary Management Platforms

Feature	Yummly	MyFitnessPal	FitGenie	Spoonacular	Proposed System
AI Recipe Generation	✘	✘	✘	Partial	**✔ Gemini 2.5 Flash**
Weekly Meal Planner + AI Auto-Fill	✘	✘	✔	✘	**✔ Calendar + AI Auto-Fill**
Grocery Auto-Sync	✘	✘	✘	✘	**✔ Atomic sync**
BMR/TDEE Nutrition Tracking	✘	✔	✔	✔	**✔ BMR/TDEE personalised**
Native Cross-Platform App	✘	✔	✘	✘	**✔ Capacitor Android/iOS**
Subscription Tier Management	✔	✔	✔	✔	**✔ Free / Pro / Annual**

III. PROPOSED SYSTEM

The System is organized into eight tightly integrated subsystems. The**AI Recipe Generation Module** uses Google Gemini 2.5 Flash with structured JSON prompting conditioned on available ingredients, dietary restrictions (vegetarian, vegan, gluten-free, keto, paleo), allergies, cuisine preference, meal type, and individualized calorie and macro targets. Each recipe includes title, description, ingredient list with quantities, step-by-step instructions with optional timers, a complete nutrition breakdown, cooking time, difficulty level, serving count, and storage tips. A real-time progress tracker displays contextual messages throughout the up-to-5-minute generation window. The **Meal Planning Module** provides a calendar-based weekly planner with breakfast, lunch, dinner, and snack slots supporting manual entry, recipe-from-history selection, and an AI Auto-Fill Week feature that generates balanced meals while excluding recently used ingredients to ensure dietary variety and minimize food waste.

The**Smart Grocery List Module** performs atomic synchronization: on each meal plan update, all meal-sourced items are replaced while custom user entries are preserved, preventing data loss during sync operations. Items are grouped by food category, displayed with quantity and unit information, and support check-off, quantity editing, and search. The **Health Dashboard Module** provides real-time aggregation of daily calorie and macronutrient intake from all planned meals, BMI calculation with health range indicators, timestamped weight history with trend visualization, streak counters, and colour-coded goal deviation indicators (green for within 10 percent deviation, yellow for 10–20 percent, red for over 20 percent). The **Notification System** delivers native daily reminders at 08:00, 12:00, and 18:00 via Capacitor Local Notifications, persisting across application restarts through the native OS scheduler.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

IV. ARCHITECTURE AND IMPLEMENTATION

The System architecture follows a five-layer modular design: (1)**Client Layer** — web browsers and Capacitor-wrapped Android and iOS apps; (2) **Frontend Layer** — React 19 with TypeScript and Vite, Tailwind CSS, Axios REST, and Supabase Realtime channels; (3) **Backend Layer** — Node.js with Express 4.18.2 and TypeScript 5.3.2, housing JWT authentication middleware, plan-limit enforcement middleware, Gemini Service, Meal Plan Auto-Fill Service, Image Scraper Service, Recipe Transformer, and a node-cron Scheduler; (4) **Data Layer** — Supabase PostgreSQL (8 tables, row-level security), Capacitor Preferences for offline caching, and localStorage with a 24-hour TTL; and (5) **External Services Layer** — Google Gemini 2.5 Flash API, Pixabay API, and Google Play Billing via RTDN webhooks. Table II details the full technology stack.

Table II. System Technology Stack

Layer	Technology	Version / Details
Frontend	React 19 + TypeScript + Vite + Tailwind CSS	React 19.1.0, TypeScript 5.8.3, Vite 6.3.5
Backend	Node.js + Express 4.18.2 + TypeScript 5.3.2	Zod 3.22.4, Helmet 8.1.0, express-rate-limit
Database	Supabase PostgreSQL + Realtime + Auth + RLS	supabase-js 2.39.0, 8 core tables, JWT sessions
AI Service	Google Gemini 2.5 Flash	@google/generative-ai 0.4.0, JSON mode, 4-min timeout
Mobile	Capacitor Core 7.4.2 (Android + iOS)	LocalNotifications, Preferences, Network, StatusBar
Image Sourcing	Pixabay API + JSDOM 24.0.0 scraper	30-second timeout; placeholder fallback on failure
Deployment	Railway / Render (backend) + Vercel Edge CDN (frontend)	node-cron scheduler; Google Play Billing RTDN webhook

The recipe generation service uses the @google/generative-ai SDK to invoke the gemini-2.5-flash model with responseMimeType set to application/json, ensuring structured output without post-processing cleanup. The prompt engineering pipeline constructs a complete JSON-formatted system prompt specifying role, ingredient constraints, dietary and allergy restrictions, cuisine, meal type, and explicit calorie and macro targets. The Gemini response is validated against a Zod 3.22.4 schema enforcing all required fields. Nutrition goal computation uses the Mifflin-St Jeor equation: $BMR = (10 \times weight_{kg}) + (6.25 \times height_{cm}) - (5 \times age) + G$, where $G = +5$ for males and -161 for females. $TDEE = BMR \times A + \Delta$, where A is the activity multiplier (1.2 sedentary to 1.9 extra active) and Δ adjusts for goal (−500 kcal for weight loss, 0 for maintenance, +500 kcal for gain). Macro targets are distributed as 30 percent protein, 45 percent carbohydrates, and 25 percent fat by caloric contribution. The Supabase PostgreSQL schema comprises eight core tables with row-level security: users, recipe_history, saved_recipes, meal_planner, user_nutrition_goals, weight_history, subscriptions, and ai_generation_log. Row-level security policies enforce complete user data isolation. Offline support is provided via Capacitor Preferences for recipe and meal plan caching and localStorage for recipe images with a 24-hour TTL.

V. TESTING AND RESULTS

The System was subjected to a four-phase testing programme covering unit testing, integration testing, performance load testing, and user acceptance testing. Unit testing validated 11 individual components, all of which passed. Table III presents the complete unit test case results covering the most critical system functions.

Table III. Unit Test Case Results

TC	Test Case Description	Expected Result	Status
01	User login with valid Supabase credentials	Session token issued	✔ **Pass**

© 2026 IJIRCCE | Volume 14, Issue 3, March 2026 DOI: 10.15680/IJIRCCE.2026.1403055

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

TC	Test Case Description	Expected Result	Status
02	Recipe generation with valid ingredients and dietary preferences	Complete recipe JSON	✔ **Pass**
03	Recipe generation with daily AI limit exceeded	Error 1500 returned	✔ **Pass**
04	Meal plan addition to breakfast slot	Supabase entry created	✔ **Pass**
05	Grocery sync after meal plan change (custom items preserved)	Atomic sync confirmed	✔ **Pass**
06	BMR/TDEE computation: male, 75 kg, 175 cm, age 28, moderate	BMR 1752.5, TDEE 2716.4	✔ **Pass**
07	AI auto-fill week with 3 empty days (9 meal slots)	9 slots filled	✔ **Pass**
08	Default notification scheduling (08:00, 12:00, 18:00)	3 notifications active	✔ **Pass**
09	Invalid input (empty ingredients) submitted to generation endpoint	Zod validation error	✔ **Pass**
10	Save recipe to favorites (INSERT saved_recipes)	Entry inserted correctly	✔ **Pass**
11	Image scraper 30-second timeout — placeholder fallback	Placeholder; generation continues	✔ **Pass**

Integration testing verified correct interaction between the React frontend and Express backend REST APIs under JWT authorization, end-to-end Gemini service invocation through the recipe transformer and image scraper, meal-plan-to-grocery synchronization, Supabase Realtime propagation to the dashboard, Capacitor notification scheduling triggered by application lifecycle events, and subscription limit enforcement across both direct generation and auto-fill week flows. Performance load testing yielded average API response times of 180 ms at 100 concurrent users, 420 ms at 500 concurrent users, and 780 ms at 1,000 concurrent users. AI recipe generation averaged 2.5 minutes end-to-end with a 99th percentile of 4.5 minutes and a timeout rate of 2 percent, confirming consistent operation within the 5-minute target. Supabase indexed query response averaged 15 ms and Realtime channel propagation latency averaged 320 ms. Table IV summarizes user acceptance testing results.

Table IV. User Acceptance Testing Results (n = 25, 2 weeks, 2026)

Metric	Score / Value	Scale / Unit
Recipe generation satisfaction	**4.6**	/ 5.0 (Likert)
Meal planning usefulness	**4.8**	/ 5.0 (Likert)
Grocery list utility	**4.5**	/ 5.0 (Likert)
Nutrition tracking accuracy	**4.7**	/ 5.0 (Likert)
Overall user satisfaction	**4.6**	**/ 5.0 (Likert)**
Average app engagement	**4.2**	sessions / week / user
Recipe generations within 5-minute window	**95%**	Success rate
Average AI generation time (end-to-end)	**2.5 min (avg), 4.5 min (P99)**	minutes
API response time @ 100 concurrent users	**180 ms**	avg. response time
Supabase Realtime propagation latency	**320 ms**	avg. channel propagation
Compatibility (Android 11–14, iOS 15–17, Chrome/Firefox/Edge/Safari)	**100%**	UI consistency

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

VI. CONCLUSION AND FUTURE WORK

This paper has presented the design, implementation, and empirical evaluation of an AI-Powered Smart Dietary Management and Personalized Recipe Generation System integrating Google Gemini 2.5 Flash LLM-based recipe generation, calendar-based meal planning with intelligent AI auto-fill, atomic grocery list synchronization, real-time BMR/TDEE-grounded nutrition tracking, native push notifications, and multi-tier subscription management into a single cross-platform application deployed on Android, iOS, and web in 2026. All 11 unit test cases passed. User acceptance testing yielded an overall satisfaction rating of 4.6/5.0 with 95 percent of AI recipe generations completing within the 5-minute target, demonstrating that the System effectively resolves the fragmentation and cognitive overhead of current dietary management workflows. The System demonstrates how modern large language models, real-time database synchronization, nutrition science, and mobile-first engineering can be unified to reduce meal planning effort, minimize food waste, and promote sustained healthy eating habits.

Future work will focus on seven directions: (i) voice command integration for hands-free recipe navigation during cooking; (ii) camera-based barcode scanning for packaged food nutritional data extraction; (iii) wearable fitness device integration (Apple Health, Google Fit, Garmin) for real-time TDEE adjustment; (iv) grocery delivery API integration enabling one-click ordering; (v) predictive meal analytics using machine learning on historical usage patterns and seasonal ingredient availability; (vi) clinical dietary management support including electronic health record integration for medically prescribed dietary plans; and (vii) a transition from the monolithic Express backend to a microservices architecture for independent scalability across functional domains at large-deployment scale.

REFERENCES

1. H. H. Lee et al., “RecipeGPT: Generative Pre-training Based Cooking Recipe Generation,” in Companion Proc. Web Conference 2020, pp. 181–184.
2. M. Bien et al., “RecipeNLG: A Cooking Recipes Dataset for Semi-Structured Text Generation,” in Proc. 13th Int. Conf. Natural Language Generation, 2020, pp. 22–28.
3. D. Choi, J. Hwang, M. Gim, and D. Park, “RecipeBuild: Learning to Build Recipe Ideas Using LM-Based Recipe Contextual Modeling,” IEEE Access, vol. 13, pp. 9788–9802, 2025.
4. J. Marin et al., “Recipe1M+: A Dataset for Learning Cross-Modal Embeddings for Cooking Recipes,” IEEE TPAMI, vol. 43, no. 1, pp. 187–203, 2021.
5. C. A. Celis-Morales et al., “Effect of Personalized Nutrition on Health-Related Behaviour Change,” Int. J. Epidemiol., vol. 46, no. 2, pp. 578–588, 2017.
6. United Nations Environment Programme, “UNEP Food Waste Index Report 2021,” Nairobi, 2021.
7. R. S. H. Istepanian and J. Lacal, “Emerging Mobile Communication Technologies for Health,” in Proc. IEEE EMBS 2003, pp. 1414–1416.
8. J. Wei et al., “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models,” in Proc. NeurIPS 2022.
9. D. Li and M. J. Zaki, “Reciptor: An Effective Pretrained Model for Recipe Representation Learning,” in Proc. 26th ACM SIGKDD, 2020, pp. 1719–1727.
10. M. Gim et al., “RecipeBowl: A Cooking Recommender Using Set Transformer,” IEEE Access, vol. 9, pp. 143623–143633, 2021.