

International Journal Research Publication Analysis

Page: 01-17

AURACART: A HEALTH-AWARE SMART GROCERY PLATFORM TO CLOSE THE NUTRITION GAP IN ONLINE FOOD DELIVERY

***Kartik Linge and Krushna Kalmegh, Prof. N. J. Padole**

MCA, P.G. Department of Computer Science and Technology H.V.P. Mandal's Degree

College of Physical Education, Amravati, Maharashtra, India.

*P.G. Department of Computer Science and Technology, H.V.P. Mandal's Degree College of
Physical Education, Amravati.*

Article Received: 06 September 2026

Article Revised: 26 September 2026

Published on: 16 October 2026

***Corresponding Author: Kartik Linge**

MCA, P.G. Department of Computer Science and Technology H.V.P.

Mandal's Degree College of Physical Education, Amravati, Maharashtra,
India.

DOI: <https://doi-doi.org/101555/ijrpa.6516>

ABSTRACT

Online grocery and food delivery apps make buying food quick and easy, but they are built around speed, price and catalogue size. They do little to help users judge whether their basket is healthy. This paper points out that gap: none of the platforms we compared combines grocery ordering, AI, nutrition guidance and personalization. We present AuraCart, a mobile-first web platform that does. It looks at the shopping cart as a whole. It gives the cart a health score from 0 to 100, warns about junk food, suggests healthier swaps, recommends recipes from the cart contents, and builds weekly and budget meal plans. We describe the architecture, give the formulas behind each feature, and walk through example carts, including one where two swaps raise the score from 58 to 82. We also discuss the business model, privacy, data quality and scalability, and give a plan for testing the system with real users. The results shown here are worked examples, not user-study data.

INDEX TERMS: Food Delivery, E-Commerce, Health Recommender System, Nutrition, Meal Planning, Artificial Intelligence, Personalization.

I. INTRODUCTION

Ordering groceries online has become routine, especially in cities. Apps like Blinkit and BigBasket bring milk, vegetables or a packet of chips to the door within minutes, and many students and working families now depend on them. The same ease works against healthy

eating, though. Adding chips to the cart takes one tap, exactly like adding spinach, and nothing on the screen tells the buyer the difference. The World Health Organization lists unhealthy diet among the main global health risks and links a healthy diet to lower rates of diabetes, heart disease, stroke and some cancers [1].

Most of these apps are built to sell. Their suggestions, such as “frequently bought together”, come from sales data and not from nutrition. The information needed to do better does exist. Packaged food in India must carry a nutrition label under FSSAI rules [2], but the label is on the pack and people rarely read it while shopping. Front-of-pack schemes like Nutri-Score [3] show that one simple indicator helps people compare products. They work one product at a time, however, and a household buys a whole basket.

When we compared existing services, one thing stood out. Blinkit is mainly about fast delivery. BigBasket is a large grocery catalogue. Amazon Fresh also has a large catalogue but little personalization, and Yummly only does recipes. We could not find a platform that puts grocery ordering, AI, health guidance and personalization in one place. That is the gap this paper works on.

Our answer is AuraCart, a smart grocery platform with a health layer on top of normal shopping. It tries to help the user decide what to buy, what to cook and what to avoid. We built a web prototype, and in this paper we describe the design, the scoring methods behind it and a plan to evaluate it. The contributions are a description of the gap in current platforms, the AuraCart design, formulas for the cart health score, healthy swaps, recipe ranking and budget meal selection, worked examples with an evaluation plan, and a discussion of the business model and risks.

Section II reviews related work and Section III states the problem. Sections IV and V describe the system and its methods. Section VI covers the business model, Section VII the implementation and evaluation, Section VIII the risks and limits, and Section IX future work. Section X concludes.

II. RELATED WORK

A. Online Grocery and Food Delivery Platforms

Grocery apps share a common design: a searchable catalogue, a cart, delivery, and a recommendation module. The recommender usually works by collaborative or content-based filtering, which predicts what a user may want from past behavior and from similar items [4], [5]. This raises relevance and basket size. What it optimizes is sales, so nothing in the method checks whether the recommended basket is healthy. On the platforms we looked at, the user

never sees an analysis of this kind.

B. Food and Recipe Recommender Systems

Food recommendation is an active research area. Min *et al.* review frameworks for food recommendation and name health, context and multimodal data as open problems [6], and their wider survey of food computing covers recognition, recommendation and monitoring [7]. Freyne and Berkovsky broke recipes into ingredients and learned ingredient preferences from ratings [8]. Teng *et al.* built networks of ingredients that go together and ingredients that can replace each other [9], which is close to what we need for healthy swaps. Content-based methods, which match item descriptions to a user profile [10], sit underneath many of these systems.

Healthiness has received attention too. Trattner and Elswailer found that many recipes on popular web portals are unhealthy by established standards, and that recommenders learned from user behavior can inherit the problem [11]. Elswailer *et al.* showed that choice biases can be used to push people toward healthier recipes, for example by offering healthier dishes that look like ones they already like [12]. All of this work is about recipes and meal planning. None of it sits inside a live grocery cart and checkout.

C. Nutrition Labelling and Nudging

A summary label like Nutri-Score [3] saves people the effort of reading a nutrition table. In behavioral economics, a design choice that steers behavior without taking options away is called a nudge [13]. We use the same idea for the whole basket. AuraCart never blocks a purchase. It shows what the basket looks like nutritionally and offers an easy alternative.

D. Comparison of Platforms

Table I summarizes the main focus of the platforms we compared. It follows the competitor analysis done for AuraCart and shows each platform's main positioning. Platforms change, so some of them may have added features since.

Table I Comparison of Platform Focus.

Platform	Primary focus	Grocery	Integrated health AI
Blinkit	Fast delivery	Yes	No
BigBasket	Grocery catalogue	Yes	No
Amazon Fresh	Large catalogue	Yes	No
Yummly	Recipes only	No	No
AuraCart	Food decision intelligence	Yes	Yes

The table confirms the gap described above. None of the platforms combines grocery ordering, AI, health analysis and personalization.

E.Positioning of This Work

To put this together: recommender techniques for food already exist [5], [6], [10]. Studies show that personalization without a health goal can repeat bad eating patterns [11] and that nudges can help [12], [13]. Most of this was tested on recipes or in lab settings, not in a real grocery flow. AuraCart sits in that missing spot. It brings health-aware recommendation to the moment of purchase and treats the cart as the thing to be analyzed.

III. PROBLEM STATEMENT AND OBJECTIVES

A.Problem Statement

People who order food and groceries online decide without seeing what their basket adds up to nutritionally. We split the problem into three gaps. The *information gap* is that nutrition data exists but is not summarized for the basket when the user buys. The *decision gap* is that the user gets no help deciding what to cook with the items or what to leave out. The *context gap* is that budget and personal goals are ignored, even though they matter a lot to students and middle-class families.

B. Objectives

We set five objectives: (1) design a platform that joins grocery ordering with an AI health layer; (2) measure basket healthiness with an easy 0 to 100 cart health score; (3) detect junk food in the cart and suggest healthier substitutes; (4) recommend recipes from the cart and build weekly and budget meal plans; and (5) give vendors visibility and analytics so that the platform works for both sides of the market.

C.Research Questions

RQ1: Can a cart-level score make the nutritional quality of a basket clear to ordinary users?

RQ2: Can healthier substitutes be suggested in a way that respects price and preference, so that users actually accept them? *RQ3*: Can recipes and affordable meal plans be generated from the cart and a stated budget?

D.Scope and Assumptions

The prototype deals with packaged and fresh grocery items sold through one marketplace, for people who shop for themselves or a small household. We assume that each product can be mapped to a food group and a processing tier, that vendors give correct prices and stock, and that users are willing to share a short profile (diet type, goals, budget). Managing medical conditions through diet is outside the scope. The platform gives general guidance and does not replace professional advice. Restaurant delivery is also left out for now, though the same scoring could later be applied to menu items.

IV. PROPOSED SYSTEM: AURACART

A. Overview

AuraCart is a mobile-first, web-enabled platform that combines e-commerce, AI and nutrition information. It has two kinds of users: customers who buy, and vendors who sell groceries. The mission is to make every food decision smarter, healthier and data-driven. The longer-term aim is to become a leading AI-powered healthy grocery platform in India.

B. Functional Modules

AuraCart has seven modules. The *Smart Grocery Platform* handles search, cart, online ordering and live product suggestions. The *AI Recipe Recommendation System* proposes recipes and cooking ideas from what is in the cart. The *Health Awareness Engine* spots junk food in the cart and suggests healthier alternatives. The *Smart Health Score System* gives the cart a score from 0 to 100, with tips for improving it. The *Weekly Meal Planner* builds a plan from the user's preferences and goals. The *Budget Meal Generator* suggests meals that fit a budget, which helps students and middle-class families in particular. The seventh module, the *AI Food Scanner*, is future work: the user will upload a photo, and the system will detect the ingredients and recommend recipes. Two more features support long-term awareness. The Monthly Health Report summarizes buying trends, and the Healthy Swap Engine handles substitution.

C. System Architecture

Fig. 1 shows the layers. At the top are the customer interface and the vendor dashboard. Below them is the application layer, which does the usual commerce work: catalogue, cart, orders, login and vendor management. The AI and health-intelligence layer is what makes AuraCart different. It holds the health score engine, the junk alert and swap module, the recipe recommender, the weekly planner and the budget meal generator. At the bottom, the data layer stores products with their nutrition attributes, recipes, user profiles and order history.

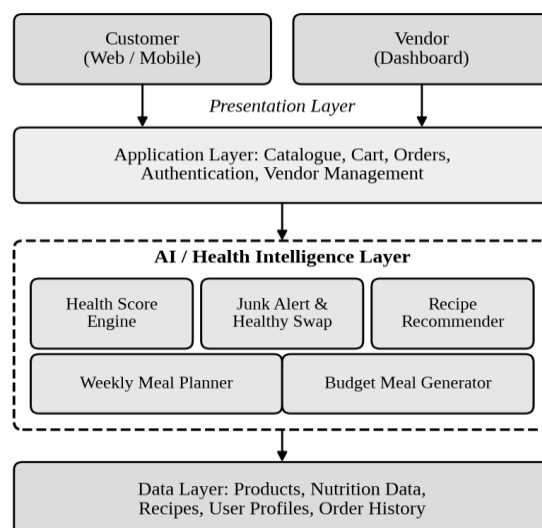


Fig. 1. Layered architecture of AuraCart, showing the AI / health-intelligence layer between the commerce services and the data layer.

The health layer only reads the cart and does not change the checkout logic. Because of this, it can be updated by itself, scaled separately, and switched off for users who want plain shopping.

D. Data Requirements

We need three data sets. The *product data* holds each item's name, price, vendor, food group, processing tier and, where available, values from the label such as sugar, sodium, saturated fat and fibre. The *recipe data* holds ingredients, quantities, estimated cost and preparation time. The *user profile* holds diet type (for example vegetarian), goals, budget and purchase history. Reliable nutrition data matters a great deal here, and we come back to it in Section VIII.

E. User Workflow

A normal session goes like this. The user signs in and sets a diet type, goals and, if wanted, a budget. Then the user searches the catalogue and adds items to the cart. After every change the cart panel shows the health score, any junk alerts and one-tap swap options. At any point the user can open the recipe view, which lists dishes the current cart can make along with the extra ingredients needed, or ask for a weekly plan or a budget meal set. At checkout the order is placed as usual, and the vendor receives it together with anonymous, aggregated demand information. After the order, the purchase data goes into the monthly health report and updates the user profile.

F. Data Model

Table II lists the main entities in the data layer and who uses them. Nutrition attributes are kept with the product, not inside the recommender, so fixing a product's data immediately

changes every score, swap and recipe that uses it.

Table II Principal Data Entities.

Entity	Key attributes	Used by
Product	Id, name, price, vendor, food group, tier, label values	Score, swap engine
Recipe	Id, ingredients, quantities, cost, preparation time	Recommender, budget generator
User	Id, dietary type, goals, budget	Personalization
Cart Order	Items, quantities, status, timestamps	Analysis, monthly report
Vendor	Id, catalogue, stock, subscription plan	Marketplace, analytics

G. Non-Functional Requirements

We set four non-functional requirements. *Responsiveness*: cart analysis must finish within the time of a normal cart update, so shopping does not slow down. *Explainability*: every score and swap comes with a short reason in plain language. *Privacy*: personal data are collected only for stated purposes, and the user decides whether the health layer is on. *Extensibility*: tiers, weights and recipes are stored as configuration, so a nutritionist can adjust them and new modules such as the food scanner can be added without redesigning the commerce side.

V. METHODOLOGY

A. Product Classification

Each product i gets a food group g_i (vegetables, grains, dairy, pulses, snacks, beverages and so on) and a processing tier t_i . Each tier has a base score. A small adjustment $\delta_i \in [-10, +10]$, taken from label values, is then added: more fibre raises the score, and high sugar, sodium or saturated fat lowers it. So $s_i = \text{clamp}(stier + \delta_i, 0, 100)$. Table III shows the tiers in the prototype. The numbers can be changed, so a qualified nutritionist can recalibrate them.

Table III: Processing Tiers and Base Scores.

Tier	Typical examples	Base s
Fresh	Vegetables, fruits, sprouts	90
Minimally processed	Milk, oats, whole grains, pulses	70
Processed	Packaged bread, jam, flavoured yoghurt	40
Ultra-processed (junk)	Chips, soft drinks, instant noodles	15

B. Cart Health Score

Suppose the cart has items $i = 1, \dots, n$ with quantities q_i .

The base score is the quantity-weighted average

$$B = \sum q_i s_i / \sum q_i \quad (1)$$

A healthy diet uses several food groups, so we add a bonus for variety. Let G_h be the set of distinct food groups that have at least one item with $s_i \geq 70$. Then

$$D = \min(10, 2|G_h|) \quad (2)$$

$$H = \min(100, B + D) \quad (3)$$

We read the score in three bands: 0–39 is poor, 40–69 is moderate and 70–100 is good. Next to the number, the interface lists the lowest-scoring items and any healthy food groups that are missing. H is only a weighted average plus a capped bonus, so a user can see exactly why the score changed when an item is added or removed. We think this helps people trust it.

C. Junk Alert and Healthy Swap Engine

An item is flagged as junk when $s_i < 30$. For each flagged item i we collect candidate products j that serve the same purpose (a crunchy snack, a cold drink), match the user's diet, score clearly higher and do not cost much more. The candidates are ranked by

$$W(i,j) = \omega_1(s_j - s_i)/100 + \omega_2 \text{sim}(i,j) + \omega_3(1 - \Delta p_{ij}) \quad (4)$$

where $\text{sim}(i,j) \in [0,1]$ measures how close the category and taste are, $\Delta p_{ij} = \max(0, (p_j - p_i)/p_i)$ is the relative price rise, and $\omega_1 + \omega_2 + \omega_3 = 1$. In plain words, a good swap should be healthier, familiar and affordable. As discussed in Section II, the original item stays available and the user makes the choice.

D. Recipe Recommendation

Let C be the set of ingredients in the cart and I_r the ingredients of recipe r . Coverage is $\text{cov}(r) = |C \cap I_r| / |I_r|$. The recipe health score H_r uses the same tier method on the recipe's ingredients, and $\text{pref}(r,u) \in [0,1]$ shows how well the recipe fits the user's diet and past choices. Recipes are ranked by

$$\text{Score}(r) = \alpha \text{cov}(r) + \beta H_r/100 + \gamma \text{pref}(r,u) \quad (5)$$

with $\alpha + \beta + \gamma = 1$. High coverage means the user can cook the dish with little extra shopping, and the missing ingredients can be added to the cart with one tap, so the recommendation helps the sale too. The health term stops the recommender from pushing popular but unhealthy dishes, the problem reported in [11].

E. Weekly Meal Planner and Budget Meal Generator

The weekly planner picks meals from the ranked list using two rules: no recipe is used more than twice a week, and each day covers at least three food groups. The budget meal generator is a similar selection problem. Given a budget T and candidate recipes with estimated costs c_r , it picks the set of recipes that

$$\max \sum x_r H_r \text{ s.t. } \sum x_r c_r \leq T, x_r \in \{0,1\} \quad (6)$$

This is the 0/1 knapsack problem. The candidate list is short, so exact dynamic programming is practical. For longer lists, a greedy rule that sorts by H_r/c_r is a fast approximation. Algorithm 1 puts the pieces together.

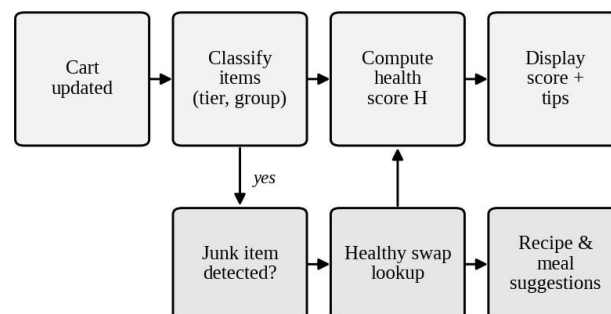
F. Cart-Analysis Procedure

Algorithm 1: Cart analysis

Input : cart $\{(i, q_i)\}$, user profile u Output: score H , alerts, swaps, recipes

- 1 for each item i in the cart:
- 2 look up (g_i, t_i, s_i)
- 3 $B \leftarrow \text{sum}(q_i * s_i) / \text{sum}(q_i)$
- 4 $D \leftarrow \min(10, 2 * |G_h|)$; $H \leftarrow \min(100, B + D)$
- 5 for each item i with $s_i < 30$:
- 6 raise junk alert for i
- 7 swaps(i) $\leftarrow$ top- k candidates by $W(i, j)$
- 8 $R \leftarrow$ top- k recipes ranked by $\text{Score}(r)$
- 9 return H , alerts, swaps, R

Fig. 2 shows when this runs. The analysis is repeated whenever the cart changes, so the score and suggestions always match the current basket.



Analysis is re-run on every add / remove / quantity change

Fig. 2. Event-driven cart-analysis pipeline. Junk detection leads to swap lookup, after which the score is recomputed.

G. Properties of the Score

A few properties follow directly from the definitions. First, the score is bounded: since $0 \leq s_i \leq 100$ we get $0 \leq B \leq 100$, and H is capped at 100. Second, a swap never lowers the score. Suppose a flagged item i ($s_i < 30$) is replaced, at the same quantity, by an item j with $s_j \geq s_i$. The numerator of B changes by $q(s_j - s_i) \geq 0$ and the denominator stays the same, so B cannot fall. The bonus D cannot fall either, because item i was below 70 and so added no group to G_h , while item j can only add one. So following a suggestion never lowers the score the user sees. Third, changing one item's score among n equal-quantity items moves B by at most $100/n$. Large baskets are therefore stable and small baskets react strongly, which seems right to us because a small basket is easy to improve.

H. Computational Complexity

Scoring a cart with n items needs one lookup per item, so it runs in $O(n)$. Swap search is restricted to the same category. With a category index holding m products per category, getting the top k costs $O(m \log k)$ per flagged item. Recipe ranking uses an inverted index from ingredients to recipes, so only recipes that share an ingredient with the cart are scored. The budget selection by dynamic programming takes $O(N \cdot T)$ for N candidate recipes and an integer budget T in whole rupees, which is fine for the short lists the ranking step produces. These costs are small, so the analysis can be repeated after every cart change.

I. Parameter Selection

The weights in (4) and (5) are design parameters, not results. As starting values we suggest $\omega = (0.5, 0.3, 0.2)$ for swaps, so that health gain counts most but similarity and price still matter, and $(\alpha, \beta, \gamma) = (0.4, 0.4, 0.2)$ for recipes, so that convenience and healthiness count equally. They should be tuned in the user study, for example until swap acceptance and recipe relevance stop improving. Later they could be set for each user from feedback.

VI. BUSINESS AND REVENUE MODEL

AuraCart is a marketplace with two sides. Vendors, including local vegetable sellers, get visibility, customer analytics and smart inventory suggestions. Customers get the health layer, which is what sets the platform apart. Personalized recommendations should keep users coming back, and encouraging healthier buying should build trust over time.

We expect revenue from five sources: commission on product sales, vendor subscription plans, premium AI diet features such as advanced diet plans and personal coaching,

sponsored product recommendations, and partnerships with healthy-food brands. Sponsored placement becomes a problem if it is allowed to clash with the health promise. We therefore suggest that sponsored items are clearly labelled, never change the health score, and appear in recommendations only if they also pass a minimum score. Later partnerships with local vendors, health brands, fitness apps and nutritionists would add both supply and expert knowledge.

Growth would happen in stages. First, improve AI personalization using swap and recipe feedback from a pilot group of students and households. Second, bring in local vendors and expand the mobile app, so that fresh produce, which is both the healthiest and the least labelled category, enters the catalogue. Third, develop health partnerships and the premium tier. Each stage depends on the one before. Without good personalization the premium features have little to offer, and without vendor supply the recommendations cannot be fulfilled.

Table IV summarizes what the platform is meant to deliver to each stakeholder.

Table IV Stakeholder Value.

VII.IMPLEMENTATION AND EVALUATION

A. Prototype

The web prototype of AuraCart covers the grocery ordering flow, the cart health score, junk alerts and healthy swaps, recipe suggestions, the weekly meal planner and the budget meal generator. The AI food scanner is left for future work. The prototype follows the layers of Fig. 1, so the health logic stays separate from the commerce services.

B.Illustrative Scenario

To show how the method behaves, take a cart with five items of quantity one: spinach (fresh, $s = 90$), oats (minimally processed, 70), milk (70), potato chips (ultra-processed, 15) and a cola drink (15). By (1) the base score is $B = (90+70+70+15+15)/5 = 52$. Three healthy groups are present (vegetables, grains, dairy), so $D = 6$ and $H = 58$ by (3), which is in the moderate band. The engine flags the chips and the cola as junk and offers roasted chickpeas (70) and buttermilk

(70) as swaps in the same categories at similar prices. Suppose the user accepts both. Then $B = (90+70+70+70+70)/5 = 74$. The cart now has four healthy groups (the chickpeas add pulses), so $D = 8$ and $H = 82$, which is in the good band. Fig. 3 compares the item scores and the cart score before and after the swap.

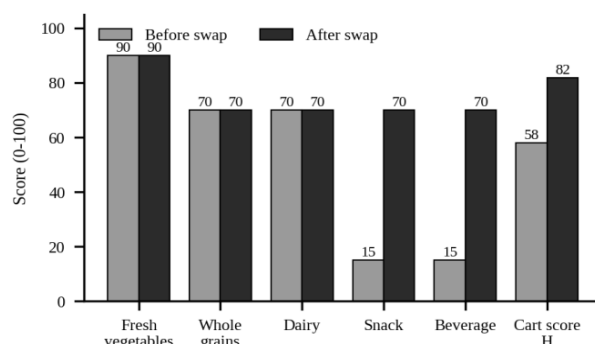


Fig. 3. Illustrative effect of two healthy swaps on item scores and on the cart health score H (values computed with Eqs. 1–3; example data, not measured results).

The example is built to show how the score works. It is not a measured result. It does show that the score moves in proportion to the change and that the reason can be read straight from the inputs.

C.Mapping Gaps to Features

Table V maps each gap from Section III to the AuraCart feature that addresses it and the effect we expect.

Stakeholder	Value delivered
Customers	Smart grocery shopping, health insights and AI-based food guidance
Vendors	Increased visibility, customer analytics and smart inventory suggestions
Health brands	Access to health-conscious buyers; sponsored placement bound by score rules
Nutritionists, fitness apps	A channel for expert plans and integrations (future scope)

Table V: Gap-to-Feature Mapping.

Gap	AuraCart feature	Expected effect
Information	Smart Cart Health Score; Monthly Health Report	Visible, interpretable basket quality
Decision: buy / avoid	Junk Food Alert; Healthy Swap Engine	More healthy choices at the point of purchase
Decision: cook	AI Recipe Recommendation	Cart items turned into meals
Context	Weekly Planner; Budget Meal Generator	Affordable plans matched to goals

D.Behavior Across Scenario Carts

To see how the score behaves beyond one example, we computed it for three more carts of five single-quantity items, using the tiers in Table III. Cart A (chips, cola, instant noodles, biscuits, bread) is a convenience-heavy basket. Cart B (vegetables, fruit, wheat flour, lentils, milk) is a balanced household basket. Cart C (instant noodles, bread, banana, milk, oats) is a student basket, where replacing the noodles with flattened rice (poha, minimally processed) is a typical swap. Table VI gives the results. All values follow directly from Eqs. (1)–(3) and

are illustrative.

Table VI: Cart Health Scores for Scenario Carts.

Cart	B	$ G_h $	D	H	Band
A. Convenience-heavy	25	0	0	25	Poor
B. Balanced household	78	5	10	88	Good
C. Student, before swap	57	3	6	63	Moderate
C. Student, after swap	68	3	6	74	Good
Sec. VII-B, before swap	52	3	6	58	Moderate
Sec. VII-B, after swap	74	4	8	82	Good

The ordering is what we expected. The convenience-heavy cart is in the poor band. The balanced cart reaches the good band through both its item quality and its variety bonus. The student cart goes from moderate to good with one cheap swap. The variety bonus also does its job: Cart B gets the full ten points for covering five healthy groups, while Cart A gets none.

E. Budget Meal Example

We illustrate the budget generator with five candidate recipes and a budget of $T = \text{INR } 150$ (Table VII). The costs and scores are made up to show the selection steps. Sorting by H_r/c_r gives oats porridge, vegetable poha and dal with rice. Together they cost INR 135 and score 240. Adding instant noodles would go over budget (INR 160), and the paneer sandwich costs more and gives less health value per rupee. We also checked every combination of the five recipes, and none that fits the budget scores above 240, so the greedy choice is optimal here. The cheapest recipe, instant noodles, is also the least healthy, which is why the objective uses health score and not price alone.

Table VII: Illustrative Budget Meal Selection. ($t = \text{inr } 150$)

Recipe	H_r	Cost (INR)	H_r/c_r
Oats porridge	80	35	2.29
Vegetable poha	82	40	2.05
Dal and rice	78	60	1.30
Instant noodles	30	25	1.20
Paneer sandwich	65	70	0.93

F. Evaluation Plan

We plan to test the system from the user side and the system side. For usability we will use the System Usability Scale [14] with students and household shoppers. To measure the effect on behavior, we will compare the mean cart health score between a control interface without the health layer and the AuraCart interface, and we will record the swap acceptance rate

(accepted swaps divided by swaps offered). Recommendation quality will be checked with precision at k of the recipe suggestions that participants judge relevant, and with the share of recommended recipes in the good band. For system quality we will use the characteristics in ISO/IEC 25010 [15], mainly performance efficiency (response time of the cart analysis), reliability and security. No results from this study are reported here, and the numbers in Section VII-B should be read as an illustration only.

We have three hypotheses. *H1*: users who see the health layer will build carts with a higher mean score than control users. *H2*: swaps that are both similar to the original item and close in price will be accepted more often than swaps that lack either property, which would support the form of (4). *H3*: participants will rate usability above the commonly used benchmark average for the scale. Participants will be recruited from students and household shoppers and assigned at random to the control or the treatment interface. Each will do a shopping task with a fixed list of needs and a fixed budget, then fill in a questionnaire. We will compare groups with standard two-sample tests and report effect sizes along with significance.

G. DISCUSSION

We made two design choices on purpose. The score is simple. A black-box model might predict healthiness better, but a tier-and-adjustment score can be checked and calibrated by a nutritionist, and that matters when we are asking people to change their behavior. The second choice is that swaps are suggestions and not forced replacements. Whether someone accepts a swap depends on taste, price and habit, which is why (4) gives weight to similarity and price as well as to the health gain.

H. Threats to Validity

Several things limit what can be concluded. The tier-based score is a simplification and may misclassify some items, especially regional or home-made foods. The nutritionist calibration described in Section V-A is meant to reduce this. People in a lab-style shopping task may act differently from real shoppers, so a field trial will be needed. A sample from one region may not represent users elsewhere in India.

Finally, a short-term change in cart contents does not prove a long-term change in diet. The monthly health report makes longer follow-up possible in future work.

VIII. CHALLENGES, RISKS AND LIMITATIONS

Data accuracy is the first concern. Every output of the system depends on nutrition data. Manufacturer labels can be incomplete, and loose or regional foods may have no label at all.

The prototype therefore uses the tier assignment as a baseline and treats label-based adjustments as a refinement. Curation by a qualified nutritionist and regular audits are needed.

The recommender will need continuous improvement. New users have no history (the cold-start problem), and a system trained only on behavior can reinforce existing unhealthy habits [11]. The health term in (5) reduces this risk but does not remove it.

User adoption is another worry. Changing food habits takes time, and many people pick a platform for speed or price. Health guidance therefore has to take little effort and must not get in the way, and users must be able to dismiss or disable it. Competition is also strong. Established players have large user bases and delivery networks, so AuraCart has to rely on its health layer and its vendor relationships and not on delivery speed.

On scalability, a large user base means frequent cart updates, and each one triggers an analysis. Caching of product scores, asynchronous processing of recommendations and horizontal scaling of the intelligence layer will be needed as the user base grows.

Privacy needs care. Purchase history and diet goals are personal data and can reveal health-related behavior. The platform should collect only what it needs, ask for informed consent and use data only for stated purposes, in line with India's Digital Personal Data Protection Act, 2023 [16].

Relying on third-party APIs for catalogue, payment or nutrition data brings availability and cost risks. The health score is a guide and not medical advice. It simplifies nutrition and may not suit users with special medical needs. We also want to avoid numeric scores encouraging an unhealthy focus on food, which is another reason to let users hide the score.

Indian diets vary by region, religion and income, and a score calibrated on a narrow set of foods can disadvantage traditional staples. Tier assignments and swap candidates should be reviewed with regional foods in mind. Diet preferences such as vegetarian, vegan and Jain should be respected in every suggestion, and the budget mode should never push low-income users to spend more in order to reach a higher score.

Table VIII collects the main risks and how we plan to reduce them.

Table VIII: Risks and Mitigations.

Risk	Consequence	Mitigation
Data accuracy	Misleading scores and swaps	Nutritionist curation; label audits; vendor data validation
Recommendation bias	Reinforcement of junk-food habits	Health term in (5); diversity checks
Privacy	Misuse of purchase data	Consent; minimization; access control; legal compliance
Low adoption	Guidance ignored	Low-effort, dismissible interface; clear explanations
Third-party APIs	Outage or cost increase	Caching; fallbacks; alternative providers
Scalability	Latency under load	Cached scores; asynchronous jobs; horizontal scaling
Sponsored items	Loss of user trust	Clear labelling; score independent of sponsorship

IX. FUTURE SCOPE

There is plenty left to add. The AI Food Scanner will use image-based food recognition, a well-developed area of food computing [7], so users can photograph what they have in the kitchen and get recipe ideas. Working with local vendors and vegetable markets would bring in fresh produce that has no label. Partnerships with fitness apps and nutritionists could bring in user goals and expert-checked plans. Support for Hindi and Marathi would open the platform to more people, and premium coaching and advanced diet plans would develop the subscription model. On the technical side, learned ranking models trained on swap-acceptance feedback could improve personalization, and the tier-based score should be compared with established nutrient-profiling systems in a controlled study.

X. CONCLUSION

We started from a simple observation: online grocery and food delivery apps are built for speed and choice, and they leave the user alone with the question of whether the basket is healthy. AuraCart adds a health layer to the normal shopping flow. The cart health score, junk alerts, healthy swaps, recipe recommendations and budget meal plans turn the cart from a shopping list into a tool for better decisions. We described each part formally, showed how it behaves on example carts and set out how we plan to test it with users. The next step is the user study in Section VII-F, followed by the extensions in Section IX.

ACKNOWLEDGMENT

The authors thank Prof. Shirsagar for guidance and encouragement throughout this work.

REFERENCES

1. World Health Organization, "Healthy diet," Fact Sheet, 2020. [Online]. Available: <https://www.who.int/news-room/fact-sheets/detail/healthy-diet>
2. Food Safety and Standards Authority of India, "Food Safety and Standards (Labelling and Display) Regulations, 2020," Government of India, New Delhi, India, 2020.
3. C. Julia and S. Hercberg, "Development of a new front-of-pack nutrition label in France: The five-colour Nutri-Score," *Public Health Panorama*, vol. 3, no. 4, pp. 537–542, 2017.
4. P. Resnick and H. R. Varian, "Recommender systems," *Commun. ACM*, vol. 40, no. 3, pp. 56–58, 1997.
5. Ricci, L. Rokach, and B. Shapira, Eds., *Recommender Systems Handbook*, 2nd ed. New York, NY, USA: Springer, 2015.
6. W. Min, S. Jiang, and R. Jain, "Food recommendation: Framework, existing solutions, and challenges," *IEEE Trans. Multimedia*, vol. 22, no. 10, pp. 2659–2671, Oct. 2020.
7. W. Min, S. Jiang, L. Liu, Y. Rui, and R. Jain, "A survey on food computing," *ACM Comput. Surv.*, vol. 52, no. 5, Art. no. 92, 2019.
8. Freyne and S. Berkovsky, "Intelligent food planning: Personalized recipe recommendation," in *Proc. 15th Int. Conf. Intelligent User Interfaces (IUI)*, 2010, pp. 321–324.
9. C.-Y. Teng, Y.-R. Lin, and L. A. Adamic, "Recipe recommendation using ingredient networks," in *Proc. 4th Annu. ACM Web Sci. Conf.*, 2012, pp. 298–307.
10. P. Lops, M. de Gemmis, and G. Semeraro, "Content-based recommender systems: State of the art and trends," in *Recommender Systems Handbook*, F. Ricci *et al.*, Eds. Boston, MA, USA: Springer, 2011, pp. 73–105.
11. C. Trattner and D. Elswailer, "Investigating the healthiness of internet-sourced recipes: Implications for meal planning and recommender systems," in *Proc. 26th Int. Conf. World Wide Web (WWW)*, 2017, pp. 489–498.
12. D. Elswailer, C. Trattner, and M. Harvey, "Exploiting food choice biases for healthier recipe recommendation," in *Proc. 40th Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, 2017, pp. 575–584.
13. R. H. Thaler and C. R. Sunstein, *Nudge: Improving Decisions About Health, Wealth, and Happiness*. New Haven, CT, USA: Yale Univ. Press, 2008.
14. Brooke, "SUS: A quick and dirty usability scale," in *Usability Evaluation in Industry*. London, U.K.: Taylor & Francis, 1996, pp. 189–194.
15. *Systems and Software Engineering—Systems and Software Quality Requirements and Evaluation (SQuaRE)—System and Software Quality Models*, ISO/IEC 25010:2011, 2011.
16. *The Digital Personal Data Protection Act, 2023*, Act No. 22 of 2023, Government of India, 2023.