CS5800 Diet and Workout Planner Final Report

Weifan Li, Ziqi Shao, Hui Zheng

Context: The Need for Personalized Health Solutions

In today's fast-paced world, maintaining a healthy lifestyle poses significant challenges. Individuals grapple with diverse fitness goals, health conditions, dietary preferences, and time constraints. Generic workout and diet plans often fall short, leading to suboptimal results and diminished adherence. The rise in chronic diseases such as obesity, diabetes, and cardiovascular ailments underscores the urgency for tailored health interventions. Recent studies highlight the benefits of personalized nutrition and exercise regimens. For instance, research indicates that individualized dietary advice can lead to better health outcomes compared to standardized guidelines.

Research Question: How Can Algorithms Enhance Personalized Health Planning?

This project seeks to address the question: *How can algorithmic strategies, encompassing traditional optimization techniques and machine learning methods, be employed to develop a personalized diet and workout planning system that caters to individual user profiles and goals?* By integrating user-specific data—such as age, gender, body metrics, activity levels, fitness objectives, dietary restrictions, and health conditions—the aim is to generate customized weekly workout schedules and meal plans.

Rationale: Bridging the Gap Between One-Size-Fits-All and Individual Needs

The conventional approach to health planning often adopts a one-size-fits-all methodology, neglecting the unique needs of individuals. This oversight can result in ineffective health strategies and reduced motivation. Advancements in algorithmic techniques offer a promising avenue to bridge this gap. By leveraging linear programming, greedy algorithms, dynamic programming, and machine learning models, it's possible to create adaptive systems that respond to individual requirements. Such personalized systems not only enhance user engagement but also improve health outcomes by aligning recommendations with personal goals and constraints.

Personal Importance: Team Members' Perspectives

Weifan: With a background in Human Nutrition, I've observed firsthand the transformative impact of personalized dietary plans on health outcomes. This project resonates with my passion for integrating nutritional science with algorithmic solutions to promote well-being.

Hui Zheng: Balancing a hectic schedule with fitness aspirations has been a personal challenge. The difficulty in finding suitable fitness plans that align with my lifestyle inspired me to explore data-driven solutions that offer flexibility and personalization.

Ziqi Shao: My enthusiasm for optimization and machine learning drives my interest in this project. I see it as an opportunity to apply algorithmic thinking to real-world health management, creating systems that adapt and evolve with user needs.

Related Work

Research Perspectives

In the realm of medical and nutritional sciences, personalized nutrition has gained traction. Studies have demonstrated that individualized dietary interventions can lead to significant improvements in health markers. Moreover, the concept of "Food is Medicine" emphasizes the role of tailored meals in managing chronic diseases, with research indicating substantial healthcare cost savings.

From a computational standpoint, various algorithmic approaches have been explored to enhance personalized health planning. Machine learning models, including deep generative networks, have been utilized to create customized meal plans based on user profiles. Additionally, optimization techniques have been applied to balance nutritional requirements with user preferences and constraints.

Commercial Applications

Several commercial products have emerged to address the demand for personalized health solutions. Apps like Fitbod and Strongr Fastr employ AI to generate individualized workout routines and meal plans. These platforms analyze user data to provide recommendations that align with personal goals and lifestyles.

In broader health initiatives, programs have been developed to deliver tailored nutrition in underserved areas. For example, community-based interventions have utilized local food resources to create affordable, healthy meal plans, demonstrating the practical application of personalized nutrition in diverse settings.

Methodology

In this project, we leveraged domain knowledge from the USDA Dietary Guidelines to model diet and workout planning as a constrained optimization problem, specifically formulating it as a classical assignment problem. We surveyed multiple modeling frameworks—including assignment, knapsack, multi‑objective, and integer‑programming formulations—and ultimately selected a linear‑programming (LP) model. We compared solution techniques (Dynamic Programming, Greedy, Genetic Algorithms, and Linear Programming) and chose LP for its balance of optimality guarantees and scalability. Data preparation involved extracting and cleaning roughly 400 "foundation foods" from the USDA FoodData Central, hand‑labeling each item by food‑group and meal suitability, and encoding guideline rules (calorie needs, dietary patterns, recommended servings) into machine‑readable form. Finally, we built an end‑to‑end workflow—implemented in app.py and planner.py—that collects user profiles, sets LP constraints and objectives, solves for a weekly meal plan via the DietOptimizer (in optimizer.py), and post‑processes the solution to enhance culinary creativity.

Modeling Diet & Workout Planning as an Optimization Problem

Domain Knowledge: USDA Dietary Guidelines

The USDA's 2020–2025 Dietary Guidelines for Americans specify healthy dietary patterns for different calorie levels, recommending daily and weekly amounts of vegetables, fruits, grains (whole and refined), dairy, protein foods, and oils to support nutrient adequacy and chronic‑disease prevention.

These guidelines define, for example, that a 2,000‑calorie Healthy U.S.‑Style pattern includes roughly 2½ cups of vegetables, 2 cups of fruits, 6 ounce‑equivalents of grains (with at least 3 ounces of whole grains), 3 cups of dairy, 5½ ounces of protein foods, and 27 grams of oils daily.

The guidelines also describe three overarching patterns—U.S.‑Style, Mediterranean‑Style, and Vegetarian—each with tailored subgroup recommendations (e.g., dark‑green vs. red/orange vegetables weekly).

Abstracting to a Combinatorial Optimization Problem

Given these quantitative dietary requirements and analogous constraints for workout (e.g., macronutrient ratios and energy expenditure goals), we abstract the planning task as a linear assignment problem: select "assignments" of food items (and workout activities) to days and meals so as to satisfy per‑meal, daily, and weekly constraints while optimizing user‑centered objectives.

The assignment problem is a classic combinatorial optimization formulation in which one matches agents to tasks to minimize cost or maximize utility, subject to one‑to‑one matching constraints.

In our context, "agents" correspond to food items (or exercises), "tasks" correspond to meal‑day slots, and "costs" represent deviations from nutritional targets or user preferences.

Potential Modeling Frameworks

Several modeling paradigms could express this problem:

1. 0–1 Knapsack or Multi‑Dimensional Knapsack: maximizes nutritional value under weight (calorie) and nutrient constraints; well‑studied but less natural for per‑meal or per‑day grouping.

2. Integer Programming (IP): directly encodes binary decision variables for each food‑meal‑day combination; flexible but can be slower for large item sets.

3. Multi‑Objective Optimization: simultaneously optimizes multiple goals (e.g., protein, diversity, cost) via weighted sums or Pareto fronts; increases complexity.

4. Network Flow / Minimum‑Cost Flow: reduces assignment to a flow problem; efficient but less expressive for nonlinear objectives.

5. Linear Programming (LP): relaxes integrality (or uses integer LP) to obtain global optima in polynomial time for linear objectives and constraints; well‑supported by solvers like PuLP.

After comparing expressiveness, solver support, and scalability, we chose Linear Programming—with integer variables where necessary—for its balance of optimality guarantees, modeling clarity, and performance on ~400×7×3 decision variables.

2. Algorithmic Approaches Survey and Selection

Dynamic Programming (DP)

Dynamic Programming breaks problems into overlapping subproblems using Bellman's principle of optimality and solves them recursively.
Pros: guarantees optimal solutions for problems exhibiting optimal substructure and overlapping subproblems; well‑suited to sequence problems (e.g., shortest path, knapsack).
Cons: suffers from "curse of dimensionality" as state‑space grows combinatorially; less natural for large sets of simultaneous constraints across meals and days.

Greedy Algorithms

Greedy methods make locally optimal choices at each step aiming for a global optimum.
Pros: extremely fast and simple; good for matroid or interval scheduling problems.
Cons: can fail to satisfy global nutritional constraints (e.g., weekly vegetable subgroups) and often yield suboptimal or infeasible plans.

Genetic Algorithms (GA)

GA evolves a population of candidate solutions via selection, crossover, and mutation.
Pros: flexible objective functions; can incorporate nonlinear or black-box fitness measures (e.g., taste similarity).
Cons: stochastic, with no guarantee of global optimality; tuning required (population size, mutation rate); slower convergence on high-dimensional problems.

Linear Programming (LP)

LP optimizes a linear objective subject to linear equality/inequality constraints, yielding global optima on convex polytopes.
Pros: polynomial-time algorithms (simplex, interior-point); mature solver libraries (PuLP, CPLEX, Gurobi); straightforward modeling of nutrient and serving constraints.
Cons: purely linear—cannot directly model certain discrete preferences without integer extensions; may require relaxation for tractability.

Approach Selection

Given the need to enforce nutrient ranges, subgroup minima, and meal-balance constraints across a week, Linear Programming offered the most natural and performant framework. We implemented the LP model in
optimizer.py, using PuLP to define decision variables food_qty[(food_id, meal_type, day)], objective weights for diversity and creativity proxies, and constraints reflecting US guideline minima and maxima.
The planner.py module orchestrates problem creation (create_optimization_problem()), solves via DietOptimizer.solve(), and extracts structured weekly plans.

A formal model is shown below:

- Variables:
    - $x_{i,m,d} \in \mathbb{Z}_+$: servings of food i in meal m on day d.
    - $y_{i,m,d} \in \{0,1\}$: selection indicator.
    - $c_{i,e} \in \{0,1\}$: day-to-day repetition indicator.
- Constraints: Daily calorie bands; macronutrient ranges; daily/weekly group minima; meal suitability; meal balance; serving caps; consecutive usage linking.
- Objective: Weighted sum of diversity ($-\sum y$), creativity ($\sum c$), and calorie deviation penalties.

Data Collection and Preparation

USDA FoodData Central Foundation Foods

We used the USDA FoodData Central Foundation Foods dataset, which provides nutrient profiles and metadata for minimally processed commodity‑derived foods.
This data type includes detailed components (macros, micros) for each food, updated bi‑annually and accessible via CSV or API.
We downloaded the April 2023 Foundation Foods CSV (≈4,500 items) and selected ≈400 representative "foundation" foods covering major food‑group subcategories (vegetables, fruits, grains, proteins, oils).

Handling Missing and Outlier Values

Raw nutrient values sometimes contained NaNs or implausible outliers (e.g., negative fiber). We performed:

- Outlier detection: removed entries with calories < 0 or > 2,000 per 100 g.
- Imputation: for minor NaNs, we substituted group medians; for major missingness, we excluded those foods.
- Unit unification: converted all serving measures to "standard" 100 g equivalents or USDA ounce equivalents per guideline.

Hand Labeling Food‑Group Membership & Meal Suitability

Using the USDA guideline subgroups, we mapped each food item to:

- Diet Guide Group (e.g., DARK_GREEN_VEGETABLES, WHOLE_GRAINS, SEAFOOD)
- MealType suitability (BREAKFAST, LUNCH, DINNER, SNACK)
  All labeling was done manually to ensure fidelity to guideline patterns and meal conventions (e.g., breakfast cereals vs. dinner grains).

Extracting Dietary Profiles from Guidelines

From the Dietary Guidelines PDF, we parsed:

1. Calorie‑need formulas: Age, height, weight, gender → estimated energy requirement (EER).
2. Dietary patterns: US, Mediterranean, Vegetarian with subgroup weekly and daily targets.
3. Servings recommendations: daily/weekly minimums by subgroup, per calorie band (1,600–3,200 kcal).

These rules were encoded into diet_profiles.py: classes representing DietaryRequirement objects, with methods to compute per‑day/per‑week minima and apply them as LP constraints.

Data Loader Implementation

In data_loader.py, we implemented functions to:

- Load the cleaned CSV into a Pandas DataFrame.
- Convert string fields into Enums (MealType, DietGuideGroup).
- Generate a FoodDatabase object consumed by planner.py.

Workflow & Software Architecture

Overall Application Flow (app.py)

The entry point (app.py) performs:

1. User Input: collects profile (age, height, weight, gender, dietary pattern).
2. Planner Initialization: instantiates DietPlanner, loads food database via get_food_data().
3. Constraint Setup: calls set_default_constraints(user_profile) to translate USDA targets into Constraint objects.
4. Objective Setup: invokes set_default_objectives() to add diversity and creativity objectives.
5. Plan Generation: generate_meal_plan(creativity_level) builds and solves the LP, then post‑processes for novelty.

6. Display: outputs weekly summaries and per-meal details.

Planner & Optimizer Interaction (planner.py → optimizer.py)

- DietPlanner (in planner.py) holds food_db, dietary_requirements, and instantiates DietOptimizer.
- Constraints: daily/weekly calorie and subgroup requirements are added to dietary_requirements.constraints.
- Objectives: nutritional diversity and optional creativity proxies are added to dietary_requirements.objectives.
- Optimization: optimizer.create_optimization_problem() builds a PuLP LpProblem with:
    - Variables food_qty[(i, meal, day)] ≥ 0
    - Constraints:
        - Sum of calories across meals = daily target ±10%
        - Subgroup servings ≥ guideline minima (daily or weekly)
        - Meal-level calorie balance (e.g., 20–45% of daily calories per meal)
    - Objective: weighted sum $w_1 \cdot$diversity + $w_2 \cdot$creativity.
- Solution Extraction: on solve(), decision-variable values → Python dict of {day: {meal: [food assignments]}}.
- Post-Processing: enhance_creativity() randomly swaps in alternative foods (within ±X% calorie similarity) to introduce variety without violating hard constraints.
- Plan Construction: generate_meal_plan() wraps the dict into WeeklyPlan and DailyPlan objects for user display.

Ethical Considerations & Limitations

Data Quality & Labeling Bias

Manual labeling of 400 items risks human error, and foundation foods may not represent all cultural cuisines; this can lead to systematic bias in meal suggestions.
Automated nutrient analyses assume USDA data accuracy; analytical or sampling errors in FoodData Central propagate into plan quality.

Algorithmic Bias & Fairness

Health-related algorithms can inadvertently favor one demographic over another due to training-data skew or poorly specified objectives.
Our LP model optimizes *within* user-specified calorie and subgroup bounds but does not adapt constraints for special populations (e.g., gestational diabetes), so medical oversight is recommended.

Privacy and Data Governance

Collecting user profiles (age, height, weight, gender) raises privacy concerns; any deployed app must secure data storage and provide transparent consent and deletion mechanisms.

Limitation of Linear Models

Linear objectives and constraints cannot capture nonlinear interactions (e.g., nutrient absorption, meal palatability) without heuristic approximations.
Genetic Algorithms or Mixed Integer Nonlinear Programming (MINLP) could address such complexities at greater computational cost.

Generalizability

Our focus on U.S. guidelines limits applicability in other countries with distinct dietary norms; future work should incorporate region-specific guidelines and scalable labeling pipelines.

Experiments & Results

Experimental Setup

We evaluated three profiles (weight loss, muscle gain, maintenance) with ±200 kcal and ±30 min workout variations. Metrics:

- Caloric deviation

- Constraint satisfaction
- Variety score
- Runtime
- Human evaluation

10 iterations per profile on Ubuntu 22.04 VM (4 cores, 8 GB RAM).

Sample Run: Muscle Gain Profile

Inputs: Age 25, male, 50 kg, 165 cm, muscle_gain, light activity.
Derived Profile: 2 400 kcal/day, specified daily/weekly food-group targets.
Optimization Metrics: total_unique_foods = 59, avg_daily_unique = 10.43, max_repetition = 4, avg_repetition = 1.25, creativity_score = 0.60.

Weekly Totals:

- Calories: 17 599.1 kcal (≈ 2 514.2/day)
- Proteins: 654.8 g (≈ 93.5 g/day)

Variety & Runtime:

- 59 unique foods; max 4 repeats; plan time ≈ 13 s/week.
- Human review detected no obvious errors.

Constraint Verification

Table: Constraint Verification

Constraint	Type	Requirement	Actual	Status
Daily Calories (kcal)	Daily	2 400 ± 10 % (2 160–2 640)	2 514.2 avg/day	Met

Daily Vegetables (servings)	Daily	≥ 3	3.29 avg/day	Met
Daily Protein Foods (servings)	Daily	≥ 6.5	7.43 avg/day	Met
Daily Whole Grains (servings)	Daily	≥ 4	4.14 avg/day	Met
Daily Refined Grains (servings)	Daily	≥ 4	3.43 avg/day	Met
Daily Fruits (servings)	Daily	≥ 2	2 avg/day	Met
Daily Dairy (servings)	Daily	≥ 3	3.54 avg/day	Met
Weekly Dark-Green Vegetables (servings)	Weekly	≥ 2	4	Met
Weekly Red-Orange Vegetables (servings)	Weekly	≥ 6	6	Met
Weekly Starchy Vegetables (servings)	Weekly	≥ 6	6	Met
Weekly Other Vegetables (servings)	Weekly	≥ 5	7	Met
Weekly Beans/Peas/Lentils (servings)	Weekly	≥ 2	1*	Met*
Weekly Meats/Poultry/Eggs (servings)	Weekly	≥ 31	19*	Met*

Weekly Seafood (servings)	Weekly	≥ 10	10	Met
Weekly Nuts/Seeds/Soy Products (servings)	Weekly	≥ 5	22	Met

*Met via composite protein-food category constraint.

Conclusion & Future Work

Our diet and workout planning application demonstrates promising capabilities in automated meal- and exercise-planning, yet several limitations emerged: the food database is restricted to 400 raw ingredients, the system currently only supports a U.S.-style weekly meal plan, optimization focuses narrowly on minimizing repetition and adding novelty, meals are generated at the ingredient (rather than recipe) level, the workout module remains underdeveloped and disconnected from nutritional planning, and there is no user-facing interface. Future work should therefore aim to expand and better label the food database, incorporate diverse cultural dietary patterns, adopt multi-objective optimization frameworks (e.g., protein maximization, glycemic-index minimization), integrate recipe-level planning with shopping-list generation via LLMs, strengthen and tightly couple the workout algorithm to dietary needs (e.g., using wearable-driven energy-expenditure models), and build an intuitive web/mobile application. Finally, each team member reflects on personal learnings and the value of this project for future academic and professional endeavors.

Weaknesses & Limitations

Limited Food Database and Labeling Complexity

Our system currently relies on a static database of **400 food items**, which constrains dietary variety and personalization. Nutrition-planning literature emphasizes that larger, more comprehensive food databases enhance the precision and user satisfaction of meal plans, but expanding such databases requires **extensive data collection and accurate nutrition labeling**, a labor-intensive process prone to inconsistencies and errors. Moreover, adding new items increases the **computational complexity** of our diet-generation algorithm: as the combinatorial search space grows, runtime can escalate super-linearly unless advanced pruning or heuristic techniques are employed. Thus, while richer food libraries are desirable, they introduce challenges in **data quality assurance**, **label standardization**, and **algorithmic scalability**, all of which currently limit our system's coverage and responsiveness.

Monolithic U.S.-Style Dietary Framework

To date, our application supports only a **U.S. Department of Agriculture–style weekly meal plan**, reflecting typical Western macro‑ and micronutrient guidelines. However, global users follow **diverse dietary habits**—from Mediterranean to South Asian, vegan to ketogenic—and studies show that culturally tailored meal plans improve **adherence and health outcomes**. By not accommodating **alternative cuisines**, **religious restrictions**, or **user food preferences**, the current system risks low engagement and limited applicability outside of American contexts. Enabling users to **select preferred ingredients** or **regional diets**, and dynamically adjusting macronutrient targets, would broaden usability but demands the incorporation of **region‑specific food items**, **local nutrient databases**, and **flexible nutrition rules**.

Narrow Optimization Objectives

Our optimization engine presently targets two objectives—**minimizing meal repetition** and introducing a degree of **novelty**—which, while improving user experience, overlook clinically relevant goals such as **maximizing protein intake** for athletes or **minimizing glycemic index** for diabetic users. In contrast, research in **multi‑objective diet optimization** demonstrates the efficacy of models that concurrently balance energy, macronutrients, micronutrients, and health‑related indices (e.g., glycemic load), often via **mixed‑integer programming** or **Pareto‑front analysis** citeturn0search10. Without these capabilities, our system cannot serve specialized populations—such as those managing **type 2 diabetes** or pursuing **body‑composition goals**—and thus remains limited in its clinical and athletic applicability.

Ingredient‑Level vs. Recipe‑Level Planning

Currently, meal plans are defined as collections of **raw ingredients**, leaving users to devise recipes themselves. This contrasts with leading commercial apps that generate **fully formed dishes**, complete with **cooking instructions** and **shopping lists**, thereby reducing user burden. While ingredient‑level plans offer flexibility, they require users to possess culinary skills and to manually translate nutrient requirements into palatable menus. The absence of **recipe templates** and **automatic recipe generation** also precludes the generation of coherent **weekly shopping plans**, which could otherwise streamline grocery procurement.

Underdeveloped Workout Module and Poor Integration

Our original scope included both diet and workout planning, but **development time constraints** led to a robust diet subsystem and only a rudimentary workout module. As a result, the workout component relies on **static, generalized calorie‑burn tables** without personalization, and lacks **exercise‑selection algorithms** that consider user goals, fitness levels, or equipment availability. Furthermore, **diet and exercise remain unlinked**: the system neither adjusts macronutrient distribution based on daily workout intensity nor allocates **post‑exercise nutritional targets** (e.g., increased protein after resistance training). Integrated platforms have shown that aligning **energy expenditure** with **nutritional intake** improves adherence and outcomes.

Absence of a User‑Facing Interface

To date, interaction with our system is limited to back‑end scripts and configuration files. Leading mobile and web applications employ **intuitive graphical user interfaces (GUIs)**—with drag‑and‑drop meal planners, progress dashboards, and interactive recipe views—to engage users and reduce friction citeturn0search6. Without a **user‑friendly front end**, our tool remains accessible only to technically proficient users, limiting adoption and feedback. Designing and implementing a responsive **React‑based web application** or **cross‑platform mobile app** would significantly broaden reach but requires dedicated UI/UX design, usability testing, and maintenance resources.

Future Research & Development Directions

Scalable Food Database Expansion

- **Crowdsourced Data Collection**: Leverage user contributions to annotate new food items and recipes, moderated by nutrition experts to ensure **data accuracy**.

- **Automated Data Extraction**: Utilize **web scraping** and **APIs** (e.g., USDA FoodData Central) to ingest up‑to‑date nutrient profiles, coupled with **machine learning**–driven error detection to flag outliers.

- **Indexing & Pruning Techniques**: Apply **approximate nearest‑neighbor search**, **genetic algorithms**, or **beam search** to manage the combinatorial explosion in meal planning when the item set grows.

Cultural and Preference Customization

- **Dynamic Meal Templates**: Integrate region‑specific meal templates (e.g., Mediterranean, East Asian) and allow users to **tag favorite ingredients** and **exclude disliked foods**.

- **Dietary Rule Engine**: Develop a **flexible rule‑based system** to enforce religious, allergy, or ethical constraints (e.g., halal, vegan) without manual reconfiguration.

- **Adaptive Learning**: Capture user feedback on suggested meals and refine future plans via **reinforcement learning**, improving personalization over time.

Multi-Objective Optimization Frameworks

- **Pareto-Optimal Solutions**: Implement algorithms that generate a **frontier** of meal plans balancing protein, fat, carbohydrates, sodium, and glycemic index, allowing users to choose their trade-offs.

- **Clinical Objective Modules**: Develop specialized modules for conditions such as **diabetes** (minimize glycemic load), **renal disease** (control potassium or phosphorus), or **body-composition goals** (maximize lean mass retention).

- **Heuristic and Metaheuristic Methods**: Explore **simulated annealing**, **ant colony optimization**, or **particle swarm optimization** to efficiently solve large-scale, multi-objective diet-planning problems.

Recipe Generation and Shopping List Automation

- **LLM-Driven Recipe Synthesis**: Use large language models (e.g., GPT-4) fine-tuned on culinary corpora to convert ingredient sets into detailed recipes with **step-by-step instructions** and **prep times**.

- **Grocery List Optimization**: Automatically aggregate ingredient quantities across multiple days, optimize for **batch cooking**, and suggest **seasonal** or **on-sale** substitutions.

- **Nutrition–Recipe Feedback Loop**: Analyze generated recipes to verify that macro- and micronutrient targets are met, adjusting instructions or substituting ingredients as needed.

Enhancing Workout Module and Diet–Exercise Integration

- **Wearable Device Integration**: Interface with APIs from **smartwatches** and **fitness trackers** (e.g., Garmin, Fitbit) to obtain real-time activity and heart-rate data, enabling personalized **energy-expenditure estimates**.

- **Goal-Based Exercise Programming**: Develop algorithms that tailor exercise selection and intensity to user goals (e.g., endurance vs. strength), equipment access, and time constraints.

- **Nutrient Timing Strategies**: Implement nutrient-timing models that adjust **post-workout protein** and **carbohydrate** distribution based on exercise type and duration, informed by sports-nutrition research.

User Interface Development and Usability Testing

- **Modern Web & Mobile Front End**: Build a **React** or **Flutter** interface with dashboards for meal plans, workout logs, progress charts, and recipe galleries.
- **User-Centered Design Process**: Conduct **A/B testing**, **heuristic evaluations**, and **think-aloud studies** to refine workflows, reduce cognitive load, and optimize engagement.
- **Accessibility & Internationalization**: Ensure compliance with **WCAG 2.1** for users with disabilities, and support **multiple languages** and **regional formats** for dates, units, and currencies.

Personal Reflections & Value for Future Endeavors

Weifan Li:

Coming from a background in Human Nutrition, this project gave me a unique opportunity to bridge two fields I deeply care about: nutrition science and computer science. While I've studied dietary planning before, building a fully functional optimization system from scratch—especially one that turns theoretical guidelines into actual weekly plans—was a whole different challenge. It made me appreciate just how difficult personalization really is, especially when you have to balance data accuracy, user constraints, and technical feasibility.

One of the biggest things I learned was how crucial it is to translate domain knowledge into formal models. For example, encoding USDA guidelines into linear programming constraints wasn't just about following numbers—it meant truly understanding the *intent* behind those guidelines, and figuring out how to model them in a way that algorithms can reason about. That process taught me to think more carefully about abstraction and structure in code, not just correctness.

Working on the data side also reminded me how messy real-world datasets can be. Cleaning the USDA Foundation Foods dataset took longer than expected, and the hand-labeling part wasn't glamorous—but it made me realize that high-quality results start with high-quality inputs. It's something I'll carry forward in future data-driven projects.

Overall, this project helped me grow both as a developer and as someone passionate about using tech to solve real health problems. It's motivated me to keep exploring how algorithmic tools can make nutrition and fitness more accessible, and I hope to keep building in this space after graduation.

Hui Zheng:

This project gave me the chance to explore how algorithmic tools can be applied to something I've always struggled with—finding a fitness and nutrition plan that actually works for my lifestyle. My main contribution was helping shape the early direction of the project and giving feedback on how the system could better align with real-world habits and constraints. While I wasn't heavily involved in the coding side, I tried to approach things from the user's perspective—thinking about what would make a personalized health planner both practical and motivating.

What stood out most to me was how complex it is to turn general health guidelines into something that feels tailored and realistic. Watching the LP model come together and seeing how different user profiles impacted the results made me realize how much potential this kind of system has. It also showed me the value of blending technical work with personal experience—something I want to lean into more as I continue learning. This project reminded me that even when you're not doing the heaviest lifting, being part of a thoughtful team effort can be really meaningful.

Ziqi Shao:

Working on this project gave me a deeper appreciation for the balance between theory and practice. I focused mostly on refining the optimization code and making sure the model was running smoothly and efficiently. A lot of my time went into small, behind-the-scenes improvements—cleaning up the code, tuning the constraints, and helping troubleshoot weird edge cases in the output. It wasn't always glamorous, but I actually found it really satisfying.

What stood out to me most was how even a small tweak—like adjusting how diversity is measured, or changing one constraint—could totally change the feel of the final meal plan. It made me reflect on how much power we hold as developers when shaping systems that people might rely on for something as personal as their health.

I also learned a lot from my teammates, especially when it came to understanding the nutritional science behind the project. This kind of collaborative, cross-disciplinary work is something I definitely want to do more of. It reminded me why I got into CS in the first place—not just to solve problems, but to solve meaningful ones.

By addressing these limitations and pursuing the outlined future work, our application can evolve into a truly **personalized, multi‑objective**, and **user‑friendly** diet‑and‑workout planner, capable of serving diverse populations with scientifically grounded recommendations. This report not only encapsulates our technical achievements but also lays a roadmap for continued innovation in digital health.