

# NutriSwap: AI-Powered Sustainable Food Discovery Platform

## 1. Executive Summary

**NutriSwap** (also referred to internally as AltFinder) is an AI-powered food discovery and analysis platform developed to help users make healthier, more sustainable dietary choices. The system leverages intelligent algorithms to compare food items side-by-side, analyzing nutritional content, safety (allergens), and environmental impact. Its core feature is the "Smart Swap" engine, which identifies and recommends healthier alternatives to processed or less nutritious foods. The primary goal is to simplify complex nutritional data into actionable insights through an intuitive "Dark Mode" web interface.

## 2. Introduction

In the modern food landscape, consumers are often overwhelmed by nutritional labels, hidden allergens, and the environmental footprint of their diet. Traditional calorie tracking apps often lack context—telling a user *what* they ate, but not *what they could have eaten instead*. **NutriSwap** was conceived as a direct solution to this problem—a smart tool designed to bridge the gap between data and decision-making by offering instant comparisons, allergen filtering (e.g., No Dairy, No Gluten), and sustainability scoring.

## 3. Project Objectives

- To develop a robust **Search & Discovery Engine** that allows users to find food items and view detailed metadata immediately.
- To design a **Comparison Tool** that visualizes nutritional differences (Macros, Sugar, Sodium) between two food items side-by-side.
- To implement a **Dietary Safety System** that filters results based on specific health goals (Diabetic, Heart Safe) and allergens.
- To create a centralized **User Dashboard** for tracking search history, favorite swaps, and aggregated nutritional analytics.

## 4. Project Scope

The project focuses on developing a responsive web-based application. The scope includes:

- **Food Search Results:** A grid view of food items with quick-access "Why is this better?" analysis.
- **Nutrition Explorer Modal:** A detailed view displaying macro breakdowns and eco-impact scores.

- **Smart Swap Engine:** Logic to suggest alternatives based on caloric density and nutrient profiles.
- **Out of Scope:** E-commerce integration (purchasing food), social sharing features, and live barcode scanning (for this phase).

## 5. Methodology

The project followed an agile component-based development approach.

- **Data Modeling:** The process began by defining the data models (`models.py`) for Food items, Nutrition facts, and Sustainability metrics.
- **Backend Development:** APIs were constructed using FastAPI to serve food data and handle search queries (`check_db.py`, `main.py`).
- **Frontend Construction:** The UI was built using React and Tailwind CSS, focusing on modular components like `AnalyticsCard.jsx`, `ComparisonPreview.jsx`, and `SmartSwapButton.jsx`.
- **Animation Integration:** Framer Motion was integrated to provide smooth transitions for modals and data visualizations.

## 6. Technology Stack

- **Frontend:** React (Vite), JavaScript (JSX).
  - *Justification:* React's component-based architecture allowed for reusable UI elements like the `MetricBar` and `ApplImage` components.
- **Styling:** Tailwind CSS.
  - *Justification:* Enables rapid UI development with a consistent, modern "Dark Mode" aesthetic visible in the dashboard.
- **Backend:** Python, FastAPI.
  - *Justification:* High performance for API requests and easy integration with data science libraries if needed for future AI expansion.
- **Database:** SQLite / SQLAlchemy.
  - *Justification:* Lightweight and efficient for managing relational data between Food Items, Nutritional Info, and Allergens.
- **Development Tools:** VS Code, Postman (implied for API testing), Git.

## 7. Detailed Description

**NutriSwap** is a comprehensive platform for dietary optimization.

- **Search & Filter Workflow:** Users interact with a search bar. The system filters the database not just by name, but by dietary tags (e.g., "Exclude Dairy").
- **Comparison Engine:** When a user selects two items, the `ComparisonView.jsx` component renders. It uses `MetricBar` components to visually display the "winner" for specific categories (e.g., Lower Sodium, Higher Protein).
- **Eco-Impact Scoring:** Beyond nutrition, the system calculates a sustainability score, likely based on water usage and carbon footprint data stored in the backend, helping users make eco-conscious choices.
- **User Dashboard:** A personalized space (`user-dashboard/index.jsx`) that displays `AchievementBadge` and `SearchHistoryItem`, gamifying the experience of finding healthy foods.

## 8. Resources Used

- **Libraries:** `framer-motion` for animations, `react-router-dom` for navigation, `lucide-react` (implied) for icons.
- **Assets:** Custom food imagery and a consistent design system (`AppIcon`, `AppImage`) stored in the `assets/images` directory.
- **Data Source:** A seeded database (`seed.py`) containing nutritional values for various cuisines including Indian, Chinese, and Global dishes.

## 9. Implementation Plan

1. **Phase 1: Database & Backend Logic:** Created `models.py` and `check_db.py` to ensure data integrity and relationships between foods and allergens.
2. **Phase 2: API Development:** Established endpoints to fetch food lists and individual details.
3. **Phase 3: Component Design:** Built atomic components (`SmartSwapButton`, `ActionButtons`) and layout wrappers (`Header`, `UserMenu`).
4. **Phase 4: Feature Integration:** Connected the comparison tool and nutrition modal to the real backend data.
5. **Phase 5: UI Polish:** Implemented the landing page animations ("Eat Smarter. Live Better") and error handling (404 Page).

## 10. Results and Deliverables

- A fully functional **Food Search Interface** capable of filtering complex dietary requirements.
- A **Comparison Modal** that provides instant visual feedback on nutritional value.

- A **Responsive Landing Page** with modern aesthetics and smooth animations.
- A **Robust Codebase** organized into clear directories (pages, components, utils) for scalability.

## 11. Challenges and Risks

- **Data Accuracy:** Ensuring the nutritional data in `seed.py` accurately reflects real-world food items was critical for user trust.
- **UI Complexity:** Managing the state for the `NutritionExplorerModal` and ensuring it overlays correctly without disrupting the user flow required careful implementation of `AnimatePresence`.
- **Image Handling:** Handling broken or missing image links required creating a robust `ApplImage.jsx` component with fallback logic.

## 12. Lessons Learned

The NutriSwap project highlighted the importance of modular architecture in React. By separating `MetricBar` and `NutrientChart` into their own components, the code became much easier to maintain. Additionally, integrating accessibility features (like high-contrast dark mode) from the start proved more efficient than adding them later.

## 13. Recommendations

- **Mobile App:** Develop a React Native version to allow users to scan foods in supermarkets.
- **Personalization:** Expand the `DietaryPreferences.jsx` to allow users to set specific macro goals (e.g., "I want to eat 150g of protein daily").
- **Community Features:** Allow users to submit their own "Swaps" and have them verified by the AI/Admins.

## 14. Conclusion

NutriSwap successfully demonstrates how technology can simplify nutrition. By automating the comparison process and highlighting "Smart Swaps," the system empowers users to take control of their health and environmental impact. It transforms static nutritional labels into an interactive, educational experience.

## 15. References

- Official documentation for React and Vite.
- Official documentation for FastAPI and SQLAlchemy.
- Tailwind CSS utility documentation.

