NutriNode: AI-Native Consumer Health Scanner

"Code To Innovate" Submission for UDGIAM ENCODE 2026
Problem Statement: Designing AI-Native Consumer Health Experiences

📱 Live Prototype

[View the Live Simulation Here]

💡 The Core Concept

NutriNode is not just a barcode scanner; it is a reasoning engine. Traditional health apps act as database browsers—they fetch data ("20g Sugar") and leave the interpretation to the user. NutriNode uses **Generative UI** to interpret that data *for* the user based on their specific health context (e.g., Inflammation, Diabetes, Veganism).

Instead of static forms, NutriNode generates a unique interface for every scan:

- **For a Diabetic User:** It generates a high-priority "Glucose Warning Card."
- **For an Athlete:** It generates a "Protein Density Badge" for the exact same product.

🏗️ System Architecture

High-Level Design

The system follows the **Thesys "Generative UI" Pattern**:

1. **Input:** Image/Barcode from User.
2. **Data Layer:** Fetch raw ingredients from **OpenFoodFacts API**.
3. **Reasoning Layer: Thesys C1 SDK** (powered by LLM) acts as the decision maker. It compares user profile tags against product ingredients.
4. **UI Generation:** Instead of text, the AI outputs a JSON DSL describing which UI components (HazardCard, BenefitBadge, Gauge) to render.
5. **Rendering:** The frontend (React/Next.js) maps this DSL to visual components using the **Crayon SDK**.

```
graph TD
    User[User Scan] --> Frontend
    Frontend -->|Barcode| API_Route
    API_Route -->|Fetch| OpenFoodFacts
    OpenFoodFacts -->|Raw Data| AI_Engine

    subgraph "Reasoning Engine (Thesys)"
```

```
    AI_Engine[LLM / Thesys C1]
    Context[User Profile: Inflammation]
    AI_Engine -->|Compare| Context
  end

  AI_Engine -->|Generates UI DSL| Frontend
  Frontend -->|Renders| DynamicUI[Dynamic Cards & Warnings]
```

🛠 Tech Stack

- **Frontend:** React, Next.js 14 (App Router), Tailwind CSS
- **AI/Middleware:** Thesys C1 SDK (for streaming UI generation)
- **Component Library:** Lucide React (Icons), Custom "Glassmorphism" UI
- **Data Source:** OpenFoodFacts API (World)
- **Deployment:** Vercel

🚀 Setup & Installation

To run the full development version locally:

1. **Clone the repository**
   git clone
   https://github.com/yourusername/nutrinode-encode2026.git
   cd nutrinode-encode2026

2. **Install Dependencies**
   npm install

3. Environment Setup
   Create a .env.local file with your Thesys credentials:
   THESYS_API_KEY=your_key_here
   OPENFOODFACTS_USER_AGENT=NutriNode-Encode2026

4. **Run Development Server**
   npm run dev

🧠 The Reasoning Engine (Prompt Strategy)

The core of NutriNode is the system prompt located in src/app/api/chat/route.ts. We instruct the AI to behave not as a chatbot, but as a **UI Generator**.

System Prompt Extract:

> "You are an expert clinical nutritionist. Do not output conversational text. You must output a JSON array of UI Components. Analyze the provided ingredients against the user's goal ('Reduce Inflammation').
>
> If an ingredient is inflammatory (e.g., High Fructose Corn Syrup), generate a HazardCard with riskLevel: high.
> If an ingredient is beneficial (e.g., Omega-3), generate a BenefitCard.
> Always explain 'Why' in 1 sentence."

🔮 Future Roadmap

- **Web Search Fallback:** Integrate **Exa.ai** to analyze products not found in OpenFoodFacts by searching manufacturer websites.
- **Multimodal Input:** Allow users to snap photos of restaurant menus, not just barcodes.

Submitted by Team Tech V for IIT Guwahati UDGIAM 2026.