NutriAI - Project Design Phase

1. System Architecture Design

The NutriAI system follows a client-server architecture. The frontend is developed using Streamlit, while the backend logic is handled in Python. The AI processing is powered by Google's Gemini 2.5 Flash model integrated through LangChain.

---

2. Component Design

Frontend Component: - Streamlit UI - Text input form - Submit button - Output display section Backend Component: - PromptTemplate for structured prompts - GoogleGenerativeAI model configuration - API key management using environment variables

---

3. Data Flow Design

Step 1: User enters food items. Step 2: Input is formatted using PromptTemplate. Step 3: The formatted prompt is sent to the Gemini model. Step 4: AI generates nutritional information. Step 5: The result is displayed on the Streamlit interface.

---

4. Database Design

Currently, NutriAI does not use a database. All responses are generated dynamically using the AI model. Future versions may include a database to store user queries and responses.

---

5. User Interface Design

The UI is designed to be minimal and user-friendly. It contains: - Title header - Text area for food input - Submit button - Loading spinner - Nutritional output section

---

6. Security Design

- API keys are stored securely using environment variables. - No sensitive user data is stored. - Secure API communication with Google Generative AI services.

---

7. Scalability Considerations

The system can be scaled by: - Deploying on Streamlit Cloud or cloud platforms - Adding caching mechanisms - Integrating a database for faster repeated queries

---