## **PROMPT 1: Initial Expert Setup**

You are an expert Product Specification Document (PSD) writer with 15+ years of experience creating winning product specs for tech companies. I need to create a comprehensive PSD for a calorie tracker application for an AI challenge competition.

I have a code base with this structure:

- Backend folder (Node.js, runs on port 9000)

- Frontend folder (Vite-based, runs on port 4173)

- Mock-service folder (runs on port 9002)

- Docker compose setup with all three services

- Google OAuth integration (VITE\_GOOGLE\_CLIENT\_ID)

- Uses JWT authentication

As a PSD expert, conduct a self-consultation where you ask yourself 5 critical questions about this project and then answer them based on your analysis of the technical stack and industry knowledge of calorie tracking applications. After this consultation, create a comprehensive PSD outline.

## **PROMPT 2: Technical Architecture Deep Dive**

Based on our initial consultation, now act as a senior technical architect and create a detailed Technical Architecture section for the PSD.

Given the stack:

- Node.js backend (port 9000)

- Vite frontend (port 4173)

- Mock service (port 9002)

- Docker containerization

- Google OAuth + JWT

Create a comprehensive technical architecture specification including:

1. System architecture diagram (ASCII/text format)

2. Technology stack details with justifications

3. Database design considerations

4. Security architecture

5. Deployment strategy

6. Scalability considerations

Make this enterprise-grade and competitive quality.

## **PROMPT 3: Functional Requirements Generation**

Now create comprehensive Functional Requirements for the calorie tracker PSD. Based on modern calorie tracking applications and the sophisticated technical stack we discussed, generate:

1. Authentication & User Management requirements (FR-001 through FR-006)

2. Food Database & Search requirements (FR-007 through FR-012)

3. Meal Logging & Tracking requirements (FR-013 through FR-018)

4. Analytics & Progress Tracking requirements (FR-019 through FR-024)

5. Goal Setting & Recommendations requirements (FR-025 through FR-030)

Each requirement should be specific, measurable, and implementation-ready. Use professional FR-XXX formatting.

## **PROMPT 4: User Stories with Acceptance Criteria**

Create detailed User Stories for the calorie tracker application in proper Agile format. For each story include:

- As a [user type]

- I want to [action]

- So that [benefit]

- Acceptance Criteria (specific, testable)

Create stories for:

1. Authentication (Google OAuth + email/password)

2. Quick meal logging (the core value proposition)

3. Progress tracking and analytics

4. Goal setting and management

5. Advanced features (AI recommendations, social features)

Make them comprehensive enough for a development team to implement.

## **PROMPT 5: Non-Functional Requirements**

Generate comprehensive Non-Functional Requirements (NFRs) for the calorie tracker. Given that this is a health application with Google OAuth integration and enterprise-grade architecture, create NFRs for:

1. Performance Requirements (specific metrics)

2. Security Requirements (OAuth, JWT, encryption)

3. Scalability Requirements (user growth, data volume)

4. Reliability Requirements (uptime, backup, recovery)

5. Usability Requirements (mobile-first, accessibility)

Use NFR-XXX formatting and include specific, measurable criteria that would impress technical stakeholders.

## **PROMPT 6: Data Model & Database Design**

Create a comprehensive Data Model specification for the calorie tracker application. Design database entities for:

1. Users (with Google OAuth fields, preferences, goals)

2. Foods (comprehensive nutrition database)

3. Meal Entries (logging with portions, timing)

4. Goals and Progress Tracking

5. User Preferences and Settings

Present as SQL-like schema definitions with:

- Proper data types and constraints

- Relationships and foreign keys

- Indexes for performance

- Security considerations

Make it production-ready for a PostgreSQL database.

## **PROMPT 7: API Specification**

Design a comprehensive REST API specification for the calorie tracker. Based on the Node.js backend (port 9000) and the features we've discussed, create:

1. Authentication endpoints (including Google OAuth flow)

2. User management endpoints

3. Food database endpoints (search, CRUD operations)

4. Meal tracking endpoints

5. Analytics and reporting endpoints

For each endpoint, include:

- HTTP method and URL

- Request/response formats

- Authentication requirements

- Error handling

Make it detailed enough for frontend developers to implement against.

## **PROMPT 8: Security & Compliance Framework**

Create a detailed Security and Compliance section for the PSD. Given that this handles health data and uses Google OAuth, address:

1. Authentication security (JWT implementation, OAuth2 flow)

2. Data protection (encryption, privacy)

3. API security (rate limiting, validation)

4. Compliance requirements (GDPR, CCPA, health data)

5. Security monitoring and incident response

Include specific security measures, compliance checklists, and implementation guidelines that would satisfy enterprise security teams.

## **PROMPT 9: User Interface & Experience Design**

Generate User Interface specifications for the calorie tracker. Given the Vite frontend (port 4173) and mobile-first approach, create:

1. Layout structure and navigation design

2. Key UI components (dashboard, food search, meal logging)

3. Responsive design specifications

4. Accessibility requirements

5. User flow descriptions

Include wireframe descriptions and UX considerations for optimal meal logging experience. Focus on the <30-second meal logging goal.

## **PROMPT 10: Testing & Quality Assurance**

Create comprehensive Testing and Quality Assurance specifications. Given the full-stack architecture with mock service (port 9002), design:

1. Unit testing strategy (backend and frontend)

2. Integration testing approach

3. End-to-end testing framework

4. Performance testing requirements

5. Security testing procedures

Include specific testing frameworks, coverage targets, and CI/CD integration. Make it suitable for a professional development team.

## **PROMPT 11: Deployment & Operations**

Generate Deployment and Operations specifications. Based on the Docker setup and three-service architecture, create:

1. Docker configuration and containerization strategy

2. Production deployment architecture

3. Monitoring and logging requirements

4. Backup and disaster recovery procedures

5. DevOps pipeline and CI/CD

Include specific configurations, metrics, and operational procedures for enterprise deployment.

## **PROMPT 12: Business Context & Success Metrics**

Create the Business Context section of the PSD including:

1. Product vision and business objectives

2. Target market analysis and user personas

3. Competitive landscape and differentiation

4. Revenue model and business case

5. Success metrics and KPIs

6. Risk assessment and mitigation strategies

Make it comprehensive enough for executive stakeholders and business decision-makers.

## **PROMPT 13: Final Assembly & Polish**

Now combine all the sections we've created into a final, polished Product Specification Document. Create:

1. Professional document header with version control

2. Executive summary

3. Table of contents

4. All sections in logical order

5. Conclusion with next steps

6. Future roadmap and enhancements

7. Appendices with supporting information

Format it as a complete, enterprise-grade PSD. Ensure consistency, professional formatting, and comprehensive coverage.

### 

### **Enhancement Techniques I used after each prompt:**

* What additional details would make this section more comprehensive and competitive?
* Add specific metrics and industry best practices to enhance this section.
* What did I miss that would impress technical stakeholders?