**Product Requirements Document (PRD)**

**📌 Product Title:**

**Calorie Tracker Web Application**

**📍 Problem Statement:**

With rising awareness around health and nutrition, individuals seek simple tools to monitor their daily calorie intake and manage their diet goals. Existing solutions are often bloated, expensive, or lack personalized interfaces.

**🎯 Goal:**

Build a web-based calorie tracking application that allows users to:

* Log food and calories
* View consumption history
* Set and monitor calorie goals
* Authenticate with Google
* Interact with real or mock APIs

**🧰 Tech Stack:**

| **Layer** | **Technology** |
| --- | --- |
| Frontend | React, Vite |
| Backend | NestJS, TypeORM, SQLite |
| Auth | Google OAuth (via @react-oauth/google) |
| Charts | Chart.js, react-chartjs-2 |
| Infra | Docker, Docker Compose |
| Dev Tools | ESLint, Prettier, Jest |

**🔩 Features:**

**1. User Authentication**

* Google OAuth login (frontend integration)
* JWT-based backend session/token handling

**2. Calorie Entry Management**

* Create/read/update/delete (CRUD) entries
* Input: food name, calorie amount, timestamp

**3. Dashboard**

* Daily/weekly calorie summaries
* Graphical representation via Chart.js

**4. Mock API**

* Available on port 9002 for local testing
* Simulates real API for foods or authentication

**5. Environment Setup**

* Dockerized support for full-stack run
* Manual dev setup with separate ports:
  + Frontend: 9001
  + Backend: 9000
  + Mock service: 9002

**🔐 Security Considerations:**

* OAuth tokens handled via secure frontend flow
* Backend validates tokens using google-auth-library
* Input validation with class-validator

**⚙️ Developer Experience:**

* ESLint and Prettier configured
* Nest CLI for backend structure
* Hot-reload for both frontend (vite) and backend (nest start --watch)
* Unit and e2e test scripts included

**📦 Deliverables:**

* Full working codebase
* Docker setup for local deployment
* README with setup instructions
* Functional test coverage (Jest)
* Mock services for offline dev