Product Requirements Document (PRD)

Product Name: Powerbuilder

Platform: Web Application (Mobile-Responsive)

Tech Stack: React (Frontend), Firebase (Auth, Firestore, Hosting), Gemini API (AI Engine)

Document Status: Draft v1.0

1. Product Overview

Powerbuilder is a personalized powerbuilding web application designed to optimize both strength and muscle hypertrophy. By merging modern 5/3/1 strength programming with AI-driven progressive overload, the app dynamically adapts to a user's performance. It serves as a comprehensive fitness hub, tracking workouts, nutrition, sleep, and body weight, providing a holistic approach to physical development.

2. Target Audience

- **Intermediate to Advanced Lifters:** Users who understand basic barbell movements but need structured, periodized programming (5/3/1) to break plateaus.
- **Powerbuilders:** Individuals who want the heavy strength progression of powerlifting combined with the aesthetic accessory volume of bodybuilding.
- **Data-Driven Athletes:** Users who want to track every variable (sleep, macros, RPE) to maximize their gains.

3. Problem Statement

Generic workout templates do not adapt to an individual's daily readiness, recovery, or specific weak points. Existing apps either track workouts passively without offering progression advice, or they rigidly force users into a linear progression model that leads to burnout. Furthermore, users often have to juggle three separate apps for workouts, calorie tracking, and habit (sleep/weight) logging.

4. MVP Goals & Success Metrics (Portfolio Context)

- **Goal:** Build a fully functional, impressive web app prototype that demonstrates frontend UI/UX skills, backend database modeling, and AI API integration.
- **Success Metrics:** * Seamless user onboarding and personalized plan generation.
    - Flawless state management during a live workout session.
    - Successful read/write operations to Firebase without latency.
    - A clean, "dark-mode" premium UI that looks spectacular on a portfolio.

5. Core Features (MVP Scope)

5.1. User Onboarding & AI Profiling

- **Data Collection:** Collect age, weight, gender, goal (cut/bulk/recomp), and current 1-Rep Maxes (1RM) for the Big 4 (Squat, Bench, Deadlift, Overhead Press).
- **AI Plan Generation:** * Calculate starting Training Maxes (TM) based on 5/3/1 principles (usually 90% of true 1RM).
    - Generate a personalized daily macro and caloric split.
    - Generate an initial 4-day workout split with specific bodybuilding accessory movements tailored to user goals.

5.2. AI-Powered Workout Engine

- **5/3/1 Core Progression:** Automatically calculate the required weights for the main lifts based on the current week's wave (e.g., Week 1: 3x5, Week 2: 3x3, Week 3: 5/3/1).
- **Active Workout Logging:** * UI to log completed reps and weight.
    - **Crucial Feature:** RPE (Rate of Perceived Exertion) or RIR (Reps in Reserve) input for *every set*.
    - Rest timer built into the active workout view.
- **AI Progressive Overload:** Upon finishing a workout, the AI analyzes the RPE and AMRAP (As Many Reps As Possible) sets to adjust the next cycle's Training Max and tweak accessory volume (e.g., if user struggled with lockout on bench, AI suggests more triceps work next week).

5.3. Nutrition & Macro Tracker

- **Macro Dashboard:** Visual progress bars for Daily Calories, Protein, Carbs, and Fats.
- **Meal Logging:** Simple text-based meal entry. For the MVP, the AI can parse natural language (e.g., "I ate 200g of chicken breast and a cup of rice") to estimate macros, saving the user from needing a massive food database.

5.4. Health & Recovery Tracking

- **Daily Log:** Quick input for Morning Body Weight and Hours Slept.
- **Trend Visualization:** Simple line charts showing body weight trends versus average sleep and strength progression.

6. Technical Architecture & Database Design (Firebase)

6.1. Tech Stack

- **Frontend:** React.js, Tailwind CSS (for rapid, premium styling), Recharts (for data visualization).
- **Backend:** Firebase Authentication (Email/Password or Google), Cloud Firestore (NoSQL

database).
- **Hosting:** Firebase Hosting.
- **AI Integration:** Google Gemini API (used for parsing meal logs and generating end-of-cycle progression updates).

6.2. Firestore Data Model (High-Level)

- users/{userId}: Stores profile info, current 1RMs, Training Maxes, target macros.
- users/{userId}/workouts/{workoutId}: Stores date, type (e.g., "Bench Day"), and an array of sets (exercise, weight, reps, RPE).
- users/{userId}/nutrition/{date}: Stores daily consumed macros and log of meals.
- users/{userId}/daily_logs/{date}: Stores sleep duration and body weight.

7. Out of Scope for MVP (Future Roadmap)

- Social feeds or sharing workouts with friends.
- Extensive barcode scanning for food (relying on AI natural language parsing for the MVP).
- Apple Health/Google Fit integrations.
- Video library of exercise forms.

8. User Flow Summary

1. **Sign Up:** User authenticates via Firebase.
2. **Onboard:** User inputs maxes and goals -> AI generates the 5/3/1 base and macro targets.
3. **Daily Use:** User opens the app -> sees today's macro goals and today's prescribed workout.
4. **Train:** User clicks "Start Workout", inputs reps/RPE per set, and finishes the session.
5. **Review:** AI updates the dashboard, showing strength progression and preparing the next session's weights.