IRON TRACKER

Product Requirements Document v1.0

The gym tracker that knows your machines

February 2026 | Status: Draft | Confidential

Table of Contents

1. Product Overview

1.1 Problem Statement

Gym-goers who train on multiple machines for the same exercise have no way to accurately track their progress per machine. A user who performs Chest Press on three different manufacturer machines must either create awkward custom exercise names (Chest Press - Hammer, Chest Press - Cybex) or accept that their history conflates fundamentally different resistance profiles. The weights are not comparable, the seat settings differ, and the strength curves vary. Existing apps treat all Chest Press sets as interchangeable, producing misleading progress charts and useless weight suggestions.

Additionally, every mainstream fitness app forces users to start a workout session before logging anything, adding friction to spontaneous training (a quick set of pull-ups between tasks, a single warmup set to test a new machine). This session-first ceremony slows logging, discourages ad-hoc tracking, and does not reflect how many experienced lifters actually train.

1.2 Solution

Iron Tracker is a mobile-first web application that introduces two novel concepts: machine-as-first-class-entity and set-centric logging. Users create equipment variants under each exercise (Chest Press > Hammer Strength Plate-Loaded, Chest Press > Life Fitness Selectorized) with per-machine saved settings, distinct weight histories, and independent progress tracking. Logging starts with a single tap on an exercise—no session ceremony required. Sets are automatically grouped into sessions for history browsing.

1.3 Target User

Primary: Intermediate-to-advanced gym-goers (6+ months experience) who train at a gym with multiple machines for the same movements, care about progressive overload, and want precise per-machine tracking. Typical age 20-45, training 3-6 days/week.

Secondary: Gym newcomers who would benefit from AI machine identification (photograph a machine, get exercise info + form tips) and guided progressive overload. The low-friction set logging reduces the intimidation of workout tracking.

Anti-persona: Home gym owners with one piece of equipment per exercise (they gain no benefit from machine variants). Casual fitness users who don’t track weights (they need a different product).

1.4 Success Metrics

Metric	Target (6 months)	Measurement
Weekly active users	2,000	Users who log ≥1 set in a 7-day window
Sets logged per active user per week	≥20	Average across WAU cohort
Machine variant adoption	≥40% of users with 2+ variants	Users with >1 equipment variant on any exercise
Set logging speed	<5 seconds for repeat set	P50 time from exercise selection to set confirmed
7-day retention	≥60%	Users who return within 7 days of first session
30-day retention	≥35%	Users still active after 30 days

2. User Stories by Feature Area

2.1 Set Logging (Core)

US-101: As a gym-goer, I want to log a set by tapping an exercise and confirming weight/reps so that I can track my training with minimal friction.

US-102: As a returning user, I want my last set’s weight and reps pre-filled so that I can log identical sets in one tap.

US-103: As a user logging multiple sets, I want an auto-starting rest timer after each logged set so that I know when to begin my next set.

US-104: As a user who trains spontaneously, I want to log a set without starting a workout session so that I can track ad-hoc training.

US-105: As a user reviewing my history, I want sets automatically grouped into sessions (by time proximity) so that I can browse workout-level summaries when needed.

US-106: As an advanced lifter, I want to log RPE, RIR, set type (warmup, working, dropset, AMRAP), and tempo so that my training data captures intensity context.

US-107: As a user who made a mistake, I want to edit or delete a recently logged set inline (swipe-to-edit, swipe-to-delete) so that corrections require minimal effort.

2.2 Machine/Equipment Variant Management

US-201: As a user with multiple chest press machines at my gym, I want to create named equipment variants under Chest Press so that each machine has its own weight history.

US-202: As a user switching machines mid-workout, I want to change the active variant with one tap (chip selection) so that it doesn’t slow my logging.

US-203: As a user who forgets machine settings, I want to save seat position, pad height, and cable settings per variant so that I can replicate my setup each visit.

US-204: As a new user encountering an unfamiliar machine, I want to photograph it and receive AI-identified exercise names, muscles targeted, and form tips so that I can learn and start logging immediately.

US-205: As a user viewing progress, I want to toggle between machine-specific and all-variant-aggregated progress charts so that I can see both per-machine and overall exercise trends.

US-206: As a user who only uses one machine for an exercise, I want the variant picker hidden so that it doesn’t add unnecessary interface complexity.

2.3 Progress Tracking and Visualization

US-301: As a user, I want to see my estimated 1RM trend for each exercise over time so that I can measure strength progress.

US-302: As a user, I want to see weekly training volume per muscle group so that I can ensure balanced programming.

US-303: As a user, I want real-time PR detection with celebratory feedback when I hit a new personal record so that milestones feel rewarding.

US-304: As a user, I want a PR table per exercise showing records across rep ranges (1RM, 3RM, 5RM, 8RM, 10RM) so that I can track strength at every rep range.

US-305: As a user, I want a muscle group heatmap showing training frequency and volume distribution so that I can identify imbalances.

US-306: As a user, I want a GitHub-style training calendar showing workout frequency so that I can see my consistency at a glance.

2.4 AI and Recommendations

US-401: As a user, I want AI-suggested weight for my next set based on my recent performance and target RPE so that progressive overload is guided.

US-402: As a user, I want deload recommendations when the system detects accumulated fatigue so that I avoid overtraining.

US-403: As a user with a photo of a gym machine, I want AI to identify the machine and pre-fill an equipment variant so that setup is fast and accurate.

US-404: As a premium user, I want to ask the AI coach questions about my training data in natural language so that I can get personalized insights.

US-405: As a user, I want weekly AI-generated progress summaries highlighting PRs, volume trends, and recommendations so that I receive actionable feedback.

2.5 Soreness and Recovery

US-501: As a user, I want to be prompted for muscle soreness ratings 1-3 days after training so that recovery data improves recommendations.

US-502: As a user, I want to see a muscle recovery status indicator per muscle group so that I know which muscles are ready to train.

2.6 Account and Settings

US-601: As a new user, I want to sign up with email or social login so that onboarding is fast.

US-602: As a user, I want to set my preferred weight unit (kg/lb) and it applies globally so that I don’t have to convert.

US-603: As a user, I want to customize rest timer defaults per exercise type so that the timer matches my training style.

US-604: As a user, I want to export my data (CSV, JSON) so that I own my training data.

US-605: As a user, I want dark mode as the default with a light mode toggle so that the app is comfortable in gym lighting.

3. Feature Priority Matrix

Features are prioritized using MoSCoW for the v1 release. The guiding principle is: everything needed for a gym-goer to open the app, log 5 sets across 2 machines, and see their progress chart must be in Must Have.

3.1 Must Have (v1.0 Launch)

Feature	User Stories	Est. Effort
Set logging with pre-fill and one-tap confirm	US-101, US-102	2 weeks
Equipment variant CRUD with per-machine settings	US-201, US-202, US-203	2 weeks
Progressive disclosure for variant picker	US-206	3 days
Auto rest timer with per-exercise defaults	US-103	1 week

Feature	User Stories	Est. Effort
Exercise library (seeded ~400 exercises)	—	1 week
Set history timeline (grouped into sessions)	US-104, US-105	1.5 weeks
Estimated 1RM trend chart per exercise	US-301	1 week
Basic PR detection (1RM, volume)	US-303	1 week
Auth (email + Google social)	US-601	3 days
Weight unit preference (kg/lb)	US-602	2 days
Dark mode default	US-605	2 days
Offline-first with optimistic updates	—	1.5 weeks
PWA installability	—	3 days

3.2 Should Have (v1.1, 4-6 weeks post-launch)

Feature	User Stories	Est. Effort
AI machine photo identification	US-204, US-403	1.5 weeks
Full PR table per exercise (all rep ranges)	US-304	1 week
Weekly volume per muscle group chart	US-302	1 week
Muscle group heatmap	US-305	1 week
Training frequency calendar	US-306	3 days
RPE/RIR and advanced set types	US-106	1 week
Swipe-to-edit, swipe-to-delete sets	US-107	3 days
Customizable rest timer defaults	US-603	2 days
Soreness tracking prompts	US-501	1 week
Multi-variant vs aggregated chart toggle	US-205	3 days

3.3 Could Have (v1.2+)

Feature	User Stories	Est. Effort
AI weight suggestions	US-401	2 weeks
Deload recommendations	US-402	1 week

Feature	User Stories	Est. Effort
Recovery status indicator	US-502	1 week
Weekly AI progress summary	US-405	1 week
Data export (CSV, JSON)	US-604	3 days
Tempo logging	US-106 (partial)	3 days

3.4 Won’t Have (v1, revisit later)

Natural language AI coach (US-404), social features, workout sharing, workout plan builder/templates, Apple Watch/WearOS companion, integration with heart rate monitors, video form check, in-app purchases/premium tier (keep free while validating product-market fit).

4. Functional Requirements

4.1 Set Logging

FR-101: The system shall allow a user to log a set by selecting an exercise, optionally selecting an equipment variant, entering weight and reps, and confirming. The minimum interaction for a repeated set (same weight, same reps, same variant) shall be 1 tap (confirm button).

FR-102: The system shall pre-fill weight, reps, and equipment variant from the user’s most recent set of that exercise. If the user has never logged the exercise, weight defaults to 0 and reps to 10.

FR-103: The system shall provide inline stepper buttons for weight (±2.5 and ±5) and reps (±1) adjacent to the input fields.

FR-104: Tapping the weight or rep field shall open a focused numeric keypad as a bottom sheet overlay, not a full-screen transition.

FR-105: The system shall auto-start a rest timer upon set confirmation. The timer duration defaults to the exercise’s last-used rest time, or a category default (compound: 180s, isolation: 90s, hypertrophy: 60s).

FR-106: The system shall persist all sets to the local cache immediately (optimistic update) and sync to the server when connectivity allows.

FR-107: The system shall assign a session_group to each set based on the date of logging. Sets logged within a 90-minute inactivity gap shall be considered part of the same session for display purposes.

4.2 Equipment Variant Management

FR-201: Each exercise may have 0 or more equipment variants per user. Exercises with 0 variants shall log sets with variant_id = NULL (generic mode).

FR-202: When an exercise has exactly 1 variant, the variant picker shall be hidden and the single variant auto-selected.

FR-203: When an exercise has 2+ variants, a horizontal row of Material Design 3 FilterChip components shall display below the exercise name, with the most-recently-used variant pre-selected.

FR-204: Each variant shall store: name (required), equipment_type (enum), manufacturer (optional), weight_increment (optional), seat_settings (JSONB, optional), notes (text, optional), photo_url (optional).

FR-205: The system shall allow creating a new variant from a bottom sheet form. The AI photo identification flow shall pre-fill variant fields when available.

FR-206: The system shall track last_used_at on each variant and sort the chip row by most recently used, ensuring the most relevant machine is pre-selected.

4.3 Progress and Analytics

FR-301: The system shall compute estimated 1RM for each exercise using the Epley formula: e1RM = weight × (1 + reps / 30). For sets with >12 reps, the Brzycki formula shall be used instead.

FR-302: The system shall detect personal records in real-time upon set logging. PR dimensions: new estimated 1RM, new rep max at a given weight, new max weight at a given rep count.

FR-303: Upon PR detection, the system shall display an animated celebratory banner with the PR details, which auto-dismisses after 4 seconds.

FR-304: The system shall generate pre-computed analytics for: estimated 1RM trend (per exercise, per variant), weekly volume per muscle group, training frequency (sessions per week), and PR history.

FR-305: Charts shall support date-range filtering and pinch-to-zoom on mobile.

4.4 AI Features

FR-401: The machine photo identification endpoint shall accept a JPEG/PNG image (<5MB), send it to Claude Sonnet via the FastAPI proxy, and return: exercise name(s), equipment type, estimated manufacturer, target muscles (primary and secondary), and 2-3 form tips. Response time target: <5 seconds.

FR-402: All AI interactions shall be rate-limited to 10 requests per user per day on the free tier.

FR-403: AI responses shall be cached by image hash to avoid duplicate API costs for repeated photos of the same machine.

5. Non-Functional Requirements

5.1 Performance

NFR-101: First Contentful Paint shall be <1.5 seconds on a 4G connection.

NFR-102: Time to Interactive shall be <3 seconds on a 4G connection.

NFR-103: Set logging (tap to confirmed) shall complete in <200ms locally (optimistic), with server sync completing within 2 seconds.

NFR-104: Exercise search shall return results within 100ms for a library of 500+ exercises.

NFR-105: The application bundle shall be <200KB gzipped for the initial route.

5.2 Reliability

NFR-201: The app shall function fully offline for set logging, exercise browsing, and history viewing. Offline-logged sets shall sync automatically when connectivity is restored.

NFR-202: The system shall handle sync conflicts using last-write-wins based on the logged_at timestamp. If a conflict is detected (same set ID with different data), the most recent logged_at wins.

NFR-203: Backend availability target: 99.5% uptime (Render's SLA). The frontend and offline features shall remain fully functional during backend downtime.

5.3 Security

NFR-301: All user data shall be isolated via Supabase Row Level Security. No user shall be able to access another user's sets, variants, or profile data through any API path.

NFR-302: The Anthropic API key shall never be exposed to the frontend. All Claude interactions shall be proxied through the FastAPI backend.

NFR-303: User passwords shall be managed entirely by Supabase Auth (bcrypt hashing, no plaintext storage).

5.4 Accessibility

NFR-401: The app shall achieve WCAG 2.1 AA compliance. All interactive elements shall have ARIA labels. Color shall not be the sole indicator of state. Minimum touch target size: 48x48dp (56dp preferred for primary actions).

NFR-402: All charts shall have text-based alternative descriptions.

5.5 Compatibility

NFR-501: Supported browsers: Chrome 90+, Safari 15+, Firefox 100+, Edge 90+.

NFR-502: Supported viewport widths: 320px (minimum) to 1920px. Primary optimization target: 375px-428px (modern smartphone range).

NFR-503: The app shall be installable as a PWA on both Android and iOS (via Add to Home Screen).

6. Release Plan

6.1 Phase 1: Core Logger (Weeks 1-6)

All Must Have features. Users can sign up, create equipment variants, log sets with pre-fill and one-tap confirm, view history grouped into sessions, see 1RM trend charts, receive PR celebrations, and use the app offline. The app is installable as a PWA.

6.2 Phase 2: Intelligence (Weeks 7-12)

All Should Have features. AI machine identification, full PR tables, muscle group visualizations, soreness tracking, advanced set types (RPE/RIR), and the training frequency calendar. This phase transforms the app from a logger into an analytics platform.

6.3 Phase 3: Coaching (Weeks 13-20)

All Could Have features. AI weight suggestions, deload recommendations, recovery indicators, and weekly AI progress summaries. This phase adds the intelligence layer that justifies a future premium tier.

7. Competitive Positioning

Across 15+ fitness apps analyzed (Strong, Hevy, JEFIT, FitNotes, Fitbod, Dr. Muscle, Alpha Progression, Setgraph, StrengthLog, GymBook, Gymaholic, RepCount, GymStreak, Boostcamp, Liftosaur), Iron Tracker is differentiated by a combination that no single competitor offers:

Capability	Iron Tracker	Best Competitor
Machine-specific tracking	Native (variant hierarchy)	None (custom exercise workaround)
Set-centric logging	Yes (no session ceremony)	Setgraph only
Saved machine settings	Yes (seat, pad, cable per variant)	None
AI machine identification	Yes (photo → exercise + tips)	None
Offline-first	Yes (queue + optimistic)	Strong, Hevy (limited)
Per-machine progress charts	Yes (variant toggle)	None
Free tier logging	Unlimited	Strong (3 routines), Fitbod (3 workouts)
1-tap repeat set	Yes (pre-fill + confirm)	Setgraph, RepCount

8. Constraints and Assumptions

8.1 Constraints

Budget: Infrastructure cost must remain under $15/month until 5,000 WAU (Netlify free, Render $7, Supabase free, Upstash free). No native mobile apps in v1 (PWA only). Two-person development team with limited design resources. Claude API costs must be capped at $50/month regardless of usage.

8.2 Assumptions

Users are willing to spend 30-60 seconds once to set up their equipment variants, in exchange for permanently better tracking accuracy. Gym WiFi is unreliable, making offline-first a genuine requirement, not a nice-to-have. Dark mode is the preferred default for gym environments. Users who train on multiple machines represent a sufficient market segment to build a business on. A mobile-first PWA provides acceptable UX compared to native apps for a v1 launch.

8.3 Dependencies

Supabase Auth for user management. Supabase PostgreSQL for data storage. Anthropic Claude API for AI features (machine ID, coaching). Render for backend hosting. Netlify for frontend hosting. These services have published SLAs and redundancy; the primary risk is Render cold starts for AI features, mitigated by the $7/month always-on plan.

End of Product Requirements Document.