

Software Requirements Specification (SRS)

****Project Title:**** Hypertrophy Builder App

****Company:**** Kinertia Labs

****Document Version:**** 1.0

****Date:**** [Insert Date]

1. Introduction

1.1 Purpose

The purpose of this document is to specify the requirements for the Kinertia Labs Hypertrophy Builder app a web-based fitness application designed to guide beginner users through hypertrophy training with smart workout planning, tracking, and progressive overload education.

1.2 Scope

This application allows users to:

- Register and complete an onboarding questionnaire
- Receive a beginner-friendly, personalized 3-day workout split
- Log exercises with sets, reps, and weight
- Track weekly progress
- Learn fitness terms and best practices via short tooltips and guidance

1.3 Definitions, Acronyms and Abbreviations

- ****Hypertrophy****: Muscle growth through strength training

- **RIR**: Reps In Reserve used to measure effort
- **UI**: User Interface
- **UX**: User Experience
- **API**: Application Programming Interface

1.4 References

- NSCA Guidelines on Hypertrophy Training
- Supabase Documentation: <https://supabase.com/docs>
- IEEE SRS Template (Std 830-1998)

1.5 Overview

Section 2 describes system interfaces, functional and non-functional requirements, and design constraints.

2. Specific Requirements

2.1 External Interface Requirements

2.1.1 User Interfaces

- Mobile-first design for modern browsers
- Accessible dashboard with workout status and quick-start buttons
- Workout logging screen with input fields for reps, sets, and weight
- Calendar view to track workout history
- Exercise library with search and filters

2.1.2 Hardware Interfaces

- No special hardware required
- Runs on any smartphone, tablet, or desktop with internet access

2.1.3 Software Interfaces

- Frontend: React or similar JS framework (Lovable-generated if applicable)
- Backend: Supabase for auth, database, and REST API support
- Deployment: Vercel, Netlify, or Lovable.dev hosting

2.1.4 Communication Protocols

- HTTPS for all client-server communications
- RESTful API calls to Supabase backend

2.2 Functional Requirements

- FR1: Users must be able to register and log in securely
- FR2: Users complete an onboarding quiz (experience level, training frequency, equipment access)
- FR3: The app generates a workout split based on quiz results
- FR4: Users can log sets, reps, weight, and optional RIR
- FR5: Users can view and revisit previous workout logs
- FR6: Users can browse a basic exercise library with descriptions and tips
- FR7: The app provides educational tooltips and short guides

2.3 Non-Functional Requirements

2.3.1 Reliability

- The app shall autosave logs during a workout session
- Downtime should not exceed 1% monthly

2.3.2 Availability

- System available 24/7 via cloud hosting
- User data backed up weekly

2.3.3 Security

- All data transmitted over HTTPS
- Supabase handles user auth securely using JWT
- Sensitive info like `.env` values are excluded from public repo

2.3.4 Maintainability

- Modular frontend and backend structure
- Code changes and bug fixes tracked via Git version control

2.3.5 Portability

- Web-based and responsive for mobile, tablet, and desktop
- Built using platform-agnostic frameworks (HTML/CSS/JS)

2.3.6 Performance

- Page load time under 2 seconds
- Rest timer and log input responsiveness within 100ms

2.4 Design Constraints

- Must support browser-based interaction (no native app yet)
- Database schema must be compatible with Supabase
- UI must remain accessible and readable on devices 320px wide