

# Eden – Product & Implementation Summary

## 1. Product Vision

**Eden** is a personal health coaching platform focused on **primespan** – the years where you actually feel strong, clear, and able to do what you care about (not just longevity or lab numbers).

**Core concept:** Health is organized into **five domains**:

- **Heart** – cardio & blood markers (VO2max, resting HR, blood pressure)
- **Frame** – strength & body structure (body composition, body fat %)
- **Metabolism** – energy & blood sugar (HbA1c, fasting glucose)
- **Recovery** – sleep & HRV
- **Mind** – focus & cognition

**Live deployment:** <https://eden-jade.vercel.app> (tryeden.health landing page)

---

## 2. Technical Architecture

### Stack

- **Frontend:** Next.js 14 (App Router), TypeScript, Tailwind CSS
- **Backend:** Next.js API routes (serverless)
- **Database & Auth:** Supabase (PostgreSQL + Auth + Storage)
- **AI:** OpenAI GPT-4.1-mini
- **Design System:** Apple Human Interface Guidelines (iOS-style UI)

### Key Directories

```
app/
├─ page.tsx                # Login (magic link)
├─ chat/page.tsx           # Main experience – AI coach chat
├─ dashboard/page.tsx      # Pentagon visualization + metrics
├─ data/page.tsx           # Data sources + Apple Health import
├─ auth/callback/route.ts  # Magic link handler
├─ api/
│   └─ eden-coach/
│       ├── message/route.ts # Chat endpoint (main AI logic)
│       ├── history/route.ts # Fetch conversation history
│       └─ plan/route.ts     # Weekly plan generation
│   └─ apple-health/process/ # Apple Health XML parsing
│   └─ dev/reset-user/route.ts # Dev tool: wipe user data
└─

lib/
├─ context/
│   ├── buildEdenContext.ts  # Aggregates profile + snapshot + plan
│   ├── getUserSnapshot.ts   # Latest metrics per category
│   └─ deriveUserProfileFromMessages.ts # Extract profile from chat
├─ plans/
│   └─ createWeeklyPlanForUser.ts # LLM-driven weekly plan generator
├─ supabase/
│   └─ server.ts             # Server-side Supabase client
```

```
| └─ client.ts          # Browser-side Supabase client
| └─ auth.ts           # requireAuth() helper
```

---

### 3. Supabase Database Schema

#### Core Tables

##### eden\_user\_profile

Stores user basics extracted from conversations.

Column	Type	Notes
id	uuid	PK
user_id	uuid	FK → auth.users
first_name	text	
age	integer	
sex_at_birth	text	"male" / "female"
height_cm	integer	
weight_kg	numeric	
primary_goal	text	User's main health goal
created_at	timestamp	
updated_at	timestamp	

##### eden\_metric\_categories

The five health domains.

Column	Type	Notes
category_code	text	PK: heart, frame, metabolism, recovery, mind
name	text	Display name
sort_order	integer	

##### eden\_metric\_definitions

What metrics Eden tracks.

Column	Type	Notes
id	uuid	PK
metric_code	text	e.g. "vo2max", "hrv", "sleep"
name	text	Human-readable name

category_code	text	FK → eden_metric_categories
unit	text	e.g. "ml/kg/min", "bpm"
display_order	integer	

#### eden\_metric\_values

Actual measurements per user.

Column	Type	Notes
id	uuid	PK
user_id	uuid	FK → auth.users
metric_id	uuid	FK → eden_metric_definitions
value	numeric	The measurement
measured_at	timestamp	When the value was recorded

#### eden\_user\_snapshots

Point-in-time aggregation of user state (stored as JSON).

Column	Type	Notes
id	uuid	PK
user_id	uuid	
snapshot_json	jsonb	Full UserSnapshot object
created_at	timestamp	

#### eden\_conversations

One per user per channel.

Column	Type	Notes
id	uuid	PK
user_id	uuid	
channel	text	"web" or "whatsapp"
last_message_at	timestamp	
created_at	timestamp	

#### eden\_messages

Chat history.

Column	Type	Notes
id	uuid	PK
conversation_id	uuid	FK → eden_conversations
role	text	"user" or "assistant"
content	text	Message text
created_at	timestamp	

#### eden\_plans

Weekly coaching plans.

Column	Type	Notes
id	uuid	PK
user_id	uuid	
snapshot_id	uuid	Optional: snapshot used
start_date	date	
end_date	date	
status	text	"active" / "completed"
focus_summary	text	1-2 sentence weekly focus
llm_raw	jsonb	Raw LLM response
created_at	timestamp	

#### eden\_plan\_actions

Individual actions within a plan.

Column	Type	Notes
id	uuid	PK
plan_id	uuid	FK → eden_plans
priority	integer	1, 2, 3, ...
title	text	Action title
description	text	Why & how
metric_code	text	Optional target metric
target_value	text	e.g. "7+ hours"
cadence	text	e.g. "daily", "3x/week"

### eden\_user\_personas (stub)

Future: archetype/persona for the user.

Column	Type
id	uuid
user_id	uuid
created_at	timestamp

### apple\_health\_imports

Tracks Apple Health file uploads.

Column	Type	Notes
id	uuid	PK
user_id	uuid	
file_path	text	Path in Supabase storage
file_size	bigint	
status	text	pending/completed/failed
created_at	timestamp	

### Storage Bucket

- apple\_health\_uploads – stores raw .zip exports from iPhone

### Row-Level Security (RLS)

All tables have RLS enabled. Users can only read/write their own data:

```
-- Example policy pattern (on each table):  
CREATE POLICY "Users can manage their own data" ON eden_user_profile  
FOR ALL USING (auth.uid() = user_id);
```

## 4. Key Flows

### 4.1 Authentication

1. User enters email on / (login page)
2. Supabase sends magic link email
3. Link redirects to /auth/callback?code=...
4. Callback exchanges code for session → redirects to /chat
5. Middleware protects /chat , /dashboard , /data

### 4.2 Chat with Eden ( POST /api/eden-coach/message )

1. Parse message from request body
2. Auth check → get user ID
3. Get or create conversation (eden\_conversations)
4. Insert user message into eden\_messages
5. Extract profile from recent messages (deriveUserProfileFromMessages)
  - Uses GPT-4.1-mini to extract: age, sex, height, weight, goal, name
  - Updates eden\_user\_profile if new info found
6. Build EDEN\_CONTEXT via buildEdenContext():
  - profile (eden\_user\_profile)
  - snapshot (getUserSnapshot → eden\_metric\_values)
  - persona (eden\_user\_personas - stub)
  - plan (eden\_plans + eden\_plan\_actions)
7. Fetch last 10 messages for conversation history
8. Build OpenAI messages:
  - System prompt (coaching personality, primespan focus)
  - Context summary (natural language, not raw JSON)
  - Conversation history
9. Call GPT-4.1-mini
10. Insert assistant reply into eden\_messages
11. Return { reply: "..."} }

#### 4.3 Data Import (Apple Health)

1. User uploads .zip export on /data page
2. File goes to Supabase storage ( apple\_health\_uploads bucket)
3. POST /api/apple-health/process parses XML
4. Extracts: VO2max, resting HR, HRV, blood pressure, body fat, sleep
5. Inserts into eden\_metric\_values
6. Redirects to /dashboard

#### 4.4 Weekly Plan Creation (manual/API)

lib/plans/createWeeklyPlanForUser.ts :

1. Build EDEN\_CONTEXT
2. Fetch recent messages
3. Call GPT-4.1-mini with plan-specific prompt
4. Parse JSON response: { focusSummary, actions[] }
5. Mark existing active plans as "completed"
6. Insert new plan + actions

**Note:** Auto-plan creation was removed. Coach can propose plans conversationally.

---

### 5. Frontend Pages

#### /chat (main experience)

- Server component
- Fetches active plan for "This week's focus" banner
- Embeds EdenCoachChat client component
- Loads history on mount via /api/eden-coach/history
- iMessage-style UI

## **/dashboard**

- Server component
- Pentagon radar chart (5 domains)
- Category scores (0-100) with status indicators
- Individual metrics with values and units
- No chat – just data visualization

## **/data**

- Apple Health upload card
- Coming soon: Labs & Wearables
- Dev tool: Reset user data button

---

## **6. System Prompt (Eden's Personality)**

You are **Eden**, an expert health & performance coach focused on extending a person's **primespan** – the years where they actually feel strong, clear, and able to do what they care about.

### **### Your job**

- Help the user figure out **what matters most right now** and **what to actually do about it**.
- Keep things practical, realistic, and humane. You are not a doctor, you are a coach.

### **### Context you receive**

- **Profile**: basics like age, goals, constraints, time available.
- **Health snapshot**: their current state across Heart, Frame, Metabolism, Recovery, and Mind.
- **Weekly plan**: if one exists, it's a focus they're working on this week.

### **### How to coach**

- Sound like a thoughtful human, not a chatbot. Short paragraphs, natural language.
- Ask **one question at a time**. Don't rapid-fire.
- Give **1-3 concrete suggestions**, not 10 vague ideas.
- Acknowledge real constraints (time, energy, injuries).

### **### On weekly plans**

- If there's a plan, use it as a reference – but don't keep re-printing it.
- If there's no plan yet, that's fine. Get to know them first.
- Never rush someone into a plan.

### **### Safety**

- You're not a doctor. Don't diagnose.
- No extreme advice.

---








## **7. Environment Variables**

```
NEXT_PUBLIC_SUPABASE_URL=https://xxx.supabase.co
NEXT_PUBLIC_SUPABASE_ANON_KEY=eyJ...
OPENAI_API_KEY=sk-...
```

---

## 8. Current State & Known Issues

### Working

-  Magic link auth
-  Chat with context-aware AI coach
-  Profile extraction from conversations
-  Apple Health data import
-  Pentagon visualization with scores
-  Chat history persistence
-  Dev reset tool

### Recent Fixes

- Fixed conversation history query (was fetching oldest 10, now most recent 10)
- Added debounce to prevent double-sends
- Removed auto-plan creation (too aggressive)
- Simplified context injection (natural language vs raw JSON)

---

*Document generated: December 2024*