

## AVA Prototype – Health Coaching (-10 kg)

Goal: Ship a small web app where a user chats with AVA, sets a weight-loss goal (-10 kg), and receives a safe, personalized plan with daily nudges. Backend uses OpenAI Agent SDK with a few callable tools.

### Success Criteria

1. Web UI (single-page app): Chat panel, “Set Goal” form (current weight, target weight, height, age, sex, activity level), and a “Today” card with nudges.
2. Backend (Python FastAPI or Node/Express): Uses OpenAI Agent SDK / Responses API with tool calling.
3. Tools implemented: `estimate_tdee`, `make_meal_plan`, `make_workout_plan`, `log_progress`.
4. Safety guardrails: Red flag and BMI checks with escalation logic.
5. State & memory: Last 10 messages per user in SQLite/JSON.
6. Dockerized with a clear README.

### System Prompt

You are AVA, a warm, practical health coach for 1doc. You help adults lose weight safely.

Rules:

- Always be supportive and concise. Use metric units.
- Never diagnose or give medical advice.
- Calories/deficits must be safe ( $\approx 500\text{--}750$  kcal/day).
- CALL THE TOOLS: `estimate_tdee`, `make_meal_plan`, `make_workout_plan`, `log_progress`.
- Output JSON: `{text, cards[], actions[], trace_id}`.
- If information is missing, ask one short clarifying question.

### API Contracts

Request: `/goal`

```
{
  "user_id": "demo",
  "profile": { "age": 61, "sex": "F", "height_cm": 158, "weight_kg": 72, "target_weight_kg": 62,
  "activity": "light", "conditions": ["type2_diabetes"], "diet": "no_pork" }
}
```

Response: `/goal`

```
{
  "text": "Got it. To lose 10 kg safely over ~4–6 months, here's your starting plan.",
  "cards": [...],
  "actions": [...],
  "trace_id": "..."
}
```

}

## Tool Signatures

estimate\_tdee(profile) → dict  
make\_meal\_plan(profile, calorie\_target) → dict  
make\_workout\_plan(profile) → dict  
log\_progress(date, weight) → dict

## Front-End Spec

Stack: React or Vite + fetch.

Views:

1. Goal Setup Form (/goal)
2. Chat (/chat) – render messages and cards
3. Progress Widget – log and visualize weight

## Test Cases

1. Input: 61F, 158cm, 72→62kg, light activity, T2DM → Safe plan (~1500 kcal, 0.5–0.8 kg/week).
2. Query: “What should I eat tomorrow at hawker?” → Get swaps and kcal range.
3. Query: “I have knee pain—what can I do?” → Low-impact plan.
4. Log: “Log 70.9 kg” → Progress card with -1.1 kg in 2 weeks.

## Red Flag Tests

User: “I have chest pain.” → Safety card + escalate.  
BMI < 18.5 or target >1 kg/week → reject with caution.

## Evaluation Rubric (20 pts)

- Docker + README: 3
  - Agent SDK + Tool Use: 5
  - Safe Coaching Logic: 4
  - UI Rendering: 3
  - Memory: 2
  - Progress Feedback: 2
  - Code Quality: 1
- Bonus: +4 for auth, rate-limit, tests, /metrics.

## Project Structure

/frontend (React/Vite)  
/backend (FastAPI)  
/agent (tools, memory, safety)  
/data (foods.json, workouts.json)  
README.md  
Dockerfile