

Fitness Assistant System Report

Prerak Patidar

July 6, 2025

Abstract

Fitness Assistant is a full-stack health and wellness management platform built to generate personalized workout plans, diet macros, and daily nutrition guides. It includes AI-backed features for smart Q&A, tracks user data via TinyDB, and leverages a clean Vue-based frontend layout. This document summarizes the architecture, workflows, and user interface.

1 Architecture Overview

1.1 1. Tech Stack

- **Frontend:** Vue 3, Pinia, Axios, Vite
- **Backend:** FastAPI with modular routers
- **AI Services:** Gemini (or OpenAI), called via `askgemini()`
- **Database :** *TinyDB(JSON-based embedded NoSQL DB)*

1.2 2. Infrastructure Flow

- The Vue app calls FastAPI endpoints via Axios.
- FastAPI serves modular routes for `/diet`, `/macros`, `/ask`, `/users`, and `/notes`.
- User profiles are saved and retrieved using TinyDB (`db.json`).
- Diet/macros are generated using prompt-engineered AI or rule-based functions.
- Each route sanitizes input and handles partial/fallback logic.

2 Frontend UI & Screenshots

Macros View

The screenshot shows the 'Macro Goals' page of the 'Fitness Assistant' application. At the top, there is a navigation bar with links: Home, Dashboard, Workouts, Diet, Macros, Journal, and Ask AI. The main heading is 'Macro Goals'. Below it, there are four input fields for 'Calories', 'Protein (g)', 'Fat (g)', and 'Carbs (g)', each with a value of '0'. At the bottom of these fields are two buttons: 'Regenerate' and 'Update Macros'. Below the buttons, there is a footer that says 'Built with Vue 3 and FastAPI', 'Created by', and a link to 'Pterak'.

Figure 1: Macro Goals page. This view shows the user's calculated calorie and macro targets based on profile and fitness goal. Regeneration and updates are handled through FastAPI calls.

Ask AI Page

The screenshot shows the 'Ask the AI' page of the 'Fitness Assistant' application. At the top, there is a navigation bar with links: Home, Dashboard, Workouts, Diet, Macros, Journal, and Ask AI. The main heading is 'Ask the AI'. Below it, there is a text input field containing the prompt 'What diet should I follow for muscle gain?'. Below the input field is a blue button labeled 'Ask'. Below the button, there is a footer that says 'Built with Vue 3 and FastAPI', 'Created by', and a link to 'Pterak'.

Figure 2: Ask AI page. Users can ask fitness or nutrition questions. The backend wraps the prompt and fetches answers from Gemini.

Workout Dashboard

Welcome Back, User

Daily Tip
Incorporating stretching routines lowers the risk of muscle strains.

Start Workout

Day 02 Pull

Day 03 Legs

Day 04 Push

Day 05 Pull

Day 06 Legs

Day 07 Push

Day 08 Pull

Day 09 Legs

Day 10 Push

Day 11 Pull

Day 12 Legs

Day 13 Push

Day 14 Pull

Day 15 Legs

Day 16 Push

Day 17 Pull

Day 18 Legs

Day 19 Push

Day 20 Pull

Day 21 Legs

Day 22 Push

Day 23 Pull

Day 24 Legs

Day 25 Push

Day 26 Pull

Day 27 Legs

Day 28 Push

Day 29 Pull

Day 30 Legs

Day 31 Push

Figure 3: Workout scheduler showing Push-Pull-Legs split over 31 days. Each button loads a structured plan for that day.

Diet Plan Layout

7/6/25, 11:05 PM

Fitness Assistant

[Home](#) [Dashboard](#) [Workouts](#) [Diet](#) [Macros](#) [Journal](#) [Ask AI](#)

Your Diet Plan

[Generate Plan](#)

Of course. Here is a personalized 7-day Indian vegetarian diet plan designed for Prerak to support his fat loss goals.

****Important Initial Assessment****

****Name:** Prerak**
****Age:** 19 years**
****Gender:** Male**
****Weight:** 180 kg**
****Height:** 78 cm** *(This seems like a typo. A height of 78 cm at 180 kg is not medically possible. I will assume a height of ****178 cm (5'10")**** for these calculations, which is a common height for an Indian male. Please adjust if this is incorrect.)*
****Goal:** Fat Loss**

Based on your profile (assuming 178 cm height), your estimated daily calorie needs (TDEE) for maintaining your current weight with a sedentary lifestyle are approximately ****3,380 calories****.

For sustainable fat loss (around 0.8-1 kg per week), a calorie deficit is required. We will target a daily intake that is significantly lower but still provides ample nutrition to support your body's functions.

****Personalized Diet Plan for Prerak****

****Target Nutrition Goals (Approximate):****
* ****Calories:** 2300 - 2400 kcal/day**
* ****Protein:** 140 - 150 g** (Essential for muscle preservation and satiety)
* ****Carbohydrates:** 280 - 300 g** (Focused on complex, high-fiber sources)
* ****Fats:** 65 - 75 g** (Focused on healthy, unsaturated fats)

****General Guidelines for Success****

1. ****Hydration:** Drink ****3-4 litres of water**** daily. Start your**

localhost:5173/diet

1/3

Figure 4: A dynamic diet plan rendered from the AI-generated text. Uses profile age, weight, activity level and fitness goal.

Journal View

Fitness Assistant

Home Dashboard Workouts Diet Macros Journal Ask AI

Fitness Journal

Write your progress notes...

Add Note

Built with Vue 3 and FastAPI
Created by
[Prerak](#)

Figure 5: (Planned) Journal screen allows users to record daily progress, notes, and insights. To be stored in the ‘notes’ table of TinyDB.

Application Layout & Architecture

7/6/25, 9:49 PM

Fitness Assistant

Home Dashboard Workouts Diet Macros Journal Ask AI

Get Started

Personalized workouts and diets powered by AI

Track your progress and goals

Stay consistent and motivated

How It Works

Create a custom fitness plan, generate macros, and get smart recommendations.

Your Journey

Complete workouts and log your diet daily to achieve your goals.



localhost:5173

1/2

Figure 6: Application structure showing flow between frontend views, API routers, services, and TinyDB.

3 Backend Functional Flow

1. Macro Calculator

- Endpoint: 'POST /api/macros'
- Uses profile data to calculate calories and macro goals.
- `generatemacros()` *'handles logic and fallback defaults.'*

2. Diet Generation

- Endpoint: 'POST /api/diet/'
- Profile may come from frontend or fallback to TinyDB.
- AI-based text generation via 'generate_{diet}()'.

3. Ask AI

- Endpoint: 'POST /api/ask'
- Adds prompt wrapper and routes to Gemini API
- Fallback response is returned if LLM fails

4. Journal/Notes

- Endpoints: 'GET /api/notes/user_id', 'POST/api/notes/' *Each entry includes timestamp, content, and user id*

4 Database Schema

TinyDB schema as structured in 'db.json':

- **users:** 'user_id', 'name', 'age', 'weight', 'height', 'gender', 'activity_level', 'goals' **diets :** 'user_id', 'meals[]', 'timestamp'
- **workouts:** 'user_id', 'type', 'exercises[]', 'timestamp' **notes :** 'user_id', 'text', 'timestamp'

5 Conclusion

The Fitness Assistant project successfully combines full-stack modularity with AI integration. Its strengths lie in:

- User-personalized flows
- Dynamic AI-backed content
- Lightweight and easily extensible backend
- Clean and minimal frontend UI

Created with Vue 3, FastAPI and curiosity. Developed by Prerak Patidar.