# Ayurveda Health Journey

## 1. Introduction

Ayurveda Health Journey is a comprehensive and interactive web application designed to promote holistic wellness through the ancient science of Ayurveda. The platform empowers users to explore Ayurvedic principles, understand their unique body constitution (Prakruti), and adopt personalized lifestyle practices that support balance and health.

Developed using React.js for the frontend and Node.js/Express.js for the backend, the application ensures a seamless, secure, and dynamic user experience. It integrates user authentication, data management, and real-time API interactions to deliver a complete health-tracking and community engagement platform.

## 2. User Guide

**Access the App:**

* Open your browser and navigate to http://localhost:3000 (after starting the dev server).
* If not logged in, you'll be redirected to /login.

**Registration and Login:**

* Register: Go to /register, enter a valid email and password, and click 'Register'.
* Login: At /login, enter your credentials. Successful login stores session data in localStorage and redirects to /dashboard.
* Security Note: Passwords are stored in localStorage for demo purposes. Use secure tokens in production.

**Navigation:**

* Use app links or browser history to switch between pages (Dashboard, Assessment, Reports, Wellness Hub, Profile, Community).

## 3. Key Features

**Dashboard (/dashboard):**

* Displays welcome message, health stats, and quick actions.
* Calculates a 0–100% health score based on assessments, mood, journal consistency, and Prakruti balance.
* Shows Prakruti distribution and progress bars for Vata, Pitta, Kapha.
* Provides daily tips, recent activities, and reminders.

**Prakruti Assessment (/assessment):**

* Multi-step form covering physical, mental, emotional, and lifestyle traits.
* Calculates dominant Prakruti via trait-matching algorithm.
* Displays personalized recommendations and stores past assessments.

**Wellness Hub (/wellness-hub):**

* Meal Planner based on Prakruti (e.g., Vata: warm foods).
* Journal for tracking mood, symptoms, and insights.
* Integration with latest assessment results.

**Reports (/reports):**

* List of all assessments with options to view details or export data.
* Displays detailed comparison of doshas and user traits.

**Profile (/profile):**

* Allows user to edit email and password.
* Logout clears local storage and redirects to login.

**Community (/community):**

* Displays messages and allows users to post and interact in real-time.

## 4. Developer Guide

**Project Architecture**

* Frontend: React.js (SPA with React Router).
* State Management: useState and useEffect hooks.
* Data Flow: Axios for API calls; localStorage for auth persistence.
* Styling: CSS Modules and Tailwind-like utilities.
* Backend: Node.js/Express with MongoDB (Mongoose).
* Icons: Lucide React.

**Key Algorithms:**

* Prakruti Calculation: Trait-based fuzzy matching algorithm.
* Health Score: Weighted average from user metrics.

## 5. Setup and Development

**Prerequisites:**

* Node.js 18+ installed.
* MongoDB running and connected.

**Frontend Setup:**

* git clone <repo>
* cd ayurveda-health-journey
* npm install
* npm start (runs at http://localhost:3000)

**Backend Setup:**

* Install dependencies: express, mongoose, bcryptjs.
* Define models for User, Assessment, Journal, Message.
* Run server: node server.js (http://localhost:5000)

**Development Workflow:**

* Hot reload on frontend.
* Use nodemon for backend auto-restart.
* React DevTools for debugging.

## 6. API Overview

**Authentication:**

* POST /auth/register – Create a new user (hashed password).
* POST /auth/login – Authenticate user credentials.
* GET /auth/me – Retrieve current user data.
* PUT /auth/me – Update email or password.
* DELETE /auth/me – Remove user account.
* POST /auth/logout – Client-side logout.

**Assessments:**

* POST /assessments – Create new Prakruti assessment.
* GET /assessments – List all user assessments.
* GET /assessments/:id – Retrieve a specific assessment.

**Journal:**

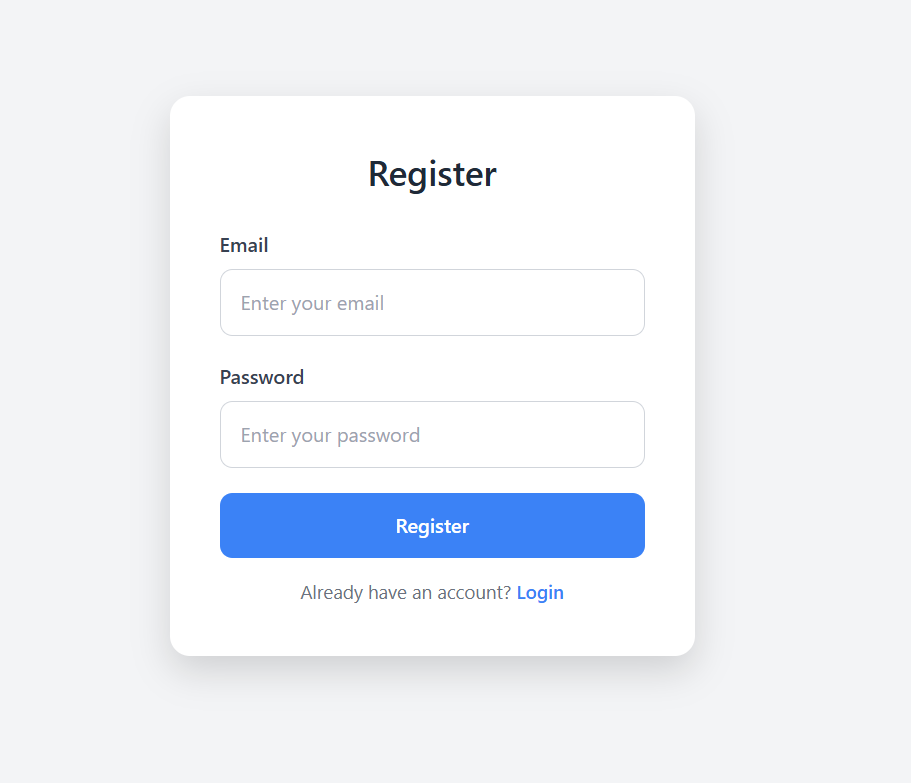
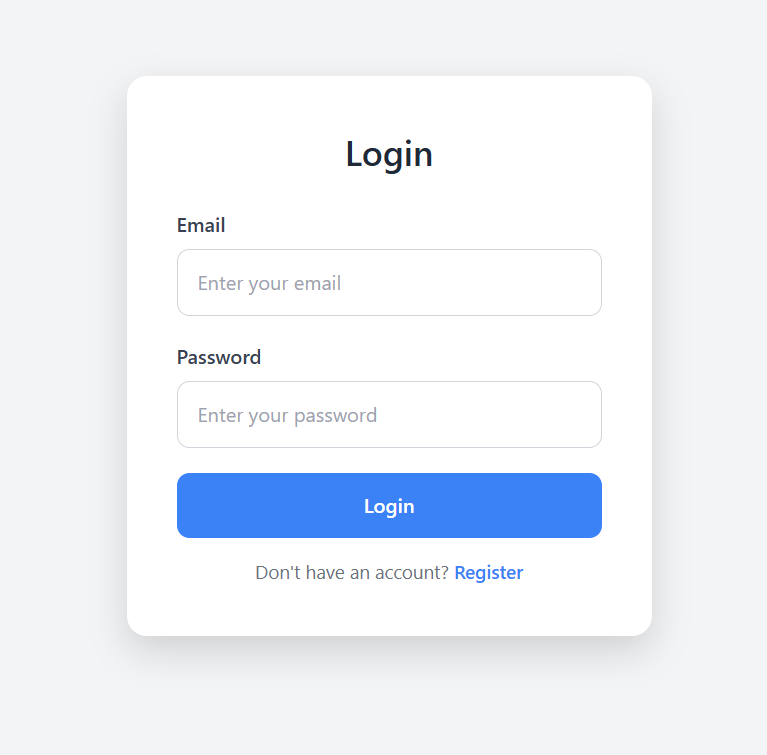
* POST /journal – Create a wellness journal entry.
* GET /journal – List all journal entries.
* PUT /journal/:id – Update existing entry.
* DELETE /journal/:id – Delete entry.

**Messages (Community):**

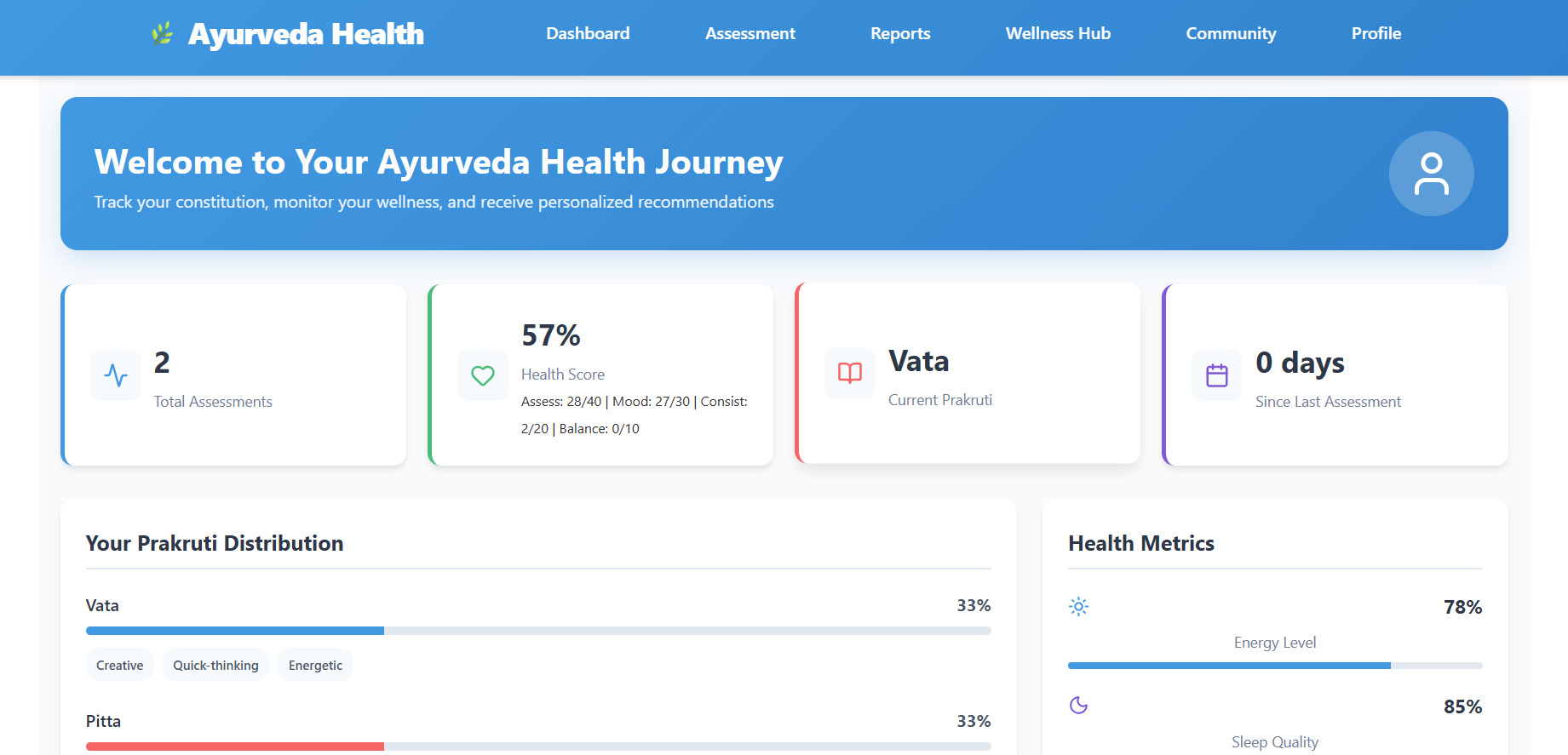
* POST /messages – Send new message to global chat.
* GET /messages – Retrieve latest 50 messages.

## 7. Project Screenshots

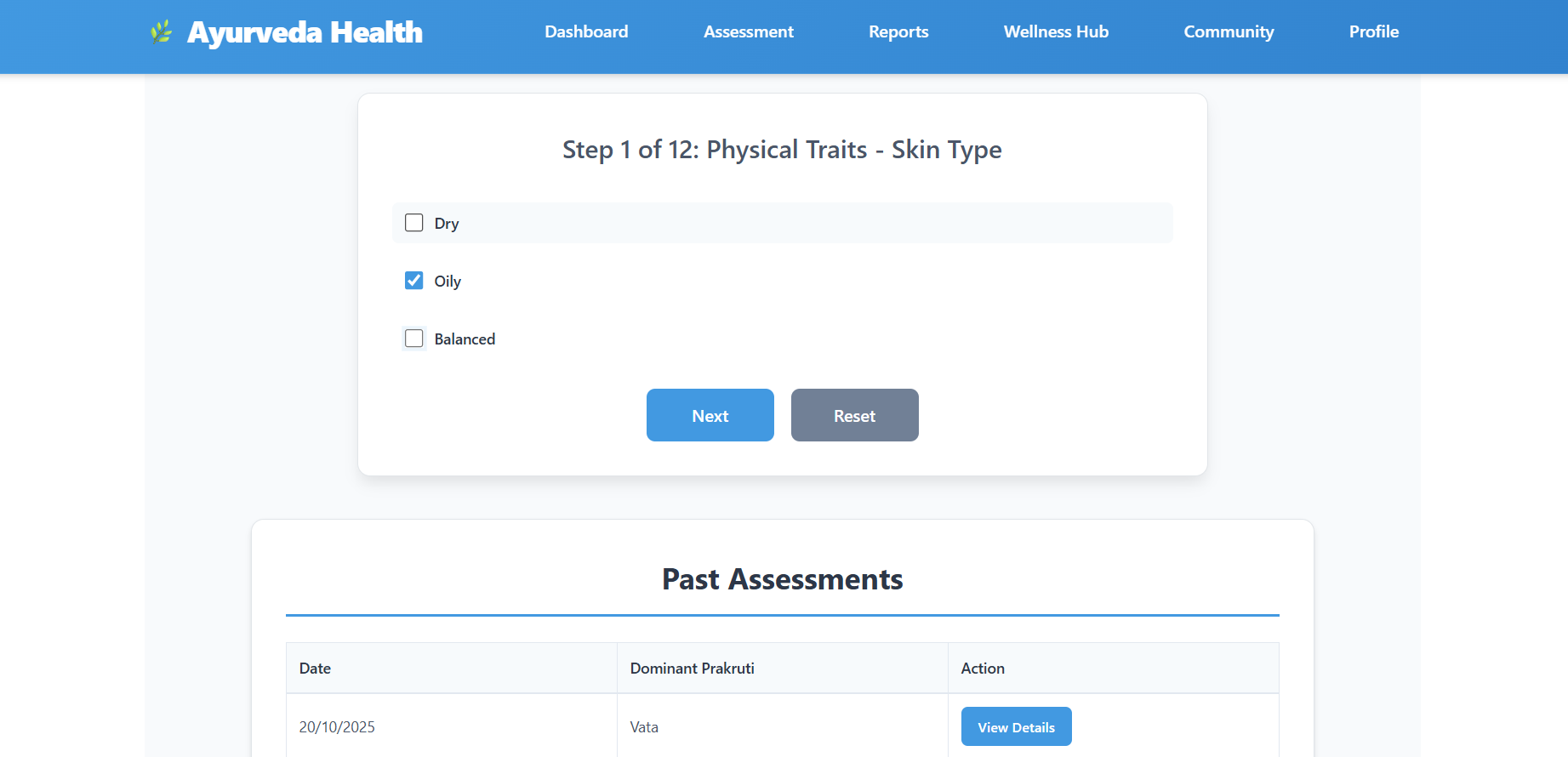
**Login & Registration:**

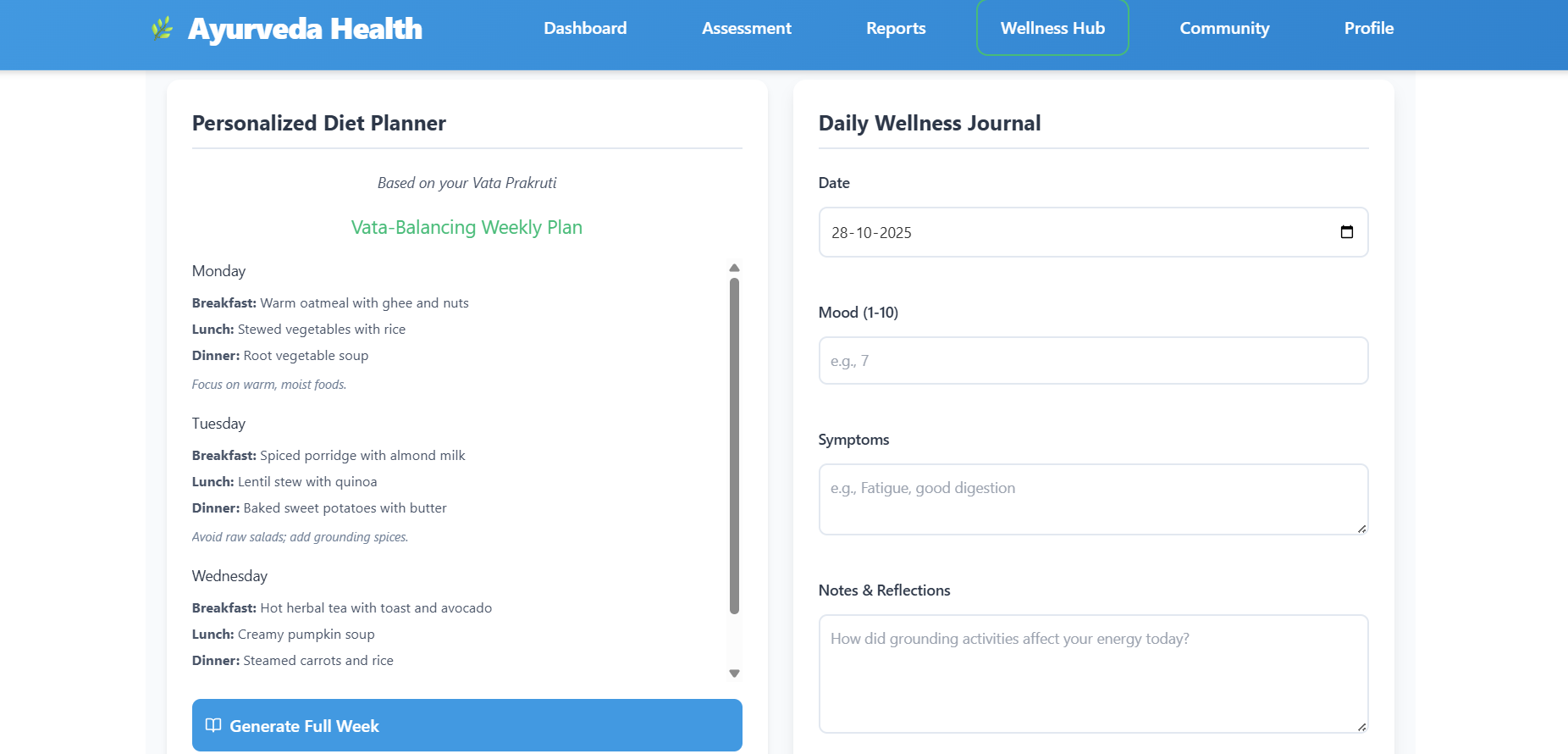
****

**Dashboard:**

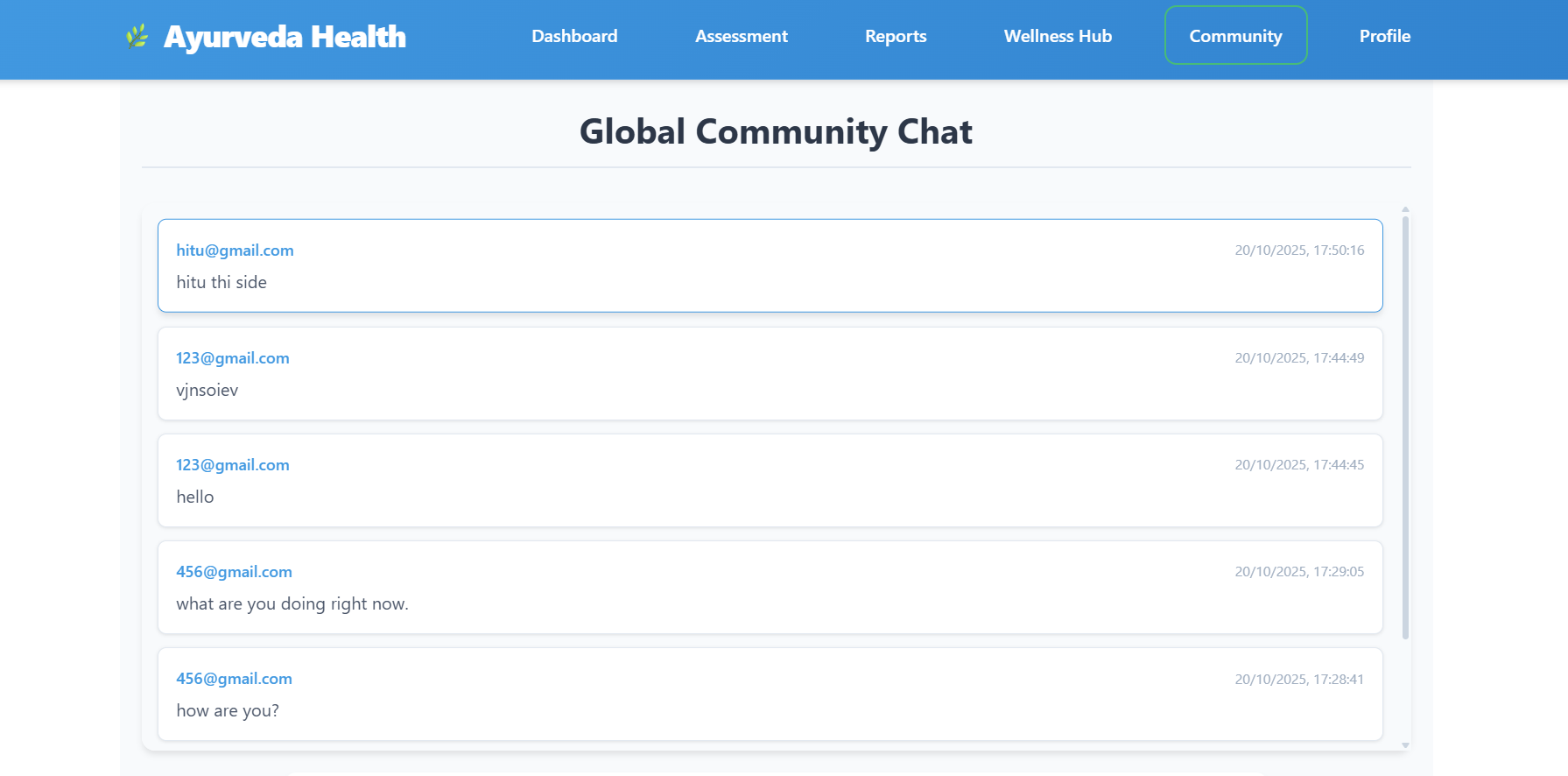


**Assessment**:

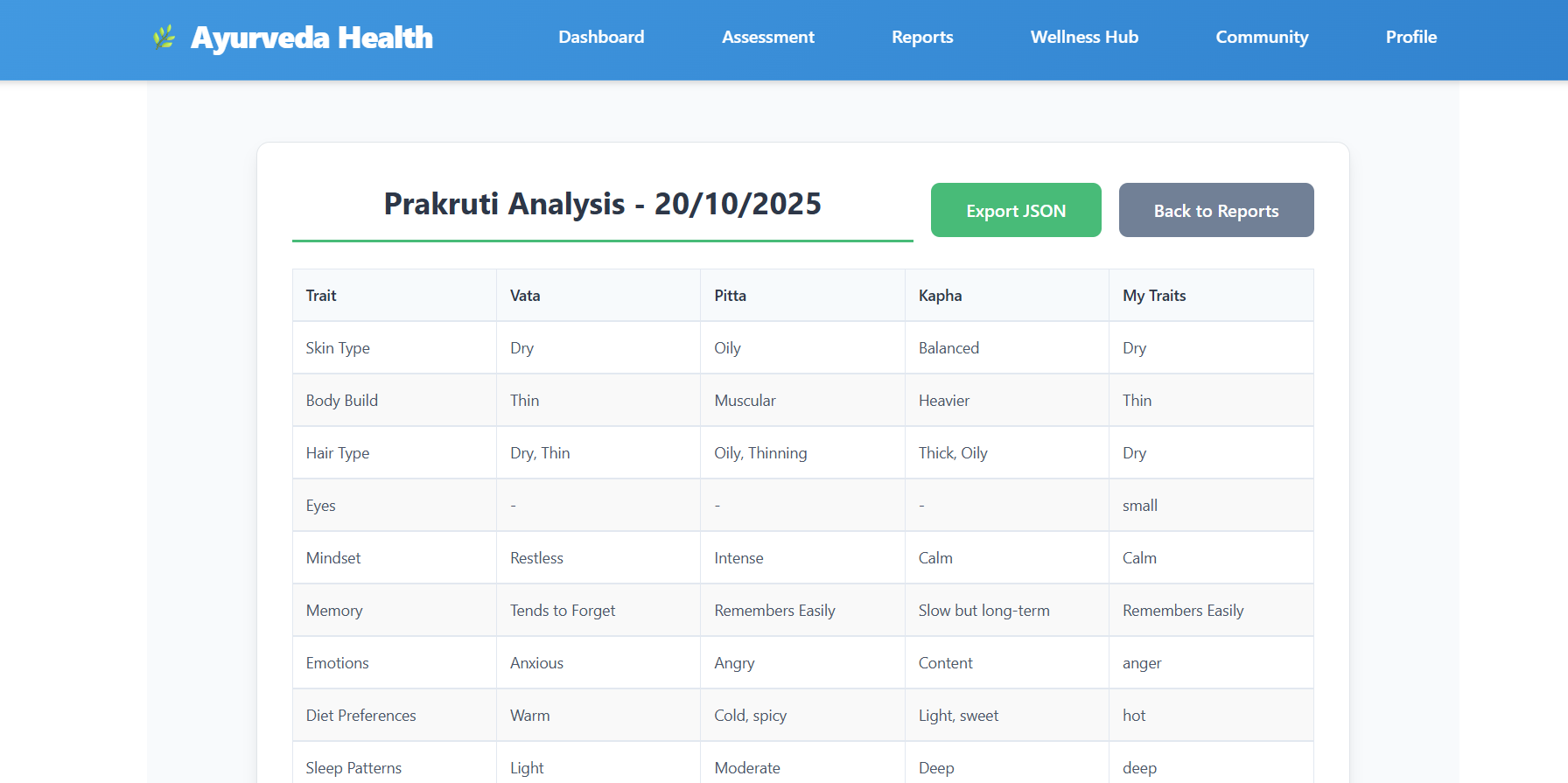


**WellnessHub:**

**Community:**



**Reports :**



**Profile:**

