End-to-End Report: Gemini-Powered Fitness Chatbot API

This document explains the flow of a Flask-based fitness chatbot API, integrated with the Gemini Flash model and Node.js backend.

Components:

- Flask backend with Gemini LLM
- REST API for query or plan generation
- Audio and Text query support
- Node.js backend as data source

API Routes:

---

1. GET `/` - Health check
2. POST `/api/generate_plan` - Auto-generates a 7-day workout plan using Gemini
3. POST `/api/query` - Accepts either text or base64-encoded audio, responds with Gemini output

Data Flow (Backend)

---

1. Fetch user data via `loader.py` from Node.js endpoints:
- /api/users
- /api/DailyProgressLog
- /api/WorkoutPlan
- /api/exercises
2. Construct full user context for Gemini.
3. Prompt Gemini with query + user context.
4. Return JSON response to client.

Audio Handling Flow

- Base64 WAV audio received via /api/query
- Converted to audio stream -> speech-to-text -> passed to Gemini

Required Python Packages

- Flask
- Flask-Session
- requests
- google-generativeai
- python-dotenv
- pytz
- SpeechRecognition
- pydub
- gtts

Tech Stack Summary:

- Gemini 1.5 Flash for LLM response
- Node.js backend for user/workout/log data
- Audio transcription using SpeechRecognition
- JSON-based API response

Output:

- Personalized 7-day workout and nutrition plans
- Smart answers to health/fitness queries
- Audio/text support