FitMate - AI Fitness Coach Agent

Graduation Project Documentation

Student Name: Moamen **Project Title:** FitMate: Full Fitness Coach Agent **Date:** November 30, 2025

Table of Contents

1. Project Planning

1.1 Introduction

FitMate is an advanced AI-powered fitness coaching application designed to democratize access to personalized health guidance. In a world where personal trainers and nutritionists can be prohibitively expensive, FitMate leverages Large Language Models (LLMs) and Retrieval-Augmented Generation (RAG) to provide expert-level advice on workouts, nutrition, and recovery 24/7.

1.2 Problem Statement

- **Accessibility:** Quality fitness coaching is often expensive and inaccessible to the general public.
- **Information Overload:** The internet is flooded with conflicting health advice, making it hard for beginners to know where to start.
- **Personalization:** Generic workout apps lack the ability to adapt to specific user injuries, equipment availability, or unique goals.

1.3 Proposed Solution

FitMate solves these problems by providing a conversational AI interface that: * Understands natural language queries in English and Arabic. * Uses **RAG (Retrieval-Augmented Generation)** to ingest specific knowledge (PDFs, research papers) for evidence-based answers. * Performs **Web Search**

to fetch the latest health trends and studies. * Offers a **"Pro" style interface** that mimics premium SaaS platforms.

1.4 Project Scope

- **Core Domain:** Fitness, Bodybuilding, Nutrition, Sleep, Recovery, Mobility.
- **Out of Scope:** Medical diagnosis (non-fitness related), Legal advice, Financial planning.

---

2. Stakeholder Analysis

2.1 Primary Stakeholders (Users)

- **Beginners:** Individuals new to fitness needing step-by-step guidance.
- **Intermediate/Advanced Athletes:** Users looking for specific program optimizations or latest research.
- **Rehab Patients:** Users needing safe exercises around injuries (supported by the "Rehab & Mobility" coaching mode).

2.2 Secondary Stakeholders

- **Developers:** Responsible for maintaining the LLM pipeline, vector store, and UI.
- **Content Providers:** Fitness experts whose PDFs/Guides are uploaded to the RAG system.

2.3 Key Requirements

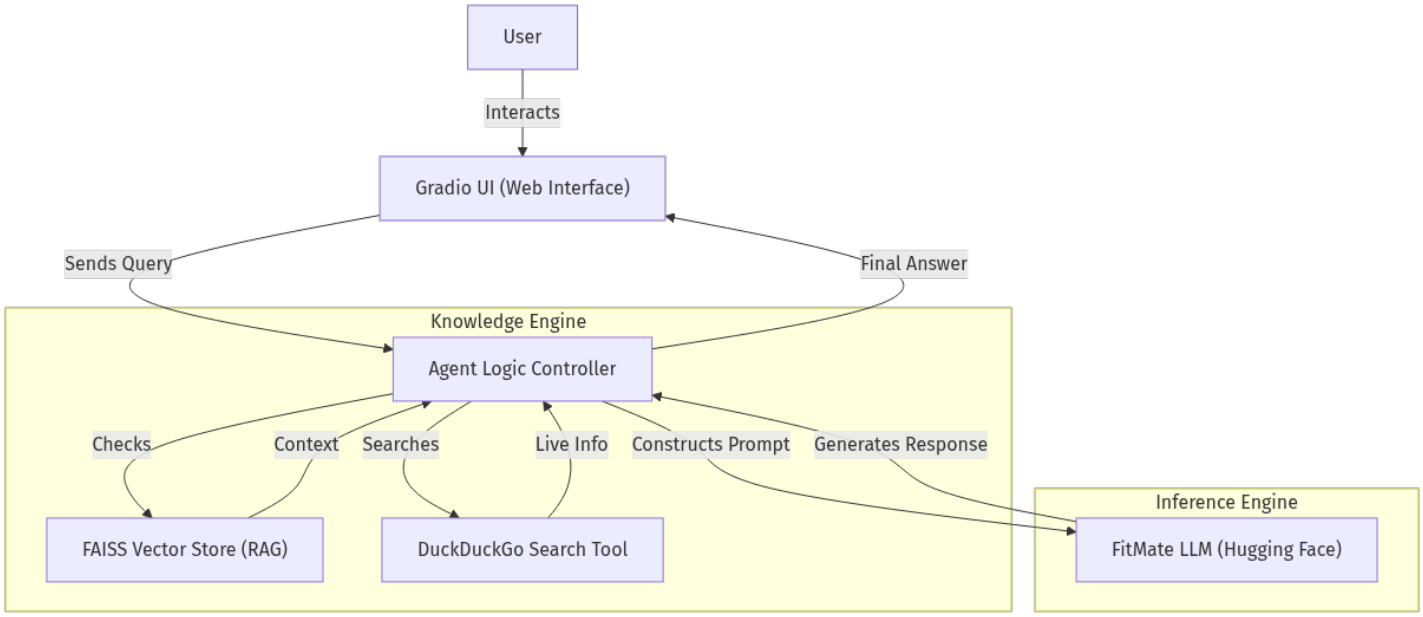
- **Accuracy:** Advice must be safe and effective.
- **Latency:** Responses should be generated reasonably fast (using 4-bit quantization).
- **Usability:** The UI must be intuitive and professional.

---

3. System Architecture & Database Design

3.1 High-Level Architecture

FitMate follows a **Serverless / Local-First** architecture powered by Python.



3.2 Data Persistence Strategy

Unlike traditional CRUD apps, FitMate uses a **Session-State Architecture** optimized for privacy and speed.

- **User Sessions:** Managed via `gradio.State`. Chat history is ephemeral and exists for the duration of the browser session.

  Vector Database (FAISS):
  - **Type:** In-memory Vector Index (`IndexFlatL2`).
  - **Data:** High-dimensional embeddings (384-dim) of uploaded PDF chunks.
  - **Model:** `sentence-transformers/all-MiniLM-L6-v2`.
  - **Purpose:** Allows the AI to "read" and "remember" uploaded documents instantly without an external SQL database.

3.3 Technology Stack

- **Frontend:** Gradio (Python-based UI library).
- **LLM Backend:** `transformers`, `bitsandbytes` (4-bit quantization), `accelerate`.
- **Embedding Model:** `sentence-transformers`.
- **Search:** `duckduckgo-search`.
- **PDF Processing:** `PyPDF2`.

---

4. UI/UX Design

4.1 Design Philosophy: "Professional & Futuristic"

The User Interface is designed to evoke trust and premium quality, moving away from "notebook" aesthetics to a "SaaS Product" look.

4.2 Color Palette

- **Primary Background:** Deep Space Purple (`#0f0518` to `#1a0b2e` gradient).
- **Accent Color:** Electric Violet (`#7c3aed`) & Neon Pink (`#db2777`) for gradients.
- **Text:** Soft White / Lavender (`#e9d5ff`) for readability.
- **Cards/Panels:** Glassmorphism effect (semi-transparent purple with blur).

4.3 Layout Structure

Sidebar Navigation:

- Persistent access to Chat History.
- "New Chat" and "Delete Chat" actions.
- **Pro Badge:** Visual indicator of premium status.

Main Dashboard:

- **Header:** Breadcrumbs and Status Pills (e.g., "■ Turbo Mode").

  Tabs:

  - `■ Home`: Landing page with value proposition and feature grid.
  - `■ Chat`: The core conversational interface.
  - `■ Knowledge Base`: Drag-and-drop PDF upload for RAG.
  - `■■ Settings`: Profile customization and Model parameters.

4.4 User Experience (UX) Flow

1. **Onboarding:** User sees the "Home" tab explaining features.
2. **Action:** User switches to "Chat" or "Knowledge Base".
3. **Interaction:** User types a query -> System shows "Thinking..." -> Response streams back.
4. **Customization:** User can toggle "Web Search" or change "Coaching Mode" (e.g., Fat Loss vs. Muscle Gain) in Settings.

---

5. Implementation Details

5.1 The FitMate Model

- **Base:** Fine-tuned Causal Language Model.
- **Optimization:** Loaded in **4-bit precision (NF4)** using `bitsandbytes` to run efficiently on consumer GPUs (or Colab/Modal free tiers).

- **System Prompting:** A robust system prompt enforces the persona of a "Professional Fitness Coach" and handles language detection (English/Arabic).

5.2 Retrieval-Augmented Generation (RAG) Pipeline

1. **Ingestion:** User uploads a PDF.
2. **Chunking:** Text is split into 500-character chunks.
3. **Embedding:** Chunks are converted to vectors.
4. **Retrieval:** When a user asks a question, the system finds the top 3 most similar chunks.
5. **Generation:** These chunks are injected into the LLM's context window to answer the question accurately.

5.3 Code Structure

- `fitmate.ipynb`: The main entry point containing all logic.
- `run_agent()`: Central function coordinating RAG, Web Search, and LLM generation.
- `Gradio Blocks`: The declarative UI definition code.

---

6. Future Work

- **Mobile App:** Convert the Gradio interface to a React Native mobile app.
- **Voice Interface:** Add Speech-to-Text (Whisper) for voice coaching.
- **Vision Capabilities:** Allow users to upload photos of meals for calorie tracking.
- **User Accounts:** Implement persistent SQL database (PostgreSQL) for long-term history storage.