Evidence-Based Fitness Q&A Chatbot: Leveraging PubMed Research for Scientific Fitness Guidance

Satya Harish, Rahul Gudivada, Elenta Suzan Jacob
Northeastern University – DS 5110: Essentials of Data Science

Fall 2025

1 Project Overview and Objectives

The fitness industry suffers from widespread misinformation, with conflicting advice often leading to ineffective training methods and potential injuries. Our project addresses this by developing an AI-powered fitness chatbot that provides evidence-based advice exclusively from peer-reviewed PubMed research. By integrating PubMed's comprehensive scientific database with Claude AI's natural language processing capabilities, we create a trusted resource for science-informed fitness decisions.

Primary Goal: Build an intelligent chatbot that answers fitness questions about nutrition, exercise, recovery, and injury prevention using only peer-reviewed research with proper academic citations.

Key Objectives:
(1) Implement a pipeline retrieving and synthesizing PubMed research into accessible answers,
(2) Develop a PostgreSQL caching system to minimize API calls and improve response times (target 60%+ cache hit rate),
(3) Create natural language processing for query interpretation and synonym mapping,
(4) Implement quality scoring prioritizing high-impact research (meta-analyses, recent publications, high-impact journals),
(5) Deploy a Streamlit web interface for natural language interactions.

Success Metrics: Response accuracy with 100% valid citations, response time under 5 seconds for cached queries and under 15-20 seconds for new queries, 60%+ cache hit rate, proper citation formatting in all responses, and strict PubMed rate limit compliance (3 requests/second).

Scope: In scope includes text-based Streamlit interface, PubMed E-utilities API integration, local PostgreSQL caching, rate-limiting mechanisms, NLP for question understanding, and coverage of strength training, cardio, nutrition, recovery, and injury prevention. Out of scope includes personalized workout plans, user accounts, mobile apps, multi-language support, and cloud deployment.

2 Team Composition and Member Contributions

Our three-member team brings complementary skills aligned with specific project components, ensuring efficient development across the data science pipeline.

2.1 Satya Harish

Expertise: PostgreSQL design and optimization, advanced SQL, schema design, indexing strategies, cache management.

Responsibilities: Design cache-first schema with five tables (Papers, Queries, Query-Paper Map-

pings, API Logs, Synonyms); create optimized indexes for full-text search using PostgreSQL's tsvector and GIN indexing; implement cache eviction removing papers unused 50+ days; develop 8+ analytical SQL queries for performance monitoring; optimize database through query tuning.

Learning: PostgreSQL full-text search (tsvector/tsquery, ts_rank) – 3-4 days.

2.2 Elenta Suzan Jacob

Expertise: Python programming, REST API integration, ETL with pandas, rate limiting, error handling, data validation, automated scheduling.

Responsibilities: Integrate PubMed E-utilities API via Biopython to fetch top 10 papers prioritizing meta-analyses; build cache-check logic (database first, API on miss); track paper usage and auto-cache at 20-30 usage threshold; implement quality scoring based on impact factor, recency, and study type; maintain synonym dictionary; conduct integration testing with 20+ questions; implement rate limiting (3 req/sec) and error handling with retry logic; run daily cleanup jobs; coordinate end-to-end pipeline integration.

Learning: Biopython for PubMed XML parsing and E-utilities (eSearch, eFetch) – 2-3 days.

2.3 Rahul Gudivada

Expertise: Python programming, API integration, frontend development, UI design, analytics implementation.

Responsibilities: Build Streamlit chat interface with loading indicators; integrate Claude API for NLP and response generation; design prompt templates communicating questions and research context to Claude; implement sentence-transformers for semantic similarity matching of queries to cached questions; coordinate cache lookups and API calls; format papers into Claude prompts; parse Claude responses with proper citations; log usage metrics and collect user feedback; create documentation (README, setup guide, user guide); develop presentation materials and demo video.

Learning: Streamlit framework and sentence-transformers – 2-3 days.

3 Technical Architecture: Tools and Technologies

Our technology stack leverages Python expertise while incorporating specialized tools for database management, API integration, and NLP.

3.1 Backend & Data Pipeline

Python 3.9+: Primary language for team proficiency and extensive data science libraries.
PostgreSQL 14+: RDBMS with robust full-text search, excellent for structured metadata and unstructured abstracts.
Biopython: PubMed E-utilities API wrapper with XML parsing for eSearch and eFetch endpoints.
psycopg2: PostgreSQL adapter for database connections and queries.
pandas: ETL operations transforming PubMed XML into structured data.

3.2 LLM & Natural Language Processing

Anthropic Python SDK: Claude API client for research synthesis and response generation.
sentence-transformers: Semantic embeddings enabling similarity-based cache lookups matching questions despite wording differences.
NLTK: Lightweight NLP for keyword extraction, stopword removal, and text normalization.

3.3 Frontend & Development Tools

Streamlit: Python-native web framework with built-in chat components, enabling interactive UI without HTML/CSS/JavaScript.
Git/GitHub: Version control and collaboration.
pytest: Testing framework.
python-dotenv: Secure API key management.
DBeaver: Database visualization.
Postman: API testing.

3.4 Technical Rationale

This stack was selected for:
(1) Leveraging Python expertise across team members,
(2) PostgreSQL's full-text search perfect for research abstracts,
(3) Streamlit eliminating separate frontend learning while maintaining professional quality,
(4) Strong documentation and community support for all tools.

3.5 API Integration Architecture

PubMed E-utilities API: Provides programmatic access to 36+ million biomedical citations. We use eSearch for finding article IDs by keywords, then eFetch for retrieving metadata (titles, abstracts, authors, dates, journals). Respects 3 requests/second rate limit through throttling.

Claude API (Anthropic): Provides advanced NLP capabilities. Receives user questions and research abstracts, then synthesizes findings across studies into clear, evidence-based responses with citations. Acts as the translation layer between scientific research and accessible fitness advice.

3.6 System Workflow

User questions follow this pipeline:
(1) **Query normalization** – remove stopwords, apply synonyms, extract keywords;
(2) **Cache lookup** – check PostgreSQL for similar cached questions using semantic similarity;
(3) **PubMed search** – on cache miss, query API for top 10 papers;
(4) **Quality scoring** – rank by impact factor, recency, study type;
(5) **Claude synthesis** – send question and papers to Claude for response;
(6) **Cache storage** – store new papers and mappings;
(7) **Response delivery** – display formatted answer with citations via Streamlit.

This cache-first architecture minimizes API costs, improves response times for common questions, and scales efficiently as the knowledge base grows organically based on user needs.

4 Conclusion

Our evidence-based fitness chatbot applies data science principles to combat real-world misinformation. By integrating PubMed's research database with Claude's language capabilities, we democratize scientific knowledge and promote evidence-based fitness practices. The project spans database design, ETL engineering, API integration, NLP, and UI development, providing hands-on experience building an end-to-end data product. Over five weeks, we will deliver a functional chatbot answering fitness questions with scientific rigor, maintaining academic integrity through proper citations, and serving as a model for combating misinformation through intelligent data systems.