# FitFact: Research-Backed Fitness Q&A Chatbot

Satya Harish(002300329), Elenta Suzan Jacob(002530281), Rahul Gudivada(002560822)
DS 5110 - Essentials of Data Science

## 1 Project Overview

**Primary Goal:** Build an AI-powered fitness chatbot providing evidence-based advice exclusively from peer-reviewed PubMed research, combating widespread fitness misinformation.

**Expected Outcomes:** Functional chatbot answering fitness questions (nutrition, exercise, recovery) with responses backed by scientific research, proper citations (100% rate), intelligent caching system, user-friendly web interface, and PubMed API rate limit compliance.

**Success Metrics:** Response accuracy validated against expert answers; response time under 5 seconds (cached) and 15-20 seconds (new queries); 60%+ cache hit rate; proper citation formatting in all responses.

### 1.1 Scope

**In Scope:** Streamlit web interface, PubMed E-utilities API integration, PostgreSQL caching database, rate-limiting mechanism, NLP for query understanding, automated citation generation, quality scoring (prioritizing meta-analyses), fitness domains (strength training, cardio, nutrition, recovery, injury prevention).

**Out of Scope:** Personalized workout plans, payment/authentication, mobile app, multi-language support, cloud deployment.

**Architecture:** Cache-first system: (1) User query → (2) PostgreSQL cache check via semantic similarity → (3) Cache hit returns stored response → (4) Cache miss triggers PubMed search (top 10 papers) → (5) Abstracts sent to Claude API → (6) Claude synthesizes cited response → (7) Response cached → (8) Display to user with source links.

## 2 Team Responsibilities

### 2.1 Satya Harish - Database Engineer & Data Architect

**Skills:** PostgreSQL design/optimization, advanced SQL, cache management, ER modeling, indexing.

**Tasks:** Design cache-first schema (papers, queries, responses, mappings, API logs, synonyms); implement full-text search (tsvector, tsquery, GIN indexes); create efficient indexes; develop cache eviction queries (remove unused 50+ days papers); write 8+ SQL analytical reports; optimize performance.

### 2.2 Elenta Suzan Jacob - Data Pipeline Engineer & ETL Specialist

**Skills:** Python, REST API integration, ETL pipelines (pandas), rate limiting, error handling, data validation, automation.

**Tasks:** Integrate PubMed API using Biopython (XML parsing); build cache-check logic; fetch/prioritize top 10 papers (meta-analyses first); implement quality scoring (journal impact, recency, study type); track paper usage (times_used counter, auto-cache at 20-30 threshold); maintain synonym dictionary; run daily cleanup jobs.

### 2.3 Rahul Gudivada - NLP Engineer & Frontend Developer

**Skills:** Python, API integration, LLM prompt engineering, UI design, analytics.

**Tasks:** Build Streamlit chat interface; integrate Claude API; use sentence-transformers for semantic similarity (improve cache hits); coordinate cache lookups/API calls; engineer prompts for cited responses;

parse/display Claude responses with citations; implement analytics (query logging, paper tracking, feedback); use NLTK for keyword extraction.

## 3 Technical Stack

**Language:** Python 3.9+ (team's strength)

**Backend/Data:** PostgreSQL 14+ with pg_trgm (full-text search), Biopython (PubMed API wrapper/XML parsing), psycopg2 (PostgreSQL adapter), pandas (ETL).

**LLM/NLP:** Anthropic Python SDK (Claude API), sentence-transformers (semantic similarity for cache optimization), NLTK (keyword extraction, stopword removal).

**Frontend:** Streamlit (Python-native, rapid prototyping, chat components).

**Dev Tools:** Git/GitHub (version control), pytest (testing), python-dotenv (API keys), DBeaver (DB visualization), Postman (API testing).

**Rationale:** Leverages Python expertise across team; PostgreSQL ideal for abstract searching; Streamlit eliminates separate frontend framework; strong documentation/community support.

**Learning Requirements:** Biopython-Elenta (2-3 days), PostgreSQL full-text search-Satya (3-4 days), Streamlit-Rahul (2-3 days), sentence-transformers-Rahul (1-2 days). Total: 1-2 weeks parallel learning.

## 4 Data Sources and Management

**Primary Source:** PubMed E-utilities API (https://www.ncbi.nlm.nih.gov/books/NBK25501/) - open-access peer-reviewed biomedical literature. No pre-existing dataset; system builds knowledge base dynamically from user queries.

**Usage Policies:** 3 requests/second (no key), 10 requests/second (with key). Rate limiting implemented; caching minimizes redundant calls.

**Collection Strategy:** Focus on resistance training, cardio, protein/creatine supplementation, recovery, injury prevention. Prioritize meta-analyses, systematic reviews, RCTs, recent publications (last 5 years).

**Quality Scoring:** Based on study type (meta-analyses highest), publication recency (5 years), journal impact factor, citation count.

**Governance:** Preserve all PubMed citations/metadata for traceability; audit log tracks updates/deletions; cache eviction (unused 50+ days); synonym dictionary for query normalization.

## 5 Development Timeline

**Week 1-3 (Foundation):** Database schema/implementation, PubMed API POC, Claude API testing, GitHub setup. *Milestone: Database operational, APIs connected.*

**Week 4-6 (Data Pipeline):** Automated ETL, 500+ papers collected, quality scoring, semantic search. *Milestone: End-to-end pipeline functional.*

**Week 7-9 (Interface):** Streamlit interface, cache optimization, citations, misinformation detection, analytics. *Milestone: MVP chatbot functional.*

**Week 10-12 (Testing):** Unit tests, accuracy validation, analytics dashboard, documentation, presentation. *Milestone: Project complete.*

## 6 Risk Management

**PubMed API Rate Limits:** Aggressive caching, batch requests.

**Claude API Costs:** Usage limits, prompt optimization.

**Data Quality:** Validation checks, quality scoring.

**Response Accuracy:** Expert validation, testing.

**Learning Curve:** Week 1 learning sprint, pair programming.

**Scope Creep:** Strict MVP adherence.

## 7 Project Tracking and Repository

**Progress Tracking:** Excel tracker with Task Name, Assigned To, Start/End Dates, Status, Dependencies, Notes. Updated weekly, included in submissions.

**GitHub Repository:** https://github.com/rahulg2469/FitFact-Chatbot

**Repository Contents:** Source code (database, ETL, LLM, frontend), database schema/scripts, requirements.txt, .env.example, README (setup instructions), this PDF, Excel tracker, documentation/ER diagrams.

**Version Control:** `main` (production), `dev` (integration), feature branches. Pull requests required for `main` merges, code reviews mandatory.

## 8 Submission Verification

- Three-page project introduction
- Project scope, objectives, outcomes defined
- Team contributions documented
- Technical stack listed
- Data source specified (PubMed API)
- Excel progress tracker included
- GitHub repository link provided
- PDF uploaded to repository
- Setup instructions documented
- Risk management outlined

**Readiness:** Comprehensive project plan complete covering objectives, scope, team composition, technical stack, timeline, and risk management. Ready for implementation with all foundational planning complete and team prepared to execute assigned responsibilities. All requirements met; documentation thorough; team committed to delivering functional, high-quality fitness chatbot combating misinformation through evidence-based research.

**GitHub Repository URL:** https://github.com/rahulg2469/FitFact-Chatbot