RAG Chatbot Evaluator MVP - Detailed Report

# Overview

This MVP evaluates RAG chatbot answers against ground-truth context using a judge LLM. It scores correctness, completeness, and relevance on a 1-5 scale, provides reasoning, and aggregates results into a failure mode report.

# Implementation Details

- Language: Python  
- Libraries: openai (for LLM API), python-docx (for this report)  
- Input: JSON file with mock data (questions, RAG answers, ground-truth)  
- Output: evaluations.json (detailed scores/reasoning), failure\_report.json (aggregated low scores)  
- LLM: OpenAI GPT-4 (or mock mode for testing)  
- No front-end; command-line script

# Mock Data

Sample data includes 3 questions with varying answer quality:

1. Capital of France - Correct answer

2. Author of Pride and Prejudice - Correct answer

3. Boiling point of water - Incorrect answer (90°C instead of 100°C)

# Evaluation Process

For each RAG answer:  
- Prompt LLM to score correctness, completeness, relevance (1-5)  
- Provide reasoning for each score  
- Aggregate low scores (<3) into failure report

# Testing Results

- Happy path: Script loads data, evaluates answers, generates reports  
- Error handling: Mock mode for missing API key  
- Output verification: Scores and reasoning match expected values  
- Failure report: Correctly identifies low-scoring answers

# Example Outputs

Evaluations JSON includes scores like:

Correctness: 5 (accurate), Completeness: 4 (lacks context), Relevance: 5 (direct)

Failure report highlights issues: e.g., "correctness: Incorrect boiling point"

# Setup and Usage

1. Install dependencies: pip install -r requirements.txt  
2. Set OPENAI\_API\_KEY (optional for mock mode)  
3. Run: python evaluator.py  
4. Check output files: evaluations.json, failure\_report.json

# Limitations and Extensions

- Limited to 3 samples; can scale to more data  
- Uses single LLM; could add multiple judges  
- No UI; could add web interface  
- Simple aggregation; could add charts/visualizations  
- Mock mode for testing; real API for production