# FinExpert: Financial Reasoning Assistant

**1.Executive Summary**

**FinExpert** is an intelligent, LLM-powered application designed to analyze complex financial documents and answer multi-step numerical questions with precision. Developed to assist financial analysts, auditors, and decision-makers, the system leverages large language models to process structured and semi-structured data—such as tables, textual narratives, and figures—to generate accurate, context-aware answers.

This report provides a comprehensive overview of the system architecture, core components, LLM integration, evaluation strategy, performance results, and identified areas for improvement.

**2. System Architecture**

The Financial Reasoning Assistant follows a **modular and extensible architecture** composed of four key layers:

* **Frontend**: A lightweight, interactive interface built using Streamlit, enabling user interaction and real-time output.
* **Backend**: A Python-based engine responsible for data ingestion, preprocessing, prompt generation, and output formatting.
* **LLM Integration**: Dual support for both OpenAI and Groq language model APIs allows flexibility in inference based on accuracy, latency, or cost needs.

**3. Key System Components**

**3.1 Frontend Interface**

The user interface, built in **Streamlit**, provides a clean, user-friendly experience. Key functionalities include:

* Upload and preview of structured financial JSON data
* Natural language question submission
* Real-time numeric answer display
* Chat history tracking for multi-turn conversational queries

A screenshot of a computer

AI-generated content may be incorrect.

**3.2 Backend Processing**

The backend performs critical processing tasks that ensure both accuracy and resilience:

* Intelligent parsing of financial statements (tables and narrative)
* Normalization of numeric formats (e.g., $, %, millions/billions)
* Execution of arithmetic programs generated by the LLM
* Exception handling and structured logging for robust debugging

**4. Language Model Integration**

The assistant integrates two high-performance language models using their API keys

**🔹 OpenAI GPT-4 Turbo**

**🔹 Groq (LLM Accelerator)**

**5. Evaluation & Performance**

To evaluate the system’s effectiveness, a dedicated script was created using the **ConvFinQA dev.json** dataset as a benchmark. The evaluation pipeline uses **GPT-4 Turbo** with the same prompt structure as the production system to ensure consistency.

Each data entry is processed by constructing a prompt that combines pre\_text, post\_text, and table\_ori data when available. If not, a fallback representation of the table is used.

**Key Features of the Evaluation Process:**

* **Answer Normalization**: Strips whitespace and formatting symbols ($, %, commas), converts number words ("million", "billion") into numeric form, and supports both numerical and string comparisons.
* **Context Building**: Merges all available context and table sources intelligently for optimal model grounding.
* **Accuracy Tracking**: Measures exact match accuracy against ground truth, logs mismatches, and saves all results in JSON format for post-analysis.
* **Output Summary**: Displays key stats, highlights incorrect predictions, and offers insights into common reasoning errors.

To run the evaluation, developers must ensure that the .env file contains valid API keys for both OpenAI and Groq.

**📊 Key Evaluation Metrics:**

* **Accuracy**: 66.61%
* **Processing Success Rate**: 100.00%
* **Error Rate**: 0.00%
* **Total Questions Evaluated**: 542
* **Correctly Answered**: 361

These results indicate solid baseline performance, especially considering the multi-hop reasoning and context sensitivity required by the dataset.

**6. Areas for Improvement**

While the Financial Reasoning Assistant performs well on many fronts, several enhancements could improve its accuracy, reliability, and flexibility:

**6.1 Answer Normalization**

* Enhance support for numeric variants (e.g., thousands, millions, different currency symbols)
* Improve conversion logic between absolute and relative units
* Refine percent/rate-of-change calculations to match real-world standards

**6.2 Contextual and Temporal Reasoning**

* Expand handling of multi-year trends and rolling aggregates
* Improve comprehension of footnotes, merged cells, and explanatory narrative
* Incorporate richer context memory for long multi-turn interactions

**6.3 Advanced Reasoning Models**

* Explore integration of domain-optimized reasoning models such as **Toolformer**, **TAPAS**, **TUTA**, or **DSPy agents**
* Experiment with **chain-of-thought prompting** combined with **program validators** to catch logical errors
* Incorporate hybrid neural-symbolic reasoning systems for enhanced transparency and traceability

**6.4 Fine-Tuning for Accuracy**

* Fine-tune LLMs (e.g., GPT variants, T5, FLAN) using **ConvFinQA** to increase domain familiarity
* Train models to generate DSLs or calculation chains aligned with financial best practices
* Use reinforcement learning or instruction tuning based on answer correctness to improve consistency

The full implementation of the **FinExpert**, including source code, prompt template, evaluation script is available on GitHub:

🔗 **GitHub Repository**: <https://github.com/your-username/financial-reasoning-assistant>  
*(Replace with your actual repository URL)*

To run the application locally, please refer to the detailed instructions provided in the README.md file located at the root of the repository.