My Take-Home Assignment: Oculis - AI Financial Assistant

Tanvi Pooranmal Meena

02

Architectural Decisions: RAG Implementation

Building a Robust AI Assistant

01

Full-Stack Strategy

Clear separation of concerns with React Frontend and Flask Backend.

02

RAG Approach

Chose Retrieval-Augmented Generation (RAG) over fine-tuning for factual accuracy, scalability, and cost-effectiveness.

03

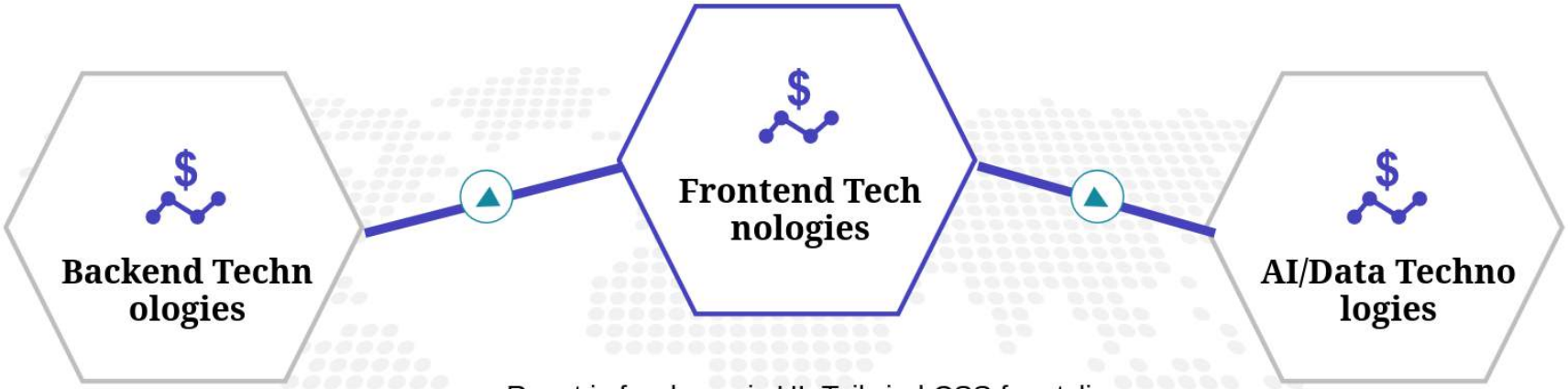
Granular Data Chunking

Broke down financial sheets into labeled facts for precise retrieval.

03

Technology Stack: Components Used

Technologies Utilized



React.js for dynamic UI; Tailwind CSS for styling.

Flask as API server; SQLAlchemy (SQLite) for data persistence; Pandas for data parsing and manipulation.

LangChain for building intelligent applications; ChromaDB for semantic search; Google Gemini API for NLP.

04

Data Ingestion: Handling Diverse Formats

Building the Foundation: Data Pipeline

01 **My Contribution**

Developed backend logic to handle diverse financial file formats.

02 **Key Feature**

Enabled dynamic multi-year data parsing from single files using Pandas.

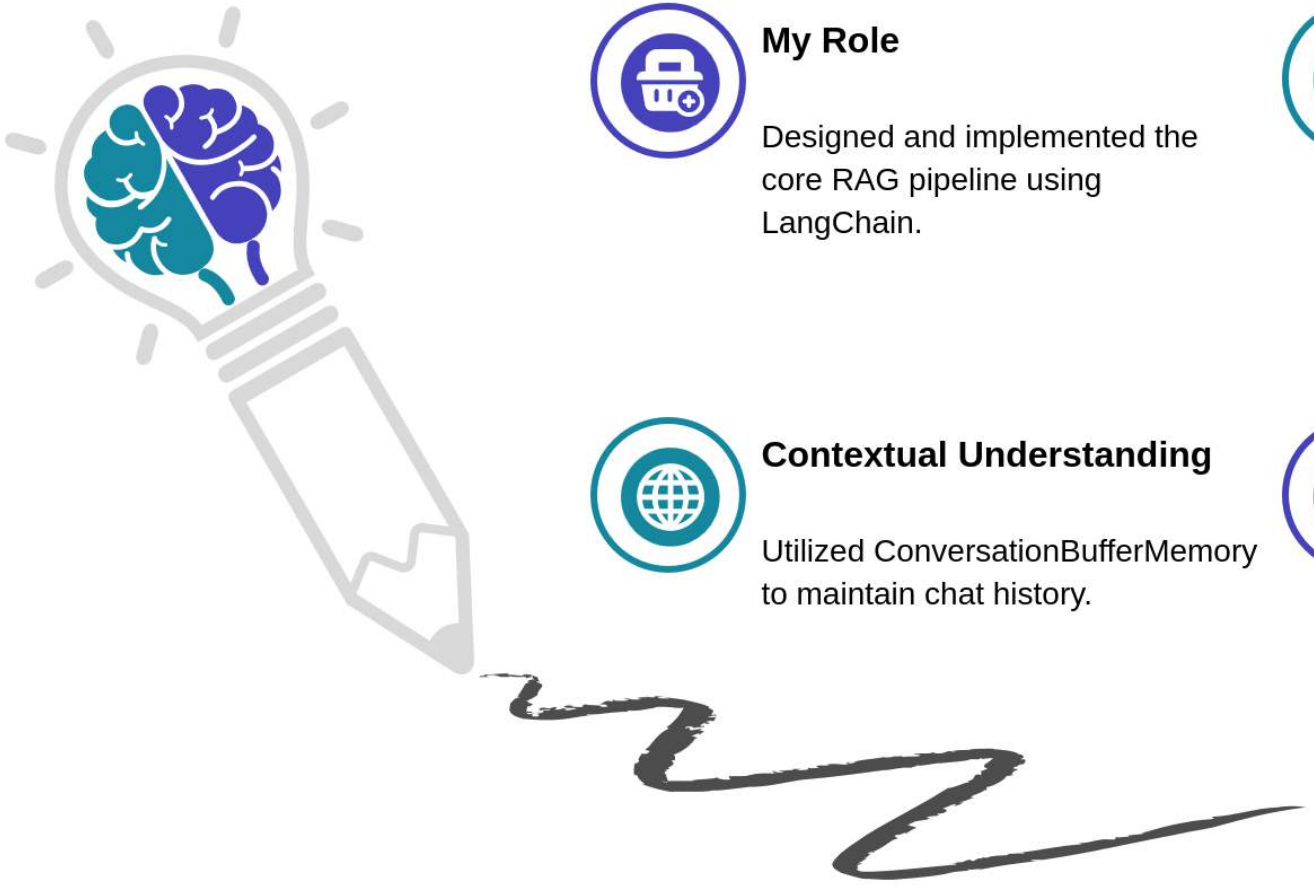
03 **Data Integrity**

Implemented logic to delete/replace existing data, preventing duplicates.

05

AI Interaction: Implementing RAG Pipeline

The Brain: Crafting AI Interaction



My Role

Designed and implemented the core RAG pipeline using LangChain.

Precision Retrieval

Configured ChromaDB with metadata filtering for relevant data search.

Contextual Understanding

Utilized ConversationBufferMemory to maintain chat history.

Critical Prompt Engineering

Refined a restrictive system pro mpt for Gemini to ensure accura te output.

06

Problem-Solving (I): LLM Hallucination

Overcoming Challenges: My Problem-Solving

Challenge 1

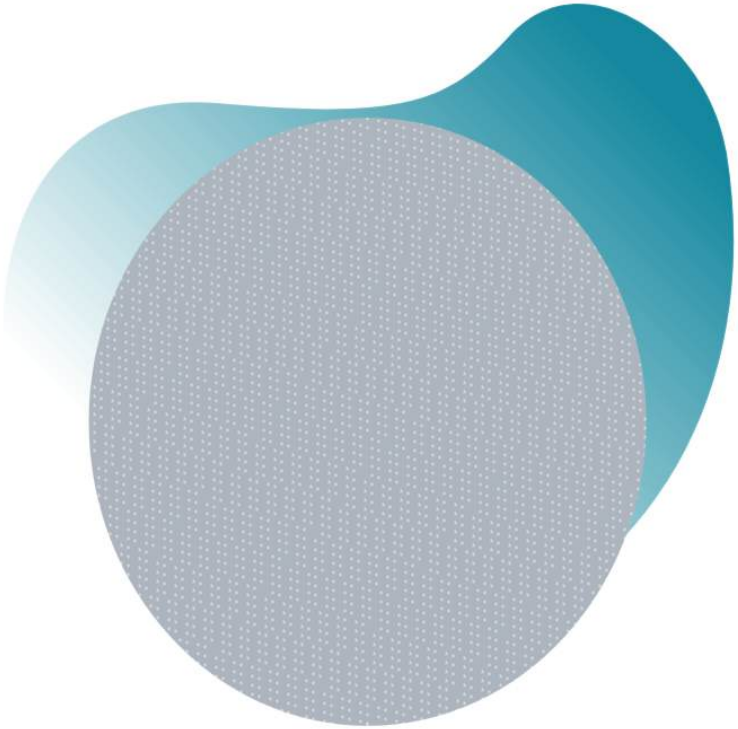
LLM hallucination of numbers despite correct context.

My Solution

Iterated on system prompt to enforce strict context usage.

Learning

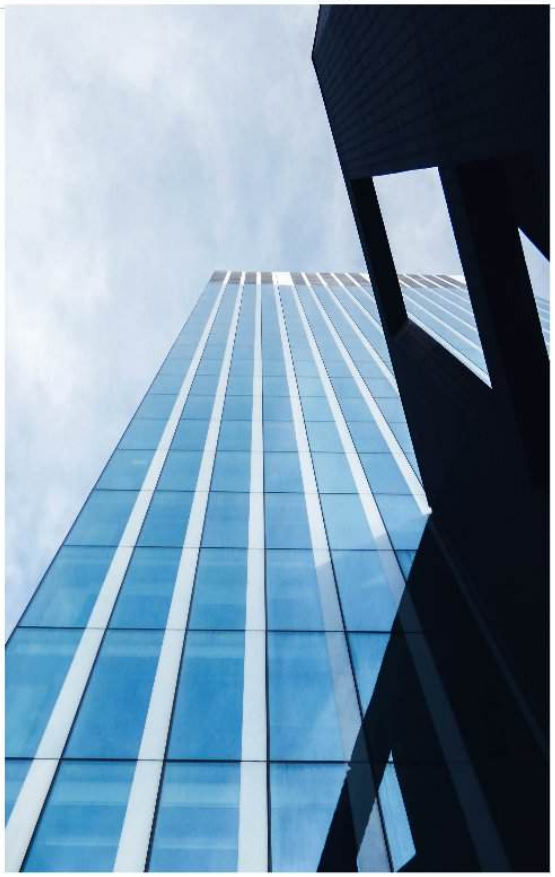
Effective prompt engineering is vital in RAG.



07

Problem-Solving (II): ChromaDB Filtering

Enhancing Usability and UI



01

Challenge 2

Complex ChromaDB filtering syntax.

02

My Solution

Deep-dived into ChromaDB documentation to apply operators correctly.

03

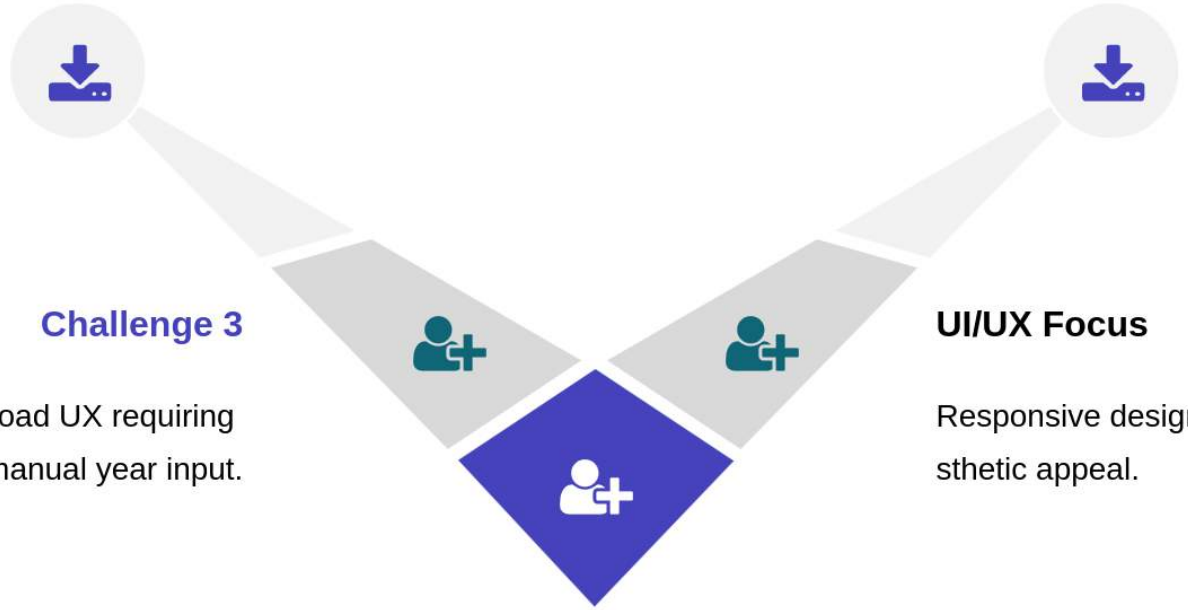
Learning

Understanding library-specific query syntax is critical.

08

UI Enhancements: Improving User Experience

UI & Backend Refinements



Challenge 3

Suboptimal file upload UX requiring manual year input.

UI/UX Focus

Responsive design, clear feedback, and aesthetic appeal.

My Solution

Dynamically detect years within uploaded files using Pandas.

10

10. Future Enhancements: My Visio
n

10.1 Beyond the Take-Home

Data Visualization

Integrate charting libraries for visual trends.

Advanced Querying

Support comparative analysis.

Authentication & Security

Implement robust user management.

Scalability

Containerization and cloud deployment.

Error Handling

Provide specific user feedback for errors.

11

Key Learnings: Personal Growth

My Key Learnings & Growth

RAG System Design

Practical experience in end-to-end RAG pipeline design and implementation.

Full-Stack Integration

Deepened understanding of frontend, backend, database interactions.

Problem-Solving

Stronger debugging skills and approaches to complex issues.

UX-Driven Development

Iterating on design based on user interaction principles.

Proactive Learning

Successfully integrate new technologies within tight timeframe.

12

Conclusion

Thank you for listening.