



University of Sri Jayewardenepura

● Progress Evaluation

Knowledge Graph-based Retrieval-Augmented Generation System for Domain-Specific Information Extraction with Glossary-Aided Responses

Group 15

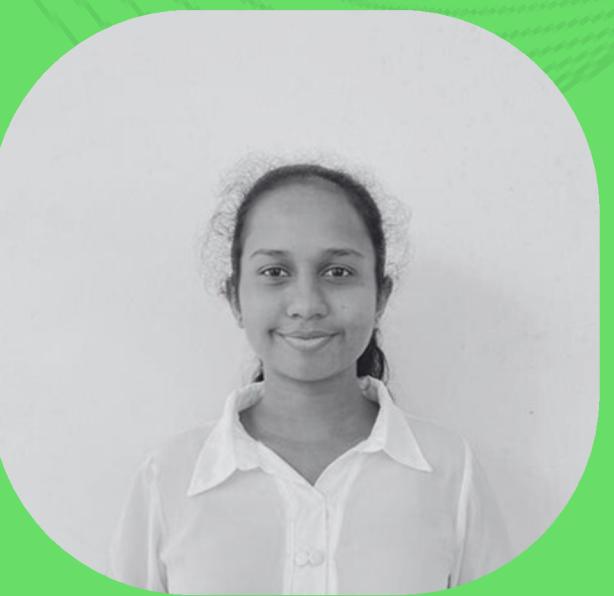


Meet Our Team ↴



Kavinda Maduranga

WEERASINGHA W.G.K.M.
ICT/20/956



Nipuni Nishadini

De Silva K.N.N.C.
ICT/20/826



Dulan Jayawikrama

Jayawickrama D.S.K.
ICT/20/862

Our Supervisors



Main Supervisor

Dr. Chamara Liyanage
Academic Supervisor,
Department of ICT,
University of Sri Jayewardenepura

External Supervisor

Mr. Hiran Wijesingha
Assistant Director IT,
Sri Lanka Tea Board

Content Outline ↴

1. Introduction & Background
2. Research Problem
3. Objectives & Outcomes
4. Methodology
5. Architecture Design
6. Technologies
7. Frontend Design
8. Backend Design
9. TO-DOS



Introduction & Background

AI Powered System

Development of an agent system for interacting domain specific knowledge.

Graph-RAG Model

Retrieval-Augmented Generation with a knowledge graph.

Web Portal

Includes a user-friendly document management portal for easy interaction.

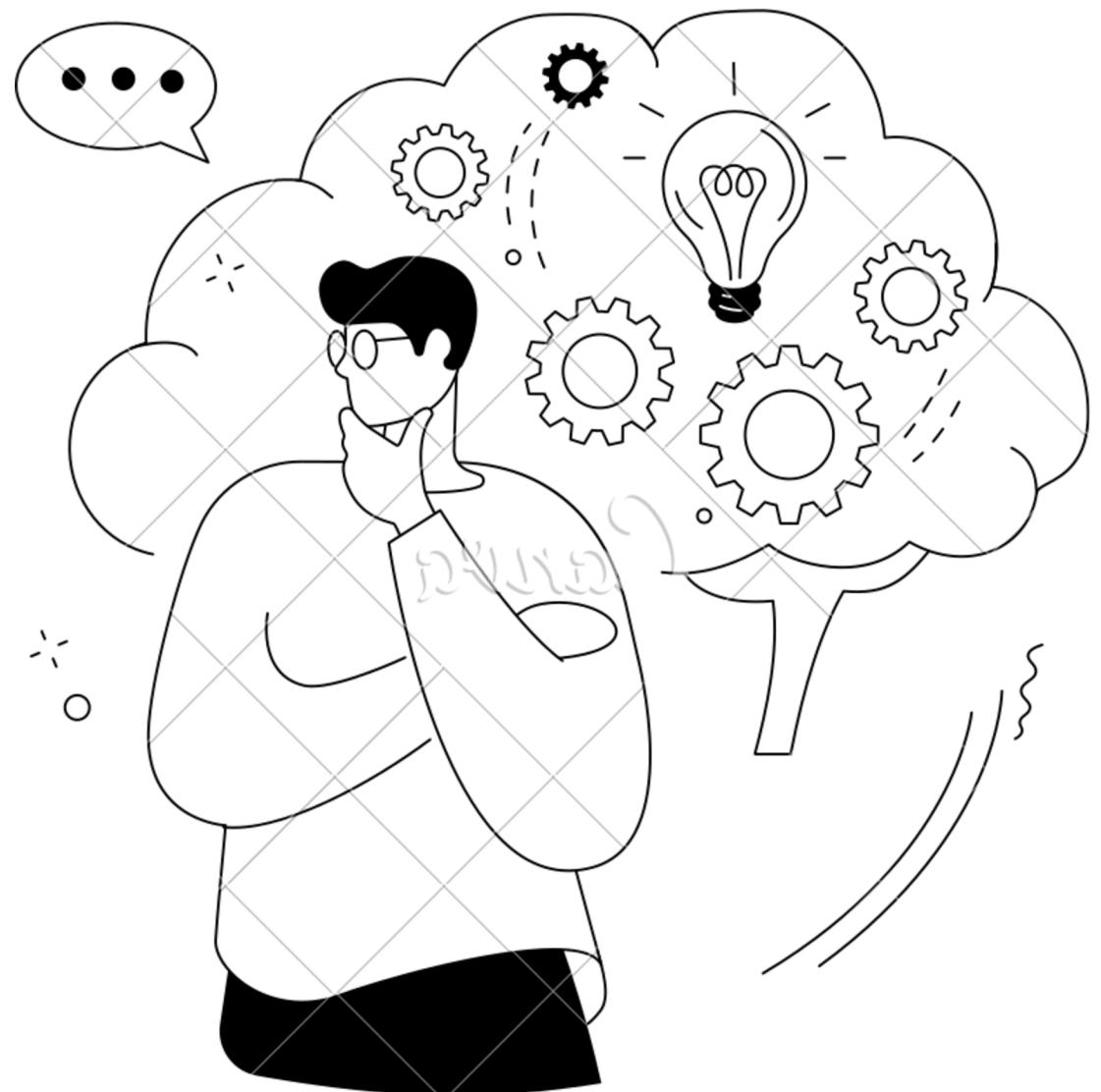
Domain-specific Glossary

Industry-specific terms will be integrated to improve the accuracy of the AI-generated response.



Research Problem

- Organizations with long histories struggle to manage and retrieve critical information from large archives of historical documents, which are essential for decision-making.
- Many still rely on outdated, manual methods, increasing the risk of losing valuable domain expertise as experienced personnel retire.
- This leaves newer employees without the deep knowledge needed for effective decisions. The lack of an efficient knowledge transfer process worsens the issue, creating a knowledge gap.
- An automated system is urgently needed to manage, retrieve, and contextualize this data, ensuring that institutional knowledge is preserved and easily accessible for future decision-making.





Objectives & Outcomes ↴

Main objective

Develop a graph RAG system integrated to improve organizations' ability to manage, retrieve, and utilize historical documents. This system will aid decision-making by providing accurate, contextually relevant information based on decades of documented institutional actions, decisions, and domain-specific knowledge.

Web-based document management portal

LLMs integration for context-aware responses

Structured knowledge base from historical document

Domain-specific glossary

User-friendly interface

Systems future needs (Scalability and adaptability)



Methodology ↓

1 Data Collection and Preprocessing

- Document Management Portal
- Text Extraction

2 Knowledge Graph Creation

- Knowledge Graph Construction
- Automation

3 Graph based Context Retrieval

- Context retrieval by graph traversal

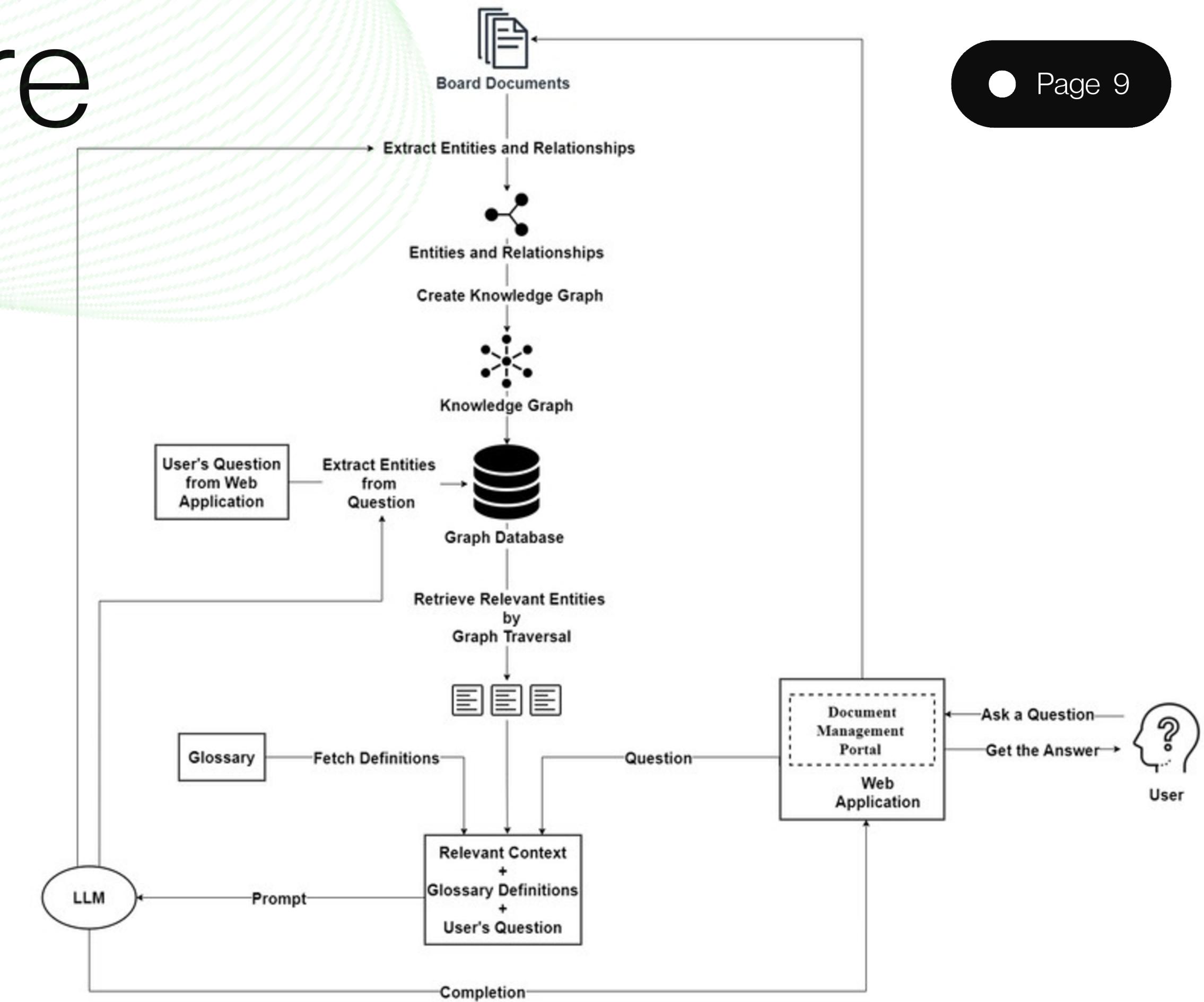
4 Integration with LLM and Glossary

- LLM Integration
- Glossary Integration

5 Answer Generation and Delivery

- LLM Processing
- Frontend Delivery

Architecture Design





Technologies ↴



Graph database management system designed to store and query connected data efficiently.



A framework for building applications that integrate LLMs with external data sources.



Large language model developed by Meta for various natural language processing tasks.



A cloud platform for building, deploying, and managing applications globally. It offers scalable and secure services for various needs.



Popular JavaScript library for building interactive user interfaces, especially for web applications.

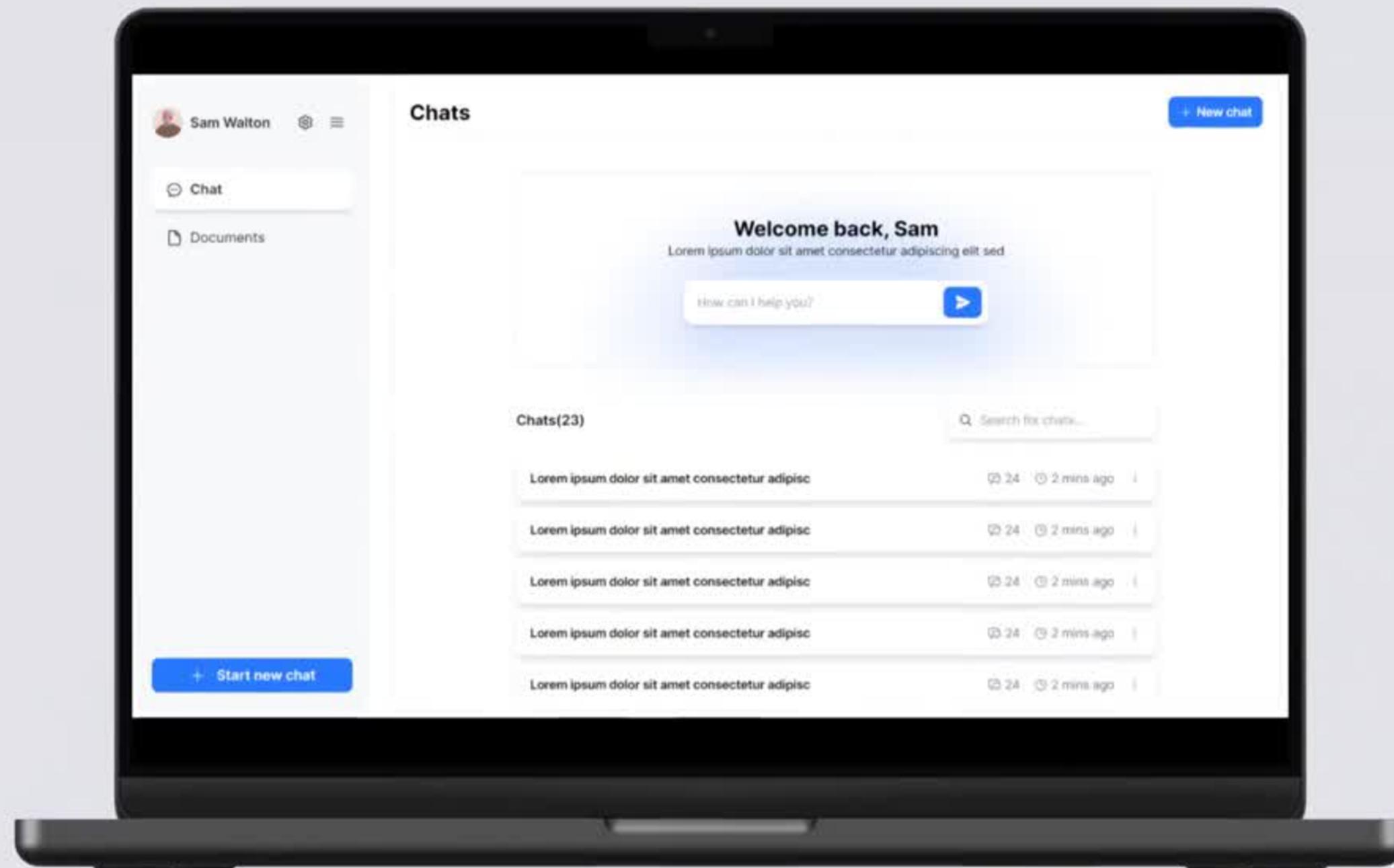


Modern, high-performance web framework for building APIs with Python, based on standard Python type hints.

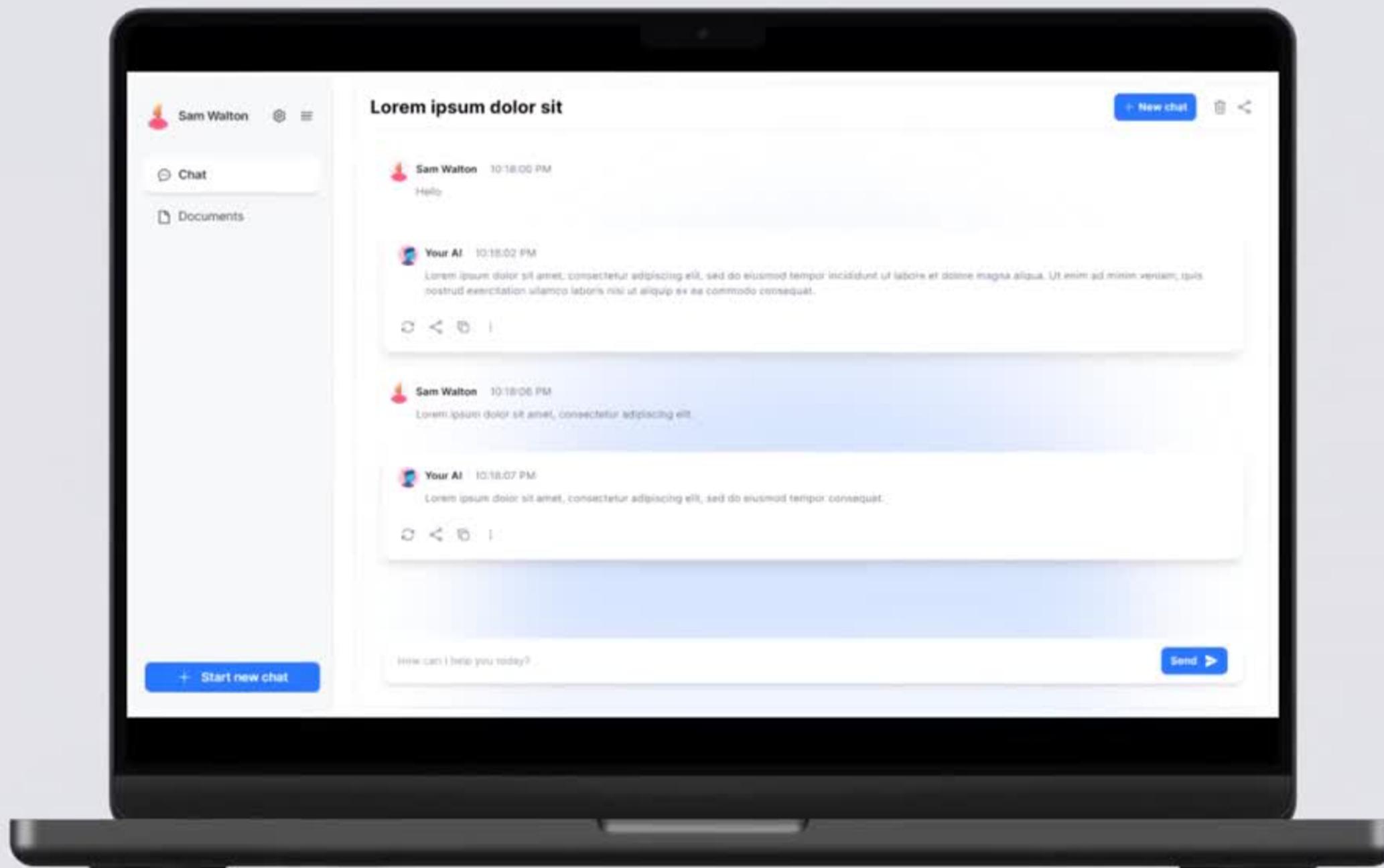


NoSQL database that stores data flexibly in JSON-like documents, ideal for modern, scalable applications.

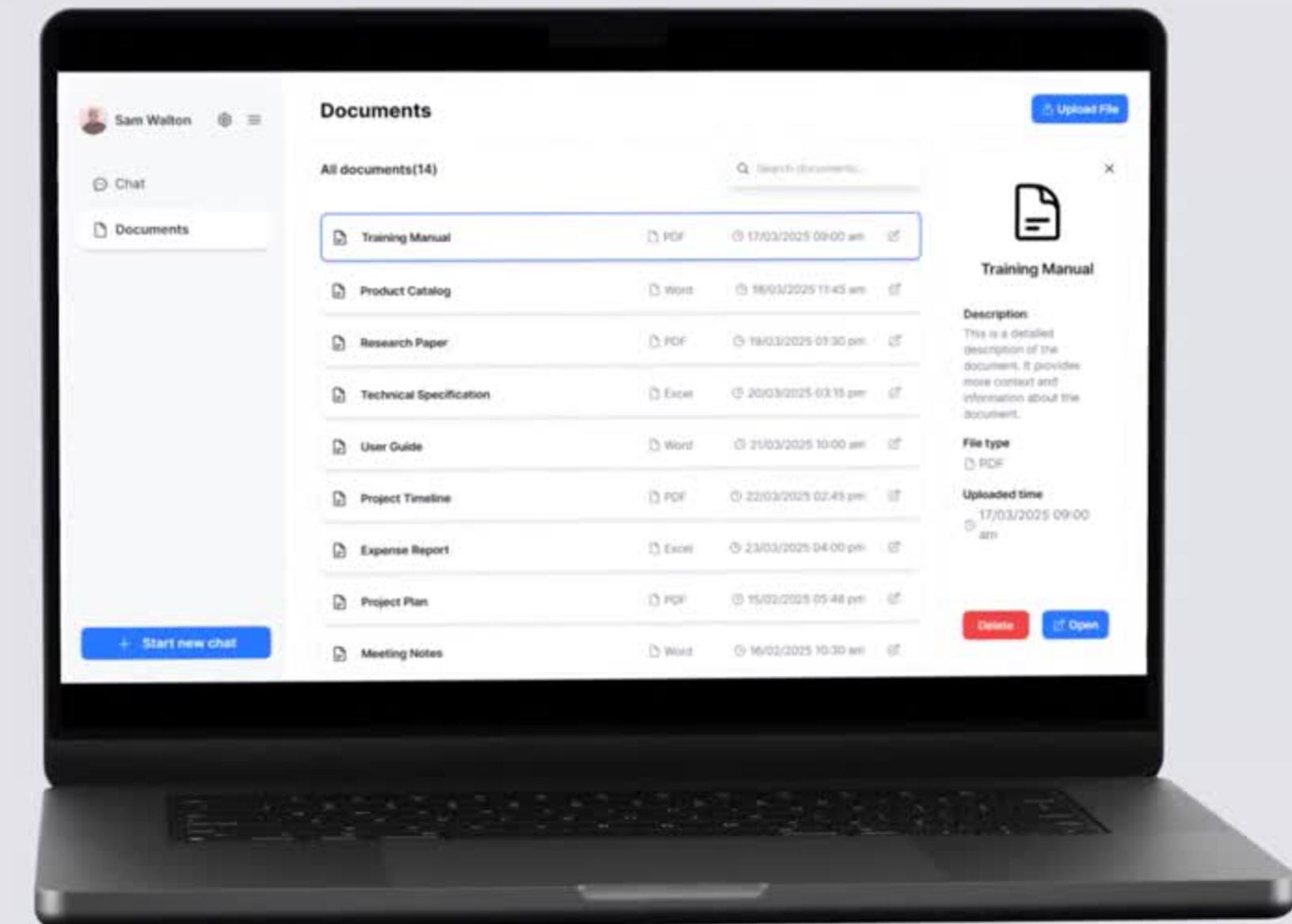
Frontend Design ↴



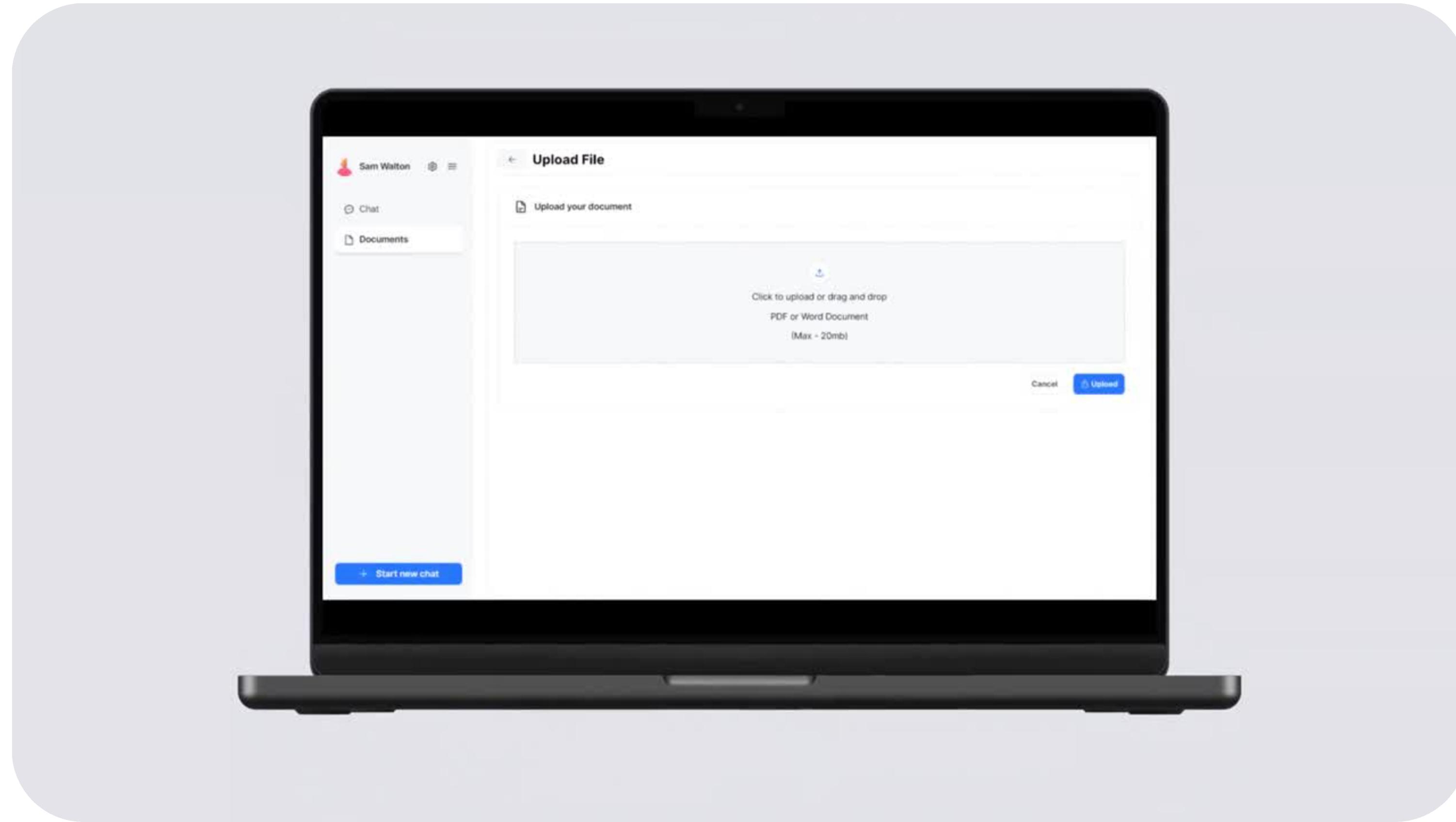
Frontend Design ↴



Frontend Design ↴



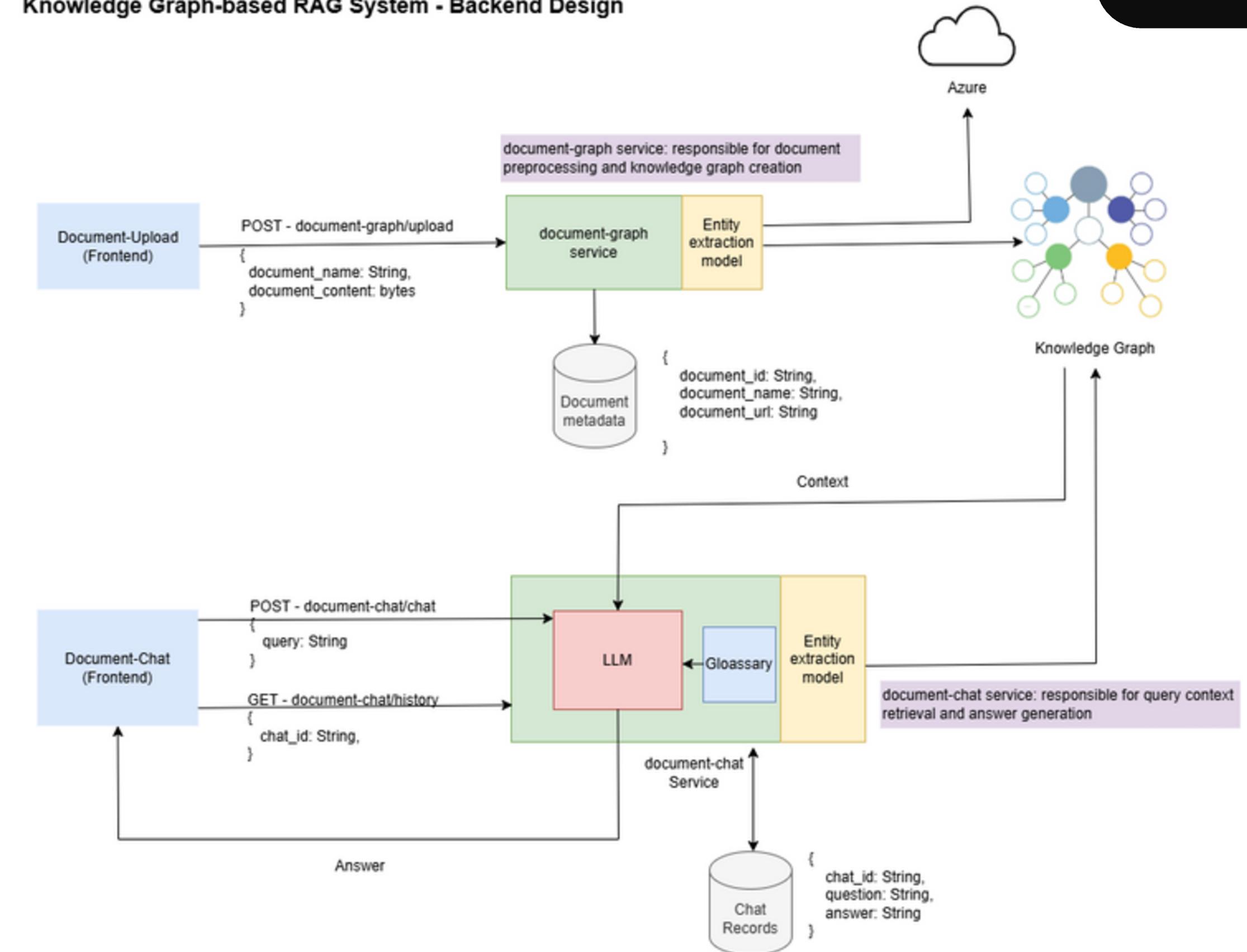
Frontend Design ↴



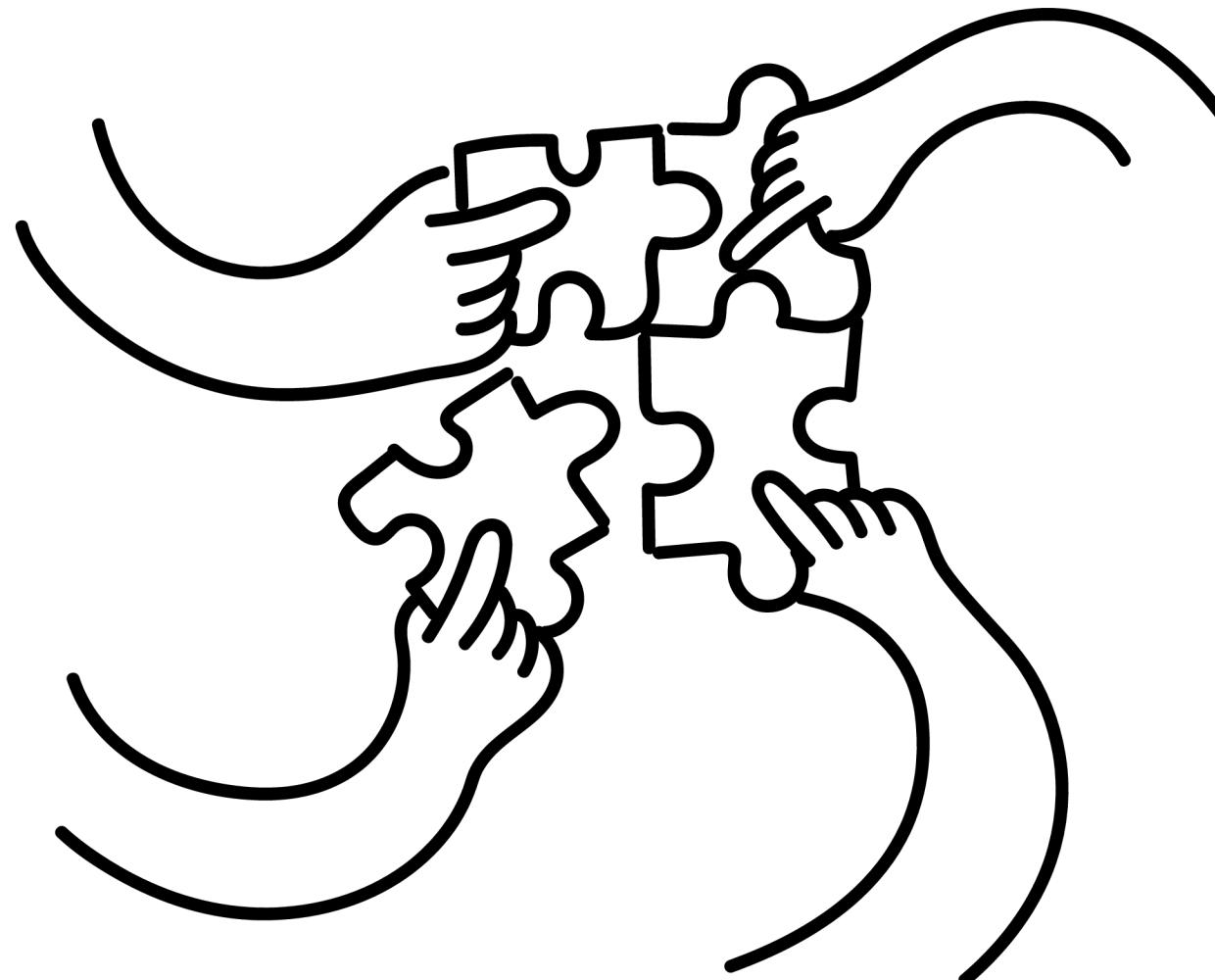
Backend Design ↴

Knowledge Graph-based RAG System - Backend Design

● Page 16



To-Do ↓



1. Configure Storage service
2. Complete Backend
3. API integration in Frontend
4. Implement chat history feature
5. Neo4j self-hosting
6. LLM response optimization
7. Knowledge Graph optimization
8. Review paper writing



Have Questions?

Thank
You!