Aayush Subramaniam - 50605671
Meghna Verma - 50604872

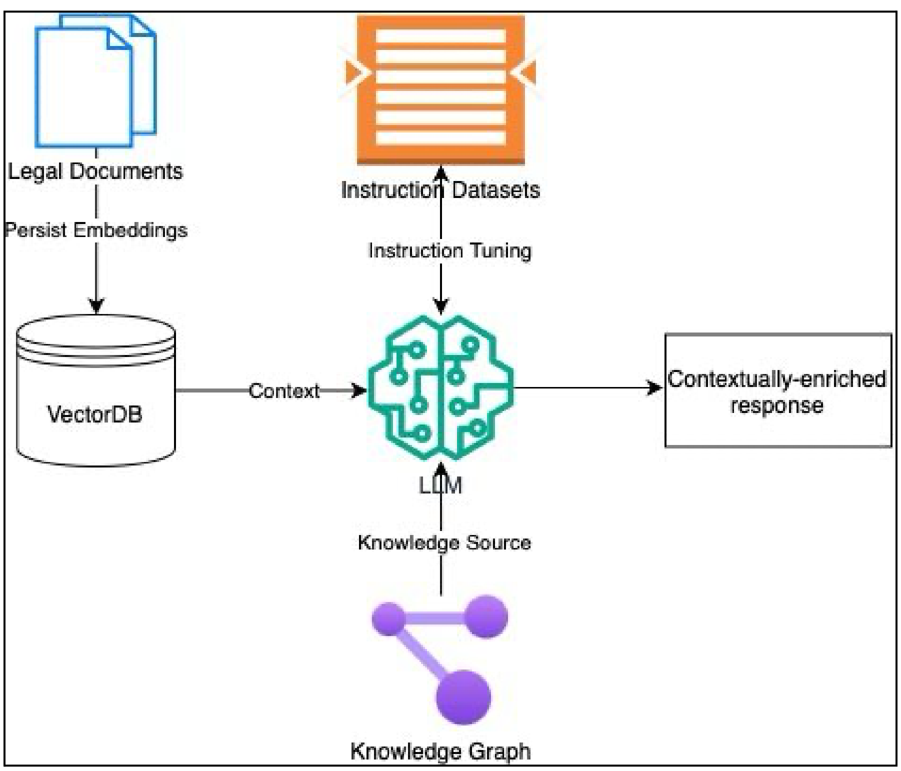
CSE 676-B Deep Learning
Final Project
Project Checkpoint

Contextually-Aware LLMs Using Knowledge Graphs

Objective :

We are planning to build a Question-Answering platform for the legislative domain by combining **Retrieval-Augmented Generation (RAG)** and **Knowledge Graphs** to provide contextually rich, accurate answers to legal questions. We aim to enable intelligent interaction with large collections of legal documents and provide detailed responses grounded in factual knowledge. Our main objective is to create a system that is lightweight and accessible. For this, we will leverage multiple **small language models** within the RAG pipeline, augmented by knowledge graphs, to achieve accuracy comparable to state-of-the-art large-scale LLMs. This will enable deployment on **edge devices and low-resource environments**, extending usability beyond centralized servers or cloud-based setups.

Architecture :



Aayush Subramaniam - 50605671
Meghna Verma - 50604872

Progress So Far:

1. **RAG pipeline implementation:**

a. We implemented a standard RAG pipeline using three small models relevant to QA tasks using the LangChain framework:
   i. google/flan-t5-base
   ii. declare-lab/flan-alpaca-base
   iii. allenai/unifiedqa-t5-base
b. We used **SentenceTransformers (all-MiniLM-L6-v2)** for embedding generation.
c. We used **LangChain's RecursiveCharacterTextSplitter** to chunk legal documents into manageable segments.
d. We used **FAISS** as the vector store for efficient document retrieval.
e. Created a Prompt with relevant contexts from the FAISS vector store.

2. Evaluation Metrics:

We evaluated responses generated by the baseline models via a basic RAG pipeline using the SQuAD evaluation metrics - Exact Match and F1 score, which allowed us to benchmark our system's response quality against standard QA datasets.

3. FAISS Vector DB Setup

a. We created a vector database of legal document chunks using a scalable embedding process and enabled persistent storage and reuse of the FAISS index for query answering.
b. In our system, we are :
   i. Reading all documents from a folder.
   ii. Splitting them into chunks.
   iii. Embedding them into vector form.
   iv. And finally, storing them in a FAISS database for fast similarity-based retrieval.

4. QA Baseline model :

a. First, we load the FAISS index.
b. We retrieve the top 4 relevant chunks using Maximal Marginal Relevance (MMR).
c. Then, format the context into a QA-friendly prompt.
d. Finally, we use a small text2text model (future steps: Using a legal-aware model) to generate a concise, context-grounded answer.
e. This will work well for our use case as legal documents are often long and complex, and chunking + retrieval helps isolate the relevant portions, and the prompting structure keeps the output factual, concise, and user-friendly.

Aayush Subramaniam - 50605671
Meghna Verma - 50604872

Further Implementation Plan :

1. **Use Case implementation : Legislative Domain.**
    a. Our final QA system will be specifically designed for legal professionals, researchers, and law students and will be tailored for law-related questions.
    b. Legal documents like case laws, regulations, and statutes will be ingested, chunked, embedded, and indexed so that when the user inputs legal queries, the system will respond with contextually enriched answers.

2. **Knowledge Graph integration :**
We will integrate a Custom-built graph Knowledge-Graph using Neo4j for specific laws, sections, and precedents to:

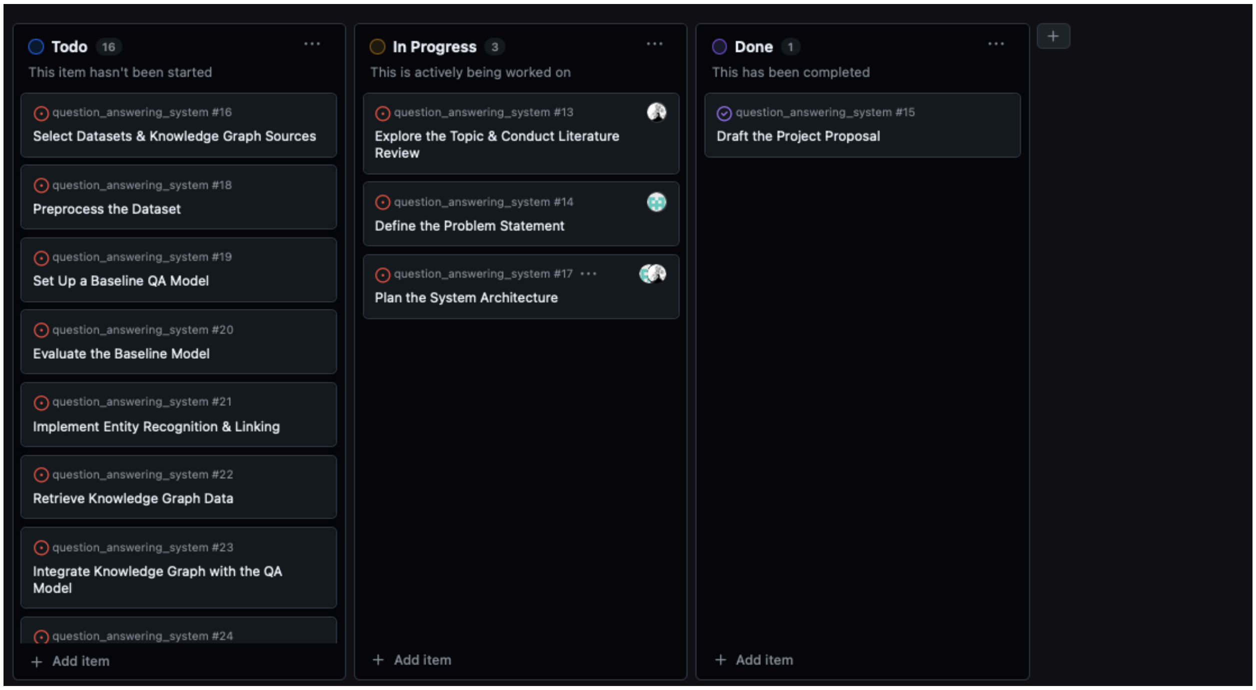
    i. Link legal entities and cases.
    ii. Provide structured context to augment generative answers.
    iii. Improve fact-grounding and traceability in legal answers.

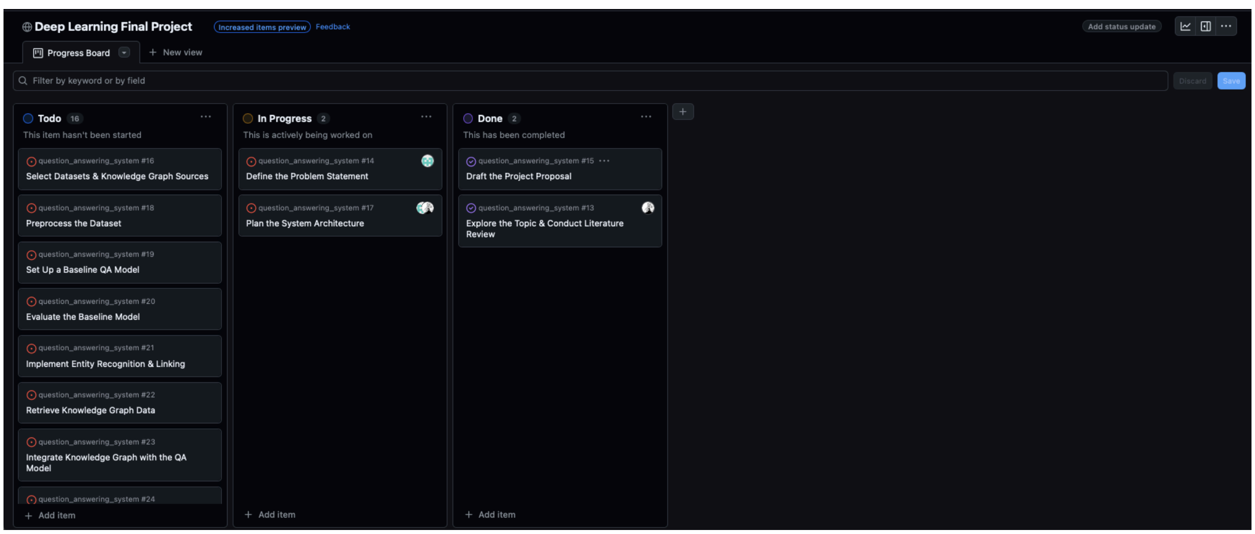
3. **Final Architecture will include :**
    a. **Document Ingestion & Chunking**: Using LangChain's loaders and splitters We will load raw legal texts from different sources and use character-based chunking to split long texts.
    b. **Embedding & Retrieval**: SentenceTransformer + FAISS We will use SentenceTransformer to convert chunks to dense vectors and retrieve similar chunks according to the user's legal question by storing all document chunks and allowing efficient semantic retrieval with FAISS.
    c. **Answer Generation**:
        i. We will use Small transformer-based LLMs (listed above) to generate a coherent answer from the retrieved contexts and Knowledge Graph.
        ii. After this, we will instruct tune our models with legal information to give more accurate results.
    d. **Knowledge Graph Augmentation**:
        i. We will identify Legal entities and the relationships between those entities via NER + linking.
        ii. During answer generation, contextual facts and additional knowledge will be retrieved from the KG to improve answer precision and reduce hallucinations.

4. **Evaluation Metrics :**
    a. We will include custom legal domain RAG benchmarks to examine retriever performance and downstream question-answering performance.
    b. And will work on hallucination reduction with KG support.
    c. LegalBench - https://hazyresearch.stanford.edu/legalbench/getting-started/

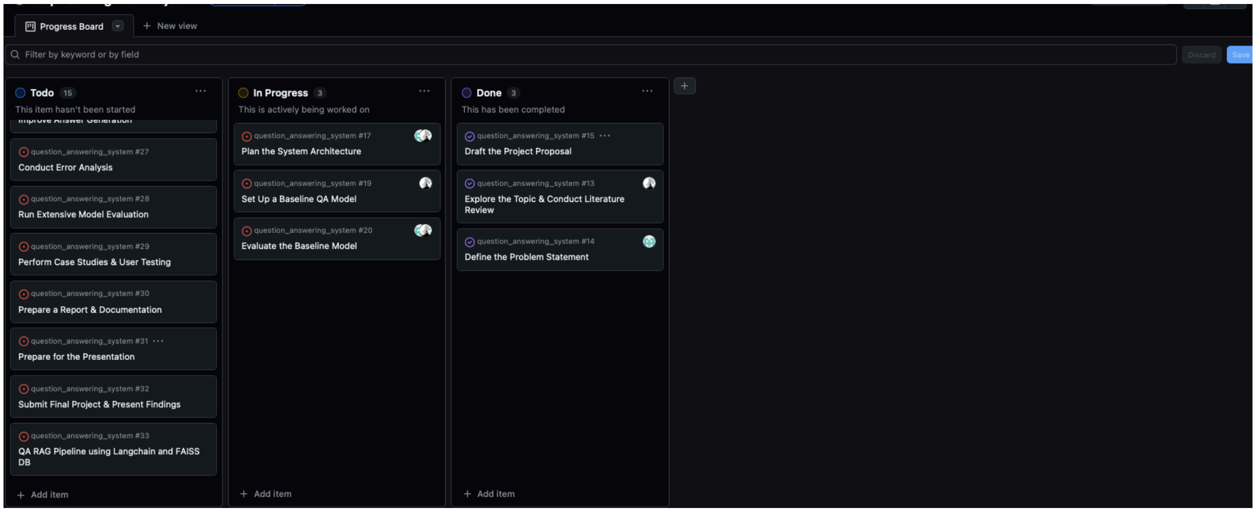
Aayush Subramaniam - 50605671
Meghna Verma - 50604872

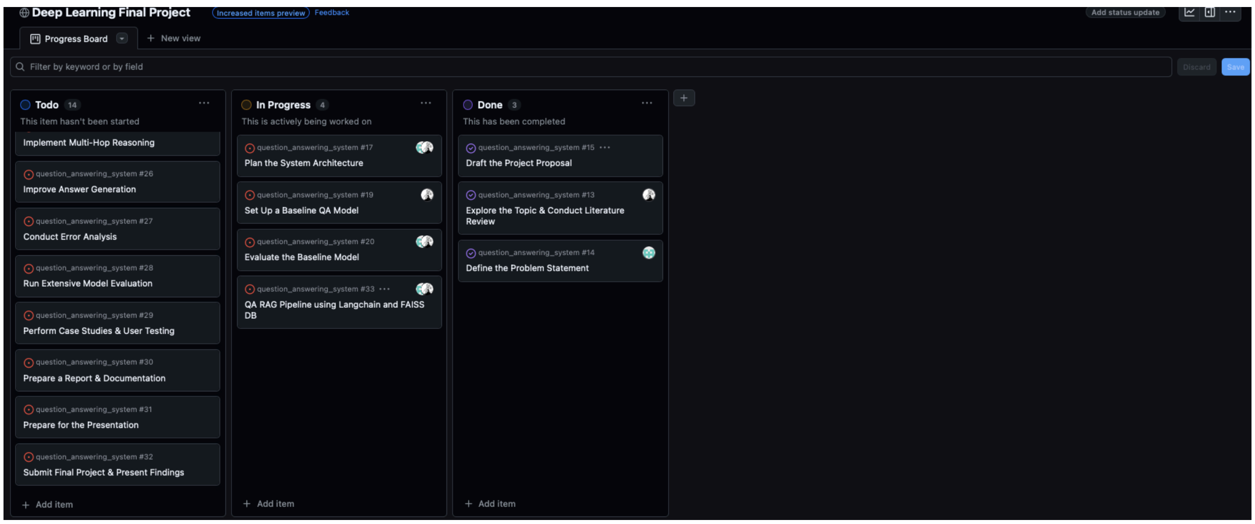
Project Board Screenshots

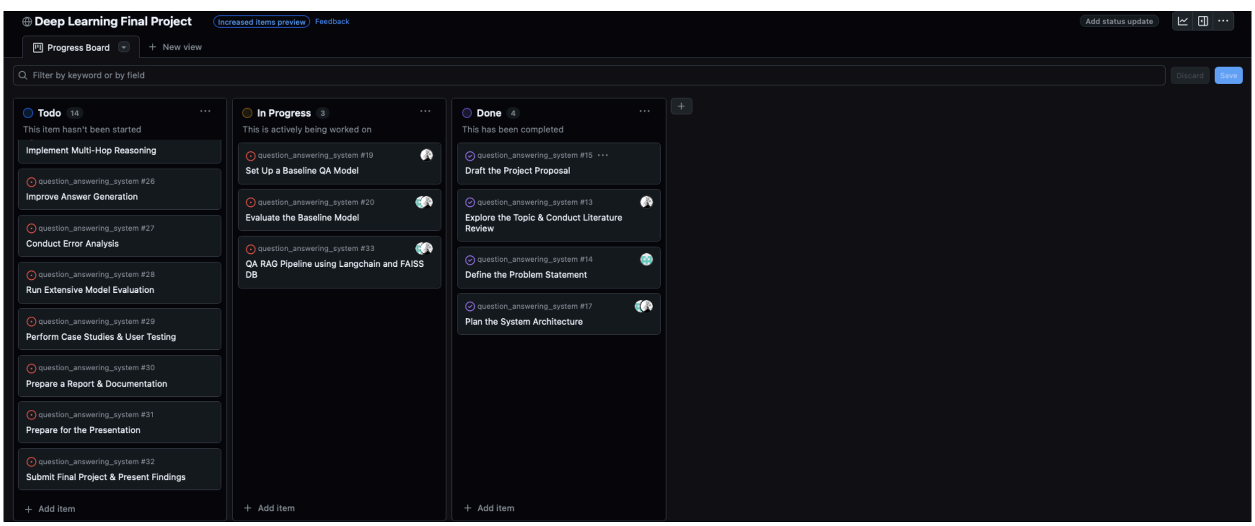




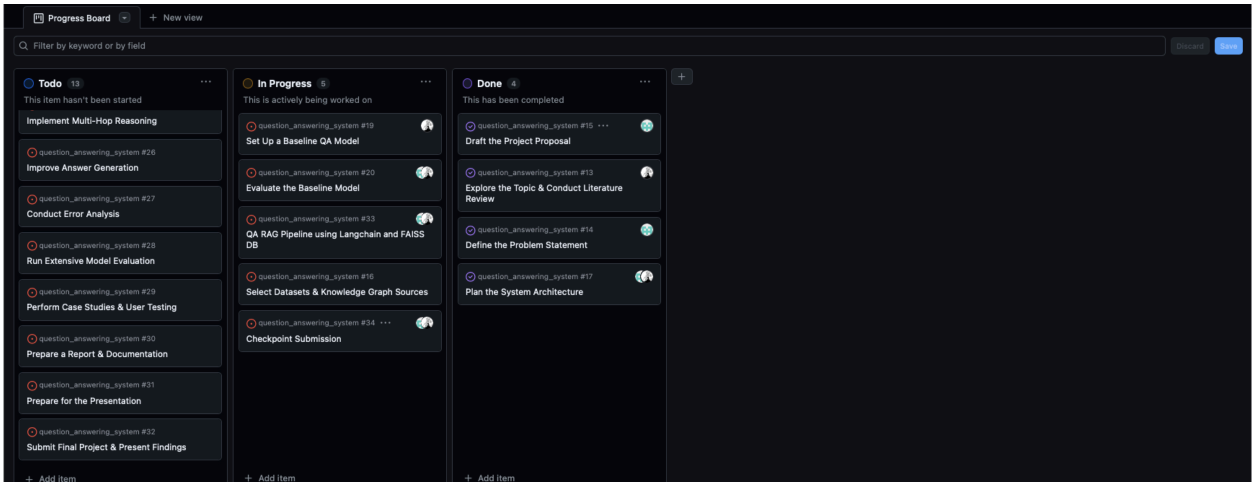
Aayush Subramaniam - 50605671
Meghna Verma - 50604872

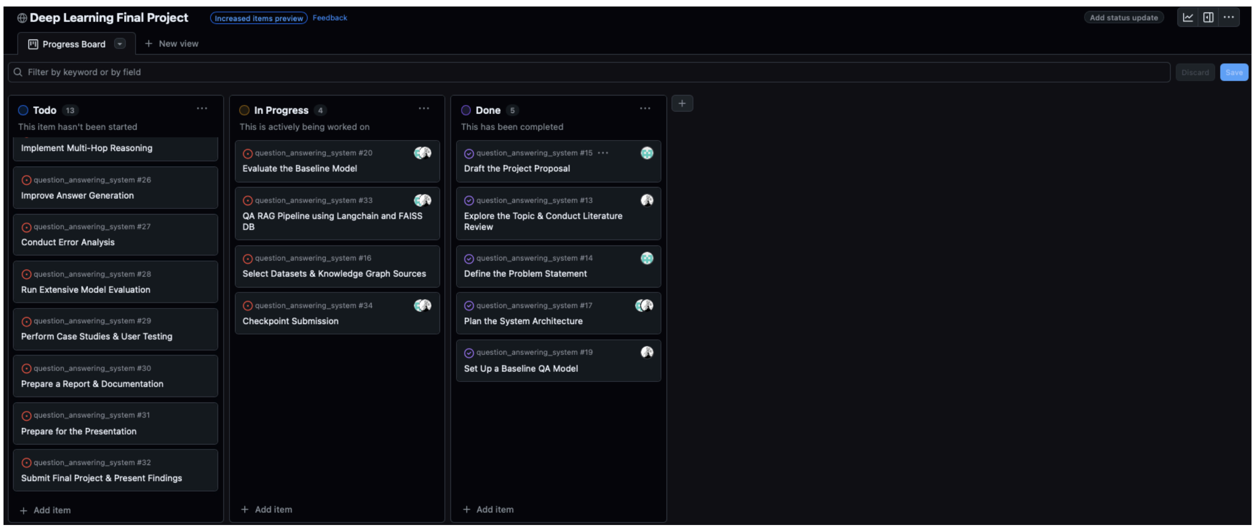


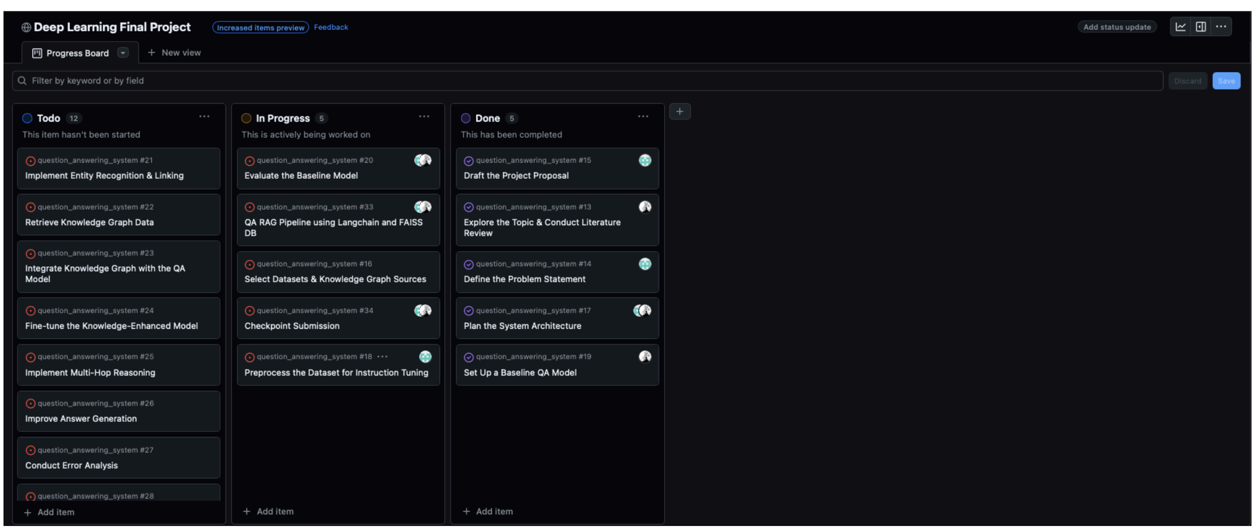




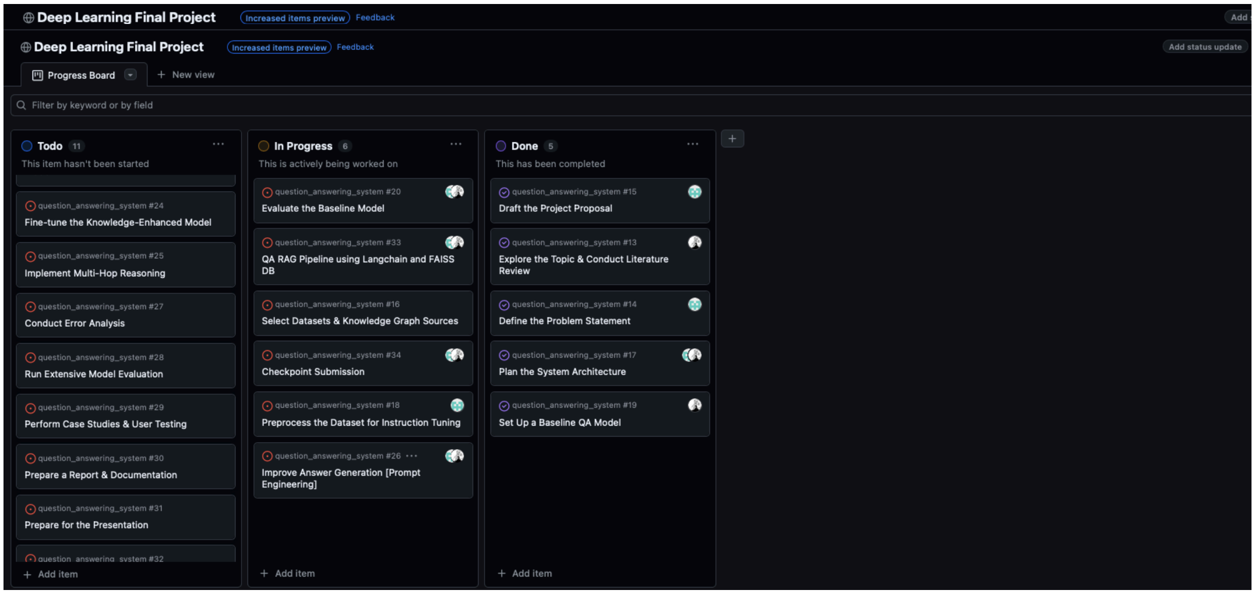
Aayush Subramaniam - 50605671
Meghna Verma - 50604872

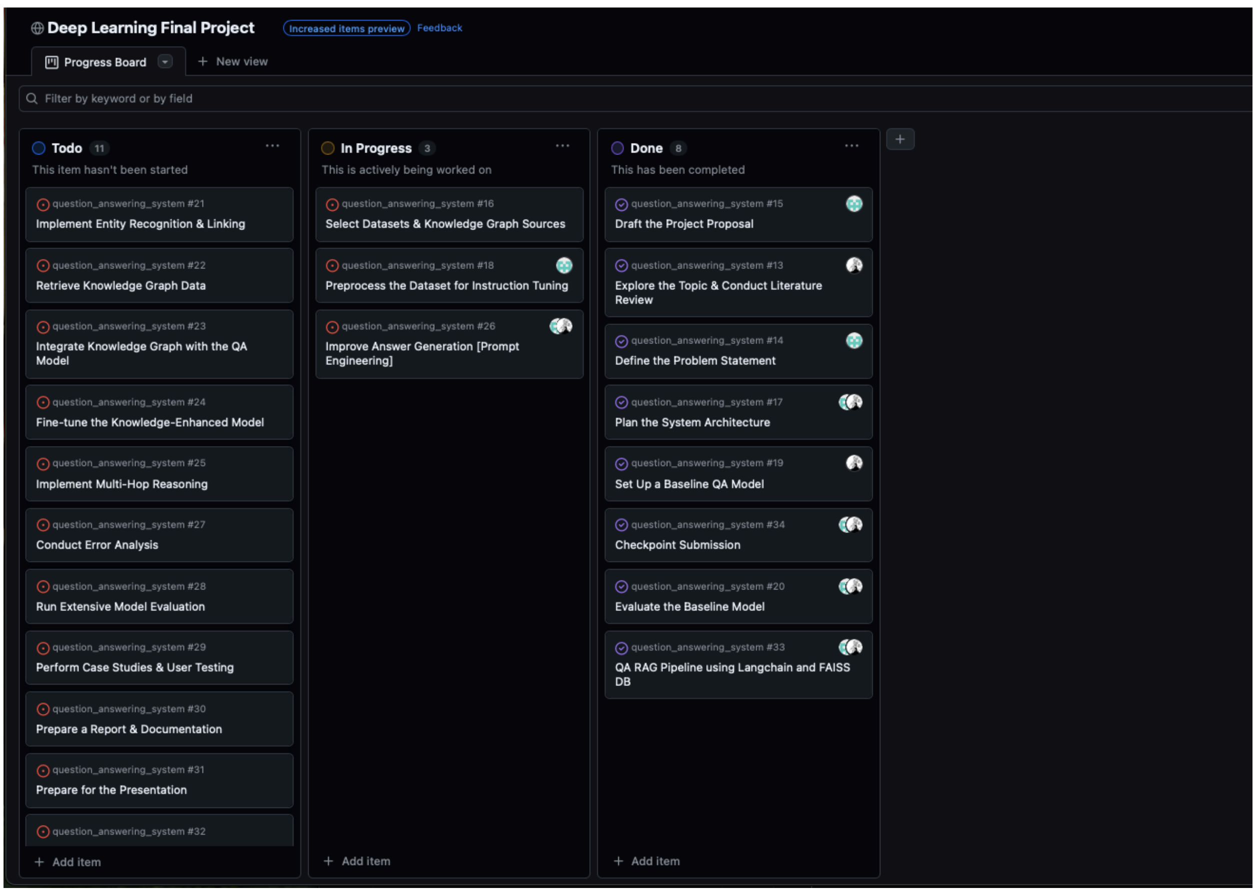






Aayush Subramaniam - 50605671
Meghna Verma - 50604872





Project Board Link - https://github.com/users/Meghna0327/projects/1