This is a comprehensive assignment with multiple components. Below is a step-by-step approach to help you complete each task:

Step 1: Web Crawling

Set Up the Web Crawler:

Use Python libraries like requests and BeautifulSoup for web scraping.

Consider using Scrapy for more advanced crawling requirements.

Scrape Data up to 5 Levels Deep:

Implement a recursive function to follow links up to 5 levels.

Ensure you handle cases where links might lead to non-HTML content or dead links.

Step 2: Data Chunking and Vector Database Creation

Data Chunking Based on Semantic Similarity:

Use libraries like spaCy or nltk for text preprocessing.

Use models like BERT for semantic similarity to chunk data.

Convert Data to Embedding Vectors:

Use sentence-transformers to convert chunks to embeddings.

Create a Vector Database using MILVUS:

Install and set up MILVUS.

Store embedding vectors with metadata.

Step 3: Retrieval and Re-ranking

Query Expansion and Hybrid Retrieval:

Use techniques like synonym expansion or context-aware query expansion.

Combine BM25 (from elastic search) with BERT-based retrieval.

Re-rank Results:

Use a re-ranking model to sort retrieved data based on relevance.

Step 4: Question Answering

Integrate an LLM:

Use a pre-trained model from Huggingface or OpenAI.

Generate Answers:

Pass re-ranked data to the LLM for answering the user query.

Step 5: User Interface (Optional)

Build a UI with Streamlit:

Create a simple interface for users to input queries and display results.