Handling Large and Complex Tables:
Suppose a user inputs the research topic: "Handling Large and Complex Tables in NLP."

The application must:

1. **Search Agent**: Fetch related papers on handling large and complex tables from platforms such as Arxiv, focusing on papers published in the last five years (e.g., 2019-2024). Store these papers in a time-series database.
2. **Database Agent**: Store the collected papers and allow querying of papers based on specific timeframes, such as retrieving all papers related to handling large tables from 2019 onwards.
3. **Frontend Application**: Display all the fetched papers in a user-friendly interface using Streamlit, allowing users to browse the papers easily.
4. **Q&A Agent**: Provide a question-answering feature where users can interact with the research papers. Example questions include:
   - "Summarize the advancements in handling large and complex tables over the last five years."
   - "What techniques are proposed in Paper X for efficient processing of large tables?"
   - "How does Paper Y deal with missing data in large tables?"
   - "Which paper presents the most effective approach for managing large and complex tables, and why?"
   - (Reference: Paper H - Section 5.1 for detailed methodology).
5. **Future Works Agent**: Suggest potential improvements or research opportunities based on the reviewed papers. Generate a review paper summarizing the state-of-the-art techniques and propose new directions for future work, such as improving data imputation methods, enhancing table structure understanding, and reducing computational costs for processing large tables.

Chat Interaction Example:
User: "Show me all the papers related to handling large and complex tables in the last 5 years."
Bot: "Here are the papers on handling large and complex tables from 2019 to 2024. [List of papers displayed]"
User: "Summarize the advancements made in these papers."
Bot: "The advancements in handling large and complex tables over the past five years include improvements in table parsing techniques, better handling of missing data, and efficient use of transformer-based models for table understanding. Notable contributions include Paper A, which presents a novel table parsing method, and Paper B, which proposes an effective data imputation strategy. Reference: Paper A - Section 3.2 | Paper B - Section 4.1"
User: "What techniques are proposed in Paper B for efficient processing of large tables?"
Bot: "Paper B proposes a hierarchical parsing approach combined with a data reduction mechanism to efficiently process large tables, significantly reducing computational overhead while maintaining data integrity. Reference: Paper B - Section 4.1"

User: "What future research can be done based on these papers?"
Bot: "Based on the reviewed papers, future research could focus on developing more effective data imputation methods for missing table values, improving the scalability of table understanding models, and integrating multimodal features to enhance table interpretation. Reference: Paper A - Section 5 | Paper C - Section 6.2"
User: "Which paper presents the most effective approach for managing large and complex tables?"
Bot: "According to the papers reviewed, Paper D presents the most effective approach for managing large and complex tables. This approach leverages a combination of neural attention mechanisms and adaptive data sampling to optimize table handling, leading to significant improvements in efficiency and accuracy. Reference: Paper D - Section 4.4 | Paper E - Section 3.7"