Webpage Question Answering

- Here the major steps I have done is the conversion of the URL to text, then it is given to OpenAI's language models for processing text and performing question answering.

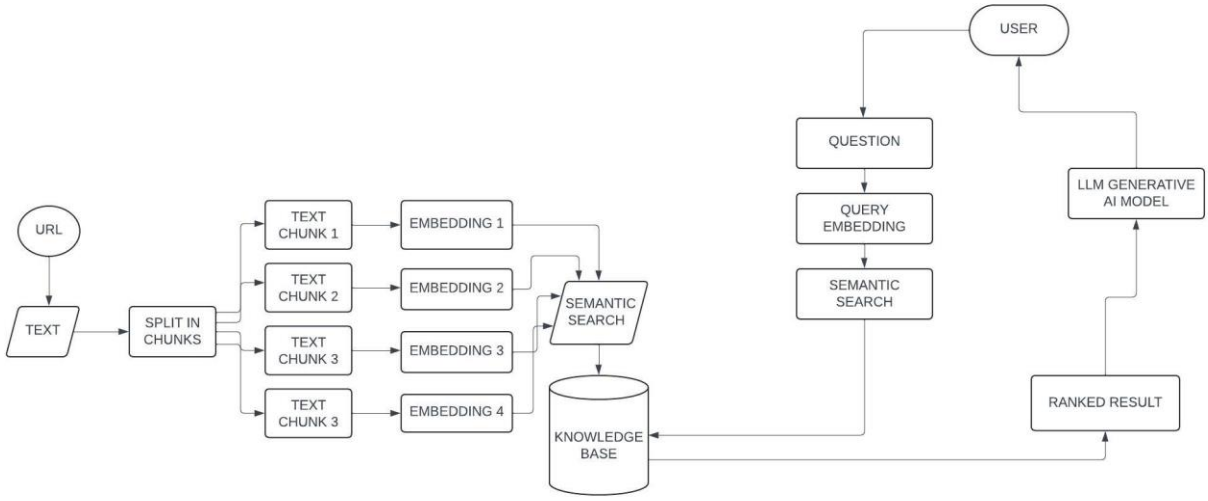

FIG : BLOCK DIAGRAM OF IMPLEMENTATION

- **STEP 1 : URL TO TEXT CONVERSION:**
  - Here the '**BeautifulSoup'is the package that is used for the** parsing HTML documents.
  - This finds all **<p>** (paragraph) tags in the HTML document. This returns a list of all paragraphs.
  - Extracts the text from each paragraph tag and joins them into a single string with spaces between. This is used to consolidate the textual content into a manageable form.
- **STEP 2: TEXT BASED QUERY:**
  - This is the method of for extracting text from a PDF, splitting it into manageable pieces, and then querying these pieces using a Language Model to answer a specific question.

  - The text is split using **CharacterTextSplitter**, which divides the long **raw_text** into smaller chunks or segments.
  - Converts text chunks into vector embeddings using an embedding model from OpenAI. These embeddings represent text in a high-dimensional space where similar texts are closer together, enabling semantic search and comparison.
  - Uses the FAISS library to create an efficient search structure for these embeddings. FAISS is optimized for fast similarity search and clustering of large datasets.
  - Searches the indexed embeddings for chunks most relevant to the given query. This step locates the parts of the PDF content that are most likely to contain information pertinent to the query.
- **NOTE:** API key for OPEN AI integration is done from reference : https://medium.com/towards-artificial-intelligence/getting-started-with-openai-the-lingua-franca-of-ai-13864b3c886a