Course Conclusion

Congratulations! You've successfully completed the **Generative AI Applications with RAG and LangChain** course, where you've applied your knowledge to real-world scenarios, refining your skills in using document loaders, text-splitting strategies, vector databases, and retrievers within LangChain. You've built a QA bot, integrated a simple Gradio interface, and explored the nuances of retrieval-augmented generation (RAG).

At this point, you know that:

- LangChain uses document loaders, which are connectors that gather data and convert it into a compatible format.
- TextLoader class is used to load plain text files.
- PyPDFLoader class or PyMuPDF Loader is used for PDF files.
- UnstructuredMarkdownLoader is used for Markdown files.
- Use the JSONLoader class to load JSON files.
- CSV Loader is used for CSV files.
- Beautiful Soup or WebBaseLoader is used to load and parse an online webpage.
- WebBaseLoader is used to load multiple websites.
- UnstructuredFileLoader is used for unknown or varied file formats.
- LangChain uses text splitters to split a long document into smaller chunks.
- Text splitters operate along two axes: The method used to break the text and how the chunk is measured.
- Key parameters of a text splitter: Separator, chunk size, chunk overlap, and length function.
- Commonly used splitters: Split by Character, Recursively Split by Character, Split Code, and Markdown Header Text Splitter.
- Embeddings from data sources can be stored using a vector store.
- A vector database retrieves information based on queries using similarity search.
- Chroma DB is a vector store supported by LangChain that saves embeddings along with metadata.

- To construct the Chroma DB vector database, import the Chroma class from LangChain vector stores and call the chunks and embedding model.
- A similarity search process starts with a query, which the embedding model converts into a numerical vector format.
- The vector database compares the query vector to all the vectors in its storage to find the ones most similar to the query.
- A LangChain retriever is an interface that returns documents based on an unstructured query.
- Vector store-based retriever retrieves documents from a vector database using similarity search or MMR.
- Similarity search is when the retriever accepts a query and retrieves the most similar data.
- MMR is a technique used to balance the relevance and diversity of retrieved results.
- Gradio is an open-source Python library for creating customizable web-based user interfaces.
- To set it up:
    - Write the Python code
    - Create the Gradio interface
    - Launch the Gradio server using the launch method
    - Access the web interface through a local or public URL provided by Gradio
- To code a simple text input and output with the Gradio interface, use the gr.Interface function.
- To upload or drop files with Gradio, use gr.File command.
- The Multi-Query Retriever uses an LLM to create different versions of the query to generate richer documents.
- The Self-Query Retriever converts the query into a string and a metadata filter.
- The Parent Document Retriever has a parent splitter to split text into large chunks and a child splitter for small chunks.