Team: Code Crafters

Problem Statement:

Running GenAI on Intel AI Laptops and Simple LLM Inference on CPU and fine-tuning of LLM Models using Intel Open VINO.

Unique Idea Brief (Solution):

- The ideology is to create a Gen AI application that can summarize text provided by the user in the text or Pdf form.
- Based on the underlying text, summarized content can be provided in short.
- User queries can be generatively answered by analysing and processing the text provided by the user.

Features Offered:

1) Summarization of input text.
2) Answering to questions asked by the user.
3) Downloading the Summary.

Process flow.

- **User Interaction:**
  - **Start**: User accesses the web application.
  - **File Upload**: User uploads a PDF or text file via the Streamlit interface.

- **Text Extraction and Preprocessing**:
  - **Extract Text**: If a PDF/text is uploaded, text is extracted using PyPDF2.
  - **Preprocess Text:** The extracted text is cleaned and tokenized.

- **Embedding Creation:**
  - **Generate Embeddings**: The pre-processed text is passed through the BAAI/bge-large-en-v1.5 model to generate embeddings**.**

OpenVINO

- **Store Embeddings**: Embeddings are stored in a FAISS index for efficient retrieval.

- **Query Processing and Reranking:**
  - **Convert Query to Embeddings**: The user's query is converted into embeddings using the BAAI/bge-large-en-v1.5 model.
  - **Retrieve Similar Embeddings**: The FAISS index is queried to retrieve embeddings similar to the query embeddings.
  - **Rerank Results**: The initial search results are reranked using the BAAI/bge-reranker-base model to prioritize the most relevant results.

- **Summarization**
  - Generate Summary: The relevant text sections are summarized using the facebook/bart-large-cnn model.

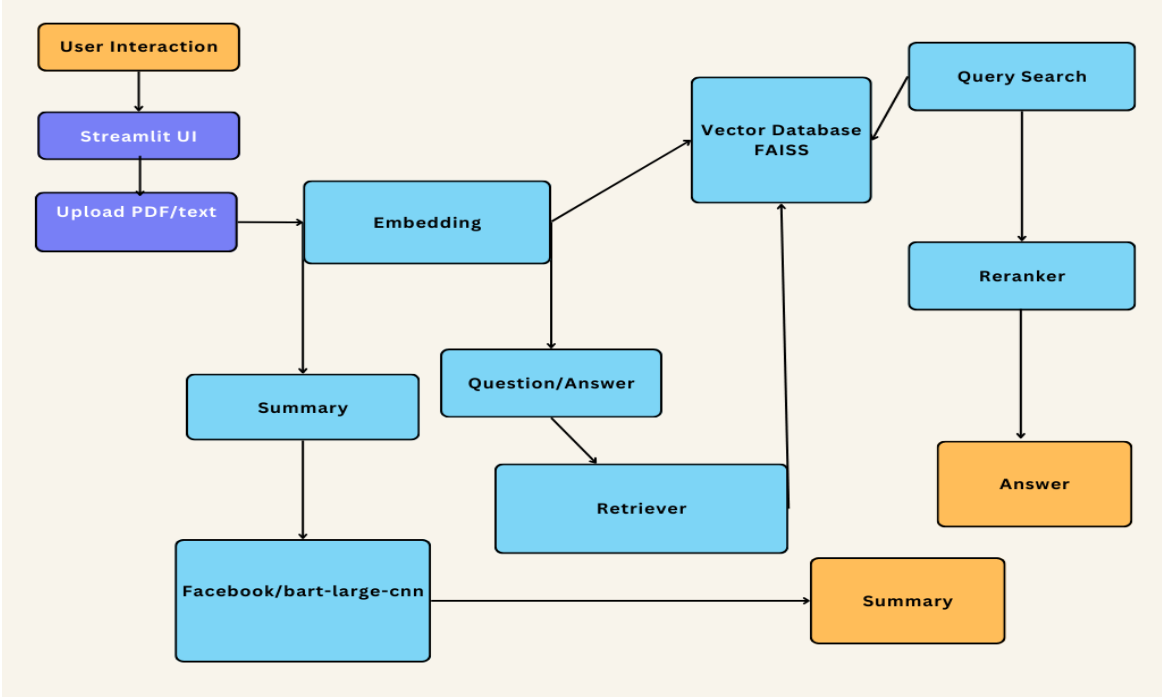
OpenVINO

- **Fine Tuning, LLM Inference and Optimization with OpenVINO™:**
  - Models are optimized with Intel OpenVINO™ to enhance performance on Intel CPUs**.**
  - Fine tuned models are used to perform inference tasks, generating embeddings, reranking results and summarizing text.

- **Output:**
  - Display Summary: The summarized text and answers are displayed to the user through the Streamlit interface.

OpenVINO™

Architecture Diagram:



Streamlit

Technologies used:

- **Python:** Overall application development and backend logic.
  i)Streamlit: Crafting an intuitive User Interface.
  ii) PyPDF2: Text extraction from user input document.
- **LLM models:**
  i) facebook/bart-large-cnn: For Summarizing the input text document.
  ii) **BAAI/bge-large-en-v1.5:** Transforms input text into embeddings.
  iii) BAAI/bge-reranker-base: Re-ranks answers for optimized user queries.
- **Vector Database:** FAISS, for storing extracted embeddings and relevant data retrieval.

OpenVINO

Team members and contribution:

1) **Ajinkya Nanivadekar:**
   - **FAISS implementation: Handled the implementation of preprocessing the input text, reliable extraction of data and integration with the vector database.**
   - **Ensured model integration and their functioning used for summarization and embedding creation.**

2) **Swaraj Kadu:**
   - **Integrated Open Vino with the project and applied necessary optimizations to enhance inference speed on intel CPU.**
   - **Developed the Streamlit intuitive user interface.**
   - **Improved response times by trying out other models to fit in with OpenVino**

Conclusion:

Thus, we have successfully learnt the application of GenAI using Intel Open-Vino tool by creation of a Text Summarizer that can summarize and answer user queries.