Setting up the environment:

1. First create a virtual environment with python version >= 3.9.
2. Now pip install the following packages: sentence-transformers, langchain, langchain_community, pymilvus.
3. Activate the environment.

Start the milvus container:

1. Download the official docker-compose.yml file from milvus documentation.
2. To change port no. and all use: nano docker-compose.yml
3. Create and start the container using docker compose -d.
4. This will create three containers etcd, milvus and minio.

Creating the script.

1. Import all the packages
2. Now connect to the milvus client using localhost and port number
3. Now call the sentence transformer model and set the device to the gpu id available.
4. Now create a milvus collection i.e. a database and set the column names(fields) details such as name, data type, max length, etc. Also set the index type to IVF_FLAT and metric (clustering) type to L2.
5. Also set the dimension of the collection to be that of the sentence transformer model.
6. Now call the recursive text better function to split the text With a Chance size of 512 and chunk overlap of 100.
7. Now create an empty list ids_array to store the id's of every batch and also create three Global variables count, refer and length
8. Now create a for loop and select the judgement DB database and also set the chunk size to the chunk size variable now create a reference variable where the date of judgement and the case title for each row for the entire batch.
9. Also create a temporary file temp.csv where the text for the judgement is stored row by row and also use the CSV loader to select the text from the temp.csv file.
10. Split documents into smaller chunks using the text splitter.
11. Iterate over the text chunks to create unique chunk IDs based on the document row number. Ensure that chunks from the same document are given sequential IDs.
12. Assign the generated chunk IDs to each text chunk and attach reference metadata to each chunk and prepare them for embedding.
13. Generate unique IDs for each chunk and update the reference length.
14. Prepare the text chunks for embedding in batches.

15. Use the SentenceTransformer model to generate embeddings for each batch. Utilize GPU for faster processing.
16. Create a list of dictionaries containing chunk IDs, embeddings, document text, and metadata.
17. Insert the data into Milvus in batches, flushing the collection after each batch.