**SSEGPT framework**

**Data Collection and Preprocessing:**

* Gather domain-specific data including PPT and textbook, homework of EM624, captions of course videos, student emails
* Preprocess the data to remove noise
* Convert text, tables, and images into a text format for model input
* Use spacy tokenizer for tokenization (or Roberta tokenizer)
* Use sentence-transformers/all-roberta-large-v1 for embedding
* Insert embeddings to Pinecone vector index

**LLM selection:**

* Select one model as the main LLM for SSEGPT

**Large Language Model (LLM):**

* Implement the selected LLM as the core component of the SSE tutor.
* Pinecone index will retrieve sentences based on the similarity (cosine metric) and that data will be used by the LLM to generate responses to user queries.
* When a user inputs a question or prompt, pass it to the language model.
* Process the model's response to ensure coherence, clarity, and relevance to the query.

**Fine tuning:**

* Fine-tune the model on domain-specific data to enhance its knowledge and relevance to the subject.

Backend API

* Create the API in the backend using Flask
* It will accept the user query and pass it to the model function
* The model function will use the model to generate a response and return it
* The API then returns the response to the Frontend UI which will then be displayed

**User Interface (UI):**

* Design the UI to input questions or prompts and display model-generated responses.
* Develop a user-friendly interface for students to interact with the SSE tutor.

**Feedback and Correction:**

* Implement a mechanism to provide feedback and corrections to the model's responses.
* Allow users to rate the quality of the answers and suggest corrections if needed.

Workflow/Pipeline:

1. The user interacts with the SSE tutor through the UI, inputting questions or prompts related to the subject.
2. The input is passed to the large language model (LLM), which generates a response based on its training and fine-tuning on domain-specific data.
3. The generated response is processed to ensure it is coherent, accurate, and relevant. This step may involve filtering out inappropriate or irrelevant content.
4. The processed response is displayed to the user through the UI.
5. The user has the option to provide feedback and corrections to the response. This feedback is used to improve the model over time.
6. The process repeats as the user continues to interact with the SSE tutor, allowing for iterative learning and improvement of the system's responses.