Course 02830 scope:
The course 02830 is a project-based course in digital media engineering. The key requirement is that the project must involve **technical design and prototyping**, leveraging the students´ engineering skills while also expanding their technical competencies.
Requirements:

- A prototype
- Iterative testing
- Project planning
- A report (graded by course officials)

Project idea:
We aim to create an AI agent that combines the functionality of a daily journal and a writing-idea generator. It should suggest creative writing prompts or themes based on journal history.

Project plan:
Overview:
MVP -> data layer -> memory -> token optimization -> RAG -> UX -> report writing

Week	Aim	Tools	Details
1-2	Minimum Viable Product (MVP)	FastAPI, Pydantic, OpenAI API	Build a simple web app where a user submits a journal entry, and it returns a theme summary/creative writing suggestions using LLM. 1. Create a FastAPI for the journal 2. Define a JournalEntry using Pydantic 3. When a user submits an entry, call OpenAI's chat/completions endpoint to summarize it. 4. Return summary to the user.(JSON with summary, suggestion)
3-4	Data layer	Suggestions from supervisor, preferably free trial/student services	Move data (journal entries) from local to cloud storage. 1. Set up cloud DB

			2. Create a workflow that saves new journal entries to storage
5-6	Embeddings and basic memory	OpenAI embeddings	Give the app a simple memory system so it can recall past journal entries and suggest ideas based on them. 1. User writes entry -> convert entry to embedding 2. Save entry + embedding (date, text, embedding) 3. Group similar entries 4. When sending to LLM for summary/writing suggestions, send instead today's entry + top 2-3 similar past entries + main themes
7-8	Token optimization		Goal: Optimize input size so that long entries are saved efficiently, and input to openAI is efficient (not waste tokens on irrelevant history) as LLMs have a limit on how much text we can send. 1. Decide which entries to send to the AI vs. which are stored for retrieval only, and summarize longer entries to reduce token use. Or save only today + last few entries. The rest are embeddings only(?) 2. Use Pydantic to structure input and output
9-10	Improve basic memory (RAG		

	memory?)		1. User writes entry -> automatically retrieve relevant past entries using embeddings (top-N matches)
11	UX Design		Spend some time making the front-end/the prototype a bit nicer.
12 - 13	Report writing		We expect to write the report during the other weeks as well. However, we also expect the project plan not to hold up perfectly; thus, we plan for a buffer period as well.