AI-Powered Supplementary Learning Platform for University Courses

Problem Statement

University courses often rely on fragmented learning resources—slides, PDFs, lab codes, and external references—making it difficult for students to search, revise, and generate structured learning materials efficiently.
Your task is to design and build an AI-powered supplementary learning platform that enhances an existing course by organizing content, enabling intelligent retrieval, generating validated learning materials, and providing a conversational interface for seamless interaction.

System Overview

The platform supports two primary course components:

- Theory
- Lab

Admins (course instructors or TAs) manage content, while students interact with the system through both traditional UI elements and an AI-driven chat interface.

Part 1: Content Management System (CMS)

Build a content management system where admins can upload, organize, and maintain course materials, including:

- Lecture slides
- PDFs
- Code files
- Notes or supplementary references

Requirements

- Content must be categorized under Theory or Lab
- Students should be able to browse and access materials as references
- Metadata such as topic, week, tags, and content type should be supported

Part 2: Intelligent Search Engine (Syntax-Aware / RAG-Based)

Implement an intelligent search system that allows students to query course materials using natural language.

Requirements

- Semantic search beyond simple keyword matching
- Retrieval-Augmented Generation (RAG)–based approaches are encouraged
- Search results should return relevant documents, excerpts, or code snippets

Bonus: Syntax-aware or structure-aware search for lab/code materials

Part 3: AI-Generated Learning Materials

The platform should generate new learning materials based on existing course content. You can use the system you generated in part 2 as a tool for internal context and for external context a MCP server wrapper over wikipedia or other knowledgebase.

Input

A topic, concept, or natural language prompt from the user

Output

Theory

- Generated learning materials such as:
- Reading notes
- Slides
- PDFs

Content must be coherent, structured, and grounded in uploaded materials
Bonus: Ability to include relevant images, diagrams, or visual aids

Lab

- Generated code-centric learning materials

Code must be:

- Syntactically correct
- Relevant to the given topic
- Supported programming languages should be clearly specified

Part 4: Content Validation & Evaluation System

Since AI-generated content may contain inaccuracies, design a validation and evaluation mechanism for generated outputs.
Possible Approaches

- Syntax checking, compilation, or linting for generated code
- Reference grounding checks against uploaded materials
- Rule-based or rubric-based evaluation
- Automated test cases for lab code
- AI-assisted self-evaluation with explainability

Goal

- Ensure generated content is:
    - Correct

    - Relevant
    - Academically reliable

Part 5: Conversational Chat Interface

In addition to standard UI components, the platform must provide a chat-based interface through which users can access core system features.

Requirements

Through the chat interface, users should be able to:

- Search course materials (Part 2)
- Request summaries or explanations of existing content (Part 1)
- Generate theory or lab materials (Part 3)
- Ask follow-up questions contextually
- Receive grounded responses based on course data

Notes

- The chat interface should maintain conversational context
- Responses must be backed by uploaded materials when applicable
- The chat UI should act as an alternative interaction layer, not a replacement for the core system

Expected Deliverables

- A working prototype (web app, API, or demo)
- Explanation of system architecture and AI components
- Sample interactions (search, generation, chat)
- Description of validation strategy
- Evaluation Criteria
- Content organization and usability
- Effectiveness of semantic search and retrieval
- Quality and correctness of generated materials
- Robustness of validation mechanisms
- Design and usefulness of the chat interface
- Overall system coherence and user experience

Bonus Tasks

1. **Handwritten Notes Digitization:** Implement a feature to convert handwritten class notes into nicely organized, digitized notes, potentially using formats like LaTeX or similar structured markups for academic quality.
2. **Content-to-Video Generation:** Develop an ability to convert provided course contents into video summaries or explanatory clips, similar to tools like NotebookLM, to facilitate different learning styles.
3. **Community and Bot Support:** Integrate a social media-like community component where students can discuss problems. This component should include a bot support feature that can automatically generate and provide grounded replies when the intended receiver is unavailable.