CISmate Knowledge Base — Demo Ingestion PDF

Version 1.0 · August 2025

1. Overview

CISmate is a smart academic planning assistant for Computer Information Systems students. It helps students generate semester schedules, validate prerequisites, estimate course difficulty, and chat with an AI assistant grounded in university study plans. This PDF is designed to be ingested into a vector store (pgvector) to power Retrieval■Augmented Generation (RAG) for CISmate's chatbot.

2. Core Features

• GPA Calculator — compute cumulative and semester GPA updates using credit■weighted formulas.
• Course Directory — browse courses with descriptions, credit hours, prerequisites, and difficulty.
• Schedule Builder — generate conflict■free timetables respecting prerequisites and load limits.
• Prerequisite Validation — ensure required courses are completed or co■requisites met.
• Difficulty Insight — labels like Easy/Medium/Hard based on peer feedback and historic data.
• Advisor Review — submit plans for feedback and approval via an advisor portal.
• Chatbot — answers academic and policy questions using RAG over official materials.

3. Data Model (Simplified)

Entities include Course, Professor, Plan, Enrollment, and Resource. Course has fields: courseCode, courseName, creditHours, prerequisites, difficulty, category, lab/project flags, semestersOffered, assessment methods, and external resources.

3.1 Sample Courses

Code	Name	CrH	Prerequisites	Offered	Diffic
1904101	Fundamentals of IT	3	—	Fall/Spring	Easy
1901242	Data Structures	3	1902110 (Intro to Programming) or OOP	Fall/Spring	Mediu
1902224	Database Management Systems	3	1901242 (Data Structures)	Fall/Spring	Mediu
1902372	Software Engineering	3	1902224 (DBMS)	Fall/Spring	Mediu
1904120	Web Applications Development	3	1901242 (Data Structures)	Fall/Spring	Mediu
1902214	Advanced Java Programming	3	OOP, Data Structures	Fall/Spring	Hard

4. Policies & Rules (Demo)

• Maximum Credit Load: Normally 18 credit hours per semester for good standing; advisors may approve exceptions.
• Prerequisite Enforcement: Students must satisfy all prerequisites before enrolling.

• Attendance: Departments may require a minimum attendance (e.g., 80%); consult official policy.
• Graduation Requirements: Fulfill all university, faculty, and major requirements with minimum GPA thresholds.

5. GPA Calculation (Reference)

Cumulative GPA after a new semester can be computed by:
newCGPA = (prevCGPA × prevHours + termGPA × termHours) ÷ (prevHours + termHours)

6. Advisor Review Workflow

• Student builds a plan → system validates prerequisites and conflicts.
• Student submits plan → advisor receives a notification.
• Advisor reviews and comments → approve or request changes.
• Approved plan is stored as the official path for the next semester.

7. Retrieval Examples (for Chatbot)

The following Q&A; pairs are intentionally included to support RAG queries during development.
Q1: What are the prerequisites for Web Applications Development?
A1: Web Applications Development (1904120) typically requires Data Structures (1901242).

Q2: Is Data Structures offered in Fall?
A2: Yes, Data Structures is usually offered in both Fall and Spring.

Q3: How do I update my cumulative GPA after this term?
A3: Use the formula in Section 5, combining previous credits and GPA with this term's.

Q4: Can I take 21 credit hours?
A4: The normal maximum is 18; exceeding it usually requires advisor approval per policy.

8. Glossary

RAG. Retrieval■Augmented Generation: augment LLMs with relevant context retrieved from a knowledge base.
Embedding. A numeric vector representation of text for semantic search with pgvector.
Chunk. A small piece of text (e.g., 200–800 tokens) stored with its embedding.
Similarity Search. Find chunks close to a query vector by cosine distance.

9. Verification Note

For any official dates, policies, or graduation rules, always verify with the university's Registrar or the department handbook. This PDF is a demo and not an official document.