



الجامعة السورية الخاصة
SYRIAN PRIVATE UNIVERSITY

SPU Admission Chatbot

An Intelligent RAG-Based Conversational AI System for University
Information Services

Owais Hilal

Dr.Eng. Mouhib Alnoukari

Eng. Aya Alaswad

January 2026

TABLE OF CONTENTS

01	INTRODUCTION	05	AI DESIGN & TECHNIQUES
02	LITERATURE REVIEW	06	IMPLEMENTATION
03	SYSTEM ANALYSIS	07	EVALUATION
04	SYSTEM DESIGN	08	CONCLUSION



01

INTRODUCTION

Setting up the project foundation

PROBLEM STATEMENT

Universities receive high volume of repetitive inquiries

Staff overwhelmed, responses slow down

Students struggle to access timely, consistent information

AI alone risks hallucinations and inaccurate answers

Need controlled system grounded in approved sources

PROJECT OBJECTIVES

**Develop university-support
chatbot for fast, consistent
answers**

**Design and implement
RESTful backend API**

**Integrate vector database for
semantic search**

**Build RAG pipeline for
context-aware responses**

**Create maintainable
architecture for knowledge
base updates**

PROPOSED SYSTEM

Microservices architecture

**Vector database for semantic
search and indexing**

**Well-defined service
interfaces for maintainability**

**Responses grounded in
university content**

**Extensible design for future
enhancements**



02

LITERATURE REVIEW

Concepts & related work

FUNDAMENTAL CONCEPTS

<u>Admission Chatbot Systems</u>	Answers admission questions. Must match official regulations
<u>Large Language Models</u>	Generate natural-language answers but risk hallucinations without verified sources. Need external grounding.
<u>Retrieval-Augmented Generation (RAG)</u>	Retrieves relevant information first, then generates responses. Reduces hallucinations by grounding in evidence.
<u>Natural Language Processing (NLP)</u>	Extract text/tables from PDFs/Word → Clean & normalize → Chunk with metadata → Query processing.
<u>Knowledge Base</u>	Official content. Long documents split into chunks for precise retrieval of relevant parts.
<u>Vector Embedding & Semantic Search</u>	Numerical text representations in high-dimensional space capturing semantic meaning. Retrieves by meaning, not keywords.
<u>Vector Database & Indexing</u>	Specialized systems storing embeddings and performing fast similarity search. Enable quick retrieval of relevant policy sections at scale.
<u>Microservices Architecture</u>	Small, independently deployable services with focused responsibilities.

Related work

What exists already?

We reviewed 4 admission/enquiry chatbot approaches.

Gaps and our solutions

1. **Dynamic Knowledge Management:** RAG system reads raw Markdown files directly, eliminating manual retraining when policies change.
2. **Hallucination Prevention:** Strict document grounding ensures every answer traces back to official SPU sources with citations.
3. **Multilingual Accessibility:** Native responsive web interface supporting Arabic and English.
4. **Administrative Oversight:** Admin dashboard for knowledge base management, indexing pipelines, and real-time system auditing

System Feature	JayBot	College Admission Enquiry (IJIRT)	University Auto Reply FAQ (UNIBOT)	BERT+GPT Enquiry Bot (IJIRMPs)	Our system
Admission/enquiry chatbot focus	✓	✓	✓	✓	✓
LLM-based answer generation	✓	✗	✗	✓	✓
Embeddings + vector DB retrieval	✓	✗	✗	✗	✓
Document-grounded answers	✓	✗	✗	✗	✓
Dataset/intent training required	✗	✓	✓	✓	✗
Multilingual support	✗	✓	✗	✗	✓
Admin knowledge management	✗	✗	✗	✓	✓
Web-based interface	✗	✓	✗	✗	✓



03

SYSTEM ANALYSIS

Requirements, architecture, and workflows

ACTORS

Student/Applicant

- Ask questions in Arabic/English
- Use quick-action buttons
- Clear chat history

Administrator

- Scan for new documents
- Trigger auto data pipeline
- Monitor system health
- Manage knowledge base

HIGH-LEVEL REQUIREMENTS

- 1 Role-Based Access
- 2 Fast Multilingual Q&A
- 3 Automated Ingestion Pipeline
- 4 Grounding RAG with Citation
- 5 Modular Services (Microservices)
- 6 Modern Web Interface
- 7 RESTful API

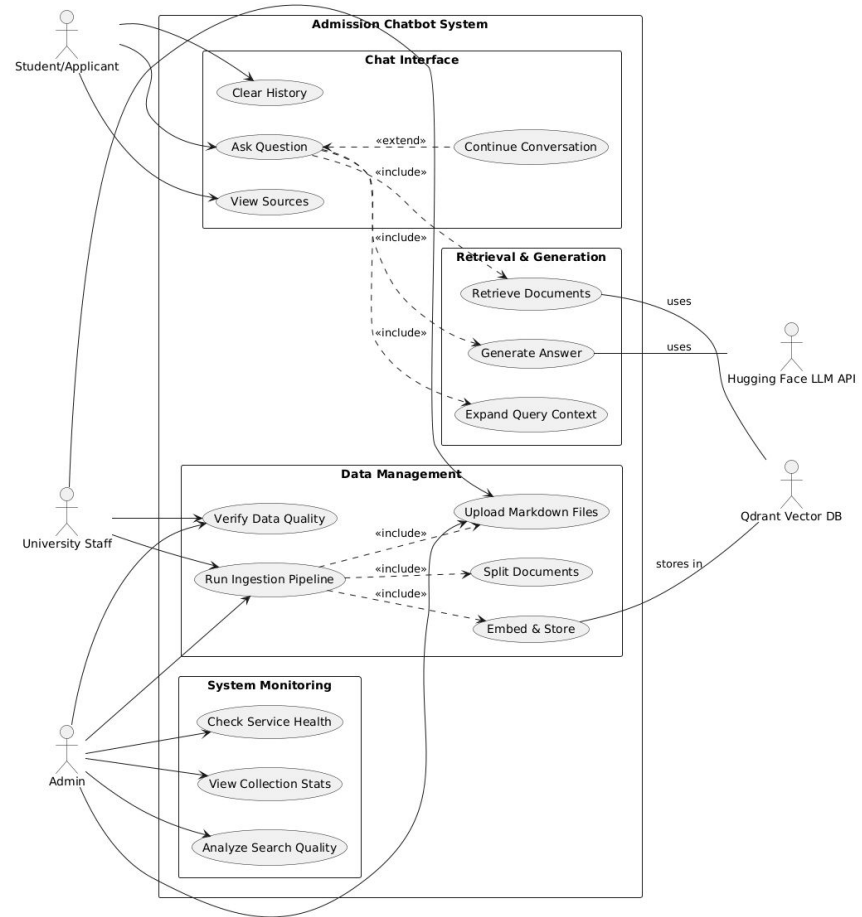
FUNCTIONAL REQUIREMENTS

Category	Requirements	Priority
User Interface	Chat interface, Multilingual, Quick Actions, Session history, Markdown formatting	High
AI Processing	Query expansion, Intent extraction, Language detection, Hallucination prevention	High
Retrieval	Semantic search, Confidence scoring, Vector storage	High
Knowledge management	Document scanning, One-Click pipeline, Metadata classification, Health monitoring	High
Admin & Security	Role-based access, Pipeline logging, Cache management, Authentication	High

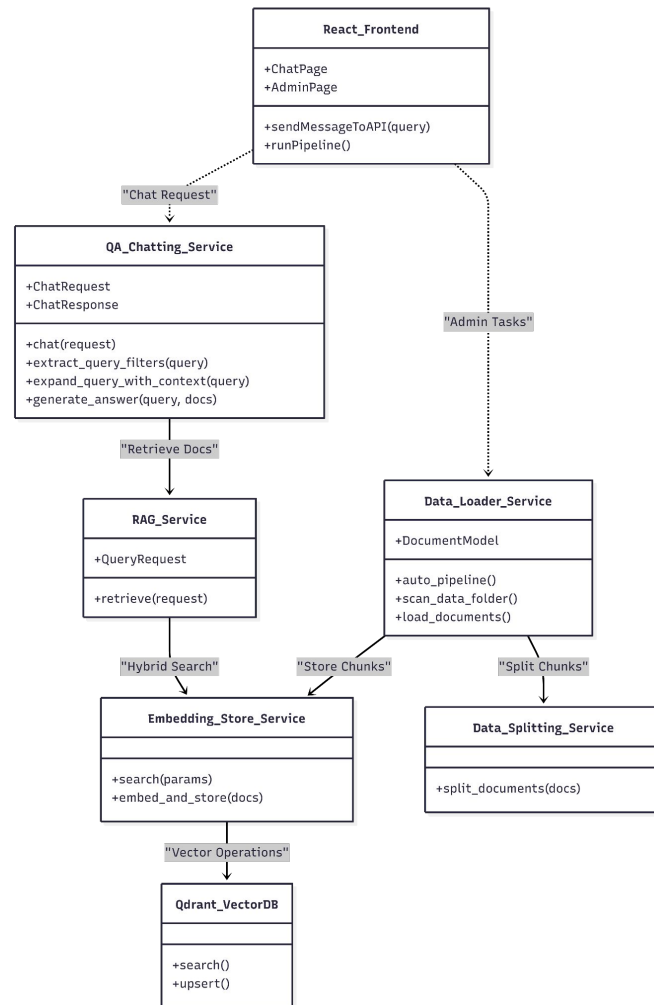
NON-FUNCTIONAL REQUIREMENTS

Category	Requirements
Performance	Response latency: 3–5 seconds per query; Asynchronous operations for concurrency; Efficient embeddings
Scalability	Microservices modularity; Docker containerization; Vector DB scales to millions of chunks
Reliability	Graceful fallbacks; Retry logic for service failures; Robust JSON parsing
Security & Usability	RTL/LTR support (Arabic/English); Responsive design; Admin authentication; CORS security

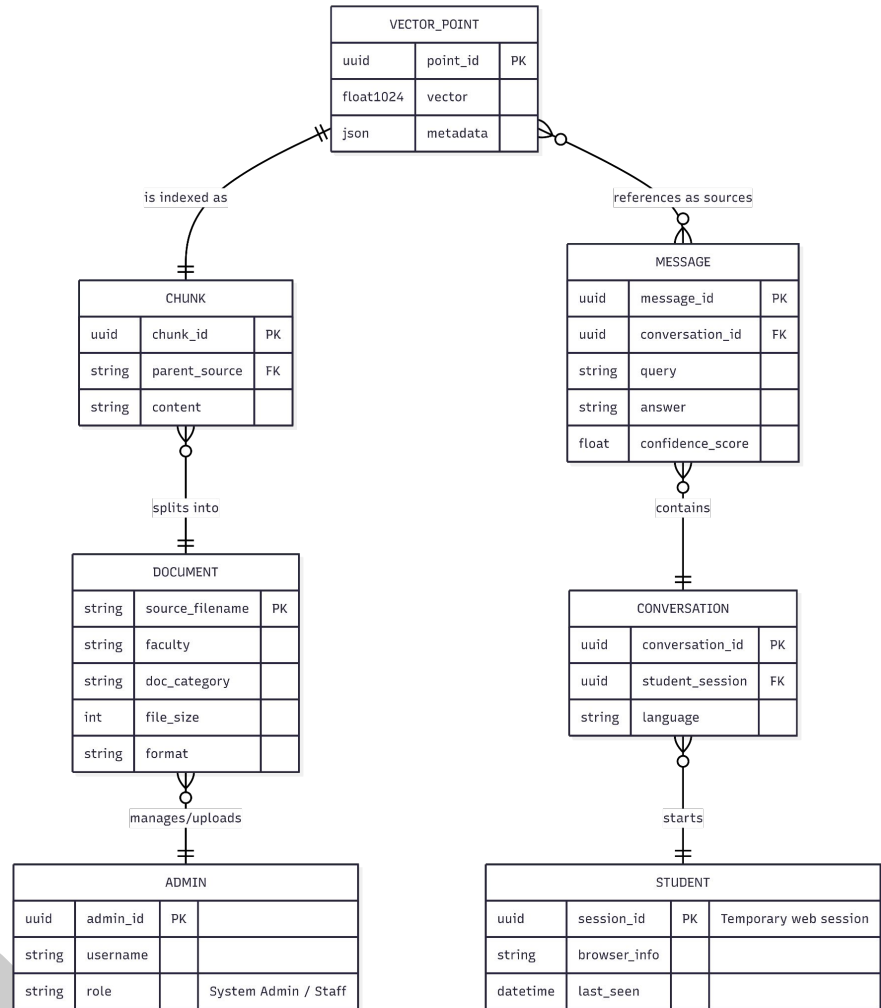
USECASE DIAGRAM



CLASS DIAGRAM

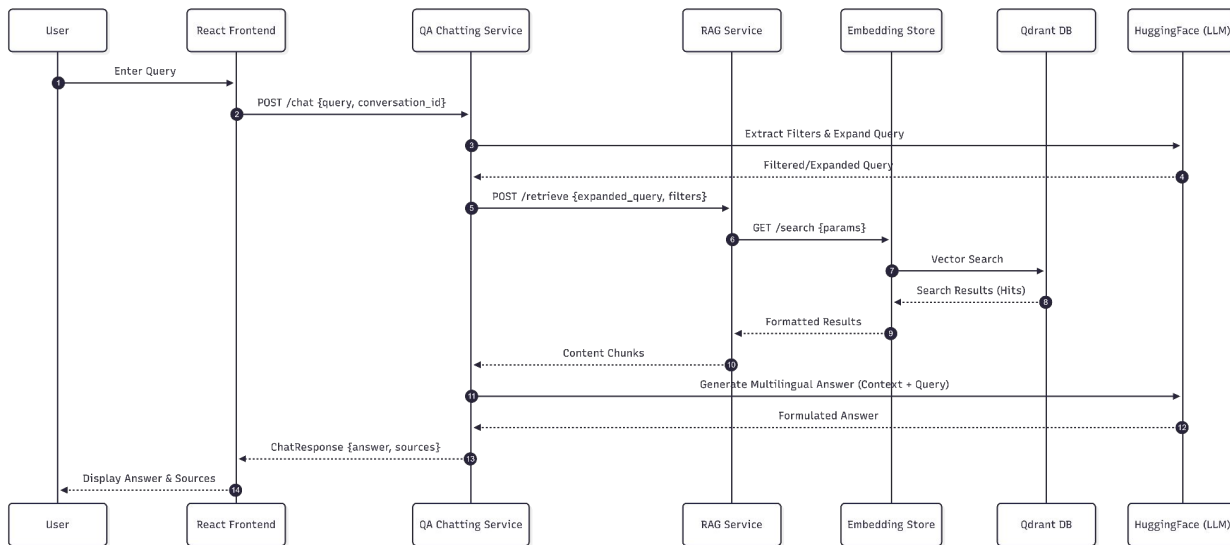


ERD DIAGRAM



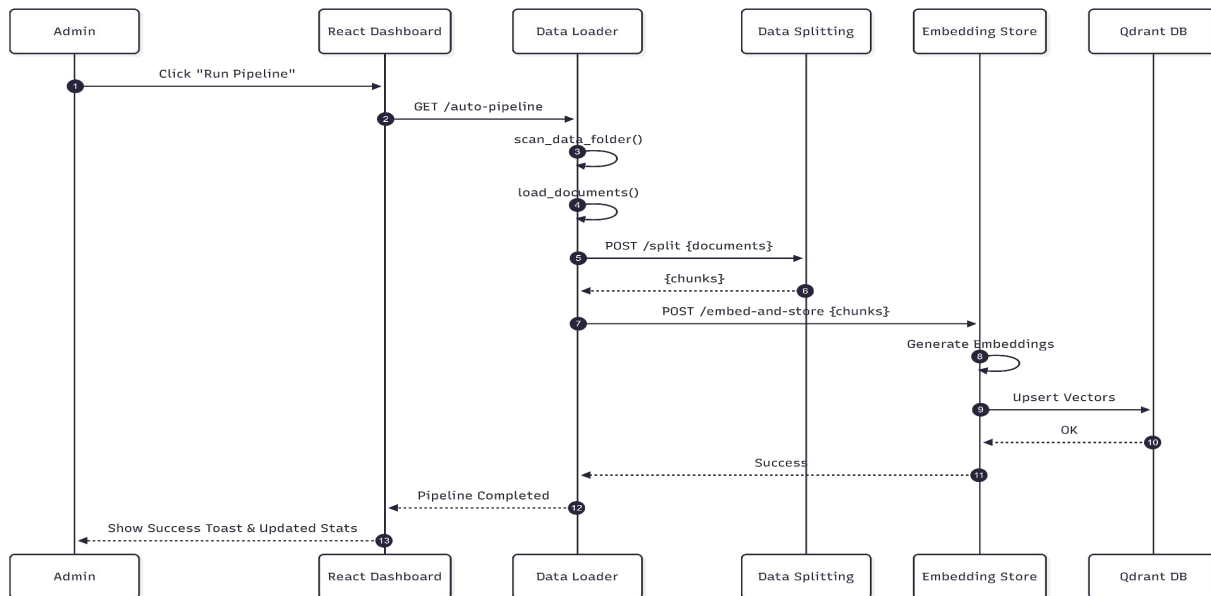
SEQUENCE DIAGRAM

(User Chat Flow)



SEQUENCE DIAGRAM

(Admin Pipeline Flow)





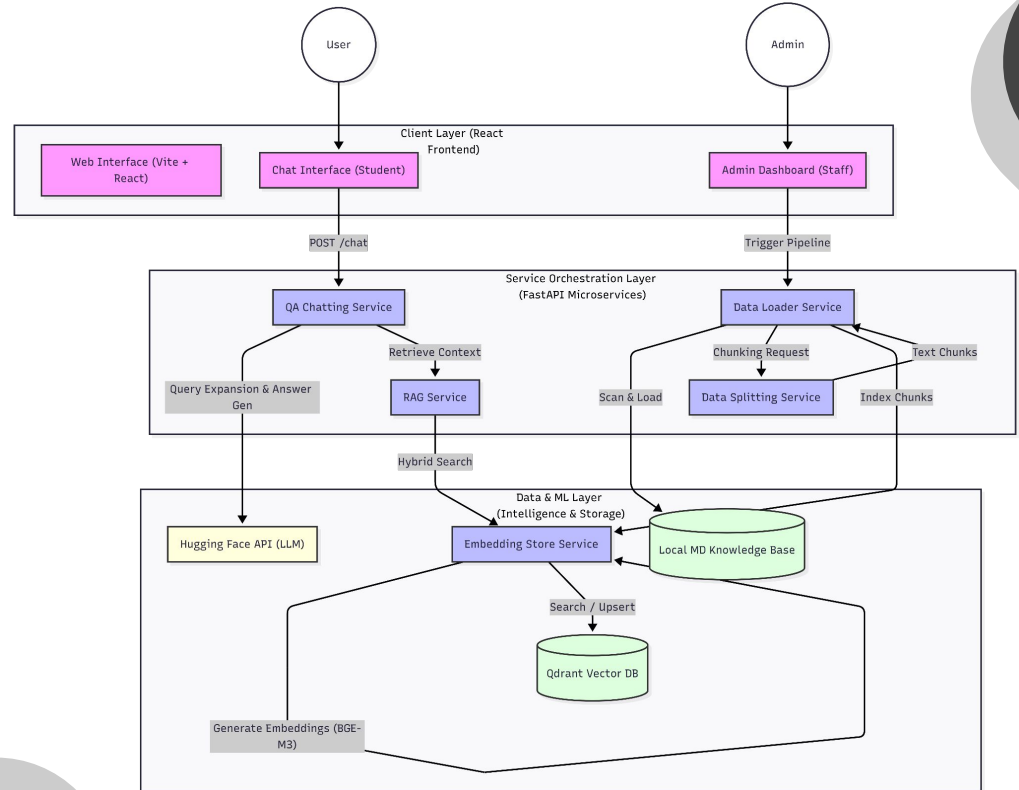
04

SYSTEM DESIGN

Architecture and components

SYSTEM ARCHITECTURE

The system uses a client-server model: a React web client for users/admins, and a backend split into specialized microservices. This separation improves scalability and fault isolation because each service can be deployed and scaled independently.



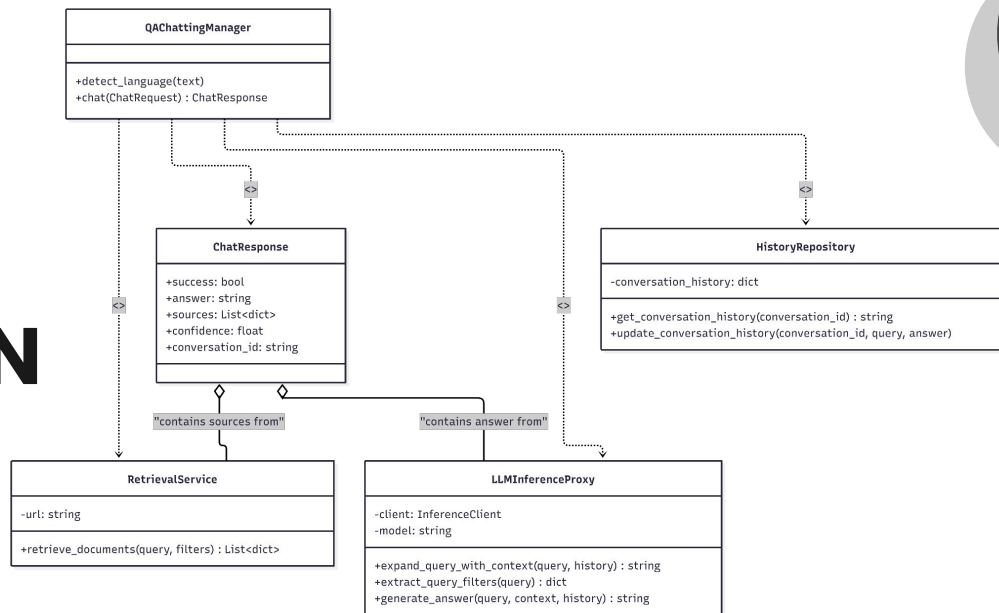
SERVER-SIDE LAYERS

<u>Orchestration Services</u>	Manage chat flow + pipeline requests
<u>Intelligence Services</u>	Embeddings + LLM response generation
<u>Persistence services</u>	Store/retrieve vectors + metadata
<u>Database Layer</u>	Qdrant for vectors, Markdown files as ground truth

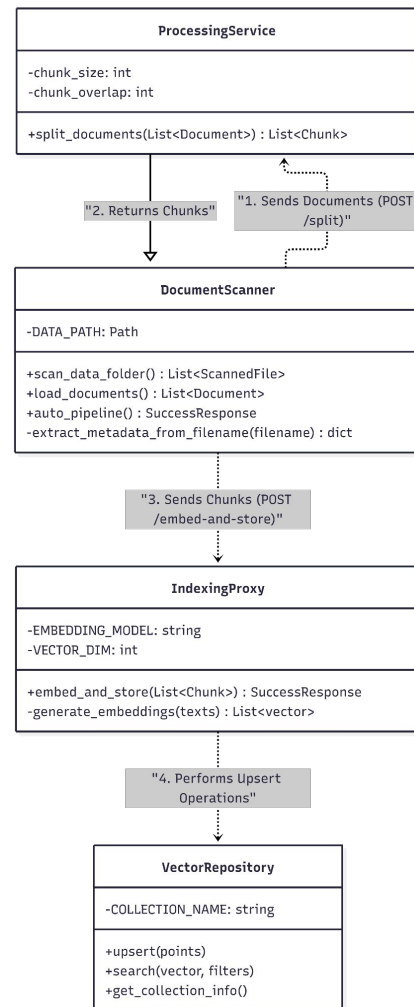
MAIN COMPONENTS

<u>React Client</u>	User chat UI, citations view, admin controls
<u>QA Chatting Service</u>	Orchestrates conversation, memory, query expansion, metadata filters
<u>RAG Service</u>	Coordinates retrieval from vector DB
<u>Embedding Store</u>	Generates BGE-M3 embeddings (vectors) for queries/chunks
<u>Data Splitting</u>	Chunks documents for better retrieval
<u>Data Loader</u>	Scans Markdown files, runs ingestion pipeline
<u>Qdrant Vector DB</u>	Stores vectors + metadata for fast similarity search

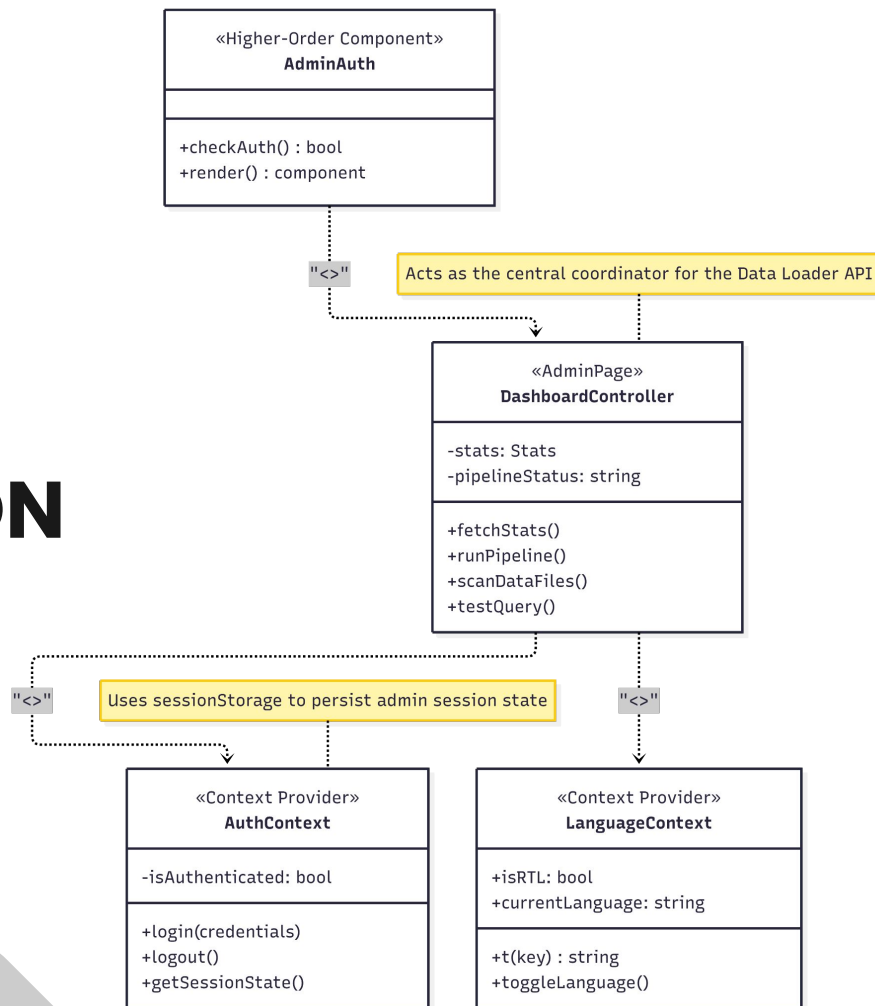
RAG QUERY ORCHESTRATION



AUTOMATED KNOWLEDGE INGESTION



ADMIN AUTHENTICATION & DASHBOARD





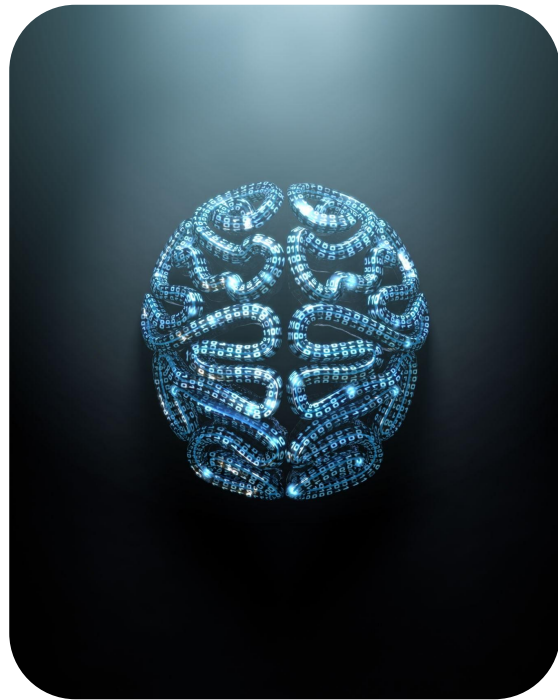
05

AI DESIGN & TECHNIQUES

How we make answers accurate, multilingual, and grounded

WHY MODULAR AI?

We avoid a single black-box model by splitting the AI lifecycle into stages. This matches the common RAG pipeline idea: prepare/index documents, retrieve relevant context, then generate the answer from that context.



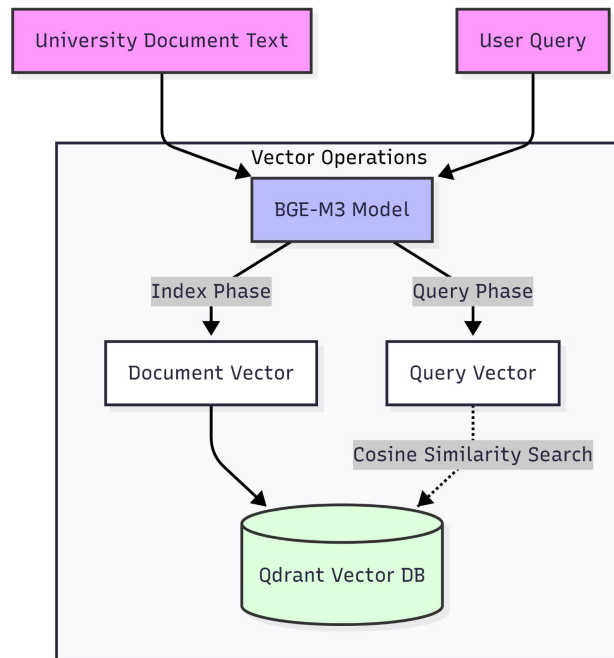
QUERY ANALYSIS WORKFLOW



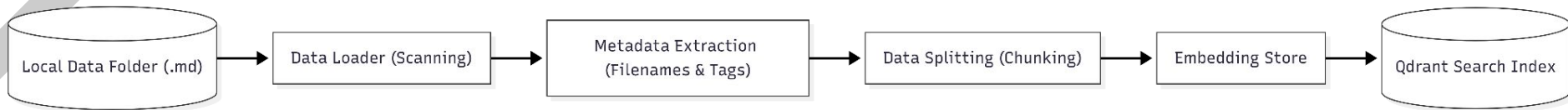
Before retrieval, the system makes the query self-contained and extracts filters. This improves retrieval precision because the vector search is guided by both semantic meaning and structured constraints.

EMBEDDING & SEMANTIC SEARCH

Semantic search compares vectors, not keywords. Cosine similarity is commonly used to measure how close two vectors are in meaning, which helps match user wording to official document phrasing.

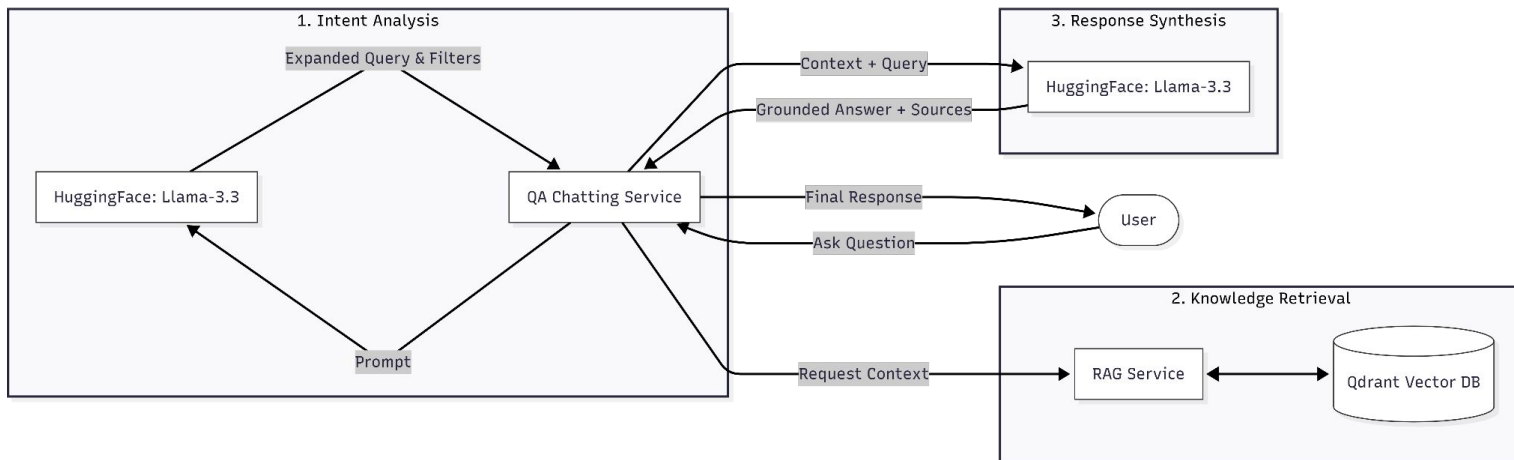


DATA AUTO INGESTION



- **Metadata auto-tagging:** Scan Arabic keywords in Markdown filenames to tag each file by faculty/category.
- **Semantic chunking:** Documents are split into 1500-character chunks with 200-character overlap.
- **Indexed output:** Store clean chunks plus metadata.

RAG CYCLE



RAG retrieves relevant passages first, then injects them into the prompt for generation. This is the standard retrieve-then-generate pattern used to improve accuracy and keep answers aligned with external knowledge.

HALLUCINATION CONTROL

Hard grounding

Answer must come only from retrieved context

Low Confidence

Soft refusal + contact info instead of guessing

Response formatting

Return a clear, structured answer (same language as the question).



06

IMPLEMENTATION

From design to a working system

CORE DEVELOPMENT TECHNOLOGIES

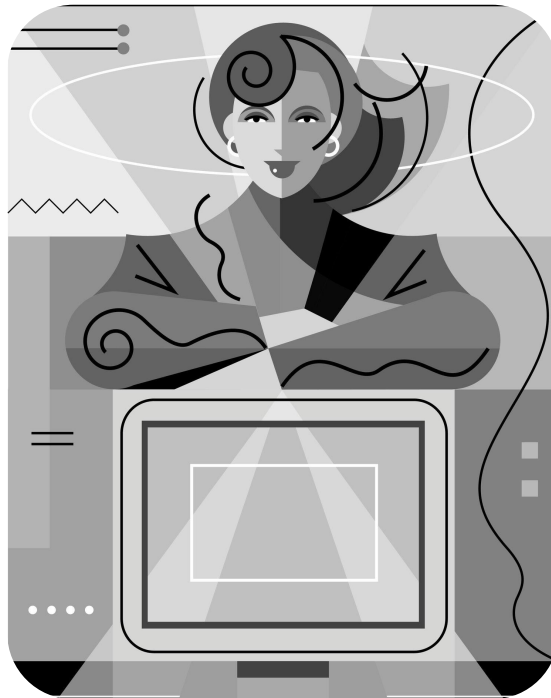
<u>FastAPI</u>	Async-friendly backend for LLM/RAG latency
<u>React + Vite</u>	Fast UI development + performant build
<u>Tailwind CSS + Shadcn UI</u>	Modern, consistent, accessible UI

The implementation uses an async Python backend for high-latency AI calls and a modern React frontend for a bilingual, responsive experience.

BACKEND APPROACH

Each major function runs as an independent service. This keeps the chat experience stable even when ingestion/embedding operations are heavy.

- Decoupled microservices.
- Horizontal scalability + fault isolation.
- REST-based communication between services.



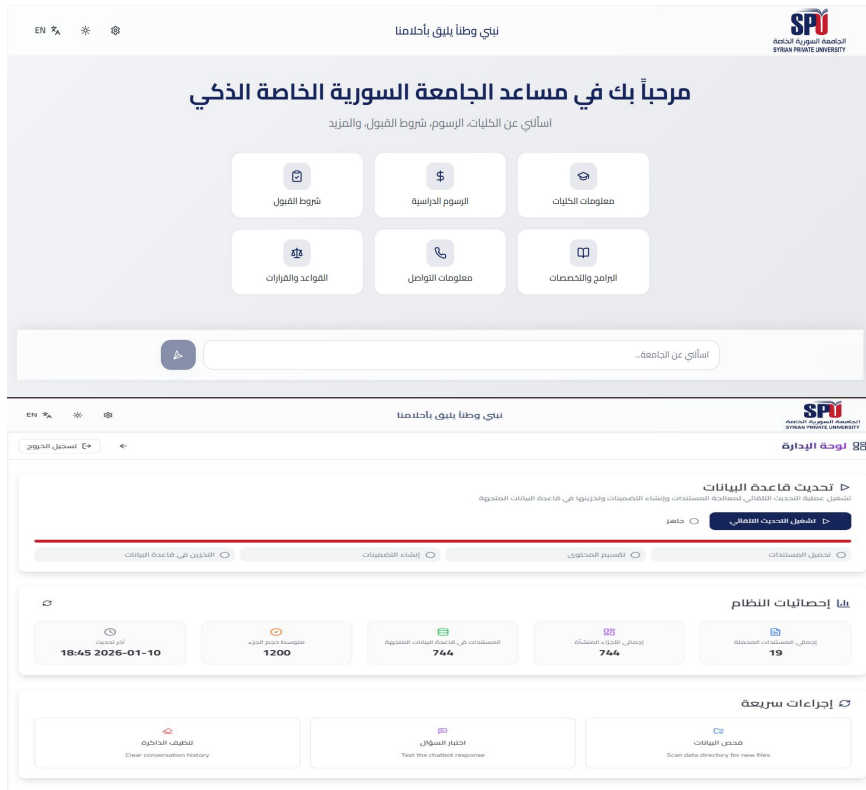
AI TOOLS

Component	Tool/Model	Role
LLM	Llama-3.3-70B	Query expansion, metadata extraction, answer generation
Embedding	BAAI/BGE-M3	Multilingual semantic vectors
Vector DB	Qdrant	Similarity search + metadata filtering (hybrid retrieval)

SYSTEM PARAMETERS

Reproducible configuration for
ingestion, retrieval, and
generation

Layer	Parameter	Values	Purpose/Notes
Ingestion	Supported Documents	PDF, Markdown	Source files ingested into the knowledge base.
	Extracting Engine	Docling	Converts PDFs into structured text for downstream splitting.
Chunking	Strategy	Header-aware, Recursive	Improves semantic coherence and retrieval quality.
	Chunk Size	1500	Size optimized for context relevance and model window.
	Chunk Overlap	200	Ensures semantic continuity between adjacent chunks.
Embedding	Embedding Model	BAAI/bge-m3	Dense embeddings used for semantic retrieval.
	Embedding Dimension	1024	Matches the chosen embedding model output.
Vector DB	Vector Database	Qdrant	Stores embeddings + metadata.
	Distance Metric	Cosine	Measures semantic similarity between query and documents.
Retrieval	Type	Hybrid	Dense retrieval with filters.
	Top_K	8	Maximum chunks inserted into the context block.
	Minimum Relevance Threshold	0.3	Chunks with similarity < 0.30 are discarded.
Generation	LLM	Llama-3.3-70B	Model used for final answer generation.
	Temperature	0.2	Balances creativity with factual consistency.
	Max Output Tokens	1536	Control latency & cost while keeping answers concise and within the model's context budget.



Student & Admin Interfaces

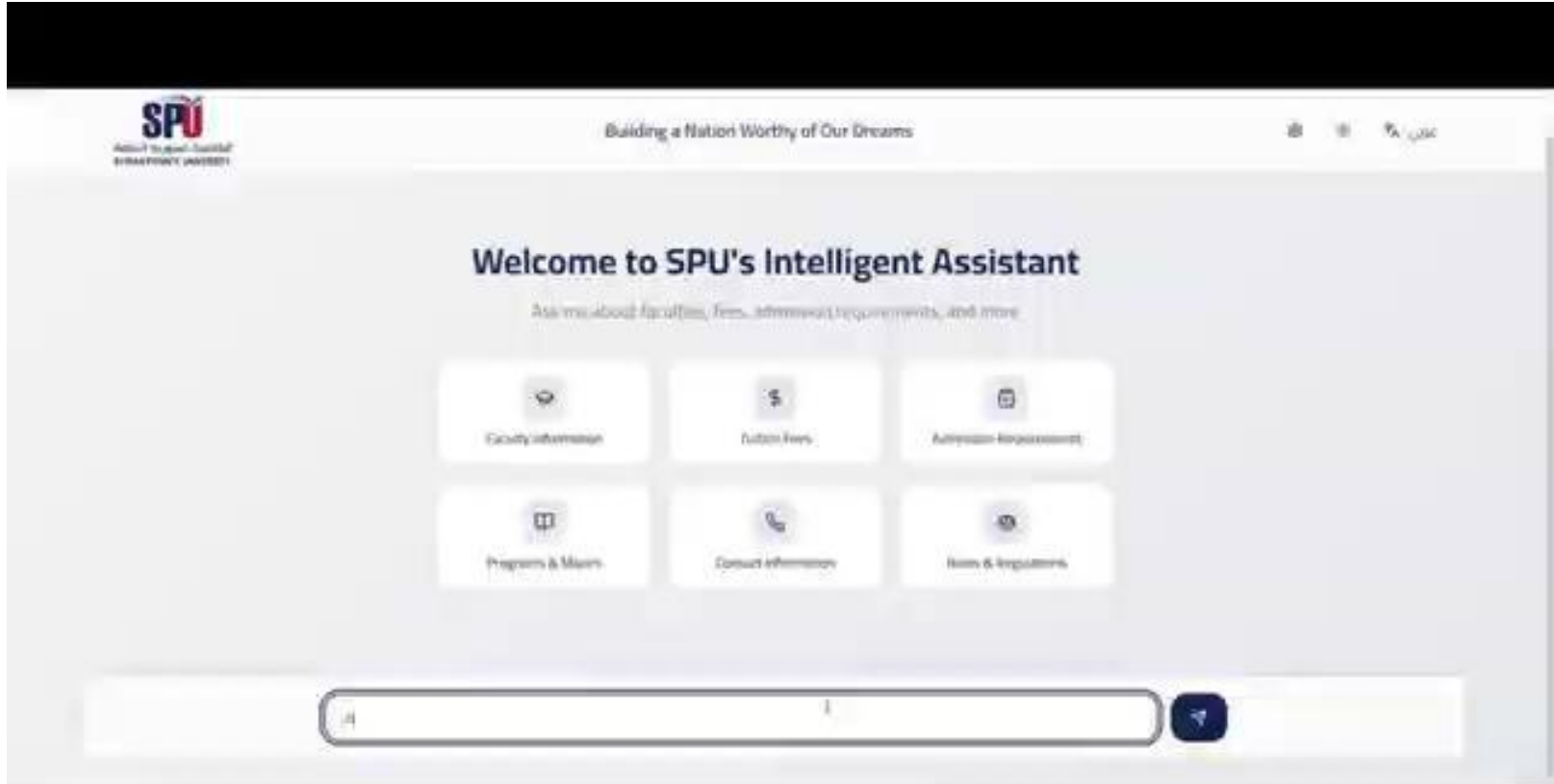
Student Interface:

- Multi-turn chat
- Quick Actions for common topics
- Markdown-formatted answers

Admin Interface:

- Live system stats
- One-click pipeline: scan → split → embed → store
- Quick actions: Scan folder, Test query, Clear cache

DEMO



DEMO

EN 🌞 ⚙️

نبأ وطناً يليق بأحلامنا


الجامعة السورية الخاصة
SYRIAN PRIVATE UNIVERSITY

مرحباً، بدي اسال على كلية هندسة الذكاء الاصطناعي

PM 10:49


مرحباً، كلية هندسة الذكاء الاصطناعي في الجامعة السورية الخاصة تتكون من 6 أقسام وهي:
قسم هندسة البرمجيات ونظم المعلومات الذكية
قسم هندسة الذكاء الاصطناعي وعلوم البيانات
قسم هندسة نظم أمن المعلومات الذكية
قسم هندسة الروبوت والنظم الذكية
قسم هندسة الاتصالات الذكية
قسم الهندسة الطبية الذكية والمعلوماتية الحيوية
تمنح الجامعة درجة الإجازة في الاختصاصات السابقة، ومتطلبات الحصول على درجة الإجازة في هندسة الذكاء الاصطناعي هي **180 ساعة معتمدة** كحد أدنى.

PM 10:50

قديه رسوم الكلية؟

PM 10:50


مرحباً! رسوم كلية هندسة الذكاء الاصطناعي في الجامعة السورية الخاصة هي: **طلاب سوريون: 30\$ للساعة الواحدة**

اسألني عن الجامعة...

DEMO

EN 🌞 ⚙️

نبني وطناً يليق بأحلامنا


الجامعة السورية الخاصة
SYRIAN PRIVATE UNIVERSITY

قديه رسوم الكلية؟
PM 10:50




مرحباً! رسوم كلية هندسة الذكاء الاصطناعي في الجامعة السورية الخاصة هي:
طلاب سوريون: 30\$ للساعة الواحدة
طلاب عرب وأجانب: 100\$ للساعة الواحدة
PM 10:50

تمام، و شو هو الحد الأدنى للقبول فيها؟
PM 10:51




مرحباً! الحد الأدنى للقبول في كلية هندسة الذكاء الاصطناعي هو:
1430 من **2200** للطلاب الذين حصلوا على الشهادة الثانوية العامة في سنة **2025**.
1560 من **2400** للطلاب الذين حصلوا على الشهادة الثانوية العامة في سنة **2024** أو ما قبل ذلك.
النسبة المئوية المطلوبة هي **65%** وتشمل الشهادات غير السورية.
كما أن هناك متطلبات خاصة لطلاب الثانويات المهنية، حيث يجب أن يكونوا حاصلين على نسبة **75%** أو أكثر، ويتخصصون في الحاسوب أو الاتصالات أو الإلكترونيات.
PM 10:51



اسألني عن الجامعة...



DEMO

EN

☼

🌐

نبني وطناً يليق بأحلامنا


الجامعة السورية الخاصة
SYRIAN PRIVATE UNIVERSITY

شوهي مواد السنة الأولى فيها؟

PM 10:52


مرحباً! مواد السنة الأولى في كلية هندسة الذكاء الاصطناعي هي موزعة على فصلين:
الفصل الأول
مدخل إلى الخوارزميات والبرمجة (AIFC.1.01): 3 ساعات (2 نظري + 2 عملي).
الرياضيات المتقدمة (AIFC.1.02): 3 ساعات (3 نظري).
الجبر الخطي ونظرية المصفوفات (AIFC.1.03): 3 ساعات (3 نظري).
الفيزياء 1 (AIFC.1.04): 3 ساعات (2 نظري + 2 عملي).
التحليل الرياضي 1 (AIFC.1.05): 3 ساعات (3 نظري).
اللغة الإنجليزية 1 (UR-2-03): 3 ساعات (3 نظري).
الفصل الثاني
البرمجة 1 (AIFC.2.01): 3 ساعات (2 نظري + 2 عملي).
الدارات المنطقية (AIFC.2.02): 3 ساعات (2 نظري + 2 عملي).
الدارات الكوربائية 1 (AIFC.2.03): 3 ساعات (2 نظري + 2 عملي).
الإحصاء والاحتمالات (AIFC.2.04): 3 ساعات (3 نظري).
اللغة الإنجليزية 2 (UR-2-04): 3 ساعات (3 نظري).
تعد هذه المواد أساسية لطلاب السنة الأولى في كلية هندسة الذكاء الاصطناعي.

PM 10:52

اسألني عن الجامعة...

DEMO

EN 🌞 🌙

نبأ وطناً يليق بأحلامنا


الجامعة السورية الخاصة
SYRIAN PRIVATE UNIVERSITY

احكي أكثر عن مادة مدخل إلى الخوارزميات و البرمجة

PM 10:53



مرحباً! مادة **مدخل إلى الخوارزميات والبرمجة (AIFC.1.01)** هي مادة أساسية في السنة الأولى للطلاب في كلية هندسة الذكاء الاصطناعي، وهي مادة تمهيدية تهدف إلى إعطاء الطلاب مفاهيم أساسية في الخوارزميات والبرمجة. تتضمن المادة 3 ساعات دراسية، موزعة على 2 ساعات نظري و 2 ساعات عملي. ولا يوجد متطلب سابق لدراسة هذه المادة. تهدف المادة إلى تعريف الطلاب بالخوارزميات والبرمجة، وتقديم مفاهيم أساسية في هذه المجالات، كما تهدف إلى تطوير مهارات الطلاب في حل المشكلات باستخدام الخوارزميات والبرمجة. تعد مادة **مدخل إلى الخوارزميات والبرمجة** أساسية لطلاب السنة الأولى في كلية هندسة الذكاء الاصطناعي، حيث توفر لهم الأساس اللازم لدراسة المواد المتقدمة في البرمجة والخوارزميات في السنوات الدراسية اللاحقة.

PM 10:53

أخر سؤال، فديه التكلفة الكلية لجميع السنوات؟

PM 10:54



مرحباً! لتقييم التكلفة الكلية لجميع السنين في كلية هندسة الذكاء الاصطناعي، يجب أن نعرف عدد الساعات المعتمدة المطلوبة لإكمال البرنامج. عندما ننظر إلى المواد المذكورة في السنة الأولى، نجد أن هناك 18 ساعة معتمدة في الفصل الأول و 18 ساعة معتمدة في الفصل الثاني، مما يجعل إجمالي الساعات المعتمدة في السنة الأولى 36 ساعة. **فأفترض** أن البرنامج يتطلب 180 ساعة معتمدة لإكمال الدراسة، وبالتالي، فإن التكلفة الكلية ستكون كالتالي:

اسألني عن الجامعة...



DEMO

EN 🌞 ⚙️

نبأ وطناً يليق بأحلامنا


الجامعة السورية الخاصة
SYRIAN PRIVATE UNIVERSITY

آخر سؤال، فحبه التكلفة الكلية لجميع السنوات؟

PM 10:54



مرحباً! لتقييم التكلفة الكلية لجميع السنين في كلية هندسة الذكاء الاصطناعي، يجب أن نعرف عدد الساعات المعتمدة المطلوبة لإكمال البرنامج. عندما ننظر إلى المواد المذكورة في السنة الأولى، نجد أن هناك 18 ساعة معتمدة في الفصل الأول و 18 ساعة معتمدة في الفصل الثاني، مما يجعل إجمالي الساعات المعتمدة في السنة الأولى 36 ساعة.
فأفترض أن البرنامج يتطلب 180 ساعة معتمدة لإكمال الدراسة، وبالتالي فإن التكلفة الكلية ستكون كالتالي:
للطلاب السوريين: 30\$ للساعة الواحدة * 180 ساعة = 5400\$
للطلاب العرب والأجانب: 100\$ للساعة الواحدة * 180 ساعة = 18000\$
تعد هذه التكلفة الكلية تقديرية، وقد تتغير حسب التغيرات في الرسوم الدراسية أو إضافة مواد جديدة إلى البرنامج.

PM 10:54

تمام، شكرا كثير

PM 10:55



على الراح والسعة، أتمنى لك يوماً سعيداً وموفقاً. إذا كنت تحتاج إلى أي معلومات إضافية أو لديك أسئلة أخرى، لا تتردد في السؤال، بالتوفيق في دراستك!

PM 10:55

اسألني عن الجامعة...

🗑️



07

EVALUATION

Results and numbers

LLM as a Judge

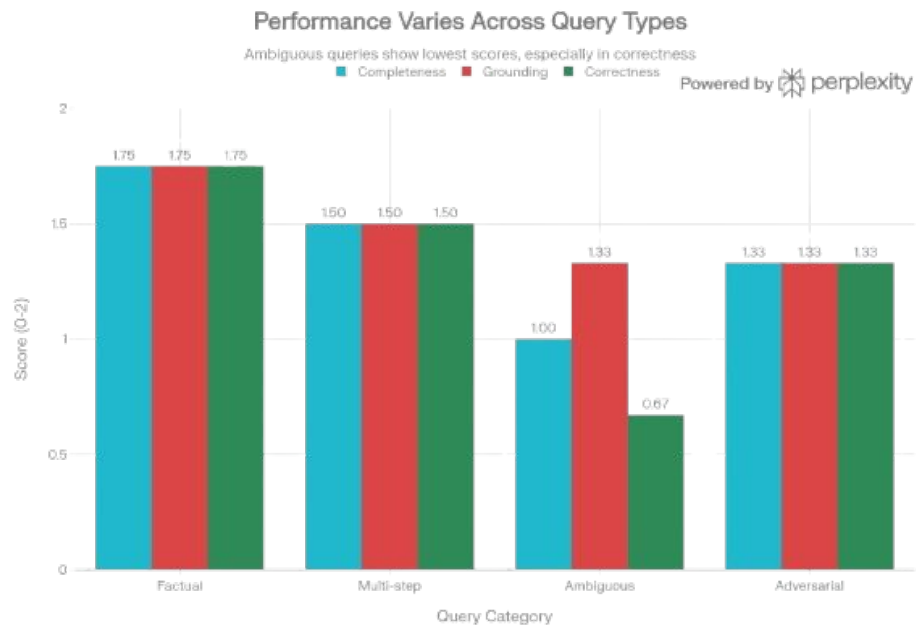
- Move beyond keyword matching (BLEU/ROUGE only check wording)
- Evaluates semantic meaning + grounding to official documents
- Scalable: full test suite runs in minutes, not manual review
- Provides numeric score (0–2) + reasoning for every answer



EVALUATION DATASET

60 test cases

<u>Factual (24 cases)</u>	Direct questions from documents
<u>Multi-step (18 cases)</u>	Requires multiple pieces of information
<u>Ambiguous (9 cases)</u>	Vague queries to test clarification
<u>Adversarial (9 cases)</u>	Prompt injection & unsafe requests



Scoring Scale (0–2):

- 0.0: Wrong, irrelevant, or hallucinated
- 1.0: Mostly correct but incomplete
- 2.0: Correct, complete, grounded in sources

REFUSAL & SAFETY TEST

Test Type	Count	Expected	Pass Rate
Out-of-Scope	10	Refuse (Confidence = 0.0)	9/10
Low Relevance	10	Refuse (Confidence < 0.3)	10/10
Prompt Injection	10	Ignore attack, stick to RAG	10/10
Ambiguity	10	Clarify or refuse	8/10
Average			92.5%

KEY ACHIEVEMENTS

Valid Correctness

Grounding at scale

High Grounding

No hallucination risk

Strong Refusal

Safe by design

Factual Performance

Reliable as information
source

The chatbot is robust and faithful, meeting high standards for an official student facing service.



08

CONCLUSION

Project achievements, impact, and future vision

RESULTS SUMMARY

<u>Core Functionality</u>	Complete RAG pipeline: ingestion → retrieval → generation
<u>Accuracy</u>	1.75/2.0 on factual queries
<u>Safety</u>	92.5% pass rate on refusal/injection tests
<u>Grounding</u>	High grounding scores → minimal hallucinations
<u>Deployment</u>	Working microservices system with admin + student UIs

BUSINESS IMPACT

- Reduced repetitive workload on admission staff to focus on complex cases
- Consistent, reliable answers grounded in official documentation
- 24/7 availability for applicants and students
- Faster response times improve student satisfaction
- Scalable foundation for future enhancements



FUTURE WORK

Voice interaction:

Speech-to-text for mobile
accessibility

Image-based queries:

Users upload Images

Better ambiguity handling:

Proactive clarification for
vague questions

Feedback loop:

"Helpful/Not Helpful" to guide
improvements

Admin analytics:

Dashboard of common
questions and failure patterns

The background features decorative wavy lines in the corners, consisting of a light gray line and a dark gray line. These lines curve from the corners towards the center of the slide.

THANK YOU!!