C-LASS

CYBERSECURITY LEARNING
WITH AI FOR STATIC SYSTEMS

Design Review 3

Kayden Vicenti
Team Lead, Customer Communicator

Sean Golez
Recorder, and Architect

Colton Leighton
Database Manager

William Barnett
Release Manager, and Architect

Scott LaRocca
CS Faculty Mentor

Dr. Lan Zhang
Client

Problem Statement and Solution

PAIN POINTS

- No centralized source for class concepts
- Lack of structured, guided study materials
- Excessive dependency on direct faculty support
- GenAI is unreliable and inaccurate for niche topics

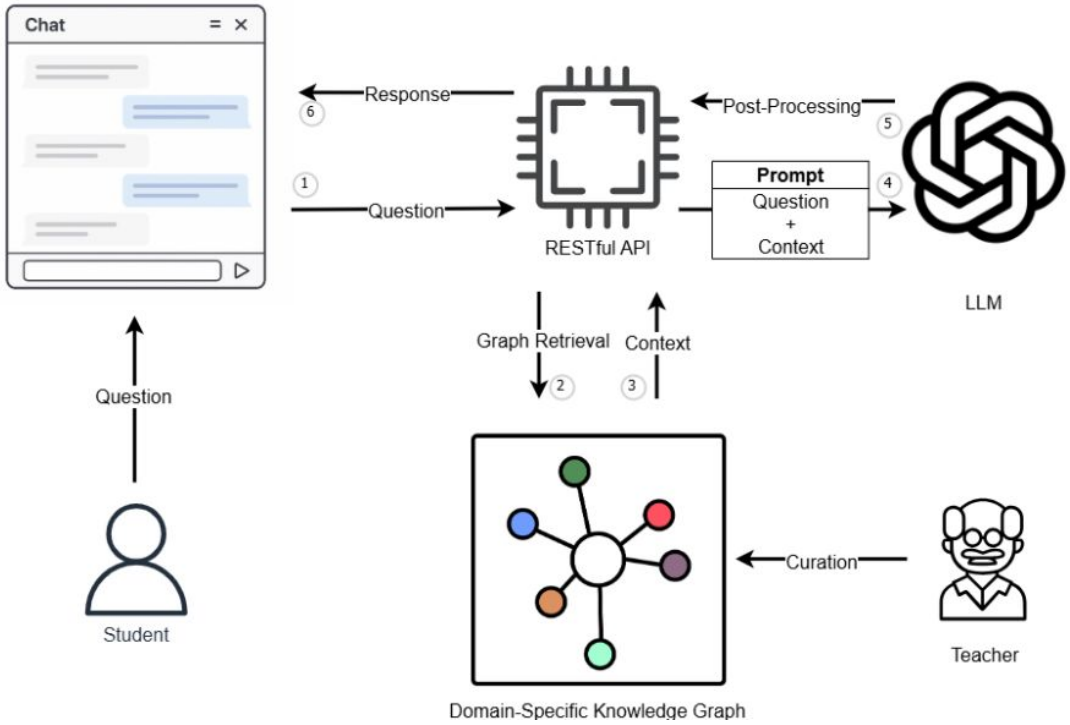
SOLUTION

- An AI chat application that:
- Aligns with instructor requirements
- Has domain specific knowledge
- Gives reliable and accurate tutoring

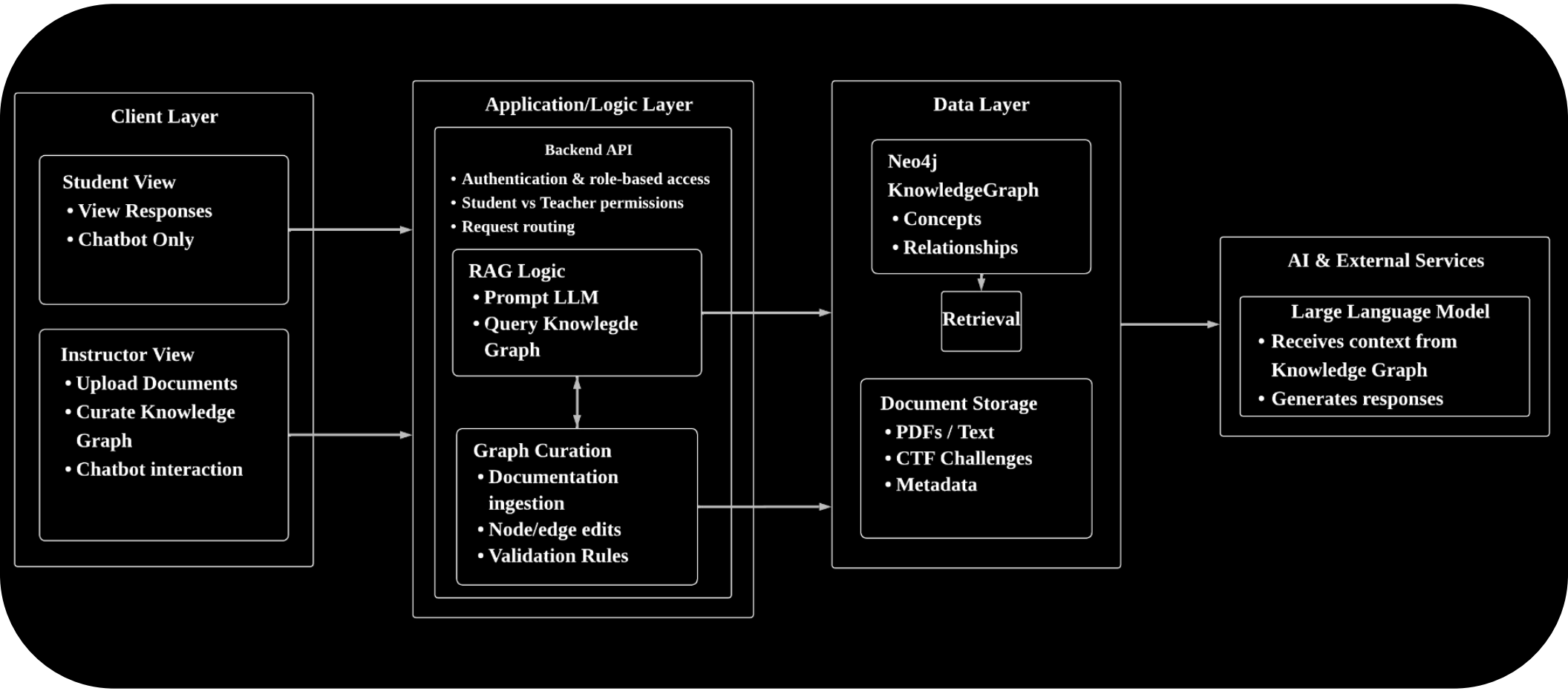
SOLUTION

KGTutor

- A chatbot that students are encouraged to use
- Instructor curates the knowledge graph
- The “tutor” utilizes the context to teach students



Software Design Overview



Testing Plan

UNIT TESTING

- Test normal inputs, boundary conditions, and invalid inputs for each unit
- Authentication such as password hashing and signin/signup
- Pytest to test backend logic

INTEGRATION

- Verify connected components work correctly:
  - RAG chat generation
  - Knowledge graph curation
  - PDF upload
  - Multi chat persistence

USABILITY

- Client feedback, student task completion, LLM response evaluation
- User will rate chatbot responses based on helpfulness, clarity, and guidance level
- Knowledge graph curation ease of use testing

LLM TESTING

- A collection of 5-10 CTF queries are evaluated against three criteria: fabrication, instructional tone, and custom prompt compliance
- Regression to catch degradation after graph or prompt changes

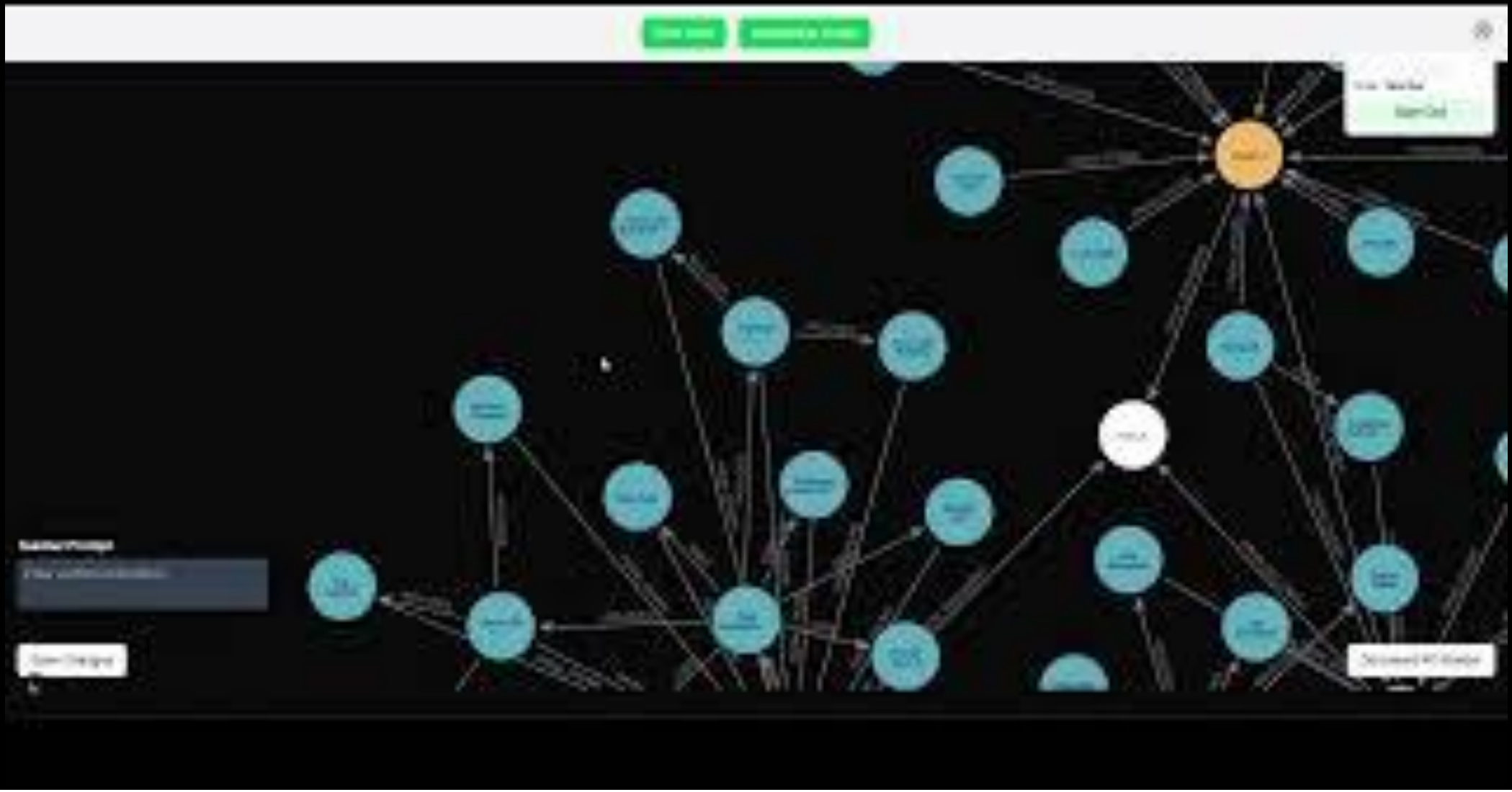
Challenges

KG Visualization	• Dependency on Neo4j database and wrappers • Mitigated through consistent version building and testing
Security	• Prompt-injection risks • Mitigated through sanitization and validation layers
LLM Accuracy	• RAG and strategic prompt engineering • Accuracy Evaluation
Scalability	• LLM and database hosting is expensive • No funding to keep up with linear cost scaling • This product is a proof-of-concept

CLASS Implementation Timeline - Gantt Chart

TASK TITLE	PHASE ONE			PHASE TWO			PHASE THREE			PHASE FOUR		
	WEEK 5	WEEK 6	WEEK 7	WEEK 8	WEEK 9	WEEK 10	WEEK 11	WEEK 12	WEEK 13	WEEK 14	WEEK 15	WEEK 16
	2/9/2026	2/16/2026	2/23/2026	3/2/2026	3/16/2026	3/23/2026	3/30/2026	4/6/2026	4/13/2026	4/20/2026	4/27/2026	5/4/2026
Setup & Design												
Finalize Software Design												
UGRADS Registration												
Design Review #2												
Alpha I/II Tech Demo												
Development												
Database												
Hosting Environment												
Application Logic												
User Interface												
Design Review #3												
Full-Stack Integration												
Capstone Poster												
Testing & Optimization												
Software Tesing Plan												
Unit Testing												
Integration Testing												
LLM Testing												
User Testing												
Finalization & Delivery												
Final Team Website												
UGRADS Capstone Presentation												
Acceptance Testing												
Final Project Report												
User Manual												
Deployment												

Alpha II Demonstration



Conclusion

Summary

- Guided personal tutor
- Education-oriented Chatbot
- Class curated content relevant to course load
- Knowledge Graph visualisation

Next Steps

- Refining UI/UX
- Integrate Testing
- Refining RAG & model pipelines
- Documenting the project and creating a user manual

Questions?

THANK YOU