4365 Final Presentation: Clarify

Tyler Cady

Georgia Institute of Technology

Introduction

Overview

Clarify is a next-generation educational forum designed to combat knowledge obsolescence.

It integrates traditional forum structures with modern features:

(1) **streamlined course setup:** instructors can upload Canvas CSVs for automatic enrollment
(2) **anonymous identify:** users are assigned persistent, forum-specific pseudonyms
(3) **AI-generated thread summaries** and **intelligent search** (text & semantic)
(4) **topical knowledge graphs:** interactive visual aids connecting related concepts

Motivation

Inspired by firsthand TA experience at Georgia Tech.

Addresses issues in existing forums:

(1) disorganization in large classes
(2) semester-based silos leading to repeated questions
(3) poor surfacing of relevant past knowledge

Clarify aims to make information timeless, discoverable, and accessible.

Skill Growth

L1: general technical skills – daily practice with TypeScript, SQL, and Supabase

L2: integration & problem-solving – combining AI, authentication, and frontend/backend tools

L3: trend recognition – adapting cutting-edge AI and vector search technology for education use

Proposed Solution

Related Work

Ed Discussion: rich media support but lacks content discoverability

Piazza: good analytics and status tracking but too externally focused

Clarify redefines forums as knowledge ecosystems, not just discussion boards.

Technologies

Core Stack: Next.js, Supabase, Tailwind, TypeScript

AI & Search: OpenAI (GPT + embeddings), Postgres vector search

Visualization: D3.js for graph rendering

Deployment & Security: Vercel, Resend (emails), Cloudflare DNS

Flexible tech selection based on practical evaluation.

Design & Implementation

UI inspired by modern SaaS design (e.g., Stripe, Brex).

Built in feature-first order for scalability and stability.

Development flow:

(1) auth & enrollment, **(2)** core interactions (threads/comments), **(3)** AI summaries/search, and **(4)** graph-based knowledge graphs

Figma used for key pages as-needed, GitHub used for tracking/versioning.

Evaluation

Success criteria:

(1) fully deployed system
(2) at least 3 of 4 advanced features complete
(3) pre-populated data for demo reliability

Outcome: all criteria met and exceed = system fully functional

Datasets

Added real data to avoid an empty, unrealistic system:

- **Coursera Forums** – 100k real course threads
- **Education Dialogue Dataset** – student-teacher LLM conversations
- **Stack Exchange Dump** – massive Q&A corpus

Used Python scripts to parse and populate Supabase with embeddings.

Timeline

Task Breakdown & Schedule

Weeks	Milestone	Description
5-7	Authentication & Enrollment	Canvas CSVs, 2FA setup via Resend, and Supabase authentication
7-9	Core Forum Development	Anonymous threads, comments, tagging, and upvotes
9-11	AI Summarization & Search	GPT summaries and full-text + semantic search
11-13	Knowledge Graphs	Semantic thread visualization via D3.js
13-15	Testing & Deployment	Bug fixes, pagination, and final hosting on Vercel

Work Distribution

Solo-built by Tyler Cady – full-stack design, development, and deployment.

Executed Solution & Results
(with demos)

Authentication & Enrollment

Canvas roster schema used to auto-assign roles & accesses.

Resend integration for 2FA emails.

Admin dashboard to manage course creation & enrollment.

Students added automatically based on uploaded CSV.

Core Forum Development

Features include:

(1) Anonymous discussion with persistent, unique pseudonyms
(2) Thread creation/editing/deletion
(3) Commenting, replying, and upvoting

AdjectiveAnimal handles per user/thread to preserve anonymity.

Built-in SQL procedures for cleanup (e.g., course deletion).

AI Summarization & Search

GPT-3.5 used to summarize threads + comments in real-time.

Search options:

(1) full-text search (trigram match)
(2) semantic search (cosine similarity on OpenAI embeddings)

Search scoped to course or dashboard-wide and there exists filtering by tags.

Summaries regenerate only if content updates (for performance & cost).

Knowledge Graphs

Interactive, dynamic visualizations:

(1) force-directed & bubble graphs
(2) responsive, zoomable UI with thread details on click

Thread relationships determined by similarity thresholds on embeddings.

Backend built to scale – fetching & rendering graphs efficiently even with large data.

Testing & Deployment

Conducted manual + experimental testing during development.

Client-side pagination added for visual performance.

Schemas finalized, dependencies resolved.

Live now at: https://clarify-4365.vercel.app/

Conclusion

Self-Evaluation

Scope: broad and impactful – multiple novel features fully delivered

Match: near-perfect execution of original proposal

Factual: real data, real AI integration, and live app proves viability

Future Work

Feature additions: **(1)** image/media input, **(2)** character limits, and **(3)** post analytics

Tech improvements: **(1)** refined AI prompt engineering, **(2)** tuned graph hyperparameters, and **(3)** explore advanced visualization methods (timeline graphs, radial trees, etc.)

Deliverables

GitHub: https://github.com/tylerrcady/clarify

Live app: https://clarify-4365.vercel.app/

Admin access available upon request via email.

Documentation: https://github.com/tylerrcady/clarify/blob/main/final-report.pdf

Skills Learned

Frontend/backend mastery with real-world tools.

AI integration with cost/performance optimization.

Full-stack ownerships from architecture to deployment.

Realistic project planning, timeline adaptation, and execution.

Thank you!