Dashboard Overview

Figure 1: Dashboard Overview

Question Clusters

Figure 2: Question Clusters

OpenTA Changelog

[Unreleased] - 2025-11-09

Professor Console - New Features

1. Dashboard Overview Get a comprehensive view of student engagement and course analytics at a glance.

Features: - Total questions asked by students - Average response confidence scores - Active students count - Unresolved questions tracker

---

2. Question Clustering Automatically group similar student questions using AI-powered semantic analysis.

Features: - Semantic clustering using embeddings and cosine similarity - Adjustable similarity threshold (default: 0.75) - View all questions within each cluster - Identify common pain points and knowledge gaps

How it works: - Questions are embedded using the same model as retrieval - Similar questions are grouped based on semantic similarity - Professors can see patterns in student confusion

---

3. Canonical Answers Create verified, reusable answers for frequently asked questions.

Features: - Create canonical answers for question clusters - Markdown editor with preview - Add citations from course materials - Publish/unpublish answers - Students automatically see canonical answers when asking similar questions

Workflow: 1. Review a question cluster 2. Write a comprehensive answer in markdown 3. Add relevant citations 4. Publish for students to access

Benefits: - Reduce repetitive answering - Ensure consistent, high-quality responses - Students get instant verified answers - Build a knowledge base over time

---

Canonical Answer Creation

Figure 3: Canonical Answer Creation

Student FAQ Page

Figure 4: Student FAQ Page

4. Student FAQ Page Students can browse all published canonical answers in one place.

Features: - Clean, searchable list of professor-verified answers - Expandable cards for each answer - Citations displayed as pill badges - Visual indicator for professor-verified content

---

5. Enhanced Chat Integration Canonical answers are automatically checked before AI processing.

Features: - When students ask questions, system first checks for matching canonical answers - If found (similarity > 0.75), returns professor-verified answer with high confidence - Marked with [**Professor-Verified Answer**] badge - Falls back to AI-generated response if no match

Benefits: - Faster responses for common questions - Guaranteed accuracy for frequently asked topics - Reduced AI hallucination risk

---

6. Student Analytics Track individual student engagement and identify students who need help.

Features: - Questions asked per student - Average confidence of responses received - Last activity timestamp - Identify struggling students

---

7. Content Gap Analysis Identify topics where students are confused or course materials are insufficient.

Features: - Low-confidence response tracking - Frequently asked topics without good answers - Unresolved question patterns - Helps improve course materials

---

Chat with Canonical Answer

Figure 5: Chat with Canonical Answer

Student Analytics

Figure 6: Student Analytics

Content Gaps

Figure 7: Content Gaps

Frontend Redesign

New Design System Implemented a calm, minimalist aesthetic inspired by modern AI interfaces.

Design Features: - Warm neutral color palette (#FAF7F2 canvas, #E46E58 coral accent) - Lora serif font for headings, Inter sans for body - Pill-shaped buttons with subtle shadows - Lucide icons (1.5px stroke, rounded style) - Smooth micro-interactions (120-150ms transitions)

Improved Navigation Unified sidebar navigation across all student-facing pages.

Features: - Consistent layout: Chat, FAQ, Study Plan, Assignment Help - Clear visual hierarchy with active states - "New Chat" button prominently displayed - Recent chats section (placeholder)

Optimized Layouts Better use of screen space across all pages.

Improvements: - Compact headers save vertical space - Wider content areas (max-w-3xl to 5xl depending on page) - Smaller input areas (2 rows instead of 3) - More room for chat messages and content

---

Backend Improvements

Multi-Agent Framework Refactored backend to use a multi-agent architecture.

Components: - **Orchestrator**: Routes requests to appropriate agents - **QA Agent**: Handles general questions with retrieval - **Assignment Helper**: Pro-

New Design System

Figure 8: New Design System

Student Navigation

Figure 9: Student Navigation

Layout Comparison

Figure 10: Layout Comparison

vides Socratic guidance - **Study Plan Agent**: Creates personalized study plans

Benefits: - Better separation of concerns - Easier to add new agent types - Shared memory and tool system - More maintainable codebase

---

Professor Service New service layer for professor console features.

Features: - Question logging and tracking - Semantic clustering algorithms - Canonical answer management - Analytics aggregation - Guardrail settings

---

Technical Updates

Dependencies Added: - `lucide-react` - Icon library - `react-markdown` - Markdown rendering in chat - `openai>=1.0.0` - OpenAI API integration

Configuration: - Extended Tailwind config with design system tokens - Added CSS variables for theming - Google Fonts integration (Lora + Inter)

Installation & Setup

Prerequisites

- Node.js 18+
- Python 3.9+
- OpenAI API key

Backend Setup

```
cd backend
pip install -r requirements.txt
cp .env.example .env
# Add your OPENAI_API_KEY to .env
python main.py
```

Frontend Setup

```
cd frontend
npm install
npm run dev
```

Access

- Student Console: http://localhost:3000/student
- Professor Console: http://localhost:3000/professor
- Login: http://localhost:3000/login

Future Enhancements

Planned Features: - [ ] Real-time collaboration for canonical answers - [ ] Export analytics reports - [ ] Email notifications for professors - [ ] Dark mode support - [ ] Mobile-responsive design - [ ] Search functionality in FAQ - [ ] Question upvoting by students - [ ] Integration with LMS platforms

---

Contributors

- Jessica Wang (@jessicayanwang)

License

MIT License