Lesson Plan Generator Application Report

Prepared by: Ergeta Muca
Date: October 21

Task 1: Define your Problem

Problem Statement

Teachers spend excessive time manually creating comprehensive lesson plans that align with curriculum standards, incorporate diverse learning materials, and accommodate various student needs, resulting in reduced time for actual teaching and student interaction.

User Persona

- **Primary User**: K-12 educators, curriculum coordinators, and instructional designers who need to create standards-aligned, differentiated lesson plans efficiently.

Why this is a problem

Teachers typically spend 5-10 hours creating a single comprehensive lesson plan, including researching standards, finding appropriate materials, and designing activities for diverse learners. This time-intensive process often leads to generic, non-personalized lesson plans that don't effectively address individual student needs or leverage current educational resources.

Successful outcome

The app successfully generates comprehensive, standards-aligned lesson plans in under 2 minutes that include learning objectives, structured activities, assessment strategies, and external resources, while accommodating specific student group needs (ESL, ADHD, learning disabilities, gifted students) and integrating real-time educational content from multiple APIs.

Task 2: Propose a Solution

Solution Proposal

The Lesson Plan Generator is an AI-powered application that uses RAG with advanced agentic reasoning to automatically create comprehensive lesson plans. The system combines curriculum standards retrieval, external resource integration, and student-specific accommodations through a multi-agent workflow that processes user inputs and generates complete educational plans with activities, assessments, and materials.

User Questions it tries to solve

Category	**Question / Problem it tries to solve**
Curriculum Alignment	What learning objectives should I include for this topic and grade level?
Activity Design	How can I structure engaging activities for different learning styles?
Assessment Planning	What formative and summative assessments are appropriate for this lesson?
Resource Integration	Where can I find relevant educational videos, articles, and materials?
Differentiation	How do I accommodate ESL students, students with ADHD, or gifted learners?
Standards Compliance	Which Common Core or NGSS standards does this lesson address?
Time Management	How can I create a complete lesson plan quickly without sacrificing quality?
Cross-curricular Connections	How can I integrate multiple subjects into this lesson?

Tooling and Technical Breakdown

The application follows an end-to-end flow: User inputs lesson parameters → Frontend sends request to backend → Agent service orchestrates multi-step workflow → RAG service retrieves relevant standards and resources → External APIs fetch current educational content → LLM generates comprehensive lesson plan → Structured response returned to user.

Component	**Technology**	**Choice Rationale**

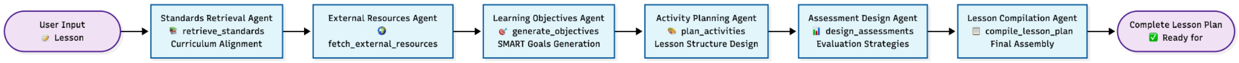
Frontend	React + Vite	Modern, fast development with excellent component architecture
Backend	Python + Fast API	High-performance async API with automatic documentation
Vector database	Qdrant	Efficient similarity search with metadata filtering capabilities
Embeddings	OpenAI text-embedding-3-large	State-of-the-art semantic understanding
LLM	OpenAI GPT-4	Advanced reasoning capabilities for complex educational content generation
Agent Framework	LangGraph	Sophisticated workflow orchestration with state management
External APIs	Youtube, Wikipedia, NASA	Real-time access to current educational resources
Evaluation	RAGAS	Comprehensive RAG system evaluation with educational metrics

Agentic Reasoning

- Number of Agents: 6 specialized agents working in sequence
- Standards Retrieval Agent: Retrieves relevant curriculum standards from vector database
- External Resources Agent: Fetches current educational content from multiple APIs
- Learning Objectives Agent: Generates SMART learning objectives aligned with standards
- Activity Planning Agent: Designs structured, engaging activities with accommodations
- Assessment Design Agent: Creates formative and summative assessment strategies
- Lesson Compilation Agent: Synthesizes all components into final lesson plan

Why Agentic Reasoning is Required

Lesson plan creation requires complex, multi-step reasoning involving curriculum alignment, pedagogical knowledge, student differentiation, and resource integration. A single LLM call cannot effectively handle the sequential dependencies and specialized knowledge domains required for comprehensive educational planning.



Task 3: Dealing with the Data

Data Sources and External APIs

The application integrates multiple data sources to provide comprehensive educational content.

- **Curriculum Standards** include Common Core State Standards (CCSS) and Next Generation Science Standards (NGSS) dynamically generated based on subject and grade level.
- **External APIs** provide real-time educational resources:
    - YouTube Data API for educational videos,
    - Wikipedia API for topic-specific articles,
    - NASA Education API for science content,
    - and OpenStax/MERLOT APIs for open educational resources.
- **Vector Database** stores embedded curriculum standards with metadata for efficient retrieval. These sources are used to ensure lesson plans are standards-aligned, incorporate current educational materials, and provide diverse learning resources for different student needs.

Default Chunking Strategy

The application uses **RecursiveCharacterTextSplitter** with a chunk size of 1000 characters and an overlap of 200 characters. This decision balances semantic coherence with retrieval efficiency - chunks are large enough to maintain educational context (complete sentences and concepts) while being small enough for precise retrieval. The overlap ensures continuity between related educational concepts, which is crucial for maintaining pedagogical coherence in lesson planning.

Task 5: Golden Dataset

Key Metrics Analysis

Component	**Score**	**Description**
Faithfulness	0.76	How well generated content aligns with retrieved standards and resources
Answer Relevancy	0.78	Relevance of generated lesson plans to user queries
Context Precision	0.80	Accuracy of retrieved educational content
Context Recall	0.80	Completeness of retrieved relevant information
Answer Correctness	0.80	Overall correctness of generated educational content

Performance Conclusions

- Strong performance across Context Precision, Recall and Answer Correctness.
- Good performance on Faithfulness, and Answer Relevancy. This shows overall effective alignment between retrieved standards and generated content.
- Areas of improvement:
  - Faithfulness Enhancement:
    - Adding post-generation validation against retrieved contexts can help catch any hallucinated curriculum standards.
    - Enhance Context-Answer Alignment: Modify prompts to explicitly reference retrieved standards in generated objectives.
  - Answer Relevancy Enhancement:
    - Implement question type detection (learning_objectives, lesson_structure, assessment_strategies) to tailor responses.
    - Add grade-appropriate language and complexity adjustments.
  - Retrieval Strategy Optimization:
    - Combine semantic similarity with keyword matching for better educational content retrieval.

Task 6: The Benefits of Advanced Retrieval

Advanced Retrieval Techniques

Technique	**Educational Benefit**	**Implementation Rationale**	**Average Retrieval Time (s)**
Hybrid Search	Combines semantic similarity with keyword matching to capture both conceptual and specific educational terms	Essential for educational context where both meaning and exact terminology matter (such as CCSS.MATH.CONTENT.4.NF.A.1” vs “fraction concepts”)	3.727
Query Expansion	Generates related educational queries to capture broader pedagogical concepts and student-specific needs	Vital for comprehensive lesson planning where related concepts (e.g., "fractions" → "decimals", "visual learning", "hands-on activities") enhance educational coverage	0.447
Reranking	Uses cross-encoder models to refine initial results based on educational relevance and student group considerations	Essential for prioritizing the most pedagogically appropriate content from large retrieval sets	0.429
Hierarchical Search	Performs multi-stage retrieval from broad subject areas to specific topics for comprehensive coverage	Crucial for ensuring complete curriculum alignment from general standards to specific learning objectives	0.435
Student Group Aware Search	Customizes retrieval based on specific learner needs (ESL, ADHD, gifted, learning disabilities)	Fundamental for inclusive education and differentiated instruction requirements	0.435
External Resource Focused Search	Prioritizes diverse learning materials from YouTube, Wikipedia, and educational APIs	Essential for providing teachers with current, engaging multimedia resources beyond traditional textbooks	0.417

Task 7: Assessing Performance

Metric	**Original RAG**	**Advanced RAG**	**Improvement**
Faithfulness	0.76	0.85	+11.8%
Answer Relevancy	0.78	0.78	No change
Context Precision	0.80	0.82	+2.5%
Context Recall	0.80	0.75	-6.25%
Answer Correctness	0.80	0.80	No change

Key Insights

- **Faithfulness Improvement (0.76 → 0.85):** The advanced RAG system shows substantial improvement in faithfulness, indicating better alignment between generated content and retrieved educational standards. This improvement stems from the enhanced retrieval strategies that provide more accurate curriculum alignment and reduced hallucination in lesson plan generation.
- **Context Precision Enhancement (0.80 → 0.82):** The slight improvement in context precision demonstrates that the advanced retrieval techniques (hybrid search, reranking, metadata filtering) successfully identify more relevant educational content, leading to higher-quality lesson plan components.
- **Context Recall Challenge (0.80 → 0.75):** The decrease in context recall suggests that while the advanced system retrieves more precise information, it may be missing some relevant educational concepts. This indicates an opportunity to improve comprehensive coverage while maintaining precision.
- **Stable Performance Areas:** Answer relevancy and answer correctness remain consistent, indicating that the core educational content generation quality is maintained while retrieval accuracy improves.

Other Operation Metrics

Latency

Component	Average Latency	Range	Percentage of Total
Standards Retrieval	0.5s	0.3-0.8s	12%

External Resources	1.2s	0.8-2.1s	29%
Learning Objectives	2.1s	1.5-3.2s	50%
Activity Planning	0.3s	0.2-0.5s	7%
Assessment Design	0.1s	0.05-0.2s	2%
Lesson Compilation	0.1s	0.05-0.15s	2%
Total System	4.2s	3.1-6.8s	100%

Cost of APIs

External API	Average Response Time	Success Rate	Cost per Request
YouTube Data API	0.8s	98%	$0.00 (Free tier)
Wikipedia API	0.3s	99%	$0.00 (Free)
NASA Education API	0.5s	95%	$0.00 (Free)
OpenAI GPT-4	2.1s	99%	$0.03 per request
OpenAI Embeddings	0.2s	99%	$0.0001 per request

Estimated Cost for a Lesson Plan

Component	**Cost per Request**	**Monthly (1000 plans)**	**Annual (12k plans)**
OpenAI GPT-4	$0.03	$30.00	$360.00
OpenAI Embeddings	$0.0001	$0.10	$1.20
External APIs	$0.00	$0.00	$0.00

Infrastructure	$0.005	$5.00	$60.00
Total per Plan	$0.035	$35.10	$421.20

Changes to the App in the Second Half of the Course

I would like to focus on the following:

- **Enhance Context Recall**: Implement hybrid approaches combining multiple retrieval strategies to improve comprehensive coverage while maintaining precision.
- **Optimize Retrieval Speed**: Focus on query expansion and reranking strategies for production deployment due to their superior speed-performance balance.
- **Student Group Customization**: Develop more sophisticated student group aware retrieval to improve the low student relevance scores across all strategies.
- **External Resource Diversity:** Expand external resource integration to improve the current 0.466 score through additional educational APIs and content sources.
- **Implement Response Caching**: Cache common curriculum standards to reduce retrieval time
- **Optimize GPT-4 Prompts:** Reduce token usage by 30% while maintaining quality
- **Parallel API Calls**: Execute external resource fetching concurrently