CSI-410 Project: Designing a Learning Experience with LLMs

Standardized Test Prep Tutor

LSAT • SAT • MCAT

Part 1: LLM Powered Learning Experience Ideas

1. **Sports Statistics Analyst & Coach**
   Who: Those who want to grow their understanding of a certain sport.
   Problem: Understanding game strategy, play breakdowns, and real-time statistics during games.
   Why an LLM: An LLM can give people someone to yell at when they get mad at their team. It can also provide explanations to people who might not fully understand part of the game. If fed visual data of games it can provide coaching like chess.com's mistake counter.

2. **Video Game Gameplay Coach**
   Who: Gamers trying to learn or improve at competitive video games.
   Problem: Most people playing games are not good at them.
   Why an LLM: There is a lot of data that can be fed into an LLM during a game where it can give insight into what a player should improve upon, informed from pro-player statistics.

3. **Verified News Aggregator**
   Who: General public and students trying to stay informed.
   Problem: News is confusing; sources are hard to locate and verify, and misinformation is everywhere.
   Why an LLM: An LLM can cross-reference claims across multiple outlets, surface primary sources backing up (or contradicting) what reporters say, and present this information in a digestible way.

4. **Language Tutor (Chat & Voice)**
   Who: Language learners at any level.
   Problem: Practicing conversational skills and getting real-time grammar corrections without a human tutor.
   Why an LLM: An LLM can hold natural conversations in the desired language, correct grammar and spelling, outline rules and sentence structure, and give real time feedback.

5. **Astronomy & Astrology Tutor**

Who: Amateur astronomers and curious learners.
Problem: Identifying stars, constellations, and understanding astrological or astronomical calculations.
Why an LLM: An LLM can take user descriptions of the night sky or coordinates and help identify constellations and planets, explain the science, and explain the mathematics behind the field.

6. **Adaptive Test Prep Tutor (LSAT / SAT / MCAT)**
Who: Students preparing for major standardized tests.
Problem: Understanding why wrong answers are wrong and building the reasoning skills each test demands.
Why an LLM: An LLM can break down individual questions by type, analyze a student's specific logical errors, explain correct reasoning, and adapt its teaching to each student's weak points.

7. **Personal Gym Trainer**
Who: People new to fitness or looking to improve their routine.
Problem: Designing effective workout plans without the cost of a personal trainer.
Why an LLM: Use a parasocial relationship for good! An LLM coach can encourage people to hit their goals while explaining the science behind exercise and laying out the math to find calorie goals.

8. **Document Reformatter**
Who: Students and professionals who need to reformat written work.
Problem: Converting documents between formats such as APA, MLA, Chicago.
Why an LLM: An LLM can quickly reformat documents between citation styles.

9. **Essay Refiner**
Who: College students writing research papers and essays.
Problem: Improving arguments and prose.
Why an LLM: An LLM can give detailed suggestions while providing examples of other authors to exemplify the voice of a piece.

10. **Tech Support**
Who: Elderly users struggling with technology.
Problem: Old people don't know how to use tech.
Why an LLM: An LLM combined with a screen-watching tool like Windows Recall can help the elderly with many tech issues as well as finding settings to improve quality of life.

Part 2: Prioritization and Selection

Selected Idea: Adaptive Standardized Test Prep Tutor

We selected Idea #6—the Adaptive Test Prep Tutor covering the LSAT, SAT, and MCAT—as our project.

Why This Idea Over the Others

This idea is deeply personal: team members are actively preparing for the LSAT, which means we understand the pain of test prep firsthand. Standardized tests are usually stressful and expensive, with students often spending thousands of dollars on prep courses while still missing why an answer they gave was wrong. An answer key will tell you the correct answer, but it can't discuss where your thinking was flawed if you got the question wrong. An LLM can fill this gap because it can read a stimulus, understand the question type, analyze your specific answer choice, and explain the flaw in your reasoning.

Tradeoffs (Scope, Risk, Ambition)

Rather than building a narrow tool for just one exam, we broadened our scope to cover three major standardized tests: the LSAT, SAT, and MCAT. This this is ambitios as each test has distinct questions reasoning style. We mitigate this through our configurable prompt layer: the system prompt defines the tutor's behavior and approach. The primary tradeoff lies in depth versus breadth. Our learning experience, while covering three separate tests, may not be as comprehensive as one designed around a singular test.

Learning Value Prioritized

We prioritize cognitive development by helping the learner understand *how* they think about problems, instead of whether they got the right answer or not. The tutor is designed to identify the question type, explain why an answer is correct, why the others are not, and break down the student's thinking process. By dissecting the logical structure of the stimulus and explaining why each wrong answer fails, it helps students recognize similar problems in the future. This builds confident learners all for a fraction of the cost other tutors would charge.

Part 3: Core LLM Experience

How the Experience Works

The application is a Flask-based web app backed by the Anthropic API (Claude Sonnet). The student first selects which test they are studying for (LSAT, SAT, or MCAT). They are then presented with a test-specific form: for the LSAT, they enter the stimulus, question and answer choices, their selected answer, the correct answer, their rationale, and a self-diagnosis of why they got it wrong; for the SAT, they enter the math problem, answer

choices, their selection, the correct answer, their work, and where they got stuck; for the MCAT, they enter the passage, question and answers, their selection, the correct answer, their reasoning, and passage comprehension notes. Once submitted, the system concatenates the base system prompt with the appropriate test-specific configuration and sends the filled-in question template as the user message to Claude. The tutor then guides the student through a structured wrong-answer analysis, not by immediately giving the right answer, but by helping them discover *why* their reasoning failed. The conversation is maintained across turns via a conversation history array, allowing follow-up dialogue. This follows the Socratic Method: the tutor asks clarifying questions, names error patterns, and pushes the student to identify the flaw in their own thinking before confirming or correcting.

System Prompt

This prompt defines the tutor's core behavior, tone, and approach. It remains constant regardless of which test is being studied.

SYSTEM PROMPT

```
You are a wrong-answer analysis/post-test review tutor who helps learners
understand their missed questions on standardized tests.  Your goal is to
help them understand WHY they got the question wrong.

Your overall approach:
1.  Help learners discover for themselves WHY they got the question wrong,
rather than initially giving them the answer.
2.  Guide them to identify the points of failure in their thinking, not just
find the right answer.
3.  Focus on the process.
4.  Your initial focus should be to guide them, not just give them the
correct answer, but you can provide the correct answer if the learner asks
for an explanation after your initial analysis.
5.  Give concise answers without rambling, but be thorough in your analysis.
6.  Only your first message has to be highly structured.  After that, you can
be more flexible in your responses based on the learner's needs and the flow
of the conversation.

Adapt your questioning to the specific test type (LSAT, SAT, MCAT) and the
common error patterns associated with that test.  Use the test-specific
context provided in the user prompt to inform your analysis.

If the learner reaches a correct understanding, affirm/or disconfirm their
insight and then ask if they want to try another question or end the session.
Respond to questions, but do not go on endlessly with probing questions.
```

Configurable Prompt Layer

These prompts are swapped in depending on which test the student selects. They take test specific question , domain knowledge, and scoring context into the tutor's behavior. Below are all three configurations.

Configuration: LSAT

```
Test Context: You are analyzing LSAT Logical Reasoning questions.

Response Structure:
- Question Type: Identify the specific question type and modifier (e.g.,
''Strengthen'', ''Weaken'', ''Must-Except'', ''Parallel-Principle'', etc.)
- Question Difficulty: Subjectively assess the difficulty level of the
question (1--5 Stars)
- Initial Observation: What you notice about their reasoning approach. This
isn't an opportunity to congratulate them on what they did right, but rather
to point out something about their thought process that is relevant to the
error they made.
- Key Question: ONE clarifying question about their thought process---if it
is relevant. If the error is simply one of reading comprehension, then you
can skip this step.
- Pattern Alert: If you spot a common error pattern, name it constructively
- Adjustment: Concrete suggestion for what to examine or reframe that could
help them see the error in their thinking. This should be actionable and
specific to the question at hand, not just a general tip.

Common LSAT LR error patterns to watch for:
- Confusing necessity vs. sufficiency in conditional logic
- Missing scope shifts between stimulus and answer choices
- Falling for strength of assertions (extreme language, absolutes)
- Misidentifying the conclusion or premises in arguments
- Intermediate conclusions mistaken for the main conclusion
- Overlooking conditional logic reversals (if A then B ≠ if B then A)
- Shell game answer choices that repeat stimulus language but change meaning
- Opposite answer choices that are factually true but don't answer the
question
- Confusing strengthen/weaken with assumption questions

LSAT-specific guidance:
- The stimulus contains an argument with premises and a conclusion
- Question stems are precise---''most vulnerable to criticism'' is different
from ''assumption''
- Wrong answers often contain elements from the stimulus but shift scope
- Emphasis on prephrasing the question in their own words to ensure they
understand what is being asked
- Correct answers must be 100% defensible from the text

Question Template:
```

```
Analyze this LSAT LR wrong answer:
STIMULUS: {stimulus} | QUESTION & ANSWERS: {question} | USER SELECTED:
{selectedAnswer} | CORRECT ANSWER: {correctAnswer} | USER'S RATIONALE:
{rationale} | USER'S SELF-DIAGNOSIS: {whyWrong}
```

Configuration: SAT

```
Test Context: You are analyzing SAT Math questions.

Common SAT Math error patterns to watch for:
- Sign errors and negative number mistakes in calculations
- Misreading what the question actually asks for (finding x vs. 2x)
- Correct calculation but wrong unit or form (decimal vs. fraction)
- Algebraic manipulation errors (distribution, combining like terms)
- Rushing through ''easy'' problems and making careless errors
- Setting up equations correctly but solving incorrectly

SAT Math-specific guidance:
- Distinguish between conceptual misunderstandings vs. computational errors
- Check if they understood the problem setup or made an execution error
- Look for whether they knew the method but made a mistake in applying it
- Consider time pressure---did they skip steps?

Question Template:
Analyze this SAT Math wrong answer:
PROBLEM: {problem} | ANSWER CHOICES: {choices} | STUDENT SELECTED:
{selectedAnswer} | CORRECT ANSWER: {correctAnswer} | STUDENT'S WORK: {work} |
WHERE STUDENT GOT STUCK: {stuck}
```

Configuration: MCAT

```
Test Context: You are analyzing MCAT CARS (Critical Analysis and Reasoning)
questions.

Common MCAT CARS error patterns to watch for:
- Bringing in outside knowledge instead of pure passage-based reasoning
- Missing author's tone, stance, or attitude toward the subject
- Confusing ''could be true'' with ''must be true'' based on passage
- Selecting answers that are factually true but don't answer the question
- Missing subtle qualifiers (some, most, all, typically, usually)
- Making inferences beyond what the passage supports

MCAT CARS-specific guidance:
- Everything must be defensible from the passage text
- Author's purpose and tone matter as much as content
- Questions test reading precision, not background knowledge
```

```
- The correct answer often hinges on a single word or qualifier

Question Template:
Analyze this MCAT CARS wrong answer:
PASSAGE: {passage} | QUESTION & ANSWERS: {question} | STUDENT SELECTED:
{selectedAnswer} | CORRECT ANSWER: {correctAnswer} | STUDENT'S REASONING:
{reasoning} | PASSAGE COMPREHENSION NOTES: {notes}
```

Prompt Architecture: How the Layers Work Together

The system prompt and configurable prompt layer are concatenated at runtime by the `create_full_prompt()` function in `prompt_config.py`. The base `SYSTEM_PROMPT` string is always loaded first and establishes the tutor's methods: Our project focused on process over answers. The test-specific `test_context` from the `TEST_CONFIGS` dictionary is then appended, injecting the relevant error patterns, guidance, and response structure. The resulting combined string is passed as the `system` parameter to the Anthropic API. Meanwhile, each test configuration also contains a `question_template` with placeholder fields that get filled from the student's form submission and sent as the `user` message. The Flask backend (`app.py`) manages conversation history as an array of message objects, enabling multi-turn follow-up dialogue within a single session. This separation means that adding support for a new test (e.g., GRE, ACT, bar exam) requires only adding a new entry to `TEST_CONFIGS`—the system prompt, API integration, and frontend remain unchanged. This is the core architectural insight: pedagogical behavior is decoupled from domain knowledge.

Student Roles and AI Usage

The majority of the work on this assignment was split by working together in the same space. There is not a clear delineation on who did what in our work since we both typed on the same methods on the same computer. This was also a relatively simple program on the backend so there was little need to go back on our own and individually to develop/refactor.

The work with clear delineation was as follows:

- **Noah**: Wrote the bulk of this .tex file documenting the project and development, with only minor edits from Akiva. (hi! -Akiva)
- **Akiva**: Implemented some front-end changes separately, since the v0 account that generated the front end was on his laptop. Specifically, modified the UI to function more like a chatbot rather than a simple one-input, one-output forum response.

AI Assistance

As suggested and demonstrated by our professor, we used a generative AI tool (specifically `v0` by Vercel) to develop the front end of our website. Due to limited experience with the libraries suggested by `v0` and with TypeScript, we also made limited use of VS Code's IDE auto-completion features. Additionally, since the configurable-multiple-test design was

secondary to the application of the LSAT sections, the specific prompt-directions for those were generated generically from the LSAT ones written by us.

Otherwise, the overall design and configuration of the backend were completed manually.