

SELF LEARNING AI TUTOR SOFTWARE QUALITY PLAN

3/21/2026

OVERVIEW

1. Project Description

The Self-Learning AI Tutor is an interactive, web-based educational system designed to help students improve their understanding of mathematical concepts through adaptive, personalized instruction. The project was developed as part of a collaborative software engineering effort at Arizona State University, with a team of student developers working in an Agile environment using iterative sprints.

This project is sponsored by Dr. John Leddo and MyEdMaster LLC, an organization focused on improving student learning outcomes through innovative educational tools and personalized tutoring solutions. The partnership provides real-world context, domain expertise, and guidance in designing an effective intelligent tutoring system.

The motivation for this project stems from the limitations of traditional learning systems, which often provide static content and do not adapt to individual student needs. Many learners struggle not because they lack effort, but because they lack targeted feedback on how they are reasoning through problems. The Self-Learning AI Tutor addresses this gap by analyzing student responses, identifying patterns in their reasoning, and delivering personalized feedback, hints, and targeted practice.

The system is built using Python, Streamlit, MongoDB, and LLM-based analysis, and includes features such as:

- Step-by-step reasoning evaluation
- Adaptive hint generation
- Weak-area detection and targeted practice modes
- Persistent tracking of student progress and learning patterns

The primary stakeholders include:

- Student users seeking personalized learning support
- The development team responsible for building and maintaining the system
- Course instructors and evaluators assessing the effectiveness of the platform
- The project sponsor, MyEdMaster LLC, providing domain expertise and application context

The overall purpose of this project is to create an intelligent tutoring system that improves learning outcomes by focusing not only on correct answers, but also on the process of reasoning, enabling students to develop deeper conceptual understanding.

2. High-Level Requirements

The Self-Learning AI Tutor must meet a set of functional and quality requirements to ensure it provides effective, reliable, and meaningful educational support. The following high-level requirements define the expected behavior and performance of the system:

1. The system shall evaluate student responses and determine correctness with at least 95% accuracy compared to predefined solutions.
2. The system shall generate hints that reference at least one element of the student's submitted reasoning or answer.
3. The system shall store student progress data, including past responses and reasoning steps, in a persistent database.
4. The system shall retrieve stored student progress data from the database when requested.
5. The system shall identify at least one weak area per student based on analysis of incorrect or incomplete responses across prior interactions.
6. The system shall generate targeted practice problems aligned with identified weak areas or user-selected topics.
7. The system shall provide step-by-step feedback explaining why a student's answer is correct or incorrect.
8. The system shall allow users to complete learning activities in a non-linear order without restriction.
9. The system shall respond to user input within 2 seconds under normal operating conditions.
10. The system shall support at least 20 concurrent users without system failure or response time exceeding 3 seconds.
11. The system shall maintain data integrity such that no stored student progress is lost during normal operation.

3. Quality Metrics

The following quality metrics define how the Self-Learning AI Tutoring system's performance and functionality will be measured against the high-level requirements. Each metric is designed to quantitatively evaluate whether the system meets its intended behavior, with clearly defined acceptance targets. These metrics are directly traceable to the project's requirements to ensure comprehensive verification of system quality.

Metric Name	Description	Acceptance Level/Target	Related Requirement(s)
Answer Evaluation Accuracy	Percentage of student responses where the system's correctness evaluation matches predefined correct answers.	$\geq 95\%$	1
Hint Reference Rate	Percentage of generated hints that include at least one reference to an element from the student's submitted reasoning or answer.	$\geq 95\%$	2
Data Persistence Success Rate	Percentage of student progress data entries (responses and reasoning steps) successfully stored in the database without loss or failure.	100%	3

Data Integrity Accuracy	Percentage of stored student progress data that exactly matches the original input when retrieved from the database.	100%	3
Data Retrieval Success Rate	This measurement is the percentage of retrieval requests for student progress data that return complete and correct data matching the originally stored values.	100%	4
Weak Area Identification Accuracy	This measurement is the percentage of correctly identified weak areas compared to predefined or instructor-labeled weak areas based on student performance data.	≥ 85%	5
Targeted Practice Alignment Rate	This measurement is the percentage of generated practice problems that correctly align with the identified weak area or user-selected topic.	≥ 90%	6
Feedback Explanation Completeness Rate	Percentage of system-generated feedback messages that include a step-by-step explanation describing why the student's answer is correct or incorrect.	≥ 95%	7
Non-Linear Navigation Success Rate	Percentage of attempts where users successfully access and complete learning activities in any order without encountering navigation restrictions or blocked paths.	100%	8
System Response Time (Normal Load)	Percentage of system responses delivered within 2 seconds under normal operating conditions (≤ 20 concurrent users).	≥ 95% of responses within 2 seconds	9
Concurrent User Support Rate	Percentage of system operations completed successfully while handling at least 20 concurrent users, without system failure or response time exceeding 3 seconds	≥ 95%	10
Data Loss Prevention Rate	Percentage of stored student progress data that remains preserved without loss during normal system operation, including across multiple sessions and concurrent usage.	100%	11

4. Testing Plan

The testing plan outlines the strategies and procedures used to verify that each quality metric is satisfied. Each test is designed to directly measure its associated metric and ensure alignment with the corresponding requirement. The testing approach includes the use of predefined datasets, system-level evaluations, and functional validation to ensure the system performs reliably under expected conditions.

Accuracy Validation Test

Applies to: Answer Evaluation Accuracy (Req 1)

Test Description

A representative dataset of student responses with known expected correctness outcomes will be run through the system. The dataset will include a range of cases, such as correct responses, incorrect responses, and edge cases (e.g., partially correct or ambiguous answers). The system's evaluations will be compared to the predefined expected results, and the percentage of matching evaluations will be calculated.

Rationale

This test directly measures whether the system meets the requirement of evaluating correctness given varying inputs with at least 95% accuracy. It is well suited because it compares actual system output against known expected outcomes. Additionally, including a range of response types (correct, incorrect, and partial) ensures the system is evaluated under realistic usage conditions.

Hint Reference Validation Test

Applies to: Hint Reference Rate (Req 2)

Test Description

A representative dataset of student responses will be used to generate hints from the system. The dataset will include a range of input types, such as correct reasoning, incorrect reasoning, incomplete responses, brief answers, and invalid or nonsensical responses. For each case, the generated hint will be reviewed to determine whether it references at least one element of the student's submitted reasoning or answer. The percentage of hints meeting this condition will be calculated.

Rationale

This test directly measures whether the system satisfies the requirement that hints reference the student's submitted reasoning or answer. It is well suited because it evaluates the generated hint output against the student input that produced it. Using a varied dataset also helps verify that the requirement is met across realistic response types rather than only simple cases.

Data Persistence Validation Test

Applies to: Hint Reference Rate (Req 3)

Test Description

A representative set of student progress records, including past responses and reasoning steps, will be submitted to the system for storage. The dataset will include typical entries as well as edge cases such as incomplete responses, multi-step reasoning, and repeated submissions. After each submission, the database will be checked to confirm that the record was successfully created and stored without failure. The percentage of records successfully persisted will be calculated.

Rationale

This test directly measures whether the system satisfies the requirement to store student progress data in a persistent database. It is well suited because it verifies successful storage across a variety of realistic input cases rather than only simple submissions. Including multiple record types also helps demonstrate that persistence remains reliable under normal usage conditions.

Data Integrity Verification Test

Applies to: Data Integrity Accuracy (Req 3)

Test Description

A representative set of student progress records will be stored in the database and then retrieved. The retrieved data will be compared against the original submitted records to verify that fields such as past responses and reasoning steps were preserved exactly as entered. The percentage of retrieved records that match the original input with no missing or altered data will be calculated.

Rationale

This test directly measures whether stored student progress data maintains its integrity after persistence and retrieval. It is well suited because it compares the original input against the stored output, which provides a clear way to verify that no data was lost, modified, or corrupted during storage. This supports the requirement by confirming not only that data is saved, but that it remains accurate in the persistent database.

Data Retrieval Validation Test

Applies to: Data Retrieval Success Rate (Req 4)

Test Description

A dataset of student progress records will be inserted into the database, including responses, reasoning steps, and associated metadata. Retrieval requests will then be executed for each stored record under normal operating conditions. The retrieved data will be compared against the original stored data to verify completeness and correctness. This process will be repeated across multiple records, users, and sessions to ensure consistency. Any mismatches or missing data will be recorded. The percentage of successful retrievals that exactly match the stored data will be calculated.

Rationale

This test directly verifies that the system can reliably retrieve stored student data without loss, corruption, or inconsistency. Accurate data retrieval is critical for maintaining continuity in personalized learning experiences. By comparing retrieved data against known stored values across multiple scenarios, this test provides an objective, repeatable, and measurable validation of system correctness and reliability.

Weak Area Identification Validation Test

Applies to: Weak Area Identification Accuracy (Req 5)

Test Description

A dataset of student interactions will be used in which weak areas have been predefined by instructors or labeled based on historical performance. The system will process this dataset and generate identified weak areas for each student. The identified weak areas will be compared against the predefined labels, and matches and mismatches will be recorded. The percentage of correctly identified weak areas will be calculated. The dataset will include a variety of cases, including consistently incorrect responses, partially correct responses, and mixed performance patterns, to ensure broad coverage.

Rationale

This test ensures that the system accurately identifies areas where students struggle, which is essential for adaptive learning. Using labeled data allows for objective comparison and quantifiable evaluation. Including diverse response patterns ensures that the system is evaluated under realistic conditions. This approach provides a reliable and measurable way to validate that the system can detect meaningful knowledge gaps.

Targeted Practice Generation Validation Test

Applies to: Targeted Practice Alignment Rate (Req 6)

Test Description

For a predefined set of weak areas and user-selected topics, the system will generate practice problems. Each generated problem will be evaluated against the intended topic or weak area using a validation checklist or expert review criteria. The evaluation will determine whether the generated problem aligns with the expected concept or skill. Results will be recorded for each generated problem, and the percentage of correctly aligned practice problems will be calculated across all test cases.

Rationale

This test verifies that the system generates relevant and appropriate practice problems tailored to student needs. Alignment between generated problems and targeted topics is critical for effective remediation. Using a structured evaluation method ensures consistency and objectivity in measurement. This approach confirms that the system's adaptive functionality is working as intended and provides meaningful, focused practice opportunities.

Feedback Explanation Completeness Test

Applies to: Feedback Explanation Completeness Rate (Req 7)

Test Description

A representative dataset of student responses—including correct, incorrect, partially correct, and incomplete answers—will be submitted to the system. For each response, the system-generated feedback will be reviewed to determine whether it includes a step-by-step explanation describing *why* the answer is correct or incorrect. The percentage of feedback messages meeting this criterion will be calculated.

Rationale

This test directly measures whether the system provides meaningful, explanatory feedback rather than simple correctness labels. Evaluating a diverse set of responses ensures the system consistently produces complete explanations across realistic usage scenarios.

Non-Linear Navigation Validation Test

Applies to: Non-Linear Navigation Success Rate (Req 8)

Test Description

Test users will attempt to access and complete learning activities in various non-linear sequences (e.g., skipping ahead, revisiting earlier modules, jumping between topics). Each navigation attempt will be logged to verify that the system allows unrestricted access without blocking, forced ordering, or navigation errors. The percentage of successful non-linear navigation attempts will be calculated.

Rationale

This test verifies that the system supports flexible learning paths, a core requirement for self-directed learning. Testing multiple navigation patterns ensures the system does not implicitly enforce a linear sequence.

Normal Load Response Time Test

Applies to: System Response Time (Normal Load) (Req 9)

Test Description

The system will be evaluated under normal operating conditions (≤ 20 concurrent users). A scripted set of typical user actions—submitting answers, requesting hints, retrieving progress, and navigating activities—will be executed. Response times will be recorded, and the percentage of responses delivered within 2 seconds will be calculated.

Rationale

This test directly measures whether the system meets the required responsiveness threshold during typical usage. Using realistic user actions ensures the results reflect actual performance rather than synthetic benchmarks.

Concurrent Users / Performance

Applies to: Concurrent User Support / Load Performance (Req 10)

Test Description

A load testing tool will simulate 20 concurrent users interacting with the system. Each simulation will perform submitting responses, requesting hints, navigating between activities, and retrieving stored data. We will record response time, error rate, and system crashes or failures. Most of the requests should be completed within 3 seconds and we will calculate this time as well as any failed requests.

Rationale

This test directly evaluates the system under usage by simulating concurrent users. It ensures the system meets performance expectations and remains stable under an expected load. This will help us identify bottlenecks or failures in scalability.

Data Integrity

Applies to: Data Integrity Accuracy (Req 11)

Test Description

A dataset of user interactions that will include responses, reasoning steps, and metadata will be submitted and stored in the database. After storage, the data will be retrieved and compared with the original input. This dataset will include multiple users and sessions and will have edge cases, repeated submissions, long inputs, and incomplete data. Any issues between stored and retrieved data will be recorded.

Rationale

This test ensures that the system will reliably preserve user data without loss or corruption. This process is critical for maintaining accurate learning progress. We will compare original input with retrieved data, providing a direct and measurable way to verify integrity.