ASHESI
UNIVERSITY

Josephine Allan

Final Project

Cohort A: Web Technologies

David Sampah

Hutton

18th December 2025

CS3 Quiz Platform: Master Technical Documentation

1. Platform Architecture & Overview

The CS3 Quiz Platform is an intelligent educational ecosystem designed for third-year Computer Science students. It helps computer science to learn more about their course through quizzes specially generated for courses a typical third year computer science takes.

Core Objectives

- Using Large Language Models (LLMs) to generate questions specifically from the CS textbooks of the 11 core courses.
- Providing more than just "right or wrong" answers by utilizing semantic similarity for partial credit. This is done by comparing the model answer with the user’s answer.
- Adaptive Learning: Allowing students to target specific chapters and topics for focused study.

System Components & File Map

- Root and Entry Components: The index.php and README.md files serve as the primary entry points for the application, providing the initial landing page and a comprehensive technical overview of the project.
- Core User Interface: The main user journey—consisting of course selection, quiz execution, and result review—is managed through dashboard.php, quiz.php, and results.php.
- Logic and Authentication: Shared utilities, session security, and common UI elements like navigation and branding are handled by functions.php, header.php, and footer.php.

- AI Orchestration Engine: The system’s intelligence, including the Natural Language Processing (NLP) grading logic and API connectivity health checks, is powered by similarity-functions.php and test-gemini-api.php.
- Data and Course Logic: Definitions for the 11 core computer science courses, their specific chapters, and the underlying database connection strings are maintained in courses.php, topics.php, and database.php.
- Administrative and Development Tools: For system maintenance and setup, run-migration.php and diagnose-api.php provide essential tools for database schema updates and real-time API monitoring.

AI Orchestration: Gemini 2.5 Flash

The platform's "intelligence" is powered by the Gemini 2.5 Flash model.

A. Quiz Generation

When a student selects a topic, the system constructs a highly specific "system prompt" sent via the api/generate-quiz.php endpoint.

1. The prompt includes the textbook name and specific chapter.
2. The AI is instructed to return a structured JSON object containing questions, options, and detailed explanations.
3. The system validates the JSON schema before presenting it to the user.

B. The "Question Bank" Optimization

To maintain high performance and low API costs, the system uses question-bank-functions.php.

- Caching: New AI-generated questions are saved to the question_bank table.

- Deduplication: The system tracks user_question_history to ensure students don't see the same question twice, pulling from the bank first before requesting new ones from the AI.

The Grading System

A standout feature is the platform’s ability to grade "Calculations" and "Essays" using Natural Language Processing (NLP).

Semantic Similarity Algorithm

Located in similarity-functions.php, the engine uses a weighted hybrid approach:

- Word Overlap (50%): Identifies key technical terms.
- Keyword Matching (30%): Ensures specific required concepts are present.
- Distance (20%): Analyzes the structural similarity of the response.

Grading Requirements

- 80% - 100%: Full Credit (Expert Understanding)
- 50% - 79%: Partial Credit (Proficient Understanding)
- 20% - 49%: Minimal Credit (Developing Understanding)
- < 20%: No Credit (Conceptual Gap Identified)

Database Management

The system relies on a relational MySQL schema designed for scale and hosting flexibility.

- database.sql: The master schema including database VIEWs for real-time analytics.
- database-import-infinityfree.sql: A compatibility version specifically for shared hosting environments (like InfinityFree) that lack VIEW permissions.

- run-migration.php: A script that automates the transition from version 1.0 to 2.1, ensuring the local question_bank is correctly initialized.

Setup & Maintenance Diagnostics

For developers and administrators, the platform includes standalone health-check scripts:

- test-gemini-api.php: Directly pings Google's servers to verify the API key and model availability.
- diagnose-api.php: Checks server-side requirements like the PHP cURL extension and SSL certificate status.
- test-question-bank.php: Verifies that the database can correctly store and retrieve JSON-formatted question data.