Q-View: Academic Quality Assurance Dashboard

Documentation & User Guide

Release Date: February 2026

Platform: Universal (HTML5/JavaScript) - Client-Side Only

Architecture: Single-Page Application (SPA) / Embedded Widget Compatible

Institution-specific tool for Gulf Medical University (GMU)

1. Introduction

Q-View is a specialized academic quality assurance tool designed for Course Coordinators, QA Officers, and Program Directors. It serves as an automated "Course Portfolio Analyzer," bridging the gap between quantitative performance data (grades) and qualitative student feedback (surveys).

Q-View focuses on Course Action Planning. It uses a hybrid parsing engine to read raw course reports (PDFs) generated by GMU systems, and grade sheets (Excel/CSV), visualize performance gaps, and streamline the creation of improvement plans without requiring server-side processing or complex software installation.

Key Features

- **Automated Parsing Engine:** Automatically extracts course codes, satisfaction rates, and free-text comments from standard PDF reports.
- **Qualitative Analysis Workflow:** Features an interactive tagging system to filter student suggestions ("Valid", "Invalid", "Repeated", "Omit"), transforming raw feedback into actionable insights.
- **Smart Action Planning:** Automatically promotes "Valid" suggestions into a formal Course Action Plan, allowing faculty to assign responsibility and actions immediately.
- **Privacy-First:** Runs 100% in the browser. No student data or exam files are ever uploaded to a cloud server.

2. System Requirements

- **Device:** PC, Mac, or Tablet with a modern web browser.
- **Browser:** Google Chrome (Recommended), Microsoft Edge, Safari, or Firefox.

- **Integration:** Fully compatible with LMS platforms and website builders (e.g., Wix, SharePoint) via iframe embedding**.**
- **Internet:** Required only for initial loading of libraries (Tailwind, PDF.js, SheetJS). Once loaded, the app functions offline.

3. Input Data Standards

Q-View relies on specific file structures to function correctly.

A. Course Report Files (PDF)

- **Format:** Standard PDF documents containing course feedback.
- **Required Sections:** The parser looks for specific headers:
  - "Course Code:" / "Course Name:"
  - "Response Rate:"
  - "What Students Like"
  - "Suggestions" (or "Suggestions for Improvement")
- **Layout Support**: Capable of reading both single-column and dual-column layouts (detecting side-by-side "Likes" and "Suggestions").

B. Grade Data Files (CSV / XLSX)

- **Format:** Excel (.xlsx, .xls) or Comma Separated Values (.csv).
- **Data Logic:**
  - **Letter Grades:** The system prioritizes a column containing letter grades (A, B+, C, F, etc.) to generate the Grade Distribution Histogram.
  - **Numeric Scores:** If letter grades are absent, it falls back to the "Total" or "Final" score column to calculate distributions based on standard intervals (90+, 80-89, etc.).

4. User Manual: Curriculum Mapper Module

This module handles the initialization of the session.

4.1. Metadata Entry

Upon launching, the user enters the context for the current audit:

- **College / Program:** (e.g., College of Pharmacy / BPharm)

- **Academic Year / Cohort:** (e.g., 2024-2025 / Semester 1)

4.2. Batch Upload (Smart Matching)

- **Action:** Click the "Select Course Data Folder" area.
- **Behavior:** Users can select **all** files (PDFs and Excels) from a folder simultaneously.
- **Logic:** The system analyzes filenames (e.g., "DP 223.pdf" and "DP 223_Grades.csv"). It automatically matches the PDF report with its corresponding Grade sheet based on the course code extracted from the filename.

5. User Manual: Analysis & Action Planning

This module is the core workspace for the QA audit.

5.1. Summary Dashboard

Provides a high-level view of the cohort's performance.

- **Cohort Overview:** Displays total courses analyzed and the aggregate Average Satisfaction rate.
- **Performance Matrix:** A tabular view of all courses, allowing a quick scan of Pass Rates vs. Satisfaction Rates to identify outliers (e.g., High Grades but Low Satisfaction).

5.2. Course View (The Audit Interface)

Clicking a course tab opens the detailed audit view.

A. Quantitative Visualizations

- Grade Distribution (Histogram): Visualizes student performance. It prioritizes Letter Grade counts (A, B, C...) for accuracy but will fallback to numeric ranges if needed.
- Feedback Analysis: A bar chart visualizing the satisfaction scores for standard survey questions (Q1–Q8), categorized by "Theory" or "Practical" question sets.

B. Qualitative Analysis (The Tagging Engine) The system extracts comments from the "Suggestions" section of the PDF.

- **Garbage Filtering:** Automatically removes noise (dots, "N/A", "Nil", footer text).
- **Interactive Dropdown**: Users must tag each suggestion:

- Valid: A legitimate issue. Action: Automatically creates an entry in the "Course Action Plan."
- Invalid: Not actionable or factually incorrect.
- Repeated: Duplicate of a previous point.
- Irrelevant: Not related to the course context.
- Omit: Removes the comment entirely from the view and the final report.

C. Course Action Plan

- **Auto-Population**: Issues tagged as "Valid" appear here automatically.
- **Manual Entry:** Users can define the specific "Action" to be taken and assign "Responsibility" (e.g., Course Coordinator, Program Director).
- **Management:** Items can be added or deleted manually.

6. User Manual: Reporting & Persistence

This module handles data output and saving.

6.1. Save Session (JSON)

Allows users to pause and resume their work.

- **Format:** Generates a .json file containing all parsed data, tags, and action plans.
- **Compatibility:** because some browswers block the direct download, the application has a fallback strategy to allow copying the generated json text and saving in a notepad or textedit file as a *.json file.

6.2. Print Report (PDF Generation)

Generates a formal QA report.

- **Unified Engine:** Creates a consolidated, printable view of the entire cohort or a single course.
- **Content:** Includes the cover page (for full reports), summary matrices, charts, filtered comments (omitted items removed), and the finalized Action Plan.
- **Format:** Optimized for A4 printing with intelligent page breaks.

7. Technical Specifications

- **Language:** HTML5, JavaScript (ES6+).
- **Libraries:**
  - **React 18:** User Interface rendering.
  - **Tailwind CSS:** Responsive styling and design system.
  - **Recharts:** Data visualization (SVG-based charts).
  - **Babel Standalone:** In-browser JSX compilation.
- **Concept & Logic:** Dr. Muhammad AlShorbagy, Dean, College of Pharmacy, GMU.
- **Technical Implementation:** AI-Assisted Development.
- **Privacy Model:** Client-Side Sandbox. The application operates entirely within the user's browser memory (RAM). Refreshing the page wipes all data, ensuring compliance with strict academic data privacy regulations.