QAMatrix: The College QA Allocation System

Documentation & User Guide

Release Date: December 2025

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

Institution-specific tool for Gulf Medical University (GMU)

1. Introduction

The **College QA Allocation System** is a single-file, browser-based application designed to automate the assignment of Quality Assurance (QA) reviewers to course files.

It solves the complex "Constraint Satisfaction Problem" of matching faculty to courses while adhering to strict rules:

- **Conflict of Interest:** A Course Coordinator cannot review their own course.
- **Capacity Limits:** Different faculty ranks (Executives, Seniors, TAs) have different review quotas.
- **Load Balancing:** The system attempts to distribute work evenly among eligible reviewers.

2. Workflow Overview

The application follows a logical 4-step workflow, represented by the sidebar tabs:

- **Data Management:** Input your Reviewers and Courses (Manual or CSV).

- **Capacity Planner:** Check if you have enough reviewers and adjust quotas.

- **Matrix:** View the generated assignments.

- **Dashboard**: Visualize workload distribution and statistics.

3. Step-by-Step Guide

Tab 1: Data Management

This is where you populate the system**. Order Matters**: It is recommended to enter the Reviewers Pool first, so the auto-suggest feature works when adding Courses.

A. Reviewers Pool

This list contains everyone available to review a course file.

- **Manual Entry:** Click "+ Add Manually". Enter Name, Department, and Role.

- **CSV Import**: Click "Upload CSV".
    - ✓ **Required Columns:** `Name, Department, Role`.
    - ✓ **Role Keywords:** The system looks for keywords in the Role column to assign rank:
        - ○ "exec" -> Executive/Dean (Default Quota: 1)
        - ○ "senior" -> Senior Faculty (Default Quota: 3)
        - ○ "ta" -> TA/Preceptor (Default Quota: 5)
- **Load Demo:** Click "Load Demo" to populate with sample data for testing.

B. Programs & Courses

This list contains the courses that need to be reviewed.

- Manual Entry: Click "+ Add Manually".

    Coordinator Name: You can type any name here. It does *not* strictly have to be someone in the Reviewers pool (e.g., an external adjunct). However, if the name matches a Reviewer, the system will prevent that Reviewer from being assigned to this course.

- CSV Import: Click "Upload CSV".
    - ○ **Smart Column Detection:** The app automatically scans your CSV headers. It accepts files with 3 or 4 columns.
    - ○ **Standard Headers:** `Code, Title, Program, Coordinator,` OR: `Code, Program, Coordinator`.
    - ○ **Minimal Headers**: Code, Program, Coordinator.

Tab 2: Capacity Planner

This dashboard calculates Supply vs. Demand.

- **Demand:** Total number of courses uploaded.
- **Capacity:** Sum of (Reviewers × Quota per Role).
- **Delta:** The difference. If negative (Red), you do not have enough reviewers.

Actions:

- **Adjust Quotas:** Manually change the quota number in the boxes (e.g., change TAs from 5 to 6).

- **Auto-Balance:** Click this button to let the AI suggest quota increases. It uses a "Bottom-Up" strategy: it tries to increase the load on TAs first, then Seniors, and finally Executives, until demand is met.

- **Run Allocation:** Once the Delta is positive (Green), click this to generate the matrix.

Tab 3: Allocation Matrix

Displays the final results.

- **Grouping:** Use the dropdown at the top to group the list by Reviewer, Role, Program, or Coordinator.

- **Conflict Check:** The system guarantees that the "Assigned Reviewer" is NOT the "Coordinator".

- **Unassigned Rows:** If a course has no reviewer (due to capacity limits), it will be highlighted in Red.

Tab 4: Analytics Dashboard

Visual reports to verify fairness.

- **Reviews by Program**: Shows which programs have the most courses.

- **Workload by Role**: Shows the distribution of work across ranks.

- **Top Reviewer Load**: Identifies who has the heaviest workload.

4. CSV Templates & formats

Reviewers CSV Template

Save as .csv. Headers are case-insensitive.

```
Name, Department, Role

Dr. John Doe, Computer Science, Senior

Jane Smith, Math, TA

Prof. Albus, Administration, Executive
```

Courses CSV Template

Save as .csv.

```
Code, Title, Program, Coordinator OR, Code, Program, Coordinator
CS101, Intro to CS, CS, Dr. John Doe
MATH200, Calculus II, MATH, Jane Smith
```

5. Technical Specifications

- **Architecture:** Single-Page Application (SPA).

- **Language:** HTML5, CSS3, Vanilla JavaScript (ES6).

- **Dependencies:** Chart.js (via CDN) for visualization.

- **Privacy:** No server-side processing. No cookies. Local Storage is not used (data persists only in RAM or via manual JSON save).

- **Browser Compatibility:** Chrome, Edge, Firefox, Safari (Desktop & Mobile).

- **Concept & Logic:** Dr. Muhammad AlShorbagy, Dean, College of Pharmacy, GMU.

- **Technical Implementation:** AI-Assisted Development (Code generation).

- **Methodology:** "This single-file HTML application demonstrates a 'No-Code/Low-Code' development approach. The domain expertise, algorithm logic, and user experience design were provided by Dr. Muhammad AlShorbagy, while the source code was generated via prompt engineering using Large Language Models (LLMs)."