

MCQ System and Progress Tracking Redesign Plan

1. MCQ System Redesign

1.1 HTML Structure Standardization

- Create a single, consistent HTML template for MCQs that:
 - Has clear question formatting with space for context
 - Provides four well-formatted answer options
 - Includes a dedicated feedback area for each question
 - Contains proper buttons: "Check Answers", "Try Again", and "Check the Grammar"
 - Supports a progressive difficulty structure for 10 questions

1.2 JavaScript Module Rebuild

- Develop a new `mcq.js` with these key features:
 - Initialization with configurable options
 - Clear option selection handling
 - Comprehensive answer checking
 - Detailed feedback display including explanations for both correct and incorrect answers
 - Ability to retry the quiz fully
 - Support for tracking progress and completion
 - Smooth scrolling to the summary table when "Check the Grammar" is clicked
 - Ability to track question difficulty progression

1.3 Question Design Guidelines

- Create guidelines for question design that ensures:
 - Questions test the student's ability to use grammar, not just recognize it
 - Questions provide sufficient context
 - Early questions are simpler, with later questions more challenging
 - All questions test the specific grammar point being taught
 - Each question has a specific, helpful explanation ready

2. Progress Tracking System Redesign

2.1 Core Data Structure

- Design a clear structure for storing MCQ progress:
 - Track overall completion status

- Store scores
- Record which specific questions were answered correctly/incorrectly
- Maintain timestamp information
- Store whether feedback was viewed

2.2 Storage Management

- Implement unified local storage approach:
 - Consistent key naming structure
 - Proper serialization/deserialization
 - Error handling with fallbacks
 - Storage space management

2.3 Progress Tracking API

- Build a clean API for the MCQ module to use:
 - `markCompleted(score, totalQuestions)` function
 - `isCompleted()` check
 - `getLastScore()` retrieval
 - `resetProgress()` functionality

3. Technical Requirements

3.1 MCQ HTML Template Elements

- Container for the entire MCQ
- 10 question elements with:
 - Clear question text area
 - Four option elements
 - Feedback display area with explanation
- Control buttons:
 - "Check Answers"
 - "Try Again"
 - "Check the Grammar" (with anchor link to summary table)
- Score display area

3.2 MCQ JavaScript Functions

- `init()` - Initialize the MCQ

- `selectOption()` - Handle option selection
- `checkAnswers()` - Check all answers and provide feedback
- `showFeedback()` - Display detailed explanations
- `resetQuiz()` - Reset all selections and feedback
- `saveProgress()` - Save completion status
- `scrollToSummary()` - Scroll to grammar summary table

3.3 Progress Tracking Requirements

- Store per-activity completion status
- Record scores
- Track timestamps of completion
- Handle resuming from saved state
- Support session recovery

4. Implementation Strategy

4.1 Phase 1: MCQ Template & Module

1. Create the standardized MCQ HTML template
2. Build the new MCQ JavaScript module
3. Implement the basic progress tracking functionality

4.2 Phase 2: Question Design

1. Create a template for crafting appropriate questions
2. Develop guidelines for writing helpful explanations
3. Design a difficulty progression framework

4.3 Phase 3: Testing & Refinement

1. Test the MCQ system with various question types
2. Verify progress tracking functions correctly
3. Ensure the system works across different browsers and devices

5. MCQ Design Criteria

- a) The MCQ should test the student's ability to use the grammar, not recognize it.
- b) At the end of the quiz the student should be able to check answers and get feedback.
- c) The feedback has to include a short, useful explanation about why the answer is right or wrong.
- d) There should be a "try again" option for the student to retry the quiz.

- e) Include a "check the grammar" button which redirects the student to the summary table of that page.
- f) There should be 10 questions that start easy and gradually get harder.