

Quiz App Cheatsheet

Overview

- Single-file quiz app (quiz-app.html) — no dependencies, no build tools
- Supports multiple-choice and open-ended questions
- Uses Automata Learning Lab brand design (brutalist, no border-radius)
- Just open in browser: open quiz-app.html

JSON Quiz Format

- Multiple-choice: question, type, options[], correctAnswer (0-indexed)
- Open-ended: question, type, acceptedAnswers[] (case-insensitive match)
- Optional fields: explanation, source (shown as feedback)
- Example: {"question": "...", "type": "multiple-choice", "options": [...], "correctAnswer": 2}

App Views & State

- #initialPage — File upload + JSON format example
- #questionContainer — Quiz-taking interface with submit/skip buttons
- #resultContainer — Score display + export buttons
- State variables: questions[], currentQuestion, score, results[]

Export Formats

- CSV: Question, Your Answer, Correct Answer, Result columns
- Markdown (LLM Review): Structured for AI review with prompts
- Anki: Semicolon-delimited question;answer for flashcard import

Design System (CSS Custom Properties)

- Primary: --ink-black (#000), --warm-cream (#F5F3EB)
- Accents: --coral, --golden, --sage, --sky (+ -light tint variants)
- Typography: IBM Plex Sans (body), JetBrains Mono (buttons/code)
- Button hover: translateY(-4px) + 4px offset box-shadow

Key Functions

- handleFileUpload() — Parse JSON, enable Start button
- displayQuestion() — Render MC options or open-ended textarea
- submitAnswer() — Check answer, update score, show feedback
- showImmediateFeedback() — Animated panel with explanation + source
- downloadResults() / downloadMarkdownResults() / downloadAnkiFormat()

Folder Contents

- quiz-app.html — The entire application
- example-quiz.json — Sample quiz (The Office themed)
- assets/ — MCP architecture diagrams (Anthropic article)
- .claude/commands/ — Custom slash commands (welcome, anki)

