

PRD: QuickQuiz

Status: Draft

Authors: Robel Kebede (robel.kebede@bison.howard.edu), Trishma Garcon (Trishma.Garcon@bison.howard.edu), Xavier Green (Xavier.Green@bison.howard.edu), Dalvin Ticha (dalvin.ticha@bison.howard.edu), Caleb Orr (Caleb.Orr@bison.howard.edu)

Background

QuickQuiz is a simple web app where users create, and complete short trivia quizzes made of multiple questions with defined correct answers. Users log in with Google, create quizzes, browse quizzes from others, submit their answers, see which ones are correct, and reveal answers when stuck. This PRD defines the milestones, functional requirements, and non-functional requirements to deliver a minimal but complete challenge-style quiz app that fits within a one-week implementation timeline.

Milestone 1: Setup & Authentication

Functional Requirements

- **FR1.1:** Create a shared GitHub repository and document the basic workflow.
- **FR1.2:** Initialize React frontend.
- **FR1.3:** Initialize FastAPI backend.
- **FR1.4:** Configure Firebase Authentication for Google login.
- **FR1.5:** Allow users to log in with Google and view their display name/email in the UI.
- **FR1.6:** Allow users to log out.
- **FR1.7:** Restrict quiz creation and deletion APIs to authenticated users.

Non-Functional Requirements

- **NFR1.1:** Setup steps documented so any teammate can run frontend + backend locally in under 10 minutes.
- **NFR1.2:** Login/logout completes within a few seconds on a normal connection.
- **NFR1.3:** Secrets (API keys, Firebase config) stored securely via environment/config files, not hard-coded.
- **NFR1.4:** Clear project structure (e.g., `frontend/`, `backend/`, `backend/routers/`, `backend/models/`).
- **NFR1.5:** Login screen follows basic accessibility (labels, focus, contrast).

Milestone 2: Quiz Creation & Management (Core CRUD)

Functional Requirements

- **FR2.1:** Authenticated users can create a quiz with:
 - Title (required)
 - Optional short description
 - 1–10 questions
- **FR2.2:** Each question has:

- Question text
 - One correct answer (MVP)
- **FR2.3:** Store quizzes and questions in a persistent database (e.g., Firestore or a simple cloud-hosted DB) via FastAPI endpoints.
- **FR2.4:** Provide a “My Quizzes” page listing quizzes created by the logged-in user.
- **FR2.5:** Quiz creators can delete their own quizzes; associated questions are also removed.
- **FR2.6:** Only the quiz creator can delete their quizzes (ownership enforced in backend).
- **FR2.7:** Validate required fields on both client and server before saving.

Non-Functional Requirements

- **NFR2.1:** Quiz creation UI is simple and self-explanatory.
- **NFR2.2:** CRUD operations complete quickly for small quizzes (≤ 10 questions).
- **NFR2.3:** Show clear error messages on validation or API failures; no silent errors.
- **NFR2.4:** Use atomic operations to avoid partially saved quizzes.
- **NFR2.5:** Layout responsive on laptop and mobile.
- **NFR2.6:** Include basic tests for FastAPI routes handling quiz creation, retrieval, and deletion.

Milestone 3: Quiz Play, Scoring & Answer Reveal

Functional Requirements

- **FR3.1:** Users can browse a list of available quizzes (title + description).
- **FR3.2:** Selecting a quiz opens a play view showing its questions.
- **FR3.3:** Users can input answers for each question and submit the quiz.
- **FR3.4:** On submission:
 - Backend compares user answers to correct answers (case-insensitive, trimmed).
 - Returns which questions are correct/incorrect and total score.
- **FR3.5:** For each question, users can click “Reveal Answer” to see the correct answer after they submit or choose to reveal.
- **FR3.6:** For logged-in users, store attempt summary (quiz ID, score, timestamp) for possible future use.

Non-Functional Requirements

- **NFR3.1:** Answer checking is robust to capitalization and extra spaces.
- **NFR3.2:** Results view is clear: per-question feedback plus score.
- **NFR3.3:** Use loading states and error handling; no broken or blank pages.
- **NFR3.4:** Missing answers are treated as incorrect without causing errors.
- **NFR3.5:** Quiz play UI supports keyboard navigation and readable layout.
- **NFR3.6:** Backend enforces that quiz content and results cannot be tampered with from the client.

Milestone 4 (Stretch, If Time Allows): Leaderboard & Random Quiz

Functional Requirements

- **FR4.1:** Leaderboard page showing top users based on stats (e.g., quizzes completed or average score).
- **FR4.2:** “Random Quiz” button that fetches a random available quiz.
- **FR4.3:** Only logged-in users appear on the leaderboard.

Non-Functional Requirements

- **NFR4.1:** Leaderboard queries are efficient and do not noticeably slow down the app.
- **NFR4.2:** Random quiz fetch is fast.
- **NFR4.3:** Only safe public data (name or nickname + basic stats) is shown.
- **NFR4.4:** If no data exists, show a friendly empty-state message.