

Quiz Application: Feature Expansion Plan

Phase 1: Improve User Experience (UX/UI)

1. Add "Previous Question" support - allow users to go back and change their answers.
2. Timer per question or for the entire quiz - auto-move or warn on timeout.
3. Show progress - for example, "Question 3 of 10" indicator.
4. Add answer explanation - display after each question or at the end.
5. Disable "Next" until an option is selected - ensures participation.
6. Highlight correct and incorrect answers - use color feedback.

Phase 2: Code Improvements

1. Use Object-Oriented Programming (OOP) or ES6 modules for better structure.
2. Store user answers in a structured array instead of relying on DOM.
3. Add localStorage support - to save quiz state on refresh.
4. Add basic test cases - validate logic with unit testing (optional).

Phase 3: Advanced Features

1. Add backend (PHP + MySQL or Node.js + MongoDB) - for storing quiz data and scores.
2. Randomize question order and options - for each quiz attempt.
3. Add category & difficulty filters - users choose their quiz preferences.
4. Leaderboard - show top scores globally or by category.
5. Export results as PDF - download performance summary.
6. Voice-based quiz - for accessibility and better UX.

Recommended Tech Stack

Frontend: HTML, CSS, JavaScript (or React)

Backend: Node.js or PHP

Database: MongoDB (Node) / MySQL (PHP)

Authentication: JWT or Session-based login system

Hosting: Netlify (Frontend), Render/Vercel (Backend) or XAMPP (Local)