

MindQuest

Project Summary | Graduation Project II, Computer Engineering, An-Najah National University
Authors: Ahmad Dardook, Sayed Qutob | Supervisor: Dr. Sufyan Samara | 2026/2027

Overview

MindQuest is a full-stack web-based learning platform that tackles a common weakness of online education: static, passive content that reduces student engagement and delays formative feedback. It combines authored courses, animated and interactive lessons, embedded quizzes, progress tracking and role-based workflows in a single system.

Problem Addressed

- Limited interactivity in existing e-learning platforms, lowering motivation and retention.
- Fixed, non-adaptive lesson sequences and slow or limited feedback.
- Heavy administrative burden on instructors for organizing content and tracking progress.

Objectives

- Build an interactive learning environment using multimedia and animations.
- Support three roles: students, teachers and administrators.
- Provide course and lesson management, quizzes (formative and summative) and timely feedback.
- Offer progress tracking and reporting for individuals and cohorts.
- Support secure payment for premium course access.

Scope

Included: web platform, course and lesson management, quiz authoring and execution, progress reports, payment-based enrollment, admin and instructor management tools.

Excluded: native mobile apps, advanced AI-driven personalization, and large-scale production deployment.

Technologies

Frontend	React, Vite, Tailwind CSS, JavaScript (JSX)
Backend	Node.js, Express.js (RESTful JSON APIs)
Authentication	JWT with role-based access control (RBAC)
Databases	MySQL (users, courses, enrollments, quizzes, progress) and MongoDB (animations, notifications, event data), a hybrid approach
Tools	VS Code, Git/GitHub, Postman, XAMPP, local dev servers
Integrations	

	Third-party payment service (Stripe), Google sign-in, email notifications, AI-assisted lesson HTML and quiz generation via webhook
--	--

Main Features

Common: sign up and sign in (including Google), role-based home dashboard, notifications and messaging (WebSocket), profile management.

Students: browse and enroll in free or paid courses, view lessons and interactive animations, take quizzes with immediate feedback, track progress, view performance reports, rate courses.

Teachers: course and lesson builder (text, images, YouTube, code, tables, questions, mini-games, animations), upload animations, **Animation Studio** for authoring algorithm visualizations on a timeline/canvas editor, quiz builder with AI question generation, student progress monitoring.

Administrators: dashboard with system metrics, user management (roles, ban/unban), course approval and lifecycle management, teacher verification, and moderation of reported comments, with all actions audited.

Results and Evaluation

The working prototype met its core engineering objectives: role-aware enrollment and access control, media-rich lesson delivery, assessment data capture, derived progress reporting and payment-gated content. It shows that interactivity, assessment and governance can be integrated coherently into one maintainable platform. The solution is considered a partial resolution of the problem: effectiveness is not yet empirically proven.

Limitations

- Short development time, small team, limited budget; local/development environments only.
- Small-scale testing only; no large or longitudinal study of learning gains.
- No production-scale load testing, security hardening or full payment reconciliation.

Recommendations and Future Work

- **Near term:** improve UI/UX consistency, richer quiz feedback, database query optimization and caching.
- **Longer term:** native mobile app, adaptive and personalized learning paths, advanced instructor analytics, completion certificates and integration with providers such as Udemy or Coursera.
- **Open problems:** scalability under heavy load and long-term measurement of learning effectiveness.