

LevelUp (LearnHub) — Online Learning System

Project Report

Generated on: 26/10/2025, 8:12:43 pm

INTRODUCTION

LevelUp (also referenced as LearnHub) is a modern online learning platform designed to enable instructors to publish courses and learners to study through structured video lessons, quizzes, and interactive features. The system supports authentication, course creation, enrollment, video playback (YouTube and locally uploaded preview videos), quiz assessments, certificates, user profiles, and feedback.

LITERATURE SURVEY

Conventional LMS solutions (e.g., Moodle, Canvas) emphasize rich feature sets but can be heavy to customize. Contemporary MOOC platforms (e.g., Udemy, Coursera) offer streamlined course delivery and assessments. LevelUp focuses on a pragmatic architecture using a MERN-like stack to balance rapid development with extensibility: React + TypeScript for a fast, component-driven UI and Express + TypeScript + MongoDB for scalable APIs and persistence. File uploads are supported via Multer, while static content is served efficiently by the Node server.

MOTIVATION AND PROBLEM DEFINITION

Students and instructors need a lightweight yet complete platform to publish and consume courses without the complexity of large enterprise systems. The problem is to design an application that provides: user onboarding, course discovery, secure enrollment, reliable video playback, structured quizzes, progress tracking, and certificate generation — all with a clean developer experience.

OBJECTIVES

1. Provide a responsive, user-friendly learning experience.
2. Enable instructors to create and manage courses with media (thumbnails, preview videos).
3. Support learner enrollment and gated access to full course content.
4. Deliver assessments via quizzes and track user progress.
5. Generate verifiable course-completion certificates.
6. Maintain a modular, TypeScript-first codebase for maintainability.

PROPOSED METHODOLOGY / ARCHITECTURE

- Frontend: React + TypeScript (Vite) with protected routes, context-based auth, and modern UI.
- Backend: Express + TypeScript with RESTful APIs for auth, courses, videos, quizzes, chatbot, certificates, and enrollments. Multer handles file uploads (videos, thumbnails); static files served under /uploads and /public.
- Database: MongoDB + Mongoose models (User, Course, Video, Quiz, Enrollment, Feedback, Certificate).
- Media Handling: Disk storage for uploaded assets; preview video is stored on the Course; lessons currently use YouTube URLs (with planned support for full local lesson uploads).
- Security: JWT-based authentication; role-based authorization for instructors/admins.
- Certificates: HTML/Canvas-based styling rendered as images and served from /public/certificates.

RESULTS AND DISCUSSION

The platform supports end-to-end course flow: instructors can create courses with thumbnails and preview videos; learners can enroll and, when lessons are published, continue learning through a dedicated player. The learning page supports YouTube embeds and a fallback to play the uploaded course preview video when no lessons are available. Quizzes, certificates, feedback, and a chatbot enhance engagement. Static assets are served directly by the backend for simple deployment.

OUTCOMES

- Functional online learning application with authentication and role-based access.
- Course lifecycle: create !' publish !' enroll !' learn !' assess !' certify.
- Robust upload pipeline for thumbnails and preview videos; static serving via Express.
- Extensible foundation for adding local lesson uploads and richer analytics.

APPLICATIONS

- Academic institutions for hosting internal courses and training.
- Corporate upskilling and onboarding programs.
- Independent instructors and small training centers.

CONCLUSION

LevelUp demonstrates a practical, modern learning platform built on a TypeScript-forward web stack. It balances fast development with clear separation of concerns and sets the stage for enhancements like native file-based lessons, adaptive assessments, and richer progress analytics.

REFERENCES

- React, Vite, and React Router documentation.
- Express and Mongoose official docs.
- Multer file upload middleware docs.
- Tailwind CSS and Lucide Icons.
- Project repository codebase (LevelUp / LearnHub).