



# LearnHub – Complete LMS (Presentation)

[Download PDF](#)[Jump to TOC](#)

## Executive Overview

LearnHub is a production-ready Learning Management System with AI assistance, course and video management, interactive quizzes, progress tracking, and automated certificate generation. The UI follows a minimalist, Zara-inspired aesthetic with responsive, fast, and accessible UX.

**14**

Complete Courses

**97**

Videos with Tracking

**24**

Quizzes (Auto-grading)

**100%**

Feature Complete

Source: project README and backend seed endpoints.

## Table of Contents

- Architecture & Tech Stack
- Key Features
- Frontend
- Backend & APIs
- Data Models
- Security & Auth
- AI Integration
- Certificates
- Database Seeding
- Setup & Environment
- Pages & User Flows
- Future Enhancements

## Architecture & Tech Stack

### Frontend

- **React 18 + TypeScript** (Vite)
- **Tailwind CSS** (custom Zara-inspired theme)

- **React Router** for SPA navigation
- UI libs: @radix-ui/react-dialog, lucide-react, react-markdown, react-youtube/player

### Backend

- **Node.js + Express** with TypeScript
- **MongoDB + Mongoose**
- **JWT** authentication
- **Google Gemini AI** via @google/generative-ai
- Image processing with sharp (certificates)

Static assets and generated certificates are served from /public (backend) and frontend/public .

### Key Features

- **Authentication:** Registration, login, JWT-protected routes, profile
- **Course Management:** Catalog, categories, search/filter, enrollment
- **Video Learning:** Player, previews, per-video progress tracking
- **Quizzes:** Multiple types, auto-grading, attempts, pass criteria
- **Certificates:** Auto generation as PNG with overlay text
- **AI Assistant:** Chat, suggestions, AI-generated quizzes (Gemini)
- **Analytics:** Progress metrics per enrollment
- **Design:** Minimalist, responsive, dark/light aware

### Frontend (React + TS)

- **Pages:** Landing, Login, Register, Dashboard, Courses, Course Detail, Course Learning, Profile, Admin (Create Course/Add Video)
- **Components:** Header, Chatbot, QuizTaking, QuizList, AIQuizModal, ProfileCompletionModal, Global AI button
- **Contexts:** AuthContext , ThemeContext

Build: npm run dev with Vite; Production: npm run build .

### Backend & REST API

Base URL: http://localhost:5000

#### System

GET	/api/health	Public
-----	-------------	--------

#### Auth

POST	/api/auth/register	Public
POST	/api/auth/login	Public
POST	/api/auth/logout	Public
GET	/api/auth/profile	JWT

## Courses

GET	/api/courses	Public
GET	/api/courses/categories	Public
GET	/api/courses/:courseId	Public/JWT (enroll check)
GET	/api/courses/:courseId/feedback	Public
POST	/api/courses/:courseId/feedback	JWT
POST	/api/courses/:courseId/enroll	JWT
DELETE	/api/courses/:courseId/enroll	JWT
GET	/api/courses/user/enrollments	JWT
POST	/api/courses	JWT (instructor/admin)

## Videos

GET	/api/videos/:videoId	Public/JWT (preview rules)
GET	/api/videos/course/:courseId	Public/JWT
POST	/api/videos/:videoId/progress	JWT
POST	/api/videos	JWT (instructor/admin)
PUT	/api/videos/:videoId	JWT (instructor/admin)

## Quizzes

GET	/api/quizzes/course/:courseId	JWT
GET	/api/quizzes/:quizId	JWT
POST	/api/quizzes/:quizId/submit	JWT
GET	/api/quizzes/:quizId/results	JWT
POST	/api/quizzes	JWT (instructor/admin)
PATCH	/api/quizzes/:quizId/status	JWT (instructor/admin)

## Chatbot (AI)

POST	/api/chatbot/chat	JWT
GET	/api/chatbot/suggestions	JWT
POST	/api/chatbot/generate-quiz	JWT
GET	/api/chatbot/test	JWT (admin)

## Certificates

POST	/api/certificates/generate/:courseId	JWT
GET	/api/certificates/my-certificates	JWT
GET	/api/certificates/download/:certificateId	JWT

## Seeding

GET	/api/seed?force=true	Public
-----	----------------------	--------

## Data Models (Summary)

### User

- name, email (unique), password (hashed), avatar, role
- profile: DOB, gender, phone, country, institution, position, dept, yearOfStudy
- preferences: experienceLevel, fieldOfInterest[], learningMode, language
- career: careerGoal, profession, organization
- socialLinks: linkedin, twitter, github, website

### Course

- title, shortDescription, description, category, level, price, duration
- instructor { id, name, email }, thumbnail, previewVideo, tags[]
- rating { average, count }, enrollmentCount, isPublished
- learningOutcomes[], requirements[], importantTopics[]

### Video

- courseId, title, description, videoUrl, thumbnail
- duration (seconds), order, isPreview, resources[], transcription

### Quiz

- courseId, title, description, instructions[]
- settings: timeLimit, passingScore, maxAttempts, randomize, showCorrectAnswers, allowReview
- questions[]: id, type, options[], correctAnswers[], explanation, points, order

- `attempts[]`: `userId`, `answers[]`, `score`, `percentage`, `timeSpent`, `passed`

## Enrollment

- `userId`, `courseId`, `status` (`active/completed/...`)
- `progress`: `videosCompleted/total`, `quizzesPassed/total`, `overallPercentage`
- `videoProgress[]`: `per-video watchedDuration`, `completedAt`
- `quizAttempts[]` with scoring snapshot

## Certificate

- `userId`, `courseId`, `userName`, `courseName`, `issuedAt`
- `certificateUrl` (PNG in `backend/public/certificates`)

## Security & Auth

- **JWT** auth with bearer tokens; middleware: `authenticate`, `authorize`, `optionalAuth`
- **Password hashing** via `bcryptjs` (salt 12); password not selected by default
- **Role-based access** for instructor/admin on sensitive routes
- **CORS** configured via `CORS_ORIGINS`

## AI Integration (Gemini)

- Chat with educational system prompt, conversation history context
- Course-aware suggestions and topic-specific assistance
- AI-generated quizzes from existing course/video context
- Config via `GEMINI_API_KEY` and `@google/generative-ai`

## Certificates

- PNG generation with `sharp` using template from `frontend/public/levelUpcertificate.png`
- Overlay user name and course name; files stored under `backend/public/certificates`
- Endpoints to generate, list, and download user certificates

## Database Seeding

- Seed endpoint: `GET /api/seed?force=true`
- Creates 14 courses, 97 videos, and 24 quizzes with sample instructor

## Setup & Environment

## Backend

- Node 18+, MongoDB 5+
- Env: `MONGODB_URI`, `JWT_SECRET`, `PORT`, `NODE_ENV`, `GEMINI_API_KEY`, `CORS_ORIGINS`
- Run: `cd backend && npm i && npm run dev`

## Frontend

- Run: `cd frontend && npm i && npm run dev`
- Dev URL: `http://localhost:5173`
- Open this file directly or via dev server: `/LearnHub-Presentation.html`

## Pages & User Flows

- **Landing** → Explore courses → View details → Enroll
- **Learning** → Watch videos → Progress auto-tracked → Attempt quizzes
- **Completion** → Pass quizzes + reach progress threshold → Generate certificate
- **Dashboard** → Overview of enrollments, progress, and AI assistant
- **Admin/Instructor** → Create courses/videos, manage quizzes

## Future Enhancements

- Advanced analytics dashboards
- Social learning (forums), notifications
- Mobile apps (iOS/Android)
- Multi-language (i18n) and accessibility audits
- Automated test coverage

© LearnHub – MIT License. This presentation is auto-generated from the repository structure and source descriptions for academic demonstration.