1. Project Title

EduConnect – An Online Learning Management System

---

2. Problem Statement

In the modern education landscape, both students and instructors face challenges in managing online learning effectively. Traditional platforms often lack seamless integration between course materials, assessments, progress tracking, and communication tools.
EduConnect aims to provide a unified, user-friendly platform for managing online courses, tracking student performance, submitting assignments, and fostering instructor-student engagement—all in one place.

---

3. System Architecture

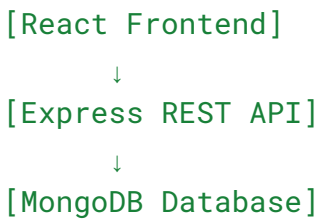
Architecture Overview:

Frontend → Backend (API) → Database

Technology Stack

- **Frontend:** React.js with React Router for smooth navigation
- **Backend:** Node.js + Express.js REST API
- **Database:** MongoDB (non-relational)
- **Authentication:** JWT-based login/signup system with role-based access (Admin, Instructor, Student)
- **Hosting:**
    - **Frontend:** Vercel
    - **Backend:** Render
    - **Database:** MongoDB Atlas

4. System Diagram (Conceptual)



---

5. Key Features

Category	Features
Authentication & Authorization	Secure user registration, login, logout; role-based access for Admin, Instructor, and Student
Course Management	Instructors can create, edit, delete, and manage courses with lessons, videos, and materials
Enrollment System	Students can browse and enroll in available courses
Assignment & Quizzes	Instructors can upload assignments and quizzes; students can submit responses
Progress Tracking	Track course completion percentage, grades, and performance analytics
Dashboard	Role-based dashboards for Admin (system overview), Instructor (course stats), and Student (learning progress)
Discussion Forum	Course-specific Q&A and discussions between students and instructors
Notifications	Real-time updates on new courses, deadlines, and grades
Hosting	Frontend and backend deployed to cloud platforms (Vercel + Render) for live accessibility

6. Tech Stack

Layer	Technologies
Frontend	React.js, React Router, Axios, TailwindCSS
Backend	Node.js, Express.js
Database	MongoDB (via Mongoose ORM)
Authentication	JWT (JSON Web Token), bcrypt.js for password hashing
Hosting	Frontend → Vercel / Netlify Backend → Render / Railway Database → MongoDB Atlas
Version Control	Git & GitHub

7. API Overview

Endpoint	Method	Description	Access
`/api/auth/signup`	POST	Register a new user (student/instructor/admin)	Public
`/api/auth/login`	POST	Authenticate user and return JWT token	Public
`/api/courses`	GET	Get all available courses	Authenticated
`/api/courses/:id`	GET	Get course details by ID	Authenticated
`/api/courses`	POST	Create a new course	Instructor only
`/api/courses/:id`	PUT	Update course details	Instructor only
`/api/courses/:id`	DELETE	Delete a course	Admin only

`/api/assignments`	GET	Get all assignments	Authenticated
`/api/assignments/:id/submit`	POST	Submit assignment	Student only
`/api/grades/:id`	GET	View student grades	Authenticated
`/api/forum/:courseId`	POST	Create or reply to discussion threads	Authenticated
