# 📚 Learning Management System (LMS)

A full-stack Learning Management System built using the \*\*MERN\*\* stack (MongoDB, Express, React, Node.js). It supports multiple user roles, allows for the creation and management of courses, and facilitates interactive learning through content uploads, assessments, and communication tools.

---

## 🚀 Features

- 🔐 \*\*Role-based Access Control\*\*: Admin, Teacher, and Student dashboards

- 📦 \*\*Course Management\*\*: Create and manage courses and modules

- 📁 \*\*Content Upload\*\*: PDFs, videos, and quizzes

- 🧪 \*\*Assessments\*\*: Assignments and MCQ-based quizzes

- 💬 \*\*Communication\*\*: Discussion forums and messaging

- 📊 \*\*Progress Tracking\*\*: Visual dashboards for users

- 🔒 \*\*JWT Authentication\*\*: Secure login & route protection

- ☁️ \*\*Cloud Ready\*\*: Easily deployable to AWS, Vercel, Render, or Netlify

---

## 🛠 Tech Stack

| Tech | Purpose |

|------------|----------------------------------|

| MongoDB | NoSQL database for data storage |

| Express.js | Backend API framework |

| React.js | Frontend UI library |

| Node.js | Backend runtime environment |

| TailwindCSS| Frontend styling framework |

| JWT | Token-based authentication |

---

## 📁 Project Structure

LMS/ ├── client/ # React frontend │ └── src/ │ └── api/ │ └── axiosInstance.js # Handles API calls ├── server/ # Node.js + Express backend │ └── server.js # Server entry point

---

## ⚙️ Local Setup Instructions

### 1. Clone the Repository

```

git clone https://github.com/yourusername/lms-project.git

cd LMS

```

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### 2. Backend Setup

```

cd server

npm install

```

Create a .env file in /server and add:

```

PORT=5000

MONGO\_URI=your\_mongodb\_connection\_string

JWT\_SECRET=your\_jwt\_secret

```

Start the server:

```

npm run dev

```

🚀 Server runs at: http://localhost:5000

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### 3. Frontend Setup

```

cd client

npm install

```

Update the API base URL in client/src/api/axiosInstance.js:

```

const instance = axios.create({

baseURL: 'http://localhost:5000/api',

});

```

Start the frontend:

```

npm run dev

```

🌐 Frontend runs at: http://localhost:3000

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### 🌐 Deployment

To deploy in production, use environment variables to avoid hardcoding URLs.

Example:

# In client/.env

```

REACT\_APP\_API\_URL=https://your-production-backend.com/api

Then in axiosInstance.js:

const instance = axios.create({

baseURL: process.env.REACT\_APP\_API\_URL,

});

```

Suggested Platforms:

Service Usage

Vercel Frontend Hosting

Render Backend Hosting

MongoDB Atlas Cloud DB

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

🧪 Testing

• Manual testing across user roles (Admin, Teacher, Student)

• Unit and Integration testing for backend APIs

• UI validation for all views and edge cases

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

📅 Project Timeline

Phase Description

🔍 Planning Requirement gathering and analysis

🎨 Design UI wireframes and DB schema

🧱 Development API, UI, integration

🧪 Testing Unit tests and user acceptance testing

🚀 Deployment Cloud hosting and final review

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

🙋 Author

👨‍💻 Abhishek Dixit

Final Year B.Tech IT Student, Amity University Maharashtra

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

📜 License

This project is for educational use. You are free to clone, fork, and modify it.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_