

Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	10 Feb 2026
Team ID	LTVIP2026TMIDS64190
Project Name	LearnHub: Your Center for Skill Enhancement
Maximum Marks	4 Marks

Technical Architecture:

LearnHub is built on a modular MERN stack architecture designed to provide secure, scalable, and structured online learning. The system follows a layered architecture model where the React frontend handles user interaction, the Node.js & Express backend manages business logic and API routing, and MongoDB serves as the centralized data storage layer.

User requests flow from the frontend through REST APIs to the backend, where authentication is verified using JWT middleware. The backend processes course management, payment verification, progress tracking, and certificate generation before interacting with the MongoDB database. Video files are handled using Multer and stored locally (or cloud-ready for future enhancement). PDFKit dynamically generates certificates once course completion criteria are satisfied.

This layered and modular design ensures security, scalability, maintainability, and smooth interaction between Admin, Teacher, and Student users.



LearnHub – Guidelines

- ✓ **Be learner-first:** Design the platform to be simple, intuitive, and accessible for students, teachers, and administrators of all technical levels.
- ✓ **Ensure structured learning:** Provide clear course modules, organized video sections, and visible progress tracking to enhance the learning experience.
- ✓ **Protect user data:** Implement secure authentication using JWT, role-based access control (Admin, Teacher, Student), and encrypted password storage.
- ✓ **Secure payments:** Ensure safe course enrollment through validated payment processes before granting course access.
- ✓ **Enable role-based control:** Allow teachers to upload and manage courses, students to enroll and track progress, and admins to monitor and control platform activities. ↗ □
- ✓ **Support scalability:** Build the system using the MERN stack with modular architecture to allow easy addition of new features.
- ✓ **Provide real-time updates:** Offer instant enrollment confirmation, section completion tracking, and certificate generation upon course completion.
- ✓ **Encourage continuous improvement:** Collect user feedback and monitor system performance to enhance usability and learning outcomes. ↗

TECHNICAL ARCHITECTURE OF LEARNHUB

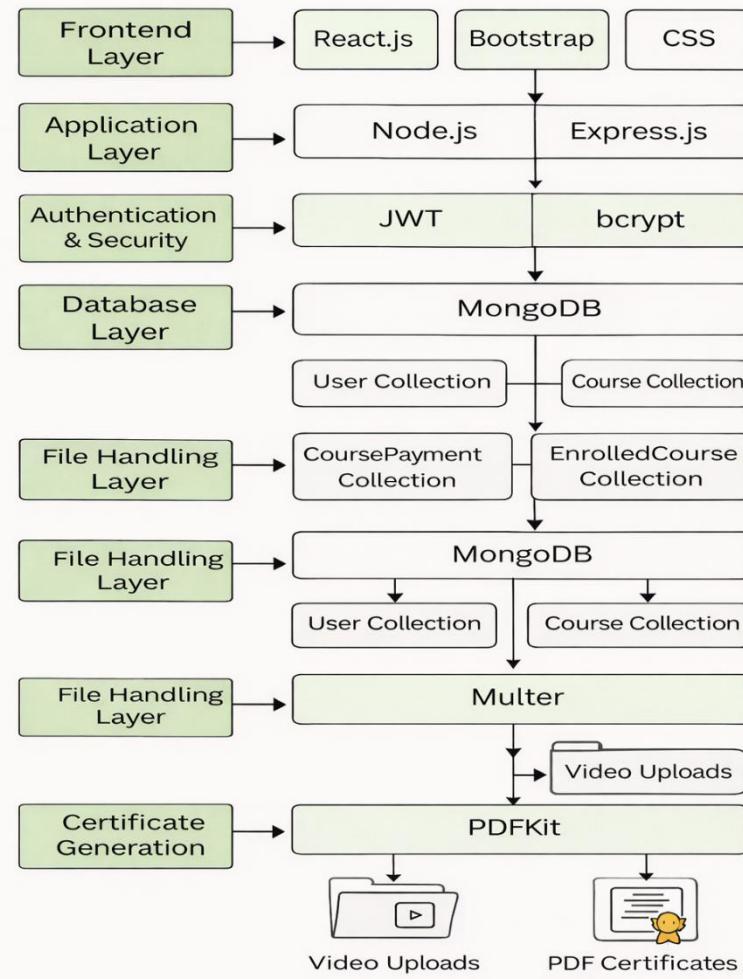


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Web-based interactive dashboards for Admin, Teacher, and Student	React.js, HTML, CSS, Bootstrap
2.	Backend Server	Handles routing, authentication, business logic, and API management	Node.js, Express.js
3.	Database	Stores users, courses, enrollments, and payment records	MongoDB (Mongoose ORM)
4.	Authentication System	Secure login and session management	JWT (JSON Web Token), bcrypt
5.	File Upload System	Upload and manage course videos	Multer
6.	Payment Module	Simulated payment validation before enrollment	Custom Express API
7.	Certificate Generator	Generates downloadable PDF certificates	PDFKit
8.	API Communication	Handles frontend-backend communication	RESTful APIs, Axios

9.	Role-Based Access Control	Restricts access based on user type (Admin, Teacher, Student)	Middleware (RBAC)
10.	Development Environment	Local development and testing	Node.js runtime, npm
11.	Infrastructure (Server / Cloud)	Deployment on local server or cloud platforms	AWS / Azure / Render / Docker (Future Ready)

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Built entirely using open-source technologies	MERN Stack
2.	Security Implementations	Password hashing and token-based authentication	Role-Based Access Control
3.	Scalable Architecture	Modular backend and component-based frontend	bcrypt, JWT
4.	Availability	Can be deployed on cloud with 24/7 availability	express + React Architecture
5.	Performance	Optimized database queries and efficient API responses	MongoDB Indexing, Async APIs
6.	Maintainability	Clean folder structure and separation of concerns	MVC-like Structure
7.	Data Integrity	Schema validation and relationship mapping	Mongoose Schema Validation
8.	Future Expandability	Future Expandability	Modular API Design