Project Design Phase

Solution Architecture

Date	**24 June 2025**
Team ID	**LTVIP2025TMID20425**
Project Name	**LearnHub: Your Center for Skill Enhancement**
Mentor Name	**Dr Shaik Salma Begam**
Maximum Marks	**4 Marks**

Solution Architecture

The **solution architecture for LearnHub** bridges the gap between the educational platform's core needs—**interactive learning, user management, secure access, and certification**—and the technology solutions provided by the **MERN stack, RESTful APIs, cloud hosting**, and scalable deployment strategies.

Goals of LearnHub's Solution Architecture

---

1. Find the Best Tech Solution to Solve Existing Learning Platform Problems

- **Tech Stack:** MERN stack (MongoDB, Express.js, React, Node.js)
- **Why MERN:** Enables fast development, seamless integration between client and server, and supports real-time user interactions, progress tracking, and efficient content delivery.

---

2. Describe the Structure, Behavior, and Components to Stakeholders

- **Frontend (React + Vite):** Handles UI/UX, routing, course browsing, and role-based views (student/instructor/admin).
- **Backend (Node.js + Express):** Manages API endpoints for registration, course content, user roles, and certificate generation.
- **Database (MongoDB):** Stores user profiles, course content, progress data, and admin configurations.
- **Authentication:** JWT-based secure login and session handling.

- **Cloud Storage (optional):** For storing course videos, certificates, and documents (e.g., AWS S3 or Firebase).

---

3. Define Features, Phases, and Requirements

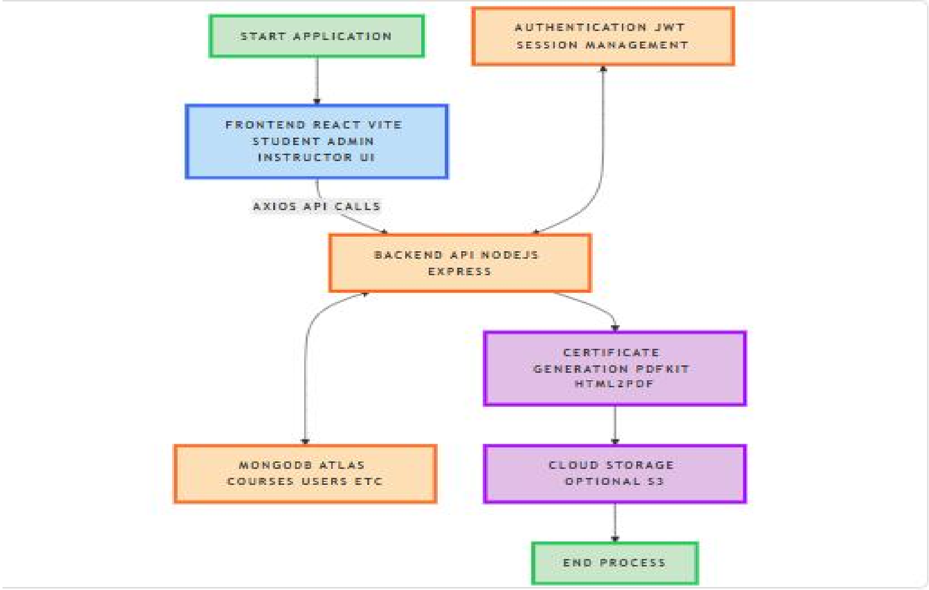
- **Core Features:**
  - User Registration & Login (JWT)
  - Course Management (CRUD by Admin/Instructor)
  - Enrollment & Progress Tracking (Student)
  - Certificate Generation
  - Dashboard for All Roles
- **Phases:**
  - Phase 1: Auth & Course Listing
  - Phase 2: Enrollment & Progress
  - Phase 3: Certificate & Admin Tools

---

4. Provide Specifications for Development and Deployment

- **REST API Specifications:** Clearly defined endpoints (e.g., `/api/users`, `/api/courses`, `/api/enroll`, `/api/certificates`)
- **Deployment Plan:** Vercel/Netlify for frontend, Render/Heroku for backend, MongoDB Atlas for cloud DB.
- **DevOps:** Use of `.env`, Docker (optional), GitHub CI/CD pipeline (optional for production).

Example - Solution Architecture Diagram:


START APPLICATION
AUTHENTICATION JWT SESSION MANAGEMENT
FRONTEND REACT VITE STUDENT ADMIN INSTRUCTOR UI
AXIOS API CALLS
BACKEND API NODEJS EXPRESS
CERTIFICATE GENERATION PDFKIT HTML2PDF
MONGODB ATLAS COURSES USERS ETC
CLOUD STORAGE OPTIONAL S3
END PROCESS