**Project Design Phase**

**Solution Architecture**

|  |  |
| --- | --- |
| Date | 23 june 2025 |
| Team ID | LTVIP2025TMID54182 |
| Project Name | LearnHub: Your Center for Skill Enhancement |
| Maximum Marks | 4 Marks |

## Solution Architecture (Long Version)

**Solution architecture** is a comprehensive framework that defines how to transform business requirements into effective technology solutions. For the **Online Learning Platform (OLP)**, the objective is to design a scalable, user-centric, and technically robust solution that facilitates seamless learning and teaching experiences for students, instructors, and administrators.

### Objectives of the Solution Architecture

1. **Bridge Business Goals with Technology Solutions**  
   The platform solves critical modern educational challenges by:
   * Offering accessible education from anywhere, anytime.
   * Supporting multiple user roles (Student, Teacher, Admin).
   * Integrating secure login, payment, and certification mechanisms.
   * Allowing teachers to create and manage content.
   * Enabling admins to monitor and control operations.
2. **Define System Structure and Characteristics**  
   The solution is based on a **modular MERN stack architecture**:
   * **Frontend:** Built with React.js using Vite, styled with Bootstrap and Material UI. This ensures high responsiveness and modular component design.
   * **Backend:** Uses Node.js with Express.js to build RESTful APIs. Middleware handles authentication and routing.
   * **Database:** MongoDB, a NoSQL database, is used with Mongoose for flexible and efficient data handling.
   * **Infrastructure:** Can be hosted on platforms like Vercel (frontend), Render or Heroku (backend), and MongoDB Atlas (database).
3. **Outline Features, Development Phases & Requirements**  
   Development is divided into iterative phases:
   * **Phase 1:** Setup user registration/login, basic course listing.
   * **Phase 2:** Implement role-based dashboards, teacher course uploads, student enrollment, and payment processing via Stripe.
   * **Phase 3:** Generate downloadable certificates, introduce admin analytics, user feedback, and moderation systems.
4. **Deliver Clear Technical Specifications**
   * **Frontend:** Vite-powered React.js interface with Material UI and Axios for API calls.
   * **Backend:** Express.js server with routes, middleware, and JWT-based security.
   * **Database:** MongoDB connected via Mongoose ODM.
   * **File Uploads:** Handled using Multer.
   * **Third-Party Services:** Stripe for payment processing and Nodemailer for email confirmations.
   * **Security:** JWT for authentication, bcrypt for password hashing, and environment-based config via dotenv.
   * **Deployment:** CI/CD deployment via Vercel (UI) and Render/Heroku (backend).

Solution Architecture Diagram (Text Form)

less

CopyEdit

[ Users (Students, Teachers, Admin) ]

|

v

[ React Frontend (Browser) ]

|

v

[ Express.js Backend Server ]

|

v

[ MongoDB Database (Cloud via Atlas) ]

|

+--> [ Stripe API - for Payments ]

|

+--> [ Nodemailer - for Email Confirmation ]

### Key Benefits of the Architecture

* **Scalability:** Components are loosely coupled and independently scalable.
* **Security:** JWT, bcrypt, and environment configurations ensure data protection.
* **Flexibility:** Easily supports new modules like mobile apps or machine learning integrations.
* **Performance:** Fast build times and API responsiveness due to optimized Vite and RESTful structure.
* **User Experience:** Intuitive UI and role-based navigation enhance accessibility and usability.

**Example - Solution Architecture Diagram:**

