**Project Design Phase-II**

**Solution Requirements (Functional & Non-functional)**

| Date | 22 June 2025 |
| --- | --- |
| Team ID | LTVIP2025TMID56577 |
| Project Name | LearnHub: Your Center for Skill Enhancement |
| Maximum Marks | 4 Marks |

### ****Technical Architecture****

The Online Learning Platform (OLP) adopts a **Client-Server Architecture**, using the **MERN Stack**:

* **Frontend (Client):** React.js (with Vite, Bootstrap, Material UI)
* **Backend (Server):** Node.js + Express.js
* **Database:** MongoDB (NoSQL)
* **Infrastructure:** Locally hosted or deployable on cloud services (e.g., Render, AWS)
* **Communication:** RESTful APIs using Axios
* **Optional Add-ons:** Stripe for payments, Nodemailer for emails

🗂 **Infrastructural Demarcation:**

* UI & logic: Browser (client)
* Server logic: Node.js server
* Data: MongoDB (Cloud or local)
* External APIs: Email, Payment (optional)
* ML model: Not applicable

### 📋 Table-1: Components & Technologies

| **S.No** | **Component** | **Description** | **Technology / Tool Used** |
| --- | --- | --- | --- |
| 1 | User Interface | Web UI | React.js, HTML, CSS, Bootstrap, Material UI |
| 2 | Application Logic-1 | Authentication (login, JWT, roles) | Node.js, Express.js, JWT, bcrypt.js |
| 3 | Application Logic-2 | Course Enrollment, Dashboard Logic | Node.js, Express.js, REST APIs |
| 4 | Application Logic-3 | Certificate Generation & Payment Integration | HTML-PDF (local), Stripe (optional) |
| 5 | Database | Course/User Data | MongoDB, Mongoose ODM |
| 6 | Cloud Database | Hosted MongoDB | MongoDB Atlas |
| 7 | File Storage | Store course media | Multer (Local), Cloudinary (Optional) |
| 8 | External API-1 | Email Notifications | Nodemailer, Gmail API |
| 9 | External API-2 | Payment Integration | Stripe API |
| 10 | Machine Learning Model | Not used in current version | - |
| 11 | Infrastructure | Local or Cloud Deployment | Localhost, Render, Vercel, AWS, etc. |

### 📋 Table-2: Application Characteristics

| **S.No** | **Characteristics** | **Description** | **Technology / Tool Used** |
| --- | --- | --- | --- |
| 1 | Open-Source Frameworks | Tools used in full stack | React.js, Node.js, Express.js |
| 2 | Security Implementations | JWT, password hashing, environment variables | bcrypt.js, JWT, dotenv |
| 3 | Scalable Architecture | Decoupled front and backend | MERN Stack, REST APIs |
| 4 | Availability | Can scale with cloud hosting | Render, AWS, Netlify |
| 5 | Performance | Fast frontend, API optimization, caching (future) | Vite, Axios, Express.js |

### 📎 References:

* <https://react.dev/>
* <https://vitejs.dev/>
* <https://expressjs.com/>
* <https://www.mongodb.com/atlas>
* <https://developer.ibm.com/patterns/>
* <https://c4model.com/>