**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 24 June2025 |
| Team ID | LTVIP2025TMID52072 |
| Project Name | Learn Hub |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

The LearnHub platform uses a full-stack MERN architecture designed to be scalable, secure, and responsive. Below is the architectural diagram (to be added separately as per submission guidelines) followed by detailed tables of components and characteristics.

**Table-1: Components & Technologies:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
| 1. | User Interface | Web-based interface for students, teachers, and admin | HTML, CSS, JavaScript, React.js, Bootstrap |
| 2. | Application Logic-1 | Authentication and Authorization | Node.js, Express.js, JWT |
| 3. | Application Logic-2 | Course creation and management logic | Node.js, Express.js |
| 4. | Application Logic-3 | Assignment of courses and section updates | Node.js, Express.js |
| 5. | Database | Stores users, courses, sections, assignments | MongoDB (NoSQL) |
| 6. | Cloud Database | (Optional if using a cloud-managed DB) | MongoDB Atlas |
| 7. | File Storage | Stores assets like thumbnails or course PDFs (optional) | Local filesystem / Cloudinary / AWS S3 |
| 8. | External API-1 | None currently (can integrate AI or analysis APIs later) | — |
| 9. | External API-2 | (Future scope for integration with EdTech APIs) | — |
| 10. | Machine Learning Model | (Optional in future for recommendation systems) | TensorFlow / scikit-learn (Future Integration) |
| 11. | Infrastructure | Cloud-ready, runs locally during development | Node.js server on Render / Vercel / Localhost |

**Table-2: Application Characteristics:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Characteristics** | **Description** | **Technology / Implementation** |
| 1. | Open-Source Frameworks | React.js, Node.js, Express.js, MongoDB | MERN Stack |
| 2. | Security Implementations | JWT authentication, password hashing with bcrypt, route protection | bcrypt, JWT, Helmet, CORS |
| 3. | Scalable Architecture | Built with 3-tier architecture: Frontend, Backend API, and Database | MERN architecture + REST APIs |
| 4. | Availability | Supports future deployment on scalable cloud infra | Can deploy on Render, Heroku, Vercel, or Kubernetes |
| 5. | Performance | REST APIs, optimized DB queries, frontend component reuse, lazy loading | Axios, React Suspense, MongoDB indexes |

This design ensures smooth delivery of learning content, robust teacher and student interactions, and easy course management. Future integrations can include ML-powered recommendations, live classrooms, and analytics dashboards.

**References:**

* <https://c4model.com/>
* <https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/>
* <https://www.ibm.com/cloud/architecture>
* <https://aws.amazon.com/architecture>
* <https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d>