**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 22 June 2025 |
| Team ID | LTVIP2025TMID54785 |
| Project Name | LearnHub: Your Center For Skill Enhancement |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

**Example: Order processing during pandemics for offline mode**

Guidelines:

Include all the processes (As an application logic / Technology Block)

Provide infrastructural demarcation (Local / Cloud)

Indicate external interfaces (third party API’s etc.)

Indicate Data Storage components / services

Indicate interface to machine learning models (if applicable)



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

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology Used** |
| 1 | **User Interface** | Web-based UI for students, teachers, and admin interactions | React.js, HTML, CSS, JavaScript, Bootstrap |
| 2 | **Application Logic – 1** | Handles routing, API calls, course logic, and business operations | Node.js, Express.js |
| 3 | **Application Logic – 2** | Role-based login, user sessions, and secure access handling | JWT (JSON Web Token), bcrypt.js |
| 4 | **Application Logic – 3** | Chat system between students and teachers/admins (future enhancement) | Socket.io (planned) |
| 5 | **Database** | Stores users, courses, enrollments, roles, and learning progress | MongoDB |
| 6 | **Cloud Database** | Cloud-hosted scalable NoSQL storage | MongoDB Atlas |
| 7 | **File Storage** | For uploading media content (course images/videos/screenshots) (optional) | Local filesystem or Cloudinary (optional) |
| 8 | **External API – 1** | Location/IP tracking features (future enhancement) | IPInfo API (optional) |
| 9 | **External API – 2** | Not used in the current version | — |
| 10 | **Machine Learning Model** | Not applicable in this version | — |
| 11 | **Infrastructure** | Hosting frontend and backend on cloud platforms | Render (Backend), Vercel / Netlify (Frontend) |

**Table-2: Application Characteristics:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Characteristics** | **Description** | **Technology Used** |
| 1 | **Open-Source Frameworks** | Core frameworks and libraries used in full-stack app development | React.js, Node.js, Express.js, MongoDB |
| 2 | **Security Implementations** | Implements secure user access, encrypted credentials, and token-based sessions | bcrypt.js, JWT, Helmet (Express middleware) |
| 3 | **Scalable Architecture** | Modular, layered design with separation of concerns (frontend, backend, database) | MERN Stack (3-tier architecture) |
| 4 | **Availability** | Deployed on cloud infrastructure ensuring high uptime and global accessibility | Render, MongoDB Atlas |
| 5 | **Performance** | Fast user interactions, asynchronous API calls, and optimized data handling | React, Axios, MongoDB |