

Project Design Phase-II Technology Stack (Architecture & Stack)

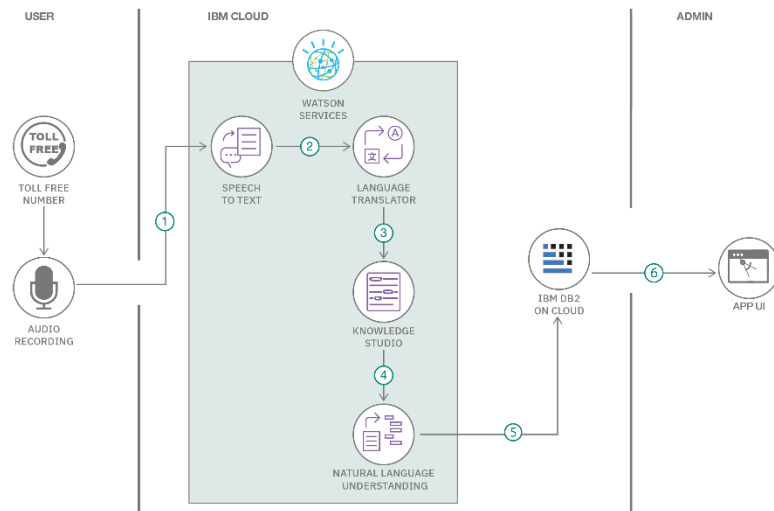
Date	24 June 3035
Team ID	LTVIP2025TMID56697
Project Name	LearnHub: Your Center for Skill Enhancemen
Maximum Marks	4 Marks

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

Reference: <https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/>



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)

S.No	Layer/Component	Description	Technology Used
1	User Interface	Web-based interface for students, teachers, admin	HTML, Tailwind CSS, JavaScript, React.js
2	Application Logic-1	Backend logic for user roles and course management	Node.js (Express.js)
3	Application Logic-2	Practice code compiler integration	Judge0 Compiler API
4	Application Logic-3	Certificate generation and role dashboards	jsPDF, Role-based Auth Logic
5	Database	Stores user, course, quiz, and progress data	MongoDB (NoSQL)
6	Cloud Database	Cloud-hosted NoSQL for scalability and availability	MongoDB Atlas
7	File Storage	Stores certificates and profile images	Cloudinary or Local Filesystem
8	External API-1	For payment gateway and invoices	Stripe API
9	External API-2	Compiler integration for coding practice	Judge0 API
10	Machine Learning Model	Not Applicable in current version	N/A
11	Infrastructure	Cloud deployment with CI/CD	Vercel (Frontend), Render (Backend), GitHub Actions

S.No	Characteristics	Description	Technology Used
1	Open-Source Frameworks	React.js, Express.js, Node.js, Tailwind CSS, jsPDF	React.js, Node.js, Tailwind CSS
2	Security Implementations	JWT Auth, HTTPS, Email Verification, Role-based Access Control	JSON Web Token (JWT), HTTPS, bcrypt, CORS
3	Scalable Architecture	Tiered architecture with separation of concerns (Frontend, Backend, Database)	MERN Stack, Microservices approach (modular APIs)
4	Availability	Cloud deployment with high uptime and fault tolerance	Vercel, Render, GitHub Actions
5	Performance	Optimized with lazy loading, CDN, efficient routing, API caching	React Lazy, Lighthouse optimization, Render CDN

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>