

Project Design Phase-II

Technology Stack (Architecture & Stack)

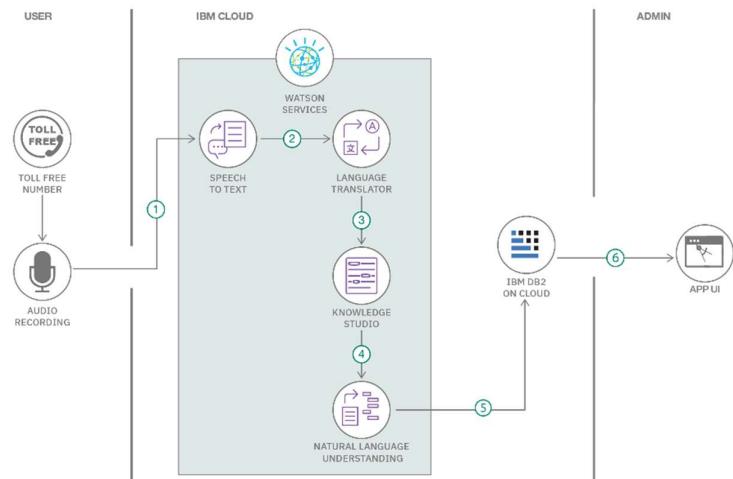
Date	25-06-2025
Team ID	LTVIP2025TMID45942
Project Name	Learn-Hub
Maximum Marks	4 Marks

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)

Architectural Diagram

I'll describe the layout for now. If you want a visual diagram, I can generate one after this.

High-Level Architecture Flow:

[User (Browser)]



[Frontend (React JS)]



[Backend (Express.js + Node.js)]



[MongoDB Database] + [File Storage (local/cloud)]



[Cloud Hosting / Local Server]

 Table-1: Components & Technologies

S.No	Component	Description	Technology
1	User Interface	Web interface for teachers, students, and admins	HTML, CSS, JavaScript, React.js
2	Application Logic-1	Course creation, video upload, enrollment logic	Node.js, Express.js
3	Application Logic-2	User authentication and role-based access	JWT (JSON Web Tokens), bcrypt.js
4	Application Logic-3	Admin operations and analytics	Express.js
5	Database	Stores user, course, progress, payment data	MongoDB (NoSQL)

S.No	Component	Description	Technology
6	Cloud Database	Optionally deploy MongoDB on cloud (e.g., MongoDB Atlas)	MongoDB Atlas
7	File Storage	Stores uploaded course content (videos, images)	Local file system or Cloudinary/AWS S3
8	External API-1	Payment integration	Razorpay API (or Stripe/PayPal)
9	External API-2	(Optional) Email OTPs / notifications	Nodemailer, SendGrid
10	Machine Learning Model (Future scope)	Quiz scoring, student progress prediction	Not yet integrated
11	Infrastructure	Deployment on local / cloud platform	Localhost, Render, Vercel, Railway

 **Table-2: Application Characteristics**

S.No	Characteristics	Description	Technology
1	Open-Source Frameworks	Frontend and backend frameworks used are open-source	React.js, Node.js, Express.js, MongoDB
2	Security Implementations	User authentication, role-based access, hashed passwords	JWT, bcrypt, HTTPS, CORS, helmet.js
3	Scalable Architecture	Modular MERN stack structure, can be split into microservices later	3-tier architecture, MVC
4	Availability	Cloud-deployable backend and frontend, fault-tolerant with cloud infra	Render/Vercel + MongoDB Atlas
5	Performance	Lightweight REST APIs, caching support, file compression possible	Axios, React Lazy Loading, Node Caching
