

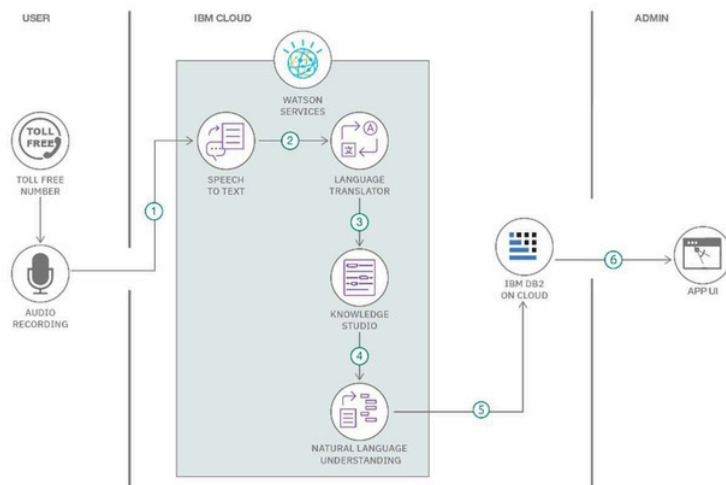
Project Design Phase-II  
Technology Stack (Architecture & Stack)

Date	June 2025
Team ID	LTVIP2025TMID56132
Project Name	ResolveNow: Your Platform for Online Complaints
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.) Table-1 :

Components & Technologies:

Indicate Data Storage components / services

Indicate interface to machine learning models (if

Table-2: Application Characteristics: applicable)

S.No	Component	User Interface	Description	Technology
1 2 3	Application Logic-1	Application	How user interacts with the application (e.g., Web UI)	React.js Node.js, Express.js Node.js,
4 5 6	Logic-2	Application Logic-3	User registration and login process	Express.js Socket.io (optional),
7 8 9	Database Cloud	Database File	Complaint submission and tracking	Express.js MongoDB MongoDB Atlas
10	Storage	External API-1	Real-time chat between user and agent	MongoDB GridFS / Base64 (optional)
11	External	API-2 Machine	Stores complaint, user, agent data	Nodemailer Twilio / Firebase OTP (N/A)
	Learning Model	Infrastructure	Cloud-hosted scalable database	Node.js server, localhost
	(Server / Cloud)		Store user-uploaded complaint images	
			Email service for notifications	
			OTP verification (if implemented)	
			(Not used currently, optional)	
			Local development environment	

S.No	Characteristics	Description	Technology Used
1	Open-Source Frameworks	Use of free, community-maintained frameworks	React.js, Node.js, Express.js
2	Security Implementations	Password hashing, secure Mongo URI, role-based access	bcrypt.js, dotenv
3	Scalable Architecture	Modular frontend/backend, MongoDB Atlas handles scaling	Express.js, MongoDB Atlas
4	Availability	Can be deployed on cloud servers or local servers for 24/7 availability	MongoDB Atlas, Vercel (optional)
5	Performance	REST APIs, optimized frontend, lazy loading	Axios, React.js, Express.js

## 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>