

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

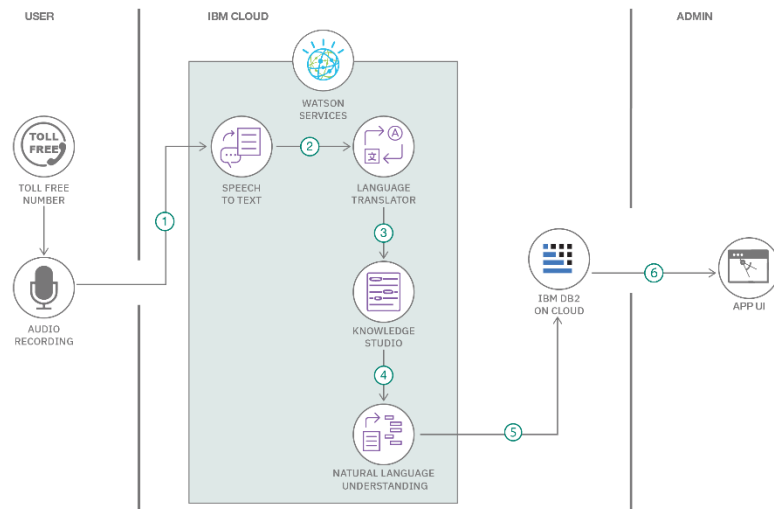
Date	16 June 2025
Team ID	LTVIP2025TMID59338a
Project Name	orderonthego: your on-demand food ordering solution
Maximum Marks	4 Marks

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	Component	Description	Technology
------	-----------	-------------	------------

1.	User Interface	Web UI for user interaction	HTML, CSS, React.js, Tailwind CSS.
2.	Application Logic-1	Customer-side logic (e.g., menu browsing, cart, orders)	JavaScript (React.js), Axios
3.	Application Logic-2	Backend logic for authentication and food operations	Node.js, Express.js
4.	Application Logic-3	Admin & Restaurant logic (approvals, menu control, etc.)	Node.js, Express.js
5.	Database	Stores all app data	MongoDB (Mongoose ODM)
6.	Cloud Database	Hosted on cloud	MongoDB Atlas
7.	File Storage	Image hosting (restaurant images, product images)	Cloudinary (or local for dev)
8.	External API-1	Weather or geo-location API (optional for extensions)	Not used currently / OpenWeather API
9.	External API-2	Payment gateway (for future use)	<i>Razorpay / Stripe</i> (future integration)
10.	Machine Learning Model	Not implemented in CraveKart Foods	N/A
11.	Infrastructure (Server / Cloud)	Server deployment	Render / Railway (for hosting) or Vercel (frontend)

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Frameworks used in client & server	React.js, Express.js, MongoDB, Node.js
2.	Security Implementations	JWT auth, bcrypt hashing, CORS, rate limits.	bcrypt, JWT, CORS
3.	Scalable Architecture	Separation of client/server, RESTful APIs, MongoDB Atlas for scaling	MERN Stack with REST APIs
4.	Availability	Deployed on cloud platforms with high uptime (Vercel/Render/Railway)	Cloud hosting + MongoDB Atlas

S.No	Characteristics	Description	Technology
5.	Performance	React-based UI with lazy loading, API pagination, CDN usage (Vercel/CDN)	React, Axios, MongoDB Indexes

References:

<https://www.mongodb.com/mern-stack>

<https://www.mongodb.com/atlas/database>

<https://expressjs.com/>

<https://reactjs.org/>

<https://jwt.io/introduction/>