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

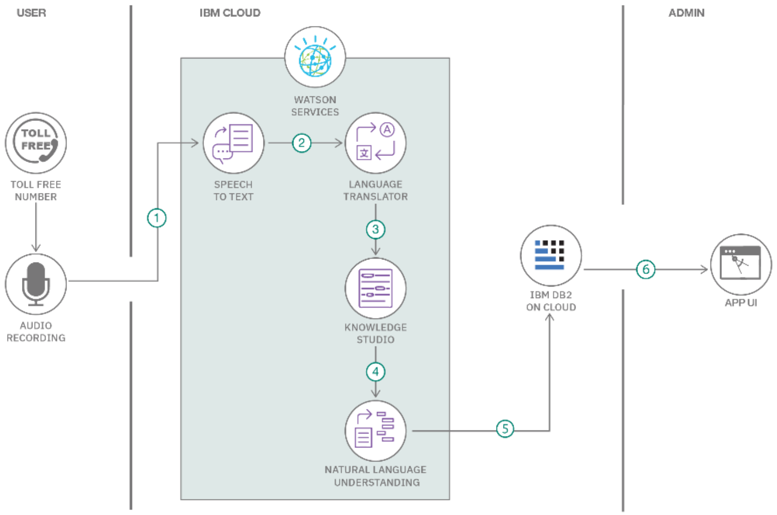
Date	31 January 3035
Team ID	LTVIP2026TMIDS87694
Project Name	EV Battery Performance and Range Monitoring System
Maximum Marks	4 Marks

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)

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1	User Interface	Responsive web dashboard + mobile app for EV charging visualization	React.js + Tailwind CSS + Tableau JS API
2	Application Logic-1	User authentication & session management	Python Flask/Django + JWT
3	Application Logic-2	EV data processing & range calculations	Python Pandas + NumPy
4	Application Logic-3	Data visualization & chart rendering	Chart.js + Plotly
5	Database	User profiles, charging history, EV datasets	PostgreSQL
6	Cloud Database	Scalable EV analytics storage	AWS RDS PostgreSQL
7	File Storage	CSV dataset uploads (EV specs, charger locations)	AWS S3
8	External API-1	Real-time charger station availability	PlugShare API / ChargePoint API
9	External API-2	Live EV battery & range data	Tesla API / OBD-II Bluetooth
10	Machine Learning Model	Range prediction based on driving patterns	Scikit-learn Random Forest
11	Infrastructure	Cloud deployment	AWS EC2 + Docker + Kubernetes

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1	Open-Source Frameworks	Frontend dashboard, data viz, backend API	React.js, Flask, Pandas, Plotly
2	Security Implementations	JWT auth, HTTPS, SQL injection protection, role-based access	JWT, bcrypt, OWASP ZAP, Cloudflare WAF
3	Scalable Architecture	Microservices: auth service, data service, viz service	Docker Compose, Kubernetes
4	Availability	Multi-AZ deployment, auto-scaling, health checks	AWS ELB, Route 53
5	Performance	CDN caching, Redis cache, optimized DB queries	Cloudflare CDN, Redis, PostgreSQL indexes

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