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

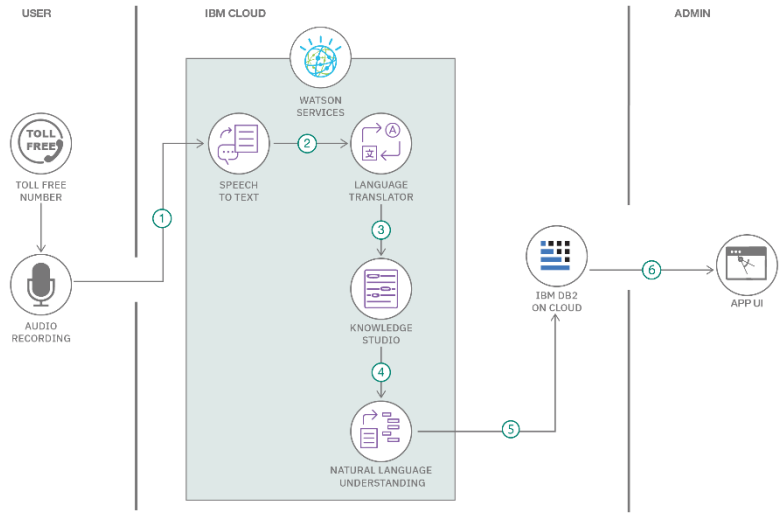
Date	31 May 3035
Team ID	LTVIP2025TMID43861
Project Name	Transfer Learning-Based Classification of Poultry Diseases for Enhanced Health Management
Maximum Marks	4 Marks

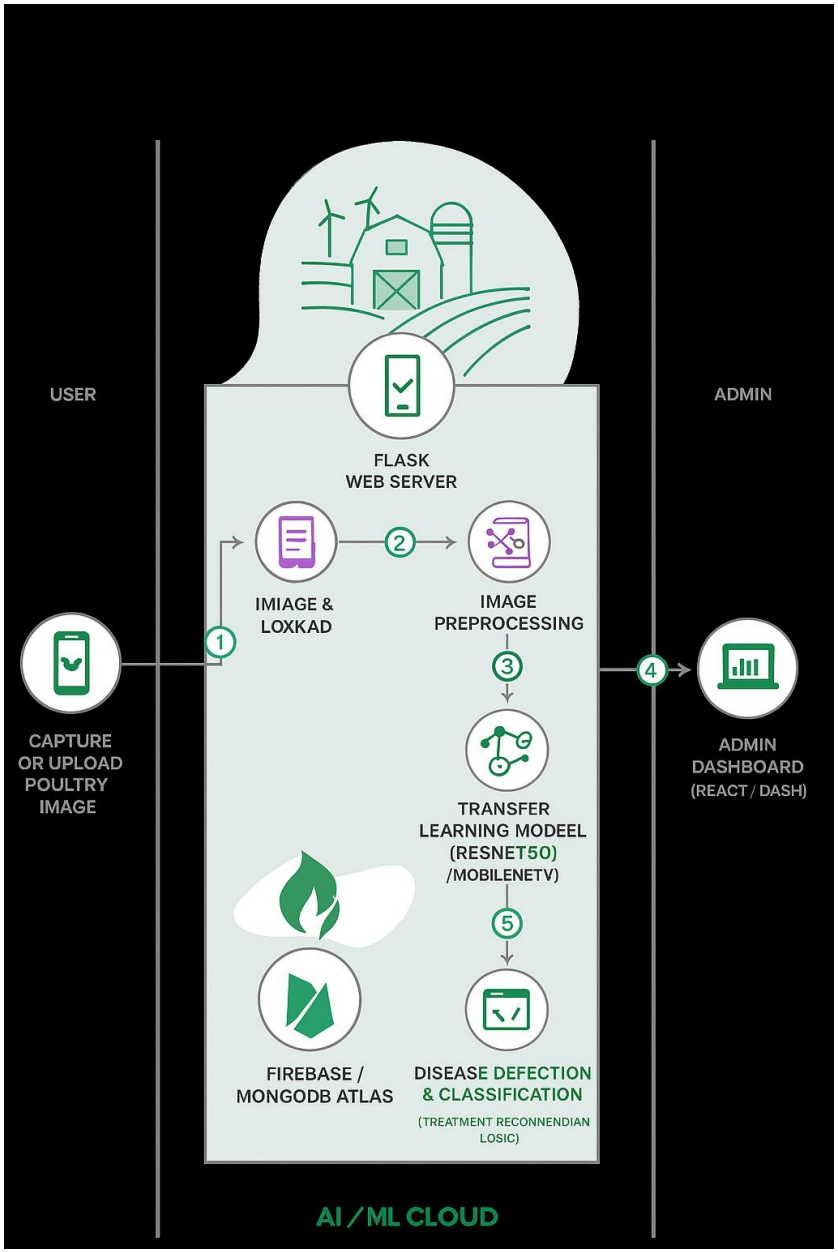
Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

Example:

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	Interface for farmers to upload images and view disease results (Mobile/Web App)	HTML, CSS, JavaScript, React.js, Bootstrap, Android (Kotlin/Java)
2.	Application Logic-1	Handles image upload, validation, and preprocessing	Python (Flask), JavaScript
3.	Application Logic-2	Logic to send image to ML model and retrieve prediction	Flask REST API / FastAPI
4.	Application Logic-3	Logic to handle notifications and database write operations	Python (Flask), Firebase Functions / Node.js
5.	Database	Stores user data, prediction results, and history	MySQL, NoSQL, etc.
6.	Cloud Database	Cloud-managed database solution	Firebase, AWS RDS, Google Cloud SQL
7.	File Storage	Storage for poultry images and model input data	Firebase Storage / AWS S3 / Google Cloud Storage
8.	External API-1	Send SMS/Email notification alerts to farmers	IBM Weather API, etc.
9.	External API-2	Weather API integration to provide suggestions based on humidity/heat (optional feature)	OpenWeatherMap API / Weatherstack API
10.	Machine Learning Model	Transfer learning model that classifies poultry diseases from images	TensorFlow/Keras using VGG16, ResNet50, or MobileNet.
11.	Infrastructure (Server / Cloud)	Deployment of app and model on cloud platform Google Cloud, Firebase Hosting, Heroku, AWS EC2	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Project uses open-source tools and libraries for web UI, machine learning, and backend logic.	React.js, Flask, TensorFlow, Keras, Scikit-learn, Firebase (Free tier)
2.	Security Implementations	Data is encrypted during transit. Secure login mechanisms used. No sensitive data stored on-device.	HTTPS, SHA-256, OAuth2.0 (Gmail login), Firebase Authentication, OWASP Top 10
3.	Scalable Architecture	Microservice-like deployment structure: frontend, backend API, ML model service. Can be deployed on scalable cloud platforms.	Flask REST API + Firebase / AWS Lambda + S3 + GCP Storage
4.	Availability	Hosted on highly available platforms. Uses cloud-based databases and storage, ensuring 24/7 uptime.	Firebase Hosting / Google Cloud Run / AWS EC2 with auto-scaling
5.	Performance	Model prediction is optimized for real-time results. Caching and CDN can be applied for fast content delivery. Image compression ensures speed.	CDN (Firebase), Model Optimization (e.g., MobileNet), LocalStorage, Caching

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