

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

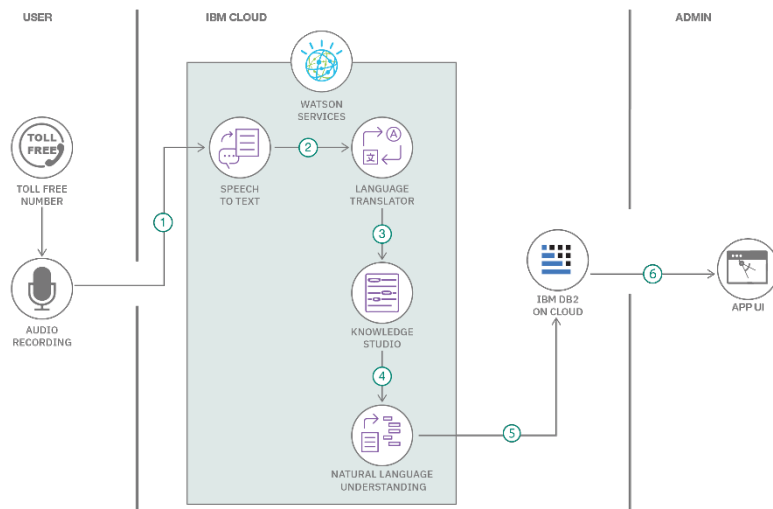
Date	June 2025
Team ID	LTVIP2025TMID33624
Project Name	Smart Sorting: Transfer Learning for Identifying Rotten Fruits and Vegetables
Maximum Marks	4 Marks

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	How the user interacts with the application: Web UI with multiple pages	HTML, CSS, JavaScript, Bootstrap, Flask-Jinja
2	Application Logic-1	Core image classification logic using ML model	Python (TensorFlow / Keras)
3	Application Logic-2	Image preprocessing and input handling	Python (OpenCV, Pillow)
4	Application Logic-3	Flask backend logic for routing and user input	Flask (Python Microframework)
5	Database	Stores user data, prediction history, and contact form submissions	SQLite / MongoDB
6	Cloud Database	Cloud-based storage for scalable deployment	Firebase Firestore / MongoDB Atlas
7	File Storage	Stores uploaded images for prediction and logging	Local filesystem / AWS S3 / Firebase Storage
8	External API-1	(Optional) Authentication or email service	Google Sign-In API / SendGrid API
9	External API-2	(Optional) Map or notification integration	Firebase Cloud Messaging (FCM)
10	Machine Learning Model	Identifies whether fruit/vegetable is fresh or rotten from images	Transfer Learning Model (VGG16 in Keras)

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1	Open-Source Frameworks	List of open-source frameworks used for web, backend, and ML components	Flask (Python), TensorFlow/Keras, Bootstrap, OpenCV
2	Security Implementations	Security mechanisms for data protection, access control, and safe communication	SHA-256 (password hashing), HTTPS, Google OAuth, Flask-Login, OWASP top 10 awareness
3	Scalable Architecture	Modular architecture allowing easy scaling of each component independently	3-tier architecture (UI, Logic, Database), Docker, REST APIs

4	Availability	Ensures app is available with minimal downtime using cloud deployment	Render/Heroku, GitHub CI/CD, Auto-deploy on push, Uptime Monitoring
5	Performance	Designed to handle multiple requests efficiently; caching and image optimization implemented	Flask Caching, Model loaded once on server, Optimized image preprocessing, CDN (optional)

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>