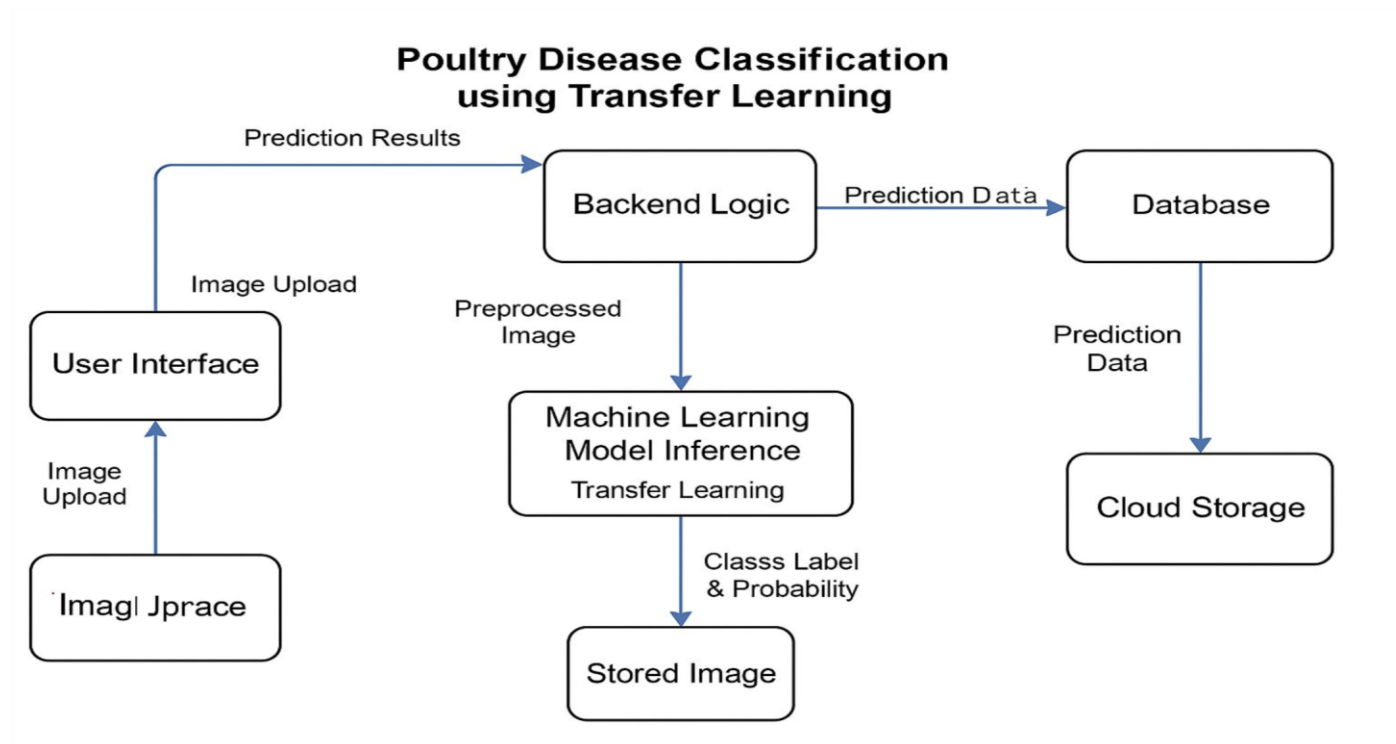


Project Design Phase-II Technology Stack (Architecture & Stack)

Date	30 June 2025
Team ID	LTVIP2025TMID37158
Project Name	Transfer Learning-Based Classification of Poultry Diseases for Enhanced Health Management
Maximum Marks	4 Marks

Technical Architecture:



S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	ML model training and inference	TensorFlow / Keras / PyTorch
2.	Security Implementations	Image upload security, access controls, API key restrictions	HTTPS, SHA-256, API Gateway

3.	Scalable Architecture	Microservice-based model deployment for easy scaling	Docker, Kubernetes (if used)
----	-----------------------	--	------------------------------

**Table-1 : Components & Technologies:**

S.No	Component	Description	Technology
1.	User Interface	Frontend for users to upload poultry images and view results	HTML, CSS, JavaScript
2.	Application Logic-1	Handles image preprocessing and routing requests	Python (Flask / Django)
3.	Application Logic-2	Disease prediction using trained CNN/Transfer Learning model	TensorFlow / PyTorch
4.	Application Logic-3	Model management and probability output parsing	Python
5.	Database	Stores user details, prediction history, and logs	MySQL / SQLite
6.	Cloud Database	Cloud storage of logs and results	Firebase / IBM Cloudant
7.	File Storage	Temporary storage of uploaded images	AWS S3 / Local File System
8.	External API-1	Optional: Weather data for poultry care advisory	OpenWeatherMap API
9.	External API-2	Optional: Disease news updates or vet services	News API / Vet API (if used)
10.	Machine Learning Model	Classifies poultry images into 4 disease categories	MobileNet / ResNet (via Transfer Learning)
11.	Infrastructure (Server / Cloud)	Deployment on cloud or local server	Localhost / AWS / IBM Cloud

**Table-2: Application Characteristics:**

S.No	Characteristics	Description	Technology
------	-----------------	-------------	------------

4.	Availability	Can be hosted on cloud with failover or load balancing	AWS EC2 / Cloud Load Balancer
5.	Performance	Uses image caching and lightweight models for fast inference	Redis (optional), CDN (optional)