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

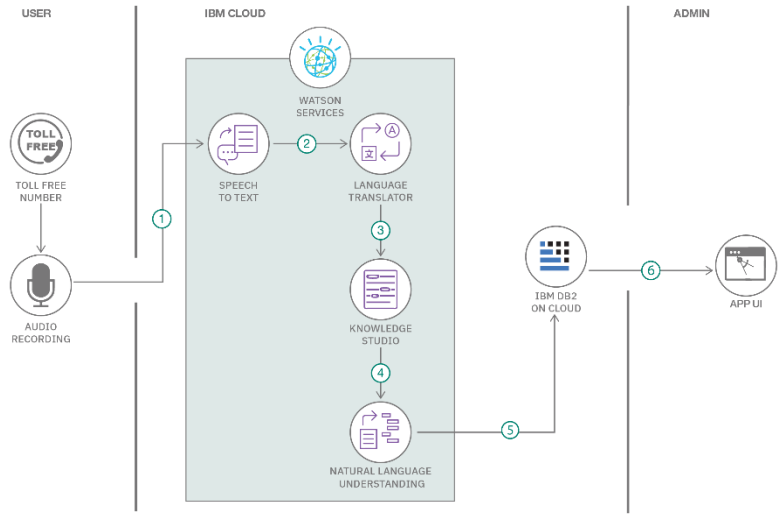
Date	27 June 2025
Team ID	LTVIP2025TMID38971
Project Name	pollen's profiling: automated classification of pollen grains
Maximum Marks	4 Marks

Technical Architecture:

The system follows a client-server architecture with a web/mobile frontend and a FastAPI backend.
A trained CNN model handles pollen image classification, integrated through the backend API.
User data, results, and logs are stored in a secure cloud database, with admin and analytics support.

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



Guidelines:

This project aims to build a system that classifies pollen grain images using a trained CNN model.
Users can register, log in, and upload images through a web or mobile interface.
The backend, built with FastAPI, handles image processing, prediction, and data storage.
The system includes an admin panel for user management, monitoring, and analytics.

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Web/Mobile interface for user interaction	HTML, CSS, JavaScript / ReactJS / Flutter
2.	Application Logic-1	Handles user registration, login, and image upload	Python (FastAPI)
3.	Application Logic-2	Processes and classifies pollen grain images	TensorFlow / Keras (CNN Model)
4.	Application Logic-3	Handles prediction result formatting and dashboard	Python (Backend Business Logic)
5.	Database	Stores user info, history, and prediction results	PostgreSQL / MongoDB
6.	Cloud Database	Cloud-hosted storage for scalable access	AWS RDS / MongoDB Atlas
7.	File Storage	Stores uploaded pollen grain images	AWS S3 / Local Filesystem
8.	External API-1	Email confirmation and notification service	SMTP / SendGrid API
9.	External API-2	Optional social login (OAuth)	Google OAuth / LinkedIn OAuth
10.	Machine Learning Model	Classifies pollen grain images into predefined types	Convolutional Neural Network (CNN)
11.	Infrastructure (Server / Cloud)	Deployment on cloud/local system	Local Server, Docker, Heroku / AWS EC2 / Kubernetes

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Backend and ML frameworks used are open-source	FastAPI, TensorFlow/Keras, ReactJS, PostgreSQL, Docker
2.	Security Implementations	Authentication, data encryption, secure access control	JWT Authentication, HTTPS, OAuth 2.0, SHA-256, OWASP guidelines

S.No	Characteristics	Description	Technology
3.	Scalable Architecture	Modular client-server architecture supports scale-out with containers	3-tier Architecture, Docker, Kubernetes (optional)
4.	Availability	Deployed on scalable infrastructure with high uptime	AWS EC2/Heroku, Load Balancer, Auto-scaling Groups
5.	Performance	Optimized API calls, image preprocessing, and model inference for quick response	FastAPI (async support), Redis Cache (optional), CDN (if used)