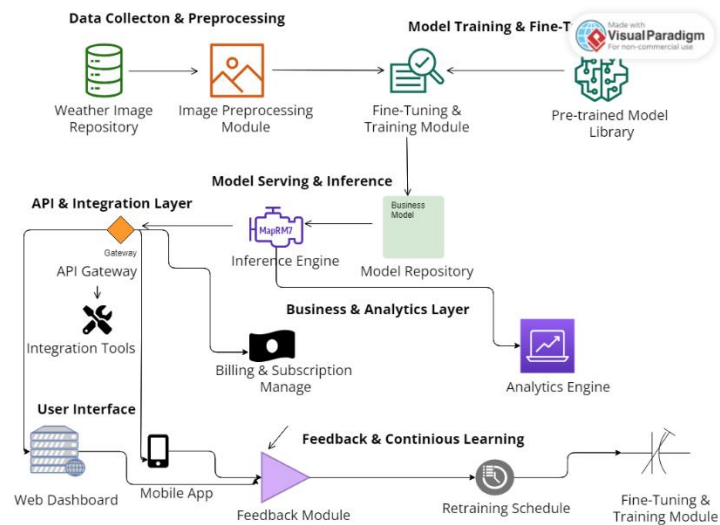


Project Design Phase-II Technology Stack (Architecture & Stack)

Date	23 October 2023
Team ID	Team-591965
Project Name	Weather Classification Using Deep Learning
Maximum Marks	4 Marks

Technical Architecture:

The Deliverable is based on the “Weather Classification System”



Guidelines:

- **Include all the processes (As an application logic / Technology Block)**

- User uploads an image for classification.

<ul style="list-style-type: none"> Image is classified based on weather conditions. User provides feedback on the classification. System refines classification based on feedback.
<ul style="list-style-type: none"> Provide infrastructural demarcation (Local / Cloud) <ul style="list-style-type: none"> Frontend/UI is cloud-hosted. Backend and ML model deployment on cloud infrastructure. Database and File storage also on the cloud.
<ul style="list-style-type: none"> Indicate external interfaces (third party API's etc.) <ul style="list-style-type: none"> Weather API (e.g., OpenWeatherMap) for external weather data. Flask API Endpoint for third-party developers.
<ul style="list-style-type: none"> Indicate Data Storage components / services <ul style="list-style-type: none"> MySQL or MongoDB for storing user data, feedback, and classifications. AWS S3 or Google Cloud Storage for uploaded weather images.
<ul style="list-style-type: none"> Indicate interface to machine learning models (if applicable) <ul style="list-style-type: none"> Transfer Learning Models (Inception V3, VGG19, Xception V3) for image classification.

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, etc.	HTML, CSS, JavaScript/ ReactJS
2.	Application Logic-1	Logic for uploading and classifying the image	Python, TensorFlow, Keras
3.	Application Logic-2	Logic for user feedback and refinement	Python, Flask API
4.	Application Logic-3	Logic for displaying weather classifications on the dashboard	JavaScript, ReactJS
5.	Database	Store user data, feedback, and classifications	MySQL, MongoDB
6.	Cloud Database	Database Service on Cloud	AWS RDS, Google Cloud SQL

7.	File Storage	Storage for uploaded weather images	AWS S3, Google Cloud Storage
8.	External API-1	Integrate with other meteorological systems	Weather API (e.g. OpenWeatherMap)
9.	External API-2	API for third-party developers	Flask API Endpoint
10.	Machine Learning Model	Weather classification model	Transfer Learning Models (Inception V3, VGG19, Xception V3)
11.	Infrastructure (Server/Cloud)	Application Deployment on Local System/Cloud	AWS EC2, Google Cloud Compute Engine

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	TensorFlow, Keras, Flask, ReactJS
2.	Security Implementations	Ensure safe image uploads, user data protection	SSL/TLS, Encryption at Rest, JWT for Authentication
3.	Scalable Architecture	Handle increasing numbers of users and data	Microservices with Docker, Kubernetes
S.No	Characteristics	Description	Technology
4.	Availability	Ensure uptime, balance loads	Load Balancers (e.g., AWS ELB), Multi-zone deployments
5.	Performance	Handle requests efficiently, quick model predictions	Caching with Redis, GPU-accelerated ML processing

