Project Design Phase
Solution Architecture

Date	19 February 2026
Team ID	LTVIP2026TMIDS75186
Project Name	Rising Waters: A Machine Learning Approach to Flood Prediction
Maximum Marks	4 Marks

Overview

The solution architecture connects the real-world problem (flood risk due to extreme rainfall and river overflow) with the technical implementation (data collection, preprocessing, machine learning modeling, and alert delivery).

The architecture defines system components, data flow, interfaces, and deployment strategy to ensure accurate and scalable flood prediction**.**

Key Goals

• Provide accurate early flood risk prediction
• Minimize false flood alarms while maintaining high detection rate
• Ensure modular structure (data layer, model layer, alert layer)
• Handle large-scale environmental and weather datasets
• Maintain reproducibility using saved models and preprocessing pipelines

Architecture Components

Frontend (UI):
Web dashboard for monitoring rainfall, river levels, and predicted flood risk.

Backend (API Layer):
Flask/FastAPI server for handling data inputs, prediction requests, and alert triggering.

Data Collection Layer:
Historical rainfall data, river water level data, temperature, humidity, and weather forecasts (from government datasets or APIs).

Preprocessing Layer:
Data cleaning, missing value handling, scaling, time-series feature engineering (rainfall intensity, water level growth rate, seasonal patterns).

ML Model Layer:
Trained flood prediction model (Logistic Regression / Random Forest / XGBoost / LSTM for time-series).

Database & Artifacts:
Storage of historical environmental data; saved trained models (pickle/joblib).

Alert System:
SMS/Email notification module for high-risk flood warnings.

Data Flow

1. Weather and river data are collected from datasets or APIs.

2. Backend receives and validates data inputs.
3. Data passes through preprocessing (cleaning, scaling, feature engineering).
4. Processed data is sent to ML model for flood risk prediction.
5. System classifies region as Low / Medium / High flood risk.
6. Results are stored in database and alerts are triggered if risk is high.
7. Dashboard displays real-time risk status and trend analysis.

Solution Architecture Diagram:

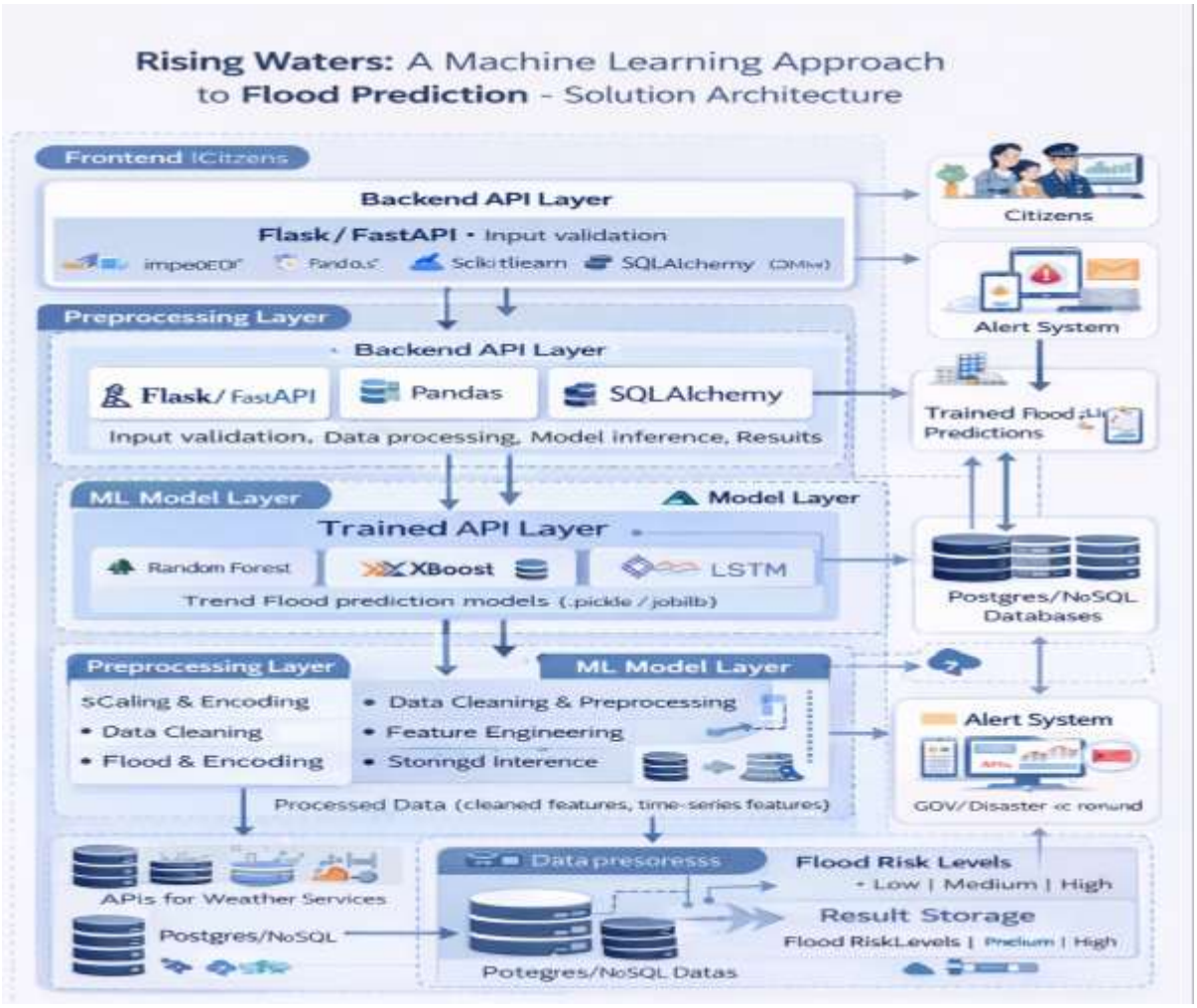

Figure 1: Architecture and data flow of the voice patient diary sample application

Non-Functional Considerations

Performance:
Real-time prediction with low latency (< few seconds per request).

Reliability:
Consistent preprocessing using saved scalers and encoders.

Security:
Secure API endpoints and protected access to dashboard.

Maintainability:
Modular architecture separating data, model, and alert systems.

Scalability:
Cloud-ready design supporting multiple districts and large datasets.

Deployability:
Compatible with Docker, cloud servers, and government monitoring systems.