Project Design Phase

3.3 Solution Architecture

Date	23 june 2025
Team ID	LTVIP2025TMID41942
Project Name	Traffic telligence : advanced traffic volume estimation with machine learning
Maximum Marks	4 Marks

Solution Architecture – TrafficTelligence

---

1. Overview

The solution architecture of TrafficTelligence follows a **three-layered structure** that connects the **user interface**, **backend logic**, and the **machine learning model** for real-time traffic volume prediction.

---

2. Architectural Layers

A. Presentation Layer (Frontend)

- **Built with:** HTML, CSS (or React optionally)
- **Role:** Takes user input (temperature, rain, snow, clouds)
- **Interaction:** Submits form data to the backend via POST request

---

B. Application Layer (Backend – Flask)

- **Built with:** Python + Flask
- **Functions:**
  - Receives user input from frontend
  - Loads the trained model (traffic_model.pkl)
  - Prepares the input and calls the model to make predictions
  - Returns the predicted traffic volume to the frontend

C. Model Layer (ML Engine)

- **Built with:** scikit-learn (Random Forest Regressor)
- **Steps:**
  - Preprocesses dataset (traffic_volume.csv)
  - Trains and evaluates the model

- Saves the model using pickle for deployment

✅ (Optional) D. Data Layer (Future Enhancement)

- **Possible Tools:** MongoDB / MySQL
- **Use:** To store user inputs, prediction logs, analytics
- **Status:** Not implemented yet, but planned in future versions

---

🔁 3. Workflow Summary

pgsql

CopyEdit

User → (Frontend Form)

→ Flask Backend (/predict)

→ Load ML Model

→ Predict Traffic Volume

→ Return Result to UI

🧰 4. Technologies Used

- **Frontend:** HTML, CSS
- **Backend:** Python, Flask
- **ML Model:** Random Forest Regressor (scikit-learn)
- **Data Handling:** pandas, numpy
- **Deployment Option:** Localhost (can be extended to Render, Heroku, etc.)

✅ Benefits of the Architecture

- Modular and easy to extend (e.g., switch ML models or frontends)
- Lightweight and fast for real-time prediction
- Compatible with full-stack upgrades (React + Node.js, etc.)

Example - Solution Architecture Diagram: