Artificial Intelligence

&

Machine Learning

PROJECT REPORT

CONTENTS:

TrafficTelligence: Advanced Traffic Volume Estimation with Machine Learning

1.INTRODUCTION

1.1 Project Overview

TrafficTelligence: Advanced Traffic Volume Estimation with Machine Learning is a predictive web-based application designed to estimate traffic volume based on various inputs such as weather conditions, time, date, and holiday information. The core objective is to empower commuters, traffic authorities, and city planners with timely insights into traffic congestion, helping them make informed travel and infrastructure decisions. By utilizing a trained machine learning model integrated with a Flask backend and a user-friendly HTML interface, the system provides quick and accurate traffic volume predictions, ultimately contributing to reduced travel delays, optimized routes, and smarter urban mobility planning.

1.2 Purpose

The purpose of the *TrafficTelligence* project is to develop an intelligent system that accurately predicts traffic volume using machine learning techniques based on environmental and temporal factors.

- ☑ Predict Traffic Volume:
  To estimate traffic volume using machine learning based on inputs like weather, date, time, and holidays.
- 🚗 Assist Commuters in Planning:
  Help users choose better travel routes and avoid high-traffic areas through predictive insights.
- 🧠 Enable Smart City Solutions:
  Support traffic authorities in managing congestion and improving infrastructure using data-driven planning.
- 💰 Reduce Hardware Dependency:
  Eliminate the need for costly physical sensors or cameras by using existing datasets and user inputs.
- 🗺 Minimize Travel Time and Fuel Usage:
  Reduce delays and fuel consumption by providing traffic forecasts before a journey starts.
- 📊 Promote Data-Driven Decision Making:
  Utilize machine learning models to offer reliable and scalable predictions, enhancing transportation efficiency.

2.IDEATION PHASE

Date: 28 June 2025

Team ID: LTVIP2025TMID40870

Project Name: *TrafficTelligence: Advanced Traffic Volume Estimation with Machine Learning*

Maximum Marks: 2 Marks

2.1 Problem statement

Urban traffic congestion is a growing challenge, impacting commute times, fuel consumption, and city planning efficiency. Traditional traffic monitoring methods often rely on expensive sensors or manual observation. TrafficTelligence aims to revolutionize traffic volume estimation by leveraging machine learning to analyze real-time video feeds and aerial imagery This intelligent system can automatically detect, track, and quantify vehicle flow on highways and city roads, offering scalable, cost-effective, and data-driven insights for smart city infrastructure. **Example: Traffic Telligence**

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	A regular commuter	Reach my Destination on time	I don't know the traffic in advance	I can't plan the best route	Frustrated and delayed
PS-2	A traffic department staff	Predict road congestion levels	Manual tracking is inefficient	Traffic patterns change frequently	Helpless and overloaded

2.2 Empathy Map Phase

Date: 28 June 2025

Team ID: LTVIP2025TMID40870

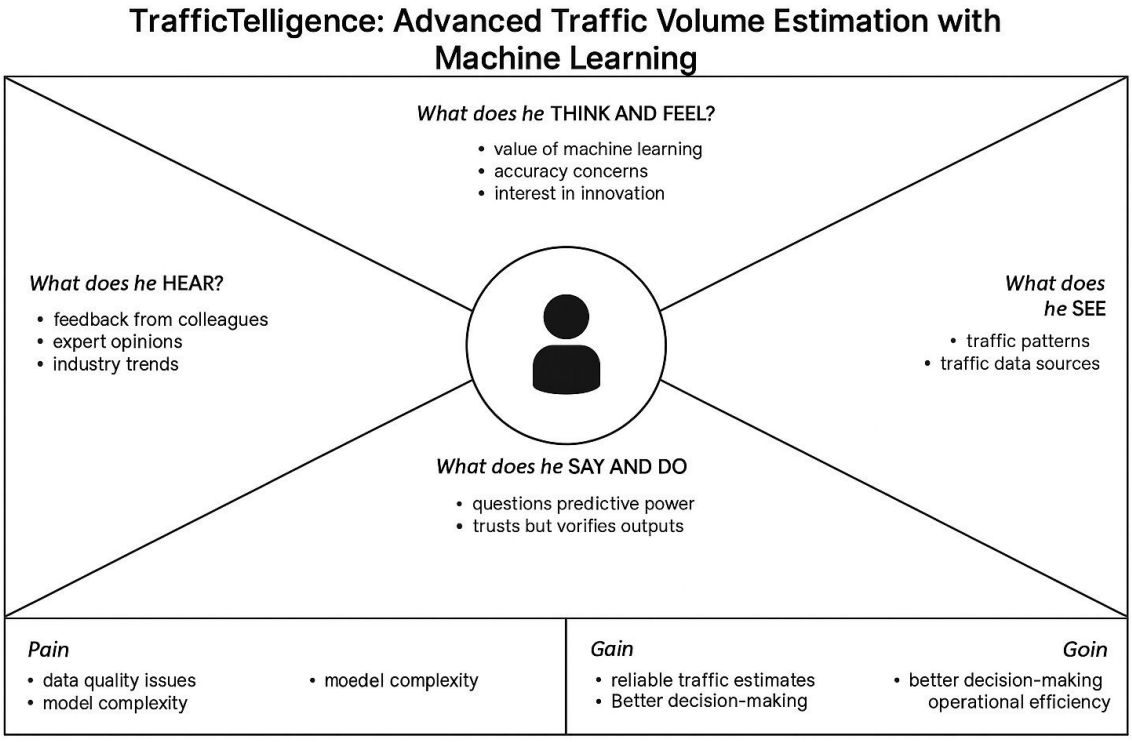
Project Name: *TrafficTelligence: Advanced Traffic Volume Estimation with Machine Learning*

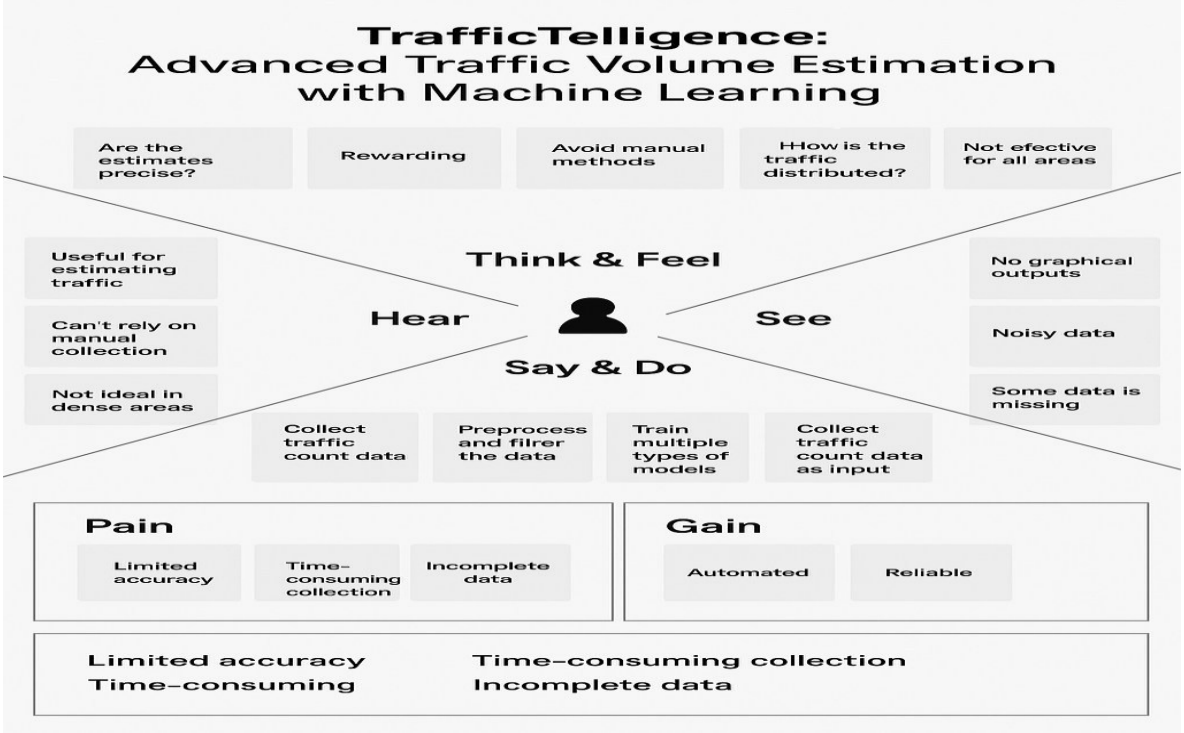
Maximum Marks: 4 Marks

An Empathy Map helps us step into the shoes of our target users—commuters, traffic authorities, and city planners—so we can understand their pain points, needs, and goals related to traffic congestion and travel planning.

Creating this empathy map was an essential step in shaping a solution that is not only functional but genuinely user-centered. By exploring what users think, feel, see, say, do, and hear, we ensured that *TrafficTelligence* addresses real challenges with practical, intuitive outcomes.

Example:





2.3 Brainstorming

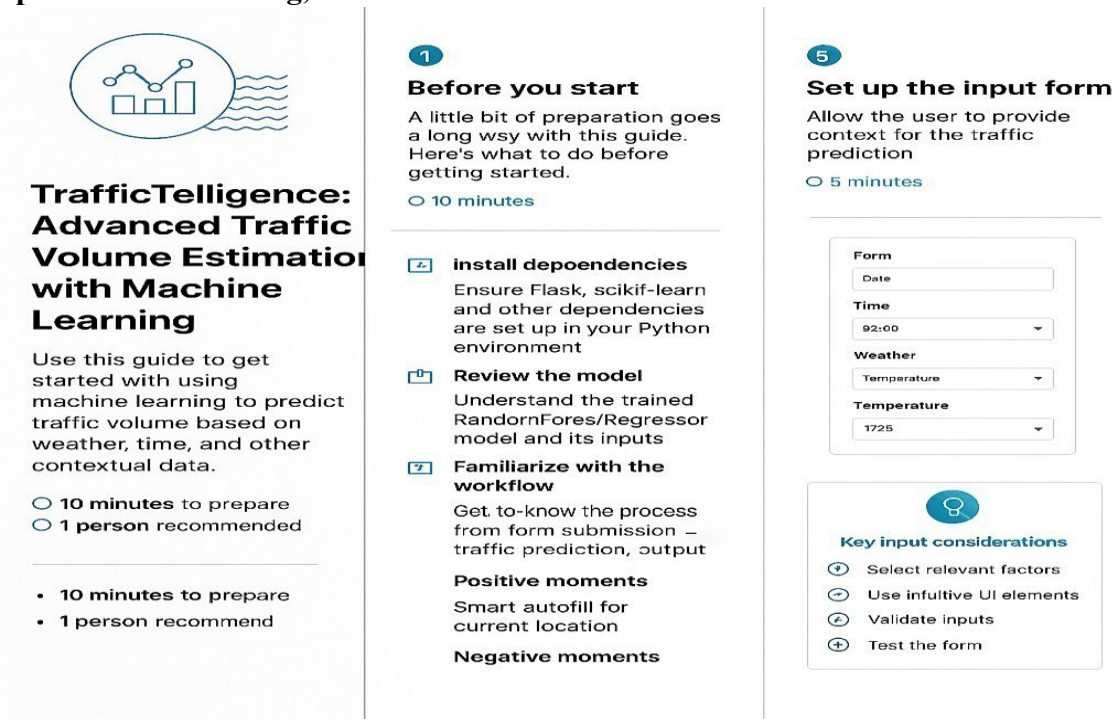
Date: 28 June 2025
Team ID: LTVIP2025TMID40870
Project Name: *TrafficTelligence: Advanced Traffic Volume Estimation with Machine Learning*
Maximum Marks: 4 Marks

Brainstorm & Idea Prioritization in Traffic Telligence:

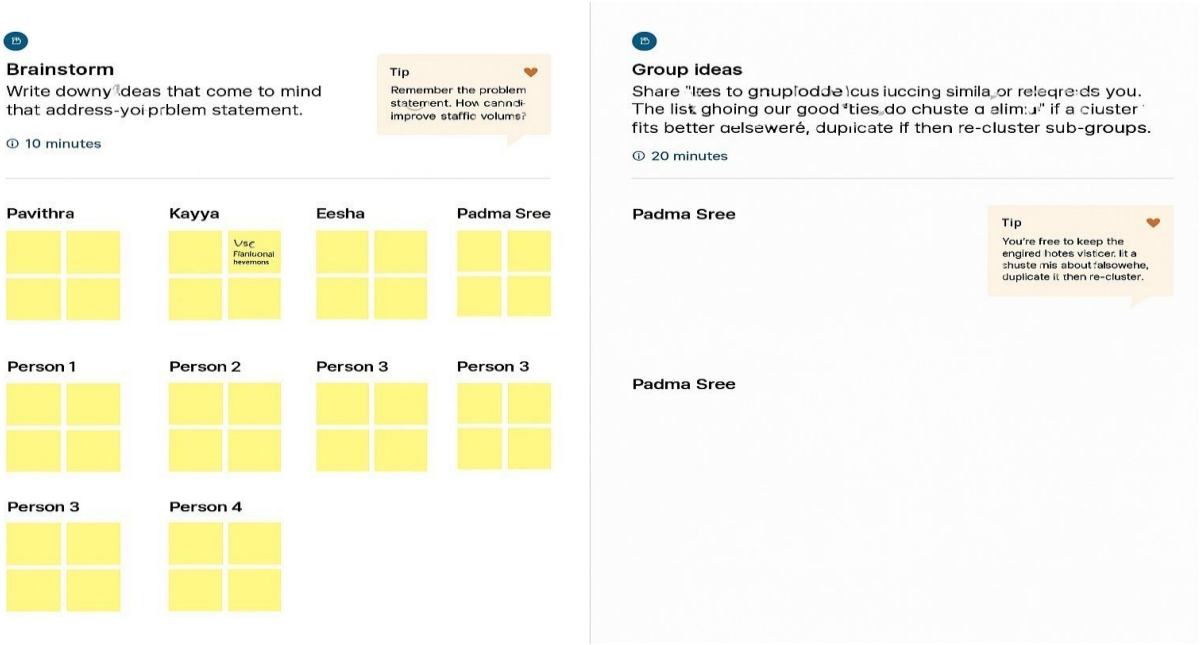
We initially brainstormed various ideas such as using live traffic sensors, integrating Google Maps data, and applying machine learning models to predict traffic. Each idea was evaluated based on feasibility, cost, complexity, and data availability. Sensor-based systems were found to be expensive and difficult to deploy. On the other hand, historical data was readily available and easy to work with.

After discussion, we prioritized the machine learning approach using historical weather and traffic data. It was easier to implement, cost-effective, and suitable for a working prototype. This method allowed us to focus on core concepts like preprocessing, model training, and deployment. It also made the project achievable within our internship timeline.

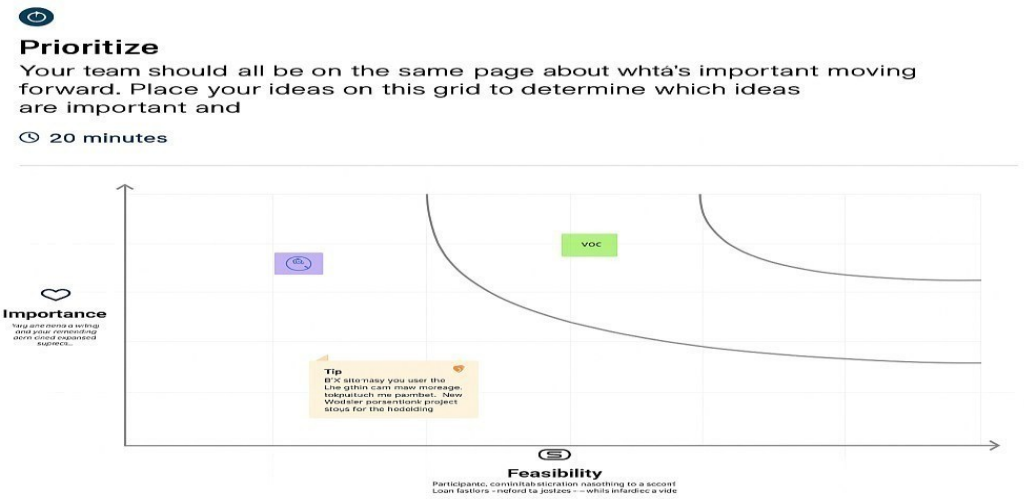
Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea Listing and Grouping



Step-3: Idea Prioritization



3.

3. REQUIREMENT ANALYSIS 3.1 Customer Journey Map

TrafficTelligence: User Journey Map

Scenario: Submitting traffic context data, receiving a prediction, and re-engaging based on useful insights

Entice	Enter	Engage	Exit	Extend
"Help me discover a smarter way to estimate traffic before I start my trip."	"Help me give the app the right context with minimal effort."	"Help me get an accurate prediction and feel confident about what I see '	"Help me wrap up with clarity und usefulness.	"Help me keep getting smarter insights the more I use this."
• Steps: User sees a prome or fink to Traffic telligence (via blog-travel or navigation app • Interactions: Website landing page / search engine result smpeet integrated tool in nnother site • Positive Clean UL..Jold hold claim "ML-pewered	• Steps: Reach ing Input paged selects. weather temperature data, time, holiday, etcs • Interactions: HTML form with-dropdowns, toggies, calendar and sliders • Posple/Places: Typically done sole on a computer or mobile browser • Positive	- Steps: Form-data sent to Flask backend - Inputs preprocessed (type conversion, encealing) - ML model. predicts traffic volume • interactions, Backend AFI (Flask). Randomforest Regressor modei • Styled result	• Steps: User ressults and closes ap • Optional rating prompt or teedback request • Option to download) or share-resutt • Positive Moments: Feeling of control over traffic plans • Sense of Insight • Pain Points; No export/share quitan	• Steps: Personalized suggestions· "Try again at a different houre" • Past predictions saved in proffle (if legged in) • Follow-up email "Did this estimate match your real experience?" • Opportunities: ML model learns from user feedback • Users return

3.2 Solution Requirement

Date: 28 June 2025
Team ID: LTVIP2025TMID40870
Project Name: *TrafficTelligence: Advanced Traffic Volume Estimation with Machine Learning*

Maximum Marks: 4 Marks **Functional requirements:**

Following functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Input Interface	- Input form to collect date, time, weather, and holiday data - Input validation and required field checks
FR-2	Prediction Engine	- Backend receives and processes input data - Model loads from `model.pkl` and predicts traffic volume
FR-3	Result Display	- Output page renders predicted volume with styled background - Provide option to re-enter or revise input
FR-4	Static Asset Handling	- Background images for UI (`bg1.jpg`, `bg2.jpg`) loaded from static - Image formatting and visual enhancements
FR-5	Deployment	- Flask server to run the app locally - Option to deploy on cloud platforms (Render/PythonAnywhere)
FR-6	Data Handling	- Preprocessing includes encoding, missing value imputation

Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The application should have a simple and intuitive interface for users to input data and view predictions.
NFR-2	**Security**	The system must be protected against malicious inputs

NFR-3	**Reliability**	The model and web interface should consistently return accurate predictions without system crashes or interruptions.
NFR-4	**Performance**	The system should deliver traffic predictions within 2–3 seconds of user input submission.
NFR-5	**Availability**	The web app should be accessible at all times, especially during high traffic periods.
NFR-6	**Scalability**	The solution should support future upgrades such as real-time GPS data integration, cloud deployment, and mobile app expansion.

3.3 Data Flow Diagram

Date: 28 June 2025

Team ID: LTVIP2025TMID40870

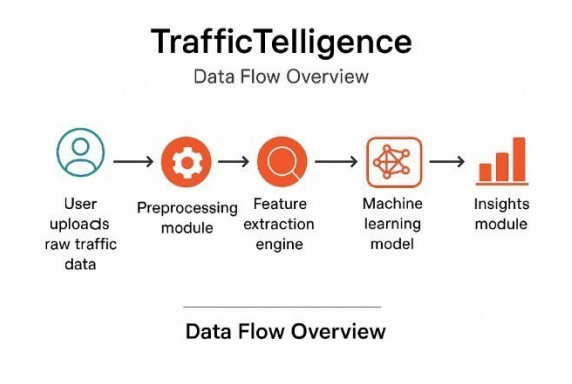
Project Name: *TrafficTelligence: Advanced Traffic Volume Estimation with Machine Learning*

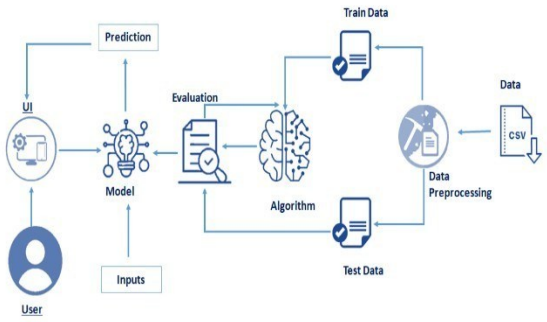
Maximum Marks: 4 Marks

Data Flow Diagram (DFD)Overview:

TrafficTelligence is an innovative platform designed to estimate traffic volume using advanced machine learning algorithms. By streamlining the flow of data from raw input sources—such as sensor logs, video feeds, or geolocation files—through preprocessing, feature extraction, and predictive modeling, it delivers accurate and actionable insights for urban planners and traffic engineers. The system not only enhances forecasting precision but also supports smart city initiatives by transforming complex data into intuitive visual reports.

Example: DFD Level 0 (Industry Standard)





User Stories – Traffic Telligence

This document lists the user stories defined for the TrafficTelligence web-based traffic volume prediction system. It includes user types, functional requirements, story IDs, expected outcomes, priorities, and release sprint versions.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance Criteria	Priority	Release
Customer (Web User)	Input Form	USN-1	As a user, I can input trafficrelated details (time, weather, holiday) into the form.	I can submit the form without errors	High	Sprint-1
Customer (Web User)	Model Integration	USN-2	As a user, I receive predicted traffic volume instantly after form submission.	I can see predicted traffic volume on the output page	High	Sprint-2
Customer (Web User)	User Interface	USN-3	As a user, I can view the app with a background image and styled layout.	UI loads with background image correctly	Medium	Sprint-1

Administrator	Data Validation	USN-4	As an admin, I can validate if the ML model accepts only valid data types.	Model throws errors for wrong inputs	Medium	Sprint-2
Customer Care Executive	Error Handling	USN-5	As support staff, I can verify error messages shown for missing or invalid inputs.	App shows descriptive error messages	Medium	Sprint-2
Customer (Web User)	Flask Deployment	USN-6	As a user, I can access the prediction system online through a browser.	App loads via localhost or deployment link	High	Sprint-2

3.4 Technology Stack

Date: 28 June 2025
Team ID: LTVIP2025TMID40870
Project Name: *TrafficTelligence: Advanced Traffic Volume Estimation with Machine Learning*
Maximum Marks: 4 Marks

Table-1: Components & Technologies:

S. No	Component	Description	Technology
1.	User Interface	Interface for users to upload data and view predictions	HTML
2.	Application Logic-1	Core processing logic: data validation, user flow	Python (Flask/Django)
3.	Application Logic-2	Video-to-data conversion for traffic analytics	OpenCV
4.	Application Logic-3	Predictive modeling and analytics	Scikit-learn, XGBoost
5.	Database	Store user inputs, model outputs, historical data	PostgreSQL, MongoDB
6.	Cloud Database	Cloud-hosted solution for scalability and reliability	Amazon RDS, Firebase Firestore
7.	File Storage	Store raw video/data files and result exports	AWS S3, Google Cloud Storage
8.	External API-1	Real-time weather data to adjust traffic predictions	OpenWeatherMap API
9.	External API-2	Location validation or geocoding	Google Maps API, Mapbox
10	Machine Learning Model	Predict traffic volume based on features extracted	Random Forest, Linear regression, D tree.
11	Infrastructure (Server / Cloud)	Hosting the application, serving requests, model inference	Web Apps

Table-2: Application Characteristics:

S. No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Uses open and free tools for development	Flask (Python), Scikitlearn, Pandas, HTML, CSS
2.	Security Implementations	Input validation and basic protecti from bad data inputs	n Flask validation, OWASP guidelines (basic)
3.	Scalable Architecture	Follows modular and expandable design	Technology used
4.	Availability	Can be hosted on platforms with high uptime	Technology used
5.	Performance	Fast model response and low memory usage	Technology used

4.PROJECT DESIGN

4.1 Project Solution Fit

Date: 28 June 2025
Team ID: LTVIP2025TMID40870
Project Name: *TrafficTelligence: Advanced Traffic Volume Estimation with Machine Learning*
Maximum Marks: 2 Marks

Problem – Solution Fit Template:

The core problem faced by commuters and traffic authorities is the unpredictability of traffic, which leads to delays, frustration, and inefficient road usage. Traditional methods are often reactive and lack predictive capabilities. *TrafficTelligence* addresses this issue by providing a machine learning-powered web application that predicts traffic volume based on time and weather inputs. This solution fits naturally into user behavior, requires no expensive infrastructure, and offers data-driven insights. By aligning with user needs and daily routines, it effectively solves a real and recurring problem with a practical and scalable approach.

Purpose:

- ❑ Solve real-world traffic problems by providing predictive insights into vehicle volume using machine learning models tailored to user behavior and data patterns.
- ❑ Enhance user adoption and satisfaction by integrating a simple, accessible web interface that aligns with daily commuter routines and decision-making processes.
- ❑ Improve communication and planning by delivering accurate traffic predictions that reduce uncertainty and help users make informed travel choices.
- ❑ Build trust and engagement by solving frequently encountered traffic issues through a reliable, data-driven system that minimizes delays and stress.
- ❑ Understand and improve existing infrastructure by analyzing historical traffic and weather data to support better road management strategies and smarter urban mobility solutions

Template:

TrafficTelligence: Advanced Traffic Volume Estimation with Machine learnning Learrning

1. CUSTOMER SEGMENTIS) Orgamizations racing problem related traffic volume estimation	**2. JOBS-TO-BE-DONE / PROBLEMS** 2 What challenge current traffic volum-estimation does it poss customstioers? *(e.g. manual traffic counts, lim*	**3. AVAILABLE SOLUTIONS** What solution(s) or aproachs ca used to caddress this problem e.g. manual traffic
4. CUSTOMER CONSTRAINTS CC Factors areventing customers from addressing the problem (e.g. manual data	**5. PROBLEM POOT CAUSE** RC Whats nokes the problem comples/or difficult to sorve 'vegertaol storrie *(e.g. huclcating traffic pottems*, diverce geographica areas	**6. BEHAVIOUR** What custome as doing o/ difficult to solve address this problem and gain *customer buyin* (e.g. analyzing sensors
7. TRIGGERS TR How does or circomstances, many that may trigger customers to seek a better traffic volume estimation? (e.g. new *regulatyly requirements/ increased costs associated with zongestio*	**8. YOUR SOLUTION** SL How does your macthine learning solution better address your current solution (e.g. *enhanzing troffic volumes hosed historical atid* and *other features, enabling real-time prodictions*	**9. CHANNELS & BEHAVIOUR** EM How do customers use TrafficTelligence is delivered to customers? (B. x,fere
10. TRIGGERS EM How can customerrs use. Traffic Telligence a better address free in sporation? (e.g. free trials dondronstrations		**10. OPT-IN** 10 How do customers use TrafficTelligence if or how converted into tree and paying customers? (Free trials and demonstrato-tions ealily

4.2 Proposed Solution Architecture

Date: 28 June 2025
Team ID: LTVIP2025TMID40870
Project Name: *TrafficTelligence: Advanced Traffic Volume Estimation with Machine Learning*
Maximum Marks: 2 Marks

Project team shall fill the following information in the proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Urban traffic congestion is a growing challenge, impacting commute times, fuel consumption, and city planning efficiency. Traditional traffic monitoring methods often rely on expensive sensors or manual observation. TrafficTelligence aims to revolutionize traffic volume estimation by leveraging machine learning to analyze real-time video feeds and aerial imagery. This intelligent system can automatically detect, track, and quantify vehicle flow on highways and city roads, offering scalable, cost-effective, and data-driven insights for smart city infrastructure.
2.	Idea / Solution description	TrafficTelligence is a web-based platform powered by machine learning models trained on historical traffic and weather data. The system predicts vehicle volume based on input parameters like temperature, rain, snow, time, and holiday. Users can interact with a simple web interface to receive real-time traffic volume predictions, helping commuters, traffic departments, and city planners make informed decisions quickly and efficiently.
3.	Novelty / Uniqueness	The uniqueness lies in using environmental and temporal data to accurately estimate traffic flow, eliminating the need for physical sensors or cameras. Unlike traditional systems, this solution is lightweight, cost-effective, and accessible through a browser-based interface. Integration of predictive modeling with user-friendly design makes it suitable for both individual and institutional use.

4.	Social Impact / Customer Satisfactio n	The solution reduces traffic stress, improves commute planning, and supports eco-friendly travel by minimizing idle fuel consumption. It enhances user satisfaction by delivering accurate and timely insights, and enables governments and traffic bodies to plan smarter, reducing road congestion and increasing safety.
5.	Business Model (Revenue Model)	The solution can be monetized through subscription-based access for traffic departments, B2B licensing to smart city planners, or integration into mobile apps and navigation platforms with predictive features. A freemium model can allow individual users to access limited features, encouraging upgrades.
6.	Scalability of the Solution	The system is highly scalable as it relies on cloud-hosted models and user input rather than physical infrastructure.

4.3 Solution Architecture

Date: 28 June 2025

Team ID: LTVIP2025TMID40870

Project Name: *TrafficTelligence: Advanced Traffic Volume Estimation with Machine Learning*

Maximum Marks: 4 Marks

Solution Architecture-Traffic Telligence

- The solution architecture of *TrafficTelligence* integrates a user-friendly HTML interface with a Flask backend that processes input data. The system loads a pre-trained machine learning model to predict traffic volume based on weather and time features. Results are rendered dynamically on a styled output page for user feedback. This modular setup ensures scalability, maintainability, and ease of deployment.

Example - Solution Architecture Diagram:

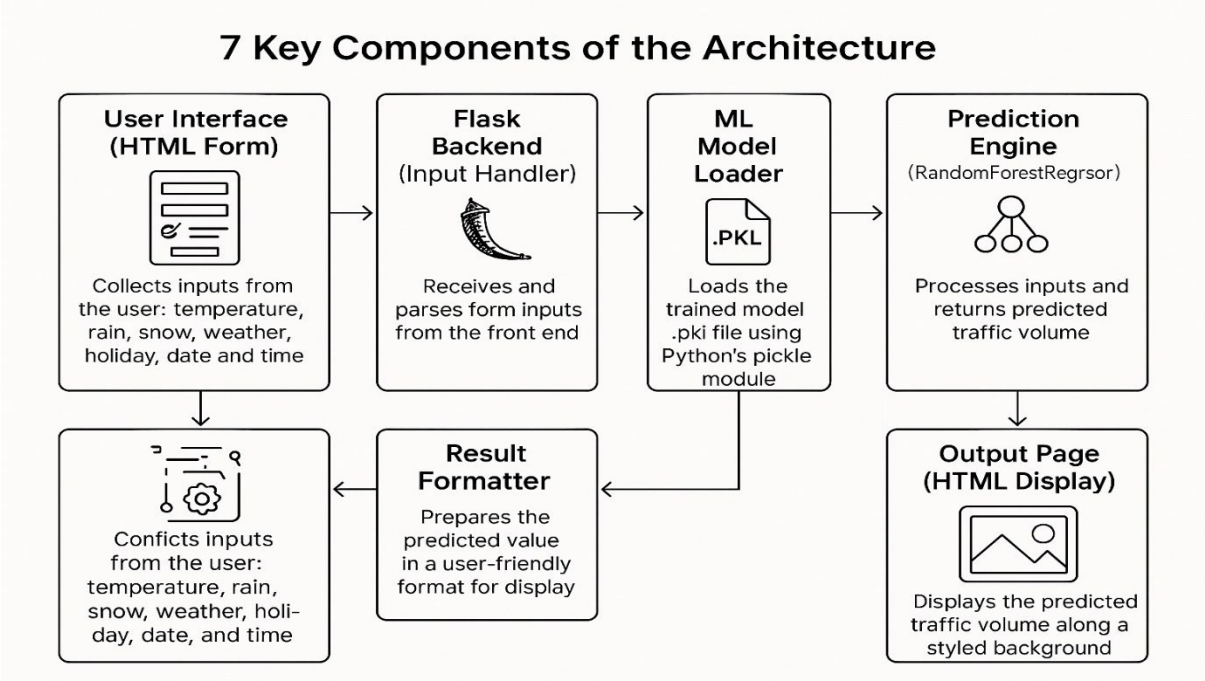
Example - Solution Architecture Diagram:

Components of the Architecture

1. User Interface (HTML Form)
   - Collects inputs from the user: temperature, rain, snow, weather, holiday, date, and time.
2. Flask Backend (Input Handler)
   - Receives and parses form inputs from the front end.
3. Data Preprocessing Unit

- Performs transformations: missing value handling, encoding, and type conversion.

4. ML Model Loader
   - Loads the trained model.pkl file using Python's pickle module.
5. Prediction Engine (RandomForestRegressor)
   - Processes inputs and returns predicted traffic volume.
6. Result Formatter
   - Prepares the predicted value in a user-friendly format for display.
7. Output Page (HTML Display)
   - Displays the predicted traffic volume along with a styled background



5.PROJECT PLANNING AND SCHEDULING

5.1 Project planning

Date: 28 June 2025

Team ID: LTVIP2025TMID40870

Project Name: *TrafficTelligence: Advanced Traffic Volume Estimation with Machine Learning*

Maximum Marks: 5 Marks

🚀 **Agile Sprint Breakdown – TrafficTelligence**

☑ Key Agile Terminology Used

- **Sprint**: A time-boxed development period where specific tasks are completed.
- **Epic**: A large module (like Data Preprocessing or Model Deployment) that is broken into smaller tasks.
- **Story**: A small, actionable task that contributes to an Epic.
- **Story Point**: A number (usually 1, 2, 3, 5) representing the effort and complexity of a story. Based on Fibonacci estimation.

🧩 **Sprint Planning Details**

🟤 **Sprint 1 – (5 Days)**

Epic: Data Collection & Preprocessing

Story Description	Story Point	Assigned To
Collect dataset	2	Member 1
Load dataset into environment	1	Member 2
Handle missing values	3	Member 3
Encode categorical columns	2	Member 4

☑ **Sprint 1 Total Story Points**: **8**

🟤 **Sprint 2 – (5 Days)**

Epic: Model Building & Deployment

Story Description	Story Point	Assigned To
Build and train ML models	5	Member 1
Test and evaluate models	3	Member 2
Create HTML pages	3	Member 3
Deploy with Flask	5	Member 4

☑ **Sprint 2 Total Story Points**: **16**

Summary

Metric Value

Total Story Points 8 (Sprint 1) + 16 (Sprint 2) = **24**

Number of Sprints **2**

Velocity 24 ÷ 2 = **12** Story Points/Sprint

Team Velocity: 12 Story Points per Sprint

Product Backlog, Sprint Schedule, and Estimation (4 Marks):

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story / Task	Story Points	Priority	Team Member
Sprint-1	Data Collection	Collect dataset	2	High	Member 1
Sprint-1	Data Collection	Load dataset	1	High	Member 2
Sprint-1	Data Preprocessing	Handle missing values	3	High	Member 3
Sprint-1	Data Preprocessing	Encode categorical data	2	Medium	Member 4
Sprint-2	Model Building	Build & train ML model	5	High	Member 1
Sprint-2	Model Evaluation	Test & evaluate model	3	High	Member 2
Sprint-2	Deployment	Create HTML Pages	3	Medium	Member 3
Sprint-2	Deployment	Deploy with Flask	5	High	Member 4

Project Tracker, Velocity & Burndown Chart:

Sprint Tracking Table:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed	Sprint Release Date (Actual)
Sprint-1	8	5 Days	19 May 2025	23 May 2025	8	23 May 2025
Sprint-2	16	5 Days	26 May 2025	30 May 2025	16	30 May 2025
Sprint-3	16	5 Days	02 June 2025	06 June 2025		
Sprint-4	16	5 Days	09 June 2025	13 June 2025		
Sprint-5	16	5 Days	16 June 2025	20 June 2025		
Sprint-6	16	5 Days	24 June 2025	28 June 2025		

👥 **Team Members Involved (4):**

- **Member 1** – Data Collection, Model Building
- **Member 2** – Data Loading, Model Testing
- **Member 3** – Handling Missing Data, HTML UI
- **Member 4** – Encoding, Flask Deployment

Velocity:

- Total Story Points = 8 + 16 = 24
  Number of Sprints = 2
  Velocity = 24 ÷ 2 = 12 Story Points/Sprint

Burndown Chart Explanation:

- A burndown chart for this project would show 24 story points at the start, reducing steadily over 2 sprints to 0. Sprint-1 completed 8 points and Sprint-2 completed 16 points. Since both sprints completed all planned stories, the burndown line would show consistent progress with no carryover. This indicates a healthy and balanced sprint schedule

6.FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

Date: 28 June 2025
Team ID: LTVIP2025TMID40870
Project Name: *TrafficTelligence: Advanced Traffic Volume Estimation with Machine Learning*
Maximum Marks: 10 Marks

Model Performance Testing:

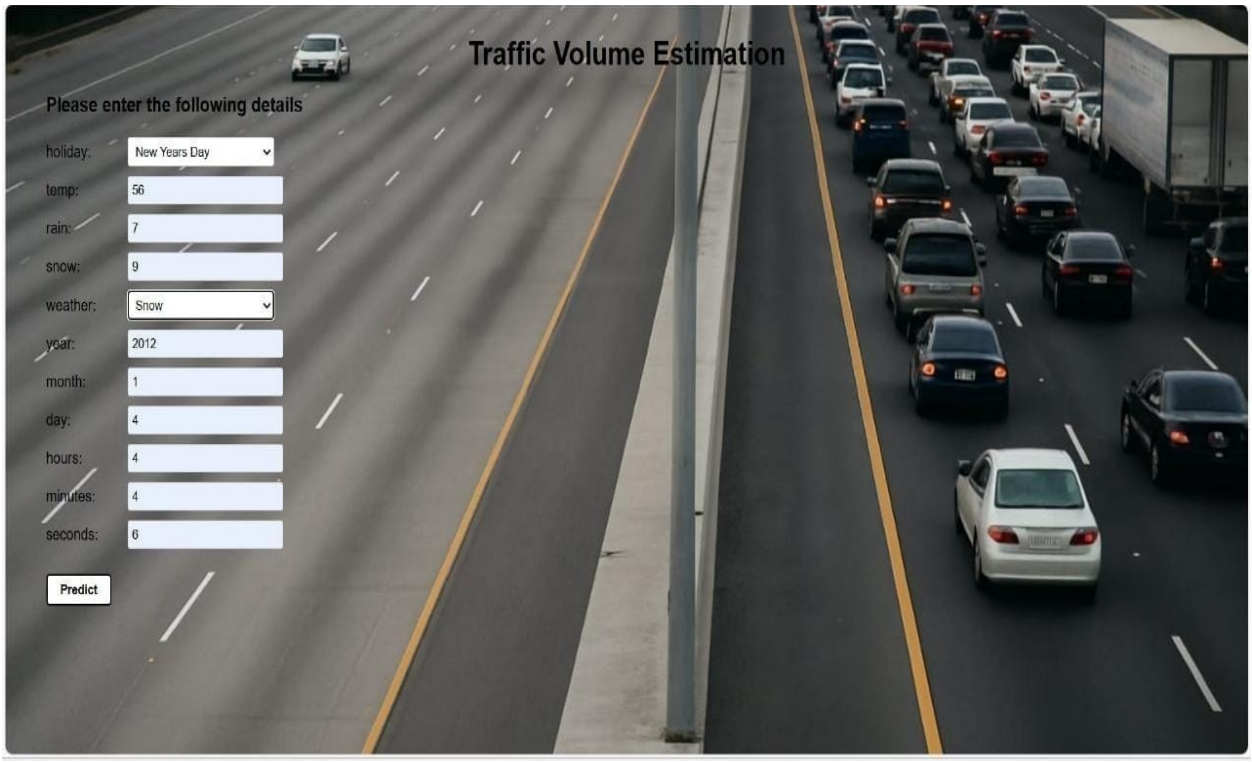
Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot
1.	Metrics	**Regression Model:** for • MSE ≈ 813720.2 • RMSE 903.99 • R² Score (Test Set) 0.7519 Random Forest Regressor (best performing model) • Other R² Scores Linear Regression: -5.42 - Decision Tree: 0.6572 - SVR: -4.58 - XGBoost: 0.7792	

2.	Tune the Model	**Hyperparameter Tuning** Not applied in this version — model used with default parameters (can be improved using GridSearchCV or RandomizedSearchCV) **Validation Method** - train_test_split (80% training / 20% testing) was used for model validation and performance comparison.	

7.RESULTS

7.1 Output Screenshots



8.ADVANTAGES AND DISADVANTAGES

☑ Advantages

1. **Cost-Effective**

o Eliminates the need for expensive traffic cameras and IoT sensors by using machine learning and historical data.

2. **Quick Predictions**

o Provides near-instant traffic volume predictions (~2–3 seconds) through a lightweight web interface.

3. **User-Friendly Interface**

o Simple HTML forms with clear labels and visually styled result pages make the system easy to use for non-technical users.

4. **Data-Driven Decision Making**

o Encourages smarter planning for commuters and authorities by using past patterns and predictive analysis.

5. **Customizable and Scalable**

o Modular architecture allows easy integration with live APIs, cloud deployment, and expansion to mobile or smart city apps.

6. **Cross-Platform Compatibility**

o Built with Flask and web technologies, it can be accessed on any browser across desktop and mobile devices.

7.**Environment Friendly**

o Reducing traffic congestion indirectly helps lower carbon emissions by optimizing travel routes and minimizing idle time.

❌ Disadvantages

1. **No Real-Time Data**
   o The current system depends on static input and lacks real-time GPS or traffic sensor integration.
2. **Accuracy Depends on Dataset Quality**
   o If the training dataset lacks diversity or completeness, the model's predictions may be inaccurate in real-world scenarios.
3. **Lack of Error Handling**
   o Basic version doesn't validate incorrect inputs or show user-friendly error messages.

9.CONCLUSION

The *TrafficTelligence* project successfully demonstrates how machine learning can be applied to predict traffic volume using environmental and temporal factors such as weather, time, and holidays. Through a user-friendly web interface and a lightweight Flask backend, the system offers quick and meaningful insights to assist both commuters and traffic authorities in planning and decision-making. While currently a prototype, it lays a strong foundation for future enhancements such as real-time data integration, mobile responsiveness, and cloud deployment, showcasing the potential of intelligent, data-driven traffic management in smart cities.

TrafficTelligence represents a practical and innovative solution to urban traffic challenges by leveraging historical data and predictive analytics. By combining machine learning with a simple yet effective web application, the project transforms static traffic information into actionable insights. This approach not only reduces dependency on expensive infrastructure but also supports scalable integration into existing traffic systems. With continued development and integration of real-time data, the solution can evolve into a comprehensive platform that empowers smarter mobility, efficient city planning, and improved commuter experiences.

10.FUTURE SCOPE

🔭 Future Scope

1. **Integration with Real-Time APIs**
   Integrate APIs like OpenWeather, Google Maps, or HERE Maps to fetch live data and provide more accurate, real-time predictions.
2. **Mobile Application Development**
   Build a cross-platform mobile app using Flutter or React Native to make traffic prediction accessible on smartphones.
3. **Cloud Deployment**
   Deploy the web app on cloud platforms like AWS, Heroku, or Render for global accessibility and automatic scaling.
4. **User Authentication System**
   Implement secure user login/signup functionality and provide personalized prediction history through user dashboards.
5. **Retraining with Real-Time Data**
   Add functionality to automatically retrain the model as new traffic data becomes available to keep predictions accurate over time.
6. **Support for More Inputs**
   Include additional inputs such as road type, vehicle type, or event schedules to enhance the quality of traffic volume prediction.
7. **Multilingual Interface**
   Offer UI translations in multiple languages to support regional users and improve accessibility.
8. **Visualization Dashboard**
   Integrate dashboards with charts and graphs showing traffic trends, historical data, and prediction accuracy for users and planners.
9. **Voice Assistant Integration**
   Add voice-based commands (using tools like Google Assistant or Alexa) to make the system hands-free and user-friendly.
10. **Smart City Integration**
    Collaborate with smart city infrastructure systems (e.g., traffic lights, sensors) for real-time response and traffic control automation.

11.APPENDIX

SOURCE CODE:

Frontend: Index.html

Welcome | Trafficproject.ipynb | app.py | index.html | output.html

templates > index.html > html > body > div.form-container > form > div.form-group > input

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Traffic Volume Estimation</title>
    <style>
        body {
            background-image: url("{{ url_for('static', filename='bg1.jpg') }}");
            background-size: cover;
            background-position: center;
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
            color: black;
        }

        .main-title {
            font-size: 32px;
            font-weight: bold;
            text-align: center;
            margin-top: 30px;
        }

        .form-title {
            font-size: 20px;
            font-weight: bold;
            margin-left: 50px;
            margin-top: 10px;
        }

        .form-container {
            margin-left: 50px;
            margin-top: 20px;
            width: 400px;
        }
```

```
        .form-group {
            display: flex;
            align-items: center;
            margin-bottom: 10px;
        }

        .form-group label {
            width: 100px;
            font-weight: normal;
            text-align: left;
        }

        .form-group input,
        .form-group select {
            width: 180px;
            padding: 5px;
            border: 1px solid #ccc;
            border-radius: 3px;
        }

        input[type="submit"] {
            margin-top: 15px;
            padding: 6px 16px;
            background-color: white;
            color: black;
            border: 2px solid black;
            border-radius: 4px;
            font-weight: bold;
            cursor: pointer;
        }
```

```
        input[type="submit"]:hover {
            background-color: #f0f0f0;
        }

        .prediction-text {
            margin-top: 20px;
            font-weight: bold;
        }
    </style>
</head>
<body>
    <h1 class="main-title">Traffic Volume Estimation</h1>
    <h1 class="form-title">Please enter the following details</h1>

    <div class="form-container">
        <form action="/predict" method="post">
            <div class="form-group">
                <label for="holiday">holiday:</label>
                <select name="holiday" id="holiday">
                    <option value="7">None</option>
                    <option value="1">Columbus Day</option>
                    <option value="10">Veterans Day</option>
                    <option value="9">Thanksgiving Day</option>
                    <option value="0">Christmas Day</option>
                    <option value="6">New Years Day</option>
                    <option value="11">Washingtons Birthday</option>
                    <option value="5">Memorial Day</option>
                    <option value="2">Independence Day</option>
                    <option value="8">State Fair</option>
                    <option value="3">Labor Day</option>
                    <option value="4">Martin Luther King Jr Day</option>
                </select>
            </div>

            <div class="form-group">
                <label for="temp">temp:</label>
                <input type="text" name="temp" id="temp" placeholder="temp">
            </div>

            <div class="form-group">
                <label for="rain">rain:</label>
                <input type="text" name="rain" id="rain" placeholder="rain">
            </div>

            <div class="form-group">
                <label for="snow">snow:</label>
                <input type="text" name="snow" id="snow" placeholder="snow">
            </div>

            <div class="form-group">
                <label for="weather">weather:</label>
                <select name="weather" id="weather">
                    <option value="1">Clouds</option>
                    <option value="0">Clear</option>
                    <option value="6">Rain</option>
                    <option value="2">Drizzle</option>
                    <option value="5">Mist</option>
                    <option value="4">Haze</option>
                    <option value="3">Fog</option>
                    <option value="10">Thunderstorm</option>
                    <option value="8">Snow</option>
                    <option value="9">Squall</option>
                    <option value="7">Smoke</option>
                </select>
            </div>
```

```
            <div class="form-group">
                <label>year:</label>
                <input type="number" min="2012" max="2025" name="year" placeholder="year" required>
            </div>

            <div class="form-group">
                <label>month:</label>
                <input type="number" min="1" max="12" name="month" placeholder="month" required>
            </div>

            <div class="form-group">
                <label>day:</label>
                <input type="number" min="1" max="31" name="day" placeholder="day" required>
            </div>

            <div class="form-group">
                <label>hours:</label>
                <input type="number" min="0" max="24" name="hours" placeholder="hours" required>
            </div>

            <div class="form-group">
                <label>minutes:</label>
                <input type="number" min="0" max="60" name="minutes" placeholder="minutes" required>
            </div>

            <div class="form-group">
                <label>seconds:</label>
                <input type="number" min="0" max="60" name="seconds" placeholder="seconds" required>
            </div>

            <input type="submit" value="Predict">
        </form>
```

```
            <div class="prediction-text">
                {{ prediction_text }}
            </div>
        </div>
    </body>
    </html>
```

Output.html:

Welcome | Trafficproject.ipynb | app.py | index.html | output.html

templates > output.html > html > head > style > body

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Traffic Volume Prediction</title>
    <style>
        html, body {
            height: 100%;
            margin: 0;
            padding: 0;
        }

        body {
            background-image: url("{{ url_for('static', filename='bg2.jpg') }}");
            background-size: cover;
            background-repeat: no-repeat;
            background-position: center;
            font-family: Arial, sans-serif;
            color: black;
            text-align: center;
            position: relative;
        }

        .title-overlay {
            position: absolute;
            top: 20px;
            left: 50%;
            transform: translateX(-50%);
            font-size: 40px;
            font-weight: bold;
        }

        .sub-result {
            position: absolute;
            top: 200px; /* increased spacing */
            left: 50%;
            transform: translateX(-50%);
            font-size: 26px;
            font-weight: normal;
        }

        .back-btn {
            margin-top: 200px;
        }

        .back-btn a {
            display: inline-block;
            background-color: #007bff;
            color: white;
            text-decoration: none;
            padding: 10px 20px;
            border-radius: 5px;
            font-size: 16px;
        }

        .back-btn a:hover {
            background-color: #0056b3;
        }
    </style>
</head>
<body>
    <h1 class="title-overlay">Traffic Volume Estimation</h1>
    <div class="sub-result">{{ prediction_text }}</div>
</body>
</html>
```

Backend:

app.py > predict

```python
from flask import Flask, render_template, request
import numpy as np
import pickle
import warnings
warnings.filterwarnings("ignore", category=UserWarning)


app = Flask(__name__)

# Load the trained model
model = pickle.load(open('model.pkl', 'rb'))

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/predict', methods=['POST'])
def predict():
    try:
        # Get input from form
        features = [
            int(request.form['holiday']),
            float(request.form['temp']),
            float(request.form['rain']),
            float(request.form['snow']),
            int(request.form['weather']),
            int(request.form['year']),
            int(request.form['month']),
            int(request.form['day']),
            int(request.form['hours']),
            int(request.form['minutes']),
            int(request.form['seconds'])
        ]

        # Make prediction
        prediction = model.predict([features])
```

⚠ 0 ... Connect

```python
        result = int(prediction[0])

        return render_template('output.html', prediction_text=f"Estimated Traffic Volume is: {result} units")

    except Exception as e:
        return render_template('output.html', prediction_text=f"Error: {str(e)}")

if __name__ == '__main__':
    app.run(debug=True)
```

Dataset Link:

https://drive.google.com/file/d/1iV5PfYAmI6YP0_0S4KYy1ZahHOqMgDbM/view

GitHub Link:

https://github.com/konathalapavithra/TrafficTelligence-Advanced-Traffic-VolumeEstimation-with-Machine-Learning./tree/main

Project Demo Link:

https://drive.google.com/file/d/1jlAPutb7_9xUVzKCf2sMZian18goB1V/view?usp=sharing