**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 24 June 2025 |
| Team ID | LTVIP2025TMID35493 |
| Project Name | TrafficTelligence: Advanced Traffic Volume Estimation with Machine Learning |
| Maximum Marks | 4 Marks |

**Technology Stack & Architecture :**

TrafficTelligence follows a robust and modular three-tier architecture, optimized for real-time traffic pattern analysis and prediction:

1. User Interface (Frontend) – Where users upload CSV files of traffic data or view analyzed results and visualizations.
2. Application Logic (Backend) – This layer handles file parsing, preprocessing, model inference, and communicates results back to the UI.
3. Storage & Model – Where trained models (e.g., traffic\_model.pkl) and uploaded data are stored for processing and historical reference.

The application is built using Flask for web integration, Scikit-learn for classical ML modeling, and Pandas/NumPy for data processing. Deployment is designed for both local environments and cloud platforms like Heroku.

**Table-1 : Components & Technologies:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
|  | User Interface | |  | | --- | | Allows data upload and displays predictions/graphs |  |  | | --- | |  | | HTML, CSS, JavaScript, Chart.js |
|  | Application Logic-1 | |  | | --- | | Handles file upload, data parsing |  |  | | --- | |  | | Flask, Python, Pandas |
|  | Application Logic-2 | |  |  |  | | --- | --- | --- | | |  | | --- | |  |  |  | | --- | | Performs preprocessing and feature extraction | |  |  | | --- | |  | | NumPy, Scikit-learn |
|  | Application Logic-3 | |  | | --- | |  |  |  | | --- | | Runs traffic prediction model and shows results |  |  | | --- | |  | | Scikit-learn, Flask |
|  | Database | |  |  |  | | --- | --- | --- | | |  | | --- | |  |  |  | | --- | | Stores uploaded files and prediction history | | | |  | | --- | | SQLite (Optional) | |
|  | Cloud Database | |  | | --- | | For storing user data/models when deployed on cloud |  |  | | --- | |  | | Firebase Storage / AWS S3 (Optional) |
|  | File Storage | |  | | --- | | Temporarily holds uploaded CSV files | | |  | | --- | | Local Filesystem | |
|  | External API-1 | |  |  | | --- | --- | | Could fetch real-time traffic/weather/road data |  | |  |  |  | |  |  |  | | OpenWeatherMap API |
|  | External API-2 | Could pull live traffic sensor data | Google Traffic API |
|  | Machine Learning Model | Trained to detect congestion patterns | Scikit-learn Model |
|  | Infrastructure (Server / Cloud) | Platform where app is hosted | Localhost / Heroku / AWS |

**Table-2: Application Characteristics:**

| **S.No** | **Characteristics** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | Open-Source Frameworks | |  |  |  |  | | --- | --- | --- | --- | | |  | | --- | | Uses reliable, free frameworks for development |  |  | | --- | |  | |  | | Flask, Pandas, NumPy, Scikit-learn |
|  | Security Implementations | Validates CSV content, blocks harmful file types | Flask file validation, secure\_filename() |
|  | Scalable Architecture | Easily extendable for real-time APIs or cloud deployment | 3-tier architecture, modular Flask App |
|  | Availability | Runs on platforms with 24/7 uptime when deployed | Heroku, AWS, GCP |
|  | Performance | |  | | --- | |  |  |  | | --- | | Efficient with fast ML predictions and lightweight processing | | Optimized Pandas operations, Pickle |

**References:**

* Scikit-learn Documentation
* Flask Official Docs
* AWS & Heroku Deployment Guides
* IBM Cloud Architecture Patterns
* UCI Traffic Dataset References
* C4 Model for Software Architecture