PROJECT DESIGN PHASE-II

TECHNOLOGY STACK (ARCHITECTURE AND STACK)

PROJECT NAME:

TrafficTelligence: Advanced Traffic Volume Estimation With Machine Learning

TECHNICAL ARCHITECTURE:

The project encompasses a systematic process, commencing with data collection and preparation, followed by exploratory data analysis and model building using various algorithms. Subsequent performance testing and hyperparameter tuning refine model accuracy. The final phase involves deploying the best model, integrating it into a web framework, and documenting the project comprehensively, including an explanatory video and step-by-step development procedure, underpinned by a technically sound architecture.

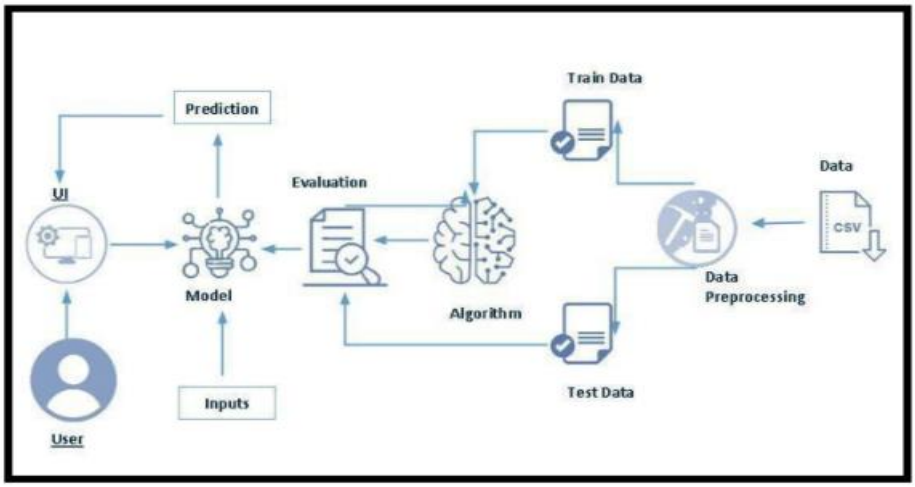

TABLE-1 COMPONENTS AND TECHNOLOGIES:

S.NO.	COMPONENT	DESCRIPTION	TECHNOLOGY
1.	USER INTERFACE	Users interact with the web application by providing input parameters, including area, date, and time, to access traffic analysis. The intuitive interface offers input fields, dropdown menus, and date/time pickers for easy data entry. Once submitted, users receive real-time traffic analysis, ensuring a seamless and informative experience.	HTML, CSS, JavaScript, ReactJS, etc
2.	DATA COLLECTION AND PREPERATION	The process begins with data collection through a provided link,	Python, Google Collab, Microsoft Excel

		followed by the importation of essential libraries. It includes meticulous handling of missing values, date-time column formatting, and a systematic correlation assessment. This professional workflow ensures the prepared dataset's reliability for insightful analysis.	Libraries: -Pandas -Numpy -Seaborn -Scikit Learn
3.	DATA ANALYSIS	Commencing with univariate analysis, which employs distplot and countplot graphs to understand individual feature characteristics. This is followed by bivariate analysis, which investigates relationships between two features. Multivariate analysis extracts inferences, including linearity and feature correlations, from the visual representations, facilitating data-driven insights and informed decision-making in a rigorous and professional manner.	Python Seaborn package
4.	MACHINE LEARNING MODEL	Robust ensemble of regression techniques, featuring linear regression for linear relationships, decision tree regressor for nonlinear patterns, random forest regressor for enhanced accuracy, SVM for complex data separation, and XGBoost for gradient boosting. This diverse array of models, professionally trained, ensures versatile and precise predictive capabilities across a wide range of data scenarios.	Python Libraries: -Scikit Learn -XGBoost
5.	DATABASE	Local storage to house all datasets, providing a secure and efficient data management solution. This local storage system ensures data integrity and accessibility within a closed network.	LOCAL
6.	INFRASTRUCTURE	Application on Local System: -	

TABLE 2: APPLICATION CHARACTERSTICS

S.NO.	CHARACTERSTICS	DESCRIPTION	TECHNOLOGY
1.	AVAILABILITY	The system's availability is conditional on its deployment. Initially, it's confined to the source	

		computer, but upon deployment on a server, it becomes accessible from any networked device. This strategy allows controlled development and testing access before enabling widespread availability across diverse systems.	
2.	PERFORMANCE	Performance metrics indicate that the decision tree model achieves 69% accuracy, while the random forest and XGBoost models exhibit superior performance, with 80% and 79% accuracy, respectively. These results reflect the effectiveness of ensemble methods in enhancing predictive capabilities and underline the potential for improved decision-making using these models.	-
3.	SCALABLE ARCHITECTURE	Methods like Random Forest and XGBoost are often considered scalable, as they can efficiently handle large datasets and take advantage of parallel processing, making them suitable for many scenarios.	-