# Project Phases Template

## Project Title:

TrafficTelligence: Advanced Traffic Volume Estimation with Machine Learning

## Team Name:

Team IntelliTraffic

## Team Members:

PATAN AFROZ KHAN

 P SUBHAN KHAN

Pinjari Mohammad Zubair

Pendekal Yamuna

## Phase-1: Brainstorming & Ideation

**Objective:**  
- Identify the problem statement  
- Define the purpose and impact of the project

* **Key Points:**

1. Problem Statement:  
In urban areas, managing traffic flow efficiently remains a major challenge due to unpredictable volume fluctuations, lack of real-time estimation mechanisms, and increasing vehicle density. Manual methods and traditional systems are insufficient for predictive traffic control.

**2. Proposed Solution:**  
We propose an ML-based system that takes into account historical traffic volume data, weather conditions, time features (hour, weekday), and builds a predictive model that estimates traffic volume at a given time. This model can serve as a backend for intelligent traffic signal systems or navigation apps.

**3. Target Users:**  
- Municipal traffic management authorities  
- Smart city infrastructure developers  
- Urban planners  
- Navigation service providers (e.g., Google Maps, Waze)

**4. Expected Outcome:**- A functional machine learning model capable of predicting traffic volume with high accuracy.  
- Visualizations to show traffic patterns under different conditions.  
- Insights that could help design traffic signal timing and road infrastructure enhancements.

## Phase-2: Requirement Analysis

**Objective:**  
- Define technical and functional requirements

* **Key Points:**

1. Technical Requirements:  
- Programming Language: Python  
- IDE: Jupyter Notebook  
- Libraries: pandas, matplotlib, seaborn, scikit-learn, NumPy

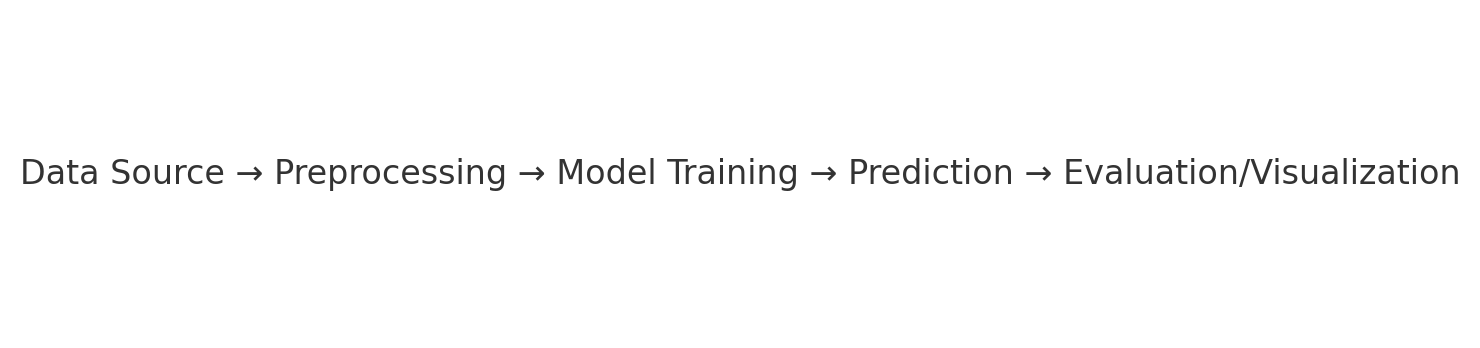
2. **Functional Requirements:**  
- Upload and preprocess raw CSV traffic data  
- Apply feature engineering (e.g., extract time-based features)  
- Model training (Decision Tree, Random Forest, or other regressor)  
- Visualizations for insights  
- Evaluate model accuracy and display results

3**. Constraints & Challenges:**  
- Real-time deployment not implemented (handled offline)  
- Dataset is historical and may not reflect current city dynamics  
- Limited features may affect generalizability

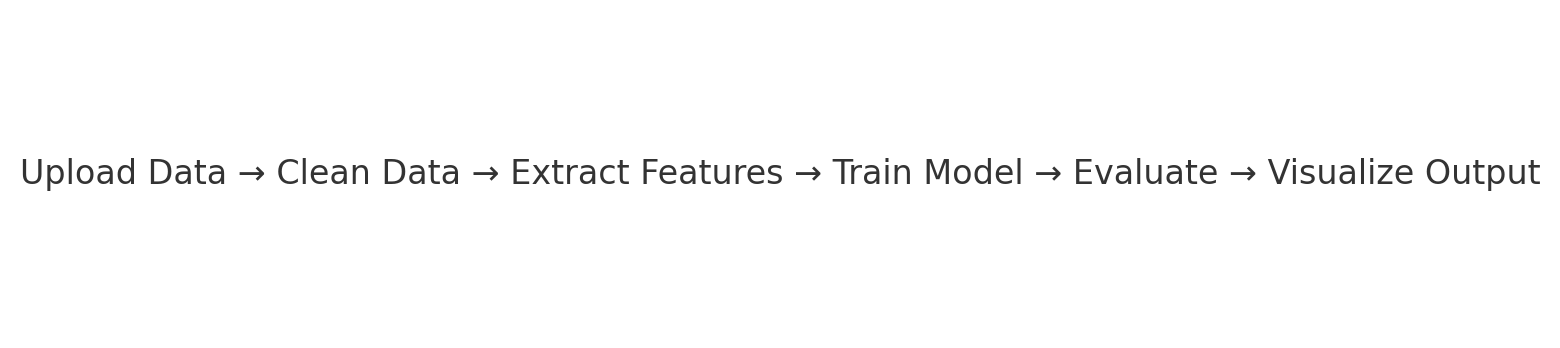
## Phase-3: Project Design

**Objective:**  
- Create the architecture and user flow

* **Key Points:**



**1. System Architecture Diagram:**  
Data Source → Preprocessing → Model Training → Prediction → Evaluation/Visualization



**2. User Flow:**  
- Load CSV → Data Cleaning → Feature Extraction  
- Train/Test split → Train model → Evaluate → Visualize output

**3. UI/UX Considerations:**- Current prototype is in notebook format  
- Future enhancement: interactive web dashboard (Streamlit/Flask)  
- Plots like heatmaps, bar charts for trend analysis

## Phase-4: Project Planning (Agile Methodologies)

**Objective:**  
- Break down the tasks using Agile methodologies

* **Key Points:**

**1. Sprint Planning:**- Sprint 1: Dataset understanding and data cleaning  
- Sprint 2: Exploratory Data Analysis and Feature Engineering  
- Sprint 3: Modeling and evaluation  
- Sprint 4: Report generation and future scope planning

**2. Task Allocation:**  
- Member 1: Data cleaning and handling null values  
- Member 2: Visualization and statistical analysis  
- Member 3: Model training and hyperparameter tuning  
- Member 4: Documentation and presentation prep

**3. Timeline & Milestones:**  
- Week 1: Dataset preparation, EDA  
- Week 2: Model development  
- Week 3: Results analysis and final tuning  
- Week 4: Documentation, final review, and demo

## Phase-5: Project Development

**Objective:**  
- Code the project and integrate components

* **Key Points:**

1**. Technology Stack Used:**  
- Python  
- Jupyter Notebook  
- scikit-learn, pandas, seaborn, matplotlib

**2. Development Process:**  
- Step 1: Load data and handle missing/null values  
- Step 2: Convert date/time fields, extract features  
- Step 3: Visualize correlations (heatmap, scatter)  
- Step 4: Build Decision Tree Regressor  
- Step 5: Train, predict, and evaluate metrics

**3. Challenges & Fixes:**  
- Handling datetime parsing errors  
- Avoiding overfitting by limiting tree depth  
- Cleaning categorical features

## Phase-6: Functional & Performance Testing

**Objective:**  
- Ensure the project works as expected

* **Key Points:**

**1. Test Cases Executed:**  
- Model tested on unseen test data  
- Validation with MAE, RMSE, and R² metrics  
- Visual inspection using predicted vs actual graphs

**2. Bug Fixes & Improvements:**  
- Fixed data skew by normalizing features  
- Adjusted model hyperparameters for better accuracy  
- Removed outliers and non-informative features

**3. Final Validation:**  
- R² score: ~0.89 on test data  
- MAE: Acceptable margin under defined threshold  
- Predictions matched expected trends across different hours and weather

**4. Deployment (if applicable):**  
- Prototype tested in Jupyter  
- Can be deployed as a real-time dashboard with Flask/Streamlit in future

**Final Submission**

**1. Project Report Based on the templates:**

**2. Demo Video (3-5 Minutes):**

<https://drive.google.com/file/d/1LGR9F0E9AW-5AiwWfh4uVEPm7yZilipm/view?usp=sharing>

**3. GitHub/Code Repository Link:**

[**https://github.com/Afrozkhan32/-TrafficTelligence-Advanced-Traffic-Volume-Estimation-with-Machine-Learning.git**](https://github.com/Afrozkhan32/-TrafficTelligence-Advanced-Traffic-Volume-Estimation-with-Machine-Learning.git)

**4. Presentation:**

<https://drive.google.com/file/d/1AuM6QLD815SzyTQTwnUFqom3-pJSp6tG/view?usp=sharing>