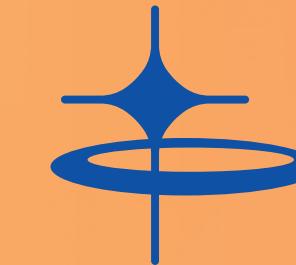
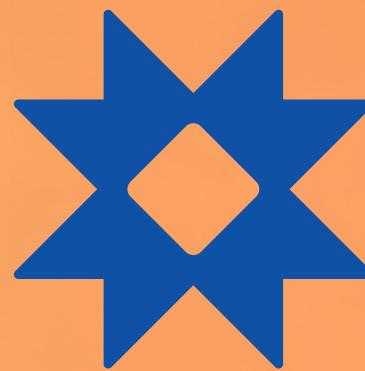
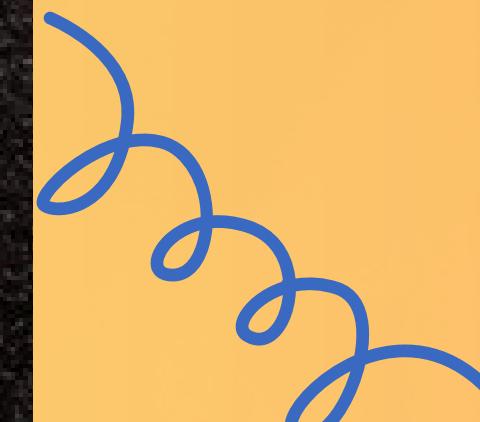




TRANSFORMING URBAN  
TRANSPORTATION SYSTEM



# SMART AI Traffic Management system



A COMPREHENSIVE OVERVIEW



## Smart Traffic Lights

Remotely controlled, AI-driven traffic lights that adapt signal timing in real-time based on live camera feeds, sensor data, and traffic flow patterns at intersections. Machine learning algorithms dynamically optimize green and red light durations to reduce congestion and improve vehicle flow efficiency.

## Traffic Managers Dashboard

An interactive, web-based dashboard (Next.js) that allows traffic managers to monitor live camera feeds, view traffic congestion levels, detect incidents, and receive AI-generated predictions. The dashboard offers insights for predictive modeling, historical traffic analysis, and real-time emergency alerts for faster decision-making.

## AI Analysis Models

Deep learning and computer vision models process live traffic data to detect vehicles, predict congestion, identify violations, and spot accidents. These models generate automatic insights, such as peak traffic times, route optimization suggestions, and anomaly detection.

# Key Components of Smart Traffic Management



# Timeline

---



## 1 .Planning & Requirement Analysis

Define system objectives, needs & scope.

✓ **Problem Identification:** Traffic congestion, accidents.

✓ **System Goals:** Real-time AI traffic control, live camera feeds, emergency handling.

✓ **Stakeholders:** NTSA, Traffic Police, City Authorities & Smart Cities.



## 2. System Design

Design the technical architecture & user interface.

✓ **Architecture:** Backend (FastAPI), Frontend (React), AI Models, IoT Devices.

✓ **Data Flow:** Cameras → AI → Dashboard → Emergency Response.

✓ **Deliverables:** System diagrams, UI Wireframes, API Specs.

# Timeline

---



## 3. Implementation (Development)

Build system components & integrate AI models.

✓ Backend: Data APIs, AI Model endpoints, Notification System.

✓ Frontend: Dashboard (Live Feeds, Insights, Maps).

✓ AI Models: Vehicle Detection, Congestion Prediction, Accident Detection.

✓ IoT: Camera feed connection.

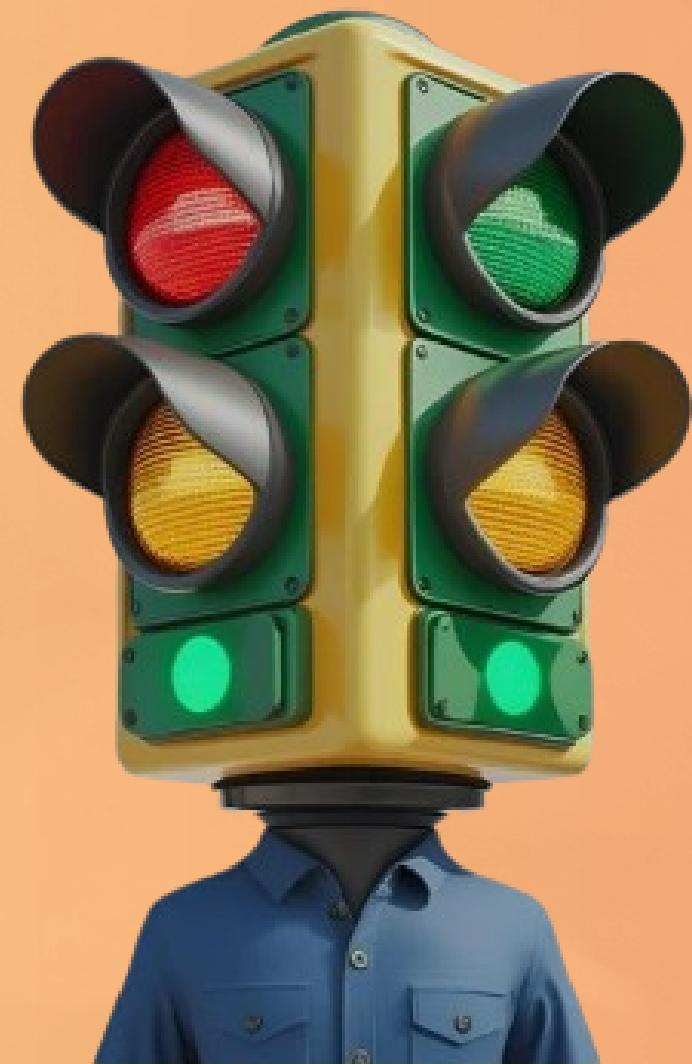
## 4. Testing & Integration

Validate functionality, performance & integration.

✓ Unit & System Testing (API, AI models).

✓ Load & Stress Testing (Simulate traffic data).

✓ Bug Fixing, Performance Optimization.





# DEVELOPMENT ROAD MAP

# Week 1 – Road Network & Data Research + API Setup



- Understand Kenyan road systems & set up data flow base.
- Research Kenyan road structures & traffic laws.
- Identify open data sources (Gov, NTSA).
- Design system architecture for data ingestion.

## **Development Task:**

Set up backend data ingestion API (Flask/FastAPI) to receive real-time feeds.

## **Deliverables:**

- ✓ Kenyan Road Analysis Report
- ✓ Backend Data Ingestion API (v1)

# Week 2 – Camera Integration

## Planning + Camera Module Skeleton



Plan camera feed integration & prototype front-end camera component.

**Tasks:**

Select camera feed standards (RTSP/HTTP).

Plan secure storage & streaming system.

**Development Task:**

Develop front-end camera module skeleton (Next.js) – placeholder for live video stream.

**Deliverables:**

- ✓ Camera Integration Technical Plan
- ✓ React Camera Module (basic frame)

# Week 3 – Live Camera Feed Integration + Stream Display



Integrate camera streams into the dashboard.

**Tasks:**

Connect to IP/CCTV feeds.

Test live streaming (on local/remote dashboard).

**Development Task:**

Implement live RTSP/HTTP stream player in the dashboard using WebRTC/FFmpeg/Janus Gateway.

**Deliverables:**

- ✓ Functional Camera Stream in Dashboard
- ✓ Stream Display & Playback Module

# Week 4 – Frame Capture + Data Storage Component



Capture image frames for AI model training.

**Tasks:**

Implement frame snapshot and save to storage.

Start labeling pipeline for sample images.

**Development Task:**

Build backend frame capture endpoint (Flask/FastAPI) and connect to storage (local/cloud).

**Deliverables:**

- ✓ Frame Capture Feature
- ✓ Labeled Image Dataset

# Week 5 – AI Traffic Detection Model + API Endpoint



Develop & train AI models for congestion & vehicle detection.

**Tasks:**

Train object detection model (YOLOv8/SSD).

Evaluate detection performance.

**Development Task:**

Expose AI inference results via backend API (FastAPI endpoint) for dashboard consumption.

**Deliverables:**

- ✓ Trained Traffic Detection AI Model
- ✓ AI Result API (JSON Response)

# Week 6 – Auto-Analysis Model + Dashboard Insights Panel



Auto-generate traffic insights & analysis.

**Tasks:**

Build AI module for real-time trend analysis.

Automate data labeling (AutoML).

**Development Task:**

Develop Insights Panel on Dashboard (NextJS) –  
Display AI-driven traffic summaries & graphs.

**Deliverables:**

- ✓ Auto-Analysis AI Model
- ✓ Insights Panel on Dashboard

# Week 7 – Accident Detection AI + Emergency Notification Feature



Detect accidents and notify authorities.

**Tasks:**

Train model to detect sudden stops/crashes.

Simulate accident scenarios for model testing.

**Development Task:**

Implement auto-notification system (SMS/Email API) for emergency alerts.

**Deliverables:**

✓ Accident Detection AI Model

✓ Emergency Notification Component

# Week 8 – System Optimization + Modular Architecture Update



Optimize performance & prepare for scaling.

**Tasks:**

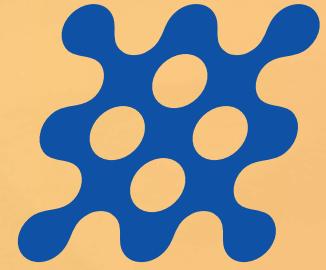
- Optimize latency, stream processing.
- Modularize components for scaling.

**Development Task:**

Refactor backend to microservices structure – Each major feature as a service.

**Deliverables:**

- ✓ Optimized System
- ✓ Modular Backend Architecture



"Every great dream begins with a dreamer. To start dreaming your future, close your eyes, envision your goals, and let your imagination guide you on the path toward possibility."

– Iyke Konzolaw

**Thank you !!!**