



FlowTrack: Intelligent Traffic Control

Optimizing Traffic, Enhancing Mobility



Objective

The **Traffic Automation System (TAS)** aims to streamline the traffic management process using modern technology for a smart city initiative. The system will manage and control traffic signals, monitor traffic patterns in real-time, detect traffic violations, and manage vehicle registration data.

This system will use **MySQL** to handle structured data like vehicle registrations, traffic violations, and traffic signals, while **MongoDB** will store unstructured data like traffic sensor logs and traffic camera footage. The system will also integrate with **AWS S3** to store media files such as violation images and video footage.

Brief Requirement

The Traffic Automation System (TAS) will allow:

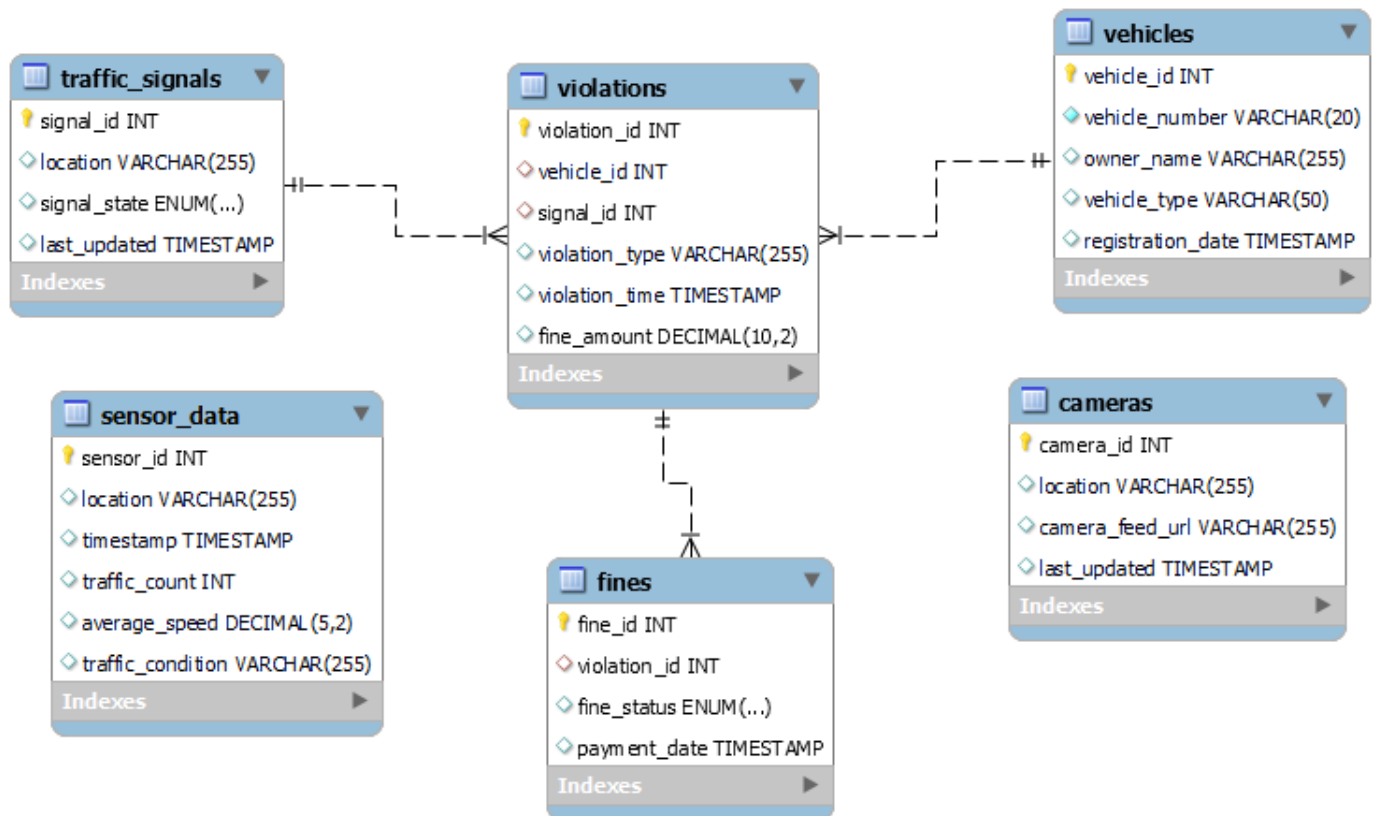
1. **Vehicle Registration:** Users can register their vehicles in the system, providing details like vehicle number, model, owner details, etc.
2. **Traffic Signal Management:** Authorities can control traffic signal timings for different areas, taking real-time traffic flow into account.
3. **Real-time Traffic Monitoring:** Using traffic sensors and cameras, the system will track the real-time traffic conditions at various intersections.
4. **Violation Detection and Management:** The system will detect traffic violations such as red-light jumping and over-speeding using camera footage and traffic sensors.
5. **Violation Report Generation:** For detected violations, the system will generate violation reports, including violation images, fines, and details of the vehicle involved.
6. **Dynamic Signal Adjustment:** Based on traffic flow, the system will adjust the traffic signal timings to optimize traffic management.

The back-end of the system will integrate:

- **MySQL** to store structured data such as vehicle registration details, violation reports, traffic signals, and fines.
- **MongoDB** to store unstructured data such as traffic sensor logs, camera footage, and sensor readings.
- Integration with **AWS S3** for storing images and videos of violations.

Database Architecture

- **MySQL:** Used for structured transactional data like:
 - **Vehicles:** Vehicle registration data.
 - **Traffic Signals:** Data related to traffic light timings and locations.
 - **Violations:** Traffic violation records (speeding, red-light violations).
 - **Fines:** Information about fines issued due to violations.
- **MongoDB:** Used for unstructured data like:
 - **Traffic Sensor Logs:** Logs of sensor readings for monitoring traffic flow.
 - **Cameras:** Camera footage from surveillance cameras at traffic intersections.
 - **Real-Time Data:** Logs of traffic flows and sensor-based adjustments in traffic signal timings.



Explanation of Key Constraints:

Foreign Keys:

- violations table links to vehicles and traffic_signals to associate violations with a specific vehicle and signal.
- finances table links to violations to track payment status for each violation.

Unique Keys:

- vehicle_number in the vehicles table is unique to ensure no two vehicles share the same number.

Default Values:

- signal_state in traffic_signals defaults to Red, assuming that when not updated, traffic lights are red.
- fine_status in the fines table defaults to 'Unpaid' to indicate that the fine is initially unpaid.

MongoDB Schema for Camera Footage

Id	Field Name	Datatype	Default Value	Additional Fields
1	camera_id	String	-	required: true
2	location	String	-	required: true
3	camera_feed_url	String	-	required: true
4	last_updated	Date	Date.now()	-
5	violation_images	[String]	-	-
6	violation_videos	[String]	-	-
7	tags	[String]	-	-

API Endpoints Table

Sr.No	API Title	API Endpoint	Sample Input	Sample Output	Status Code
1	<div>Flask</div> <div>nodejs</div> signin Admin	POST/signin	{ "email": "admin@ admin.com, "password": "1234 56" }	{ "message": "succ ess", "credentials": { "id" : 1, "email": admin @admin.com , "tok en": "lkajfdsjlj3os" }	200
2	<div>Flask</div> validate token	POST/auth/validate	{ "token": " lkajfdsjlj3os" }	{ "message": "succ ess" }	200
3	<div>Flask</div> Register Vehicle	POST /vehicles/register	{ "vehicle_number" : "ABC123", "owner_name": "John Doe", "vehicle_type": "Sedan" }	{ "message": "Vehicle registered successfully", "vehicle_id": 1 }	201
4	<div>Flask</div> Get Vehicle Details	GET /vehicles/{vehicle_id}	{ "vehicle_id": 1 }	{ "vehicle_id": 1, "vehicle_number" : "ABC123", "owner_name": "John Doe", "vehicle_type": "Sedan" }	200
5	<div>Flask</div> Update Traffic Signal State	PUT /signals/{signal_id}/state	{ "signal_state": "Green" }	{ "message": "Traffic signal updated", "signal_state": "Green" }	200
6	<div>Flask</div> Get Violation Report	GET /violations/{violation_id}	{ "violation_id": 1 }	{ "violation_id": 1, "vehicle_number" : "ABC123", "violation_type": "Red Light Jump", "fine_amount": 150.00 }	200
7	<div>Flask</div> Pay Fine	PUT /fines/{fine_id}/pay	{ "payment_date": "2024-12- 13T10:00:00" }	{ "message": "Fine paid successfully", "fine_status": "Paid" }	200
8	<div>Flask</div> Monitor Traffic Sensor Data	GET /sensors/{sensor_id}/data	{ "sensor_id": 1 }	{ "sensor_id": 1, "location": "Main Street", "traffic_count": 250, "average_speed": 45.6, "traffic_condition" : "Moderate" }	200
9	<div>nodejs</div> Get Camera Feed	GET /cameras/{camera_id}/feed	{ "camera_id": 1 }	{ "camera_feed_url" : "http://s3.amazo naws.com/camer a1_feed" }	200

10	<div>Flask</div> Generate Violation Report	POST /violations/generate	{ "vehicle_id": 1, "signal_id": 1, "violation_type": "Red Light Jump", "fine_amount": 150.00 }	{ "message": "Violation report generated", "violation_id": 1 }	201
11	<div>Flask</div> Adjust Traffic Signal Based on Sensors	PUT /signals/{signal_id}/adjust	{ "traffic_condition": "Moderate" }	{ "message": "Traffic signal adjusted successfully", "signal_state": "Green" }	200
12	<div>Flask</div> Delete Traffic Signal	DELETE /signals/{signal_id}	{ "signal_id": 1 }	{ "message": "Traffic signal deleted successfully" }	200

Authentication using NodeJS vs flask

■ nodejs ■ flask

