DBMS MiniProject
TRAFFIC MONITORING SYSTEM

Submitted by: Rishika Kinger
SRN: PES1UG21CS489
Class: 5-H

INTRODUCTION

In the ever-growing urban landscape, managing and optimizing traffic flow is a critical aspect of ensuring public safety and efficient transportation. The Traffic Monitoring System (TMS) is an innovative solution designed to provide real-time insights into traffic conditions, empowering both citizens and law enforcement to contribute to the improvement of traffic management. This system features two distinct logins: one for citizens and police officers, and another for administrators, each with specific functionalities tailored to their roles.

1. Citizen and Police Officer Login:

Citizen Access and Police Officer Access:
Citizens/ Police officers can use this login to access and add information. They can add vehicle and road data, view the number of cars and plan routes accordingly.

2. Admin Login:
The administrative login provides control over the system's core functionalities. Administrators can add new roads to the system, update road information, and manage user accounts. This access ensures the system remains dynamic and adapts to the changing traffic landscape. Additionally, administrators can generate reports based on traffic data, helping in decision-making for future infrastructure planning.

Key Features:

Traffic Monitoring: The system provides live updates on number of vehicles, enabling users to make informed decisions regarding their routes.

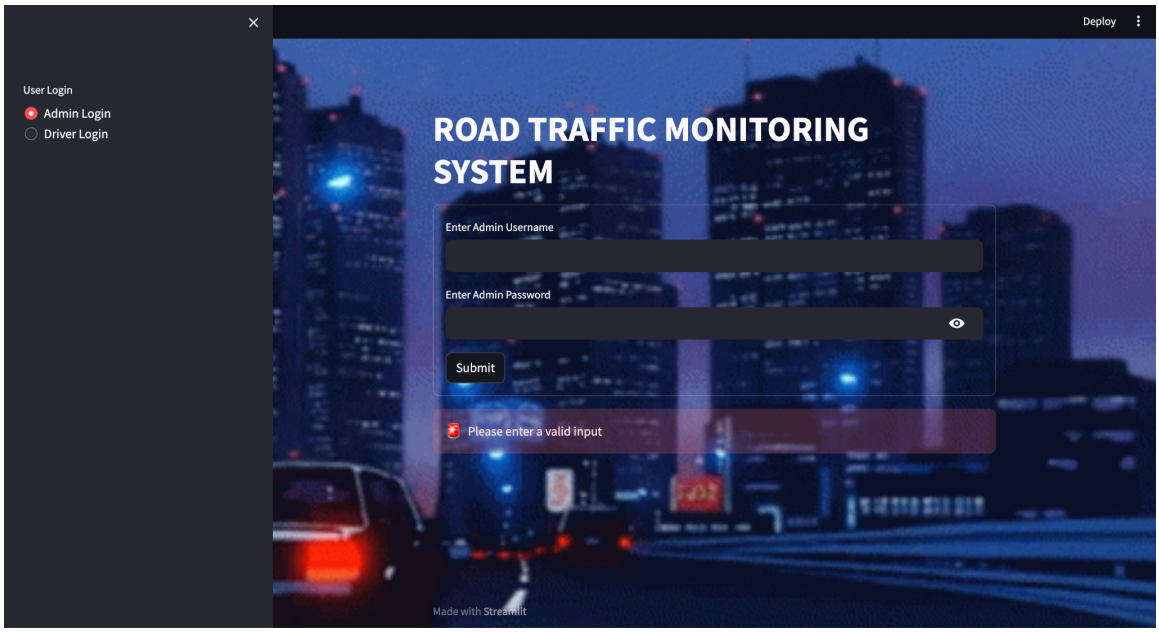
Police Patrol Integration: Police officers can update their patrol status, ensuring accurate information on roads patrolled and those requiring attention.

Road Management: Administrators can add new roads, update existing information, and ensure the system reflects the current state of the road network.

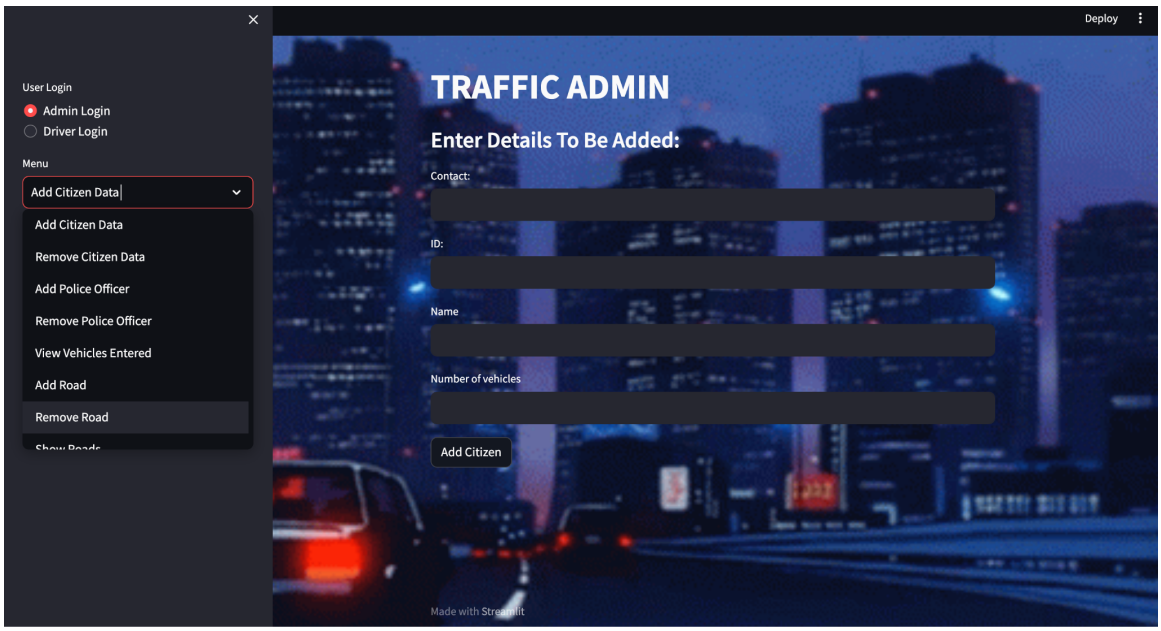
The Traffic Monitoring System aims to create a collaborative platform where citizens, law enforcement, and administrators work together to enhance traffic management and contribute to safer and more efficient transportation systems. This project can also be developed further to create a google maps like system, where data is manually entered.

SCREENSHOTS

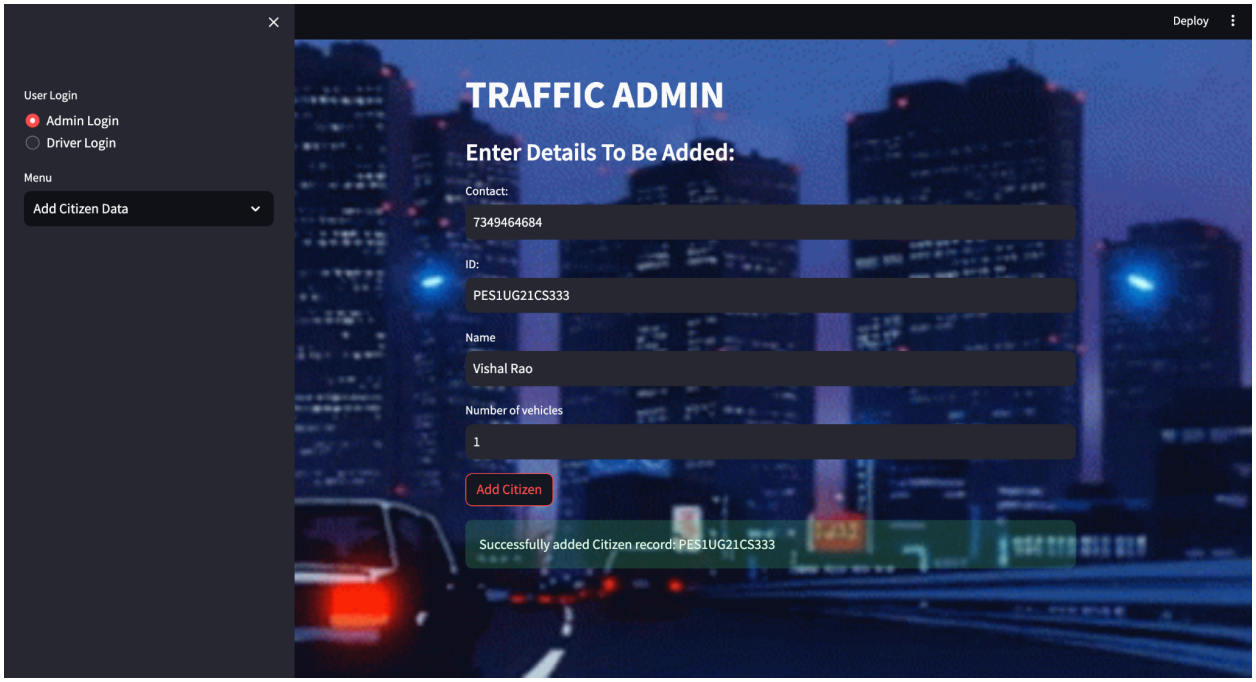
Login



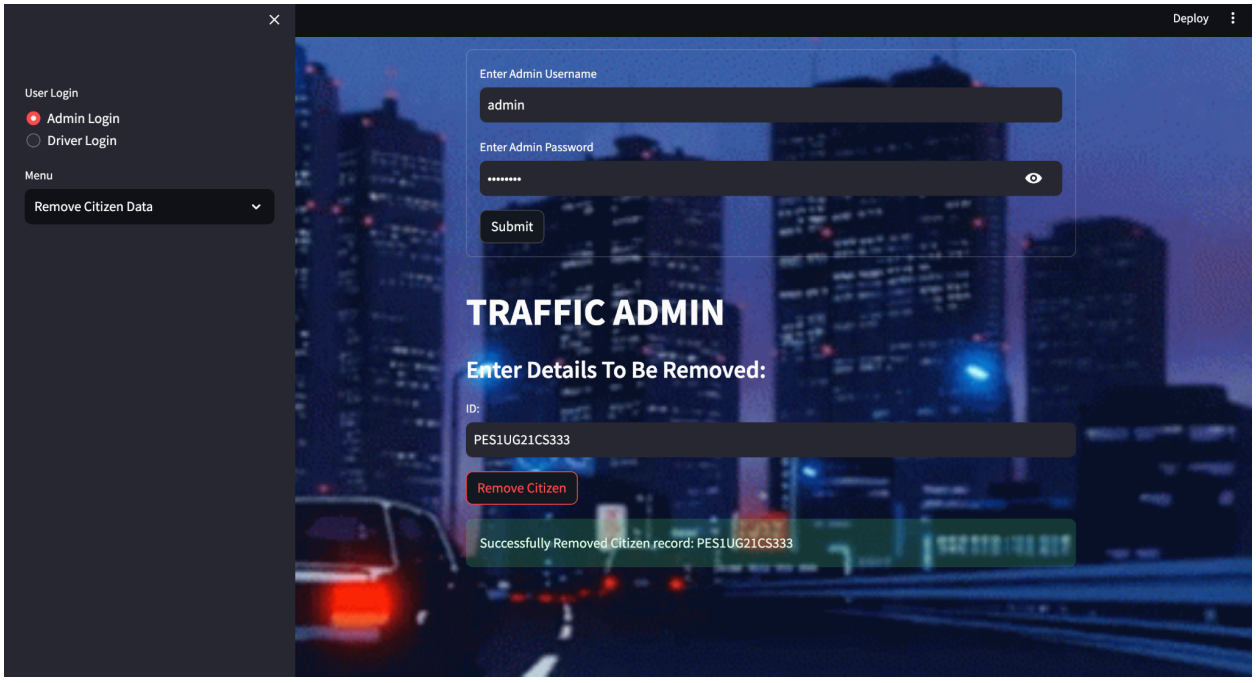
Admin dashboard



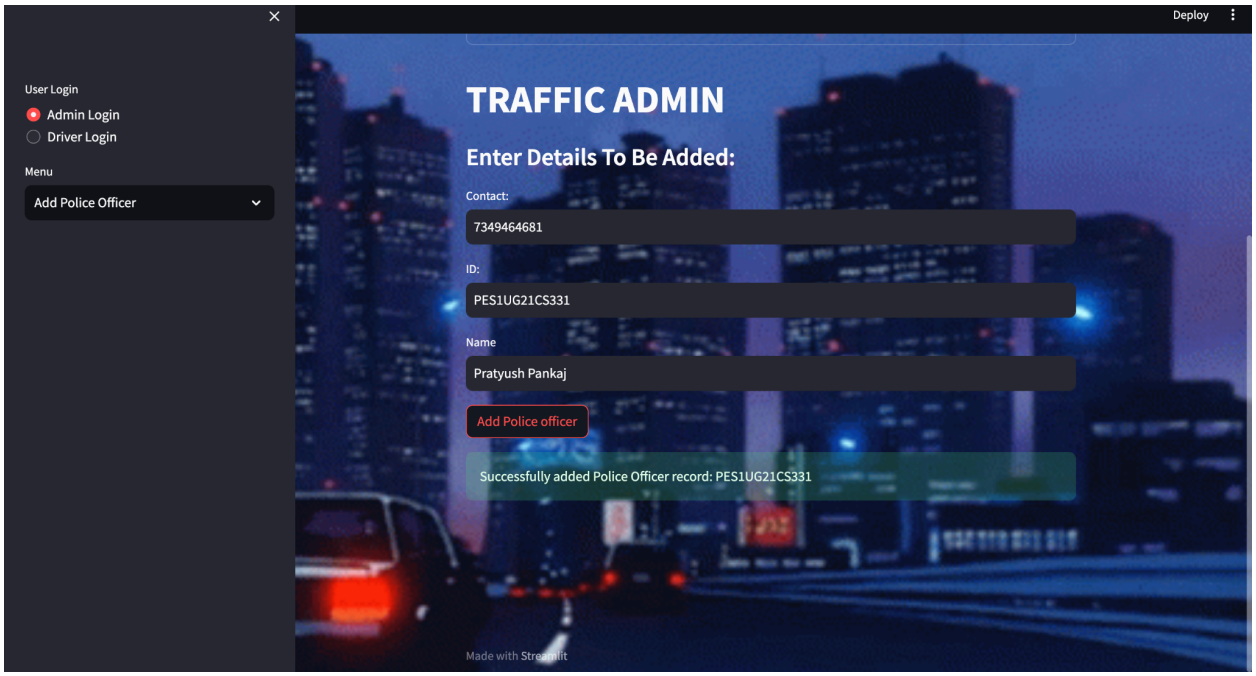
Adding citizen data



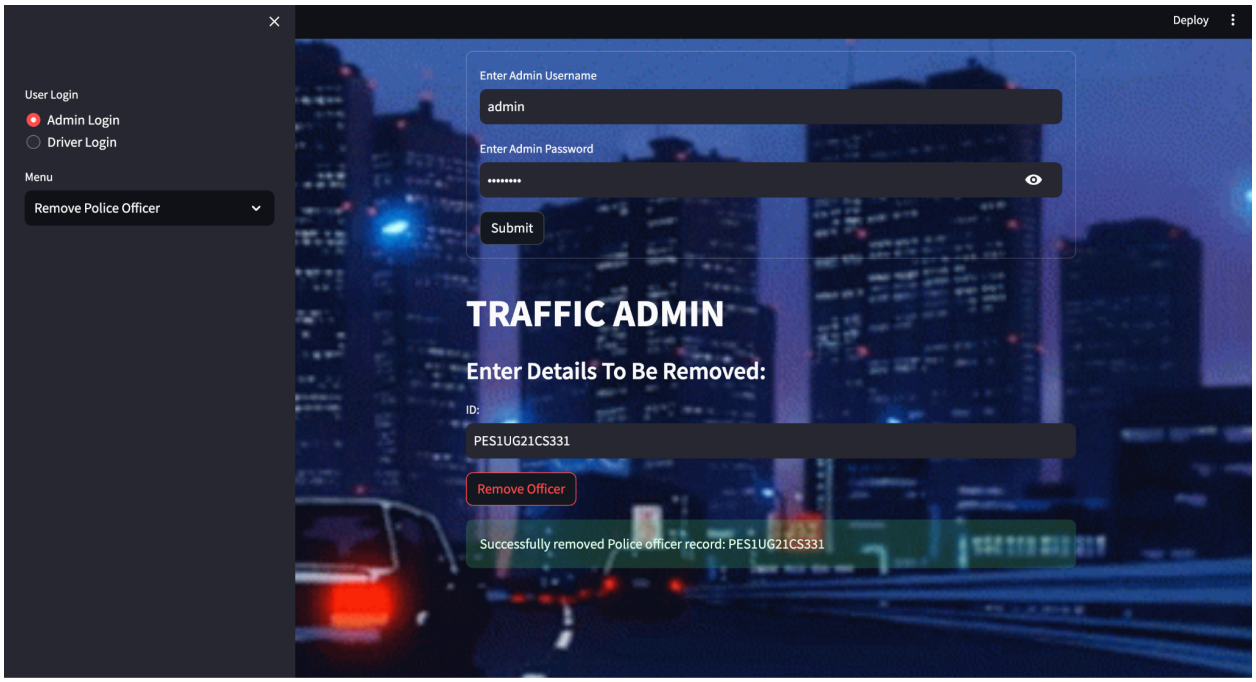
Removing citizen data



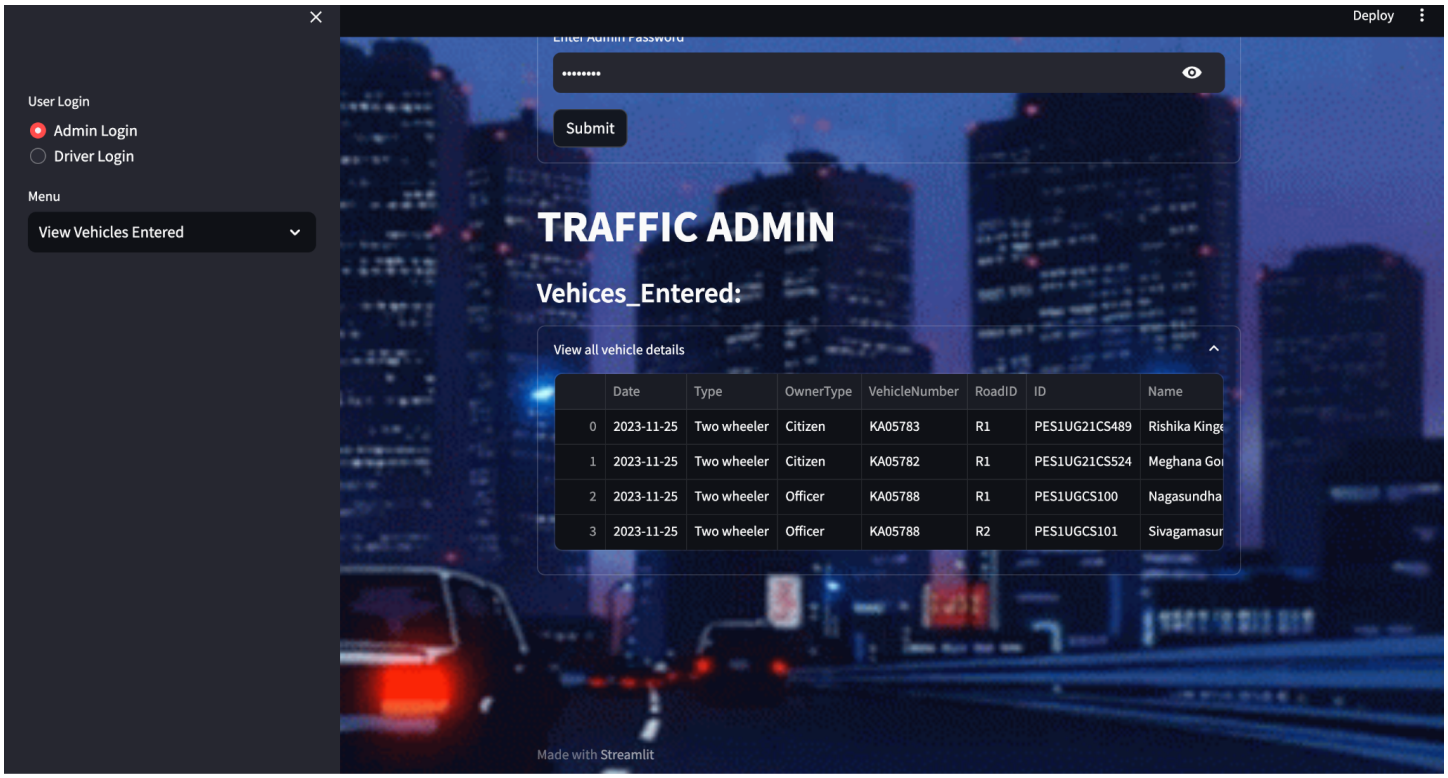
Adding police officer



Removing citizen data

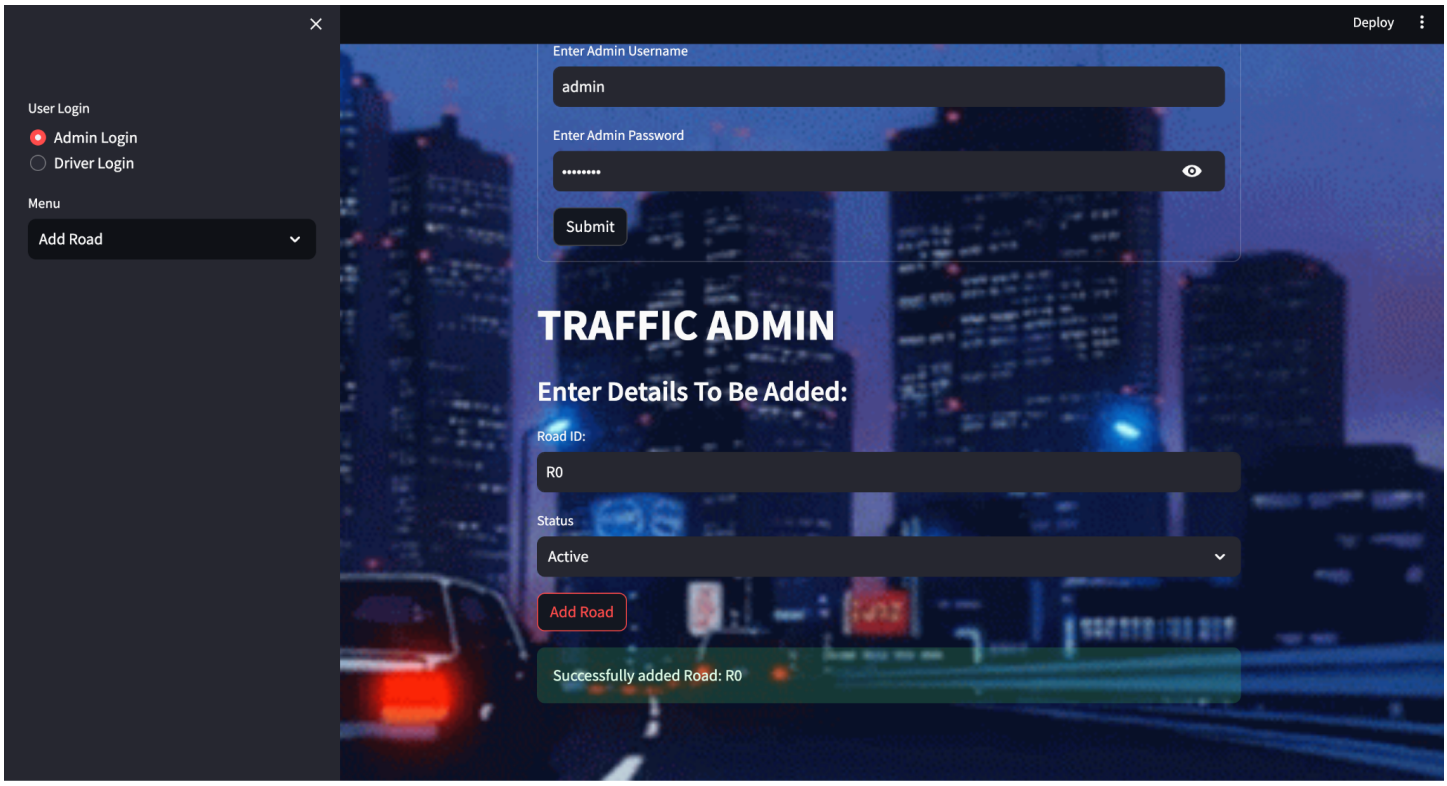


Viewing all vehicles present on roads

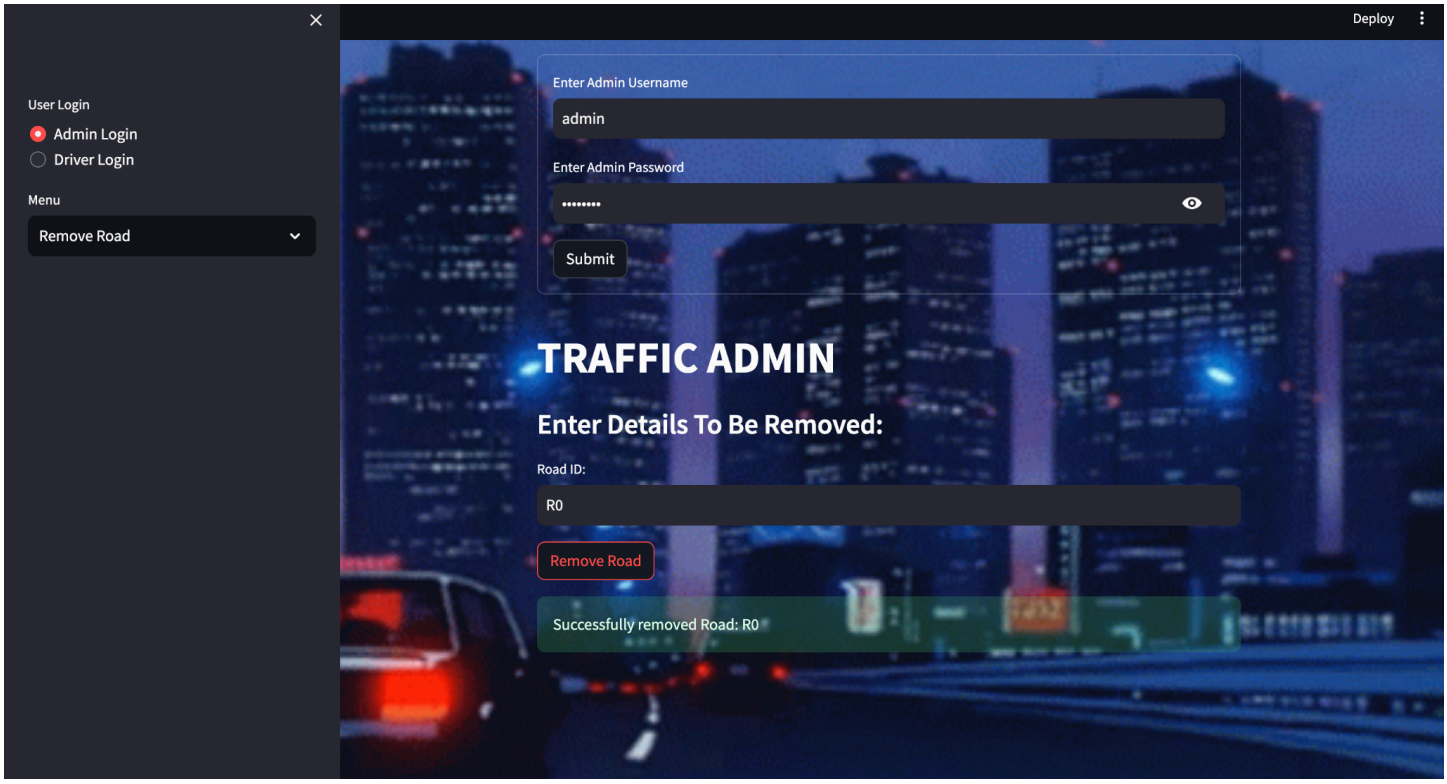


	Date	Type	OwnerType	VehicleNumber	RoadID	ID	Name	Contact
0	2023-11-25	Two wheeler	Citizen	KA05783	R1	PES1UG21CS489	Rishika Kinger	8310730145
1	2023-11-25	Two wheeler	Citizen	KA05782	R1	PES1UG21CS524	Meghana Goru	9353820323
2	2023-11-25	Two wheeler	Officer	KA05788	R1	PES1UGCS100	Nagasundhari	919299292
3	2023-11-25	Two wheeler	Officer	KA05788	R2	PES1UGCS101	Sivagamasundhari	020202002

Adding a road



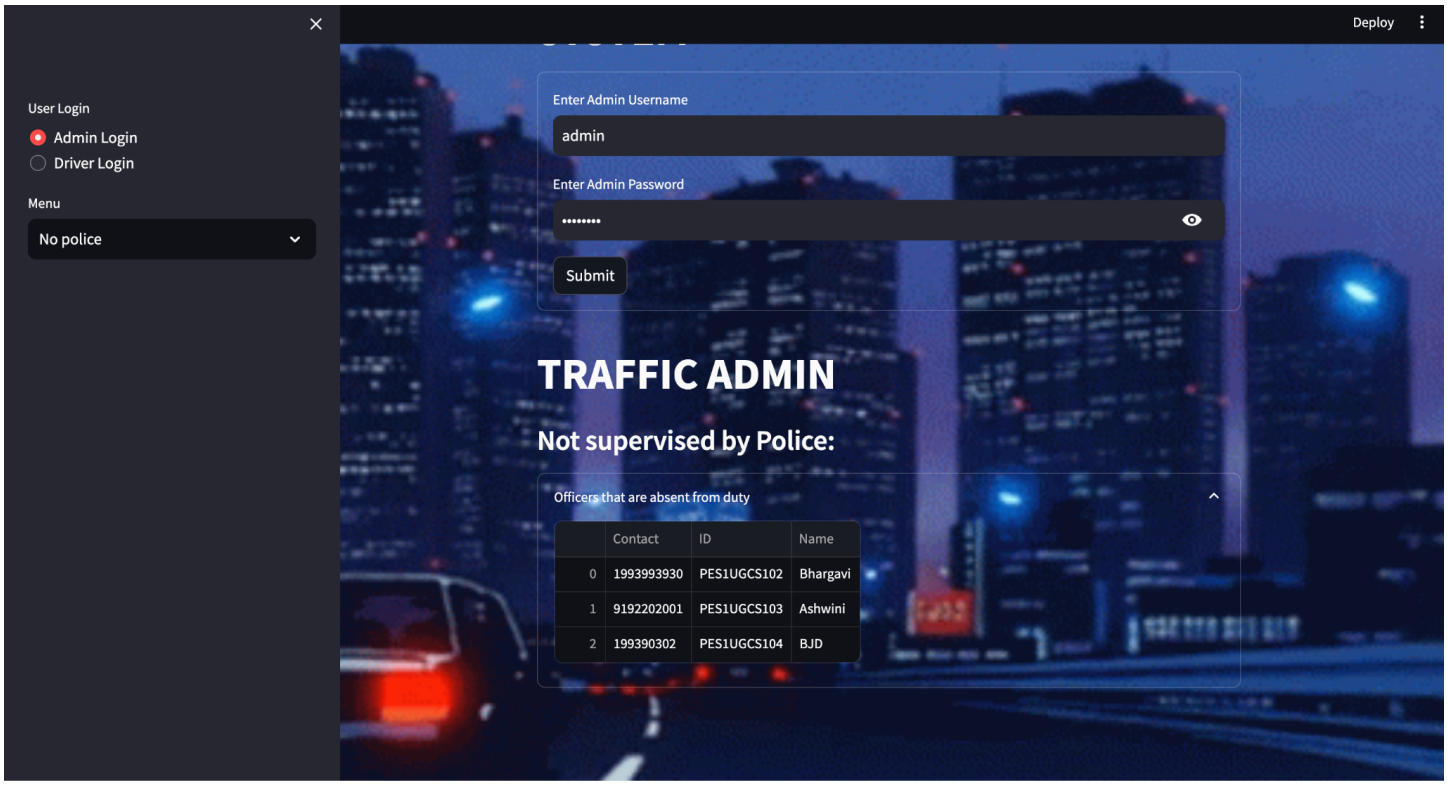
Removing a road



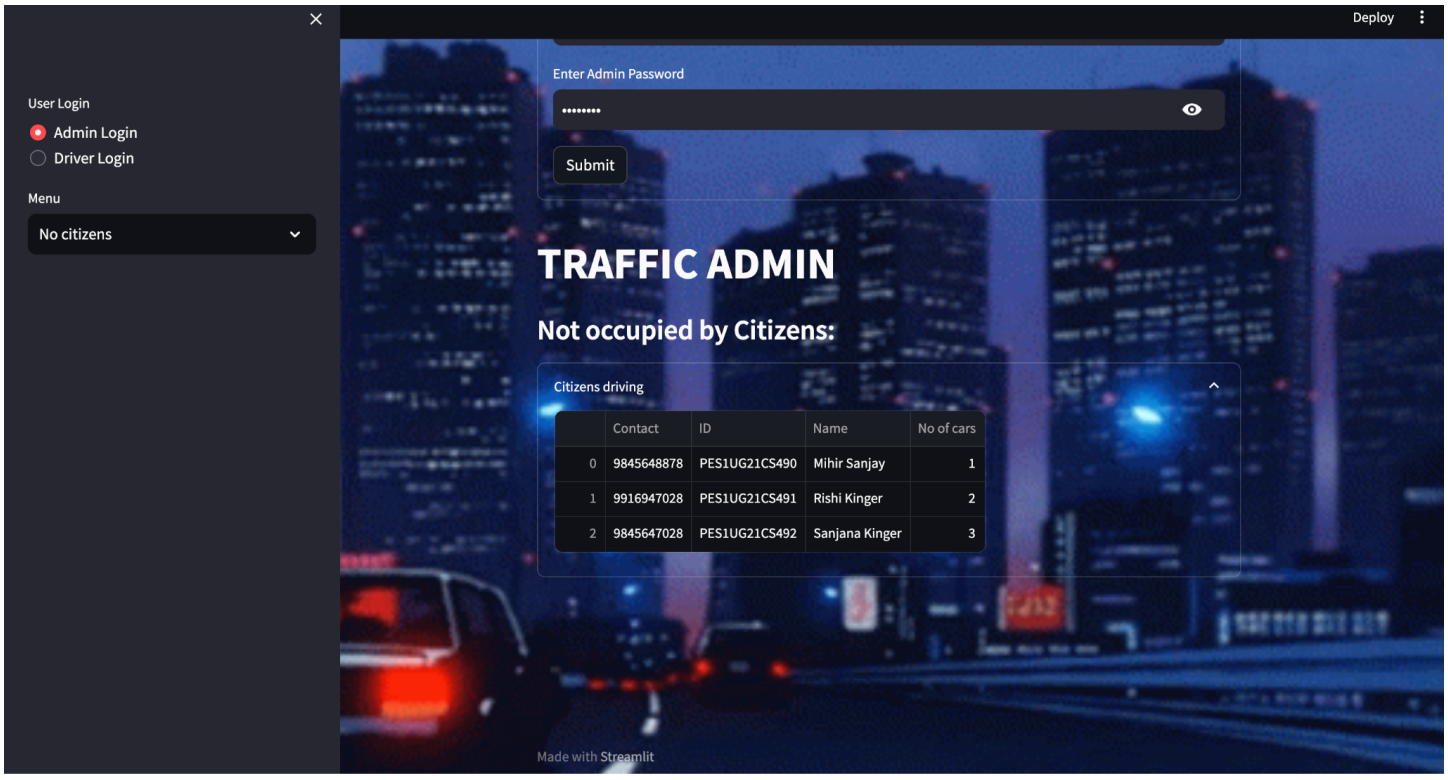
Viewing the status of all roads

Distribution of roads

Police officers off duty



Citizens off roads



Driver Dashboard

Adding a vehicle

Viewing number of vehicles


Deploy
ROAD TRAFFIC MONITORING SYSTEM
User Login
Admin Login
Driver Login
Menu
Vehicle COUNT
Enter Username
driver
Enter Password
Submit
DRIVER DASHBOARD
All the vehicles
Vehicle Count
0
5

QUERIES USED:

1. SQL file

```
create database dbmsfinal;
use dbmsfinal;

CREATE USER 'rkin'@'localhost' IDENTIFIED BY '2393';
GRANT ALL PRIVILEGES ON dbmsfinal.* TO 'rkin'@'localhost';
```

– Creating tables

```
CREATE TABLE  road(road_id varchar(10) NOT NULL, Status
VARCHAR(50) DEFAULT 'Active',PRIMARY KEY(road_id));

CREATE TABLE  Vehicle_in(Date DATETIME, Type TEXT NOT NULL,
Owner_type TEXT NOT NULL, Vehicle_no varchar(50) NOT NULL, P_id
varchar(2) NOT NULL,ID varchar(13) NOT NULL,PRIMARY KEY(ID,
Date),FOREIGN KEY(P_id) REFERENCES road(road_id));

CREATE TABLE  citizen(Contact varchar(10) NOT NULL, ID varchar(13)
NOT NULL, Name varchar(50) NOT NULL, no_of_cars INT,PRIMARY
KEY(ID));

CREATE TABLE  officer(Contact varchar(10) NOT NULL, ID varchar(13)
NOT NULL, Name varchar(50) NOT NULL,PRIMARY KEY(ID));
```

– Inserting dummy values

```
INSERT INTO citizen(Contact, ID , Name , no_of_cars) VALUES
("9353820323","PES1UG21CS524","Meghana Goru","1");
INSERT INTO citizen(Contact, ID , Name , no_of_cars) VALUES (
"8310730145","PES1UG21CS489","Rishika Kinger","1");
```

```
INSERT INTO citizen(Contact, ID , Name , no_of_cars) VALUES
("9845648878","PES1UG21CS490","Mihir Sanjay","1");
INSERT INTO citizen(Contact, ID , Name , no_of_cars) VALUES
("9916947028","PES1UG21CS491","Rishi Kinger","2");
INSERT INTO citizen(Contact, ID , Name , no_of_cars) VALUES
("9845647028","PES1UG21CS492","Sanjana Kinger","3");


INSERT INTO officer(Contact , ID , Name ) VALUES
("919299292","PES1UGCS100","Nagasundhari");
INSERT INTO officer(Contact , ID , Name ) VALUES
("020202002","PES1UGCS101","Sivagamasundhari");
INSERT INTO officer(Contact , ID , Name ) VALUES
("1993993930","PES1UGCS102","Bhargavi");
INSERT INTO officer(Contact , ID , Name ) VALUES
("9192202001","PES1UGCS103","Ashwini");
INSERT INTO officer(Contact , ID , Name ) VALUES
("199390302","PES1UGCS104","BJD");


INSERT INTO road(road_id, Status) VALUES ('R1', 'Active');
INSERT INTO road(road_id, Status) VALUES ('R2', 'Active');
INSERT INTO road(road_id, Status) VALUES ('R3', 'Active');
INSERT INTO road(road_id, Status) VALUES ('R4', 'Active');
INSERT INTO road(road_id, Status) VALUES ('R5', 'Active');
```

– Procedure

```
DELIMITER //
CREATE PROCEDURE StatusUpdation ( IN Road_id VARCHAR(2))
BEGIN
UPDATE road SET Status="Occupied" where road.road_id = Road_id ;
END;
//
Delimiter ;
```

– Trigger

```
DELIMITER //
CREATE TRIGGER SETSTATUS  AFTER INSERT ON Vehicle_in
FOR EACH ROW
BEGIN
   CALL StatusUpdation(NEW.P_id);
END //
Delimiter ;
```

2. Queries in the python file

Basic CRUD, SELECT operations

```
CREATE TABLE IF NOT EXISTS Vehicle_in(Date DATE, Type TEXT,
Owner_type TEXT, Vehicle_no varchar(50), P_id varchar(2),ID
varchar(13))
```

```
INSERT INTO Vehicle_in(Date, Type, Owner_type,Vehicle_no,
P_id,ID) VALUES (%s,%s,%s,%s,%s,%s)
```

```
SELECT COUNT(P_id) from Vehicle_in
```

```
CREATE TABLE IF NOT EXISTS citizen(Contact varchar(10), ID
varchar(13), Name varchar(50), no_of_cars INT)
```

```
CREATE TABLE IF NOT EXISTS officer(Contact varchar(10), ID
varchar(13), Name varchar(50))
```

```
CREATE TABLE IF NOT EXISTS road(road_id varchar(2), Status TEXT)
```

```
INSERT INTO citizen(Contact, ID , Name , no_of_cars) VALUES
(%s,%s,%s,%s)
```

```
DELETE FROM citizen WHERE ID="{}"
```

```
INSERT INTO officer(Contact , ID , Name ) VALUES (%s,%s,%s)
```

```
DELETE FROM officer WHERE ID="{}"
```

```
INSERT INTO road(road_id , Status) VALUES (%s,%s)
```

```
DELETE FROM road WHERE road_id="{}"
```

```
SELECT * FROM road
```

Nested Queries

```
SELECT * FROM officer WHERE officer.ID NOT IN ( SELECT ID FROM
Vehicle_in);
```

```
Select * from citizen where citizen.ID NOT IN(Select ID from
Vehicle_in)
```

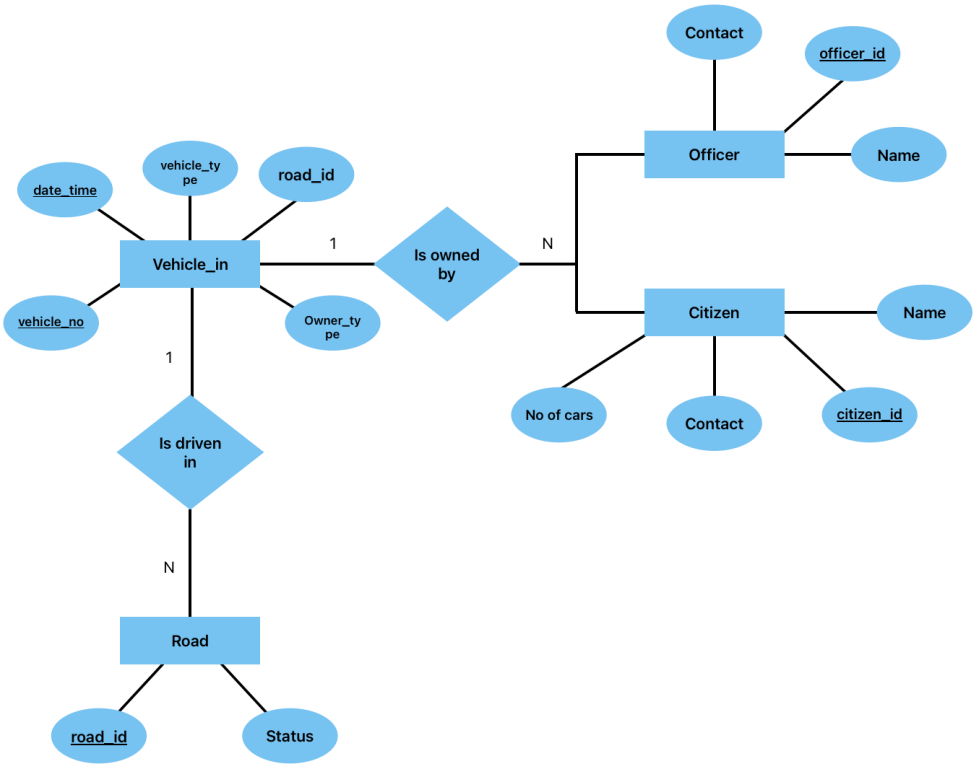
Join query

```
SELECT Vehicle_in.Date, Vehicle_in.Type, Vehicle_in.Owner_type
AS OwnerType, Vehicle_in.Vehicle_no AS VehicleNumber,
Vehicle_in.P_id AS RoadID, Vehicle_in.ID, COALESCE(officer.Name,
citizen.Name) AS Name, COALESCE(officer.Contact,
citizen.Contact) AS Contact FROM Vehicle_in LEFT JOIN officer ON
Vehicle_in.Owner_type = 'Officer' AND Vehicle_in.ID = officer.ID
LEFT JOIN citizen ON Vehicle_in.Owner_type = 'Citizen' AND
Vehicle_in.ID = citizen.ID;
```

Aggregate Query

```
SELECT count(*) as total_records,Owner_type,Date  FROM
Vehicle_in group by date,Owner_type;
```

ER DIAGRAM


Contact
officer_id
Officer
Name
vehicle_ty pe
road_id
date_time
Vehicle_in
1
Is owned by
N
vehicle_no
Owner_ty pe
Citizen
Name
1
No of cars
Contact
citizen_id
Is driven in
N
Road
road_id
Status

RELATIONSHIP SCHEMA IN THIRD NORMAL FORM