SMART TRAFFIC VIOLATION LOGGER

Name: Nithishwar T
Date of Submission: November 5, 2025
GitHub Repository: https://github.com/nithishwar17/SMART-TRAFFIC-VIOLATION-LOGGER

---

1. ABSTRACT

The **Smart Traffic Violation Logger** is a Flask-based web application developed to modernize how traffic violations are recorded and managed.
It enables traffic officers to log violations digitally by entering details such as the vehicle number, type of violation, location, fine amount, and payment status.
Each record automatically generates a unique **QR code**, which redirects to a public status page that citizens can scan to check their challan details and payment updates.

This project showcases essential Flask concepts such as routing, database integration using SQLAlchemy, QR code generation, and secure officer authentication through Flask-Login and Flask-Bcrypt. It provides a digital, transparent, and efficient alternative to the traditional manual challan process.

---

2. OBJECTIVE

- To digitalize traffic violation management using Flask and SQLite.
- To ensure secure data entry for traffic officers using login authentication.
- To generate QR codes for each violation to simplify citizen access.
- To provide a public view of violation details without login.
- To enhance transparency and reduce paperwork in traffic management systems.

---

3. PROPOSED SYSTEM

The proposed system aims to replace manual traffic challans with a **smart, database-driven platform**.
Officers can:

- Log in securely using authorized credentials.
- Add, view, and update violation records.
- Toggle payment status (Paid/Unpaid).
- Generate QR codes automatically for each violation.

Citizens can:

- Scan QR codes printed on challans.
- View violation details (vehicle number, date, location, fine, status).
- Verify payment status directly through the public page — no login required.

This approach improves efficiency, accuracy, and accessibility for both officers and citizens.

---

4. SYSTEM REQUIREMENTS

Software Requirements

- Operating System: Windows 10 / 11
- Python 3.10 or above
- Flask framework
- SQLite database
- Browser: Google Chrome / Edge
- IDE: Visual Studio Code

---

5. TECHNOLOGIES USED

Component	Technology Used
Language	Python
Framework	Flask
Database	SQLite
Frontend	HTML, CSS, Bootstrap
Libraries	Flask-Login, Flask-Bcrypt,qrcode, Pillow
Tools	Visual Studio Code, GitHub

---

6. SYSTEM ARCHITECTURE / WORKFLOW

Workflow:

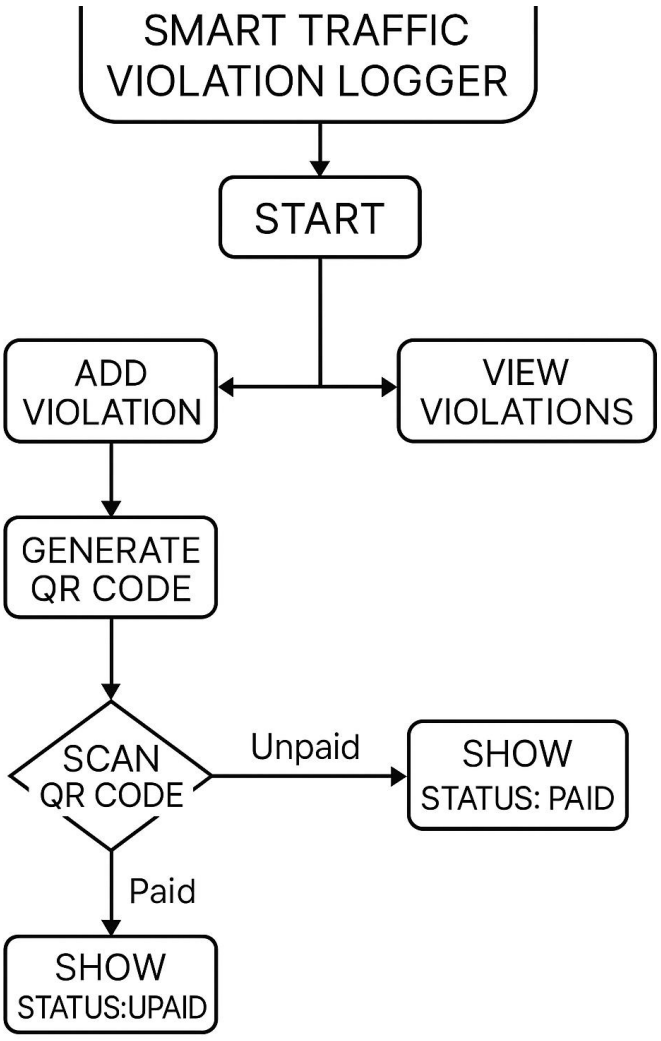
1. Officer logs into the system.
2. Adds a new traffic violation with all required details.
3. The system stores data in the database and generates a **QR code**.
4. QR code links to a public /status/<id> page showing violation details.
5. Citizens can scan the code using their mobile device.

6. Officers can toggle payment status once fines are paid.

---

7. IMPLEMENTATION STEPS

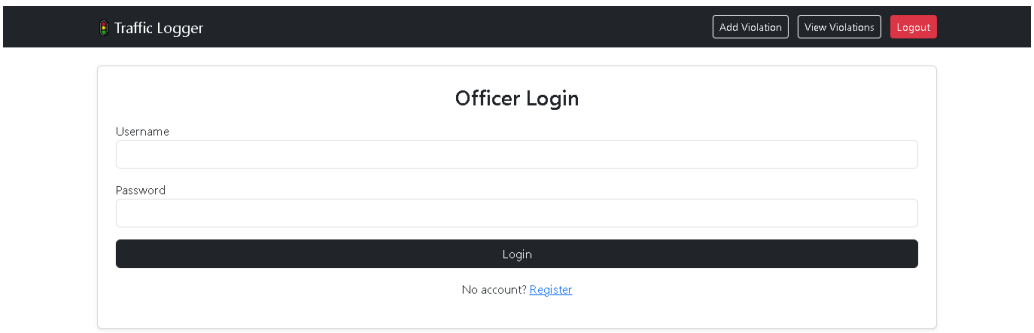
1. Initialize Flask project and create virtual environment.
2. Configure SQLite database using SQLAlchemy.
3. Create **User** and **Violation** models.
4. Implement officer authentication: register, login, and logout.
5. Build routes for adding, viewing, updating, and public status.
6. Generate and save QR codes for each record.
7. Design UI templates using Bootstrap for a responsive look.
8. Test app functionality using different users and devices.



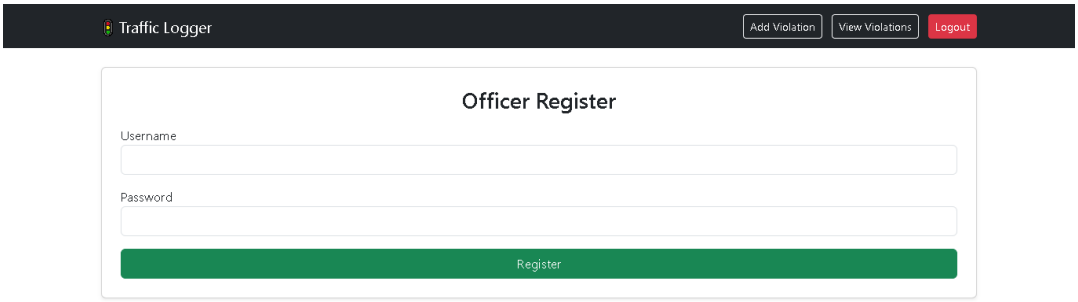
---

8. SCREENSHOTS & OUTPUT

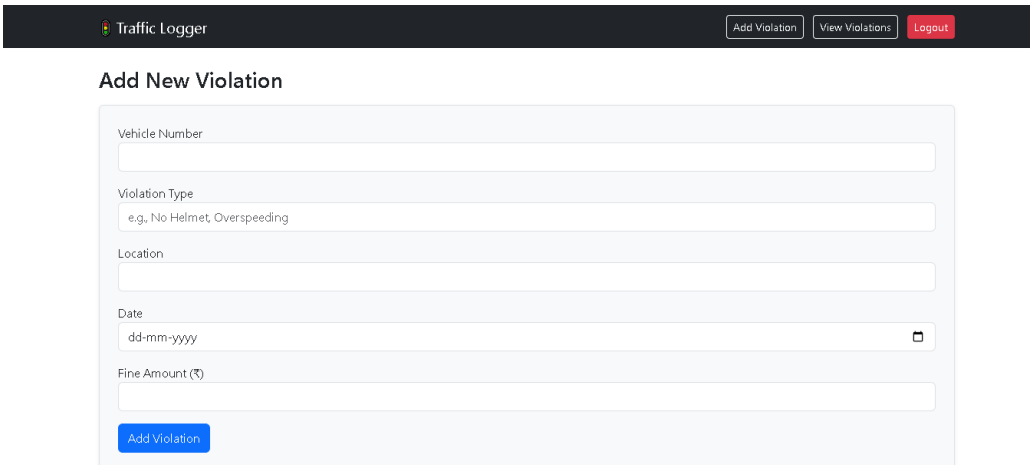
- Login Page



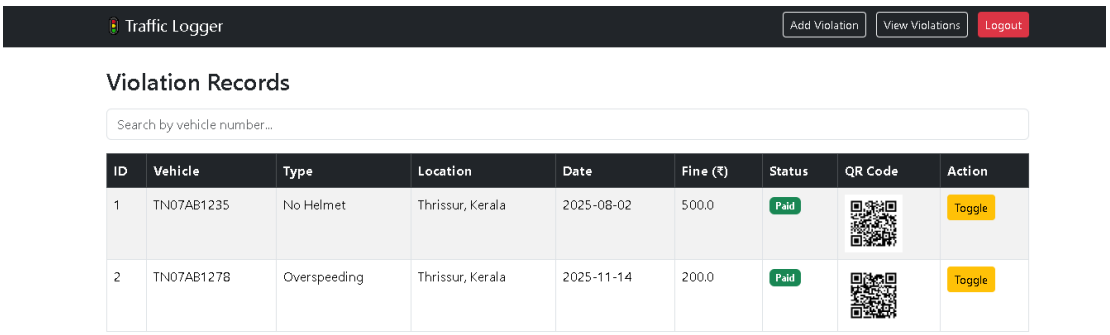
- Register Page



-
- Add Violation Form



- View Violations Page



---

9. RESULTS & DISCUSSION

The application successfully implements a digital platform for managing traffic violations.

- Officers can securely manage records through authentication.
- Public QR-based status pages enhance transparency.
- CRUD (Create, Read, Update, Delete) operations function smoothly.
- Testing confirmed stable database operations and responsive user interface.
  Overall, the system replaces manual paperwork with a user-friendly digital solution.

---

10. CONCLUSION

The **Smart Traffic Violation Logger** provides an efficient, paperless, and secure system for managing traffic violations.
It demonstrates the power of Flask and Python for real-world web applications.
By combining authentication, database integration, and QR code generation, the system offers an effective solution to modern traffic management.

---

11. FUTURE ENHANCEMENTS

- Integrate online payment gateways (Razorpay, Paytm, etc.).
- Enable SMS/email alerts to citizens after challan generation.
- Link system with the RTO database for automatic challan lookup.
- Add admin dashboard analytics and violation reports.