

E-Challan System – Project Proposal

1. Introduction

Traffic management involves monitoring vehicles, recording violations, issuing challans, managing payments, and maintaining records of vehicle owners. When handled manually, these tasks become slow, inconsistent, and error-prone.

To address this issue, we propose a **Pseudo E-Challan Generation & Traffic Violation Detection System** developed in C++ using concepts learned in this semester. The system will combine basic **computer vision**, **file handling**, and **modular programming** to automate the core workflow of challan generation, violation detection, registration, and payment processing.

2. Problem Statement

Current challan systems face several difficulties:

- Manual entry of violations
- Delays in identifying number plates
- No real-time monitoring or detection
- Higher chances of human error
- Lack of a unified digital database for challans, owners, and payments

There is a need for an **automated, file-based prototype** capable of simulating real-world traffic violation workflows, from detection to payment.

3. Proposed Solution / Methodology

We propose building a modular C++-based E-Challan System divided into clear components, each handled by a team member. The project integrates OpenCV (for detection), JSON/structs (for database storage), Model-based violation logic, and a pseudo-bank module.

System Modules

1. **OpenCV Detection Module - Violation & Challan Generation Module**
 2. **Registration Database Module**
 3. **Bank Module**
 4. **GUI + Email Notifications Module**
 5. **Individual Records Analysis Module**
-

4. Task List / Division of Responsibilities

Member 1: OpenCV Processing – Minahil Hammad (BSSE25045)

Responsibilities:

- Handle live camera feed or uploaded Videos
- Detect vehicle presence
- Extract number plate (contours + segmentation)
- Provide extracted plate text to the challan generator

C++ Concepts Required:

Pointers, dynamic memory arrays, file handling, structs, functions, Recursion, OpenCV integration.

Member 2: Database Manager – Muhammad (BSSE25002)

Responsibilities:

- Create and manage Registration Database (JSON + structs)
- Create Challan Database (JSON/text)
- Manage dynamic 2D arrays for bank/account records
- Implement search, update, and append operations

C++ Concepts Required:

OOPs Class, Pointers, dynamic memory arrays, file handling, structs, functions, Recursion.

Member 3: Finance & Bank Module – Syeda Eman Fatima (BSSE25038)

Responsibilities:

- Maintain balances
- Deduct challan amount
- Simulate transactions
- Update payment status

C++ Concepts Required:

Pointers, dynamic memory arrays, file handling, structs, functions, Recursion.

Member 4: GUI + Email Integration – Fatima Saeed (BSSE25013)

Responsibilities:

- Build program interface (console/GUI)
- Display challan history, payment info, violations
- Send automated email notifications to vehicle owners

C++ Concepts Required:

Pointers, dynamic memory arrays, file handling, structs, functions, Recursion.

Member 5: Record Analysis – Muhammad Rafay (BSSE25037)

Responsibilities:

- Analyze generated challans
- Summaries for paid/unpaid challans
- Search challans by owner E-Mail

C++ Concepts Required:

Pointers, dynamic memory arrays, file handling, structs, functions, Recursion.

5. Expected Final Results

By the end of the semester, the team will deliver:

- A fully functional C++ pseudo E-Challan System
 - Complete integration of:
 - Vehicle detection
 - Plate extraction
 - Violation identification
 - Automated challan generation
 - Email notification
 - Payment processing
 - Working pseudo-databases:
 - Registration database (JSON/structs)
 - Challan database (JSON/structs)
 - Bank/account database (JSON/structs)
 - Records database (JSON/structs)
 - Ability to store, update, delete, and retrieve records
 - Basic reports such as:
 - Vehicle owner list
 - Challan history
 - Accounts info
 - Generated Challans by System History
 - Proper C++ coding structure with modular design and comments
-

6. Conclusion / Future Directions

This project demonstrates how C++ concepts can be applied to simulate a real-world traffic enforcement system. By dividing the system into modules such as detection, database, finances, GUI, and notifications, team members will gain hands-on experience in modular programming, teamwork, and system integration.

Future Directions

- Integration with government APIs
- Real-time violation detection with IoT cameras
- License plate recognition using deep learning
- Actual banking API for online payments
- Mobile application for challan checking
- Cloud-based online challan portal
- QR-based payment receipts
- Databases Using SQL

7. References

Basics of C++:

<https://www.geeksforgeeks.org/cpp/c-plus-plus/>

Structs in C++:

https://www.w3schools.com/cpp/cpp_structs.asp

Json File Handling:

<https://www.youtube.com/watch?v=whNFPBEI-wM>

<https://www.youtube.com/watch?v=Sa8bdVogGIo>

Dynamic 2D Arrays in C++:

<https://www.geeksforgeeks.org/cpp/how-to-declare-a-2d-array-dynamically-in-c-using-new-operator/>

Basic OOP:

<https://www.youtube.com/watch?v=T1fkvT1uPj8>

OpenCV in C++:

<https://www.youtube.com/watch?v=2FYm3GOonhk>

E-Mail in C++:

https://github.com/cutelyst/simple-mail?utm_source=chatgpt.com

Qt GUI in C++:

https://www.youtube.com/watch?v=iOQAkNsbGug&list=PLBcRQYUuUM4jre_nTyKujCapKS_IcAwVsY