|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Software Requirement Specifications**  ***Version: [*1.0*]***   |  |  | | --- | --- | | Project Code | F24-204 | | Supervisor | Mr. Kashan Hussain | | Co Supervisor | - | |  |  | | Project Team | Usman Rasheed(21K-3225)  Syed Talha Minhaj(21K-3347)  Muneeb Ali(21K-4924) | | Submission Date | 12/12/2024 | |

Document History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Name of Person** | **Date** | **Description of change** |
| 1.0 | Usman Rasheed |  | Document Created |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Distribution List

|  |  |  |
| --- | --- | --- |
| **Name** | **Role** | |
| Mr. Kashan Hussain | | Supervisor |
| - | | Co- Supervisor |
|  | |  |

Document Sign-Off

|  |  |  |
| --- | --- | --- |
| **Version** | **Sign-off Authority** | **Sign-off Date** |
| 1.0 | Supervisor | 12/11/2024 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Table of Contents**

[1. Introduction 7](#_Toc184829402)

[1.1. Purpose of Document 7](#_Toc184829403)

[1.2. Intended Audience 7](#_Toc184829404)

[1.3. Abbreviations 7](#_Toc184829405)

[1.4. Document Convention 7](#_Toc184829406)

[*2.* Overall System Description 8](#_Toc184829407)

[2.1. Project Background 8](#_Toc184829408)

[2.2. Project Scope 8](#_Toc184829409)

[2.3. Not In Scope 9](#_Toc184829410)

[2.4. Project Objectives 9](#_Toc184829411)

[2.5. Stakeholders 10](#_Toc184829412)

[2.6. Operating Environment 10](#_Toc184829413)

[2.7. System Constraints 10](#_Toc184829414)

[2.8. Assumptions & Dependencies 10](#_Toc184829415)

[3. External Interface Requirements 12](#_Toc184829416)

[3.1 Hardware Interfaces 12](#_Toc184829417)

[3.2 Software Interfaces 12](#_Toc184829418)

[3.3 Communication Interfaces: 12](#_Toc184829419)

[4. Functional Requirements 13](#_Toc184829420)

[4.1. Functional Hierarchy 13](#_Toc184829421)

[4.1.1 [Use Case Diagram] 15](#_Toc184829422)

[5. Non-functional Requirements 18](#_Toc184829423)

[6. References 20](#_Toc184829424)

[7. Appendices 21](#_Toc184829425)

# Introduction

## Purpose of Document

The purpose of this Software Requirements Specification (SRS) document is to provide a detailed and structured outline of the functional and non-functional requirements for the Automated Speeding Detection System on Website. This document serves as a comprehensive guide for all stakeholders, including business users, developers, and technical teams, to ensure a shared understanding of the system’s goals, functionalities, and constraints.

* To provide a clear communication framework among stakeholders
* To act as a reference document for developers during the system’s design, implementation, and testing phases.
* To ensure that the system’s design and implementation align with the business objectives
* To establish a foundation for requirement traceability throughout the project lifecycle

## Intended Audience

* Fast NU
* Jury
* Supervisor (Kashan Hussain)
* Students of Fast NU
* Our Team(Designer, Developer, Tester)
* Potential Users of this product

## Abbreviations

* MERN = MongoDB, Express, Node.js, React.js
* IR = Infrared
* AI = Artificial Intelligence
* OCR = **Optical Character Recognition**

## Document Convention

* Font Family = Arial
* Font Size = 12 for headings, 10 for the rest of the content

# Overall System Description

## Project Background

In high-traffic areas with frequent, repeat vehicle visits such as universities, offices, and gated communities, overspeeding poses a significant safety risk. Existing systems either lack automation or rely on manual intervention, leading to inefficiencies in detecting, recording, and penalizing overspeeding drivers. Our Project aims to automate speed monitoring, license plate recognition, and fine management processes by combining technologies such as IR sensors, computer vision, OCR, and a MERN stack for data management and visualization to address those gaps.

**Problem Statement:**

In Pakistan overspeeding is a persisting problem especially in environments where repeated vehicle traffic (such as Universities, Offices, and gated communities), in these areas traditional countermeasures are either non-existent or are prone to intense labor work, are error-prone, and are susceptible to corruption. Traditional methods also fail to perform robustly, are not automated for fine management or license plate recognition, and often result in many missed violations. The project aims to create a system that would solve all these issues and be convenient to set up.

## Project Scope

This project aims to provide an automated solution for detecting overspeeding, accurate license plate recognition, and streamlining fine management to ensure a safer and more efficient system for high-traffic environments with repeated users.

**Included Functionalities:**

1. **Overspeeding Detection**:
   * Measure vehicle speed using IR sensors.
   * Detect overspeeding vehicles and log their details.
2. **License Plate Recognition**:
   * Use computer vision and EasyOCR to identify and record license plates.
3. **Data Logging and Notification**:
   * Store overspeeding data (number plate, speed, timestamp, license plate image) in a database.
   * Send email notifications to registered users with fine details.
4. **Admin Management via Webpage**:
   * Add new users (number plates, names, emails) to the system.
   * Delete user records.
   * View overspeeding records and fine payment statuses.
   * Update fines manually for cash payments.
5. **Unregistered User Handling**:
   * Record unregistered users' details for future reference.

**Excluded Functionalities**:

* manual verification of FYP status posted by the employee

## 2.3. Not In Scope

**1. Vehicle Model Identification:**

o The system will not identify or classify vehicles by type or model.

**2. Advanced Analytics:**

o Insights like driver behavior analysis or speed patterns over time.

**3. Cross-regional User Data Sharing:**

o The system will not share user data across multiple deployments or regions**.**

## 2.4. Project Objectives

**Admin Functions:**

* Add, update, and delete user data, including vehicle number plates, owner names, and emails, through the admin interface.
* View detailed overspeeding records, including speed values, timestamps, and license plate images, for all vehicles.
* Update fine payments manually for individuals paying in cash.
* Maintain system functionality and ensure compliance with data privacy and traffic regulations.

**Vehicle Monitoring Functions:**

* Detect vehicle speed in real-time using IR sensors.
* Identify vehicles exceeding the speed limit and log details (number plate, speed value, timestamp, and license plate image) into the database.
* Classify users as "Registered" or "Unregistered" based on the database and update records accordingly.

**Notification and Fine Management Functions:**

* Send email notifications to registered vehicle owners with fine details, including payment options and deadlines.
* Record and manage the status of fine payments for each violation.

**Web Interface Functions:**

* Display a list of overspeeding records with associated details, such as timestamps and fines, on a user-friendly webpage.
* Provide insights on payment statuses, allowing the admin to track who has paid their fines.
* Offer an intuitive interface for managing registered and unregistered users' data.
* These objectives ensure the system automates overspeed detection and fine management while providing admins with the tools to manage user data efficiently

## 2.5. Stakeholders

* System Admin (For the Organization)
* Registered Vehicle Owners (Employees, Students, Residents)
* Unregistered Vehicle Owners
* Administrative Staff (Managing Database and Fines)
* Traffic Authorities / Law Enforcement
* Development Team
* External Service Providers (Email Notification System, OCR Model Providers)

## Operating Environment

Anyone that uses the system must be aware that the project is broken into 4 modules, the user must ensure proper guidelines are followed for setting up each module:

* The Hardware Module: The user must ensure that the hardware module has an OS that can run python and can connect to an Arduino and a camera, the settings for the Arduino must ensure that the speed is set according to the requirements of that particular area, the user must ensure the camera and the IR Sensors are placed such that they can perform their tasks respectively when it comes to capturing speed and license plates.
* The AI Server: The user must ensure all API’s such as gmail’s smtp api, MongoDB’s Api, and the apis of other tools used in the project is set up correctly. The user must ensure to upload the Ai codebase to a server that is able to handle their specific needs, and ensure the model is installed.
* Database: User must ensure to enter relevant data according to their domain is entered with proper rights given to proper users, if they choose to update any documents then they have to update all other code accordingly.
* Website: The user must ensure that they have a stable internet.

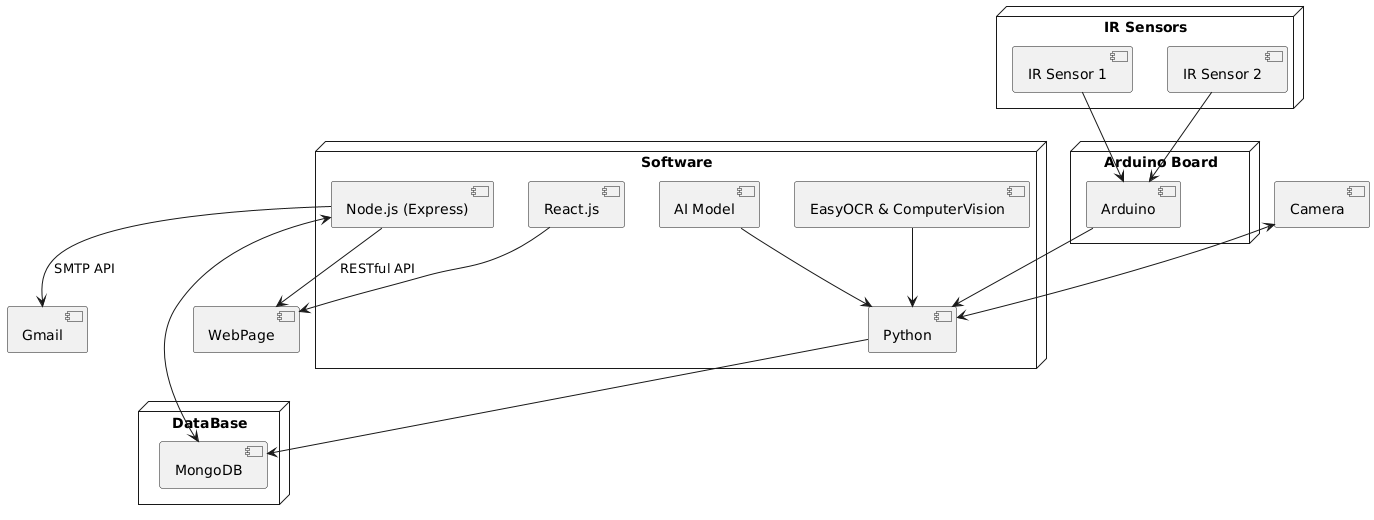
## System Constraints

* The IR sensors and camera must be positioned in locations where they can effectively detect vehicle speed and capture clear images of license plates.
* The Sensors and camera must be placed with their range in mind
* The Camera connected has a high display quality and a high frame rate
* The speed limit of the location is configured properly by the admin.
* The system requires a stable internet connection for communication between the project components.
* The AI Servers processes can process the license plate detection effectively.
* The Gmail, MongoDB and other relevant API keys are set up properly by the system administrator.
* The Hardware of the system is connected to a proper power supply.
* The system may require maintenance in the case of any hardware damage

## Assumptions & Dependencies

* Any vehicle that passes through the system has a license plate on it.
* Every user has provided correct details
* The distance between each IR Sensor has been set up correctly
* The System Admin correctly enters paid fines
  + 1. **Dependencies**
* Any vehicle that passes through the system has a license plate on it.
* Every user has provided correct details
* The distance between each IR Sensor has been set up correctly
* The System Admin correctly enters paid fines

# External Interface Requirements



## 3.1 Hardware Interfaces

* 2 IR Sensors
* Arduino Board
* A Camera
* Any Computer or laptop

## 3.2 Software Interfaces

* MongoDB
* React.js
* Node.js (with express setup)
* Python
* The AI Model
* SMTP API
* EasyOCR and ComputerVision

## 3.3 Communication Interfaces:

* The COM Port is connected for the arduino to connect with the python script
* The IR Sensors connect to the arduino through the relevant pins
* The python script connects to the arduino directly
* RESTful apis will be used to transfer data through and from the database
* The SMTP API will send emails through Gmail

# Functional Requirements

## 4.1. Functional Hierarchy

**1. Speed Detection and Overspeed Logging**

**1.1. Speed Detection**

* 1.1.1. Detect vehicle speed in real-time using IR sensors.  
  **1.2. Overspeed Logging**
* 1.2.1. Identify vehicles exceeding the speed limit.
* 1.2.2. Log overspeeding details, including:
  + 1.2.2.1. Number plate.
  + 1.2.2.2. Speed value.
  + 1.2.2.3. License plate image.
  + 1.2.2.4. Timestamp.

**2. License Plate Recognition**

* **2.1. Optical Character Recognition (OCR)**
* 2.1.1. Use computer vision and EasyOCR to read license plates.
* 2.1.2. Log recognized license plate details into the database.

**3. Database Management**

**3.1. Overspeeding Records**

* 3.1.1. Store overspeeding records, including registered and unregistered users.  
  **3.2. User Classification**
* 3.2.1. Classify users as "Registered" or "Unregistered."
* 3.2.2. Maintain registered user details:
  + 3.2.2.1. Number plate.
  + 3.2.2.2. Name.
  + 3.2.2.3. Email.

**4. Admin Features**

**4.1. User Management**

* 4.1.1. Add new user details (number plates, names, and emails).
* 4.1.2. Delete user records.  
  **4.2. Record Management**
* 4.2.1. View overspeeding records.  
  **4.3. Payment Updates**
* 4.3.1. Update fine payments for manual cash transactions.

**5. Notification System**

**5.1. Email Notifications**

* 5.1.1. Send email notifications to license plate holders for fines.
* 5.1.2. Include payment details and options in the email.

**6. Payment Tracking**

**6.1. Fine Status Management**

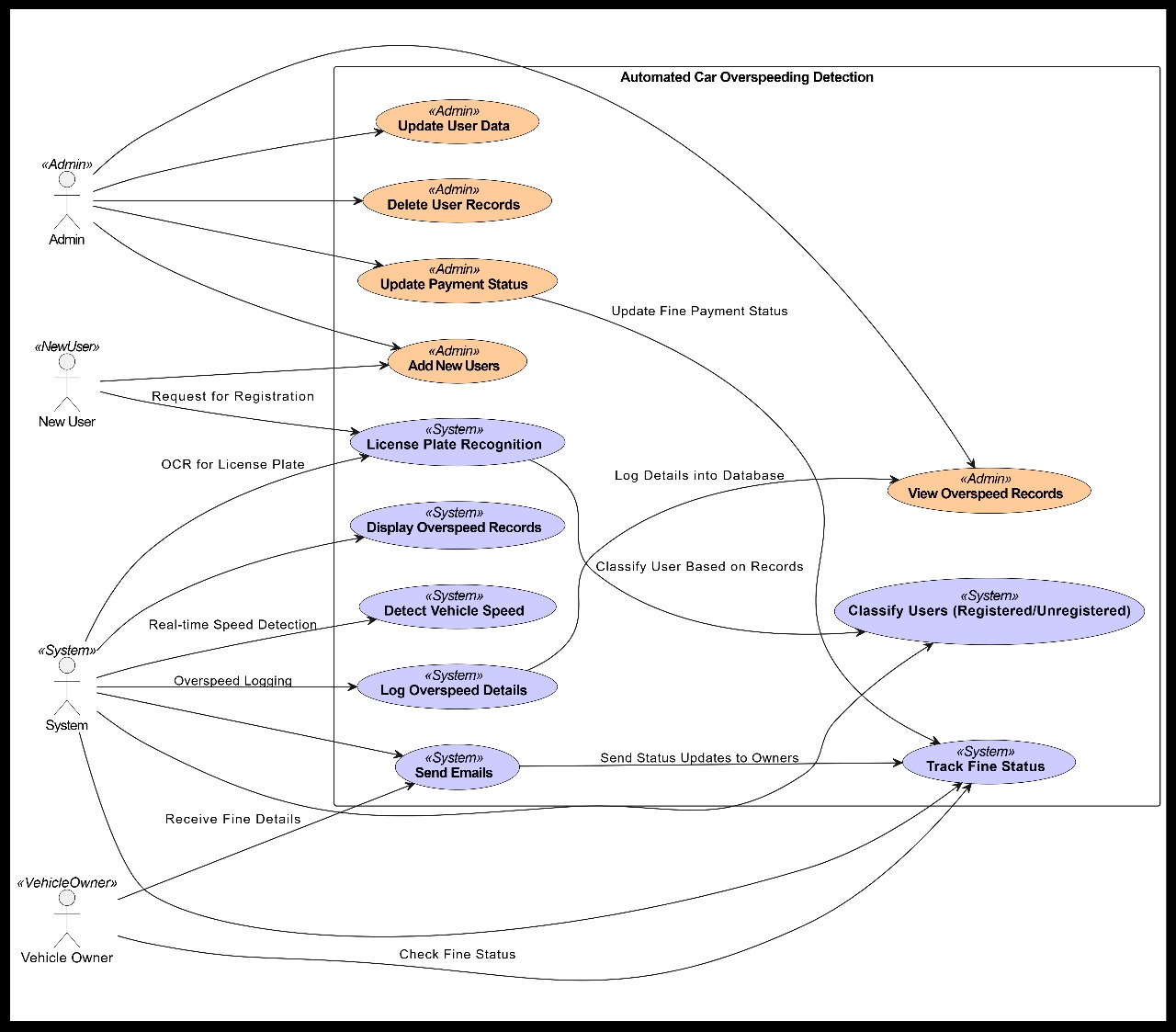
* 6.1.1. Maintain a status of fine payments.
* 6.1.2. Display fine payment statuses on the webpage.

**7. Webpage Features**

**7.1. Record Display**

* 7.1.1. Display overspeeding records with timestamps.
* 7.1.2. Display user information for both registered and unregistered users.  
  **7.2. Fine Management Interface**
* 7.2.1. Provide an interface for tracking fine payment statuses.

## 4.1.1 [Use Case Diagram]



***Admin - User Management***

|  |  |  |
| --- | --- | --- |
| Use Case Id: | UC-AU-01 | |
| Actors: | Admin | |
| Feature: | Add, Update, and Delete User Records | |
| Pre-condition: | Admin must log in successfully. | |
| Scenarios |  | |
| Step# | **Action** | Software Reaction |
| 1 | Admin selects "Manage Users" from the menu. | System displays a list of existing users and options to add, update, or delete users. |
| 2 | Admin selects "Add User" and enters details (e.g., name, email, license plate). | System validates the input for completeness and duplicate entries. |
| 3 | Admin submits the form. | System saves the new user record and confirms the addition. |
| Alternate Scenarios: | 1a: If required fields are missing, the system highlights the missing fields with error messages.  2a: If the vehicle plate number is already registered, the system notifies the admin and prevents duplication. | |
| Post Conditions: | | |
| Step# | **Description** | |
|  | User details are successfully added, updated, or deleted in the database. | |
| Use Case Cross-Referenced: | UC-OS-01: Overspeed Logging, UC-VO-01: Receive Fine Notification. | |

***Admin - Fine Payment Update***

|  |  |  |
| --- | --- | --- |
| Use Case Id: | UC-AU-02 | |
| Actors: | Admin | |
| Feature: | Update Fine Payment Status | |
| Pre-condition: | Admin must have access to payment records. | |
| Scenarios |  | |
| Step# | **Action** | Software Reaction |
| 1 | Admin selects "Fine Management" from the menu. | System displays a categorized list of fines (e.g., Paid, Unpaid). |
| 2 | Admin selects an unpaid fine to view details. | System displays the fine details, including timestamp and amount. |
| 3 | Admin marks the fine as "Paid." | System updates the payment status and reflects it in the records. |
| Alternate Scenarios: | 1a:If the database connection fails, an error message is displayed, prompting the admin to retry later. | |
| Post Conditions: | | |
| Step# | **Description** | |
|  | Fine payment status is successfully updated in the database. | |
| Use Case Cross-Referenced: | UC-VO-01: Receive Fine Notification, UC-OS-01: Overspeed Logging. | |

***Vehicle Owner - Receive Fine Notification***

|  |  |  |
| --- | --- | --- |
| Use Case Id: | UC-VO-01 | |
| Actors: | Vehicle Owner | |
| Feature: | Receive Fine Notification | |
| Pre-condition: | Vehicle owner is registered and has committed an overspeeding violation. | |
| Scenarios |  | |
| Step# | **Action** | Software Reaction |
| 1 | System sends a fine notification email to the owner. | Email contains details like timestamp, speed, fine amount, and payment options. |
| 2 | Vehicle owner reviews the email. | Owner sees detailed instructions for fine payment. |
| Alternate Scenarios: | 1a: If email delivery fails, the system retries sending the email or notifies the admin. | |
| Post Conditions: | | |
| Step# | **Description** | |
|  | Fine notification is successfully delivered to the vehicle owner. | |
| Use Case Cross-Referenced: | UC-AU-02: Update Fine Payments, UC-OS-01: Overspeed Logging. | |

***System - Speed Detection***

|  |  |  |
| --- | --- | --- |
| Use Case Id: | UC-SYS-01 | |
| Actors: | System | |
| Feature: | Detect Speed of Vehicles | |
| Pre-condition: | IR sensors and speed detection system must be operational. | |
| Scenarios |  | |
| Step# | **Action** | **Software Reaction** |
| 1 | A vehicle passes through the IR sensor's detection zone. | System measures the vehicle's speed in real-time. |
| 2 | System compares the measured speed with the set speed limit. | If speed exceeds the limit, an overspeed record is created. |
| Alternate Scenarios: | 1a: If IR sensors malfunction, the system logs an error and notifies the admin. | |
| Post Conditions: |  | |
| Step# | **Description** | |
|  | Speed data is logged, and overspeeding records are created for violators. | |
| Use Case Cross-Referenced: |  | |
| - | UC-OS-01: Overspeed Logging, UC-SYS-02: License Plate Recognition. | |

***System - License Plate Recognition***

|  |  |  |
| --- | --- | --- |
| **Use Case Id:** | **UC-SYS-02** | |
| **Actors:** | System | |
| **Feature:** | License Plate Recognition using OCR | |
| **Pre-condition:** | Vehicle must pass in the detection zone for OCR to capture the plate. | |
| **Scenarios** |  | |
| **Step#** | **Action** | **Software Reaction** |
| 1 | System captures the vehicle's license plate image. | OCR processes the image to extract license plate details. |
| 2 | OCR reads and saves the license plate data to the database. | System associates the data with the overspeeding record. |
| **Alternate Scenarios:** | 1a:If OCR fails, the image is flagged for manual verification. | |
| **Post Conditions:** | | |
| **Step**# | **Description** | |
|  | License plate details are stored in the database and linked to the overspeeding record. | |
| **Use Case Cross-Referenced:** | UC-OS-01: Overspeed Logging, UC-AU-01: Admin Updates User Records. | |

# Non-functional Requirements

* 1. ***Performance Requirements***
* The system should detect and log overspeeding with minimal delay
* The system should handle multiple data without any performance issues
* The system should be able to log and retrieve results in under 2 seconds
* The Database should be able to support 10,000 records
  1. ***Safety Requirements***
* The IR Sensors and camera should be placed such that they do not face any damage
* Data should be kept encrypted to prevent unauthorized access
* The system should include error handling in case of any malfunctions
  1. ***Security Requirements***
* All Data including license plate details and user information must be encrypted
* Only those with administrator privileges should have access to view the database and update it
* The system must be protected from unauthorized access such as SQL injections
* The Gmail API key should be securely stored
  1. ***User Documentation***
* A Detailed user manual should be created outlining how to operate and setup the system
* Instructions should be made for the administrator on how to add, remove and manage users and update fines
* A Technical document for system installation and configuration for all requirements and dependencies

# References

* Ali, S., & Singh, R. (2021). **AI-Powered Traffic Monitoring and Law Enforcement Systems**. *Journal of Intelligent Transportation Systems*, 13(2), 89-104.
* Jain, R., & Srivastava, D. (2021). **Challenges in Automated Number Plate Recognition for Traffic Systems**. *International Journal of Smart Transportation*, 11(3), 67-82.
* Khan, M., & Ahmed, S. (2022). **The Role of Smart Traffic Systems in Urban Management**. *International Journal of Urban Planning*, 14(4), 134-149.
* Kumar, P., & Verma, S. (2020). **Technological Advancements in Speed Detection Systems for Smart Cities**. *IEEE Transactions on Smart Systems*, 15(3), 76-89.
* Nguyen, H., & Patel, R. (2019). **Real-time Traffic Monitoring Using Radar and ANPR**. *IEEE Journal of Transportation Technology*, 12(1), 98-112.
* Sharma, K., Verma, R., & Gupta, A. (2020). **Deep Learning Techniques in Automated Number Plate Recognition**. *International Journal of Computer Vision Applications*, 13(2), 45-58.
* Zhao, Y., Liu, J., & Chen, H. (2021). **Advanced ANPR Systems Using Convolutional Neural Networks**. *Journal of Transportation Engineering*, 148(1), 345-359.

# Appendices