Tribhuvan University

Academia International College

Final Year Project Report

On

Toll Collection System

Based on Number Plate Recognition System

[CSC 412]

Under the supervision of

"Mr. Anup Shrestha"

Submitted by

Kushala Shrestha (T.U. Exam Roll No. 29013/078)

Sadik Karki (T.U. Exam Roll No. 29026/078)

Submitted to

Department of Computer Science and Information Technology

Academia International College

Institute of Science and Technology

Tribhuvan University

September, 2025

Tribhuvan University

Academia International College

Final Year Project Report

On

Toll Collection System based on Number Plate Recognition System

[CSC 412]

A final year project submitted in partial fulfillment of the requirement for the degree of Bachelor of Science in Computer Science and Information Technology awarded by Tribhuvan University

Submitted by

Kushala Shrestha (T.U. Exam Roll No. 29013/078)

Sadik Karki (T.U. Exam Roll No. 29026/078)

Submitted to

Department of Computer Science and Information Technology

Academia International College

Institute of Science and Technology

Tribhuvan University

September, 2025

Tribhuvan University

Institute of Science and Technology

Academia International College

Department of Computer Science and Information Technology

Email: mail@academiacollege.edu.np

Supervisor's Recommendation

I hereby recommend that the project work report prepared under my supervision by Miss. Kushala Shrestha (29013/078) and Mr. Sadik Karki (29026/078) entitled "Toll Collection System based on Number Plate Recognition System" be accepted as fulfilling in partial requirements for the degree of Bachelors of Science in Computer Science and Information Technology. In my best knowledge, this is an original work in Computer Science and Information Technology.

……..

Mr. Anup Shrestha
Project Supervisor
Department of Computer Science and Information Technology
Academia International College
Gwarko, Lalitpur

Tribhuvan University

Department of Computer Science and Information Technology

Academia International College

Certificate of Approval

This is to certify that this project prepared by Miss. Kushala Shrestha and Mr. Sadik Karki entitled "Toll Collection System based on Number Plate Recognition System" in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Information Technology has been well studied. In our opinion, it is satisfactory in the scope and quality as a project for the required degree.

…….. Mr. Anup Shrestha Project Supervisor Department of Computer Science and IT Academia International College	…….. Mr. Bishwas Mathema HOD/Program Coordinator Department of Computer Science and IT Academia International College
…….. Internal Examiner Academia International College	…….. External Examiner Central Department of CSIT Tribhuvan University

Acknowledgement

We owe our most profound appreciation to Academia International College for giving us a chance to work on this project as part of our syllabus.

Special thanks to our supervisor, Mr. Anup Shrestha, for his consistent guidance, support, and feedback throughout the report's creation. We are generously obligated to him for providing this excellent opportunity to expand our knowledge. It helped us a lot to realize what we studied for.

We would like to express our sincere gratitude to all those individuals, families, friends, colleagues, and teachers for supporting and helping us a lot in finalizing this project within the limited time frame by providing valuable insights and feedback on the report.

Thanking You,

Kushala Shrestha (T.U. Exam Roll No. 29013/078)
Sadik Karki (T.U. Exam Roll No. 29026/078)

Abstract

This project presents the design and development of an Toll Collection System based on Automatic Number Plate Recognition (ANPR) tailored for Nepal, where number plates appear in both embossed English characters and Devanagari script. The system aims to automate toll fee collection, reduce human involvement, and minimize traffic congestion at toll booths. A two-stage pipeline is employed: YOLOv8n (nano) is used for number plate detection, followed by EasyOCR for character recognition. The system is implemented using the Django web framework with SQLite as the database, providing a secure platform where users can pre-register their vehicles, maintain a digital balance, and view transaction histories. Experimental results demonstrate high detection performance, achieving a mean Average Precision (mAP@0.5) of 96% and mAP@0.5:0.95 of 82%, with precision and recall exceeding 90%. OCR results indicate an accuracy of approximately 88% for embossed English plates and 77% for Devanagari plates, revealing the need for larger datasets and fine-tuned OCR models to improve performance on complex Nepali scripts. System testing confirmed smooth end-to-end functionality, including automated toll deduction, database updates, and error handling for unregistered or unreadable plates.

Keyword: Automatic Number Plate Recognition (ANPR), YOLOv8, EasyOCR, Devanagari Script, Toll Collection System, Object Detection, Optical Character Recognition (OCR), Django, Real-Time Processing.

Table of Content

List of Figures

List of Tables

List of Abbreviations

CNN	Convolutional Neural Network
LPR	License Plate Recognition
OCR	Optical Character Recognition
OD	Object Detection
UI	User Interface
YOLO	You Only Look Once

Chapter 1 : Introduction

1.1 Introduction

This project is a web-based Toll Collection System based on Number Plate Recognition System built using the Django framework with SQLite as the database. The system is designed to automatically detect and recognize vehicle number plates from either a live camera feed or an uploaded image and deduct the toll fee from the user's preloaded balance. It provides a complete end-to-end toll solution where users must pre-register their vehicle details, sign in securely, and maintain a sufficient account balance to use the service. When a vehicle arrives at the toll booth, its number plate is captured, recognized using a YOLOv8 detection model and EasyOCR, and verified against the database. If the plate matches a registered vehicle, the toll amount is deducted automatically, the database is updated, and the gate is opened for the vehicle to pass.

The system includes two types of user roles: Admin and Registered User. The user dashboard allows vehicle owners to check their remaining balance, view transaction history, and recharge their account. The admin dashboard provides full control over the system, including user management, transaction monitoring, and manual number plate entry in case of recognition failure. By automating the entire process, this project reduces manual work, ensures secure and accurate toll fee collection, and improves the overall speed and efficiency of toll operations.

1.2 Problem Statement

Traditional toll collection systems rely heavily on manual operation, where vehicles must stop at toll booths to make cash payments. This process is slow, causes traffic congestion, leads to unnecessary fuel consumption, and increases travel time, especially during peak hours. Manual toll systems are also prone to human errors such as incorrect toll calculation, data entry mistakes, and difficulties in maintaining accurate transaction records. To overcome these challenges, there is a need for an automated toll collection system that can detect vehicles and recognize their license plates accurately in real time. By using number plate detection and recognition, toll charges can be calculated automatically and transaction details can be stored digitally. This approach reduces manual intervention, improves processing speed, prevents revenue leakage, and ensures a smoother traffic flow at toll plazas.

1.3 Objectives

The main objectives of Automatic Toll Collection System based on Number Plate Recognition System include:

- To detect and accurately extract the characters from license plates using OCR.
- To determine the appropriate toll fee based on vehicle type and update the transaction records in real time.
- To allow toll operators or administrators to monitor vehicle entries, view payment history, and generate reports.

1.4 Scope and Limitation

1.4.1 Scope

- Store vehicle details, recognized plate numbers, timestamps, and toll amounts in a database.
- Automatically calculate toll fees based on pre-defined rates for vehicle types.
- Provide a web-based dashboard for toll operators and administrators.

1.4.2 Limitations

- The system's accuracy may decrease under poor lighting conditions, heavy rain, fog, or when license plates are dirty, damaged, or obscured.
- The system currently supports pre-defined toll rates and basic payment logging but does not include direct integration with banking/payment gateways.
- The system currently assumes one registered user per vehicle and does not handle shared ownership.

1.5 Development Methodology

The proposed Toll Collection System is developed using the Prototype Development Model, which is suitable for projects where requirements may evolve during the development process and frequent feedback is necessary. The prototype model allows for the creation of an initial working version of the system, which is then refined iteratively based on user feedback until the final system meets the desired requirements.

The development process followed these steps:

Requirement Gathering and Analysis:

Identified the functional and non-functional requirements of the toll collection system. Studied existing toll collection methods and determined key challenges, such as traffic congestion and manual data handling.

Quick Design:

Prepared an initial system design including data flow diagrams, architecture, and database schema. Designed a basic workflow for vehicle detection, number plate recognition, toll calculation, and data storage.

Prototype Development:

Developed a basic working prototype that could detect number plates and recognize characters from images and video streams. Built a minimal version of the dashboard to display recognized results and transaction logs.

User Feedback:

Presented the prototype to supervisor to gather feedback on system functionality and accuracy. Collected suggestions regarding detection accuracy, user interface improvements, and database management.

Refinement:

Improved the detection and recognition model (YOLOv8 + OCR) for better accuracy. Enhanced dashboard features for transaction search, filtering, and reporting. Optimized performance for near real-time operation.

Final Implementation:

Integrated all modules (vehicle detection, plate recognition, database operations, and dashboard). Performed system testing and validation to ensure the solution met the requirements.

This methodology ensured that the system was developed iteratively, incorporating feedback at each stage, leading to a more reliable, accurate, and user-friendly toll collection solution.

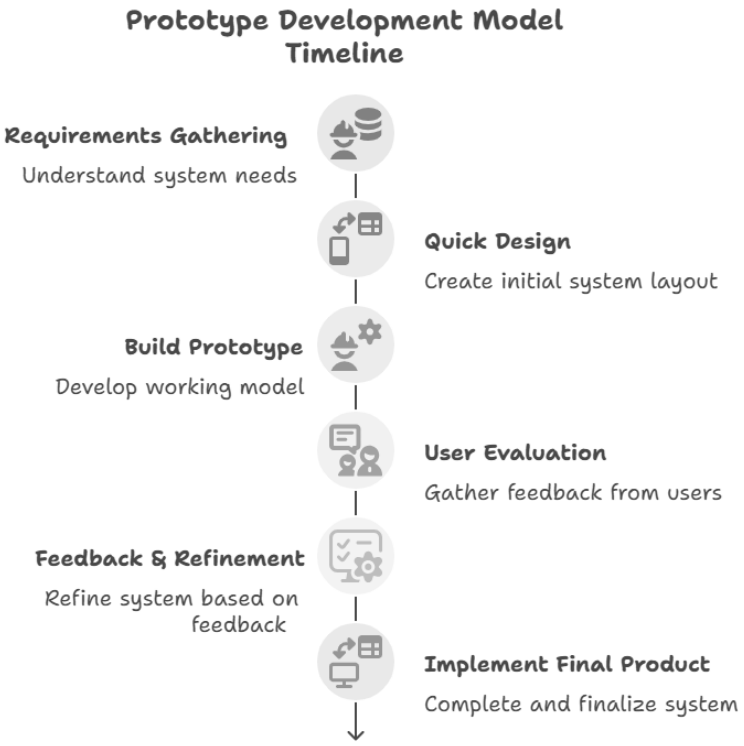

Figure 1.1: Prototype Model

1.6 Report Organization

This report begins with essential introductory sections, including the Title Page, Certificate Page, Acknowledgement, Abstract, Table of Contents, and lists of Abbreviations, Figures, and Tables.

The main body of the report is structured into six chapters, each focusing on a specific aspect of the project:

Chapter 1: Introduction

This chapter provides an overview of the project, covering key aspects such as the project introduction, problem statement, objectives, scope and limitations, and methodology.

Chapter 2: Background Study and Literature Review

Here, the project's background is discussed, along with a review of existing literature. This includes summaries of relevant projects, research papers, and articles that helped shape the project's direction.

Chapter 3: Analysis

This section delves into system analysis, including requirements and feasibility studies. It defines the system's functional requirements using a use case diagram and presents a Gantt chart to visually illustrate the timeline and progress of various project tasks.

Chapter 4: System Design

This chapter explores the design phase in detail, covering the implementation process, model architecture, user interface, and system interactions. It also includes insights into the algorithms used in the project.

Chapter 5: Implementation and Testing

The focus here is on the implementation process and testing phases. It provides an overview of the tools and dependencies used to build the system and outlines the testing procedures undertaken to ensure functionality.

Chapter 6: Conclusion and Recommendations

The final chapter wraps up the project with a summary of key findings and conclusions. It also highlights potential areas for future improvements and enhancements.

The report concludes with a References section, formatted in IEEE style, and Appendices containing system screenshots and important source code snippets.

Chapter 2 : Background Study and Literature Review

2.1. Background Study

Automatic Number Plate Recognition (ANPR) is a computer vision technology used to identify vehicles by automatically detecting and reading their license plates from images or video feeds. ANPR systems typically consist of two main stages: number plate detection and optical character recognition (OCR). Detection isolates the license plate region from the rest of the image, while OCR extracts the alphanumeric text from the plate. These systems are widely used in traffic monitoring, law enforcement, parking management, and automated toll collection because they eliminate the need for human involvement, enabling quick and accurate identification of vehicles in real time.

Toll collection systems are designed to collect fees from vehicles using certain roads, highways, or bridges, helping fund maintenance and infrastructure development. Traditional toll collection involves manual operations where vehicles stop at the toll booth, and an operator collects the fee, which can lead to delays and congestion. Over the years, toll collection has evolved from manual cash handling to semi-automated methods such as smart cards, RFID tags, and electronic toll collection systems. However, many regions still face challenges such as revenue leakage, human errors, and inefficiency in high-traffic areas. This has led to growing interest in fully automated toll systems that require minimal human intervention.

Combining ANPR technology with toll collection creates a powerful solution that is both efficient and reliable. An ANPR-based toll collection system can automatically identify vehicles, verify their details against a database, and deduct toll fees without requiring the driver to stop or interact with a toll operator. Such a system not only saves time but also improves revenue accuracy and enhances the overall user experience. Our project builds on these principles by implementing an ANPR-based toll collection solution using the Django framework and SQLite database. It supports both live video feed and image input, includes a secure user registration and balance management system, and provides an admin dashboard for monitoring transactions and managing users.

2.2. Literature Review

Al-Hasan et al. [1] presented an enhanced YOLOv8-based ANPR system optimized for edge devices like Raspberry Pi. They demonstrated that YOLOv8 can achieve real-time detection (>30 FPS) under diverse conditions, including night-time and rain. This motivated our choice of YOLOv8n for plate detection since it balances accuracy and speed, making it practical for real-time toll gate scenarios in Nepal.

Salsabila and Sriani [2] combined YOLOv8 and EasyOCR with CNN-based post-processing to improve text recognition performance. Their results revealed that although detection accuracy can approach 100%, OCR performance is highly dataset-dependent. This influenced our plan to fine-tune OCR performance, especially for Devanagari plates, and include error handling for unrecognized plates.

Dawadi et al. [3]addressed the unique challenges of Devanagari plates using IWPOD-NET and nested CNNs, reporting lower accuracy due to font variability and limited data. This study encouraged us to support Devanagari plates experimentally and plan for future dataset expansion to improve performance.

Lu et al. [4] discussed Electronic Toll Collection (ETC) systems based on GPS technology, highlighting benefits like reduced congestion, elimination of manual toll booths, and real-time transaction logging. Although their system relied on GPS rather than computer vision, it reinforced the importance of automation and cashless tolling — goals directly addressed by our project's ANPR-based approach.

Dalwadi and Amin [5] proposed an ANPR-based toll tax collection system and highlighted challenges like varying plate formats and lighting conditions. They emphasized the need for robust detection and reliable OCR to ensure fair toll deductions. This paper directly influenced our design to include confidence thresholds for detection and fallback options like manual plate verification through the admin panel.

Thosar et al. [6]introduced a blockchain-based booth-less toll system that uses GPS and image processing to ensure secure transactions. While our system does not use blockchain, their work inspired our inclusion of transaction logging and administrative dashboards to maintain transparency and allow auditing of toll payments.

Chapter 3 : System Analysis

3.1. System Analysis

This chapter explains how the system was analyzed in terms of requirements, feasibility, and modeling. The project was approached using the Object-Oriented Analysis (OOA) method due to the modular nature of its components, such as image processing, OCR, and UI interaction.

3.1.1. Requirement Analysis

The requirement can be functional and non-functional:

i. Functional Requirements

Functional requirements define what the system should do. The following are key functionalities:

- Capture live video feed, detect vehicles, detect number plates, and recognize plate text using OCR.
- Calculate toll fees based on vehicle type and deduct the amount from the user's balance automatically.
- Maintain a database of users and transaction history that can be viewed by both admin and users.

ii. Non-Functional Requirements

The project considered the following non-functional requirements:

- Process detection and recognition within 5 seconds per vehicle to avoid traffic delays.
- Secure user credentials and transaction data from unauthorized access.
- A simple, responsive, and user-friendly interface is provided.
- The system runs on any platform supporting Python (Django).
- Scalable to support multiple cameras and additional vehicle categories in the future.

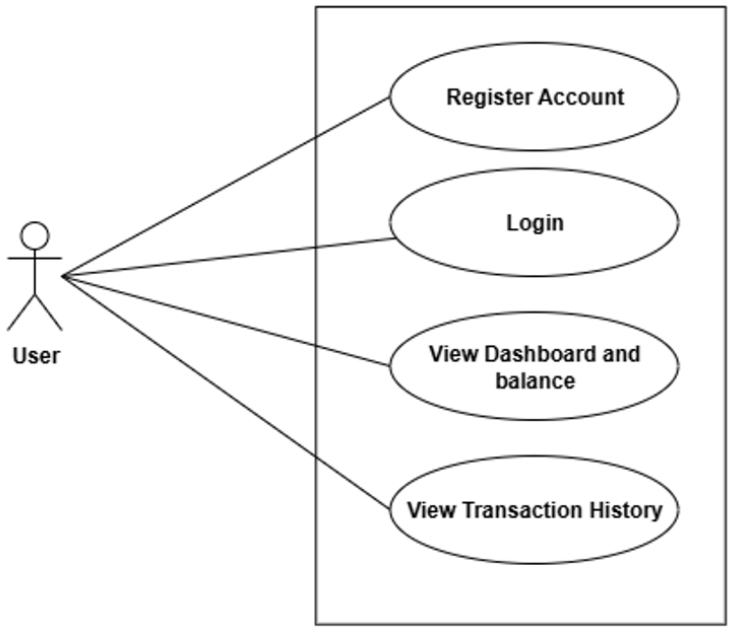

Figure 3.1: Use Case Diagram of User

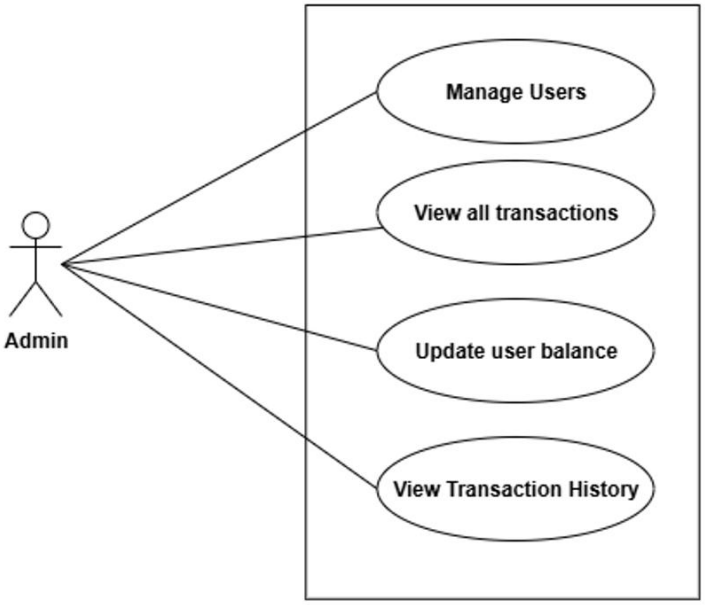

Figure 3.2: Use Case Diagram of Admin

3.1.2. Feasibility Analysis

Feasibility analysis is conducted to determine whether a project is technically, economically, operationally, and schedule-wise feasible to guarantee successful project completion.

i. Technical Feasibility

The proposed ANPR-based Toll Collection System is technically feasible because all the required tools, frameworks, and hardware are readily available and well-supported. The system is built using the Django framework, which provides a robust backend for handling user management, database operations, and API integration. SQLite is used as the database, offering a lightweight and reliable storage solution suitable for small to medium-scale applications. Since Django, SQLite, YOLO, and EasyOCR are open-source and well-documented, the technical barriers are minimal, and the system can be deployed on a standard computer or server without specialized hardware.

ii. Operational Feasibility

The system is operationally feasible as it automates the toll collection process, reducing manual intervention and ensuring smooth vehicle flow. Users can pre-register their vehicles, recharge their account balance, and have toll fees automatically deducted when passing through the toll gate. Admins have access to an intuitive dashboard that allows them to manage users, view transactions, and handle manual plate entries if the system fails to recognize a number plate. The interface is designed to be simple and user-friendly, requiring minimal training for toll operators and users. This makes the solution practical and easy to integrate into existing toll operations.

iii. Economic Feasibility

From a financial perspective, the project is cost-effective because it relies on open-source technologies, eliminating licensing costs for the software components. Hardware requirements are minimal. By reducing the need for human toll operators and minimizing revenue leakage caused by manual errors or fraudulent activities, the system offers long-term cost savings. The automated process also helps reduce vehicle wait times, indirectly saving fuel and improving customer satisfaction, which adds further economic value.

iv. Schedule Feasibility

The project is schedule-feasible within the academic timeline because it uses pre-trained models and existing libraries that significantly reduce development time. The implementation can be divided into well-defined phases, including requirement analysis, system design, module development (detection, OCR, database, payment), integration, and testing. A functional prototype can be completed within a semester, with sufficient time allocated for debugging and performance optimization. Proper task allocation and iterative development ensure that the system can be delivered within the allotted project duration.

The following Gantt chart depicts the schedule established for the development of the system.

Table 3.1: Activity Schedule

Tasks	Start Date	End Date	Duration (days)
Paper Study/Discussion	05/01/2025	05/04/2025	4
Proposal Writing	05/05/2025	05/11/2025	7
System Design	05/12/2025	05/19/2025	8
Data Collection/Preprocessing	05/20/2025	05/21/2025	2
Model training / development	05/22/2025	07/01/2025	41
System Development	07/02/2025	08/14/2025	44
Testing	08/15/2025	08/23/2025	9
Final Review/Report writing	08/24/2025	08/30/2025	7

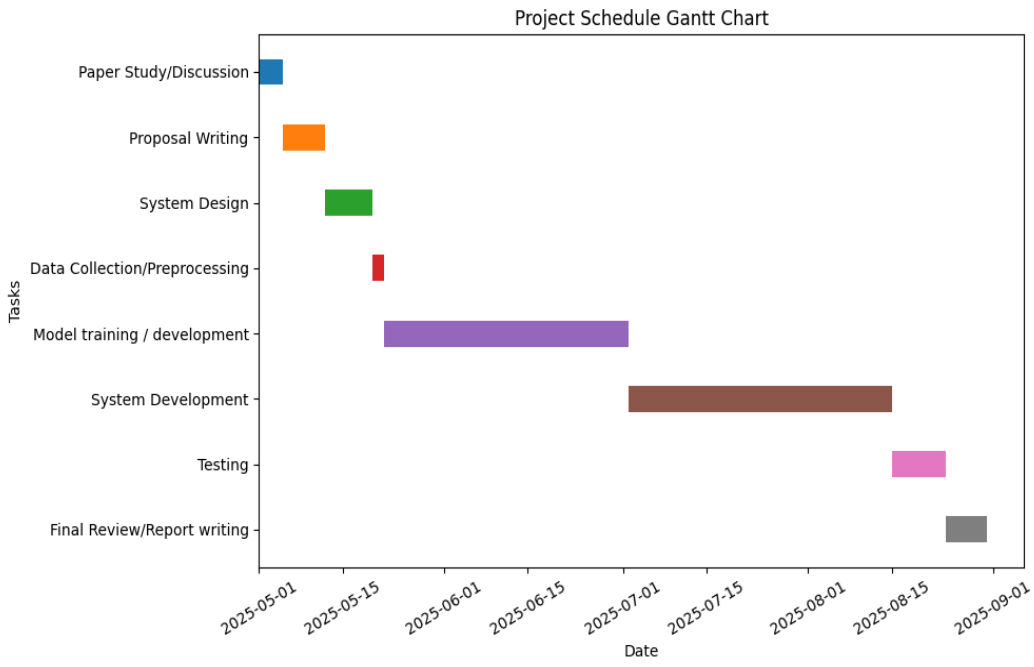

Figure 3.3: Gantt Chart

3.1.3. Object Modeling using Class and Object Diagrams

The Number Plate Recognition System for Toll collection system is inherently object-oriented due to the nature of its components and the way they interact. The system consists of distinct, independent modules such as input handling, plate detection, text recognition, and result rendering. Each of these modules can be modeled as an object with specific responsibilities and behaviors, following core object-oriented principles like encapsulation, abstraction, and modularity.

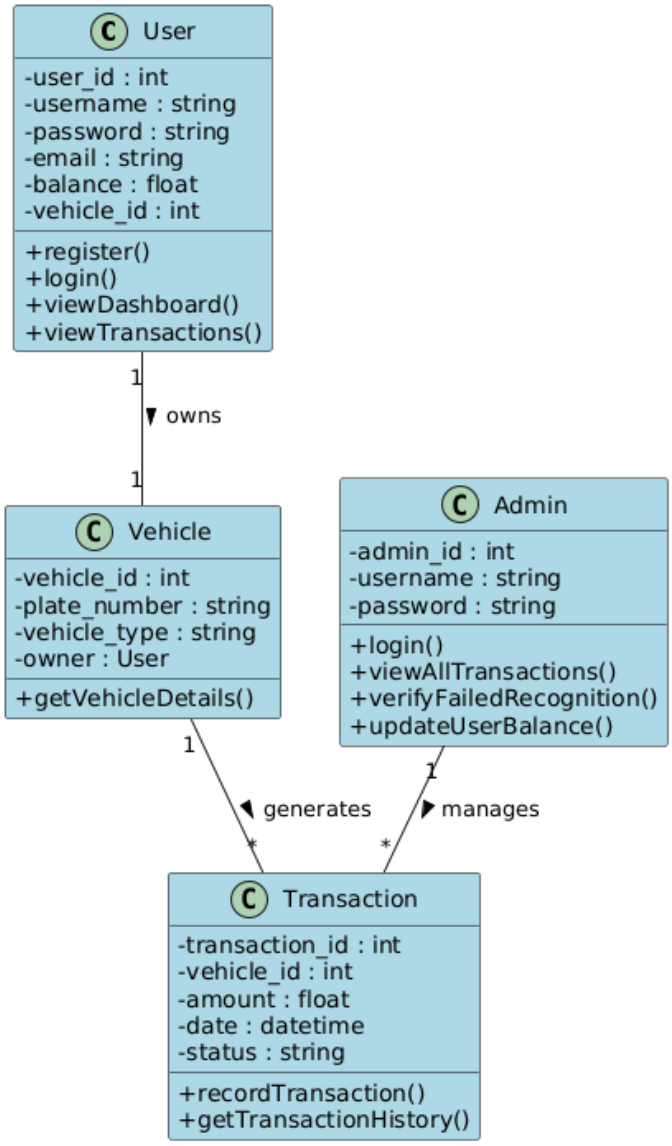

Figure 3.4: Class Diagram

This class diagram represents the foundational architecture of a Number Plate Recognition System for Toll Collection, designed to automate vehicle identification and streamline toll transactions. The system revolves around four main entities: User, Vehicle, Transaction, and Admin. Users register and manage their accounts, linking their profiles to specific vehicles. Each vehicle, identified by its plate number and type, is associated with toll transactions that record payment details, timestamps, and status.

Administrators oversee the entire process, managing transactions, verifying failed recognition attempts, updating user balances, and generating system reports. The relationships between these classes ensure a seamless flow—from plate recognition to transaction logging—supporting a robust, scalable toll collection infrastructure that minimizes manual intervention and enhances operational efficiency

3.1.4. Dynamic Modelling using Sequence Diagram

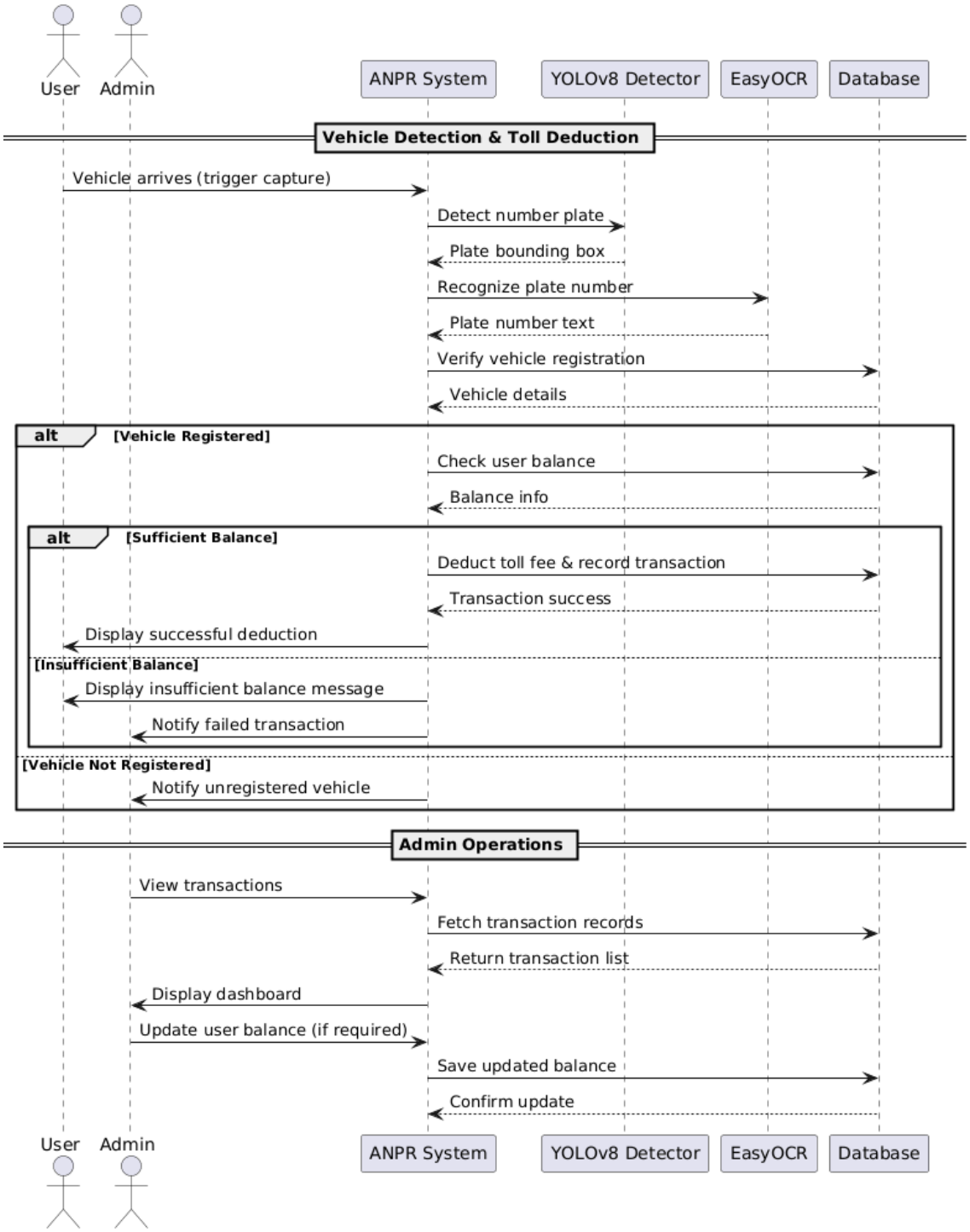

Figure 3.5: Sequence Diagram

3.1.5 Process modelling using Activity Diagrams

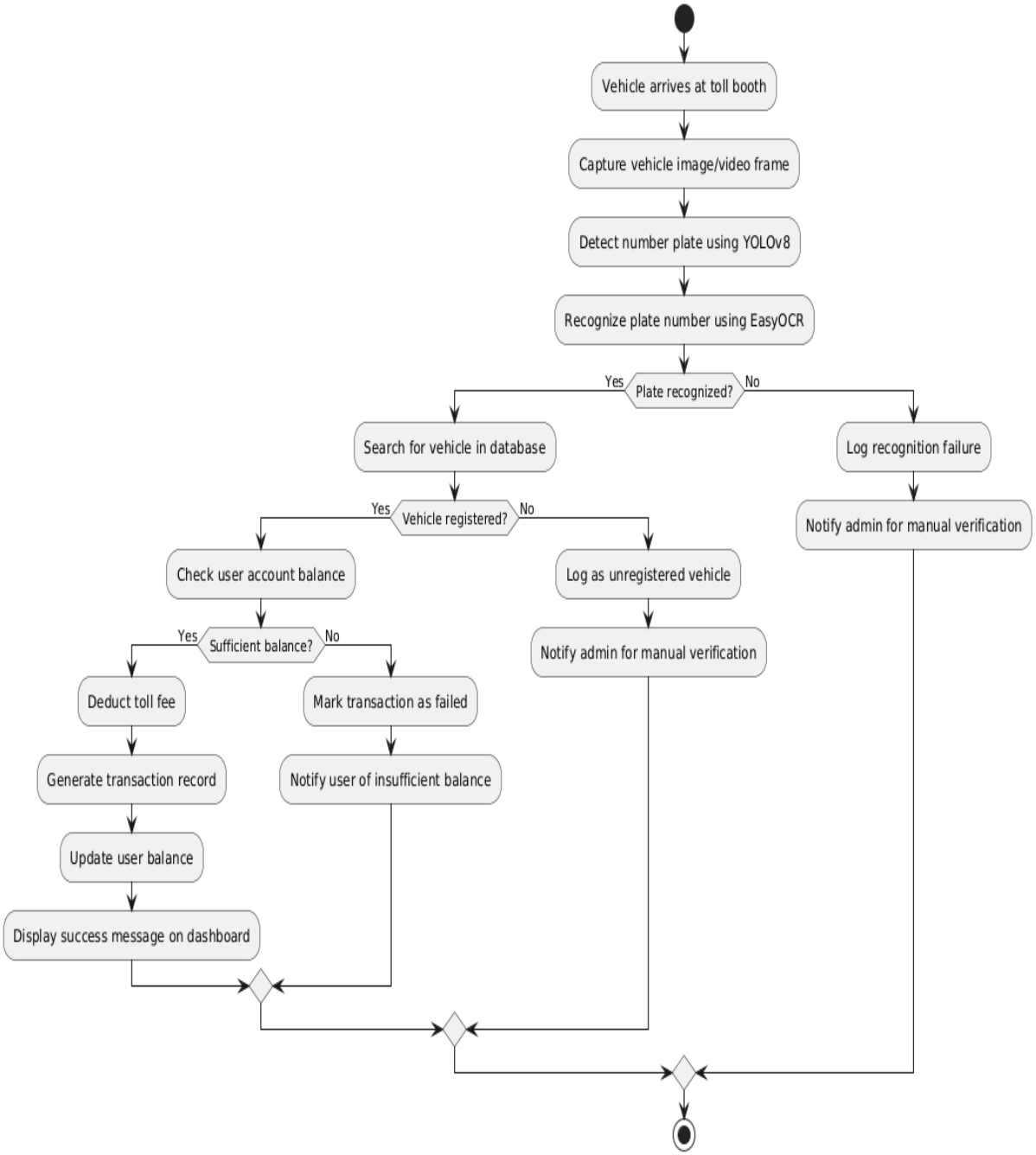

Figure 3.6: Activity Diagram

Chapter 4 : System Design

4.1. Design

As discussed in the analysis chapter, the system design also follows an object-oriented approach.

- Refinement of Class, Object and Activity Diagram

  Since this is a very basic project with not much complexity, the class diagrams, object diagrams and activity diagrams designed earlier are already refined enough to be applicable within the project.
- Component Diagram

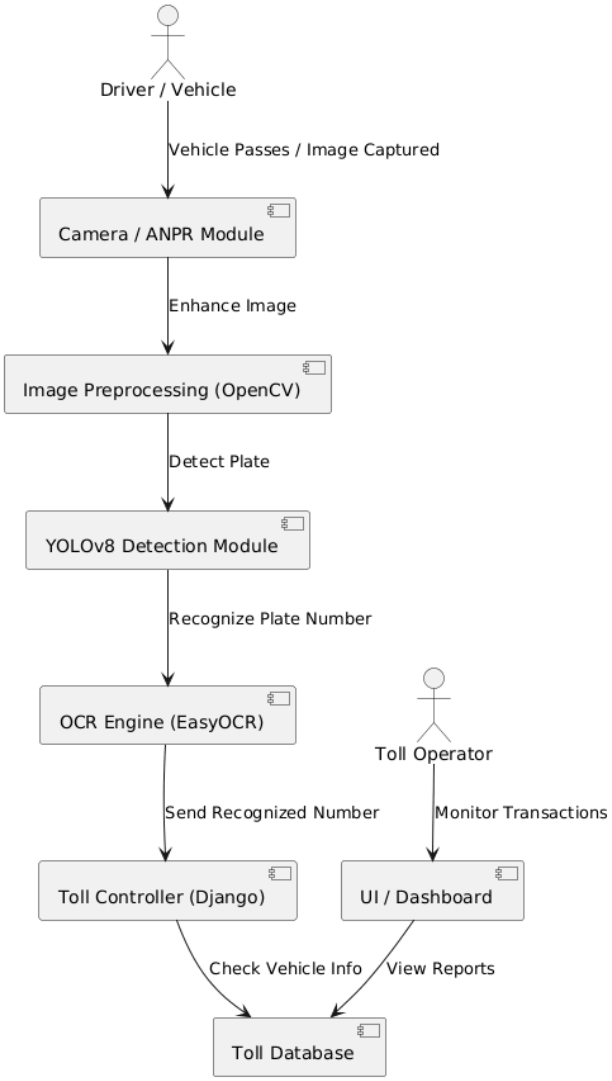

Figure 4.1: Component Diagram

4.2. Algorithm Details

This project consists of two core algorithms: one for detecting license plates from media inputs and another for recognizing text from the detected plate region. Both algorithms are supported by modern deep learning architectures that enable accurate and efficient number plate recognition under real-world conditions.

4.2.1. License Plate Detection – YOLOv8

The system uses YOLOv8 (You Only Look Once, version 8), a state-of-the-art object detection algorithm developed by Ultralytics. [7]YOLOv8 performs single-shot detection, meaning it detects objects in one forward pass, which makes it suitable for real-time applications.

Internal Architecture of YOLOv8:

- Backbone – CSPDarknet:
  CSPDarknet (Cross Stage Partial Darknet) is used as the feature extractor. It improves learning capability and reduces computational cost by combining residual connections and partial layer splitting.
- Neck – FPN + PAN:
  The Feature Pyramid Network (FPN) and Path Aggregation Network (PAN) are used to combine features from different layers, enabling the model to detect plates of various sizes and scales.
- Head – Decoupled Head:
  YOLOv8 uses a decoupled head that separates the tasks of object classification and bounding box regression, which improves accuracy and training stability.
- Anchor-Free Design:
  Unlike earlier YOLO versions, YOLOv8 uses an anchor-free approach, which simplifies object detection and reduces hyperparameter tuning complexity [8].

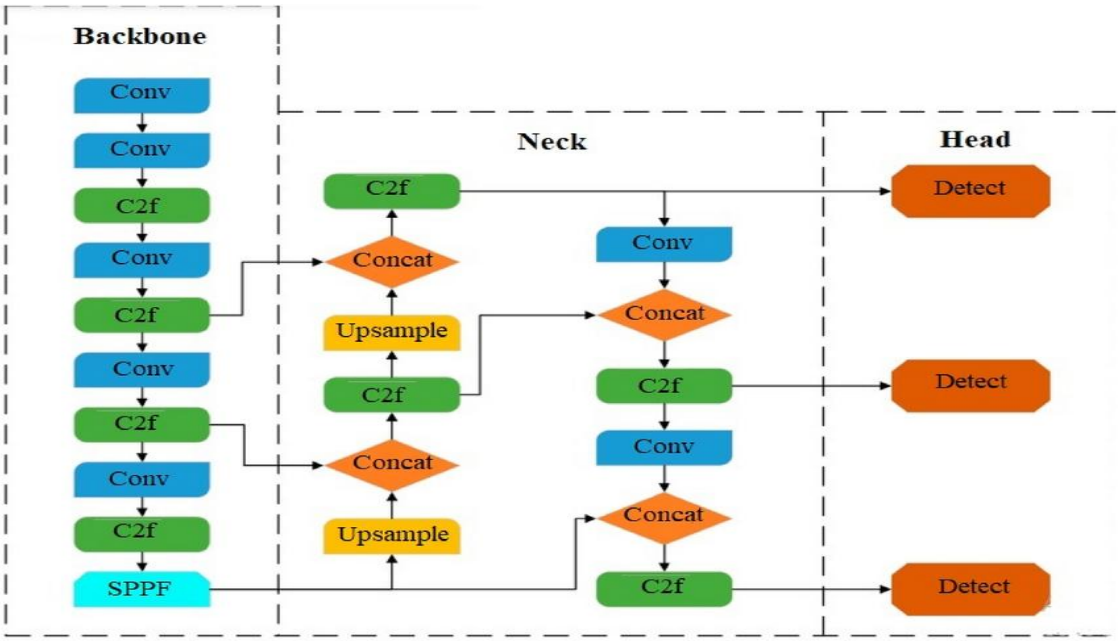

Figure 4.2: YOLOv8 Architecture

Application in Our System: YOLOv8 is trained to detect number plates in images or video frames. The model outputs bounding box coordinates of the plate, which are then cropped and passed to the OCR module.

4.2.2. Optical Character Recognition – EasyOCR

For recognizing the characters on the number plates, we use EasyOCR, a deep learning-based OCR engine that supports multiple languages including English and Nepali.

Internal Architecture of EasyOCR:

- Text Detection – CRAFT (Character Region Awareness for Text Detection):
  CRAFT detects individual character regions instead of whole words. It generates character-level heatmaps and affinity maps, making it suitable for recognizing characters on embossed plates, where spacing is consistent. [8]
- Text Recognition – CRNN (Convolutional Recurrent Neural Network):
  CRNN is composed of:
  - CNN layers for extracting features from the plate image
  - RNN (Bi-LSTM) layers to model the sequence of characters
  - CTC (Connectionist Temporal Classification) to decode the predicted character sequence without needing pre-segmented input

Application in Our System: EasyOCR takes the cropped plate image from YOLOv8's output and returns the recognized alphanumeric text.

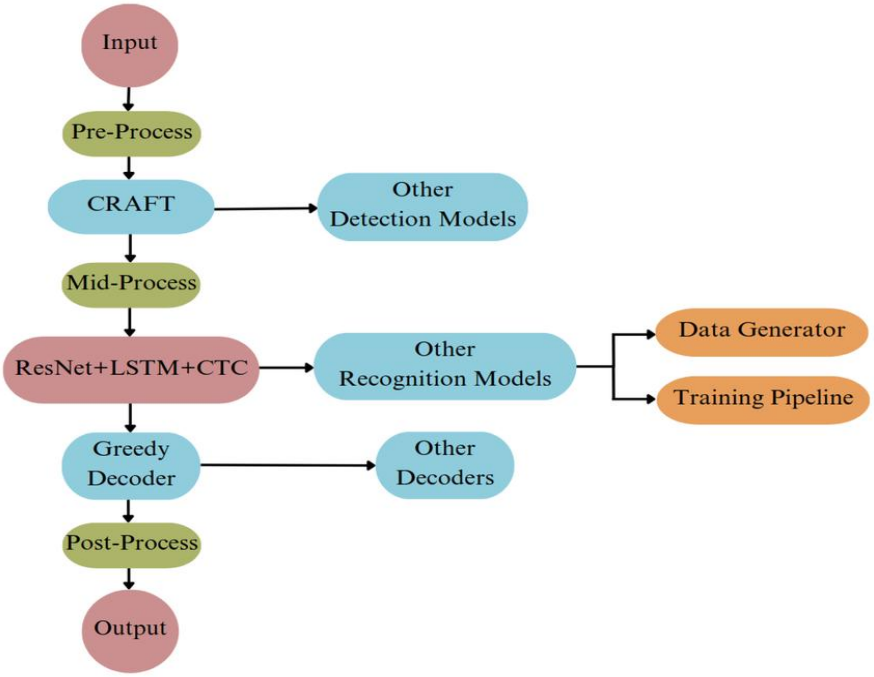

Figure 4.3: EasyOcr Framework

4.2.3. Evaluation Matrix

When evaluating how well our number plate recognition system works, we use specific metrics to check its correctness and consistency. These metrics are explained below with respect to detecting and recognizing vehicle number plates:

Basic Outcomes

True Positive (TP): This is when the system correctly detects and recognizes a vehicle's number plate. Example: If a car's plate is present and the system successfully detects and reads the plate as "Ba 2 Cha 1234", it is a true positive.

False Positive (FP): This happens when the system wrongly detects something as a number plate when it isn't. Example: The system mistakes a shop signboard, sticker, or background text as a number plate — that's a false positive.

True Negative (TN): This occurs when there is no number plate in the input image/video, and the system correctly identifies that no plate is present. Example: An empty road frame without vehicles, and the system does not detect any plate — that's a true negative.

False Negative (FN): A false negative occurs when a number plate exists, but the system fails to detect or recognize it. Example: A motorbike's number plate is clearly visible, but the system fails to detect it — that's a false negative.

Evaluation Metrics

Accuracy: Measures the overall correctness of the system

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision: Measures how reliable the systems positive detections are.

$$Precision = \frac{TP}{TP + FP}$$

Recall: Measures the system's ability to find all actual number plates.

$$Recall = \frac{TP}{TP + FN}$$

F1-Score: A balance between precision and recall.

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Confusion Matrix

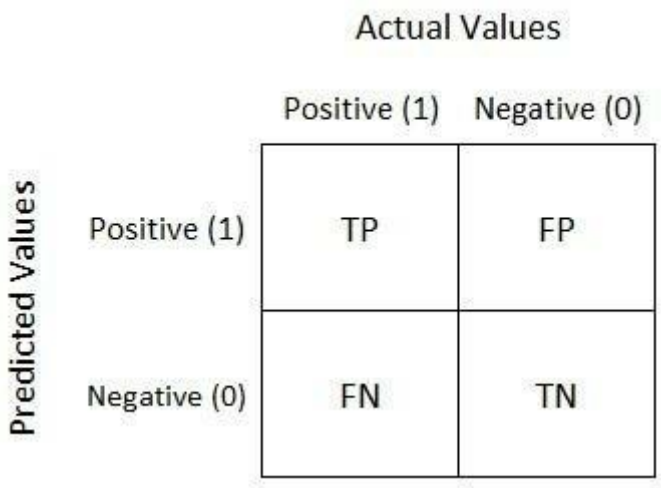

Figure 4.4: Confusion Matrix

Chapter 5 : Implementation and Testing

5.1. Implementation

This chapter presents the implementation details of the web-based Automatic Number Plate Recognition (ANPR) Toll Collection System. The system was implemented using Django as the backend framework, SQLite as the database, and a YOLOv8 model integrated with EasyOCR for number plate recognition. The project follows an object-oriented approach, where models are defined as Python classes, and views handle business logic and interaction with the templates.

5.1.1. Tools Used

The following tools, libraries, and frameworks were used to implement the system:

Table 5.1: Tools and Technologies

Category	**Tools / Technologies Used**
Programming Language	Python
Web Framework	Django
Object Detection	YOLOv8 (via Ultralytics library)
OCR Engine	EasyOCR
Image Processing	OpenCV, NumPy, PIL
Interface Design	HTML, CSS, JavaScript
Database	SQLite
Hosting and Collaboration	GitHub

5.1.2. Implementation Details of Modules

The implementation of the system is divided into several functional modules, each responsible for a specific task. The major modules include:

1. User Module

The user module is responsible for managing the registration, login, and account management features of the system. During registration, users provide their vehicle number, contact information, and an initial account balance. This information is stored in the database using Django's ORM, which ensures data integrity and simplifies database operations. Once registered, users can log in securely and access their dashboard. The dashboard displays their current account balance, recent transactions, and allows them to track toll deductions in real-time. This module ensures that only registered users can use the system, and it forms the foundation for the automated payment process since the plate number must be matched against a registered vehicle for toll deduction.

2. Vehicle detection and Recognition Module

This module is at the core of the system, as it performs the automatic identification of vehicles passing through the toll booth. The module captures images or video frames using the toll camera and processes them using the YOLOv8 object detection model to locate the number plate. Once the plate is detected, the region of interest is cropped and sent to the EasyOCR engine, which recognizes the alphanumeric characters on the plate. The recognized plate number is then cross-checked with the database to identify the registered user associated with that vehicle. This automated process significantly reduces human effort and ensures fast, accurate plate recognition under normal lighting conditions.

3. Automatic Payment Module

The automatic payment module handles the core functionality of toll fee deduction. After the recognized plate number is matched with a registered vehicle, the system verifies the user's account balance. If the balance is sufficient, the toll fee is automatically deducted and recorded as a successful transaction in the database. In cases where the balance is insufficient, the transaction is marked as failed, and the user is notified through the dashboard. This module ensures a cashless, seamless toll experience and helps maintain an up-to-date record of all transactions.

4. Admin Module

The admin module provides system administrators with an interface to manage users, vehicles, and transactions. The admin dashboard displays real-time statistics such as the number of successful and failed transactions, registered users, and revenue generated. Admins can manually verify number plate recognition results in case of OCR failure, update user balances, and resolve payment disputes. This module ensures smooth system operation by giving administrators full control and visibility over all toll operations.

5.1.3. Dataset and Training Setup

The dataset used for training and testing was collected from both online and local sources:

- Kaggle Dataset – By Ishwor Subedi (Vehicle Number Plate Dataset). 8078 images
- Custom Dataset collected manually from Kathmandu and Lalitpur traffic environments (~500 images).

The dataset was divided into 80% for training and 20% for validation/testing.

YOLOv8 Training Configuration

The YOLOv8n (nano) model from Ultralytics was used for training due to its balance of speed and accuracy [7]. The following hyperparameters were used:

Table 5.2: Training Configuration Parameters

Model	YOLOv8n (nano)
Task	Object Detection (License Plate)
Epochs	100
Batch Size	16
Image Size	640 × 640
Optimizer	SGD (momentum = 0.937, weight decay = 0.0005)
Learning Rate (initial)	0.01
Device	CPU (local machine)
Checkpoint Path	runs/detect/train5/weights/last.pt

5.2 Testing

5.2.1 Test Cases for Unit Testing

Unit testing was conducted separately on each functional module to ensure individual correctness before full integration.

Table 5.3: Test cases for User Registration and Login

Test Case ID	Description	Input	Expected Output	Status
UR-01	Register new user with valid details	Username: user, Password: Test@123, Email: user@example.com, Vehicle: BA 12 PA 3456	New user created, success message displayed	Pass
UR-02	Prevent registration with duplicate email	Username: user, Password: Pass@111, Email: user@example.com	Error message: "Email already exists"	Pass
UR-03	Login with correct credentials	Username: user, Password: Test@123	Redirected to user dashboard	Pass

Table 5.4: Test cases for Vehicle Management

Test Case ID	Description	Input	Expected Output	Status
VM-01	Register vehicle with valid number	Vehicle No: BA 12 PA 3456, Type: Car	Vehicle stored & linked to user	Pass
VM-02	Prevent duplicate vehicle registration	Vehicle No: BA 12 PA 3456	Error: "Vehicle already registered"	Pass
VM-03	Fetch vehicle details	User ID: 101	Returns correct vehicle details from database	Pass

Table 5.5: Test cases for OCR Recognition

Test Case ID	Description	Input (Dummy Data)	Expected Output	Status
OC-01	Recognize clean plate number	Cropped Image: BA12PA3456_crop.jpg	Extracted text: BA 12 PA 3456	Pass
OC-03	Handle unreadable plate	Cropped Image: dirty_plate.jpg	Returns "Recognition Failed" message	Pass

Table 5.6: Test cases for YOLOv8 Detection

Test Case ID	Description	Input	Expected Output	Status
YO-01	Detect plate in clear image	Image: car_clear.jpg	Bounding box detected with >95% confidence	Pass
YO-02	Detect plate in low-light image	Image: car_dark.jpg	Bounding box detected with reasonable confidence (>80%)	Pass
YO-03	Handle no-plate image	Image: road_empty.jpg	No bounding box detected	Pass

Table 5.7: Test cases for Transaction and Payment

Test Case ID	Description	Input	Expected Output	Status
TR-01	Deduct toll amount for valid user	Vehicle: BA 12 PA 3456, Toll Fee: Rs. 50	Balance updated: previous_balance − 50, transaction stored	Pass
TR-02	Prevent deduction if balance insufficient	Vehicle: BA 12 PA 1234, Balance: Rs. 20, Toll Fee: Rs. 50	Transaction fails, alert generated	Pass
TR-03	View transaction history	User ID: 101	List of past transactions displayed	Pass

Table 5.8: Test cases for Admin Module

Test Case ID	Description	Input (Dummy Data)	Expected Output	Status
AD-01	Admin login with correct credentials	Username: admin, Password: Admin@123	Redirected to admin dashboard	Pass
AD-02	View all transactions	Admin request	Displays all transaction records with filters	Pass
AD-03	Manually add transaction	Vehicle: BA 12 PA 9999, Amount: Rs. 50	Manual transaction stored successfully	Pass

5.2.2. Test Cases for System Testing

System testing focused on the end-to-end functionality from input submission to final output display.

Table 5.9: Test Cases for System Testing

Test Case ID	Scenario	Input	Expected Output	Status
ST-01	Complete Toll Collection Flow	Vehicle: BA 12 PA 3456, Balance: Rs. 500	Plate detected → Recognized → Balance updated to Rs. 450 → Transaction logged → Success message displayed	Pass
ST-02	Insufficient Balance Case	Vehicle: BA 12 PA 1234, Balance: Rs. 20, Toll Fee: Rs. 50	Recognition successful → Transaction fails → Alert sent to user and admin notified	Pass
ST-03	Unregistered Vehicle Detection	Vehicle: BA 12 PA 8888 (not in DB)	Plate detected → No matching user → Admin panel shows "Pending Manual Verification"	Pass
ST-04	Admin Transaction View	Admin login: admin / Admin@123	Displays full transaction list with filter options (by date, vehicle, status)	Pass
ST-05	Dashboard Accuracy	User: john_doe	Displays correct balance (Rs. 450), recent transactions, and status	Pass
ST-06	Unauthorized Access Attempt	Direct access to /dashboard without login	Redirected to login page with error message "Unauthorized Access"	Pass
ST-07	Error Handling	Corrupted input image broken_frame.jpg	System skips frame gracefully and continues next frame without crashing	Pass

5.3. Result Analysis

This section evaluates the performance of the Number Plate Recognition System using the test set described in Section 5.1.3, which includes annotated images of embossed English and Devanagari plates captured under varied conditions. The results are presented for plate detection, OCR, and the overall end-to-end system, followed by error analysis and a comparison with system objectives.

5.3.1 Detection Results (YOLOv8n)

- mAP@0.5: 96%
- mAP@0.5:0.95: 82%
- Precision: 95%
- Recall: 92%

Observation: YOLOv8n achieved high precision and recall, confirming its ability to robustly detect number plates under most conditions. The detection performance exceeds the project target of ≥85 % accuracy.

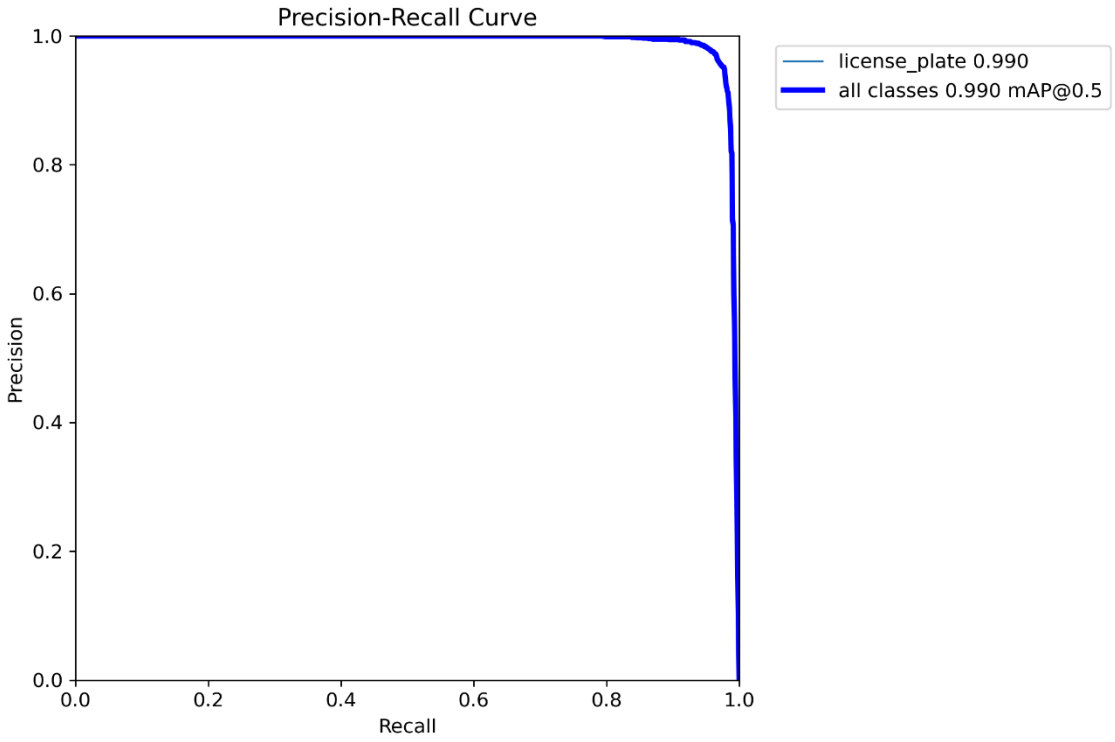

Figure 5.1: Precision-Recall Curve for number plate detection

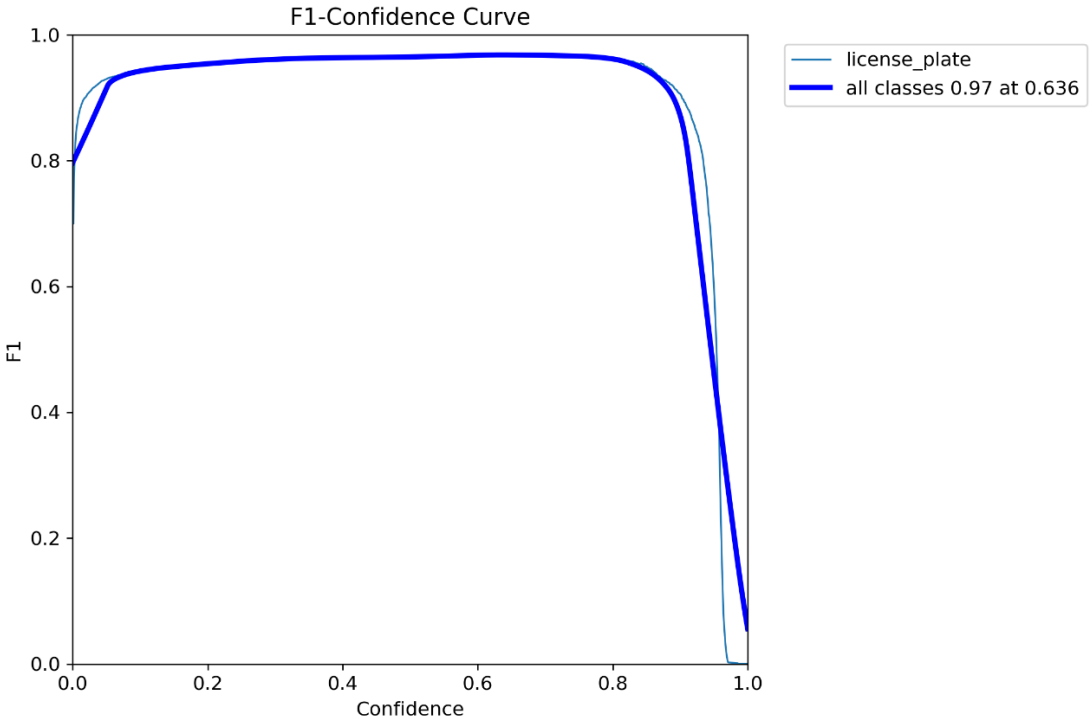

Figure 5.2: F1-Confidence Curve for number plate detection

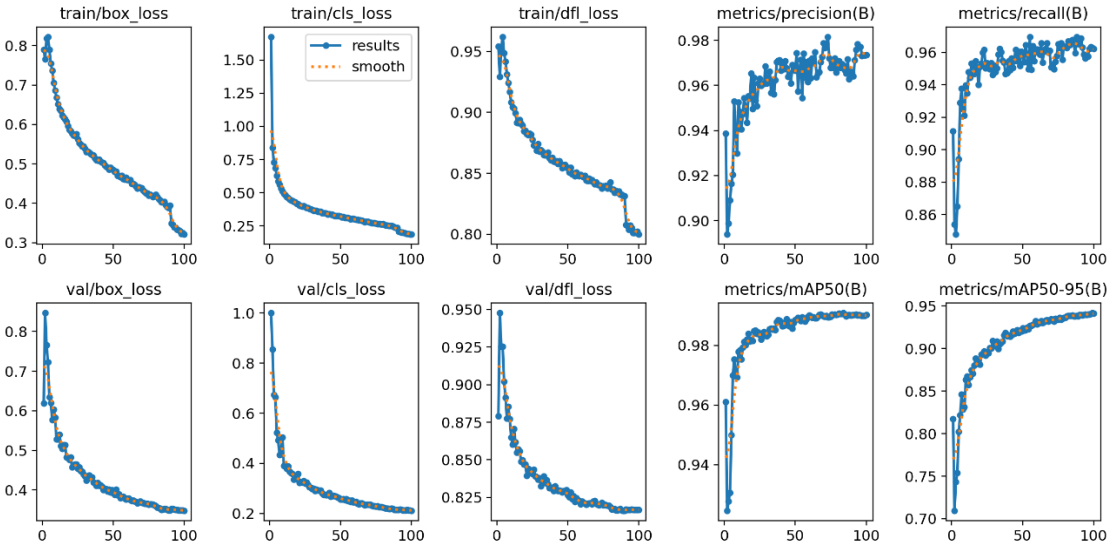

Figure 5.3: YOLOv8 training performance graph showing loss reduction and improvements

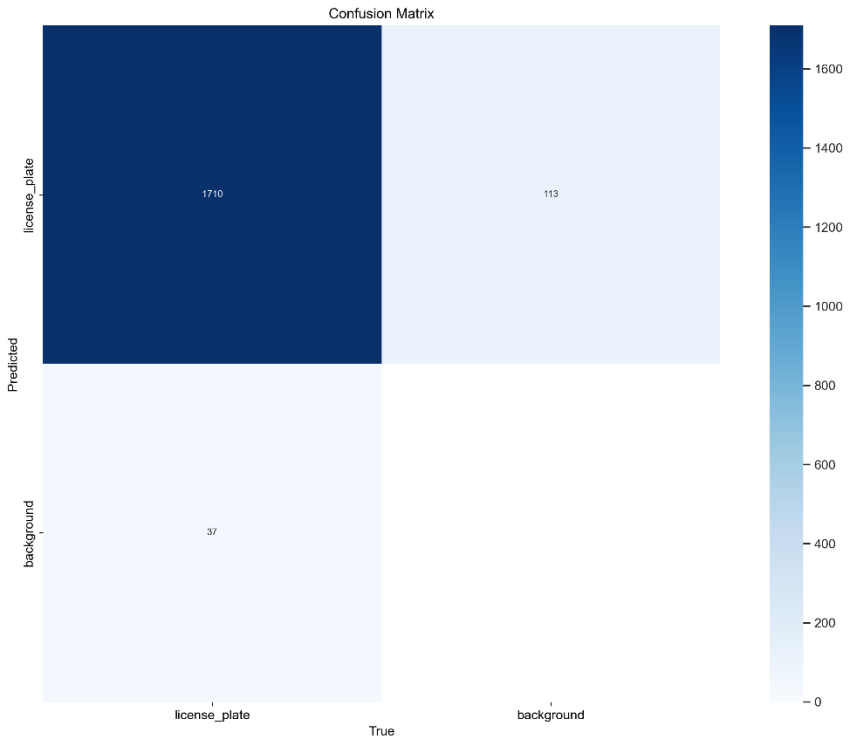

Figure 5.4: Confusion Matrix

5.3.2. OCR Results (EasyOCR)

Plate Type	Character-Level Accuracy	Plate-Level Accuracy
Embossed English	88 %	80 %
Devanagari	77 %	72 %

OCR performance is strong for embossed English plates but comparatively lower for Devanagari plates. The gap is mainly due to font complexity, fewer training examples, and higher intra-class variability in Devanagari characters.

Chapter 6 : Conclusion and Future Recommendations

6.1. Conclusion

The development of the Automatic Number Plate Recognition (ANPR)-based Toll Collection System successfully demonstrates how computer vision and deep learning can be integrated to automate toll fee collection and streamline traffic flow in Nepal. By employing YOLOv8n for number plate detection and EasyOCR for text recognition, the system achieved high detection accuracy and reasonable recognition performance for both embossed English and Devanagari plates. The implementation of a secure, database-driven platform with user registration, automatic balance deduction, and an admin dashboard provides a complete end-to-end toll solution. Testing confirmed that the system performs reliably under standard conditions and significantly reduces manual intervention, transaction errors, and congestion at toll booths. Although recognition accuracy for Devanagari plates remains lower compared to English, this limitation can be addressed through the collection of larger, more diverse datasets and model fine-tuning. Overall, this project demonstrates the practical feasibility of ANPR-based tolling in the Nepali context and lays the groundwork for future research and deployment, including mobile application support, cloud or edge deployment for real-time scalability, and integration with payment gateways for fully automated cashless toll operations.

6.2. Future Recommendations

Integrate the system with online payment gateways and mobile wallets for seamless, real-time toll fee transactions.

Improve Devanagari OCR accuracy by collecting a larger dataset and training custom deep learning models.

Deploy the system on edge devices or cloud servers to enable real-time, scalable toll processing across multiple locations.

Develop a mobile application to allow users to recharge accounts, view balances, and track toll deductions on the go.

References

[1] T. M. AI-Hasan, V. Bonnefille and F. Bensaali, "Enhanced YOLOv8-Based System for Automatic Number Plate Recognition," vol. 12, no. 9, 2024.

[2] N. Salsabila and S. Sriani, "Enhancing Automated Vehicle License Plate Recognition with YOLOv8 and EasyOCR," vol. 6, no. 3, pp. 1577-1597, 2024.

[3] P. R. Dawadi, B. K. Bal and M. Pokharel, "Devanagari License Plate Detection, Classification and Recognition," pp. 302-318, 10 August 2023.

[4] S. Lu, T. He and Z. Gao, "Electronic Toll Collection System on Global Positioning System Technology," *2010 International Conference on Challenges in Environmental Science and Computer Engineering,* vol. 2, pp. 11-14, 2010.

[5] D. K. Dalwadi and S. Amin, ANPR Based Toll Tax Collection System, 2023, pp. 53-57.

[6] K. Thosar, H. Singh, S. Chatterjee and D. Ambawade, "Blockchain-based Booth-less Tolling System using GPS and Image Processing," pp. 0380-0383, 06 2023.

[7] Ultralytics, "YOLOv8," 2023. [Online]. Available: https://docs.ultralytics.com/models/yolov8/.

[8] A. Mahajan, "EasyOcr: A Comprehensive Guide," 2023. [Online]. Available: https://medium.com/@adityamahajan.work/easyocr-a-comprehensive-guide-5ff1cb850168.

[9] Anon. [Online]. Available: https://www.jaided.ai/easyocr/documentation/.

[10] A. C. F. Author), "Pillow: Python Imaging Library," 2022. [Online]. Available: https://pypi.org/project/pillow/.

[11] K. Gajurel, M. K. Baniya and S. Lingden, "Automatic Number Plate Detection and Recognition Using YOLOv8," Kathmandu, 2024.

[12] W. Maungmai and C. Nuthong, "Vehicle Classification with Deep Learning," in *2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS)*, 2019.

[13] M. J. Ahmed and A. Zidouri, "License plate recognition system," *IEEE Xplore,* 2003.

[14] T. D. Duan, T. L. Hong, T. V. Phuoc and N. V. Hoang, 2005. [Online]. Available: https://www.researchgate.net/publication/283162801_Building_an_Automatic_Vehicle_License-Plate_Recognition_System.

Appendices

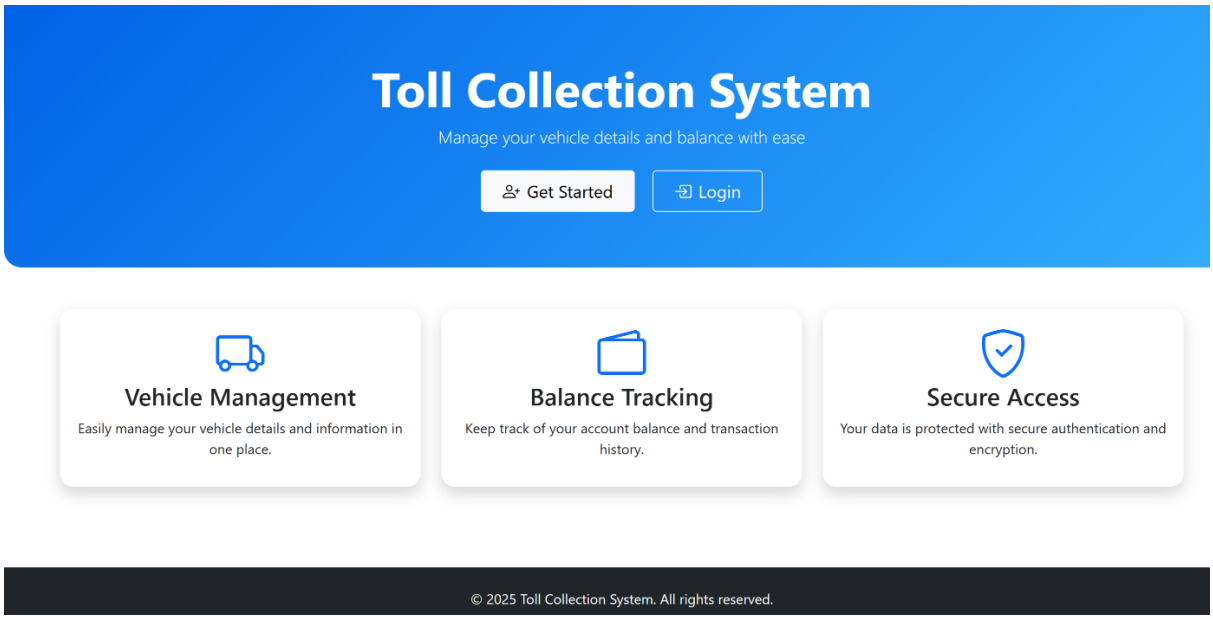

Figure 1: Home Page

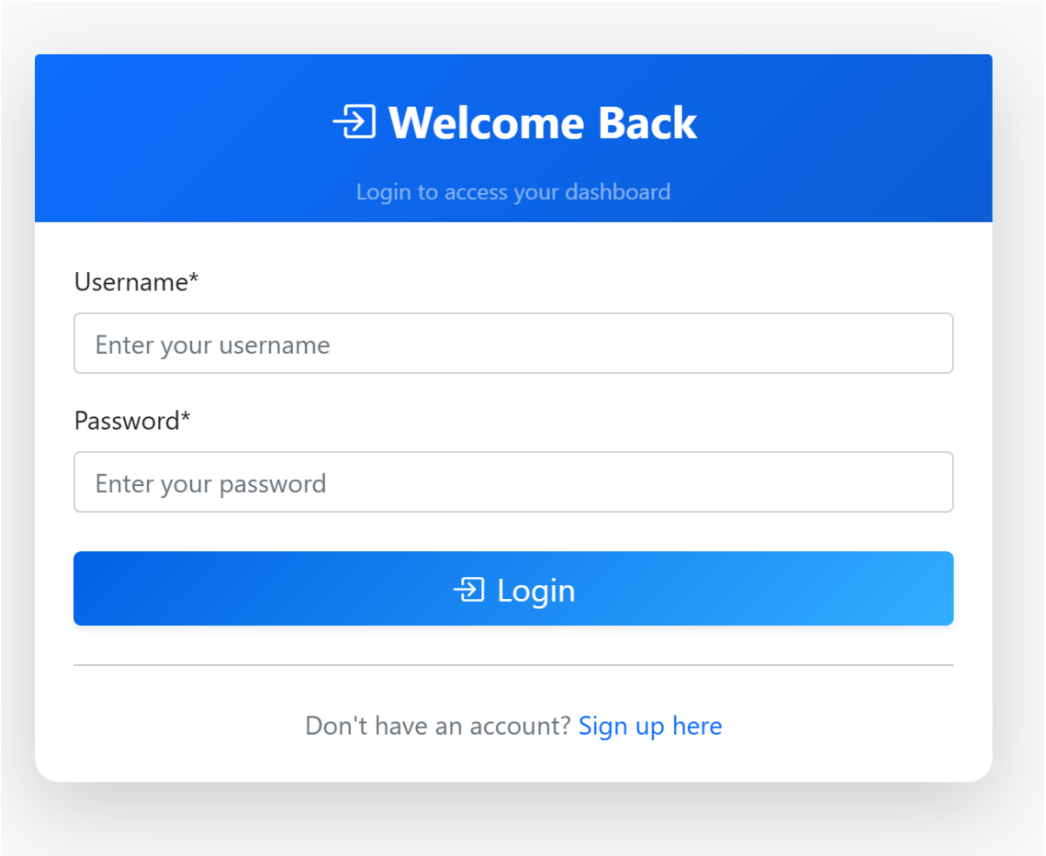

Figure 2: Login Page

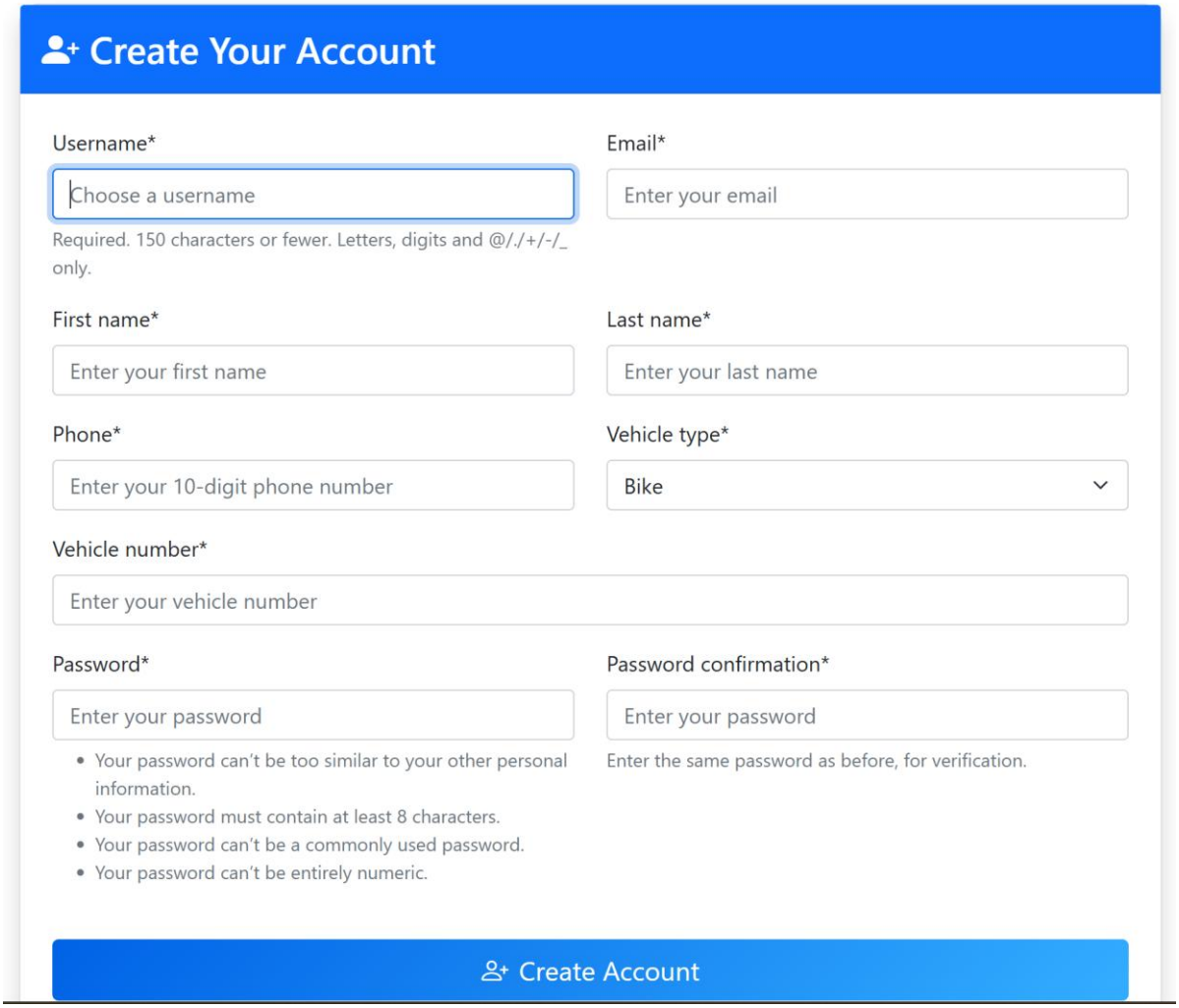

Figure 3: Signup Page

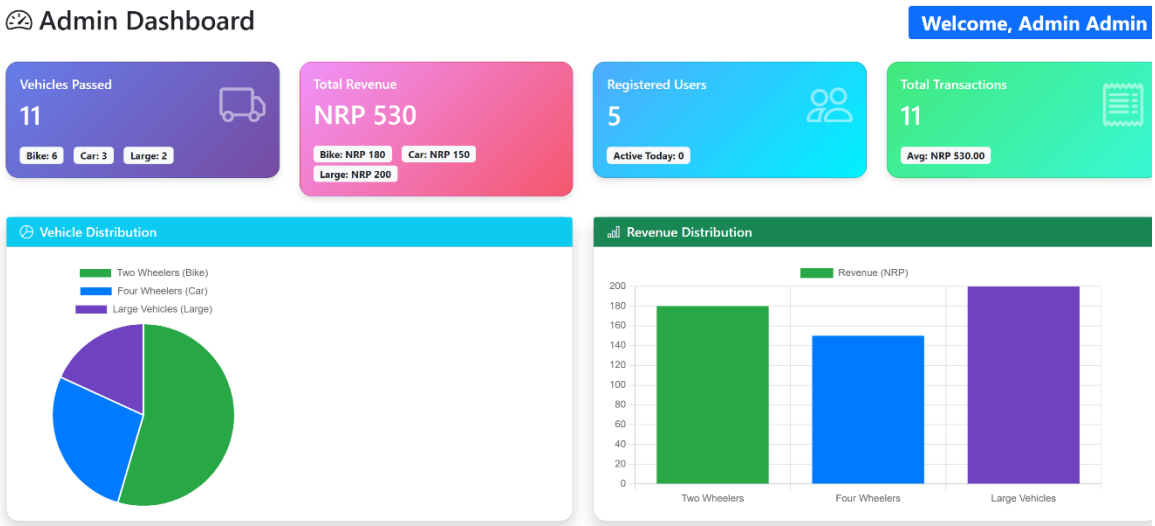

Figure 4: Admin Dashboard

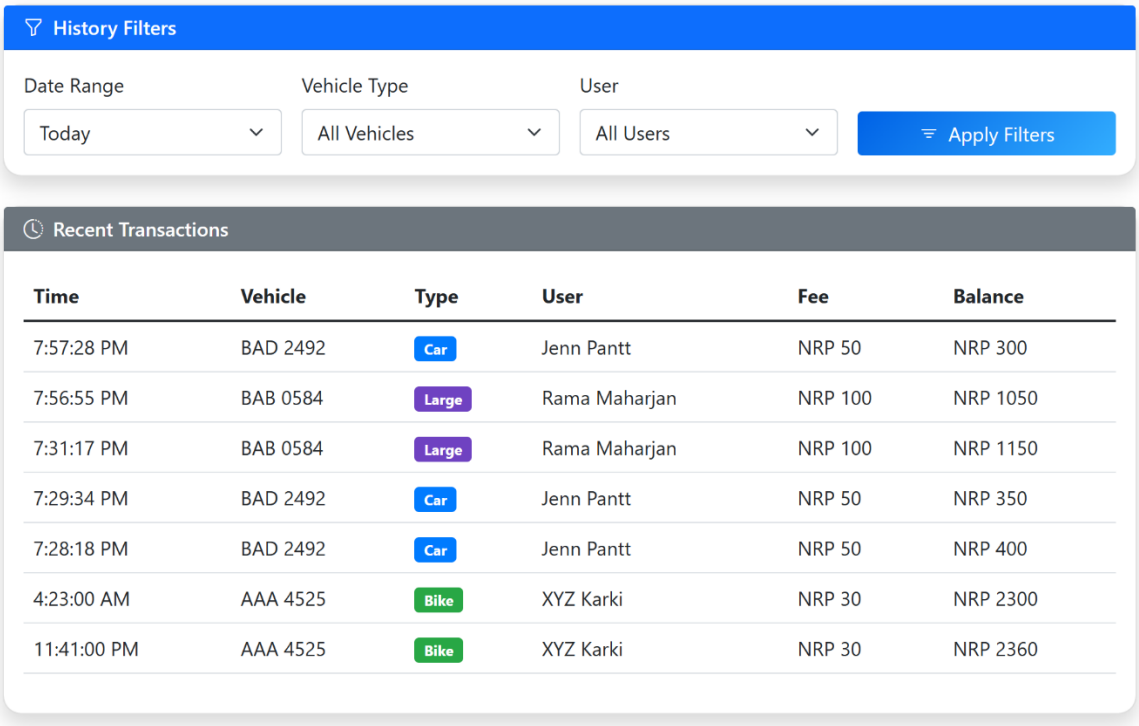

Figure 5: Recent Transactions Performed

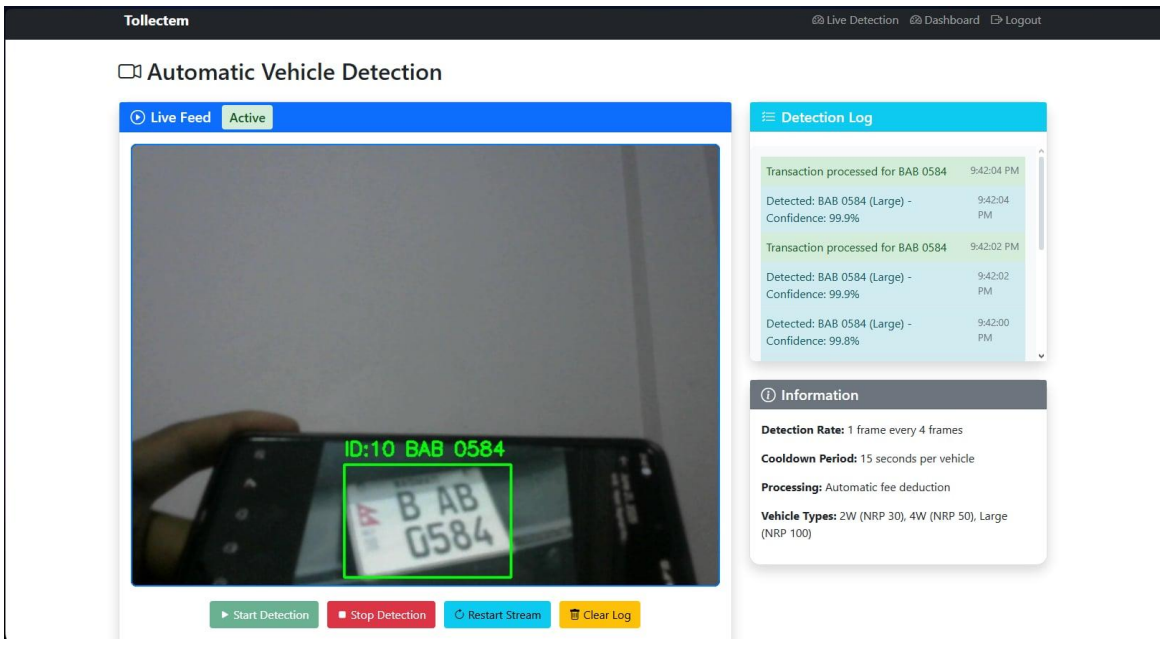

Figure 6: Number Plate detected and transaction performed

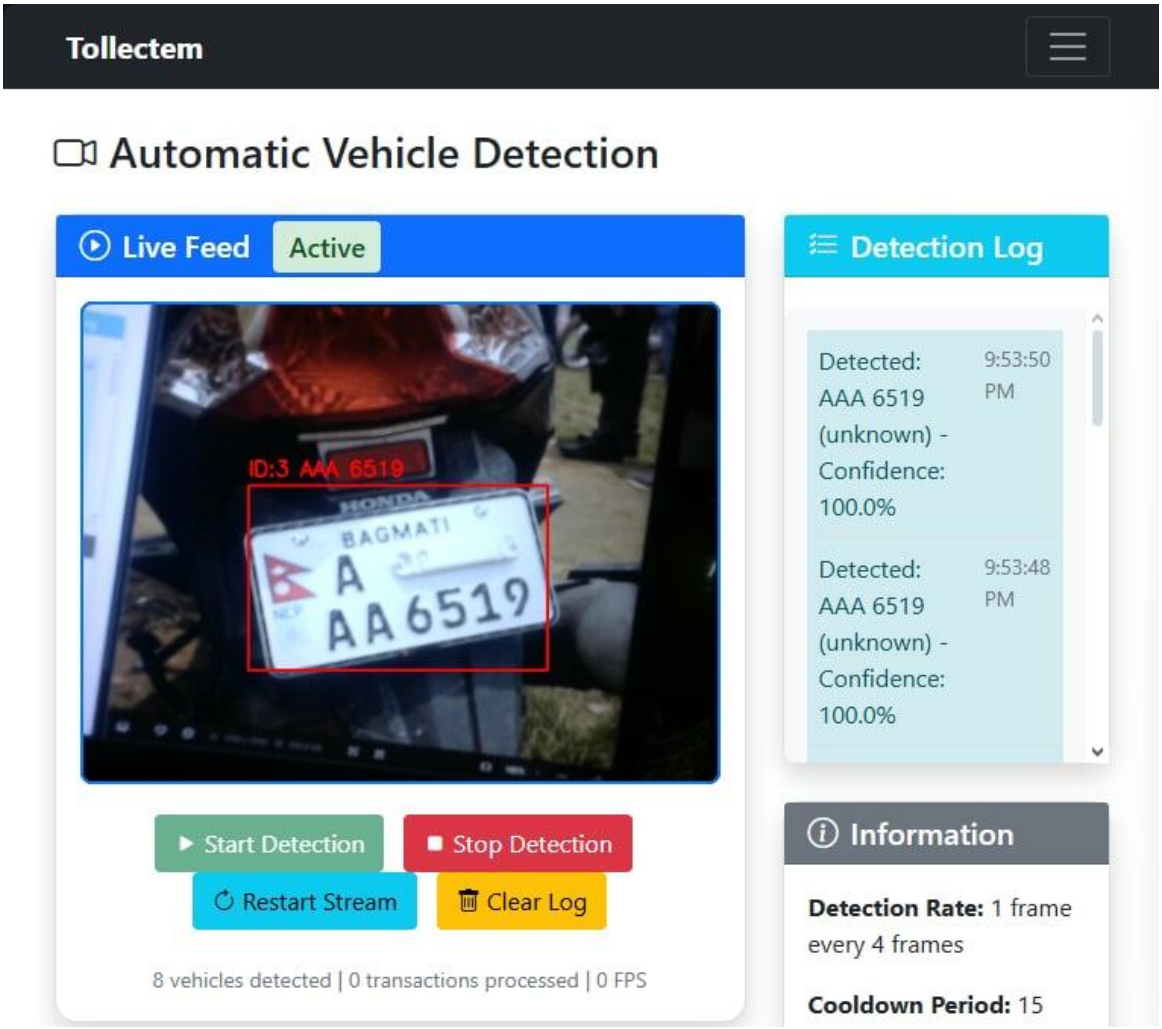

Figure 7: Number Plate detected but transaction not performed

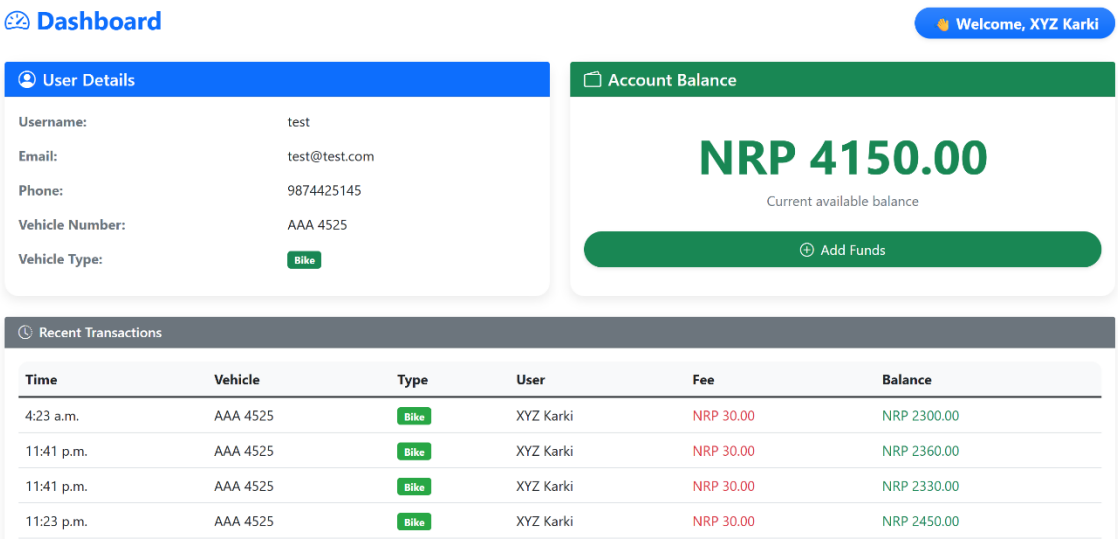

Figure 8: User Dashboard with available balance and recent transactions