Project Proposal

1 1. Project Title

VisionGuard – Automated License Plate Recognition (ALPR) System

2 2. Project Overview

The Automated License Plate Recognition (ALPR) System is a desktop application designed to accurately and efficiently identify and log license plate numbers from images or video feeds. The application will utilize computer vision techniques and optical character recognition (OCR) to extract and process license plate information.

Key Features and Functionalities:

- Real-time license plate detection from video feeds.
- Image-based license plate recognition.
- Logging and storing recognized license plate numbers.
- User-friendly interface for managing and viewing logs.

3 3. Rationale & Market Relevance

Why Choose This Project? The rise in the number of vehicles on the road necessitates efficient traffic management and security systems. An Automated License Plate Recognition system offers a solution to these challenges by providing real-time monitoring and logging of vehicles.

Real-World Problem Addressed: Traffic management authorities and security agencies require an automated solution to monitor vehicles, enforce traffic laws, and enhance security. Manual logging of vehicle information is inefficient and prone to errors. The ALPR system addresses these issues by automating the process.

Existing Solutions and Added Value: While there are existing ALPR systems, many are expensive and not easily accessible. This project aims to develop a cost-effective and user-friendly application that can be widely adopted by traffic management authorities, parking facilities, and private organizations.

4 4. Workflow & Methodology

Approach and Techniques: The ALPR system will leverage convolutional neural networks (CNNs) for license plate detection and Optical Character Recognition (OCR) for character extraction.

AI/ML Models, Tools, and Frameworks:

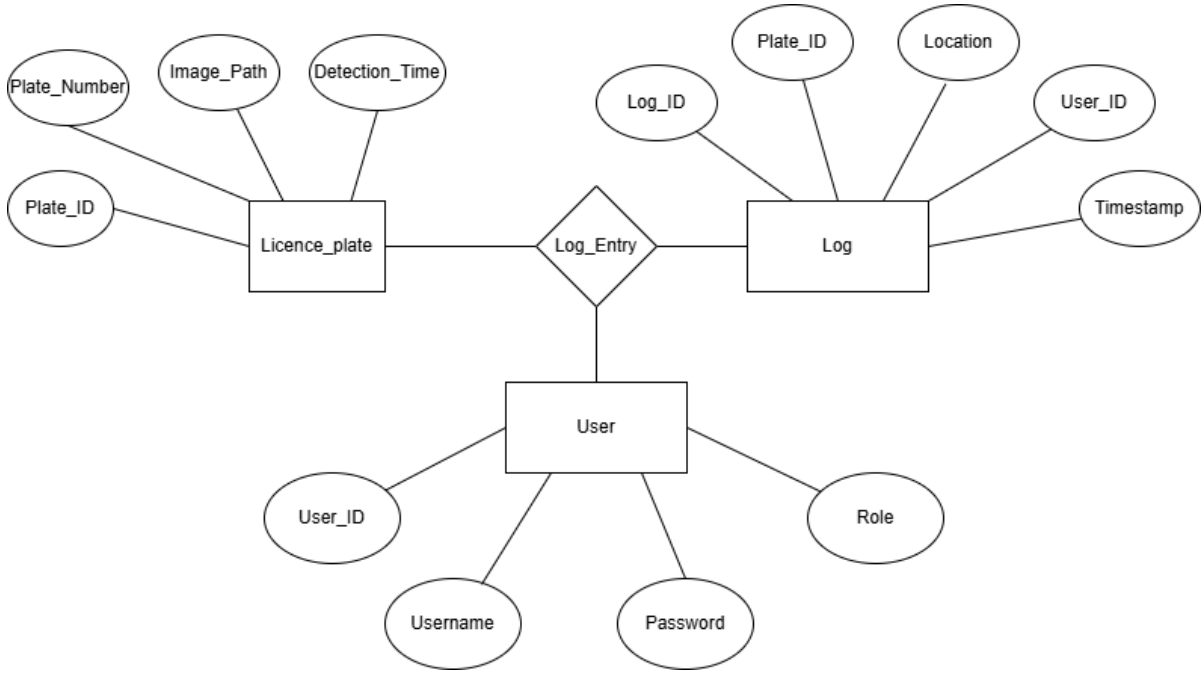
- **OpenCV**: For image and video processing.
- **Tesseract OCR**: For optical character recognition.
- **Keras/TensorFlow**: For building and training the CNN model.

Development Process:

1. **Data Collection and Preprocessing**:
    - Gather a dataset of vehicle images with visible license plates.
    - Preprocess images for training the CNN model.
2. **Model Training**:
    - Train the CNN model to detect and recognize license plates from images.
3. **Integration with OCR**:
    - Integrate Tesseract OCR to extract text from detected license plates.
4. **Development of Desktop Application**:
    - Build a user-friendly interface using PyQt or Tkinter.
    - Implement real-time video feed processing and logging features.
5. **Testing and Evaluation**:
    - Test the application on various datasets to evaluate accuracy and performance.
    - Refine the model and application based on feedback and test results.

5 5. ERD & Workflow

Entity-Relationship Diagram (ERD):



6 Workflow:

1. **User Interface:**
   - User launches the application.
   - User logs in to the system using a username and password.
2. **Image/Video Input:**
   - Application captures video feed or allows the user to upload images containing license plates.
3. **License Plate Detection:**
   - The system processes the input using a CNN model to detect license plates in the images or video frames.
4. **Optical Character Recognition (OCR):**
   - Detected license plates are processed using Tesseract OCR to extract the text from the plates.
5. **Data Logging:**
   - Extracted license plate numbers are logged into the database with relevant details (image path, detection time, location, user ID).
6. **Log Management:**
   - Users can view, search, and manage logged license plate entries through the application interface.