

**School of Computer Science**

**Masters in Applied Computing (M.A.C)**

**COMP8157**

**Advanced Database Topics**

**Professor Dr Shafaq khan**

**Project Proposal**

**by**

**Hamza Baig (110089314)**

**Arjun Kheni (student id here)**

**Meet Kevadiya (student id here)**

**Himani Rabari (student id here)**

Table of Contents

[Introduction: 3](#_Toc126074241)

[Problem statement: 3](#_Toc126074242)

[Motivation for solution: 3](#_Toc126074243)

[Methodology: 3](#_Toc126074244)

[Technology: 3](#_Toc126074245)

[References: 3](#_Toc126074246)

# Introduction:

The smart traffic system aims to improve road safety and reduce the number of road accidents by detecting traffic violations using computer vision. The system is designed to detect violations such as jumping a red light and record the plate number of the vehicles involved. The recorded information will be used to send an e-ticket to the violator.

# Problem statement:

Traffic violations such as jumping red lights, over speeding, and reckless driving are common and pose a significant threat to road safety. Conventional enforcement methods are time-consuming and not always effective. Hence, there is a need for a smarter and more efficient way to enforce traffic laws.

# Motivation for solution:

The use of computer vision technology in the smart traffic system offers several advantages over traditional enforcement methods. The system is quick and efficient in detecting violations and recording the information of the violator. It reduces the need for manual enforcement and minimizes human error. Moreover, the e-ticketing system makes it convenient for violators to pay their fines without having to visit a physical location.

# Methodology:

The smart traffic system will use computer vision algorithms and deep learning techniques to detect traffic violations. The cameras placed at intersections will capture images and feed them into the system for analysis. The system will use image processing techniques to detect the presence of a vehicle at the intersection and determine if it has jumped the red light. The license plate of the vehicle will be extracted and recorded. The recorded information will be used to generate an e-ticket and send it to the violator.

## Methodology for the project:

### SCRUM:

1. Form a cross-functional team: Scrum requires a cross-functional team with all the necessary skills to complete the project. This includes developers, testers, project managers, and stakeholders.
2. Define product backlog: A product backlog is a prioritized list of features and requirements that your team will work on. Create a product backlog for your smart traffic system project that includes all the necessary features and requirements.
3. Plan sprints: Sprints are short, time-boxed iterations of work that last 1-4 weeks. Plan sprints for your project, setting goals for each sprint and prioritizing the items in your product backlog.
4. Hold daily stand-up meetings: Scrum requires daily stand-up meetings to ensure that the team stays on track and that everyone is aware of what others are working on.
5. Review and retro: At the end of each sprint, hold a sprint review meeting to demonstrate what was completed during the sprint. Also, hold a sprint retro meeting to discuss what went well and what could be improved in the next sprint.
6. Repeat the process: Repeat the process of planning, developing, and reviewing for each sprint until all the items in the product backlog have been completed.

## Milestones for the project:

### Milestone 1 (Project Proposal - January 31st):

1. Research: Gather information about existing smart traffic systems, computer vision technologies and their applications in traffic management.
2. Define project scope: Identify the specific problem you're trying to solve and determine the requirements for the system you're proposing.
3. Outline solution: Describe the solution you're proposing, including the computer vision algorithms you plan to use and the key components of the system.
4. Create a project plan: Determine the timeline for the project, allocate resources and establish project milestones.
5. Write the project proposal: Present your research, project scope, solution, and project plan in a clear and concise proposal document.

### Milestone 2 (February 28th):

1. Develop the computer vision algorithms: Choose appropriate computer vision algorithms to detect traffic violations.
2. Implement the system components: Start implementing the system components, including the camera, computer vision algorithms, and data storage and retrieval system.
3. Test the system: Test the system components to ensure they work as expected.
4. Evaluate the performance: Evaluate the performance of the computer vision algorithms and make any necessary improvements.

### Milestone 3 (March 28th):

1. Integrate the system components: Integrate the camera, computer vision algorithms, and data storage and retrieval system into a single system.
2. Test the integrated system: Test the integrated system to ensure it functions as intended.
3. Deploy the system: Deploy the system in a real-world environment and monitor its performance.
4. Evaluate and refine the system: Evaluate the performance of the system and refine it based on the results.
5. Prepare documentation: Prepare comprehensive documentation on the system, including its design, implementation, and performance.

# Technology:

The smart traffic system will use computer vision and deep learning technologies, such as OpenCV and TensorFlow, to detect traffic violations. The system will also use image processing techniques, such as license plate recognition, to extract the license plate number of the violator's vehicle. The e-ticketing system will be developed using web technologies, such as HTML, CSS, and JavaScript.

* Computer Vision: OpenCV, TensorFlow, or PyTorch
* License Plate Recognition: ALPR (Automatic License Plate Recognition)
* Object Detection: YOLO, Faster R-CNN
* E-ticketing System: Node.js, Laravel, or Django.

# References:

1. OpenCV Library, "OpenCV - Open Source Computer Vision Library," [Online]. Available: https://opencv.org/. [Accessed: 31-Jan-2023].
2. TensorFlow, "TensorFlow - An Open Source Machine Learning Framework," [Online]. Available: https://www.tensorflow.org/. [Accessed: 31-Jan-2023].
3. PyTorch, "PyTorch - An Open Source Machine Learning Library," [Online]. Available: https://pytorch.org/. [Accessed: 31-Jan-2023].
4. Node.js, "Node.js - A JavaScript runtime built on Chrome's V8 JavaScript engine," [Online]. Available: https://nodejs.org/. [Accessed: 31-Jan-2023].
5. Laravel, "Laravel - The PHP Framework For Web Artisans," [Online]. Available: https://laravel.com/. [Accessed: 31-Jan-2023].
6. Django, "Django - The High-Level Python Web Framework," [Online]. Available: https://www.djangoproject.com/. [Accessed: 31-Jan-2023].