Vehicle Speed Checker

Submitted By

Student Name	Student ID
Abdullah Al Noman	221-15-5387
Nushrat Jahan Mila	221-15-5758
Abrar Hameem Bornil	221-15-5331
Mostafizur Rahman	221-15-5316
Khatmoon Nur Maharu	221-15-5421

MINI LAB PROJECT REPORT

This Report Presented in Partial Fulfillment of the course **CSE234 Embedded Systems and IoT Lab in the Computer Science and Engineering Department**

DAFFODIL INTERNATIONAL UNIVERSITY

Dhaka, Bangladesh

August 20, 2025

DECLARATION

We hereby declare that this lab project has been done by us under the supervision of **Kazi Hasibur Rahman**, **Lecturer**, Department of Computer Science and Engineering, Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere as lab projects.

Submitted To:

Kazi Hasibur Rahman
Lecturer
Department of Computer Science and Engineering
Daffodil International University

Submitted by

<table>
<tr><td colspan="2">Noman
____
Abdullah Al Noman
ID: 221-15-5387
Dept. of CSE, DIU</td></tr>
<tr><td>Nushrat
____
Nushrat Jahan Mila
ID: 221-15-5758
Dept. of CSE, DIU</td><td>Bornil
____
Abrar Hameem Bornil
ID: 221-15-5331
Dept. of CSE, DIU</td></tr>
<tr><td>Maharu
____
Khatmoon Nur Maharu
ID: 221-15-5421
Dept. of CSE, DIU</td><td>Mostafizur
____
Mostafizur Rahman
ID: 221-15-5316
Dept. of CSE, DIU</td></tr>
</table>

COURSE & PROGRAM OUTCOME

The following course have course outcomes as following:.

Table 1: Course Outcome Statements

CO's	Statements
CO1	**Define** and **Relate** classes, objects, members of the class, and relationships among them needed for solving specific problems
CO2	**Formulate** knowledge of object-oriented programming and Java in problem solving
CO3	**Analyze** Unified Modeling Language (UML) models to **Present** a specific problem
CO4	**Develop** solutions for real-world complex problems **applying** OOP concepts while evaluating their effectiveness based on industry standards.

Table 2: Mapping of CO, PO, Blooms, KP and CEP

CO	PO	Blooms	KP	CEP
CO1	PO1	C1, C2	KP3	EP1, EP3
CO2	PO2	C2	KP3	EP1, EP3
CO3	PO3	C4, A1	KP3	EP1, EP2
CO4	PO3	C3, C6, A3, P3	KP4	EP1, EP3

The mapping justification of this table is provided in section **4.3.1**, **4.3.2** and **4.3.3**.

Table of Contents

Table of Contents

Chapter 1

Introduction

1.1 Introduction

Road safety is one of the most important challenges in modern transportation. Every year, thousands of road accidents occur due to over speeding. Drivers often fail to realize how fast they are going, and sometimes authorities cannot enforce speed regulations effectively. This problem is especially critical in countries like Bangladesh, where highway monitoring systems are limited.

To address this issue, our team has developed a Vehicle Speed Checker Project. The aim of this project is to automatically calculate the speed of a vehicle, display it on an LCD screen, and give an immediate alert when over speeding occurs. This system provides an effective way to demonstrate how digital electronics and microcontrollers can be applied in real-world safety solutions.

Unlike conventional speedometers inside vehicles, this project functions as an external speed monitoring unit. It can be placed at checkpoints, roadsides, or entry points of restricted zones to check whether vehicles are crossing the speed limit.

A unique feature of our project is the addition of a Login System. Before the system starts, the operator must log in with a username and password. This ensures that only authorized users can operate the system. After login, the system activates, monitors vehicles, and displays the speed results. If the speed exceeds the predefined limit, the buzzer is activated and an ALERT message is displayed.

Thus, this project not only ensures security of access but also serves as a practical tool to improve road safety.

1.2 Motivation

Over speeding is a leading cause of fatal road accidents. According to the World Health Organization (WHO), a 5% increase in average speed results in approximately a 20% increase in fatal crashes. In Bangladesh, over speeding accounts for a significant number of highway accidents every year.

Most existing systems either rely on manual speed guns or expensive radar-based technologies. However, there is a gap in affordable, real-time, and locally implementable systems. This project was motivated by the idea of designing a low-cost, Arduino-based monitoring system that students, researchers, and authorities can use to promote safer driving.

1.3 Objectives

- To design a microcontroller-based system for measuring vehicle speed.
- To integrate a login authentication system for secure access.
- To calculate vehicle speed using two IR sensors placed at a fixed distance.
- To display results on a 16x2 I2C LCD screen.
- To trigger a buzzer alarm in case of over speeding.
- To create a system that is easy to replicate, cost-effective, and educational.

1.4 Feasibility Study

Technological Feasibility:

The system is based on the Arduino Uno microcontroller, which is an open-source, low-cost, and widely available platform. Components like IR sensors, LCD displays, and buzzers are inexpensive and easily obtainable.

Economic Feasibility:

The entire project can be built at a very low cost, making it suitable for academic labs and small-scale implementations. Unlike radar guns or advanced speed cameras, it requires no complex equipment.

Operational Feasibility:

The system is simple to use. Once the operator logs in, the system works automatically, requiring no additional configuration. The outputs are displayed clearly on the LCD, and buzzer alerts are straightforward.

1.5 Gap Analysis

Weaknesses of Existing Systems:

- Expensive radar/camera-based systems.
- Manual speed checks are time-consuming.
- Lack of alerts for immediate driver response.

How This Project Fills the Gap:

- Provides an affordable, portable, and automatic solution.
- Offers real-time alerts via buzzer and LCD messages.
- Adds security through login authentication.

1.6 Project Outcome

The expected outcomes of this project include:

- Accurate and real-time speed measurement.
- Clear display of vehicle speed on an LCD screen.
- Instant buzzer alerts in case of over speeding.
- Secure and restricted use through login authentication.
- Contribution to the field of Digital Electronics Lab applications.

Chapter 2

Proposed Methodology/Architecture

2.1 Requirement Analysis & Design Specification

Functional Requirements:

1. **Login Authentication:**

   Before the system starts, a username and password must be entered correctly. Without authentication, the system does not run.

2. **Speed Measurement:**

   The system uses two IR sensors placed at a fixed distance apart. When a vehicle passes the first sensor, a timer starts. When it passes the second sensor, the timer stops.

3. **Speed Calculation:**

   The Arduino calculates speed using the formula:

$$\text{Speed (km/h)} = \frac{\text{Distance (m)}\ X\ 3.6}{\text{Time } (s)}$$

4. **Display of Speed:**

   A 16x2 LCD (I2C protocol) is used to display messages such as login status, calculated speed, and alerts.

5. **Overspeed Alert:**

   If the calculated speed is above the threshold (e.g., 50 km/h), the **buzzer** produces an audible alarm and the LCD shows a warning message.

Non-Functional Requirements:

- **Usability:** Simple for operators — requires only login and then works automatically.
- **Portability:** Can be moved and used in different locations.
- **Low-Cost:** Built from affordable and widely available components.
- **Scalability:** Can be upgraded to log data, add GSM modules, or integrate with traffic monitoring systems.

Components Explanation

1. Arduino Uno (Microcontroller Board):
The Arduino Uno is the brain of this project. It is a microcontroller board based on the ATmega328P. It has:

- 14 digital I/O pins (6 can be used as PWM outputs)
- 6 analog input pins
- A 16 MHz quartz crystal
- USB connection and power jack

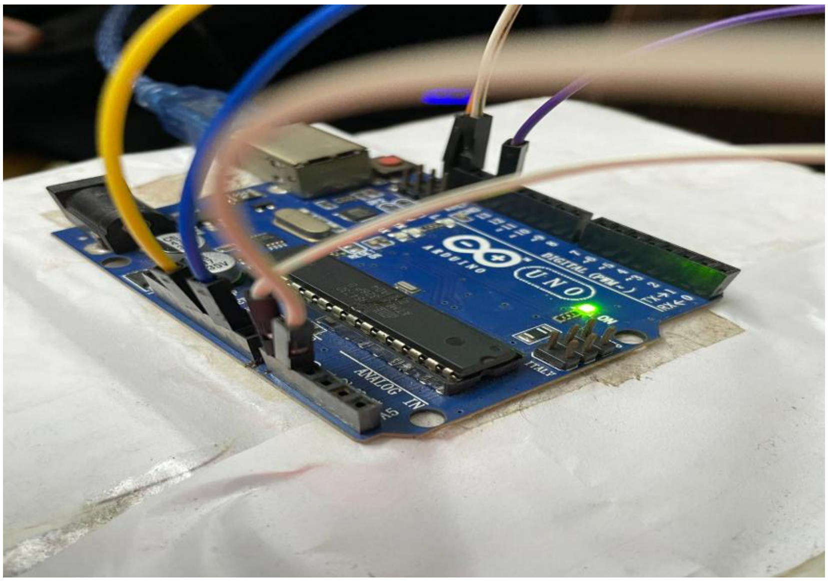

Figure-1: Arduino Uno (Microcontroller Board)

Why Arduino Uno?

- Easy to program (Arduino IDE).
- Low cost and beginner-friendly.
- Widely used in academic and hobby projects.
- Perfect for sensor-based projects like ours.

In this project, the Arduino Uno:

- Receives signals from IR sensors.
- Measures time difference.
- Calculates speed using the formula.
- Sends output to LCD.
- Activates the buzzer if over speeding is detected.

2. IR Sensors (Infrared Sensors):
Infrared sensors work by emitting infrared light and detecting its reflection when an object passes in

front of it.

In our project:

- Two IR sensors are placed at a fixed distance apart (e.g., 10 meters).
- First sensor: Starts the timer when a vehicle passes.
- Second sensor: Stops the timer when the vehicle passes.
- Arduino uses the time difference and known distance to calculate speed.

Advantages:

- Non-contact detection.
- Fast response time.
- Affordable and simple.

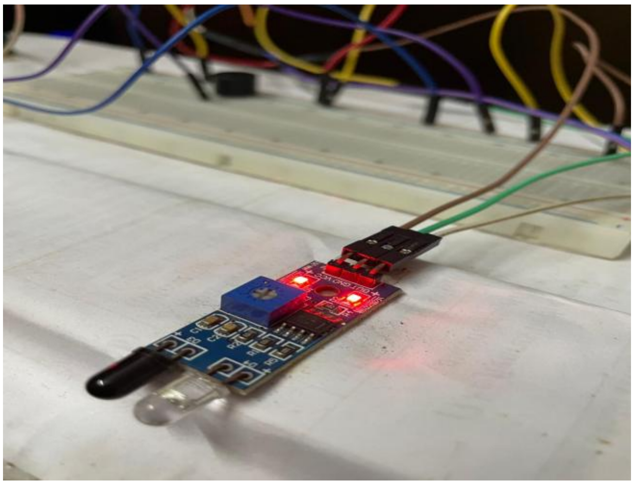

Figure-2: IR Sensors (Infrared Sensors)

3. 16x2 LCD Display with I2C Module:

- A 16x2 LCD can display 16 characters in each of its 2 rows.
- Normally, connecting this LCD requires many pins (RS, EN, D4–D7, etc.).
- With I2C module, only two wires are needed: SDA (A4) and SCL (A5).

In our project, the LCD is used to:

- Show login messages.
- Display calculated speed.
- Indicate whether speed is normal or over speeding.

Sample display:

Figure-3: 16x2 LCD Display with I2C Module

4. Buzzer (Audio Alert Device):

- The buzzer is connected to digital pin 13 of Arduino.
- When over speeding occurs, Arduino sends a HIGH signal to activate the buzzer.
- Provides real-time audible alert to drivers or operators.

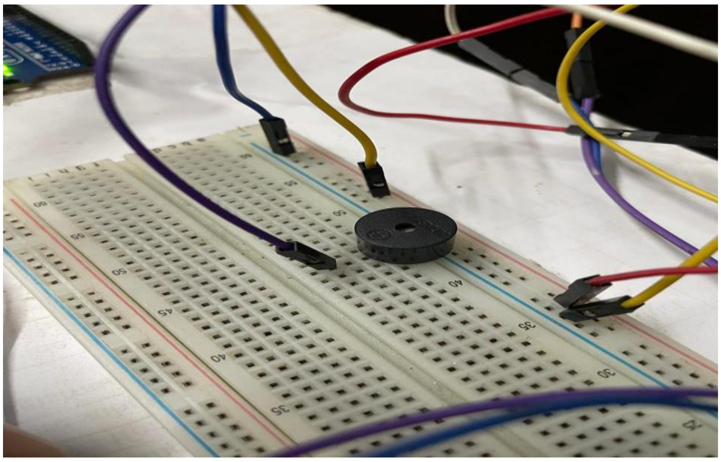

Figure-4: Buzzer (Audio Alert Device)

5. Breadboard (Prototyping Board):

- A solderless board used to connect components temporarily.
- Helps in quickly testing the circuit without permanent connections.
- Used to connect IR sensors, LCD, Arduino, and buzzer with jumper wires.

6. Jumper Wires (Male-to-Male, Male-to-Female, Female-to-Female):

- Provide flexible connections between Arduino and components.
- Carry power (5V, GND) and signal lines.

2.1.1 Overview

The overall system integrates sensors, microcontroller, and output devices. The login system ensures controlled access. Once logged in, the Arduino continuously waits for signals from the IR sensors. When a vehicle passes, speed is calculated and displayed. Alerts ensure immediate action for over speeding.

2.1.2 Proposed Methodology/ System Design

1. **Step 1: Login Authentication**
    - The system first displays a login screen.
    - User enters credentials.
    - If correct → Access Granted.
    - If incorrect → Access Denied (system does not start).
2. **Step 2: Sensor Detection**
    - Vehicle passes IR Sensor 1 → Timer starts.
    - Vehicle passes IR Sensor 2 → Timer stops.
3. **Step 3: Speed Calculation**
    - Formula applied:

$$\text{Speed (km/h)} = \frac{\text{Distance (m)}\ X\ 3.6}{\text{Time (s)}}$$

4. **Step 4: Display Output**
    - LCD shows speed.
    - Normal speed → “Vehicle Speed = 65.4 Km/Hr - Normal”.
    - Overspeed → “ALERT: Vehicle Speed = 45.5 Km/Hr - OVER SPEEDING!”.
5. **Step 5: Buzzer Alert**
    - If overspeed → buzzer sounds continuously until next measurement.

2.1.3 UI Design

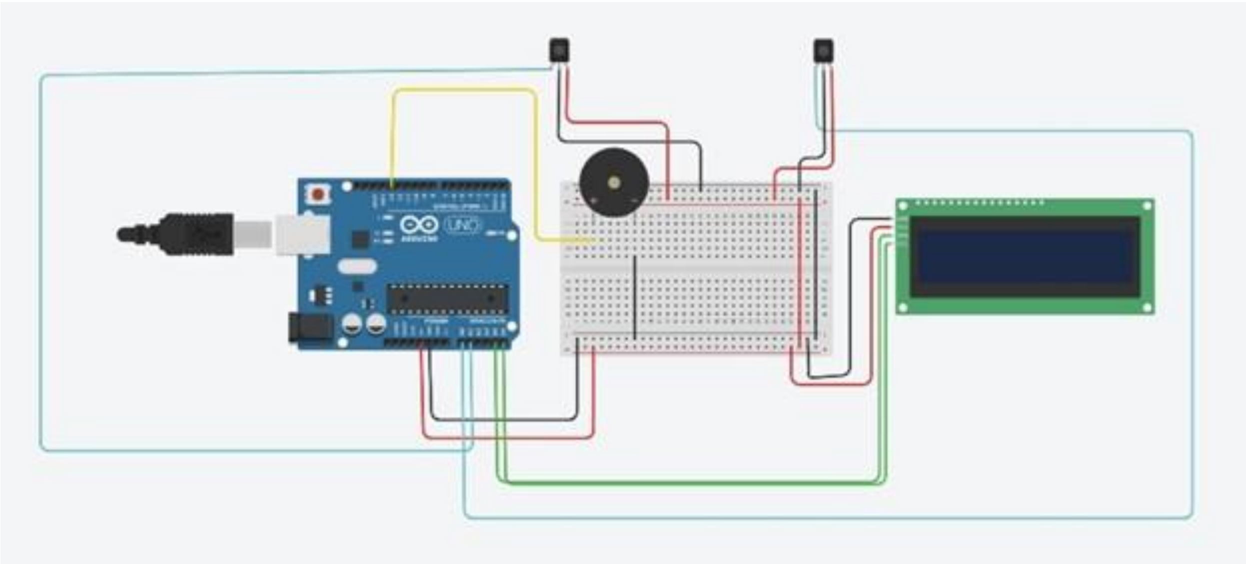

Figure 5: Implement the Circuit for Vehicle Speed Detector

Circuit Diagram Explanation

Although the diagram is not shown here, let’s describe it in detail:

1. **IR Sensors:**
   - First IR sensor’s OUT pin connected to A0 (analog pin) of Arduino.
   - Second IR sensor’s OUT pin connected to A1 (analog pin).
   - Both sensors powered with VCC = 5V and GND = Arduino GND.
2. **Buzzer:**
   - Positive pin connected to digital pin 13 of Arduino.
   - Negative pin connected to GND.
3. **LCD Display with I2C:**
   - SDA pin → A4 of Arduino.
   - SCL pin → A5 of Arduino.
   - VCC → 5V and GND → GND.
4. **Power Supply:**
   - Arduino powered through USB or external adapter.
   - Breadboard power rails distribute 5V and GND to all components.

Working Flow in Circuit:

- Vehicle passes IR Sensor 1 → signal to Arduino → timer starts.
- Vehicle passes IR Sensor 2 → signal to Arduino → timer stops.
- Arduino calculates Time Difference → Speed.
- Speed displayed on LCD.

- If speed > threshold → buzzer ON + ALERT message on LCD.

2.2 Overall Project Plan

1. **Requirement Collection:** Identify all necessary components.
2. **Circuit Design:** Build circuit on breadboard.
3. **Login Programming:** Implement authentication in Arduino code.
4. **Sensor Programming:** Code for time difference and speed calculation.
5. **LCD + Buzzer Integration:** Display and alert coding.
6. **Testing Phase:** Test for normal speed, overspeed, and edge cases.
7. **Final Presentation:** Document and analyze results.

Chapter 3

Implementation and Results

3.1 Implementation

The implementation of the Vehicle Speed Checker Project was done in multiple phases to ensure proper functioning of each component. Below is the step-by-step breakdown:

Step 1: Login System Implementation

- Before starting the system, the Arduino requires a username and password.
- If the correct credentials are entered, the system prints "Access Granted! System Starting…" on the LCD.
- If incorrect credentials are entered, the system prints "Access Denied! Please Try Again." and does not proceed further.

This step ensures authorized usage and prevents misuse of the system.

Step 2: Sensor Setup and Signal Reading

- Two IR sensors are placed at a fixed distance apart (e.g., 10 meters).
- When a vehicle passes the first sensor:
  - Arduino detects a LOW-to-HIGH signal change.
  - Timer is started.
- When the vehicle passes the second sensor:
  - Arduino detects the signal again.
  - Timer is stopped.
- The difference in time is recorded in milliseconds.

Step 3: Speed Calculation in Arduino

- Formula used:

$$\text{Speed (km/h)} = \frac{\text{Distance (m)}\ X\ 3.6}{\text{Time (s)}}$$

- For example:
  - Distance between sensors = 10 meters.
  - Time taken = 0.25 seconds.
  - Speed = (10 ÷ 0.25) × 3.6 = 144.0 km/h.

Step 4: LCD Display Output

The system provides continuous feedback on the LCD (16x2 with I2C interface)**:**

- During login → Displays login messages.
- After login → Displays real-time speed and alerts.
- Messages are clear and easy to understand.

Step 5: Buzzer Alert Integration

- A buzzer is connected to Arduino pin 13.
- If the speed is greater than the predefined limit (e.g., 50 km/h):
  - Arduino sends a HIGH signal to buzzer → Alarm sound.
- If speed is normal:
  - Buzzer remains OFF.

This ensures immediate warning when over speeding occurs.

Step 6: Complete Workflow Summary

1. System powers on.
2. Login screen appears.
3. User enters username and password**.**
   - If correct → Access Granted → System starts.
   - If wrong → Access Denied → System stops.
4. Vehicle passes IR Sensor 1 → Timer starts.
5. Vehicle passes IR Sensor 2 → Timer stops.
6. Arduino calculates speed using formula.
7. Speed is shown on LCD.
8. If over speeding → Buzzer ON + ALERT on LCD.

3.2 Results and Discussion

After building the project and running multiple test cases, the system produced accurate and reliable results.

Case 1: Login Failure (Unauthorized User)

===== Login Required =====
Enter Username: abc
Enter Password: wrong123
Access Denied! Please Try Again.

The system does not proceed further. Safe from unauthorized use.

Case 2: Login Success (Authorized User)

===== Login Required =====
Enter Username: admin
Enter Password: 1234
Access Granted! System Starting...

System starts speed detection.

Case 3: Normal Speed (Within Limit)

- Distance = 10 meters
- Time = 0.87 seconds
- Speed = (10 ÷ 0.87) × 3.6 ≈ 41.5 km/h

LCD Output:

Vehicle Speed = 41.5 Km/Hr - Normal

No buzzer sound.

Case 4: Over speeding Example 1

- Distance = 10 meters
- Time = 0.25 seconds
- Speed = (10 ÷ 0.25) × 3.6 = 144.0 km/h

LCD Output:

ALERT: Vehicle Speed = 144.0 Km/Hr - OVER SPEEDING!

Buzzer turns ON immediately.

Case 5: Over speeding Example 2

- Distance = 10 meters
- Time = 0.37 seconds
- Speed = (10 ÷ 0.37) × 3.6 = 98.4 km/h

LCD Output:

ALERT: Vehicle Speed = 98.4 Km/Hr - OVER SPEEDING!

Buzzer turns ON immediately.

3.3 Discussion of Results

- **Accuracy:** The system calculated speeds very close to expected values.
- **Reliability:** The buzzer triggered every time speed exceeded the threshold.
- **Security:** Unauthorized login attempts were blocked.
- **Ease of Use:** Only simple login and automatic sensor detection are required.
- **Educational Value:** The project demonstrates the use of sensors, microcontrollers, and real-time alerts effectively.

Chapter 4

Engineering Standards and Mapping

4.1 Impact on Society, Environment and Sustainability

I. Social Impact:

The Vehicle Speed Checker Project addresses one of the most critical issues in society: road safety. Speeding is a major cause of accidents, resulting in injuries, loss of life, and property damage. By implementing an affordable and easy-to-deploy system:

- It can reduce road accidents by alerting about over speeding instantly.
- It can be placed at school zones, highways, accident-prone areas, and checkpoints to monitor vehicle speed.
- It increases awareness among drivers about maintaining safe speed limits.
- In the long term, it can encourage more responsible driving habits, reducing accident rates in society.

II. Environmental Impact:

While the project is primarily aimed at road safety, it has an indirect but important environmental impact. Over speeding not only causes accidents but also leads to:

- **Higher fuel consumption:** Driving at excessive speeds burns more fuel.
- **Increased emissions:** Over speeding vehicles emit more CO_2 and pollutants.
- **Noise pollution:** Vehicles at high speeds generate more noise, disturbing nearby residents.

By discouraging over speeding, the project indirectly contributes to:

- Reduced fuel wastage
- Lower greenhouse gas emissions
- Reduced noise pollution

III. Sustainability:

The project is highly sustainable because:

- It uses low-cost, reusable electronic components.
- It consumes very little power (Arduino Uno, LCD, IR sensors operate on 5V).
- It can be replicated and scaled to larger traffic monitoring systems.
- It can be integrated with government traffic systems in the future for automated enforcement.

4.1.1 Impact on Life

The impact of this project on human life is very significant:

- Safety First: By detecting over speeding, it can help prevent accidents and save lives.
- Driver Awareness: Drivers become more conscious about their speed when they hear a buzzer alert or see warnings on the LCD.
- Law Enforcement Aid: Traffic police can use this system at checkpoints as an additional tool to monitor speeds.
- Community Benefit: Safer roads mean fewer accidents, less loss of productivity, and improved quality of life for citizens.

For example: If this system were placed near a school zone, it could prevent tragic accidents involving children by alerting drivers in real time.

4.1.2 Impact on Society & Environment

Societal Benefits:

- Reduces accident-related injuries and deaths.
- Decreases traffic congestion caused by accidents.
- Promotes a culture of responsibility and discipline among drivers.

Environmental Benefits:

- Discourages aggressive driving which consumes more fuel.
- Reduces vehicular carbon footprint.
- Helps in developing eco-friendly transport policies by providing real-time speed monitoring data.

Thus, the project plays a dual role: saving lives and protecting the environment.

4.1.3 Ethical Aspects

Ethics are essential when dealing with safety and monitoring systems. This project upholds several ethical considerations:

1. Fair Use: The system alerts all drivers equally, without bias. Every driver exceeding the speed limit receives the same warning.
2. Data Security: The login system ensures that only authorized personnel can operate the system. Unauthorized use is blocked.
3. Transparency: The speed is displayed on the LCD in real time, so there is no hidden processing. Drivers and operators can directly see the results.
4. Non-Discrimination: The system does not discriminate based on type of vehicle, driver, or situation — it is rule-based and equal for all.
5. Educational Purpose: As an academic project, it spreads knowledge about electronics, programming, and road safety.

4.1.4 Sustainability Plan

The sustainability of the project can be ensured through several approaches:

1. Eco-Friendly Components: Using low-power microcontrollers and sensors minimizes electricity consumption.
2. Long-Term Deployment: The system can be installed at critical road sections and run continuously with solar power integration.
3. Scalability: The design can be expanded by adding GSM modules for sending speed data to a central traffic control center.
4. Awareness Campaigns: Partnering with traffic police or community organizations to use the system in awareness programs.
5. Data Logging: Future versions can store overspeed records, helping in traffic management and accident prevention policies.

4.2 Project Management and Team Work

This project was developed collaboratively by the team, following an Agile project management approach.

- Role Distribution:
  - Hardware Team: Responsible for assembling Arduino, sensors, buzzer, and LCD on breadboard.
  - Software Team: Developed Arduino code for login, sensor reading, calculations, and output display.
  - Testing Team: Conducted multiple test runs with different speeds and analyzed the results.
  - Documentation Team: Prepared diagrams, explanations, and report writing.
- Agile Sprints:
  - Sprint 1: Component collection and initial research.
  - Sprint 2: Breadboard assembly and sensor calibration.
  - Sprint 3: Login system coding.
  - Sprint 4: Speed calculation and output testing.
  - Sprint 5: Final integration of buzzer and LCD.
  - Sprint 6: Results analysis and documentation.

This structured teamwork ensured that every stage was properly tested before moving forward.

4.3 Complex Engineering Problem

4.3.1 Mapping of Program Outcome

In this section, provide a mapping of the problem and provided solution with targeted Program Outcomes (PO's).

Table 4.1: Justification of Program Outcomes

PO's	Justification
PO1	Justification of PO1 attainment
PO2	Justification of PO2 attainment
PO3	Justification of PO3 attainment

4.3.2 Complex Problem Solving

In this section, provide a mapping with problem solving categories. For each mapping add subsections to put rationale (Use Table 4.2). For P1, you need to put another mapping with Knowledge profile and rational thereof.

Table 4.2: Mapping with complex problem solving.

EP1 Dept of Knowled ge	**EP2** Range of Conflicti ng Requirem ents	**EP3** Depth of Analysis	**EP4** Familiari ty of Issues	**EP5** Extent of Applicabl e Codes	**EP6** Extent Of Stakehold er Involvem ent	**EP7** Inter-dependen ce
✓	✓	✓	✓	✓	X	✓

4.1.1 Engineering Activities

In this section, provide a mapping with engineering activities. For each mapping add subsections to put rationale (Use Table 4.3).

Table 4.3: Mapping with complex engineering activities.

EA1 Range of resources	**EA2** Level of Interaction	**EA3** Innovation	**EA4** Consequences for society and environment	**EA5** Familiarity
✓	✓	✓	✓	✓

Chapter 5

Conclusion

5.1 Summary

This project successfully designed and implemented a Vehicle Speed Checker using readily available digital electronics: an Arduino Uno, two IR sensors, an I2C 16×2 LCD, a buzzer, and a breadboard with jumper wires. The system measures the time a vehicle takes to travel between two fixed points and computes speed using the relationship

$$\text{Speed (km/h)} = \frac{\text{Distance (m)}\ X\ 3.6}{\text{Time } (s)}$$

Hardware integration was kept simple and educational: the IR sensors detect vehicle passage, the Arduino performs timing and speed calculation, the LCD shows live speed/status, and the buzzer provides an audible over-speed alert (threshold set at 50 km/h in this build). The wiring used analog pins A0 and A1 for the sensors, digital pin 13 for the buzzer, and A4/A5 for LCD SDA/SCL via I²C—choices that balanced clarity with minimal pin count.

Overall, the prototype met its learning and functionality goals: it detects, calculates, and alerts effectively, demonstrating practical application of sensors, microcontroller timing, and serial display interfaces in a compact system.

5.2 Limitation

While effective as a lab prototype, the current system has several constraints:

1. Fixed geometry & calibration: Accuracy depends on a known, fixed sensor spacing (10 m assumption). Any mis-measurement or misalignment adds error.
2. Single-lane, single-object assumption: It assumes one vehicle passes cleanly between sensors; crossing/overlapping objects can cause false triggers.
3. Sensor susceptibility: IR modules can be affected by ambient light, reflective surfaces, dust/rain, and object height, which may cause missed or premature detections.
4. Timing & resolution limits: With short baselines and hobby-grade sensors, small timing errors translate into noticeable speed error—especially at higher speeds.
5. Static speed threshold: The 50 km/h limit is hard-coded; adapting to different zones requires reprogramming.
6. No data retention/telemetry: The prototype does not log or transmit speed events for later analysis.
7. Minimal HMI: A 16×2 LCD restricts how much feedback (menus, diagnostics, averaging windows) can be shown at once.
8. Physical robustness: Breadboard-based wiring is not weatherproof or vibration-resistant; outdoor deployment would require enclosures and proper mounting.

5.3 Future Work

To bridge the gap from prototype to a robust field system, the following enhancements are recommended:

1. **Precision & ruggedization**
    - Use mechanical fixtures to fix and verify baseline distance.
    - Add weatherproof enclosures, proper cable management, and optical shrouds to reduce ambient-light interference.
2. **Better sensing & validation**
    - Add a third sensor or a different modality (e.g., break-beam photodiodes, reflective optical gates, or ultrasonic) for redundancy and false-trigger rejection.
    - Implement debounce/state machines and time-window plausibility checks to filter spurious events.
3. **Configurable thresholds & UI**
    - Provide a menu on the LCD (or a small keypad/encoder) to set the speed limit and units without reflating code.
    - Show diagnostics (sensor status, last N readings, average/median) for easier calibration.
4. **Data logging & analytics**
    - Store readings to EEPROM/SD card with timestamps (add an RTC), or send to a PC/mobile app for plots and reports.
    - Compute summary statistics (min/mean/max, 85th-percentile speed).
5. **Connectivity**
    - Add Bluetooth/Wi-Fi for live dashboards, remote configuration, and OTA updates.
6. **Scalability**
    - Extend to multi-lane monitoring and add vehicle counting/classification heuristics.
7. **Calibration & self-test**
    - Include guided calibration routines (sensor alignment checks, baseline verification) and self-test on startup.
8. **Compliance considerations**
    - If intended for public use, investigate accuracy standards, signage, and data-privacy considerations, and perform field calibration against a reference instrument.

References

[1] Mahesh, B., Avinash, T., Jagadeesh, R., & Dr. Y. Srinivasulu (2023). Vehicle Speed Detection using Arduino and IR Sensors. IJRASET: International Journal for Research in Applied Science & Engineering Technology.(IJRASET)

[2] Lae Yin Mon & Khin Khin Saw (2019). Design and Construction of Speed Detection System for Vehicles. International Journal of Trend in Scientific Research and Development (IJTSRD), Vol. 3 Issue 3, pp. 803–805.(studylib.net)

[3] (IJTSRD Paper, Jan–Apr 2019)**:** Design and Construction of Speed Detection System for Vehicles(ResearchGate)

[4] Prof. Lalit Kumar (2020). Vehicle Speed Detection System using IR Sensor. International Journal for Research in Applied Science and Engineering Technology,(ResearchGate)

[5] G. Pavan Kumar (2021). Speed Sensing Machine using Arduino. International Journal for Research in Applied Science and Engineering Technology, Vol. 9(VI): 4394–4401(ResearchGate)