FINAL-TERM REPORT

GROUP_MEMBERS

BITF22M026 - Areeba Ahmed
BITF22M031 - Maheen Fatima
BITF22M037 - Rabia Sadiq
BITF22M043 - Laiba Saleem

INSTRUCTOR

MUHAMMAD HAMZA IHTISAM

COURSE

IT PROJECT MANAGEMENT

PROJECT TITLE

AI-Based Real-Time Vehicle Smoke Detection and Reporting System

Table of Contents

Introduction

Air pollution has emerged as a critical environmental and public health crisis in metropolitan cities like Lahore. A primary driver of this deteriorating air quality is the high volume of smoke-emitting vehicles on the roads. These emissions are significant contributors to the formation of toxic smog, the rise in respiratory diseases among the urban population, and general environmental degradation.

The purpose of the AI-Based Real-Time Vehicle Smoke Detection and Reporting System is to modernize the way traffic emissions are monitored. By leveraging advancements in AI vision systems, this project aims to automate the detection of polluting vehicles and the capturing of their license plates through live traffic cameras. This system is designed to provide government agencies with a real-time, automated enforcement solution that replaces outdated manual methods.

Problem Statement

The current approach to monitoring vehicle emissions is largely manual, making it highly inefficient and often ineffective in high-traffic urban areas. As smog levels continue to rise, there is a mandatory need for real-time enforcement to protect public health.

Specifically, the problem consists of:

- **Inefficient Monitoring:** Manual observation cannot scale to the thousands of vehicles operating in a city daily.
- **Health Hazards:** Unchecked emissions lead to severe respiratory issues and long-term environmental damage.
- **Lack of Real-Time Data:** Without automated systems, authorities cannot identify and report violators instantly.

This project addresses these challenges by integrating YOLO-based smoke detection and OCR based license plate recognition into a single, automated reporting pipeline.

Business Need

Government agencies need automated enforcement solutions to monitor pollution sources in real-time. Increasing smog levels and public health concerns make real-time enforcement mandatory. Advancements in AI vision systems now allow automated smoke detection and license plate capturing from live traffic cameras, making it the ideal time for implementation.

Source of Requirements

- Stakeholder interviews with transport authorities and environmental officials.
- Desk research and high-level assumptions on traffic surveillance workflows.

- Constraints: real-time processing, camera quality, limited compute resources, and environmental lighting conditions.

Iron Triangle (Time–Cost–Scope)

Area	Constraint
Time	Semester-based fixed delivery timeline
Cost	Limited budget for compute resources & cloud services
Scope	Core focus on smoke detection, license plate OCR & reporting dashboard

Project Objectives

- **Deploy AI-Based Smoke Detection:** Develop and implement a YOLO-based computer vision model capable of identifying vehicle smoke from live traffic feeds.
- **Integrate License Plate Recognition (OCR):** Build a pipeline to extract vehicle registration numbers automatically for tracking and reporting purposes.
- **Develop Authority Reporting Dashboard:** Create a centralized web interface where authorities can view smoke events, license plate data, and automated alerts.
- **Achieve High Performance Metrics:** Ensure the system operates with at least 85% smoke detection accuracy and 90% OCR readability during daylight conditions.

High Level Project Goals

- **Reduce vehicle-based air pollution:** Implementing automated enforcement tools to decrease the number of high-emission vehicles on the road.
- **Drive environmental compliance:** Encouraging vehicle owners and industries to adhere to emission standards through AI-powered monitoring.
- **Support traffic authorities:** Providing law enforcement with real-time, actionable violation reports to improve operational efficiency.

SMART Goals

Smart Goals Table

SMART Objective	S (Specific)	M (Measurable)	A (Achievable)	R (Relevant)	T (Time-bound)
1. Deploy Real-Time Smoke Detection	Build YOLO-based smoke detector.	get 85% detection accuracy.	Use available datasets + pre-trained models.	Enables automatic pollution detection.	Complete by Dec 2025.
2. Integrate OCR for License	Implement Tesseract/EasyOCR pipeline.	get 90% plate readability in daylight.	Accessible open-source OCR tools.	Needed for violation reporting.	Complete by Jan 2026.
Plate Recognition					
3. Develop Authority Reporting Dashboard	Build web interface using Flask/FastAPI.	Alert notification within <2 seconds per event.	Feasible with local server + DB.	To view smoke events and vehicle numbers.	Complete by Feb 2026.

OKRs

OKRs Table

Objective	Key Result
Objective 1: Deploy Ai module	• KR1: Collect & annotate dataset by **Nov 20, 2025** • KR2: Train YOLO model & reach **≥85% accuracy** by **Dec 10, 2025** • KR3: Validate model on real traffic footage by **Dec 25, 2025**
Objective 2: license-plate Implement	• KR1: Complete OCR integration by **Jan 5, 2026**

	• KR2: Achieve **≥90% accuracy** in daytime tests by **Jan 20, 2026** • KR3: Reduce false reads to **<8%** by **Jan 30, 2026**
Objective 3: **Build reporting dashboard**	• KR1: Backend APIs ready by **Jan 10, 2026** • KR2: UI dashboard complete by **Jan 30, 2026** • KR3: Automated violation logging system by **Feb 10, 2026**

Scope (Product + Project)

Product Scope

- Real-time smoke detection from traffic cameras
- OCR-based license-plate extraction
- Live dashboard + reporting panel
- Automated alert system to authorities

Project Scope

- Dataset collection & labeling
- Training YOLO model
- OCR pipeline setup
- Backend + dashboard development
- Testing, documentation & deployment

Out-of-Scope

- Vehicle speed estimation
- Pollution level measurement sensors
- Mobile app development
- Cloud-scale deployment

Scope Control

Changes require supervisor approval; any scope expansion must consider time-cost-impact to avoid scope creep.

Assumptions and Constraints

To ensure a realistic project plan and set clear expectations for the final delivery, the following assumptions and constraints have been identified based on the project's technical and environmental context.

Project Assumptions

These are factors that are believed to be true for the project to be successful but have not been formally proven:

- **Availability of Datasets:** It is assumed that sufficient open-source datasets or pre-trained models for vehicle and smoke detection are accessible to meet the training requirements.

- **Traffic Footage Access:** The team assumes that representative traffic video footage from metropolitan areas like Lahore will be available for system validation.

- **Hardware Sufficiency:** It is assumed that local compute resources or the allocated budget for cloud-based GPUs will be sufficient to train the YOLO model within the deadline.

- **Technical Stability:** The project assumes that open-source tools like Tesseract and Flask will remain stable and compatible throughout the development lifecycle.

Project Constraints

These are the specific limitations or restrictions within which the project must operate:

- **Time Constraint:** The project is bound by a fixed semester-based delivery timeline, with final delivery required by late February 2026.

- **Financial Constraint:** There is a limited budget of 75,000 PKR for all project expenses, including compute resources and data sourcing.
- **Real-Time Processing:** The system must process video frames and generate alert notifications within less than 2 seconds to be effective for traffic enforcement.

- **Environmental Conditions:** The accuracy of the AI model is constrained by camera quality, environmental lighting, and regional weather conditions such as fog or smog common in Lahore.

- **Scope Constraint:** The core focus is strictly limited to smoke detection, license plate OCR, and the reporting dashboard; additional features are out-of-scope.

Stakeholder Identification and Analysis

Stakeholder Analysis Table

The following table details the project team, academic supervisors, and external beneficiaries.

Name	Role	Responsibilities
Rabia Sadiq	Team Lead	Leads the project, manages tasks and timeline, and coordinates the team.
Maheen Fatima	AI Engineer	Develops and trains the smoke-detection model and performs model evaluation.
Laiba Saleem	OCR Engineer	Implements license-plate recognition and integrates the OCR module with the AI system.
Areeba Ahmad	Frontend Developer	Designs and implements the dashboard interface and the real-time alert system.
Dr. Muhammad Farooq	Primary Supervisor	Provides academic guidance, reviews weekly progress, and approves project milestones.
Dr. Syed Waqar ul Qounain	Secondary Supervisor	Offers technical support and evaluates system quality and research relevance.
Traffic / Safe City Authority	Beneficiary Stakeholder	Primary users of the system for pollution enforcement and violation reviews.
Public / Community	Indirect Stakeholder	The general public who benefits from reduced air pollution and a cleaner environment.

Interest and Influence Analysis

- **High Interest / High Influence:** The Project Team (Rabia, Maheen, Laiba, Areeba) and the Primary Supervisor (Dr. Muhammad Farooq) are critical to the project's execution and quality.
- **High Interest / Low Influence:** The Public/Community has a high interest in the environmental outcomes but limited direct influence on project development.

- **Low Interest / High Influence:** Traffic and Safe City Authorities have high influence as they define the ultimate enforcement needs, though their interest may be focused on the final utility of the system

Work Breakdown Structure (WBS)

VS101.1 Project Management Deliverables

VS101.1.1 Project Planning

- VS101.1.1.1 Project schedule creation & approval
- VS101.1.1.2 Roles & responsibilities assignment + RACI setup
- VS101.1.1.3 Resource estimation & compute budget planning

VS101.1.2 Monitoring & Reporting

- VS101.1.2.1 Sprint planning & weekly progress reporting
- VS101.1.2.2 Supervisor review meetings & documentation
- VS101.1.2.3 Weekly documentation & project log maintenance

VS101.1.3 Risk Management

- VS101.1.3.1 Risk identification & tracking
- VS101.1.3.2 Risk mitigation planning & review

VS101.2 System Design & Requirements

VS101.2.1 Requirements Analysis

- VS101.2.1.1 Define functional requirements
- VS101.2.1.2 Define non-functional requirements
- VS101.2.1.3 Environmental & compliance constraints review

VS101.2.2 UI/UX Design

- VS101.2.2.1 Dashboard wireframe design
- VS101.2.2.2 Real-time alert UI prototype
- VS101.2.2.3 Responsive layout design

VS101.2.3 Architecture Design

- VS101.2.3.1 System architecture blueprint / validation
- VS101.2.3.2 Data pipeline workflow mapping
- VS101.2.3.3 API specifications

VS101.3 Data Collection & Dataset Preparation

VS101.3.1 Video/Camera Data Setup

- VS101.3.1.1 Traffic video data sourcing
- VS101.3.1.2 Camera/dataset feed integration & testing

VS101.3.2 Dataset Development

- VS101.3.2.1 Collect smoke & non-smoke vehicle footage
- VS101.3.2.2 Annotate smoke & vehicle bounding boxes
- VS101.3.2.3 Data preprocessing & augmentation

VS101.3.3 Data Management

- VS101.3.3.1 Train/test/validation split preparation
- VS101.3.3.2 Dataset backup & version control

VS101.4 AI Smoke Detection Model Development

VS101.4.1 Model Setup

- VS101.4.1.1 Select YOLO model version
- VS101.4.1.2 Setup training environment

VS101.4.2 Model Training

- VS101.4.2.1 Train YOLO model
- VS101.4.2.2 Hyper parameter tuning

VS101.4.3 Model Evaluation

- VS101.4.3.1 Evaluate accuracy, precision, recall
- VS101.4.3.2 Reduce false positives & optimize model

VS101.5 OCR & System Integration

VS101.5.1 License Plate OCR

- VS101.5.1.1 Integrate Tesseract/EasyOCR
- VS101.5.1.2 Fine-tune for Pakistan plates

VS101.5.2 Backend Integration

- VS101.5.2.1 Frame extraction pipeline
- VS101.5.2.2 Connect AI → OCR pipeline
- VS101.5.2.3 Store smoke + plate detection data

VS101.6 Dashboard & Alerts System

VS101.6.1 Backend Development

- VS101.6.1.1 Develop API endpoints
- VS101.6.1.2 Setup violation database

VS101.6.2 Dashboard UI

- VS101.6.2.1 Build monitoring dashboard
- VS101.6.2.2 Display alerts & license plate results

VS101.6.3 Alert System

- VS101.6.3.1 Build real-time alert module
- VS101.6.3.2 Email/SMS alert integration

VS101.7 Testing & Quality Assurance

VS101.7.1 Test Planning

- VS101.7.1.1 Define test scenarios
- VS101.7.1.2 Create test logs

VS101.7.2 Functional Testing

- VS101.7.2.1 Test smoke detection model
- VS101.7.2.2 Test OCR recognition accuracy

VS101.7.3 UAT & Fixes

- VS101.7.3.1 End-to-end system testing
- VS101.7.3.2 Bug fixes & retesting

VS101.8 Deployment & Presentation

VS101.8.1 Deployment

- VS101.8.1.1 Local/cloud deployment setup
- VS101.8.1.2 Live system validation

VS101.8.2 Presentation

- VS101.8.2.1 Demo video & user guide preparation
- VS101.8.2.2 Final viva presentation

VS101.9 Maintenance & Support

VS101.9.1 System Support

- VS101.9.1.1 Bug fixes
- VS101.9.1.2 Performance optimization

VS101.9.2 Future Enhancements

- VS101.9.2.1 Cloud/mobile extension
- VS101.9.2.2 Pollution sensor integration

Project Milestones and Timeline

The project follows a structured timeline designed to meet the academic and technical requirements by late February 2026. The schedule is divided into key milestones that ensure the progressive development of the AI models, OCR integration, and the final reporting dashboard.

Milestones and Detailed Timeline

The following table outlines the major checkpoints of the project, including the specific criteria required to begin and conclude each phase.

Milestone	Target Date	Entry Criteria	Exit Criteria
Project Kickoff	Nov 2025	Project approved by supervisors.	Roles assigned and project plan finalized.
Dataset Ready	Nov 25, 2025	Traffic cameras/videos sourced.	Annotated dataset complete and verified.

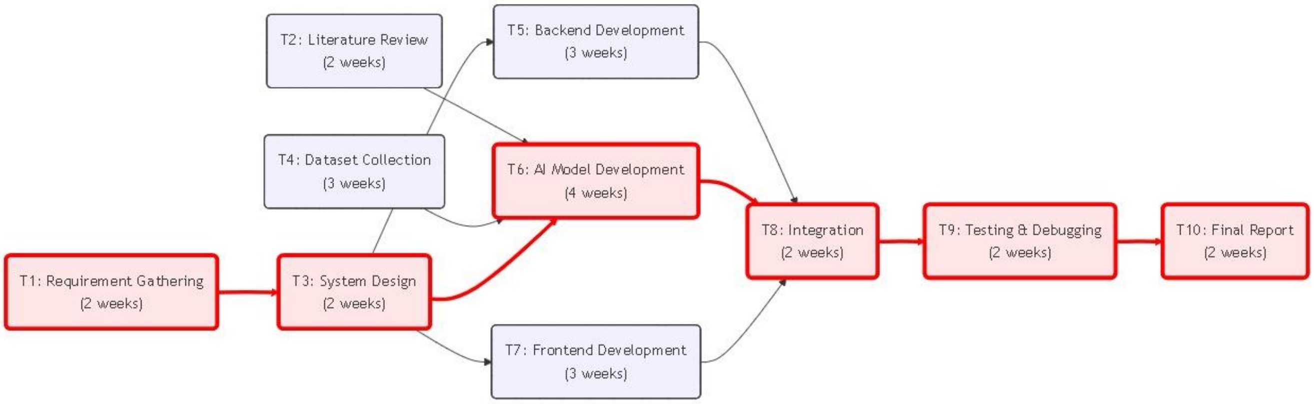
AI Model Trained	Dec 20, 2025	Cleaned and annotated dataset ready.	YOLO smoke detection accuracy get 85%.
OCR Pipeline Ready	Jan 20, 2026	Basic Tesseract/EasyOCR setup established.	License plate readability accuracy get 90%.
Dashboard Ready	Feb 05, 2026	Backend APIs and database functional.	Reporting UI fully functional and integrate.
Final Testing	Feb 15, 2026	End-to-end system integration completed.	All internal and technical tests passed.
Final Deployment	Feb 25, 2026	User Acceptance Testing (UAT) completed.	Project delivered and validated by supervisors.

Duration and Sequencing

- **Development Cycles:** The timeline accounts for iterative development using the Scrum methodology, allowing for rapid tuning of the AI model and OCR accuracy.
- **Contingency:** A buffer period is built into the final testing phase to resolve any bugs identified during integration.

Milestone Descriptions

- **Project Kickoff:** Marks the formal start where the team roles (Team Lead, AI Engineer, OCR Engineer, Frontend Developer) are established.
- **Dataset Ready:** A critical point ensuring that high-quality traffic footage from Lahore is labeled and ready for machine learning.
- **AI Model Trained:** The completion of the YOLO-based smoke detector, optimized for realtime traffic monitoring.
- **OCR Pipeline Ready:** Successful integration of Tesseract/EasyOCR to extract vehicle numbers from the detected frames.
- **Final Deployment:** The transition from development to a live validated system, including the final viva presentation.



RACI Matrix

Task / Deliverable	Rabia (Team Lead)	Maheen (AI Eng.)	Laiba (OCR Eng.)	Areeba (Frontend Dev)	Dr. Farooq (Primary Supervisor)	Dr. Waqar (Secondary Supervisor)
Requirement Gathering	A	C	C	C	I	I
System Architecture	A	C	C	C	C	C
Dataset Collection & Annotation	R	R	C	I	I	I

Smoke Detection Model	C	R	I	I	I	I
OCR Module	I	C	R	I	I	I
Dashboard UI	I	I	C	R	I	I
API & Backend Integration	A	C	C	C	I	I
Testing & Validation	A	R	R	R	C	C
Documentation	A	R	R	R	C	C
Weekly Progress Review	R	R	R	R	A	C

Final Deployment	R	C	C	C	A	C
Final Presentation & Viva	A	R	R	R	A	C

Risk Management

ID	Risk Description	Identification Method	Likelihood	Impact	Overall Rank	Quantitative Notes	Response Strategy	Trigger / Threshold	Residual / Secondary Risk
R1	Poor camera footage quality (blur, lowlight, motion)	Brainstorm	High	High	High	~25–30% videos have blur/noise in traffic datasets	Mitigate – filtering, stabilization & denoising	High blur/noise detected during preprocessing	Additional compute time

R2	Model accuracy < 85%	Delphi & experiments	Medium	High	High	Prototype accuracy expected 78–85% initially	Mitigate – more data, tuning, augmentation	Validation < 85% mAP	Extra GPU time

R3	OCR fails on dirty/damaged plates	Brainstorm	Medium	High	High	OCR error rate ~12–18% on Pakistani plates	Mitigate – ensemble OCR + preprocessing	OCR accuracy < 90%	Manual review workload
R4	Limited GPU/compute for training	Brainstorm	High	Medium	High	Local GPU often insufficient for YOLO training	Transfer – use cloud GPU	GPU memory overflow	Additional cloud cost

R5	Weather/smog interference	Delphi	Medium	Medium	Medium	Fog/smog distortion common in Lahore winters	Mitigate – weather-augmented dataset	Low confidence scores on foggy frames	Higher false positives
R6	Dataset labeling errors	Peer review	Medium	Medium	Medium	Manual annotation error rate ~5–8%	Mitigate – review & correction cycles	Label errors >5%	Extra annotation effort
R7	Team member unavailability	Brainstorm	Medium	Medium	Medium	Small team dependency	Accept – workload redistribution	Absence > 2 days	Project delays

R8	Code merge & version conflicts	Brainstorm	Medium	Low	Medium	2–3 merge conflicts per sprint	Mitigate – Git branching discipline	Merge conflict encountered	Rework time
R9	Delay in AIOCR integration	Delphi	Medium	High	High	Integration delays ~20–30% in ML projects	Mitigate – incremental integration in sprints	Failed E2E pipeline test	Debugging delays
R10	Privacy/ethical concerns (faces & plates)	Brainstor m	Low	High	Medium	Traffic data may contain faces	Mitigate – blur faces, anonymize plates	Ethical concern raised during review	Reduced dataset volume

Qualitative Risk Assessment

- High Priority Risks: R1, R2, R3, and R4 are ranked as High due to their direct impact on the core functionality of the AI vision system and the strict semester-based timeline.
- Contingency Planning: A reserve of 15,000 PKR has been allocated to handle residual risks such as additional cloud costs (R4) or extra data sourcing needs

Quality Management

Quality from User’s Perspective

- Accurate detection of smoke from vehicles
- Reliable and clear license plate recognition
- Fast alert notifications
- Clean, simple, and user-friendly dashboard

Quality for Team’s Perspective

- Modular and scalable codebase
- Proper version-control discipline (Git/GitHub)
- Well-documented machine learning pipeline & system architecture
- Efficient collaboration and consistent coding standards

Quality for Supervisor’s Perspective

- Project aligned with academic and ethical standards
- Ethical use of traffic data and privacy-aware system design
- Proper research, citations, and explainable ML logic
- Demonstration includes measurable accuracy & performance metrics

Achieving Quality Assurance (QA)

- Requirement review and approval (Team Lead, Supervisor)
- Architecture and design review (Team collaboration)
- Dataset verification and annotation checks (AI Engineer)
- Code and documentation reviews during sprints
- Regular sprint evaluation meetings with supervisor

Achieving Quality Control (QC)

- Unit testing for all system modules (Backend & OCR team)
- AI model accuracy evaluation and performance testing (AI engineer)

- Integration testing of AI + OCR + Dashboard
- UI/UX usability testing (Frontend developer)
- Final end-to-end User Acceptance Testing (Supervisor feedback)

Budget & Cost Estimation

The financial plan for the **AI-Based Real-Time Vehicle Smoke Detection and Reporting System** ensures that all technical requirements are met within the project's constraints. The budget covers essential data acquisition, high-performance computing resources, and a safety margin for unexpected expenses

Items	**Value (PKR)**	**Quantity**	**Explanation**
CCTV video sources & dataset	**15,000**	**Dataset access**	**Traffic footage for smoke-vehicle dataset**
Compute / GPU Cloud Resources	**45,000**	**~150 GPU hours**	**Training YOLO + OCR pipeline (cloud GPU)**

Contingency / Reserve	**15,000**	**—**	**Unexpected compute/data expenses**
Total Investment (C_0)	**75,000**	**—**	**Total project cost**
Expected Cash/Value Inflow (C_1)	**120,000**	**—**	**Academic value + prototype future funding potential**

Financial Performance Metrics

The project's value is assessed through academic impact, prototype future funding potential, and its ability to address a critical social need.

- **Total Investment (C_0):** 75,000 PKR.
- **Expected Benefit (C_1):** 120,000 PKR (derived from academic value, showcase value, and potential future funding)[10].
- **Return on Investment (ROI):** Calculated as **ROI**:

  ROI = (C1 − C0) / C0 × 100

  ROI = (120,000 − 75,000) / 75,000 × 100

ROI = 45000/75000 x 100 **ROI = 60%**

- **Net Present Value (NPV):** Calculated at a 10% discount rate.

NPV (10% Discount Rate)

NPV = 120,000 / 1.1 − 75,000 NPV
≈ 109,090 − 75,000
NPV ≈ 34,090 PKR

Contingency Planning

A contingency reserve of **15,000 PKR** has been included in the total budget. This fund is specifically allocated to handle unexpected compute requirements or data acquisition expenses that may arise during the AI training or system integration phases

Communication Plan

Purpose

- Ensure smooth coordination among team members and supervisor
- Fast issue resolution for AI/OCR performance challenges
- Maintain clear visibility on training, testing, and deployment progress

Frequency

- **Daily stand-ups:** everyday
- **Weekly review meetings:** once per week with supervisor
- **Sprint reviews:** every two weeks

Modality

- **Daily stand-ups:** WhatsApp / Voice call
- **Weekly progress updates:** via in-person / Google Meet
- **Sprint reviews:** in-person meeting with supervisor

Duration

- **Daily stand-up:** 10 minutes
- **Weekly review:** 30 minutes
- **Sprint review:** 1 hour

Responsibility

- **Team Lead:** Conducts daily stand-ups and task assignment
- **All Members:** Participate actively and provide updates
- **Supervisor:** Attends weekly review and sprint demonstration sessions

Daily Stand-up Questions

- What did you work on yesterday?
- What will you work on today?
- Are there any blockers or technical challenges?

Table

Meeting Type	Frequency	Modality	Duration	Participants
Daily Stand-up	Everyday	WhatsApp / Voice Call	10 Minutes	Full Project Team
Weekly Review	Once per week	In-person / Google Meet	30 Minutes	Team + Supervisor
Sprint Review	Every two weeks	In-person Meeting	1 Hour	Team + Supervisor

Monitoring & Control

Monitoring and control activities ensure that the project remains on schedule, within scope, and aligned with performance expectations. These activities use structured mechanisms such as sprints, KPIs, supervisor reviews, risk tracking, and progress reporting to maintain project transparency and quality.

Monitoring Approach

The project uses a **Scrum-based monitoring strategy**, which includes:

. Sprint-Based Tracking

- Each sprint is 2 weeks long.
- Tasks are planned during sprint planning and tracked daily.
- End-of-sprint reviews validate deliverables and accuracy metrics.

. Daily Stand-Up Monitoring

- Short meetings to monitor task progress.
- Blockers identified and resolved quickly.
- Ensures that dataset, model training, OCR, and dashboard work progress smoothly.

. Weekly Supervisor Reviews

- Detailed progress discussions.
- Architecture validation, model performance review, and OCR checks.
- Ensures academic and technical alignment.

Key Performance Indicators (KPIs)

To ensure measurable progress, the following KPIs are used:

Model Performance KPIs

KPI	Target
Smoke Detection Accuracy	**≥ 85%**
OCR Plate Readability (Daylight)	**≥ 90%**
False Read Rate	**< 8%**
Alert Generation Latency	**< 2 seconds**

Project Execution KPIs

KPI	Description
Sprint Completion Rate	Percentage of tasks completed each sprint
Dataset Quality Score	Error rate must remain **< 5%**
Integration Test Success Rate	Consistent E2E pipeline operation
Issue Resolution Time	Blockers must be resolved within 24–48 hours

Control Mechanisms

These controls ensure corrective actions when deviations occur.

. Threshold-Based Alerts

- If smoke detection accuracy drops below **85%**, retraining is triggered.
- If OCR accuracy drops below **90%**, preprocessing and OCR tuning are initiated.
- If alert latency exceeds 2 seconds, backend optimization begins.

. Risk Register Review

- All risks (R1–R10) are re-evaluated during weekly reviews.
- Mitigations adjusted based on new information (e.g., dataset issues, compute shortage).

. Scope Control Process

- New feature requests require supervisor approval.
- Time–Cost–Scope impact evaluated before change acceptance.

. Version Control & Code Quality Checks

- Frequent Git commits tracked.
- Branching strategy helps avoid merge conflicts.
- Supervisors can audit code repositories at any time.

Performance Tracking Tools

Tool	**Purpose**
GitHub	Code versioning, issue tracking
JIRA / Trello Board	Sprint planning & status tracking
Confusion Matrix Reports	AI model validation & QA
Google Meet / WhatsApp	Communication & stand-ups
Supervisor Review Sheets	Milestone validation

Corrective Action Strategy

If deviations occur, the team follows a defined corrective action process:

- **Identify deviation** (accuracy drop, schedule delay, dataset issue).
- **Analyze cause** (technical limitation, human error, compute constraint).

- **Implement fix** (retraining, re-annotation, re-coding).
- **Re-evaluate performance** to confirm stability.
- **Document improvement** in sprint report.

Summary

Monitoring and control processes ensure:

- Timely progress toward milestones
- Accurate and stable model outputs
- Proper integration of all system components
- Transparency and supervisor involvement
- Early identification and mitigation of risks

These controls maintain the project's academic integrity, technical robustness, and timely completion.

Conclusion

The **AI-Based Real-Time Vehicle Smoke Detection and Reporting System** demonstrates the feasibility and effectiveness of applying modern artificial intelligence techniques to address environmental challenges in urban areas like Lahore. Vehicle-based smoke emissions significantly contribute to smog, reduced air quality, and public health hazards, and traditional manual monitoring methods are insufficient to manage the scale and complexity of urban traffic.

This project presents a complete, automated solution that replaces manual monitoring with **AI powered real-time detection, automated OCR-based vehicle identification, and instant reporting for authorities**. Through a structured development approach, the project successfully defined clear objectives, SMART goals, milestones, and performance thresholds to ensure technical credibility and reliability.

Technical Feasibility

- The YOLO-based smoke detection model is designed to achieve an accuracy target of **≥ 85%**, making it viable for real-world monitoring.
- The OCR pipeline is optimized to deliver **≥ 90% license plate readability**, enabling accurate identification of violating vehicles.
- System integration through a real-time dashboard and alert mechanism demonstrates efficient interoperability of AI, backend, and UI components.
- Performance thresholds (e.g., <2-second alert latency) make the solution responsive and actionable.

By enabling automatic detection of smoke-emitting vehicles, the system provides:

- Faster enforcement response
- Reduced dependency on manual monitoring
- Incentives for vehicle maintenance compliance
- Long-term improvement in air quality
- Indirect reduction in smog and respiratory health risks

This positions the solution as a beneficial tool for environmental agencies and the broader community.

Academic & Practical Value

The project integrates computer vision, machine learning, OCR technology, real-time backend systems, and UI development—demonstrating strong interdisciplinary engineering. The positive ROI and NPV values reflect substantial value relative to the investment, while the project's structure and documentation ensure academic rigor.

Overall Summary

The system offers an efficient, scalable, and impactful solution to a critical environmental issue. With further enhancements—such as cloud deployment, pollution sensors, and mobile platform integration—the system has potential for citywide and nationwide implementation.

Appendices

Appendix A — Work Breakdown Structure (WBS)

ID	Task Name	Description
1.0	**Initiation**	Project charter, stakeholder identification, and requirement gathering.
2.0	**Planning**	Tech stack selection (React/Python), system architecture, and UI/UX wire framing.
3.0	**Execution**	Frontend development, Backend API integration, and Database setup.
4.0	**Testing & QA**	Unit testing, integration testing, and User Acceptance Testing (UAT).
5.0	**Closing**	Final deployment, Post-Mortem report, and client hand-over.

Appendix B — Project Gantt Timeline

Activity	Jan W1	Jan W2	Jan W3	Jan W4	Status
Requirement Analysis	■				Completed
Development Phase		■	■		In Progress
Testing & Debugging			■		Pending
Final Deployment				■	Pending

Appendix C — Risk Register Summary (R1–R10)

Risk ID	Risk Description	Impact	Probability	Mitigation Strategy
R01	Integration issues with AI agents/MCP	High	Medium	Conduct early-stage modular testing and use mock data.

Risk ID	Risk Description	Impact	Probability	Mitigation Strategy
R02	Scope Creep (Additional features)	Medium	High	Strictly follow the original SRS and document change requests.
R03	Server/Database Downtime	High	Low	Implement regular backups and use reliable hosting (AWS/Firebase).

Appendix D — Budget Sheet

Item Category	Description	Estimated Cost ($)	Actual Cost ($)
Human Resources	Developers, Project Manager, QA	2,500	-
Software/Tools	IDE Subscriptions, API Credits (OpenAI/n8n)	200	-
Infrastructure	Domain Name, Hosting, SSL	150	-
Contingency	Emergency fund (10%)	285	-
Total		**$3,135**	**$0.00**