

VigilantEye – Project Specification Document

Group Name: AICS7

Team Name: EYEQ

1. Executive Summary

VigilantEye is an AI-powered multi-agent video intelligence system designed for real-time anomaly detection, violence recognition, unauthorized access monitoring, forensics, and privacy-compliant analysis.

The system includes a web dashboard, Telegram alerting, video summarization, and automated incident reporting.

2. Scope Statement

In-Scope:

- Real-time video analysis
- Multi-agent behavior recognition
- Telegram-based alert system
- Forensics & incident reporting tools
- Web dashboard interface

Out-of-Scope:

- Physical CCTV installation
- Commercial licensing
- Production deployment

3. Requirements & Deliverables

Functional Requirements:

- Anomaly Detection ($\geq 85\%$ accuracy)
- Real-time Telegram alerts (< 5 sec)
- Video summarization (PDF/JSON)
- Forensics tools

Non-Functional Requirements:

- User-friendly interface
- GDPR/HIPAA aligned privacy handling
- Scalable cloud integration
- 30 FPS real-time performance

4. Work Schedule & Timeline

Week	Milestone	Status
Week 3	Project Kickoff	Completed
Week 4	Requirements Finalized	Completed
Week 5–6	Architecture & Design	Completed
Week 7–8	Frontend Development	Completed
Week 9–10	Integration	Completed
Week 11	Testing & Optimization	Completed
Week 12	Final Delivery	Completed

5. Roles & Responsibilities

Team Member	Role	Responsibilities
Tanzima	Backend Lead	Audio intelligence, backend architecture
Sameer	Frontend Lead	UI/UX & dashboard
OM Patel	Data Analyst	Speech + vision analysis
Abdullah	Data Scientist	Alerts + ML models
Sukhjit	Full Stack Developer	APIs, segmentation, backend
Riya	Data Scientist	Face recognition
Sri Datta	QA & Tester	Environment detection & testing
Varisdeep Singh	Frontend Developer	Documentation, testing, UI support

6. Exit Criteria

- All modules integrated successfully
- Real-time detection functional
- Telegram alerts operational
- Forensics tools validated
- Accuracy benchmarks achieved
- Full documentation completed

7. Risks & Mitigation

- Model accuracy issues → Data augmentation
- Integration delays → Early testing, CI/CD
- Performance bottlenecks → Optimization, scaling
- Timeline pressure → Strict milestone control

8. Assumptions & Constraints

Assumptions:

- Team members available
- Open-source tools sufficient
- Cloud infrastructure functioning

Constraints:

- Zero budget
- Limited timeframe
- No hardware installation

9. Communication Plan

- Weekly meetings
- Daily WhatsApp updates
- Continuous GitHub commits
- Email communication with instructor
- Demo preparation in final week