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 (≥ 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

Milestones:

- ⇒ Week 3 – Kickoff
- ⇒ Week 4 – Requirements finalized
- ⇒ Week 5–6 – Architecture & design
- ⇒ Week 7–8 – Frontend development
- ⇒ Week 9–10 – Integration
- ⇒ Week 11 – Testing & optimization
- ⇒ Week 12 – Final delivery

5. Roles & Responsibilities

- Tanzima – Backend Lead
- Sameer – Frontend Lead
- OM Patel – Data Analyst
- Abdullah – Data Scientist (Alerts)
- Sukhjit – Full Stack Developer
- Riya – Face Recognition
- Sri Datta – QA & Testing
- Varisdeep – Frontend Developer, Documentation & Testing

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