**Proposal Document: ClasUtil – Smart CCTV Classroom Monitoring System**

**Submitted by:**  
Asingo Paul  
Junior AWS Cloud Engineer | Solutions Architect  
Nairobi, Kenya  
Email: [[YourEmail@example.com](mailto:YourEmail@example.com)]  
Date: July 20, 2025

### 1. Project Title

**ClasUtil: AI-Powered Real-Time Classroom Occupancy Monitoring Using CCTV Infrastructure**

### 2. Executive Summary

ClasUtil is an innovative and scalable classroom monitoring solution designed to optimize the utilization of classroom space using existing CCTV infrastructure. By integrating AI-based people detection, a real-time web dashboard, and a cloud backend, ClasUtil provides a unified interface for lecturers, administrators, and planners to instantly know which rooms are currently in use or free. This solution eliminates manual checks, enhances scheduling, and contributes to the broader vision of smart, data-driven campus operations.

### 3. Problem Statement

Many educational institutions experience inefficiencies in classroom scheduling and utilization due to a lack of real-time visibility into classroom occupancy. These challenges include:

* Frequent double bookings or unused scheduled rooms.
* Difficulty for lecturers and planners to find free spaces quickly.
* No central system to track classroom usage history.
* Underutilization of existing security infrastructure (CCTV).

Despite having CCTV cameras installed across classrooms, most schools only use them for security purposes. ClasUtil leverages this underutilized infrastructure to add value and provide continuous operational insights.

### 4. Proposed Solution

ClasUtil will integrate with the school's existing CCTV cameras to analyze real-time classroom footage and detect occupancy using AI-powered people detection models (e.g., YOLOv8). The data will be collected and visualized on a web-based dashboard that:

* Displays live status of each classroom: Occupied ❌ or Empty ✅.
* Includes advanced filters (e.g., Building, Floor, Time, Type).
* Offers secure access for authorized staff and stakeholders.
* Tracks historical usage to inform future decisions.

This solution will be lightweight, scalable, and cloud-friendly, capable of expanding across departments and campuses.

### 5. Technical Architecture

**5.1 CCTV Integration**

* Capture real-time footage via RTSP streams from IP cameras.
* Support for various camera vendors (Hikvision, Dahua, etc.).

**5.2 AI People Detection**

* Implement YOLOv8 or similar model to detect human presence.
* Frame capture interval: Configurable (e.g., every 10 seconds).
* Lightweight processing pipeline on Flask server or cloud VM.

**5.3 Backend Services (Flask + Supabase)**

* Flask backend to manage requests, AI inference, and camera registration.
* Supabase for secure storage of room metadata, status logs, and user roles.
* WebSocket or polling for real-time dashboard updates.

**5.4 Dashboard (Tailwind CSS + Bootstrap)**

* Web-responsive dashboard displaying:
  + Room name, block, occupancy status.
  + Status icons and color codes (Red = Occupied, Green = Free).
  + Search and filter options.
  + Graphs and analytics (optional in later versions).

### 6. Features Summary

| Feature | Description |
| --- | --- |
| Real-time Detection | Automatically identifies occupancy status via CCTV and AI. |
| Secure Dashboard | Role-based access for lecturers, planners, and admins. |
| Multi-Room Support | Handles 10s to 100s of classrooms. |
| Filter & Search | Easily find available rooms by name, block, or floor. |
| Historical Logs | Optionally track room usage over time. |
| Expandability | Future support for smart alerts, booking integration. |

### 7. Benefits to the Institution

* **Operational Efficiency**: Easily identify unused classrooms in real-time.
* **Improved Scheduling**: Allocate rooms more accurately based on actual usage.
* **Reduced Time Waste**: Lecturers locate available classes faster.
* **Data-Driven Decisions**: Utilize logs and analytics to improve infrastructure planning.
* **Enhanced ROI on CCTV**: Maximizes the value of existing surveillance systems.
* **Platform for Innovation**: Encourages student/staff involvement in campus tech solutions.

### 8. Implementation Roadmap

| Phase | Activity | Timeline |
| --- | --- | --- |
| Phase 1 | Camera inventory & RTSP access testing | Week 1 |
| Phase 2 | Set up Flask backend and Supabase integration | Week 2 |
| Phase 3 | Integrate YOLOv8 people detection | Week 3-4 |
| Phase 4 | Develop and test responsive dashboard UI | Week 4-5 |
| Phase 5 | Pilot run in select classrooms | Week 6 |
| Phase 6 | Full school-wide deployment & maintenance | Week 7+ |

### 9. Required Resources

**Human Resources**:

* 1 Backend Developer
* 1 AI/ML Engineer (for model tuning)
* 1 UI/UX Designer (for dashboard)
* 1 Network Technician (camera access support)

**Hardware/Software Requirements**:

* School-provided access to classroom IP cameras (RTSP enabled).
* On-premise server or cloud-hosted VM with GPU (optional).
* Internet access for cloud database (Supabase).
* Domain for dashboard URL (e.g., clasutil.schoolname.ac.ke).

### 10. Estimated Budget

| Item | Estimated Cost (KES) |
| --- | --- |
| Development & Engineering (6 weeks) | 120,000 |
| GPU-Enabled VM (if needed) | 5,000/month |
| Dashboard Hosting (1 year) | 2,000 |
| Supabase Free Tier (or Pro) | 0 - 2,500/month |
| Domain Registration (annual) | 1,000 |
| Training & Miscellaneous | 10,000 |
| **Total Estimate** | **140,000 - 150,000 KES** |

### 11. Risk Mitigation

| Risk | Mitigation Strategy |
| --- | --- |
| Network restrictions on RTSP | Collaborate with school IT to whitelist internal IPs |
| AI model false detections | Retrain model using local classroom footage |
| Power/Internet outages | Implement local caching; retry logic for sync |
| Privacy concerns | Only use stream metadata; no video recording stored |

### 12. Sustainability and Scalability

* Designed with modular architecture for easy future expansion.
* Can be integrated with school LMS or timetable systems.
* Open to contributions from IT students as a living academic project.
* Low recurring costs make it viable long-term.

### 13. Conclusion

ClasUtil offers a forward-thinking, practical solution to a common challenge in educational institutions: how to ensure classroom spaces are used efficiently. By utilizing AI and existing infrastructure, it reduces waste, supports better scheduling, and builds a smart campus foundation. We seek your support to implement this solution for the benefit of students, staff, and the future of digital education.

**Prepared by:**  
Asingo Paul  
AWS Cloud Engineer | Developer

Sure! Here's the rewritten **detailed proposal for ClasUtil** using large, structured paragraphs that explain the system comprehensively:

### ****Proposal for ClasUtil – A Smart Classroom Monitoring and Utilization System Using CCTV and AI****

#### ****Introduction****

As educational institutions expand in size and capacity, managing space usage—particularly classroom utilization—becomes increasingly complex. Many schools and universities face challenges in optimizing room occupancy, especially when classes are canceled, relocated, or underutilized. Often, valuable resources like classrooms remain empty without administrators or lecturers knowing, leading to inefficient space use and potential scheduling conflicts. To address this gap, we propose the implementation of **ClasUtil**, a smart, AI-driven digital system designed to monitor classroom occupancy using existing CCTV infrastructure. The platform provides real-time data visualization and analytical insights to help institutions optimize space allocation, improve teaching logistics, and make data-driven infrastructure decisions.

#### ****System Overview****

ClasUtil leverages the existing CCTV network in educational institutions and integrates it with AI-powered video analytics to determine the real-time status of classrooms—whether they are empty or occupied. The system processes live video feeds using computer vision models such as YOLOv8 to detect human presence in the rooms. These detections are interpreted into occupancy status and then sent to a centralized database (Supabase), which stores and manages this data. The frontend interface, built using HTML, CSS (with Tailwind for sleek design), and JavaScript, connects to this backend via a Flask web server. Users can interact with a responsive, mobile-friendly dashboard to view room occupancy status, trends over time, and filter data by building, floor, class type, or occupancy state.

#### ****Technical Architecture****

The system is architected to be modular and scalable. At the heart of the solution is the integration between AI-powered video processing and a modern web application stack. CCTV feeds are processed locally or through edge devices capable of running the YOLOv8 model, detecting the presence of people in real-time. This detection is converted into occupancy data, which is stored in a Supabase database with timestamps and room identifiers. The Flask backend fetches the latest data entries and renders them on a live dashboard using WebSocket or AJAX polling techniques to ensure that the UI is always updated without the need for manual refreshes.

The dashboard is equipped with filters allowing users to sort or view data based on buildings, floors, and class types. Moreover, occupancy insights are visualized using embedded charts that represent usage patterns, aiding decision-makers in identifying underutilized areas or peak usage times. All data is hosted securely, and access to the admin panel can be authenticated to restrict manipulation or unauthorized access.

#### ****Benefits to the Institution****

The implementation of ClasUtil stands to bring significant benefits to both administrative and academic stakeholders. First, it improves **space efficiency** by providing real-time insights into which rooms are being used or left idle, allowing reallocation for meetings, makeup classes, or group studies. Second, it enhances **logistical planning**, reducing double bookings and scheduling conflicts. Furthermore, the system supports **infrastructure planning**, as historical usage data can guide future investments in additional classroom blocks, renovation, or reassignments.

ClasUtil also contributes to **security and compliance**, as the system helps verify actual room usage against scheduled timetables. This can be useful during audits or for maintaining transparency. Additionally, lecturers and students can benefit from being able to quickly identify available classrooms when their scheduled rooms are unavailable, improving the overall academic experience.

#### ****Implementation Plan****

The first phase will involve collaboration with the IT department to integrate the CCTV feed into our processing pipeline. Where existing infrastructure lacks clarity or coverage, additional cameras may be installed. Once feeds are accessible, a machine learning model will be fine-tuned to detect occupancy with high accuracy across various lighting and classroom conditions. Simultaneously, the web interface will be customized for the school’s layout, naming conventions, and user permissions.

After system calibration and internal testing, the platform will be deployed for pilot testing in select buildings. Feedback from users will help refine filters, UI features, and performance. Upon successful validation, the system will be rolled out across the entire institution.

#### ****Conclusion****

ClasUtil presents a powerful opportunity for educational institutions to adopt smart technologies for efficient campus management. By fusing AI and existing surveillance infrastructure, this system delivers real-time insights and data-driven decision-making tools to optimize space usage and improve academic planning. The deployment is minimally invasive, cost-effective, and scalable for future expansion. We strongly recommend the implementation of ClasUtil as a strategic move towards a smarter, more connected learning environment.

Would you like a version of this formatted into a formal PDF or Word document? I can also add budget projections, timelines, or technical diagrams if needed.