**OBJECTIVE**

The primary objectives of the **VisionixAI** project are:

1. To develop a computer vision-based system that can detect human presence in predefined zones of an indoor environment without relying on external hardware sensors.

2. To design and implement a grid-mapping algorithm that divides a room into logical zones and continuously monitors each zone using a camera feed.

3. To automate control of electrical appliances such as lights and fans by triggering ON/OFF signals based on real-time zone occupancy status with configurable delays.

4. To build a CLI-based tool that enables developers and administrators to interact with, configure, and monitor the vision system easily.

5. To ensure energy efficiency and sustainability in shared spaces like classrooms and offices by reducing unnecessary power consumption through intelligent automation.

6. To maintain system modularity and scalability for future integration with mobile/web applications, hardware controllers, and IoT platforms.

7. To create a cost-effective and non-invasive automation solution that can operate independently of traditional sensors, thus minimizing setup complexity and deployment costs.

ABSTRACT

**VisionixAI** is an innovative project that utilizes the power of Computer Vision and Artificial Intelligence (AI) to automate real-time energy optimization in indoor environments. Designed for smart classrooms, offices, and shared spaces, VisionixAI introduces a grid-based visual intelligence system that detects human presence and controls appliances like lights and fans accordingly. This eliminates the need for traditional sensors or manual intervention, enabling an efficient, contactless automation layer.

At the core of **VisionixAI** lies its zone-mapping algorithm, which divides a room into multiple logical sections and continuously monitors them through a camera feed. Using lightweight computer vision models and motion inference, it determines occupancy in each zone. When a zone remains unoccupied for a configured time, the system signals associated devices to power down, and reactivates them upon re-entry. This behaviour ensures intelligent resource usage, reduced energy wastage, and improved comfort in shared environments.

The project is implemented as a CLI-first tool to maximize portability and developer control. It integrates seamlessly with hardware controllers or APIs to trigger energy-saving actions. **VisionixAI's** modular architecture allows easy extension to mobile, web, or IoT interfaces in the future.

This project addresses the growing demand for sustainable and intelligent automation. It serves as a scalable foundation for broader smart infrastructure systems—eliminating the dependency on external sensors while leveraging vision-based intelligence to reduce operational costs and promote environmental responsibility.

In essence, **VisionixAI** represents a shift towards smarter spatial awareness through pure computer vision. It opens new possibilities for AI-driven automation and resource management, fostering both sustainability and technical innovation.

**VisionixAI — Software Requirements Specification (SRS)**

**1. Introduction**

**1.1 Purpose**

This document defines the software requirements for **VisionixAI**, a smart room automation platform that uses computer vision (YOLOv8) to detect human presence in defined spatial zones and trigger signals to control devices such as lights and fans.

**1.2 Scope**

VisionixAI is a software-only solution for now, using only a webcam and real-time inference to determine presence. It divides rooms into zones and operates via a CLI. The platform is intended for future integration with hardware (IoT devices) and dashboards via an API layer.

**2. Overall Description**

**2.1 Product Perspective**

VisionixAI acts as a middleware between video input and device logic. It runs continuously, accepts a webcam stream, processes it via a YOLOv8 model, maps presence to zones, and emits virtual ON/OFF signals.

**2.2 Product Features**

* Webcam integration (live feed or IP camera)
* Room zoning system (manual/custom grid setup)
* Person detection using YOLOv8 or Mediapipe
* Zone-based presence tracking
* CLI tool to control the stream and emit signals
* Future support for API and dashboard layers

**2.3 User Classes**

* **Admin/Developers**: Set up and run the system
* **Classroom/Office Users**: Indirect users who are detected

**3. Functional Requirements**

* **FR1:** The system shall capture frames from a webcam.
* **FR2:** The system shall detect humans using YOLOv8.
* **FR3:** The system shall divide the room into user-defined zones.
* **FR4:** The system shall map detected people to zones using centroid tracking.
* **FR5:** The system shall track the duration a person remains in a zone.
* **FR6:** The system shall trigger an ON signal if a person is present for >10 sec.
* **FR7:** The system shall trigger an OFF signal if the zone is empty for >10 sec.
* **FR8:** The CLI tool shall allow the user to start/stop streams and view zone status.

**4. Non-Functional Requirements**

* **NFR1:** The system shall process at least 10 FPS on a basic CPU.
* **NFR2:** The model shall detect persons with ≥85% accuracy.
* **NFR3:** The CLI shall be user-friendly with clear status updates.
* **NFR4:** The system shall be scalable to any room layout.

**5. System Design Overview**

* **Input:** Webcam frames
* **Processing:** YOLOv8 → Bounding Boxes → Zone Mapping → Timers
* **Output:** Zone occupancy log, ON/OFF signal via CLI/API

**6. Technologies Used**

* Python
* OpenCV
* YOLOv8 (ultralytics)
* Tkinter/CLI for UI
* Flask/FastAPI (for future API layer)
* MQTT/Socket.IO (for future hardware signals)