Technical Project Report: Room Object Detection and Reporting System

I. Executive Summary

This project delivers a complete pipeline for performing object detection on room images, deduplicating findings on a per-room basis, and generating a quantitative report detailing object presence across multiple rooms. The solution is deployed as a local web application utilizing Streamlit for the user interface and **YOLOv11** for highly accurate inference.

II. Technical Stack

Component	Technology	Role
Model	**YOLOv11**	State-of-the-art object detection engine (`best.pt` model file).
Inference Framework	`ultralytics` (v8.3.0)	Python package for model loading and running detection.
User Interface	Streamlit	Provides a local, interactive web interface for file uploads and result visualization.
Language	Python (3.9+)	Primary development language for all backend logic and UI integration.
Libraries	Pandas, Pillow, NumPy	Used for data structuring, image manipulation, and visualization.

III. System Architecture and Flow

The application follows a modular architecture split into three logical layers:

1. **Frontend (Streamlit):** Handles multi-file image uploads and allows the user to assign or edit the room name (identifier) for each file. It displays controls for `Model Path` and `Confidence Threshold`.
2. **Inference Layer (`src/yolo_inference.py`):** Loads the cached `best.pt` model. It runs inference on the uploaded image (Numpy array format), returning a list of raw detections (bounding box coordinates, class ID, confidence).
3. **Reporting Layer (`src/dedupe.py`, `src/report.py`):**
    - **Deduplication:** The core business logic is executed here. All raw detections below the set confidence threshold are filtered, and the remaining objects are reduced to a **set of unique class names** per room. (Example: 3 detections of 'chair' in one room result in a count of 1 for 'Chair' in that room).
    - **Reporting:** A Pandas DataFrame is constructed where rows represent rooms, columns represent object classes (Capitalized), and cell values are binary (1 if the object is present, 0 if not). A final `Total` row aggregates the presence count across all rooms.

IV. Model Performance Metrics

The production model, `best.pt`, was trained using the YOLOv11 architecture on the custom HomeObjects-3K dataset. The metrics below represent the overall performance averaged across all 12 classes:

Metric	Value
Precision (P)	**0.698**
Recall (R)	**0.740**
mAP50	**0.778**

Training Reference: Full per-class performance details and configuration are available in the training notebook, `Kaggle-roomobjectdetectionmodel-simplyphi.ipynb`.

V. Key Features

- **Per-Room Deduplication:** Handles redundant or overlapping detections within a single image by only recording the *presence* of an object type (count = 1).
- **Cross-Room Totals:** The total count for any object reflects the number of *unique rooms* in which that object was detected.
- **Real-time Visualization:** Annotated images with bounding boxes and capitalized labels are generated in memory and displayed instantly in the Streamlit UI.
- **Data Export:** The final aggregated report is available for download as a CSV file.
- **Robust Class Mapping:** The system accurately maps class IDs (0-11) to the designated object names, including multi-word names like "potted plant" and "photo frame."
- **No Detection Handling:** Rooms with zero detections are clearly marked in the UI and result in an all-zero row in the tabular report.