Project Requirements

Property Data RAG System

Project Objective

Build a Retrieval-Augmented Generation (RAG) system that answers questions about real estate properties using property listings and market data.

Dataset Requirements

- **Property listings dataset** (CSV/JSON format)
- **Minimum 1,000 property records**
- **Required fields:**
  - Address
  - Price
  - Bedrooms
  - Bathrooms
  - Property type
  - Listing date
  - Description
- **Optional fields:**
  - Crime score
  - Flood risk

Core Features

1. Document Ingestion

- Process property data files and create vector embeddings

2. Query Interface

- Accept natural language questions about properties

3. Retrieval System

- Find relevant properties based on user queries

4. Response Generation

- Provide accurate answers with data citations

Sample Test Questions

- "What's the average price of 3-bedroom homes?"
- "Find properties under $400 with 2+ bathrooms"
- "Which area has the most Crime?"

- "Compare prices between studio and 2 bed homes"

Technical Stack

- **Backend:** Python/FastAPI
- **Vector DB:** Pinecone, Weaviate, or ChromaDB
- **Embeddings:** OpenAI, Sentence-Transformers, or similar
- **LLM:** OpenAI GPT, Anthropic Claude, or open-source alternative
- **Frontend:** Simple web interface (Angular, Streamlit, or React)

Deliverables

1. Working RAG application with web interface
2. Source code with documentation
3. Sample dataset (synthetic or public data)
4. Demo video showing key functionality
5. Brief technical report explaining approach

Evaluation Criteria

- **Accuracy:** Correct answers to property queries
- **Code Quality:** Clean, documented, maintainable code
- **Technical Implementation:** Proper RAG pipeline design

Room-wise Unique Object Detection

Project Objective

Build a computer vision system that detects and counts unique objects in rooms, ensuring duplicates in the same room are counted only once.

Dataset Requirements

- **Input:** Images or videos of indoor rooms (living room, bedroom, office)
- **Minimum:** 500 labeled room images (synthetic or public datasets allowed such as OpenImages or COCO Indoor scenes, or custom-annotated)
- **Annotations required:**
  - Room boundaries/identification (at least per-image "room ID")
  - Object bounding boxes with class labels (TV, sofa, chair, table, bed)

Core Features

1. Object Detection

- Use YOLO (v8/v11 or similar) to detect objects within rooms

2. Room Identification

- Tag each image/frame with a room ID (metadata or scene classification/segmentation)

3. Unique Object Counting

- **Within a single room:** If the same object appears multiple times, count it as one
- **Across different rooms:** Count the object separately per room

Example:

- Two TVs in Room A → Count = 1 TV
- One TV in Room B + one TV in Room A → Count = 2 TVs

4. Room-wise Report Generation

- Output a structured report showing each room and its unique objects list

Sample Test Scenarios

Scenario 1:

- **Input:** Photo of living room with 2 sofas, 3 chairs, 2 TVs
- **Output:** Room A → Sofa: 1, Chair: 1, TV: 1

Scenario 2:

- **Input:** Apartment with 2 rooms → Room A (2 TVs), Room B (1 TV)
- **Output:** Room A → TV: 1, Room B → TV: 1 (Total unique TVs = 2)

Scenario 3:

- **Input:** Studio room with bed + chair + duplicate bed annotation error
- **Output:** Room C → Bed: 1, Chair: 1

Technical Stack

- **Backend/Model:** Python, YOLOv8/YOLOv11 (Ultralytics or custom)
- **Room Classification/Segmentation:** Metadata tags or pretrained scene classification model
- **Processing Pipeline:**
  - YOLO for object detection
  - Room-wise grouping logic
  - Deduplication of objects per room
- **Frontend:** Simple web/desktop interface (Streamlit, React, or Flask UI)
- **Output Format:** JSON/CSV + visualization (bounding boxes on images)

Deliverables

1. Working YOLO-based application that processes images and outputs room-wise unique objects
2. Source code with documentation
3. Sample annotated dataset
4. Demo video showing detection and deduplication logic in action
5. Technical report explaining:
   - Object detection approach
   - How room-wise deduplication was implemented

Evaluation Criteria

- **Accuracy:** Correct detection and unique counting per room
- **Robustness:** Handles multiple rooms and duplicate objects correctly
- **Code Quality:** Clean, modular, well-documented code
- **Practicality:** Clear pipeline from image → report
- **Innovation:** Bonus if the applicant uses techniques like scene segmentation, room classification, or embeddings for uniqueness detection

Here's a simple project requirement draft for your LiDAR-based floor plan system:

Project Requirement – LiDAR Floor Plan & Room Dimension System

Objective

Develop a system that uses a LiDAR camera to capture indoor spaces and automatically generate accurate floor plans with room dimensions.

Key Requirements

Functional Requirements

1. **LiDAR Data Capture**
    - Use a LiDAR-enabled camera to scan indoor spaces.
    - Capture 3D point cloud data of walls, doors, and furniture.
2. **Floor Plan Generation**
    - Convert LiDAR scan data into a 2D floor plan.
    - Clearly mark walls, doors, and windows.
    - Allow export of floor plan in standard formats (e.g., PDF, DXF, PNG).
3. **Room Dimension Extraction**
    - Automatically calculate room length, width, and area.
    - Provide accurate wall-to-wall measurements.
    - Support multiple rooms in one scan.
4. **User Interface**
    - Simple interface to start/stop scanning.
    - Display live preview of captured floor plan.
    - Allow user to edit or label rooms.
5. **Output & Storage**
    - Save floor plan with dimensions.
    - Store project data locally or in the cloud.

Non-Functional Requirements

- Accuracy: Room dimensions within ±2 cm.
- Usability: Minimal technical knowledge required.
- Compatibility: Work on mobile/tablet with LiDAR (iPad Pro, iPhone Pro, or LiDAR-enabled device).

Alternative Creative Project Option

Project Objective

For creative thinkers: If you have an innovative idea for applying data science to real estate that goes beyond the two projects above, you're welcome to propose and build your own prototype! We're looking for candidates who can think outside the box and identify novel applications of data science in the real estate domain.

Creative Project Guidelines

- **Innovation Focus:** Your project should demonstrate creative problem-solving and novel use of data science techniques
- **Real Estate Domain:** Must be applicable to real estate industry challenges or opportunities
- **Technical Rigor:** Should showcase solid data science/ML fundamentals
- **Practical Value:** Solve a real problem that industry professionals would find useful