

Project Concept

The AR-Warehouse Management System (AR-WMS) for your small computer store aims to augment traditional warehouse operations by overlaying augmented reality data visualization on real-world objects and shelves. This system enhances inventory management, item lookup, and stock monitoring using AR to:

- Quickly identify product locations
- Verify stock quantities visually
- Speed up picking and restocking processes
- Minimize errors and improve operational efficiency

Core interaction would be through mobile or wearable AR devices (e.g., smartphones, tablets, or AR glasses) that use camera input to recognize products and shelves and overlay information such as product details, inventory status, restock alerts, and real-time updates.

Technology Stack

- **Programming Language:** Python (for backend, computer vision, and integration)
 - **Computer Vision & AR:** OpenCV (image processing, object detection), optionally ARCore/ARKit for advanced AR rendering on mobile
 - **Machine Learning:** TensorFlow/PyTorch for training object detection models to recognize products
 - **Database:** SQLite or PostgreSQL for inventory database
 - **Backend Framework:** FastAPI or Flask (REST API to connect AR front end with backend database)
 - **Frontend (AR Interface):** Unity with Python integration (via external REST calls) or native mobile app (Android/iOS) with ARCore/ARKit
 - **Hardware:** Smartphones or AR glasses with camera capabilities
 - **Additional Tools:** GitHub for version control, Docker for containerization, CI/CD pipelines for deployment
-

Draft Proposal Outline

1. **Project Title:** AR-Warehouse Management System for Small Computer Store
2. **Background & Problem Statement:**

Traditional inventory management relies heavily on manual checks, which are prone to errors and time-consuming. AR technology offers an interactive and efficient solution for real-time inventory visualization and management.

3. Objectives:

- Develop an AR-based system to identify and locate inventory items in the warehouse.
- Integrate object detection using OpenCV and ML to recognize computer store products.
- Provide real-time inventory updates and alerts via the AR interface.

4. Methodology:

- Use OpenCV for camera capture and image processing; train models for product recognition.
- Develop backend API with FastAPI and database to maintain inventory data.
- Build AR frontend application for real-time overlay and interaction.

5. Expected Outcomes:

- Reduced search and picking time.
- Improved accuracy in stock counts and reordering.
- Enhanced warehouse staff productivity.

6. Timeline & Milestones: (see project plan below)

Project Plan (High-Level)

Phase	Duration	Key Activities	Deliverables
Requirements & Research	2 weeks	Gather store requirements, research AR tech	Detailed requirement doc
Design	2 weeks	System architecture, technology selection	Design docs, data models
Prototype Development	4 weeks	Develop detection model, basic AR overlay	Prototype MVP
Backend & API Setup	3 weeks	Develop inventory database, REST API	Functional backend system
Frontend AR App	4 weeks	AR interface development, integration	Working AR app
Testing & Refinement	3 weeks	System testing, bug fixing, usability	Tested and refined system
Deployment & Training	2 weeks	Deploy system, staff training	Deployed system, training materials