Smart Product Labeling and Traceability System

# 1. Raspberry Pi 5 (Main Control Module)

**What it is:**

Raspberry Pi 5 is the fifth-generation single-board computer in the Raspberry Pi series. It comes equipped with a quad-core ARM Cortex-A76 CPU, 8GB RAM (optional), dual HDMI ports, and an upgraded GPU for better multimedia performance. The GPIO pins allow for connecting a wide variety of sensors and actuators, making it ideal for embedded and industrial control projects.

**Why we used it:**

It acted as the central controller of our entire system. The Raspberry Pi handled data collection from sensors, image capturing, running the object detection and OCR models, controlling actuators for the conveyor system, and sending data to the Firebase database. It served as a compact yet powerful embedded computer that could run Python scripts, manage real-time logic, and even handle HTTP requests for web integration.

**Advantages:**

- Affordable and powerful edge device for embedded AI  
- Linux-based OS allows full software stack development  
- GPIO support for flexible hardware interfacing  
- Capable of running computer vision models locally  
- Easy integration with cloud services like Firebase

# 2. EasyOCR (Label Text Detection)

**What it is:**

EasyOCR is an open-source Python-based OCR tool that supports over 80 languages. It utilizes a combination of convolutional neural networks (CNNs) and attention-based LSTM decoders to read printed or handwritten text from images. It's designed to work out-of-the-box on noisy, rotated, or partial text in real-world scenarios.

**Why we used it:**

EasyOCR was used to verify the text content on printed labels. Once a PCB passed object detection, a high-resolution image of the label area was captured and processed using EasyOCR. This helped us ensure that the serial number, manufacturing date, and other critical details were clearly printed and legible before the product continued down the line.

**Advantages:**

- No need to train a custom OCR model for standard text  
- Easily integrates with OpenCV and Python scripts  
- Robust to skewed angles, lighting variations, and blur  
- Lightweight enough to run on the Raspberry Pi 5

# 3. Custom Object Detection Model (Python + CNN)

**What it is:**

We developed a lightweight convolutional neural network (CNN) model in Python to detect PCBs. This model was trained using a dataset of PCB images in various orientations and lighting conditions. It served as the first validation step to ensure the object entering the labeling station was indeed a PCB.

**Why we used it:**

Without reliable object recognition, there's a risk of labeling incorrect or misaligned items. This model helped us automate the identification process, ensuring only verified PCBs were processed further. The use of AI instead of hardcoded pixel thresholds improved robustness and reduced false triggers.

**Advantages:**

- Trained on a custom dataset for high accuracy  
- Reduced dependency on physical alignment constraints  
- Enabled dynamic rejection of non-compliant objects  
- Enhances traceability by ensuring proper object identification

# 4. Firebase (Cloud-Based Database)

**What it is:**

Firebase is a platform developed by Google that offers backend services such as Realtime Database, Authentication, and Hosting. It provides REST APIs for interacting with structured JSON data over HTTP.

**Why we used it:**

Our Firebase database stored inspection results for every unit that passed through the system. It included fields like Device ID, Batch ID, RoHS status, timestamps, and rejection reasons (if any). Firebase Authentication also allowed us to differentiate between customers (who can only verify data) and factory workers (who can modify logs and view system statistics).

**Advantages:**

- Real-time updates and sync between hardware and web interface  
- Highly scalable and secure backend  
- Easy to implement user authentication  
- No infrastructure setup or server maintenance required

# 5. Web Dashboard (Verification & Admin Panel)

**What it is:**

A responsive web application with two major portals:  
1. Customer Verification Page: Users enter a serial number (e.g., BOX\_1001) to check product authenticity.  
2. Admin Panel: Authenticated employees can log in, modify entries, view rejected logs, and monitor production.

**Why we used it:**

The dashboard made our system transparent and user-friendly. Instead of accessing databases manually, employees and customers interacted with a clean UI. This also reduced the chances of human error in log management.

**Advantages:**

- Bridges the gap between backend logs and real-world users  
- Accessible from any browser/device  
- Adds value to product authenticity by allowing customers to self-verify  
- Supports future extension to analytics or dashboard monitoring

# 6. Roller-Based Labeling System

**What it is:**

A motorized roller system that applied printed labels onto the PCBs. The label was positioned manually or via automated feeder in sync with product position.

**Why we used it:**

We wanted to simulate a real-world labeling environment. The roller system allowed consistent pressure to be applied during labeling, reducing human intervention and inconsistencies.

**Advantages:**

- Physical simulation of industrial automation  
- Precise and reliable label placement  
- Easily extendable to sticker rollers or inkjet markers  
- Minimizes manual errors

# 7. Sensors and Actuators (Automation Support)

**What it is:**

IR proximity sensors were used to detect the presence and alignment of products. Actuators (motors) controlled the movement of the conveyor belt and the label roller. These were controlled through GPIO pins of the Raspberry Pi using Python logic.

**Why we used it:**

To automate the stopping and starting of the conveyor belt based on product detection. This improved timing accuracy, ensured label alignment, and allowed rejection of faulty units.

**Advantages:**

- Accurate detection of PCB position  
- Smooth coordination between detection, labeling, and data logging  
- Real-time system response and control  
- Reduces manual labor and increases throughput