# PROJECT SENTINEL

## Retail Analytics & Optimization Platform Challenge

Project Sentinel

## THE SCENARIO

Many retail companies are facing an existential crisis in today’s competitive market. They are losing an estimated significant amount annually due to a combination of inventory shrinkage, operational inefficiencies, and customer abandonment. Their recent investment in self-checkout technology, intended to reduce costs, has instead introduced new complications, including a 34% increase in theft incidents, customer frustration, and challenges in staffing allocation.

The four critical challenges facing the modern retail industry are:

1. **Inventory shrinkage** (theft, misplacement, and record inaccuracies)
2. **Self-checkout security and efficiency issues** (scan avoidance, Ticket Switching, unexpected system crashes and scanning errors)
3. **Resource allocation inefficiencies** (staffing, checkout lanes, stock replenishment)
4. **Poor customer experience** (long wait times, difficult self-checkout processes)

## THE CHALLENGE

In this 4-hour challenge, your team of 4 must design and build a prototype for an integrated **Retail Intelligence System**. This system must:

* Process and analyze multi-modal data streams from various in-store sensors
* Monitor and optimize self-checkout operations in real-time
* Detect inventory anomalies, suspicious activities, and operational inefficiencies
* Optimize resource allocation in real-time
* Deliver actionable insights through an intuitive dashboard

Your solution should demonstrate how it would operate in a live retail environment, making it possible for store managers to make data-driven decisions instantly.

## DATA SOURCES

Your system must integrate these heterogeneous data streams:

1. **RFID Tag Data** - Real-time product location and movement near the self-checkout counter
   * Format: JSON with tag IDs, location coordinates, and timestamps
   * Update frequency: Continuous, approximately 120 readings per second
2. **Computer Vision Analytics** - Pre-processed data from store cameras around Counters
   * Format: JSON with customer counts, dwell times, and demographic estimates
   * Update frequency: 5-second intervals
3. **Point-of-Sale Transactions** - Sales data from self-checkout systems
   * Format: JSON database records with transaction details
   * Update frequency: Real-time as transactions occur
4. **Products Recognition Analytics** – Vision based products identification of self-checkout systems
   * Format: JSON with Predicted product details
   * Update frequency: Real-time as transactions occur
5. **Mobile Scanner Activity** - Employee inventory activities
   * Format: XML messages from handheld devices
   * Update frequency: Irregular, event-driven
6. **Store Layout & Fixture Information** - Physical store configuration
   * Format: JSON combination defining zones, shelves, displays, self-checkout areas, and walkways
   * Update frequency: Static, updated only when store layout changes
7. **Current Inventory Data**- All Nassery products data in current inventory
   * Format: JSON
   * Update frequency: 1-hour Interval

## TECHNICAL REQUIREMENTS

### 1. System Architecture Design

Create a detailed system architecture diagram showing:

* Data ingestion and processing pipeline
* Storage solutions (real-time and historical)
* Processing components and their relationships
* Self-checkout monitoring and intervention system
* Dashboard and alert systems

Your design must address latency requirements, data volume handling, and system resilience.

### 2. Data Fusion Engine

Develop a data integration module that:

* Synchronizes timestamped data across different sources
* Correlates self-checkout activities with RFID and camera data
* Resolves spatial coordinates between different reference systems
* Handles missing, corrupt, or delayed data
* Creates a unified data model for cross-source analysis

### 3. Self-Checkout Optimization System

Create algorithms that:

* Detect scan avoidance and barcode switching
* Unexpected system crashes and scanning errors
* Identify weight discrepancy patterns that indicate theft attempts

### 4. Anomaly Detection System (Required)

Implement detection algorithms for:

* Inventory discrepancies between expected and actual counts
* Unusual product movement patterns indicating potential theft
* Unexpected customer behavior patterns around self-checkout areas
* Supply chain disruptions and imminent stockouts

### 5. Resource Optimization Engine (Required)

Create an algorithm that:

* Predicts checkout staffing needs 15 minutes in advance
* Recommends optimal staff allocation between traditional and self-checkout supervision
* Prioritizes restocking activities based on sales velocity and inventory levels
* Balances customer experience metrics against operational costs
* Dynamically adjusts when to open/close self-checkout stations based on store traffic

### 6. Visualization Dashboard (Required)

Design and implement an intuitive interface that:

* Provides real-time store status overview
* Highlights anomalies and alerts with appropriate priority levels
* Visualizes customer flow, inventory status, and resource allocation
* Tracks self-checkout efficiency and security metrics
* Enables drill-down into specific incidents or metrics

## CHALLENGE SCENARIOS

Your system will be tested against the following scenarios using simulated data:

### Scenario 1: “The Vanishing Inventory”

Several high-value electronic items show significant count discrepancies between system records and physical RFID readings, concentrated in a specific store area near self-checkout stations.

### Scenario 2: “Self-Checkout Surge”

A promotional event creates unexpected customer traffic at self-checkout stations, overwhelming the single staff member assigned to monitor them and causing both legitimate intervention requests and potential theft opportunities.

### Scenario 3: “Organized Retail Crime”

A coordinated theft attempt involving manipulation of product tags, weight sensor deception at self-checkout, and unusual movement patterns near emergency exits.

### Scenario 4: “The Frustrated Abandonment”

Multiple customers are abandoning their shopping carts at self-checkout stations due to technical difficulties, creating inventory displacement and customer experience issues.

### Scenario 5: “Supply Chain Disruption”

Key products face imminent stockout due to delayed shipments, requiring prioritized inventory redistribution across the sales floor and special handling at self-checkout stations.

## PROVIDED RESOURCES

### Data files

* 1. Inventory\_data.csv
  2. Inventory\_data.csv
  3. Inventory\_data.csv
  4. Inventory\_data.csv
  5. Inventory\_data.csv
  6. Inventory\_data.csv

## DELIVERABLES

1. **System Architecture Documentation** (PDF)
   * Comprehensive diagram showing all system components
   * Explanation of design choices and trade-offs
   * Data flow and processing pipeline descriptions
   * Self-checkout monitoring subsystem details
2. **Working Prototype** (Code Repository)
   * Data integration module
   * Self-checkout optimization algorithms
   * Anomaly detection algorithms
   * Resource optimization engine
   * Visualization dashboard
3. **5-Minute Presentation**
   * Problem statement and solution approach
   * Architecture highlights and technical innovations
   * Live demonstration of system capabilities
   * Potential business impact for NexGen Retail

## EVALUATION CRITERIA

| Criteria | Weight | Description |
| --- | --- | --- |
| **System Design Quality** | 25% | Architecture elegance, scalability, resilience, and appropriate technology choices |
| **Technical Implementation** | 20% | Code quality, functionality, performance, and handling of edge cases |
| **Self-Checkout Optimization** | 20% | Effectiveness of security measures, customer experience improvements, and operational efficiency |
| **Real-time Analytics Capabilities** | 15% | Effectiveness of anomaly detection and resource optimization algorithms |
| **User Experience & Visualization** | 10% | Dashboard usability, information clarity, and actionability of insights |
| **Innovation & Business Impact** | 10% | Creativity, novel approaches, and potential ROI for NexGen Retail |

## ANTI-AI DESIGN ELEMENTS

This challenge has been specifically designed to test true engineering skills that go beyond what generative AI tools can provide:

1. **Complex System Design** - Requires holistic understanding of multiple technologies and their integration
2. **Spatial-Temporal Data Relationships** - Demands reasoning about physical store dynamics
3. **Custom Visualization Development** - Necessitates creative dashboard design tailored to specific business needs
4. **Self-Checkout Security Logic** - Requires nuanced understanding of physical retail operations and human behavior
5. **Multi-faceted Algorithm Design** - Involves complex decision-making across multiple variables
6. **Cross-domain Problem Solving** - Combines retail operations knowledge with technical implementation

## TECHNICAL SPECIFICATIONS

### Development Resources Provided

* Sample data files for each data source
* API documentation for simulation environment
* Testing harness for scenario evaluation
* Base visualization libraries and templates
* Virtual store environment for testing
* Self-checkout system API documentation

### Technical Constraints

* Solution must process data with sub-500ms latency
* Self-checkout security alerts must generate within 800ms of suspicious activity
* Dashboard must update within 1 second of events
* System must handle 1000+ RFID readings per second
* All processing must occur on provided development environment
* No pre-trained ML models may be used (teams must develop their own algorithms)

## SCHEDULE

| Time | Activity |
| --- | --- |
| **0:00 - 0:15** | Challenge briefing and Q&A |
| **0:15 - 1:00** | Architecture design and planning |
| **1:00 - 3:00** | Implementation and development |
| **3:00 - 3:30** | Testing against provided scenarios |
| **3:30 - 3:45** | Preparation of presentation |
| **3:45 - 4:00** | Final submissions |
| **Post-Challenge** | Judging and awards ceremony |

## TIPS FOR SUCCESS

* Focus on a minimum viable product that addresses all required components
* Prioritize system architecture and data integration early
* Create simple algorithms that work over complex ones that might fail
* Pay special attention to the self-checkout optimization component
* Design your dashboard for immediate comprehension by non-technical users
* Consider the business impact of your technical decisions

*This challenge simulates real-world retail technology problems and is designed to showcase your team’s ability to design, implement, and communicate complex technical solutions under time constraints.*