# Inventory Management System using Azure DevOps

## Introduction

The Inventory Management System project aims to build an end-to-end automated data processing and reporting solution using Azure DevOps. This system integrates SQL, MongoDB, Python, PySpark, and Azure Databricks to track and analyze inventory levels, store performance, and automate stock alerts. The solution is managed, versioned, and automated through Azure DevOps boards and pipelines.

## Objective

The main objective of this project is to automate the entire data lifecycle of inventory management from data collection and transformation to analytics, storage, and pipeline automation ensuring accurate real-time insights and reduced manual intervention.

## Tools and Technologies Used

• Azure DevOps – for project management, task tracking, and CI/CD automation  
• MySQL – for structured data storage (products, sales, stores)  
• MongoDB – for unstructured and flexible inventory data storage  
• Python (Pandas, NumPy) – for data cleaning, transformation, and calculation  
• PySpark – for distributed analytics and store-level insights  
• Azure Databricks – for ETL pipeline development and analytics

## Implementation Overview

The project was executed in multiple stages, covering database setup, data processing, ETL pipeline development, and DevOps automation. Each stage was managed using Azure DevOps work items categorized as Epics, Features, User Stories, and Tasks.

### 1. Database Setup (SQL + MongoDB)

In this phase, the database architecture for inventory management was designed. MySQL was used to create relational tables for products, stores, employees, and sales data. MongoDB was integrated for flexible data storage. CRUD operations and stored procedures were tested for consistency and data integrity.

### 2. Data Processing using Python

Python scripts using Pandas and NumPy were developed to clean raw sales and inventory data. Missing values were handled, datatypes were standardized, and duplicate records were removed. The processed dataset was used to compute revenue, profit, and discount percentages and was exported as a cleaned CSV file.

### 3. PySpark Analytics

The cleaned dataset was loaded into PySpark for large-scale analytics. Sales data was grouped by store, product, and month to identify performance trends. Underperforming products were filtered out, and summarized data was exported for visualization and further reporting.

### 4. ETL Pipeline in Azure Databricks

An automated ETL pipeline was developed in Azure Databricks to integrate cleaned data from MySQL and MongoDB. The ETL process included extraction, transformation, and loading of data into a centralized analytics table. Automated jobs were configured to refresh the data periodically, ensuring up-to-date reports.

### 5. Pipeline Automation using Azure DevOps

In the final stage, Azure DevOps was used to automate the daily stock check process. A YAML-based pipeline was created to trigger Python scripts automatically, generate reorder lists, and publish reports. All tasks were managed in Azure Boards, ensuring complete visibility of progress and collaboration among team members.

## Work Items in Azure DevOps

The project work items were created and categorized under a single Epic 'Inventory Management System'. Each feature represented a major module of the system, with corresponding user stories and detailed tasks. All work items were assigned to the user and maintained under the HexaDevOps project area path.

The following categories were used in Azure DevOps:  
• Epic – Inventory Management System  
• Features – Database Setup, Data Processing, PySpark Analytics, ETL Pipeline, DevOps Automation  
• User Stories – Defined under each feature to describe functionality  
• Tasks – Actionable steps linked to each user story

## Screenshots from Azure DevOps

Below screenshots represent the Azure DevOps board and work items created for the Inventory Management System project.

## 

*Figure 1: This screenshot represents the* ***Azure DevOps Board – Work Items View.***

## 

*Figure 1: This screenshot represents the* ***Azure DevOps Work Items for Pipeline Automation***

## Outcome

The Inventory Management System project successfully automated the entire inventory tracking workflow. Through Azure DevOps, daily processes such as stock verification, data updates, and report generation were made fully automated. The system now provides real-time insights into sales, profit margins, and low-stock alerts with minimal manual intervention.

## Conclusion

This project demonstrates the integration of cloud-based data engineering tools and DevOps automation for an end-to-end business solution. Using Azure DevOps enhanced collaboration, ensured continuous integration, and simplified deployment workflows. The Inventory Management System serves as a scalable and reliable framework for retail and supply chain automation.