**📘 Project Documentation: Intune-Like MDM Simulation**

**📌 1. Project Title**

**Intune-Like MDM Simulation Using Python and Power BI**

**🧠 2. Objective**

To simulate a lightweight Mobile Device Management (MDM) system similar to Microsoft Intune. The project enforces basic endpoint security policies, logs device compliance data, and visualizes system-wide compliance using Power BI dashboards.

**🎯 3. Purpose**

* Demonstrate an understanding of endpoint compliance monitoring.
* Showcase skills relevant to system administration, SCCM/Intune support, and compliance tracking.
* Provide hands-on experience with simulated policy enforcement using Python.

**🧰 4. Technologies Used**

| **Area** | **Tools & Languages** |
| --- | --- |
| Programming | Python 3.x |
| Database | SQLite |
| Data Visualization | Power BI Desktop |
| Data Handling | Pandas, CSV |
| OS Target | Simulated Windows 10/11 endpoints |

**📦 5. Project Modules**

**a) device\_check.py**

* Simulates endpoint data (device ID, OS, username, firewall status, etc.).

**b) mdm\_policy.py**

* Defines policy rules: minimum password length, firewall ON, encryption ON.
* Returns compliance results based on simulated device info.

**c) compliance\_logger.py**

* Logs device status (compliant/non-compliant) into a SQLite database (database.db).

**d) main.py**

* Integrates the modules and simulates multiple endpoint compliance checks in one run.

**e) generate\_csv.py**

* Exports database logs to a CSV file (compliance\_data.csv) for Power BI reporting.

**f) dashboard.pbix**

* A Power BI dashboard (user-created) that visualizes compliance trends.

**📂 6. Folder Structure**

mdm\_simulation/

├── device\_check.py

├── mdm\_policy.py

├── compliance\_logger.py

├── main.py

├── generate\_csv.py

├── database.db

├── compliance\_data.csv

├── dashboard.pbix

└── README.txt

**🛠️ 7. How It Works**

1. **Device Simulation**: Generates 10 random devices (ID, OS, firewall status, etc.)
2. **Policy Check**: Evaluates each against rules like:
   * Password ≥ 8 characters
   * Encryption enabled
   * Firewall enabled
3. **Logging**: Stores each result in SQLite with timestamp and compliance status.
4. **Data Export**: Converts logs into a CSV for analysis.
5. **Visualization**: Power BI presents:
   * Pie chart (compliance %)
   * Trend line (compliance over time)
   * Table of devices and status

**📊 8. Power BI Dashboard Highlights**

* Pie chart: Compliant vs Non-Compliant Devices
* Line chart: Compliance Trends Over Time
* Table: Device ID, User, OS, Compliance Result

**✅ 9. Key Functional Features**

| **Feature** | **Description** |
| --- | --- |
| Simulated MDM Policies | Python code mimics enterprise policy enforcement |
| Device Compliance Checking | Firewall, Encryption, Password rules evaluated |
| Centralized Logging (SQLite) | Results logged with timestamps and metadata |
| CSV Export | Structured data for analysis in BI tools |
| Power BI Dashboard | Real-time visual analytics for compliance monitoring |

🎯 **10. Relevance to Oracle Systems Analyst Role**

| **Oracle JD Requirement** | **This Project Demonstrates** |
| --- | --- |
| SCCM/Intune familiarity | Simulated Intune-like policy enforcement |
| MDM/MAM management | Policy checking and endpoint status evaluation |
| Compliance monitoring | Logging + Power BI dashboards |
| Troubleshooting & Reporting | Full cycle: Detection → Evaluation → Logging → Reporting |
| Windows endpoint support | Targeting Windows 10/11 (simulated) |