Phase 8: Data Management & Deployment (CORRECTED)

This phase focused on defining a professional, auditable strategy for metadata migration (deployment) using modern tools and ensuring system data can be reliably backed up.

Concept	Project Goal
Data Management Strategy	Establish reliable protocols for mass data import (historical records) and automated data backup for institutional compliance.
Deployment Strategy	Utilize a source-driven model with **VS Code/SFDX** for all project metadata (Objects, Fields, Flows, Apex) to ensure seamless and auditable migration from Sandbox to Production.

8.1 Data Management

8.1.1 Data Import (Initial Migration)

- **Tool: Data Loader** (Desktop application) was selected over the Data Import Wizard.
- **Purpose:** The initial import involved migrating historical records from legacy spreadsheets (e.g., campus security registers) into the new **Lost Item** and **Found Item** custom objects. Data Loader was chosen due to the high volume of records and the necessity for complex field mapping into the new custom object structure.

8.1.2 Data Integrity & Validation

- **Method:** Data integrity is primarily enforced by the **Validation Rules** (if any were created in Phase 4) and the strict **Custom Field Requirements** (e.g., Category Picklist, required fields) implemented in Phase 3.
- **Note: Duplicate Rules** were explicitly **excluded** from the MVP scope to focus resources on the core automation and matching logic.

8.1.3 Data Export & Backup

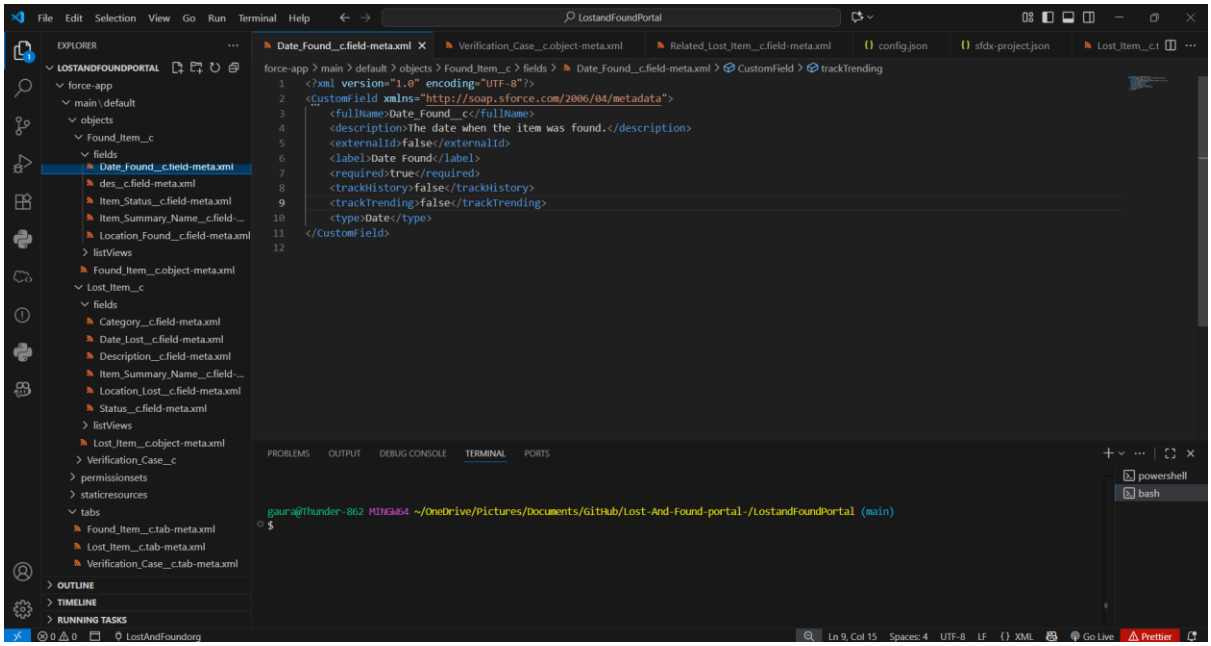
- **Strategy:** The Salesforce **Data Export Service** (Weekly Export) was configured to automatically back up all critical custom object data (Lost Item, Found Item, Case) to the institution's secure external storage location.
- **Purpose:** Ensures compliance with institutional data retention policies and provides a full backup for data recovery.

8.2 Deployment Strategy & Metadata Management

8.2.1 Source-Driven Development (VS Code & SFDX)

All declarative (Flows, Layouts) and programmatic (Apex) metadata was managed using the modern source control framework.

- **Tools: VS Code** with the Salesforce Extension Pack and the **Salesforce Command Line Interface (CLI/SFDX)**.
- **Workflow:** All work was done in a **Development Org**, pulled into a local SFDX project directory, and pushed/deployed using CLI commands (e.g., sf project deploy start).
- **Metadata Components:** This includes the two custom objects, the two Record-Triggered Flows, the QueueTools Apex class, and all relevant security (Profiles/FLS).



8.2.2 Sandbox & Lifecycle Management

- **Sandbox Usage:** The project acknowledges the standard lifecycle: Development → **Sandbox** (for UAT/testing) → Production.
- **Change Sets:** While SFDX was primary, **Change Sets** remain the standard tool for moving security settings, profiles, and smaller metadata components from the UAT Sandbox to the Production environment.