# CAMPUS EVENT MANAGEMENT SYSTEM

NAME: SHAIK RAQUEEBUL ISLAM

## Phase 8: Data Management & Deployment

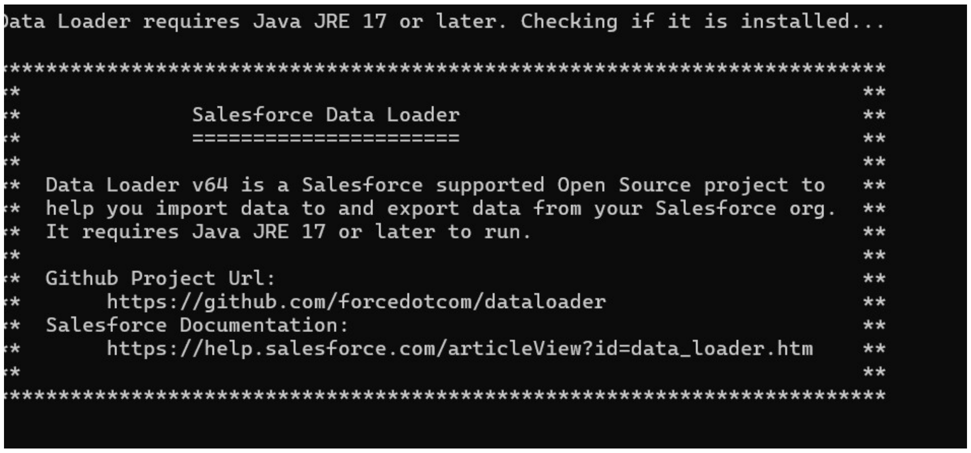
### Objective

To effectively manage bulk data operations for custom objects and deploy the Campus Event Management System components from a development environment to another Salesforce org. This ensures smooth testing, demonstration, and eventual release of the system.

---

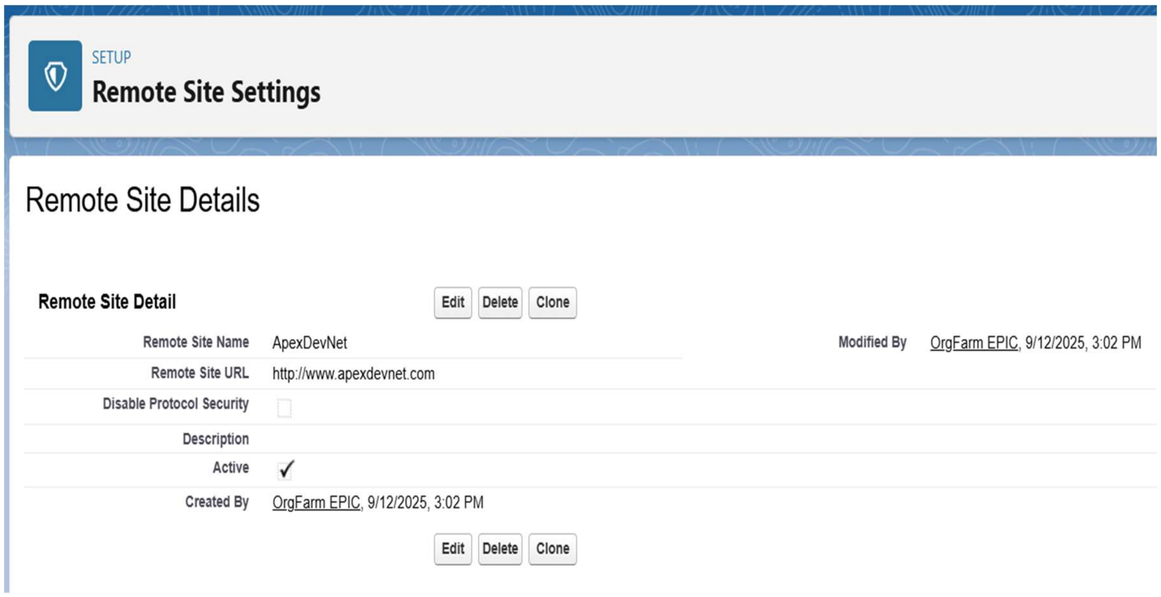
### Data Management Strategy

- Populate custom objects (**Event_Details__c**, **Participant__c**, **Feedback__c**) with realistic sample data.
- Use **Salesforce Data Loader** for high-volume data operations, including inserts, updates, and deletions.
- Prepare the system for final demos and testing by creating accurate and relationship-aware data records.



```
ata Loader requires Java JRE 17 or later. Checking if it is installed...

************************************************************************
*                                                                     **
*             Salesforce Data Loader                                  **
*             ======================                                  **
*                                                                     **
*  Data Loader v64 is a Salesforce supported Open Source project to   **
*  help you import data to and export data from your Salesforce org.  **
*  It requires Java JRE 17 or later to run.                           **
*                                                                     **
*  Github Project Url:                                                **
*       https://github.com/forcedotcom/dataloader                     **
*  Salesforce Documentation:                                          **
*       https://help.salesforce.com/articleView?id=data_loader.htm    **
*                                                                     **
************************************************************************
```


SETUP
Remote Site Settings

Remote Site Details

Remote Site Detail

| | | | |
|---|---|---|---|
| Remote Site Name | ApexDevNet | Modified By | OrgFarm EPIC, 9/12/2025, 3:02 PM |
| Remote Site URL | http://www.apexdevnet.com | | |
| Disable Protocol Security | ☐ | | |
| Description | | | |
| Active | ✓ | | |
| Created By | OrgFarm EPIC, 9/12/2025, 3:02 PM | | |

---

**Data Preparation**

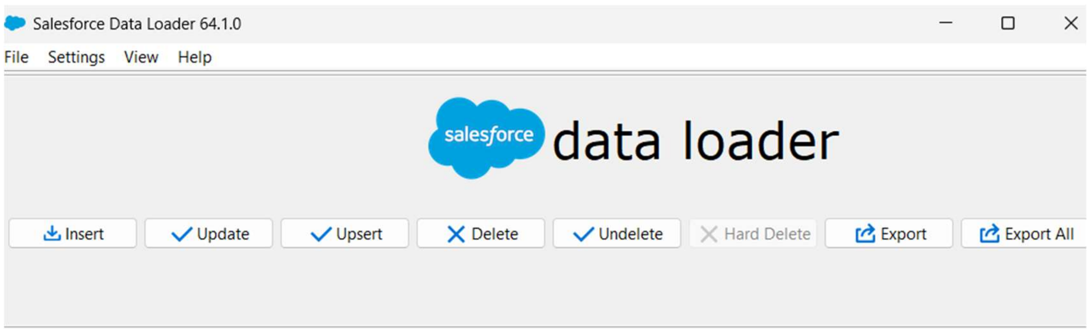
1. **File Format:**
   - Prepare CSV files using Excel or Google Sheets, with columns matching the Salesforce **API field names**.
2. **Event Data (events.csv):**
   - Columns: Event_Name__c, Event_Date__c, Location__c, Organizer__c
   - Ensure date format matches Salesforce standards (YYYY-MM-DD).
3. **Participant Data (participants.csv):**
   - Columns: Name, Participant_Email__c, Event__c
   - For lookup relationships (Event__c), use the **18-digit Salesforce record ID** of the corresponding Event.
4. **Feedback Data (feedback.csv):**
   - Columns: Participant__c, Event__c, Feedback_Rating__c, Comments__c
   - Use correct IDs for Participant and Event lookup fields.

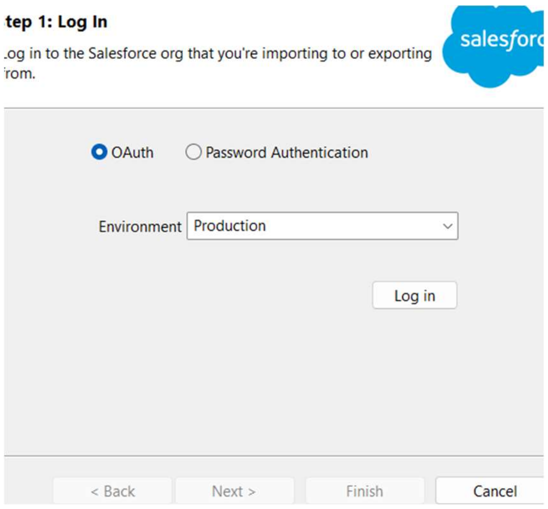
---

**Data Loading Process**

1. **Tool:** Salesforce Data Loader (Desktop application).
2. **Authentication:** Connect using **OAuth login** for your Developer Org.

3. **Action:** Use **Insert** operation for new records (Events, Participants, Feedback).
4. **Mapping:**
    - Map CSV columns to object fields in Data Loader wizard.
    - Use **Auto-Match** to reduce manual mapping effort.
5. **Verification:**
    - Check loaded records in Salesforce UI:
        - Events → "Events" tab
        - Participants → "Participants" tab
        - Feedback → "Feedback" tab

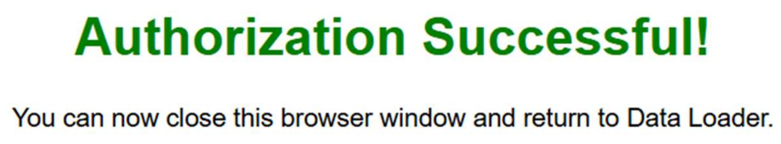



tep 1: Log In

.og in to the Salesforce org that you're importing to or exporting
'rom.

[✓ OAuth] Password Authentication

Environment: Production


Authorization Successful!

You can now close this browser window and return to Data Loader.

## Deployment Strategy

- Use **Salesforce DX (SFDX)** and **VS Code** for source-driven deployment.
- Treat all components (Apex, LWCs, layouts, pages) as **files in a local project folder**.
- Deploy systematically to a new org or sandbox for testing.

**Retrieve components from org:**

- sfdx force:source:retrieve -x manifest/package.xml

**Deploy components to target org:**

- sfdx force:source:deploy -x manifest/package.xml

**Push local DX source to scratch org:**

- sfdx force:source:push

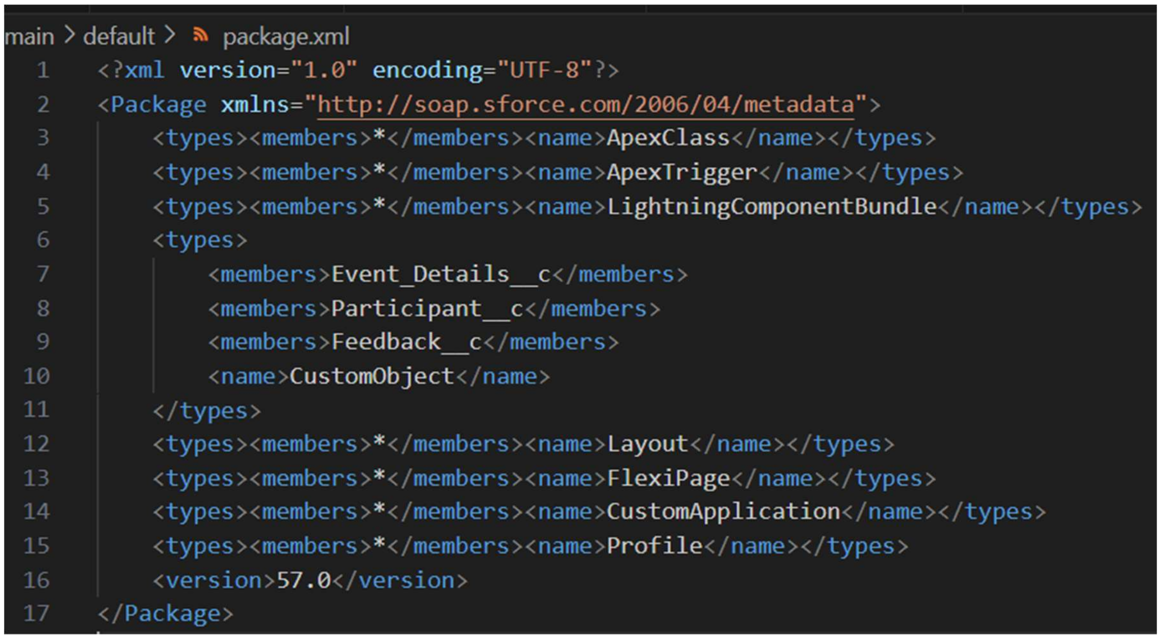
**Pull changes from scratch org to local:**

- sfdx force:source:pull

---

## Metadata Management

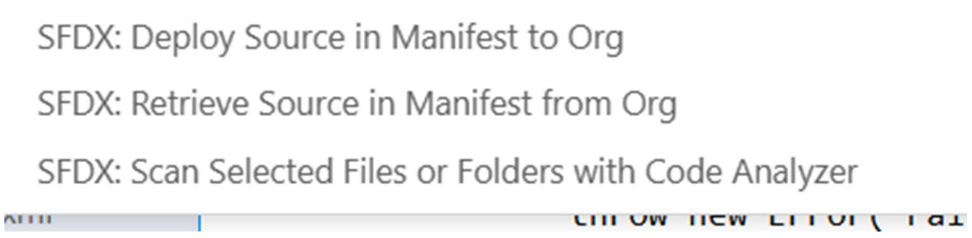
1. **package.xml Manifest:**
    - Create a manifest in manifest/package.xml listing all components:
        - ApexClass, CustomObject, LightningComponentBundle, Layout, FlexiPage, CustomApplication
    - Ensure accuracy to avoid deployment errors.



main > default > package.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types><members>*</members><name>ApexClass</name></types>
    <types><members>*</members><name>ApexTrigger</name></types>
    <types><members>*</members><name>LightningComponentBundle</name></types>
    <types>
        <members>Event_Details__c</members>
        <members>Participant__c</members>
        <members>Feedback__c</members>
        <name>CustomObject</name>
    </types>
    <types><members>*</members><name>Layout</name></types>
    <types><members>*</members><name>FlexiPage</name></types>
    <types><members>*</members><name>CustomApplication</name></types>
    <types><members>*</members><name>Profile</name></types>
    <version>57.0</version>
</Package>
```

2. **Troubleshooting:**
    - Remove non-existent components from manifest to prevent deployment failures.
    - Example: If a CustomTab or object no longer exists, delete its entry from package.xml.


SFDX: Deploy Source in Manifest to Org
SFDX: Retrieve Source in Manifest from Org
SFDX: Scan Selected Files or Folders with Code Analyzer