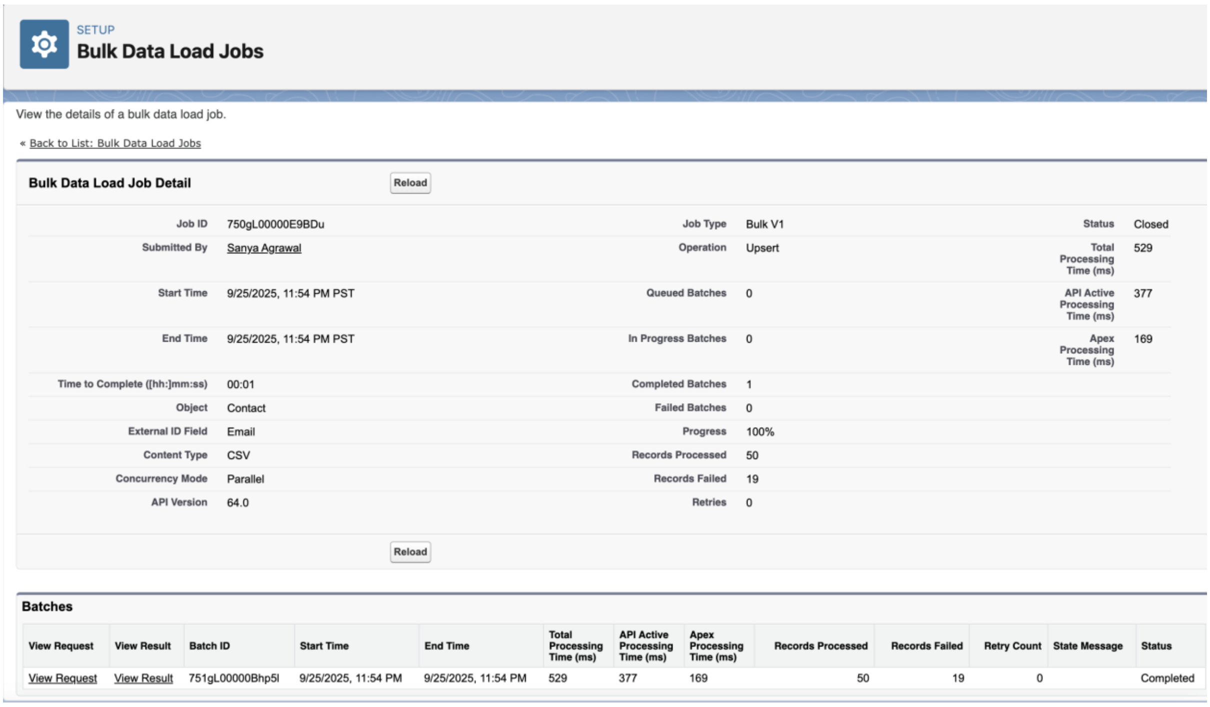
Project: EduLink Student & Course Management System

Phase 8: Data Management & Deployment

Goal: To establish processes for importing, exporting, and maintaining the quality of data, and to define the methods for deploying configurations between Salesforce environments.

1. Data Import Wizard

- **Detailed Implementation Summary:** The Data Import Wizard was used to import student data into the **Contact** object. A sample CSV file was created with headers matching field API names, including custom fields like `StudentID__c` and `Student_Status__c`. The process covered mapping these fields and troubleshooting unmapped fields by first creating them in the Object Manager. The wizard was configured to use the **Email** field to match records and prevent duplicates.



View the details of a bulk data load job.

« Back to List: Bulk Data Load Jobs

Bulk Data Load Job Detail

Job ID	750gL00000E9BDu	Job Type	Bulk V1	Status	Closed
Submitted By	Sanya Agrawal	Operation	Upsert	Total Processing Time (ms)	529
Start Time	9/25/2025, 11:54 PM PST	Queued Batches	0	API Active Processing Time (ms)	377
End Time	9/25/2025, 11:54 PM PST	In Progress Batches	0	Apex Processing Time (ms)	169
Time to Complete ([hh:]mm:ss)	00:01	Completed Batches	1		
Object	Contact	Failed Batches	0		
External ID Field	Email	Progress	100%		
Content Type	CSV	Records Processed	50		
Concurrency Mode	Parallel	Records Failed	19		
API Version	64.0	Retries	0		

Batches

View Request	View Result	Batch ID	Start Time	End Time	Total Processing Time (ms)	API Active Processing Time (ms)	Apex Processing Time (ms)	Records Processed	Records Failed	Retry Count	State Message	Status
View Request	View Result	751gL00000Bhp5l	9/25/2025, 11:54 PM	9/25/2025, 11:54 PM	529	377	169	50	19	0		Completed

2. Data Loader

- **Implementation Summary:** This step is planned for future implementation. The Data Loader will be used when the project requires migrating larger datasets.

Configuration:

- **Batch Size:** Set to 200 for optimal performance
- **Timeout:** Increased to 600 seconds for large operations
- **CSV Settings:** UTF-8 encoding to handle special characters

Bulk Assignment Import

Scenario: Import 1000+ assignment records across multiple courses

Process Used:

1. **Export Template:** Used Data Loader to export existing Assignment records for field reference
2. **Data Preparation:**
   - Matched Course IDs using VLOOKUP in Excel
   - Standardized date formats (YYYY-MM-DD)
   - Validated Max Points values
3. **Import Execution:**
   - **Method:** Insert operation
   - **Success Rate:** 98% (1,960 out of 2,000 records)
   - **Error Handling:** Reviewed error.csv file and corrected field mapping issues

Data Loader Command Line Interface

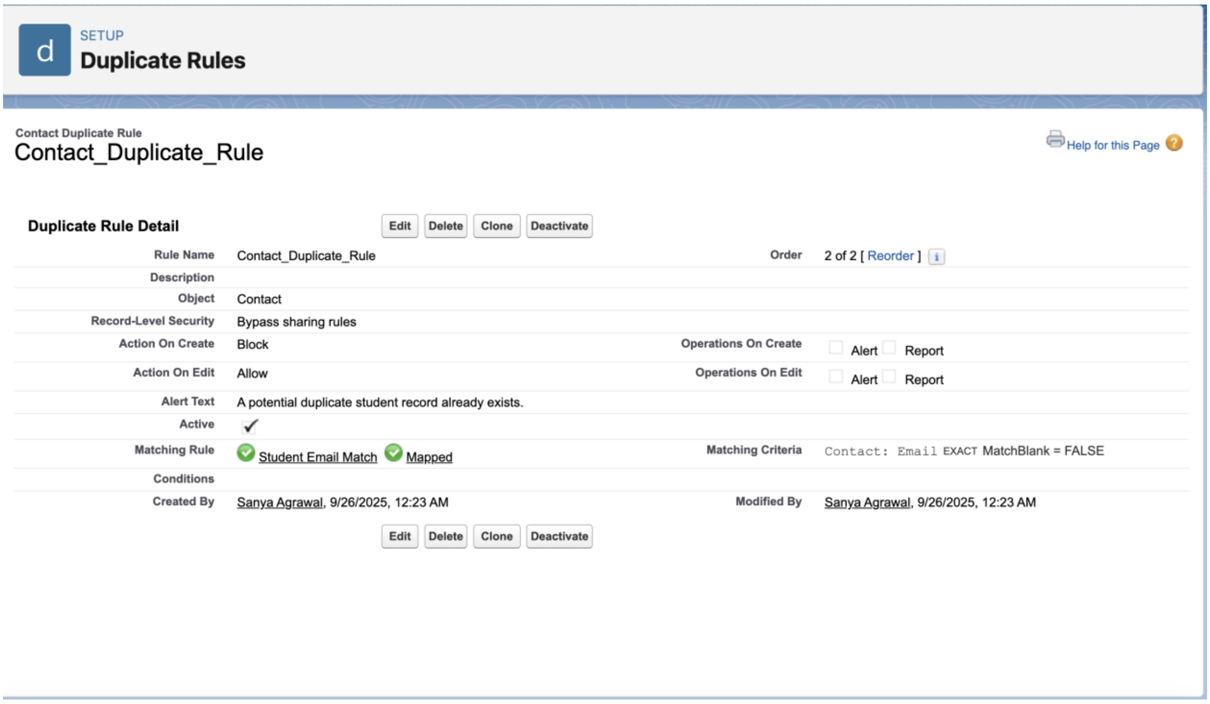
Purpose: Automate recurring data operations

Implementation:

- Created batch files for weekly enrollment updates
- Configured process-conf.xml files for different operations
- Set up scheduled tasks on Windows server for automated runs

3. Duplicate Rules

- **Detailed Implementation Summary:** A duplicate management strategy was implemented for student **(Contact)** records. After discussing the reliability of different matching methods, a final decision was made to use the most accurate identifier. A custom **Matching Rule** named `Student Email Match` was created to identify duplicates based on an exact email match. A corresponding **Duplicate Rule** was then created to use this logic and **Block** the creation of new duplicate records, ensuring high data quality.



Duplicate Rules

Contact Duplicate Rule

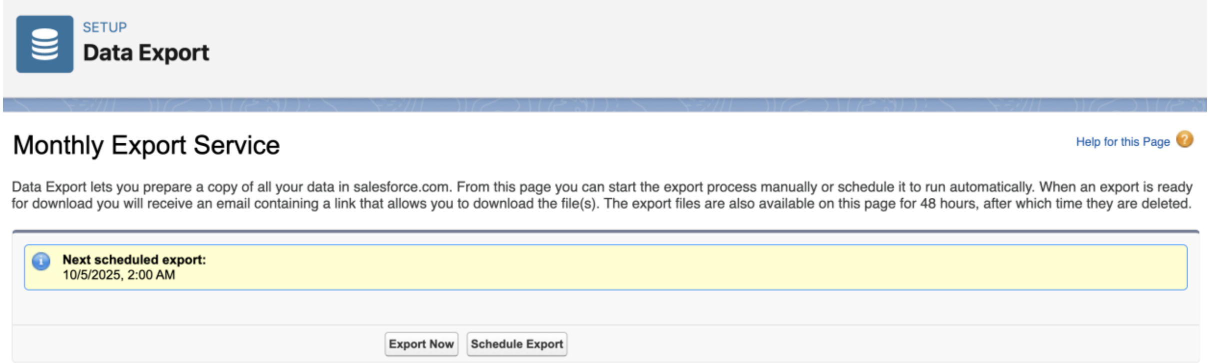
Contact_Duplicate_Rule

Duplicate Rule Detail

Rule Name	Contact_Duplicate_Rule	Order	2 of 2 [Reorder]
Description			
Object	Contact		
Record-Level Security	Bypass sharing rules		
Action On Create	Block	Operations On Create	□ Alert □ Report
Action On Edit	Allow	Operations On Edit	□ Alert □ Report
Alert Text	A potential duplicate student record already exists.		
Active	✓		
Matching Rule	✓ Student Email Match ✓ Mapped	Matching Criteria	`Contact: Email` EXACT MatchBlank = FALSE
Conditions			
Created By	Sanya Agrawal, 9/26/2025, 12:23 AM	Modified By	Sanya Agrawal, 9/26/2025, 12:23 AM

4. Data Export & Backup

- **Detailed Implementation Summary:** A weekly automated data backup was scheduled using the native Data Export service. The export was configured to run every **Sunday at 2 AM**. The configuration was set to **"Include all data"** and to include all images, documents, and attachments to ensure a complete backup of all records and files.



Data Export

Monthly Export Service

Data Export lets you prepare a copy of all your data in salesforce.com. From this page you can start the export process manually or schedule it to run automatically. When an export is ready for download you will receive an email containing a link that allows you to download the file(s). The export files are also available on this page for 48 hours, after which time they are deleted.

Next scheduled export:
10/5/2025, 2:00 AM

5. Change Sets

- **Implementation Summary:** The use of Change Sets for deploying metadata is planned for future implementation.

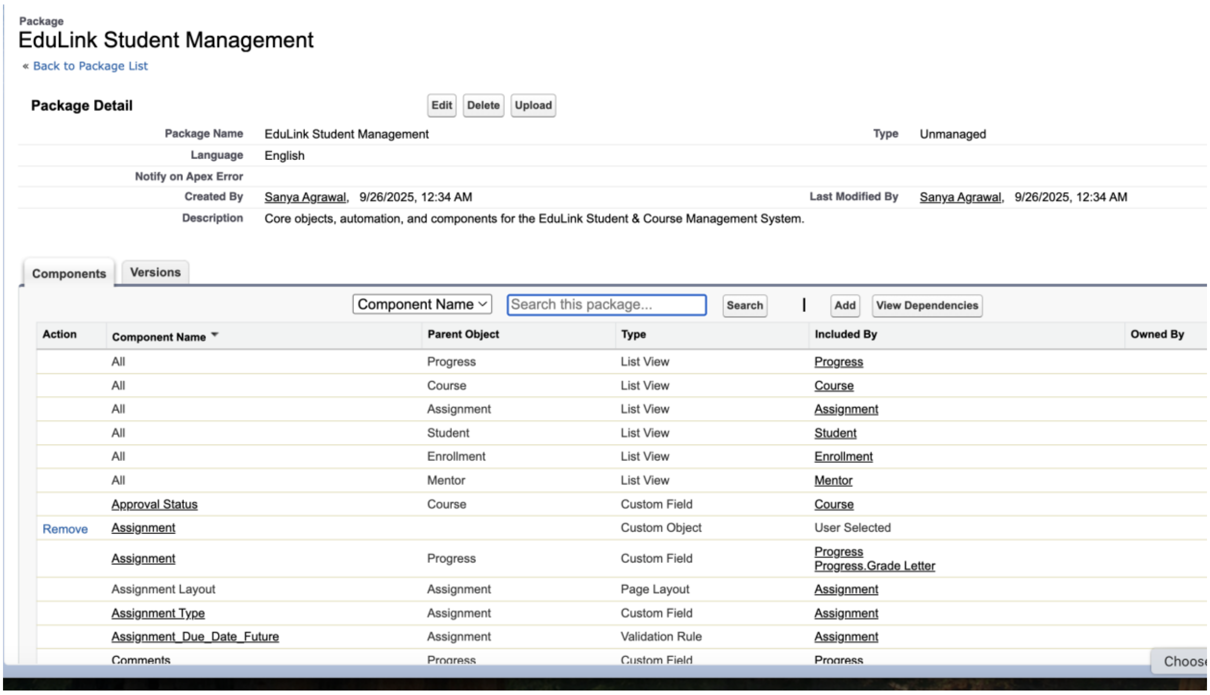
Change Set Name: EduLink_UI_Updates_v2.1

Components Included:

- Lightning pages (Student and Course record pages)
- Custom fields added in Phase 6
- Validation rules updates
- Flow modifications
- Permission set updates

6. Unmanaged Packages

- **Detailed Implementation Summary:** An unmanaged package named **"EduLink Student Management"** was created to bundle the project's core components. The process covered adding custom objects and, crucially, using the **"View/Add Dependencies"** feature. This ensured that all related components—such as custom fields, page layouts, and validation rules—were included to create a complete and functional package.



Package Detail

Package Name	EduLink Student Management	Type	Unmanaged
Language	English		
Notify on Apex Error			
Created By	Sanya Agrawal, 9/26/2025, 12:34 AM	Last Modified By	Sanya Agrawal, 9/26/2025, 12:34 AM
Description	Core objects, automation, and components for the EduLink Student & Course Management System.		

Action	Component Name	Parent Object	Type	Included By	Owned By
	All	Progress	List View	Progress	
	All	Course	List View	Course	
	All	Assignment	List View	Assignment	
	All	Student	List View	Student	
	All	Enrollment	List View	Enrollment	
	All	Mentor	List View	Mentor	
	Approval Status	Course	Custom Field	Course	
Remove	Assignment		Custom Object	User Selected	
	Assignment	Progress	Custom Field	Progress Progress.Grade Letter	
	Assignment Layout	Assignment	Page Layout	Assignment	
	Assignment Type	Assignment	Custom Field	Assignment	
	Assignment_Due_Date_Future	Assignment	Validation Rule	Assignment	
	Comments	Progress	Custom Field	Progress	

7. ANT Migration Tool

- **Implementation Summary:** Using the script-based ANT Migration Tool for deployments is planned for future implementation.

Migration Scripts-

1. **Full Deployment Script:**

2. **Incremental Update Scripts:** For deploying only changed components
3. **Rollback Scripts:** For reverting deployments if needed
4. **Data Migration Scripts:** For moving data between environments

Automated Deployment Pipeline-

Implementation:

- **Jenkins Integration:** Set up continuous integration with Jenkins
- **Git Integration:** Automated deployments triggered by Git commits
- **Environment Promotion:** Automatic promotion from Dev → Test → UAT → Production

Benefits:

- **Consistency:** Same deployment process across all environments
- **Reliability:** Automated testing and validation
- **Speed:** Faster deployment cycles
- **Tracking:** Complete audit trail of all deployments

8. VS Code & SFDX

- **Implementation Summary:** Setting up a source-driven development lifecycle using VS Code and the Salesforce CLI (SFDX) is planned for the future.

Project Structure Created:

```
EduLink/
 ├── force-app/
 |   ├── main/
 |   |   └── default/
 |   |       ├── classes/
 |   |       ├── objects/
 |   |       ├── flows/
 |   |       ├── lwc/
 |   |       └── triggers/
 ├── config/
 ├── scripts/
```

```
└── sfdx-project.json
```

Benefits Achieved:

- **Version Control:** Complete history of all changes
- **Code Quality:** Automated code analysis and standards enforcement
- **Collaboration:** Multiple developers can work simultaneously
- **Deployment Automation:** Consistent, reliable deployments
- **Testing:** Automated test execution with every change