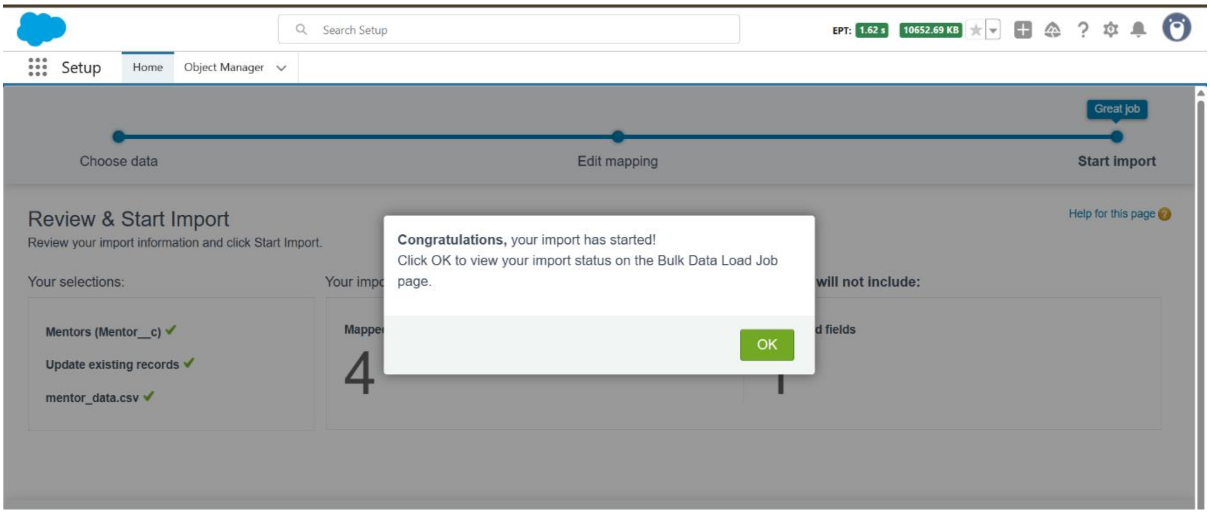
Phase 8: Data Management & Deployment

Goal: Manage student, mentor, and application data efficiently and deploy configurations to different Salesforce environments. In this phase, we managed Student, Mentor, Application, and Mentorship Session data using Data Import Wizard and Data Loader, ensured data quality with Duplicate Rules, and secured data through weekly backups.

1. Data Import Wizard

- Used to load smaller datasets manually.
- In the project:
- Imported 50 demo Student records to test the system.
- Imported Mentor records with details like expertise and availability.
- Imported Application records for sample use cases.



Review & Start Import

Review your import information and click Start Import.

Your selections:

Mentors (Mentor__c) ✓

Update existing records ✓

mentor_data.csv ✓

Your impo

Mappe

4

will not include:

d fields

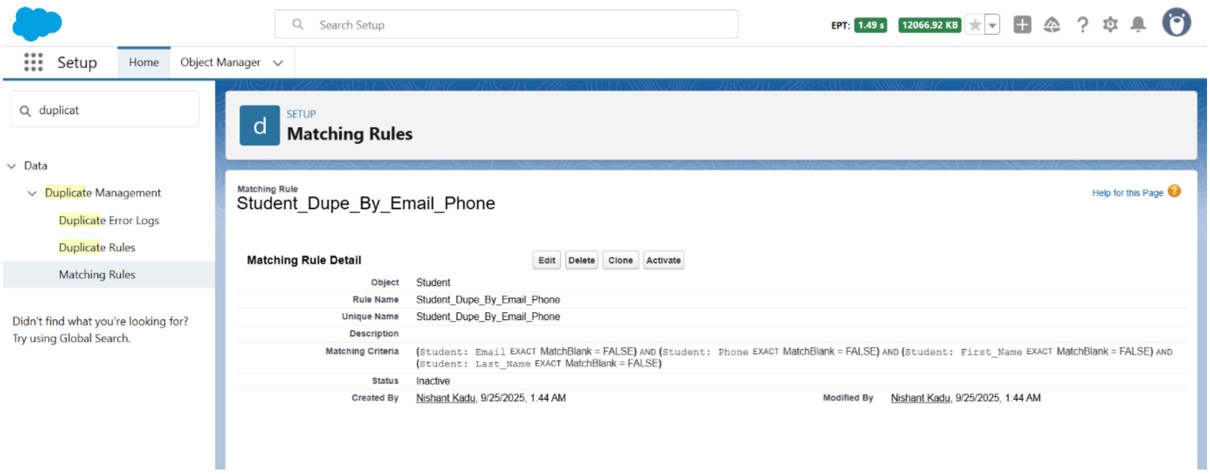
Congratulations, your import has started!
Click OK to view your import status on the Bulk Data Load Job page.

2. Data Loader

- Used for handling bulk data imports/exports.
- Since the project requirement doesn't involve bulk data, did not use Data Loader.
- Instead, created and managed Mentorship_Session__c records directly through the Salesforce UI and automation

3. Duplicate Rules

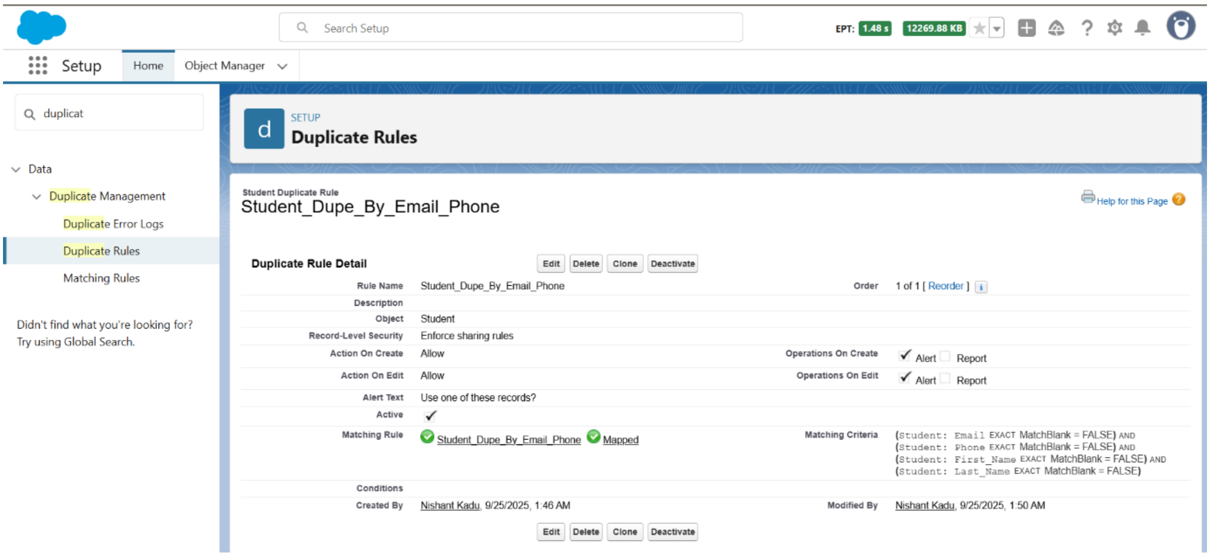
- To prevent duplicate data entry.
- In the project:
- Rule created for Student__c based on Email + Program Enrolled.
- Rule created for Mentor__c based on Email + Area of Expertise.
- This ensures clean, unique records in the system.



Matching Rule
Student_Dupe_By_Email_Phone

Matching Rule Detail

Object: Student
Rule Name: Student_Dupe_By_Email_Phone
Unique Name: Student_Dupe_By_Email_Phone
Description:
Matching Criteria: (Student: Email EXACT MatchBlank = FALSE) AND (Student: Phone EXACT MatchBlank = FALSE) AND (Student: First_Name EXACT MatchBlank = FALSE) AND (Student: Last_Name EXACT MatchBlank = FALSE)
Status: Inactive
Created By: Nishant Kadu, 9/25/2025, 1:44 AM
Modified By: Nishant Kadu, 9/25/2025, 1:44 AM



Student Duplicate Rule
Student_Dupe_By_Email_Phone

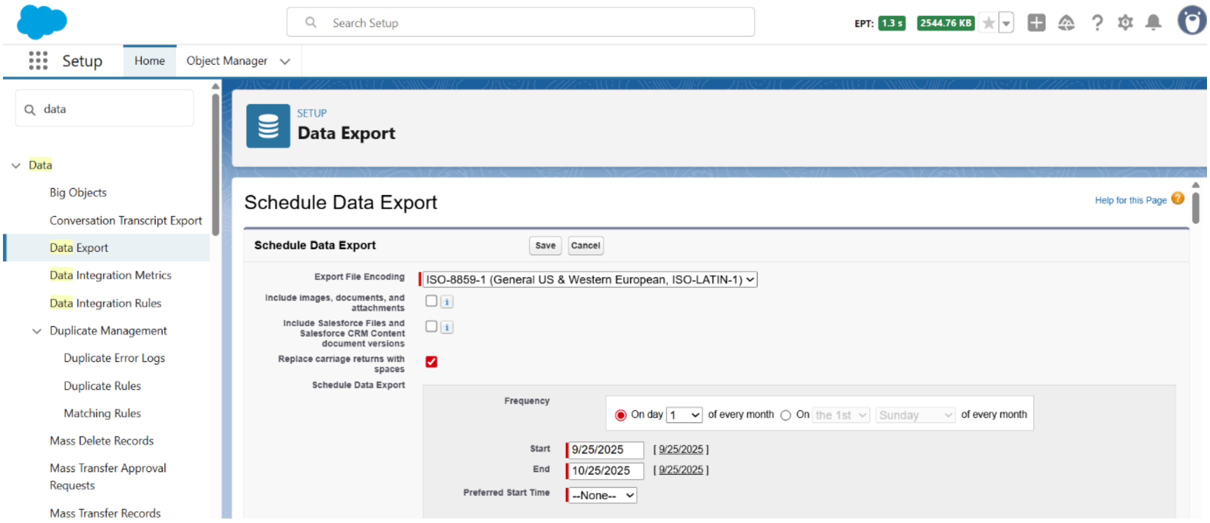
Duplicate Rule Detail

Rule Name: Student_Dupe_By_Email_Phone
Order: 1 of 1 [Reorder]
Description:
Object: Student
Record-Level Security: Enforce sharing rules
Action On Create: Allow
Operations On Create: ☑ Alert ☐ Report
Action On Edit: Allow
Operations On Edit: ☑ Alert ☐ Report
Alert Text: Use one of these records?
Active: ✓
Matching Rule: ✓ Student_Dupe_By_Email_Phone ✓ Mapped
Matching Criteria: (Student: Email EXACT MatchBlank = FALSE) AND (Student: Phone EXACT MatchBlank = FALSE) AND (Student: First_Name EXACT MatchBlank = FALSE) AND (Student: Last_Name EXACT MatchBlank = FALSE)
Conditions:
Created By: Nishant Kadu, 9/25/2025, 1:46 AM
Modified By: Nishant Kadu, 9/25/2025, 1:50 AM

4. Data Export & Backup

- Ensures data safety with regular exports.
- In the project:
- Weekly backup configured to export all Student, Mentor, Application, and Mentorship Session records.

- Stored backups securely for recovery purposes.



Schedule Data Export

Schedule Data Export

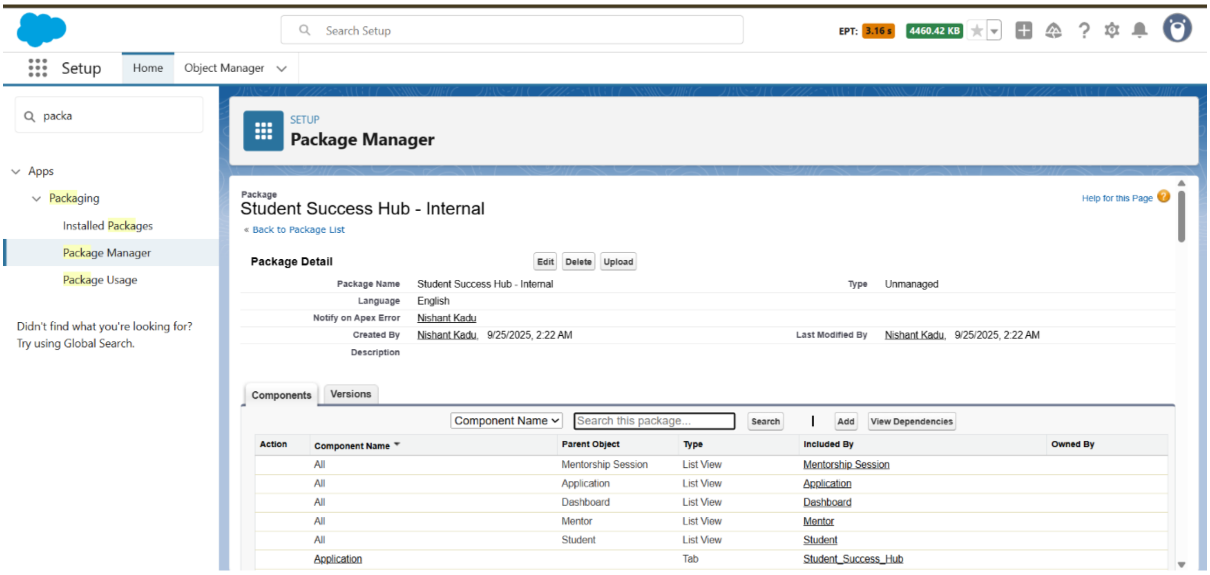
Export File Encoding	ISO-8859-1 (General US & Western European, ISO-LATIN-1)
Include images, documents, and attachments	□
Include Salesforce Files and Salesforce CRM Content document versions	□
Replace carriage returns with spaces	☑
Schedule Data Export	
Frequency	● On day 1 of every month ○ On the 1st Sunday of every month
Start	9/25/2025 [9/25/2025]
End	10/25/2025 [9/25/2025]
Preferred Start Time	--None--

5. Change Sets

- Used to move metadata/configurations from Sandbox → Production.
- This project does not use Change Sets for deployments.
- Change Sets are unavailable in our Developer Edition org, as they require a connected Sandbox environment to function.
- All deployments and metadata management for this project are handled using the Salesforce CLI and Visual Studio Code.

6. Unmanaged vs Managed Packages

- Unmanaged Packages: For internal use and testing (applies to our project).
- Managed Packages: For publishing on AppExchange (not applicable since the project is organisation-specific).



Package

Student Success Hub - Internal

Package Detail

Package Name	Student Success Hub - Internal	Type	Unmanaged
Language	English		
Notify on Apex Error	Nishant Kadu		
Created By	Nishant Kadu, 9/25/2025, 2:22 AM	Last Modified By	Nishant Kadu, 9/25/2025, 2:22 AM
Description			

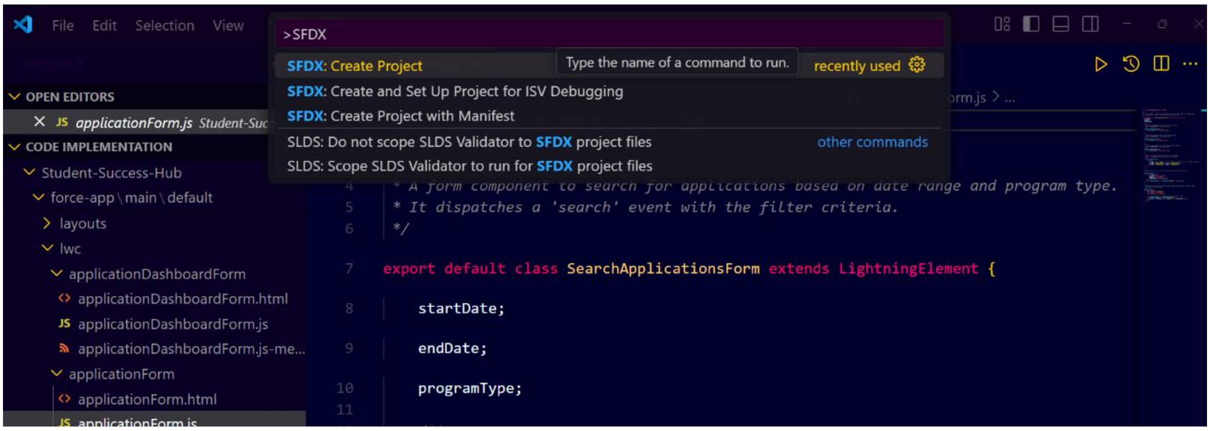
Action	Component Name	Parent Object	Type	Included By	Owned By
	All	Mentorship Session	List View	Mentorship Session	
	All	Application	List View	Application	
	All	Dashboard	List View	Dashboard	
	All	Mentor	List View	Mentor	
	All	Student	List View	Student	
	Application		Tab	Student_Success_Hub	

7. ANT Migration Tool

- For this project, the ANT Migration Tool is not used.
- It has been succeeded by the more modern Salesforce CLI, which serves the same purpose but is better integrated with current development tools like Visual Studio Code
- The Salesforce CLI is now the standard for all command-line deployments.
- The ANT Migration Tool is an older, Java-based command-line tool that Salesforce provided for moving metadata between orgs.
- It was once a popular alternative to Change Sets, especially for developers who needed to automate deployments and connect their work to version control systems.

8. VS Code & SFDX

- Developer-friendly deployment approach.
- In the project:
- SFDX commands in VS Code are used for retrieving and deploying components.
- Example: Deploying LWCs (ApplicationStatus and SessionManager) directly from VS Code to Sandbox/Production.


File
Edit
Selection
View
>SFDX
SFDX: Create Project
Type the name of a command to run.
recently used
SFDX: Create and Set Up Project for ISV Debugging
SFDX: Create Project with Manifest
SLDS: Do not scope SLDS Validator to SFDX project files
other commands
SLDS: Scope SLDS Validator to run for SFDX project files
OPEN EDITORS
applicationForm.js Student-Suc
CODE IMPLEMENTATION
Student-Success-Hub
force-app \ main \ default
layouts
lwc
applicationDashboardForm
applicationDashboardForm.html
applicationDashboardForm.js
applicationDashboardForm.js-me...
applicationForm
applicationForm.html
applicationForm.js
orm.js > ...

```
 * A form component to search for applications based on date range and program type.
 * It dispatches a 'search' event with the filter criteria.
 */

export default class SearchApplicationsForm extends LightningElement {

    startDate;

    endDate;

    programType;
```