Phase 8: Data Management & Deployment

Travel and Tourism Management System

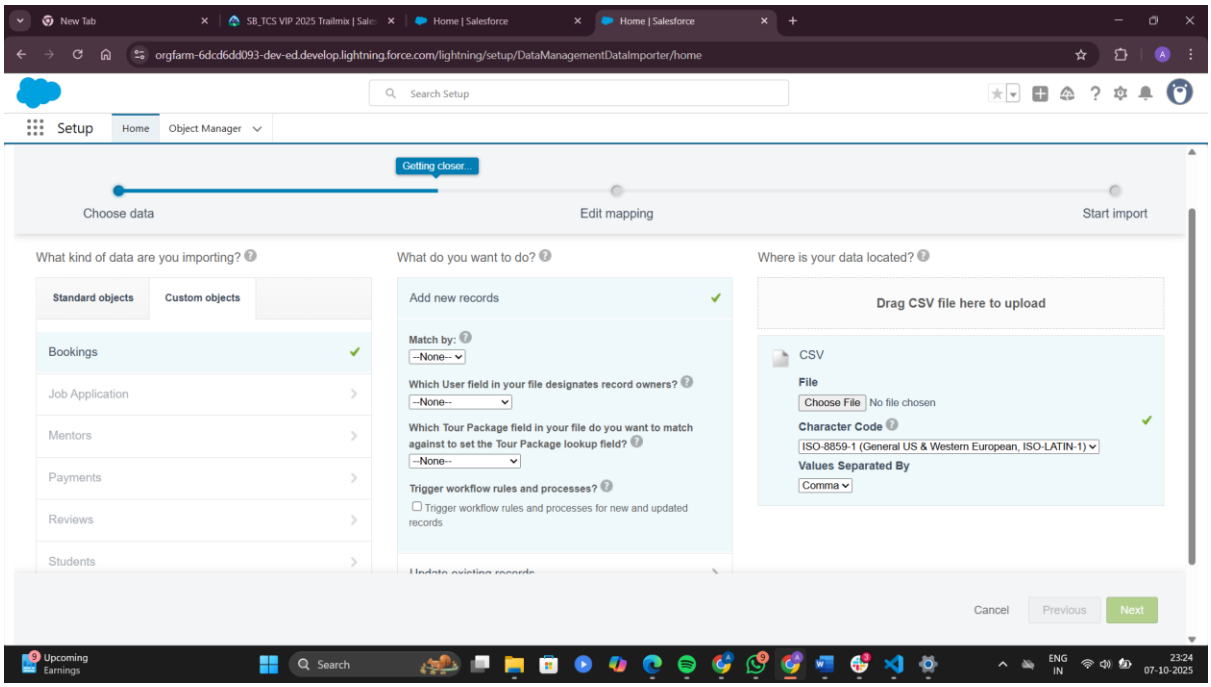
---

1. Data Import Wizard

Purpose:
Easily import data records for standard and custom objects (e.g. Booking__c, Customer__c, TourPackage__c).

Steps:

1. Go to Setup → Data → Data Import Wizard.
2. Click "Launch Wizard".
3. Select the object you want to import (e.g. Booking__c).
4. Choose to add new records or update existing data, depending on your use case.
5. Upload your prepared CSV file.
6. Map CSV columns to Salesforce fields (e.g., BookingDate → Booking_Date__c).
7. Click Next and start the import.



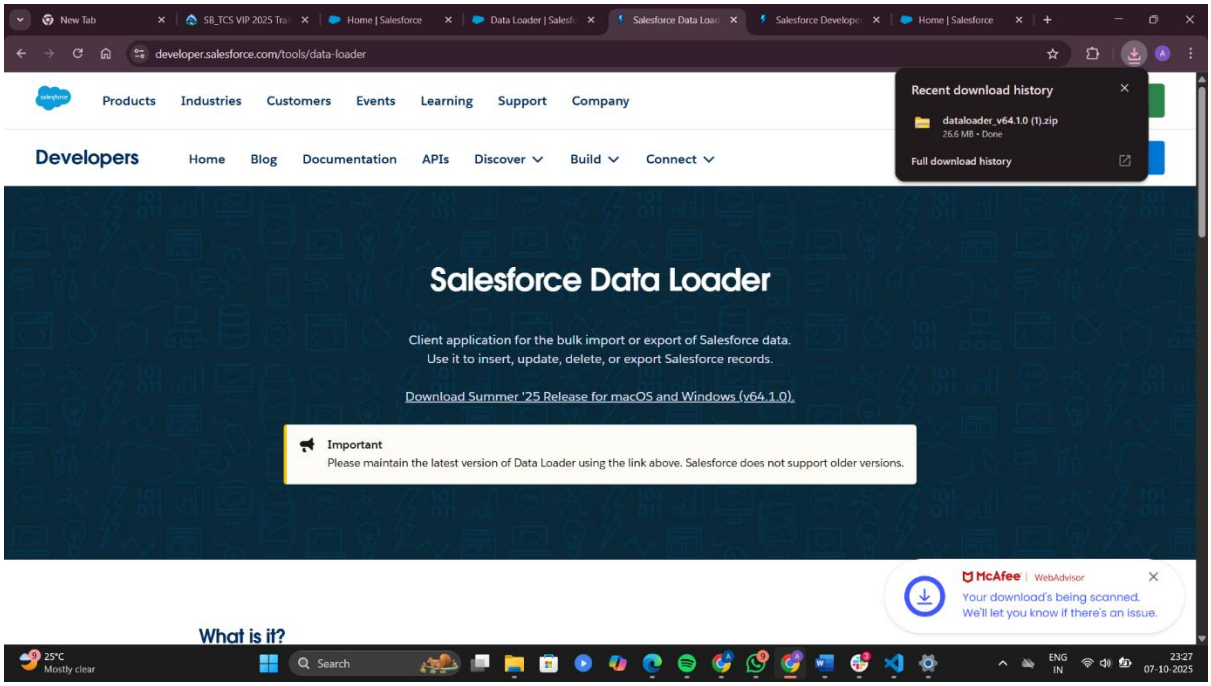
---

2. Data Loader

Purpose:
Use for large datasets (over 50,000 records), complex operations like upsert, delete, or scheduled batch imports.

Steps:

1. Download and install Salesforce Data Loader.
2. Open Data Loader and log in with Salesforce credentials.
3. Choose the operation (Insert, Update, Upsert, Delete, or Export).
4. Select the object (e.g. Booking__c).
5. Browse and select the CSV file.
6. Map CSV columns to Salesforce fields.
7. Run the operation.
8. Review success and error logs generated.

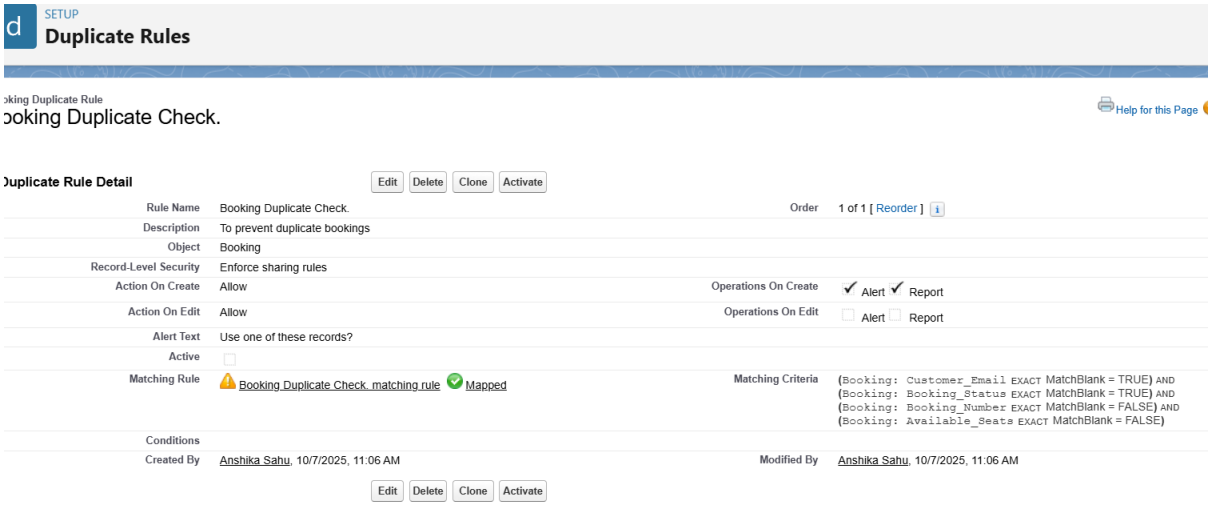


3. Duplicate Rules

Purpose:
Prevent duplicate entries for key objects like Customers or Bookings.

Steps:

1. Go to Setup → Duplicate Management → Duplicate Rules.
2. Click "New Rule" and select your object (e.g., Booking__c).
3. Create a Matching Rule that defines matching logic (e.g., Booking duplicate check).
4. Specify actions on create or update — block duplicates or allow with alert.
5. Activate the rule.



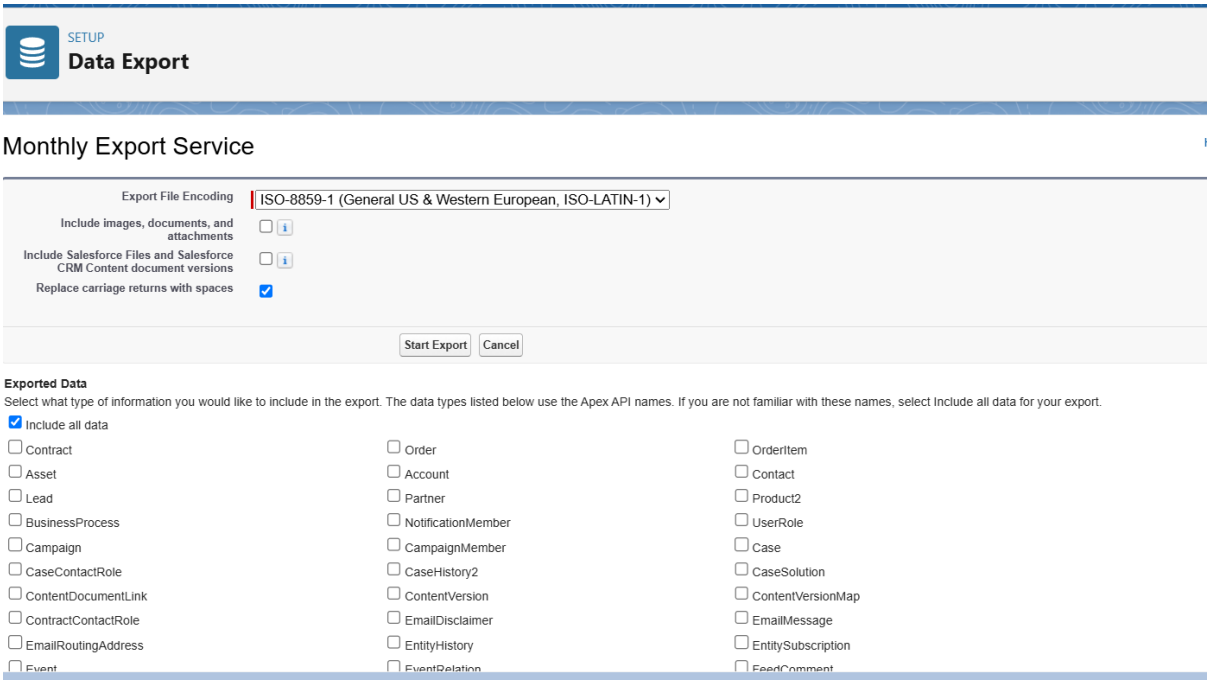
---

4. Data Export & Backup

Purpose:
Schedule regular backups or manually export data to safeguard against loss.

Steps:

1. Setup → Data → Data Export.
2. Choose “Export Now” for immediate export or schedule automated exports.
3. Select objects to include (e.g. Booking__c).
4. Optionally include attachments and documents.
5. Download the backup ZIP file containing CSV files.



---

5. Change Sets

Purpose:
Deploy metadata and configurations between Salesforce orgs (e.g., Sandbox → Production).

Steps:

1. Setup → Deployment → Outbound Change Sets.
2. Click “New” and name your change set.
3. Add required components: Objects, Fields, Apex Classes, Validation Rules, Flows.
4. Upload to target org.
5. In the target org, validate and deploy the change set.

---

6. Unmanaged vs Managed Packages

Unmanaged Packages:

- For distributing customizations for learning or demo.
- Changes after installation are not synced.

Managed Packages:

- Production-ready apps with upgrade capabilities and licensing.

Steps to create:

1. Setup → Packages → New.
2. Choose Managed or Unmanaged package.

3. Add components, upload package.

---

7. ANT Migration Tool

Purpose:
Script-based deployment via XML metadata files.

Steps:

1. Download and configure the Force.com Migration Tool (ANT).
2. Prepare build.xml and package.xml files specifying components.
3. Use commands:
    - ant retrieve
    - ant deploy
4. Validate deployments before pushing to production.

---

8. VS Code & SFDX (Salesforce CLI)

Purpose:
Modern development environment for metadata management, source control, and deployment.

Steps:

1. Install VS Code and Salesforce Extension Pack.
2. Authenticate your org using CLI:

text

sfdx force:auth:web:login -a YourOrgAlias

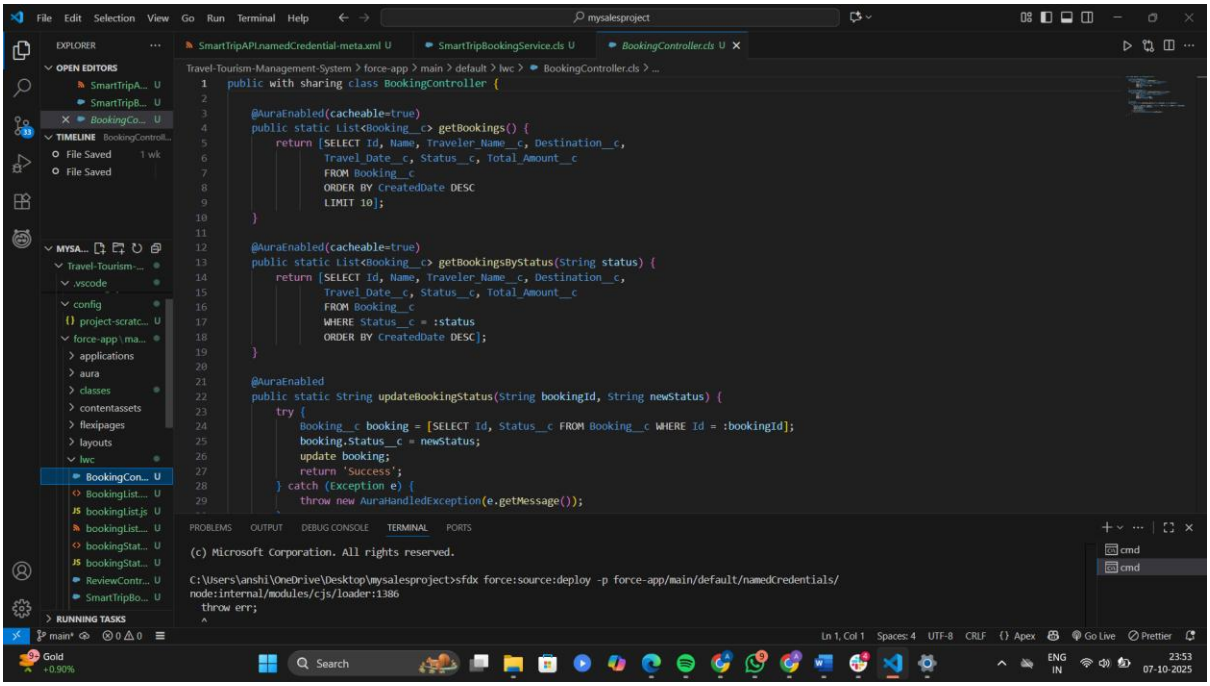
3. Use commands to retrieve and deploy metadata:

text

sfdx force:source:retrieve -m CustomObject, ApexClass

sfdx force:source:deploy -p force-app

4. Create new Lightning Web Components, Apex classes from VS Code.
5. Integrate Git for version control.


File Edit Selection View Go Run Terminal Help
mysalesproject
EXPLORER
OPEN EDITORS
SmartTripA... U
SmartTripB... U
BookingCo... U
TIMELINE BookingControll...
File Saved 1 wk
File Saved
MYSA...
Travel-Tourism-...
.vscode
config
project-scratc... U
force-app \ ma...
applications
aura
classes
contentassets
flexipages
layouts
lwc
BookingCon... U
BookingList.... U
bookingList.js U
bookingList.... U
bookingStat... U
bookingStat... U
ReviewContr... U
SmartTripBo... U
RUNNING TASKS
SmartTripAPI.namedCredential-meta.xml U
SmartTripBookingService.cls U
BookingController.cls U
Travel-Tourism-Management-System > force-app > main > default > lwc > BookingController.cls > ...
public with sharing class BookingController {
@AuraEnabled(cacheable=true)
public static List<Booking__c> getBookings() {
return [SELECT Id, Name, Traveler_Name__c, Destination__c,
Travel_Date__c, Status__c, Total_Amount__c
FROM Booking__c
ORDER BY CreatedDate DESC
LIMIT 10];
}
@AuraEnabled(cacheable=true)
public static List<Booking__c> getBookingsByStatus(String status) {
return [SELECT Id, Name, Traveler_Name__c, Destination__c,
Travel_Date__c, Status__c, Total_Amount__c
FROM Booking__c
WHERE Status__c = :status
ORDER BY CreatedDate DESC];
}
@AuraEnabled
public static String updateBookingStatus(String bookingId, String newStatus) {
try {
Booking__c booking = [SELECT Id, Status__c FROM Booking__c WHERE Id = :bookingId];
booking.Status__c = newStatus;
update booking;
return 'Success';
} catch (Exception e) {
throw new AuraHandledException(e.getMessage());
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
(c) Microsoft Corporation. All rights reserved.
C:\Users\anshi\OneDrive\Desktop\mysalesproject>sfdx force:source:deploy -p force-app/main/default/namedCredentials/
node:internal/modules/cjs/loader:1386
throw err;
^
cmd
cmd
main*
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF {} Apex Go Live Prettier
Gold
+0.90%
Search
ENG
IN
23:53
07-10-2025