

Phase 8: Data Management & Deployment

Introduction

In this phase, we focus on data handling (import, export, deduplication, backup) and deployment (moving metadata from Dev/Sandbox to Production). The goal is to ensure clean travel package, booking, and customer data, reliable backups, and smooth deployments for the **TourEase Travel & Tourism Booking System**.

1. Data Import Wizard

The Data Import Wizard is used for importing small to medium datasets (up to 50,000 records). For our project, we used it to import initial Travel Packages and Customer records.

Steps:

1. Go to **Setup** → **Data Import Wizard** → **Launch Wizard**.
2. Select the custom object **Travel_Package__c**.
3. Upload the prepared **TravelPackages.csv** file (with Package Name, Destination, Start Date, End Date, Price, and Total Seats).
4. Map CSV fields to Salesforce fields.
5. Choose record owner option → “Use values from CSV” (we used Salesforce User Email).
6. Run Import → verify in the **Travel Packages** tab.

The screenshot shows the Salesforce Data Import Wizard interface. At the top, there's a progress bar with three steps: 'Choose data' (completed), 'Edit mapping' (current step), and 'Start import'. Below the progress bar, the title is 'Edit Field Mapping: Travel Packages'. A note states: 'Your file has been auto-mapped to existing Salesforce fields, but you can edit the mappings if you wish. Unmapped fields will not be imported.' Below this is a table for mapping CSV headers to Salesforce fields.

Edit	Mapped Salesforce Object	CSV Header	Example	Example	Example
Change	Destination	Destination	Goa	Himachal Pradesh	Kerala
Change	Start Date	Start Date	15-10-2025	01-11-2025	05-12-2025
Change	End Date	End Date	20-10-2025	10-11-2025	12-12-2025
Change	Price	Price	15000	25000	18000
Change	Total Seats	Total Seats	50	40	30
Change	Available Seats	Available Seats	50	40	30
Change	Booked Seats	Booked Seats	0	0	0

At the bottom right, there are buttons for 'Cancel', 'Previous', and 'Next'. A URL bar at the very bottom shows: <https://orgfem-eb1945479c-dev-ed.develop.lightning.force.com/dataimporter/dataimporter.app>

2. Data Loader

For larger datasets (above 50,000 records), Data Loader is used. In our project, this is useful for bulk imports of **Bookings** and **Customer Records**.

Steps:

1. Download & install Salesforce Data Loader.
2. Login with Salesforce credentials (Sandbox first).
3. Choose **Insert** operation.
4. Select object (e.g., Booking__c).
5. Upload CSV file and map fields.
6. Run import and review logs for success/failure.

3. Duplicate Rules

To prevent duplicate customer records (same email address or phone number), we created a Matching Rule and Duplicate Rule.

Steps:

1. Setup → Quick Find → **Matching Rules** → **New**.
 - Object: **Customer__c** (or Contact).
 - Matching Field: **Email** or **Phone**.
 - Save & Activate.
2. Setup → Quick Find → **Duplicate Rules** → **New**.
 - Object: **Customer__c**.
 - Matching Rule: **Customer_Email_ExactMatch**.
 - Action: **Block or Allow but Alert**.
 - Save & Activate.

The screenshot shows the Salesforce 'Duplicate Rules' setup page. The left sidebar contains navigation links for Setup, Home, and Object Manager. The main content area is titled 'Duplicate Rules' and includes a 'What Are Duplicate Rules?' section. Below this, there is a table listing existing duplicate rules. The table has columns for Rule Name, Description, Object, Matching Rule, Active status, Last Modified By, and Last Modified Date. Three rules are listed: Standard Account Duplicate Rule, Standard Contact Duplicate Rule, and Standard Lead Duplicate Rule. Each rule is associated with a specific matching rule and is currently active.

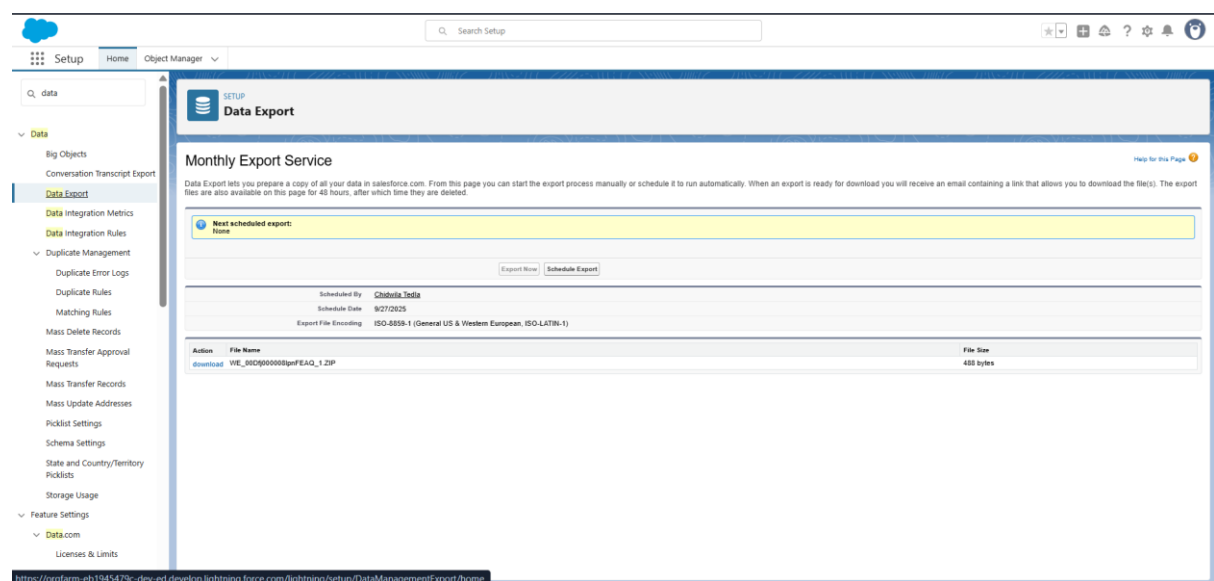
Rule Name	Description	Object	Matching Rule	Active	Last Modified By	Last Modified Date
Standard Account Duplicate Rule	Identify accounts that duplicate other accounts.	Account	Standard Account Matching Rule	✓	DEFG	9/18/2025
Standard Contact Duplicate Rule	Identify contacts that duplicate other contacts and leads.	Contact	Standard Lead Matching Rule Standard Contact Matching Rule	✓	DEFG	9/18/2025
Standard Lead Duplicate Rule	Identify leads that duplicate other leads and contacts.	Lead	Standard Lead Matching Rule Standard Contact Matching Rule	✓	DEFG	9/18/2025

4. Data Export & Backup

Regular data backups are essential for compliance and recovery.

Steps:

1. Setup → Quick Find → **Data Export**.
2. Select objects: **Travel_Package__c, Booking__c, Customer__c, Payment__c**.
3. Schedule export weekly (e.g., Saturday at midnight).
4. Salesforce generates a .zip file and sends a download link via email.



5. Unmanaged vs Managed Packages

For project sharing:

- **Unmanaged Package:** best for portfolios, training, and demos. Recipients can edit the components.
- **Managed Package:** locked, versioned, used for AppExchange distribution.

We used an **Unmanaged Package** for the TourEase project.

6. SFDX & VS Code Deployment (Best Practice)

While Change Sets are beginner-friendly, Salesforce DX (SFDX) with VS Code is best practice for developers.

Steps:

1. Install **VS Code + Salesforce Extension Pack**.
2. Authenticate Org:

3. `sfdx auth:web:login -a DevOrg`
4. Pull metadata:
5. `sfdx force:source:pull`
6. Deploy to target org:
7. `sfdx force:source:deploy -p force-app`
8. Track changes with GitHub for version control.

```

force-app > main > default > classes > TravelPackageController.cls > ...
1 public with sharing class TravelPackageController {
5     public static List<TravelPackage__c> getTravelPackages() {
13         ;
14         return packages;
15     } catch(Exception e) {
16         System.debug('Error fetching travel packages: ' + e.getMessage());
17         return new List<TravelPackage__c>();
18     }
19 }
20
21

```

```

PS C:\Users\91756\OneDrive\Desktop\TourEase\TourEase> sfdx auth:web:login -a DevOrg
Successfully authorized chidalia799@agentforce.com with org ID 0a0fj0088001pnfEAg
PS C:\Users\91756\OneDrive\Desktop\TourEase\TourEase>

```

```

1 import { LightningElement, track, api } from 'lwc';
2 import searchAirports from '@salesforce/apex/BookingAirportController.searchAirports';
3 import updateBookingAirports from '@salesforce/apex/BookingAirportController.updateBookingAirports';
4
5 export default class AirportSearch extends LightningElement {
6     @api bookingId;
7     @track keyword = '';
8     @track airports = [];
9
10    handleInput(event) {
11        this.keyword = event.target.value;
12        if(this.keyword.length >= 2){
13            searchAirports({ keyword: this.keyword })
14                .then(result => { this.airports = result; })
15                .catch(error => { console.error(error); });
16        } else {
17            this.airports = [];
18        }
19    }
20
21    handleSelect(event){
22        const index = event.target.dataset.index;
23        const selected = this.airports[index];
24
25        updateBookingAirports({
26            bookingId: this.bookingId,
27            departure: selected.name,
28            departureIATA: selected.iataCode,
29            arrival: selected.name,
30            arrivalIATA: selected.iataCode
31        });
32        this.airports = [];
33    }
34 }
35

```

```

force-app > main > default > lwc > airportSearch > JS airportSearch.js > ...

```

Conclusion

Phase 8 ensures that the TourEase Travel & Tourism Booking System is:

- **Data-ready** (imported & deduplicated),
- **Secure** (backups scheduled),
- **Deployment-ready** (Change Sets / Unmanaged Package / SFDX).

With clean data and reliable deployment methods, the project is now stable for production rollout.