

## Phase 8: Data Management & Deployment

### Objective:

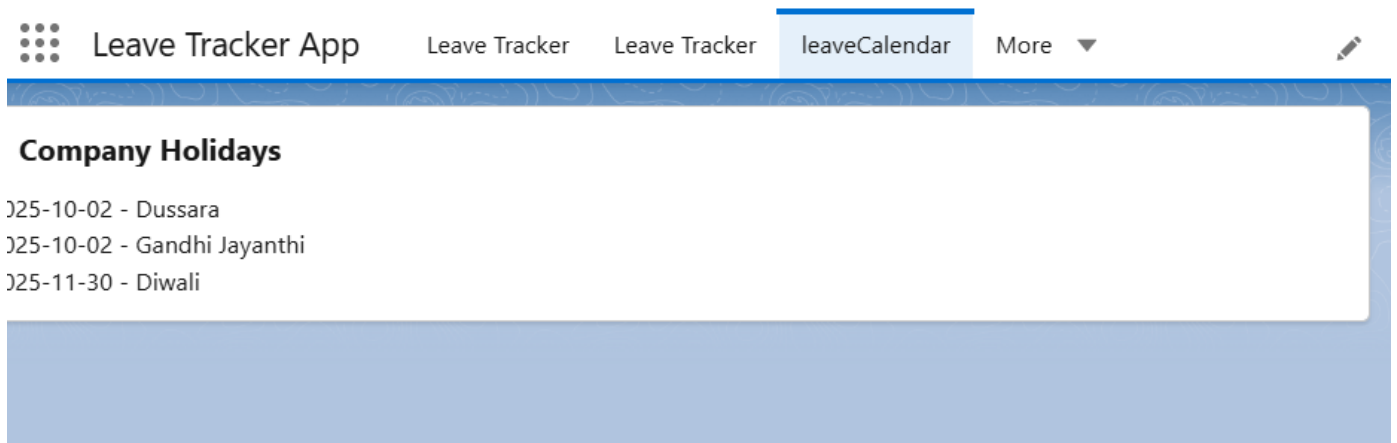
To ensure smooth and error-free deployment of Leave Tracker customizations from development to higher environments (UAT, Production) using best practices and tools.

### Data Import Wizard

- Used to load initial employee data (Users, basic LeaveRequest\_\_c records).
- Supports CSV files.
- Best for small/medium data volumes (< 50,000 records).

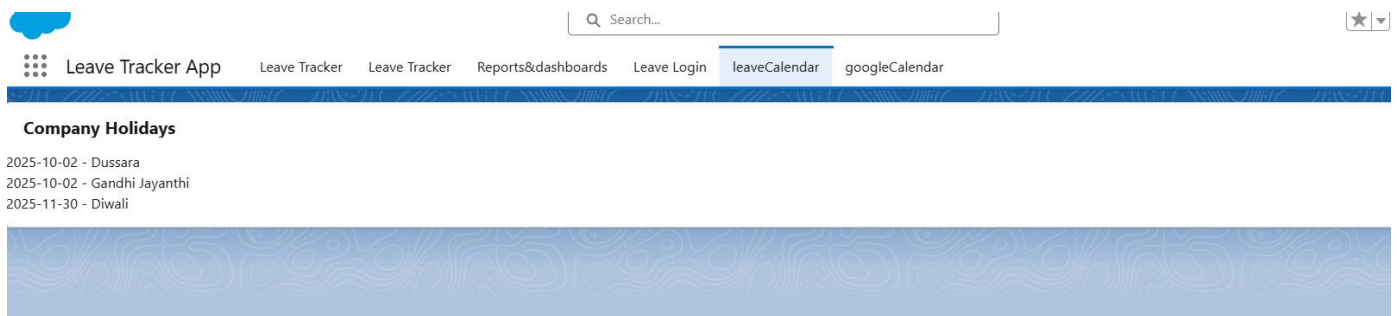
### Calendar:

- Google Calendar / Outlook Tab: Embed or sync employee calendars inside Salesforce.
- Purpose: Employees can see leave requests alongside their personal/work calendar to avoid conflicts.



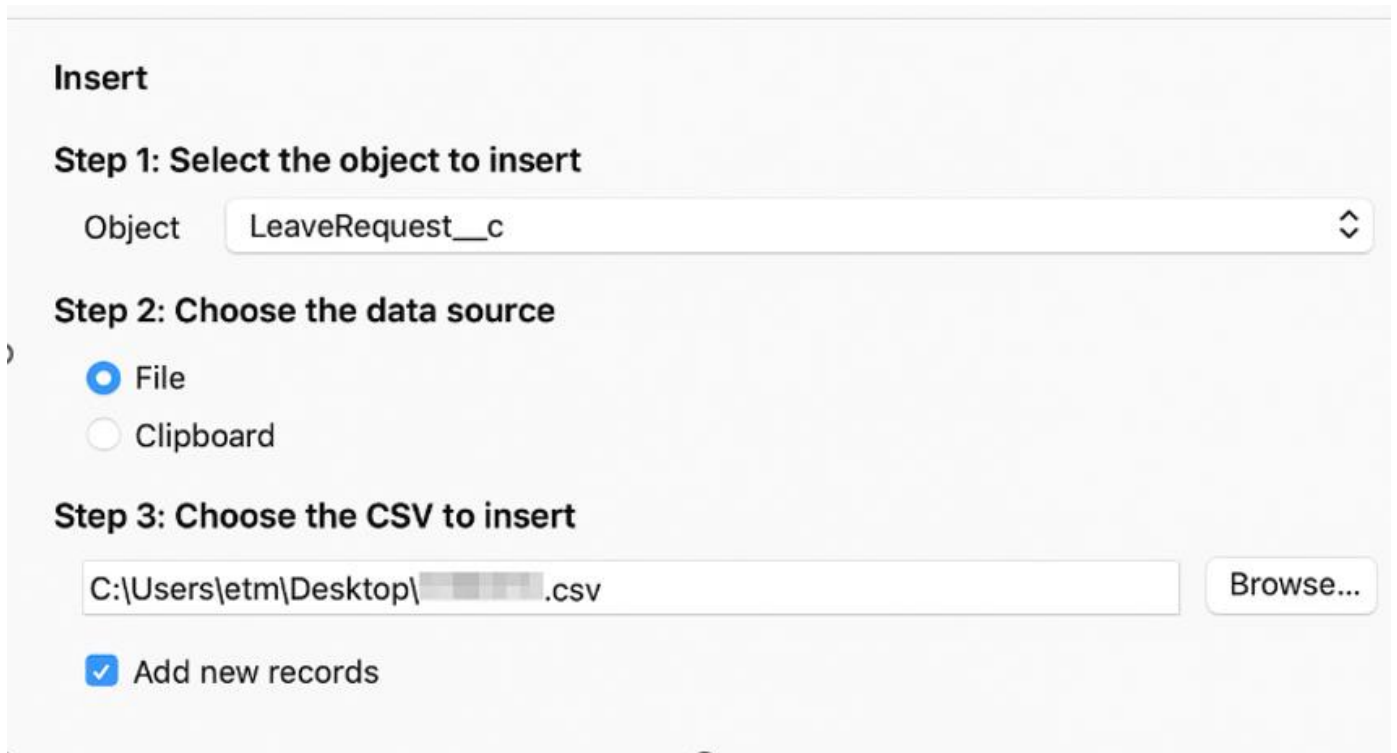
### API-Based Integrations (Advanced)

- Google Calendar API / Microsoft Graph API: Automatically push approved leaves to external calendars using Apex or middleware.
- Purpose: Real-time syncing of leave events with company calendars.



## Data Loader

- Used to perform bulk operations (insert, update, delete, upsert) for large datasets.
- Example: Bulk upload of historical leave records.
- Supports command-line interface for automation.



The screenshot shows the 'Insert' window of the Salesforce Data Loader. It is divided into three steps:

- Step 1: Select the object to insert**  
Object: LeaveRequest\_\_c
- Step 2: Choose the data source**  
☒ File  
☐ Clipboard
- Step 3: Choose the CSV to insert**  
C:\Users\etm\Desktop\[redacted].csv Browse...  
☒ Add new records

## Change Sets

Used Change Sets for deploying metadata between environments (Sandbox → Production).

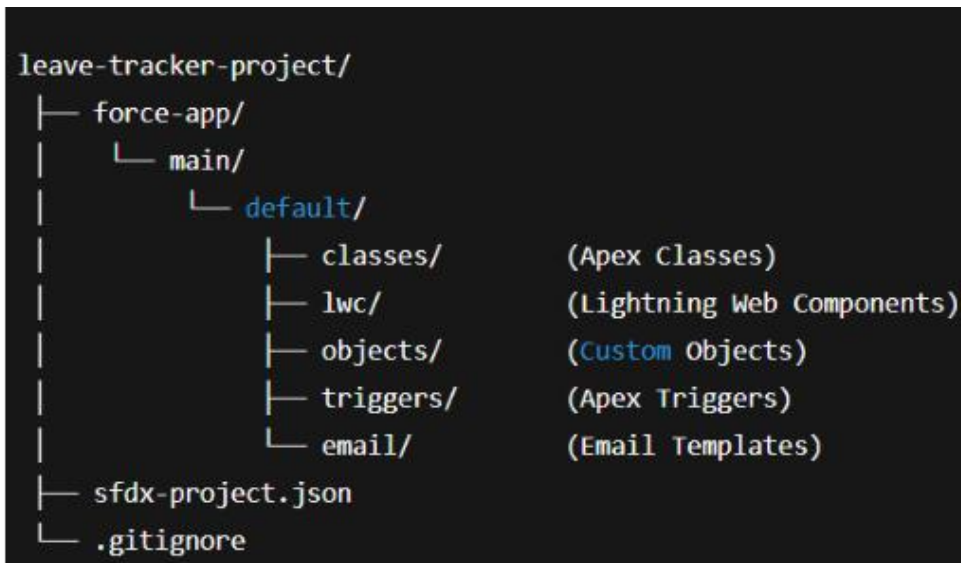
Included:

- Custom Objects (LeaveRequest\_\_c)
- Apex Classes (LeaveRequestController, Triggers)
- Lightning Web Components (applyLeave, myLeaves, leaveRequest)
- Email Templates & Flows

# VS Code Deployment Setup

## Prerequisites

- Install Visual Studio Code.
- Install Salesforce CLI (SFDX).
- Install Salesforce Extension Pack in VS Code.
- Connect Org → `sfdx force:auth:web:login -d -a DevHub`.
- Sample Project Structure (VS Code)



- Retrieve metadata from source org
  - `sfdx force:source:retrieve -m ApexClass,CustomObject,LWC`
- Deploy metadata to target org
  - `sfdx force:source:deploy -p force-app/main/default`
- Run all tests before deployment
  - `sfdx force:apex:test:run --resultformat human --codecoverage`

## Deployment Checklist

- Code Quality Check (PMD, Prettier for LWC).
- Apex Test Coverage  $\geq 75\%$ .
- Run Validation Deployment (test without committing).
- Backup Production metadata before final push.
- Post-deployment steps (activate Flows, verify Email Deliverability).

