Phase 8: Data Management & Deployment

Goal: Manage leave-related data efficiently and ensure smooth deployment between environments.

1. Data Import Wizard

- Use Salesforce **Data Import Wizard** to import small sets of data.
- Example: Import 50 demo **Employee LeaveRequest__c** records for testing.
- Simple UI-based tool → no technical setup required.

2. Data Loader

- For bulk operations (insert, update, delete).
- Example: Import **thousands of leave records** from an external HR system.
- Supports CSV-based upload and scheduled automation.

Example CLI Command:

```
sfdx force:data:tree:import --plan leaveRequestPlan.json
```

3. Duplicate Rules

- Prevents multiple **leave requests** for the same employee on the same dates.
- Example: Duplicate Rule → block if *From_Date__c* and *To_Date__c* overlap with existing leave request for the same employee.

  Example Rule:

  Block duplicate LeaveRequest__c if Employee = same and From Date/To Date overlaps

4. Data Export & Backup

- Weekly export of all **LeaveRequest__c** and **Employee__c** data.
- Backup ensures HR can recover records if any accidental deletion occurs.

5. Change Sets

- Move **LWC components (applyLeave, myLeaves, leaveRequest)**, validation rules, approval processes, and workflows from **Sandbox** → **Production**.
- Ensures configuration changes are properly tracked and deployed.

- 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

6. Unmanaged vs Managed Packages

- **Unmanaged Package:** Share code/configurations internally within the organization.
- **Managed Package:** Use if you plan to publish the **Leave Management App** on Salesforce AppExchange for external customers.

7. ANT Migration Tool

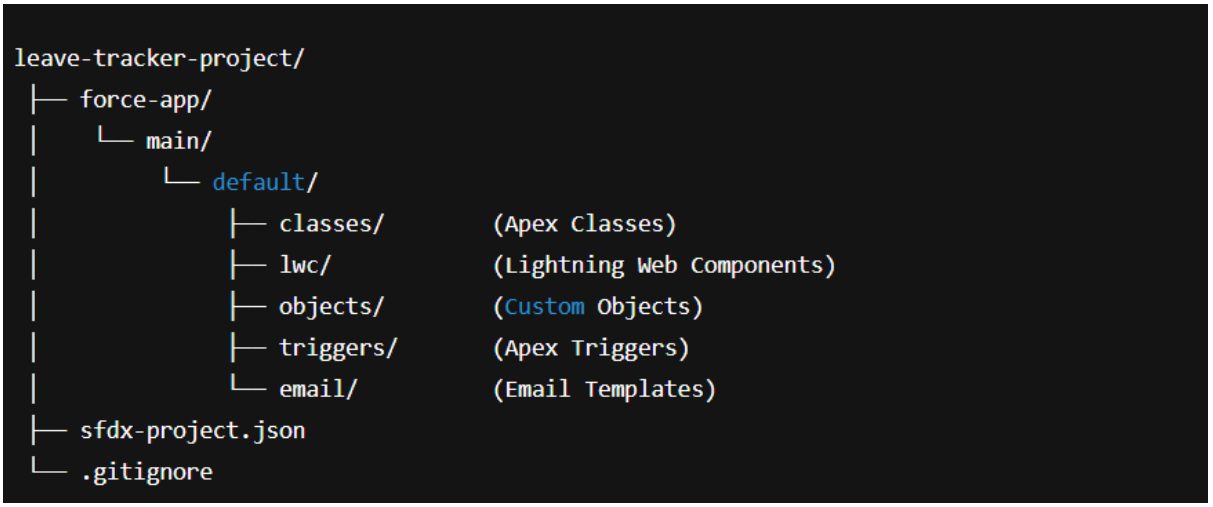
- Command-line based deployment tool.
- Example: Migrate **LeaveRequest__c custom object, Apex classes, and LWCs** between environments.
- Useful for automated deployments in CI/CD pipelines.

8. VS Code & SFDX

- Use **Salesforce DX + VS Code** for developer-friendly deployments.
- Version control with Git.
- Automates pushing/pulling metadata between **scratch orgs, sandboxes, and production**.

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 Deployment Commands

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 ≥ 75%.
✅ Run Validation Deployment (test without committing).
✅ Backup Production metadata before final push.
✅ Post-deployment steps (activate Flows, verify Email Deliverability).

Phase 8 Outcome:

- Reliable data import/export for leave records.
- Duplicate prevention ensures clean data.
- Backup strategy protects against data loss.
- Deployment tools (Change Sets, ANT, SFDX) streamline moving changes from Sandbox → Production.
- Leave Tracking App is secure, scalable, and ready for enterprise deployment.