**Salesforce Project Implementation Documentation**

**Project Title:** Salesforce E-Commerce After-Sales Support CRM  
**Domain:** Retail / E-commerce  
**Focus:** Centralized post-purchase support (returns, refunds, replacements, complaints, warranty claims)  
**Implementation Scope:** Admin + Developer concepts

**Phase 7: Integration**

**Objectives**

Enable smooth data exchange between Salesforce and external systems like **Payment Gateway, Courier Service, ERP, and Email/SMS Platforms**. This ensures real-time visibility of refunds, replacements, and delivery updates.

**Activities**

1. **API Integrations**
   * **Payment Gateway API**:
     + Automated refund status sync.
     + API callouts when refund transaction is complete.
   * **Courier Service API**:
     + Courier\_Tracking\_\_x external object fetches shipment updates.
     + Customers can see live delivery status in portal.
2. **Middleware**
   * Used **MuleSoft** to integrate Salesforce with ERP (for stock availability & order sync).
   * Mule connectors for REST APIs & database connectivity.
3. **Email/SMS Integration**
   * Integrated with **Twilio** for SMS notifications (refund initiation, replacement dispatch).
   * **SendGrid** for automated email updates.
4. **Web Services (Apex Callouts)**
   * Implemented **@future(callout=true)** for asynchronous courier updates.
   * REST API consumption for real-time refund reconciliation.
5. **Platform Events**
   * Trigger event when refund > ₹50,000 → push notification to Finance ERP.
6. **Named Credentials**
   * Secured authentication for external API endpoints.
7. **Single Sign-On (SSO)**
   * Configured SSO for agents → one login for Salesforce + ERP system.
8. **Error Handling**
   * Created **Integration\_Error\_\_c object** to log failed API calls.
   * Admin dashboard highlights integration failures.

**Deliverables**

* **Integration Architecture Diagram**
* **API Documentation** (Payment, Courier, ERP)
* **Error Handling & Retry Strategy**

**Phase 8: Data Management**

**Objectives**

Ensure **data accuracy, deduplication, backup, and compliance** while handling sensitive refund/payment details.

**Activities**

1. **Data Migration**
   * Imported **Order history** and **Customer data** using Data Loader.
   * Migrated ~50,000 records into Salesforce.
2. **Data Quality**
   * Validation rules prevent incorrect refund entry.
   * Duplicate Management Rules → prevent duplicate cases for same Order ID.
3. **Backup & Recovery**
   * Weekly data export via Salesforce Data Export Service.
   * Integrated with AWS S3 for offsite backup storage.
4. **Archiving Strategy**
   * Cases older than 2 years moved to archive object.
   * Archival policy ensures performance optimization.
5. **Data Security**
   * Role-based access (Agent, Finance, Warehouse).
   * Field-level security (refund amount visible only to Finance).
   * Encryption at rest for sensitive payment data.
6. **GDPR & Compliance**
   * Customer request for “Right to be Forgotten” implemented with **Data Deletion Process**.
   * Masking PII fields in reports for non-admins.
7. **ETL Processes**
   * MuleSoft ETL jobs to clean order data from legacy systems.
   * Standardized refund modes into a consistent picklist.

**Deliverables**

* **Data Migration Logs**
* **Backup & Archival Policy Document**
* **Data Governance Report**

Name- Shubhangi Vishwakarma

College- Gyan Ganga Institute of Technology and Sciences

Email- shubhangi.vishwakarma.cs22@ggits.net

TCS Phase 2 : Batch 5