**Smart College CRM – Student Management System on Salesforce**

**Phase 2: Org Setup & Configuration**

**Goal:** Prepare the Salesforce environment through setup and configuration to support the Student Management System, including admission, course enrollment, and fee tracking.

**1. Salesforce Edition**

* **Edition Used:** Salesforce Developer Edition (Dev Org) – free, full-featured org for testing and customization.
* **Rationale:** Suitable for project prototyping, custom objects, and testing before production.

**2. Company Profile Setup**

* **Company Name:** Smart College CRM
* **Default Locale:** English (India)
* **Currency:** Indian Rupee (INR) primary, USD enabled if needed for international courses/partners.
* **Time Zone:** Asia/Kolkata (IST)
* **Business Address:** College HQ address (configurable)  
   Configured organization name, locale, currency, and timezone.

**3. Business Hours & Holidays**

* **Standard Business Hours:** 9:00 AM – 5:00 PM IST (Monday–Friday)
* **Holidays Configured:** Republic Day (26 Jan), Independence Day (15 Aug), Gandhi Jayanti (2 Oct), Diwali, Summer Vacation.
* **Purpose:** Ensures automated processes, approvals, and notifications respect working days.

**4. Fiscal Year Settings**

* **Type:** Standard Fiscal Year (Jan–Dec)
* **Reason:** Aligns with academic year reporting and fee collection cycles.

**5. User Setup & Licenses**

* **User Types Configured:**
  1. **Student User (Community/Experience Cloud License):** Access to personal profile and course/fee info.
  2. **Admission Officer (Salesforce Platform License):** Manage applications and enrollment.
  3. **Finance Officer (Salesforce Platform License):** Manage fee collection and payment tracking.
  4. **Faculty/Admin User (Salesforce License):** Manage courses, assignments, and dashboards.
  5. **System Admin (Salesforce License):** Manage overall Salesforce org.

**6. Profiles**

* **Student Profile:** Limited access – view own data, course enrollment, fee status.
* **Admission Profile:** Create/manage applications and admissions.
* **Finance Profile:** Manage student fees and generate finance reports.
* **Faculty/Admin Profile:** Manage courses and student progress.
* **System Administrator:** Full access to all objects, settings, and reports.

**7. Roles**

* **Hierarchy Setup:**
  + System Admin (Top)
  + College Admin / Head of Departments
  + Admission Officer
  + Finance Officer
  + Faculty
  + Students  
    Ensures role hierarchy visibility (e.g., Admission Officers see all student applications, Students see only their own records).

**8. Permission Sets**

* **Additional Access via Permission Sets:**
  + **Fee Reporting Access:** For Finance Officers to create/modify reports.
  + **Course Management Access:** For Faculty/Admin to manage courses.
  + **Admission Overrides:** For Admission Officers to handle special cases.  
     Assigned permission sets according to role responsibilities.

**9. Org-Wide Defaults (OWD)**

* **Student Records:** Private – visible only to student and relevant staff.
* **Application Records:** Private – managed by Admission Officers and Admins.
* **Fee Records:** Controlled by Parent (linked to Student).
* **Course Records:** Public Read Only – all users can see available courses.

**10. Sharing Rules**

* **Admission Officers:** Access all applications.
* **Finance Officers:** Access all fee records.
* **Faculty/Admin:** Access all course and student enrollment data.

**11. Login Access Policies**

* **Students:** Allowed login 6 AM – 11 PM IST.
* **Staff/Admin:** Enforce IP restrictions and 2FA (Multi-Factor Authentication).
* **Purpose:** Ensures security and controlled access.

**12. Dev Org Setup**

* Created free Salesforce Developer Org for baseline.
* Installed VS Code + Salesforce CLI (SFDX) for development and deployment.
* Connected org to GitHub repository for version control.

**13. Sandbox Usage**

* Developer Sandbox: Testing and configuration changes.
* Full / Partial Sandbox: For User Acceptance Testing (UAT) and training.

**14. Deployment Basics**

* Configurations and custom objects built in Dev Org/Sandbox.
* Deployment to Production via Change Sets (Profiles, Roles, Objects, Validation Rules).
* Version control via GitHub (for Apex classes, Lightning components, and metadata).

**Phase 2 Deliverable**

By the end of Phase 2:

* Org setup is complete with users, profiles, roles, and permission sets.
* Security policies and login rules configured.
* OWD and sharing rules set to maintain proper data access.
* Sandbox and Dev Org ready for development of objects, fields, and automation in Phase 3.