Smart College CRM – Student Management System on Salesforce

Phase 6: User Interface Development (Admin & Lightning Components)

**Goal: To design and build an interactive, user-friendly interface for Smart College CRM that allows students, faculty, and administrative staff to manage admissions, courses, fees, and notifications efficiently using Salesforce Lightning Experience and Lightning Web Components (LWC).**

**1. Lightning App Builder**

**Purpose:** Create dashboards for different personas with relevant information and actions.

**Dashboards:**

* **Student Dashboard:** Shows enrolled courses, fee status, admission status, and notifications.
* **Faculty Dashboard:** Displays courses assigned, enrolled students, and attendance alerts.
* **Admin Dashboard:** Provides system-wide analytics on admissions, fees, course enrollments, and student performance.

**Implementation Steps:**

1. Use **Lightning App Builder** to create App Pages for each persona.
2. Drag-and-drop standard components: Report Charts, List Views, and Related Lists.
3. Insert custom LWCs as placeholders: StudentSummaryCard, FeeList, CourseEnrollmentForm.
4. Activate per App/Profile so that each persona sees their respective dashboard.

**2. Record Pages**

**Customized record pages for key objects:**

* **Student\_\_c Record Page:** Displays personal info, enrollment status, fee status, and course list.
* **Course\_\_c Record Page:** Shows course details, enrolled students, and faculty.
* **Fee\_\_c Record Page:** Displays payment history, status, and related student.

**Layout Example (Student Record Page):**

* **Left Panel:** Student details (name, contact, admission status).
* **Middle Panel:** Courses enrolled, fee status, and recent activity.
* **Tabs:** Courses (related list), Fees, Notifications.
* **Right Panel:** Activity timeline and advisor notes.

**3. Tabs**

Provide structured navigation:

* **Student Tabs:** Dashboard, My Courses, Fee Details, Notifications.
* **Faculty Tabs:** My Courses, Student List, Attendance, Notifications.
* **Admin Tabs:** All Students, All Courses, Fee Reports, System Analytics.

**4. Home Page Layouts**

* **Student Home Page:** Shows personalized greeting, upcoming course notifications, pending fee reminders, and quick links for enrollment.
* **Faculty Home Page:** Displays course schedules, student lists, attendance tracking, and upcoming events.
* **Admin Home Page:** Analytics charts (admissions, course enrollments, fees collected) and approval workflows.

**5. Utility Bar**

* **Quick Search:** Students, Courses, Fees.
* **Quick Add Student:** Admin-only LWC for adding new student records.
* **Help/FAQ Chatbot:** Guides students and faculty.
* **Recent Items:** Quick access to frequently used records.

**6. Lightning Web Components (LWC)**

**Key LWCs for the project:**

* **StudentSummaryCard:** Displays student info, fee status, and courses.
* **FeeList:** Shows all fees with status (Paid, Pending, Overdue).
* **CourseEnrollmentForm:** Allows students to enroll in courses online.
* **NotificationPanel:** Shows real-time alerts for students, faculty, and admins.
* **AttendanceScanner:** Faculty LWC for marking attendance via QR code or student ID.

**Each LWC uses Apex controllers** to fetch and update data dynamically.

**7. Apex with LWC**

**Example Use Case:** Fetch student fees dynamically.

public with sharing class FeeController {

@AuraEnabled(cacheable=true)

public static List<Fee\_\_c> getFeesByStudent(Id studentId){

return [SELECT Id, Amount\_\_c, Status\_\_c, Due\_Date\_\_c FROM Fee\_\_c WHERE Student\_\_c = :studentId];

}

}

import { LightningElement, wire, api, track } from 'lwc';

import getFeesByStudent from '@salesforce/apex/FeeController.getFeesByStudent';

export default class FeeList extends LightningElement {

@api studentId;

@track fees;

@wire(getFeesByStudent, { studentId: '$studentId' })

wiredFees({ error, data }){

if(data){

this.fees = data;

} else if(error){

console.error(error);

}

}

}

**8. Navigation Service**

* **Example:** Student clicks “View Course” → Redirects to **Course\_\_c Record Page**.

this[NavigationMixin.Navigate]({

type: 'standard\_\_recordPage',

attributes: {

recordId: courseId,

objectApiName: 'Course\_\_c',

actionName: 'view'

}

});

**Outcome of Phase 6 :**

User-friendly dashboards tailored for Students, Faculty, and Admins.  
Optimized record pages for Student, Course, and Fee management.  
LWCs enable dynamic interaction (fee list, course enrollment, attendance scanner).  
Navigation & Utility Bar improve usability and efficiency for all users