

Student Performance Tracker

Problem Statement

In many educational institutions, maintaining comprehensive records of student performance—including academic scores, attendance, and extracurricular activities—is often done manually or across multiple disconnected systems.

This leads to delayed reporting, errors in attendance calculation, lack of real-time insights into student performance, and ineffective communication with parents.

The problem:

- Tracking student attendance and academic performance is time-consuming and error-prone.
- Parents often receive delayed information about their child's performance.
- Teachers and administrators lack a unified platform to manage students' overall progress efficiently.

Solution:

Develop a Salesforce-based Student Performance Tracker that integrates student data management, attendance calculation, academic monitoring, and parent communication into a single, easy-to-use application.

This system will automate attendance updates, calculate percentages, notify parents in real-time for failed subjects, and maintain a holistic view of student performance including extracurricular activities.

Phase 1 – Problem Understanding & Industry Analysis

1. Requirement Gathering

Objective: Understand the needs of the school administration, teachers, and parents regarding student tracking and performance management.

Key Requirements Identified:

- Maintain student records (personal, contact, class, section).
- Record and calculate attendance automatically based on daily inputs.
- Record academic scores and calculate percentages and pass/fail status.
- Notify parents when a student fails an exam.
- Track extracurricular activity participation.
- Provide a consolidated dashboard for administrators/teachers.

2. Stakeholder Analysis

Stakeholder	Role	Needs / Concerns
School Administrators	Manage student records & reports	Easy-to-access dashboard, real-time updates, accurate records
Teachers	Record attendance & grades	Efficient data entry, automatic calculations, notifications
Parents	Monitor child performance	Timely alerts, detailed exam & attendance reports
Students	View performance	Transparency and visibility of progress

3. Business Process Mapping

Current Process (Manual / Fragmented):

Attendance recorded manually → Calculations done offline → Academic results recorded separately → Parent communication via calls or emails

Proposed Salesforce Process:

- Teacher records attendance → Student object auto-updates Total_Working_days__c and Number_of_Days_Present__c.
- Teacher records academic scores → Academic object auto-calculates percentage and status.
- Record-triggered flow automatically sends email notifications to parents if percentage < 35%.
- All student data (attendance, academics, extracurricular) consolidated in a Lightning component dashboard.

4. Industry-specific Use Case Analysis

Educational Institutions: Require digital transformation for record-keeping and parent communication.

Use Case: Automate performance tracking and communication in schools to improve accuracy and efficiency.

Competitive Advantage: Real-time updates, automation of repetitive tasks, consolidated view of student data, and easy accessibility via Salesforce platform.

5. AppExchange Exploration

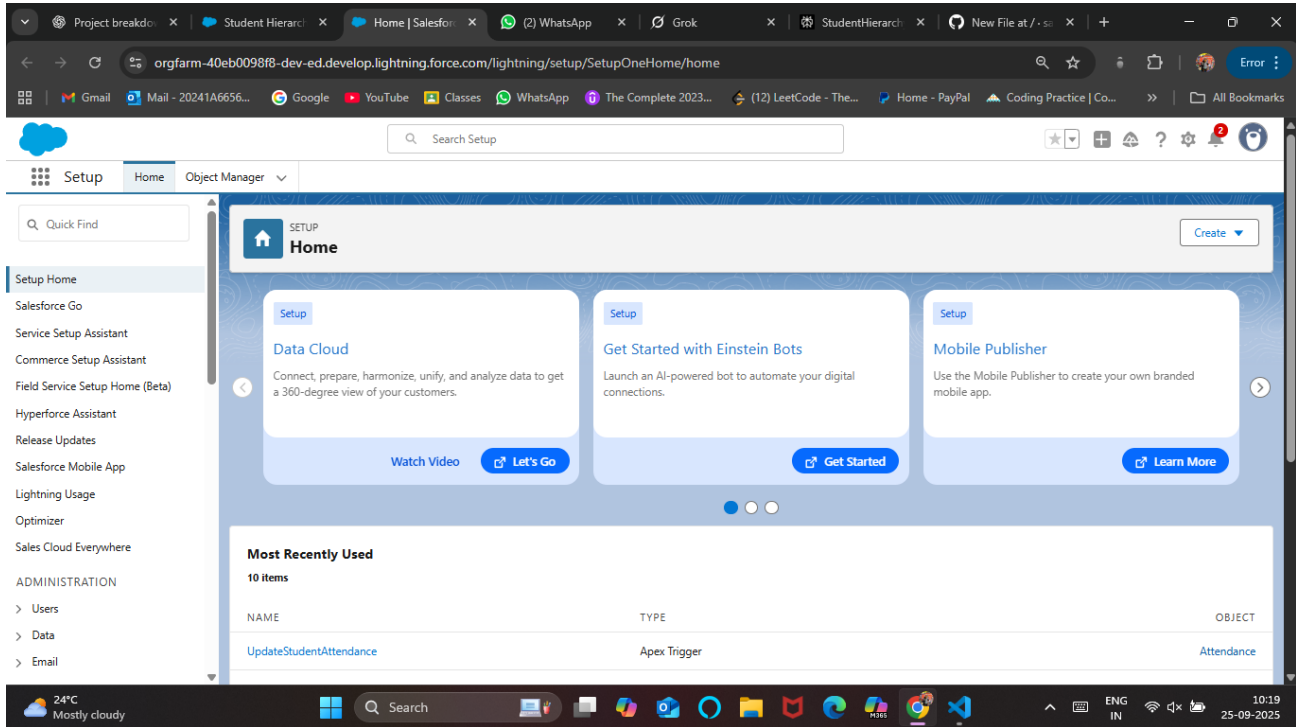
Explored existing Salesforce Education apps such as Education Data Architecture, School Management Apps, and Parent Communication Tools.

Identified gaps: Existing apps often don't integrate attendance, academic, and extracurricular activities together with automated notifications.

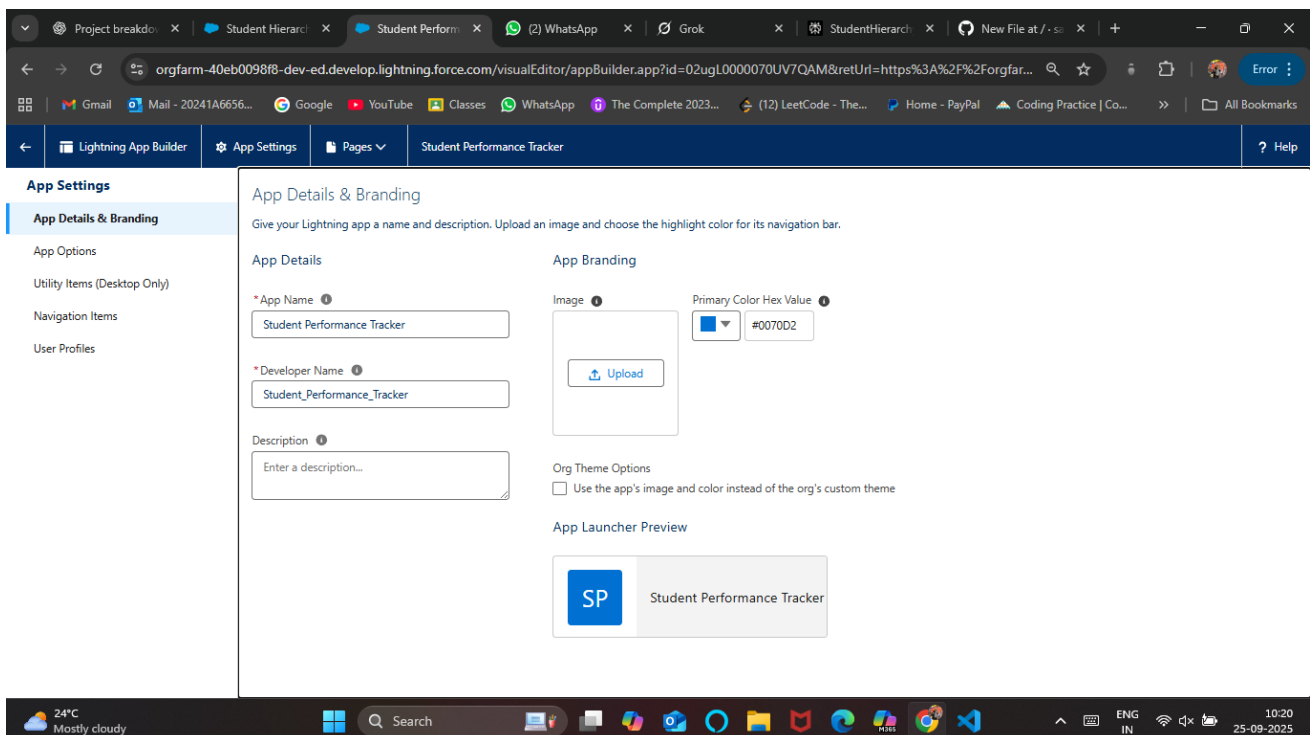
This project fills the gap with a custom-built Salesforce solution tailored for end-to-end student performance tracking.

Phase 2 – Org Setup & Configuration

- Set up a Salesforce Developer Org.
- Configured Lightning App settings, navigation items, and user profiles.
- Prepared the Salesforce environment for development and testing.



Developer Org Setup



Lightning App Creation

Phase 3 – Data Modeling & Relationships

- Created four custom objects: Student, Attendance, Academic, and Extracurricular Activity.
- Defined fields and relationships, including lookup relationships between Student and related objects.
- Used Schema Builder to visualize object relationships and ensure a structured data model.

The screenshot shows the Salesforce Object Manager interface for the 'Student' object. The 'Fields & Relationships' tab is active, displaying a list of 11 fields. The fields are sorted by Field Label. The table includes columns for Field Label, Field Name, Data Type, Controlling Field, and Indexed. The fields listed are: Attendance Percentage, Class, Contact Number, Created By, Date of Birth, Last Modified By, Number of Days Present, Owner, Parent Email, Section, Student ID, and Student Name. The 'Student ID' field is highlighted as the primary key.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Attendance Percentage	Attendance_Percentage__c	Formula (Percent)		
Class	Class__c	Picklist		
Contact Number	Contact_Number__c	Phone		
Created By	CreatedById	Lookup(User)		
Date of Birth	Date_of_Birth__c	Date		
Last Modified By	LastModifiedById	Lookup(User)		
Number of Days Present	Number_of_Days_Present__c	Number(10, 0)		
Owner	OwnerId	Lookup(User Group)		✓
Parent Email	Parent_Email__c	Email		
Section	Section__c	Picklist		
Student ID	Student_ID__c	Text(10)		✓
Student Name	Name	Text(80)		✓

Student Object

The screenshot shows the Salesforce Object Manager interface for the 'Academic' object. The 'Fields & Relationships' tab is active, displaying a list of 10 fields. The fields are sorted by Field Label. The table includes columns for Field Label, Field Name, Data Type, Controlling Field, and Indexed. The fields listed are: Created By, Exam, Last Modified By, Marks, Owner, Percentage, Status, Student ID, Subject Name, and Total Marks. The 'Student ID' field is highlighted as the primary key.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Exam	Exam__c	Text(20)		
Last Modified By	LastModifiedById	Lookup(User)		
Marks	Marks__c	Number(3, 0)		
Owner	OwnerId	Lookup(User Group)		✓
Percentage	Percentage__c	Formula (Number)		
Status	Status__c	Formula (Text)		
Student ID	Student_ID__c	Lookup(Student)		✓
Subject Name	Name	Text(80)		✓
Total Marks	Total_Marks__c	Number(3, 0)		

Academic Object

Project breakdo... x Student Hierarc... Attendance | Sai... x (2) WhatsApp x Grok x StudentHierarch... x Student-Perform... x + -

orgfarm-40eb0098f8-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01IgL000002xEz/FieldsAndRelationships/view

Search Setup

Setup Home Object Manager

SETUP > OBJECT MANAGER

Attendance

Details

Fields & Relationships
8 Items, Sorted by Field Label

Quick Find New Deleted Fields Field Dependencies Set History Tracking

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Attendance Name	Name	Auto Number		✓
Attendance Status	Attendance_Status__c	Picklist		
Created By	CreatedById	Lookup(User)		
Date	Date__c	Date		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Reason for Absence	Reason_for_Absence__c	Text(30)		
Student ID	Student_ID__c	Lookup(Student)		✓

24°C Light rain Search 10:50 25-09-2025

Attendance Object

Project breakdo... x Student Hierarc... Extracurricular A... x (2) WhatsApp x Grok x StudentHierarch... x Student-Perform... x + -

orgfarm-40eb0098f8-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01IgL000002F5f/FieldsAndRelationships/view

Search Setup

Setup Home Object Manager

SETUP > OBJECT MANAGER

Extracurricular Activity

Details

Fields & Relationships
7 Items, Sorted by Field Label

Quick Find New Deleted Fields Field Dependencies Set History Tracking

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Achievement	Achievement__c	Text(30)		
Activity Name	Name	Text(80)		✓
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Participation Date	Participation_Date__c	Date		
Student ID	Student_ID__c	Lookup(Student)		✓

24°C Light rain Search 10:50 25-09-2025

Extracurricular Activity Object

Project breakdo... x Student Hierarc... x Home | Salesfor... x (2) WhatsApp x Grok x StudentHierarc... x Student-Perform... x +

orgfarm-40eb0098f8-dev-ed.develop.lightning.force.com/lightning/setup/SchemaBuilder/home

Search Setup

Setup Home Object Manager

Schema Builder

Close Auto-Layout View Options

Elements Objects

Select objects to display on the builder.

Select from: All Objects

Quick Find...

Select All Clear All

- ☒ Academic
- ☐ Account
- ☐ Activation Target
- ☐ Activity
- ☐ Address
- ☐ Alternative Payment Method
- ☐ Appointment Category
- ☐ Appointment Invitation
- ☐ Appointment Invitee
- ☐ Appointment Topic Time Slot
- ☐ Approval Submission
- ☐ Approval Submission Detail

Attendance

Academic

Extracurricular Activity

Student

Legend

- Lookup Relationship
- Master-Detail Relationship
- Required Field

NIFTY -0.18%

Search

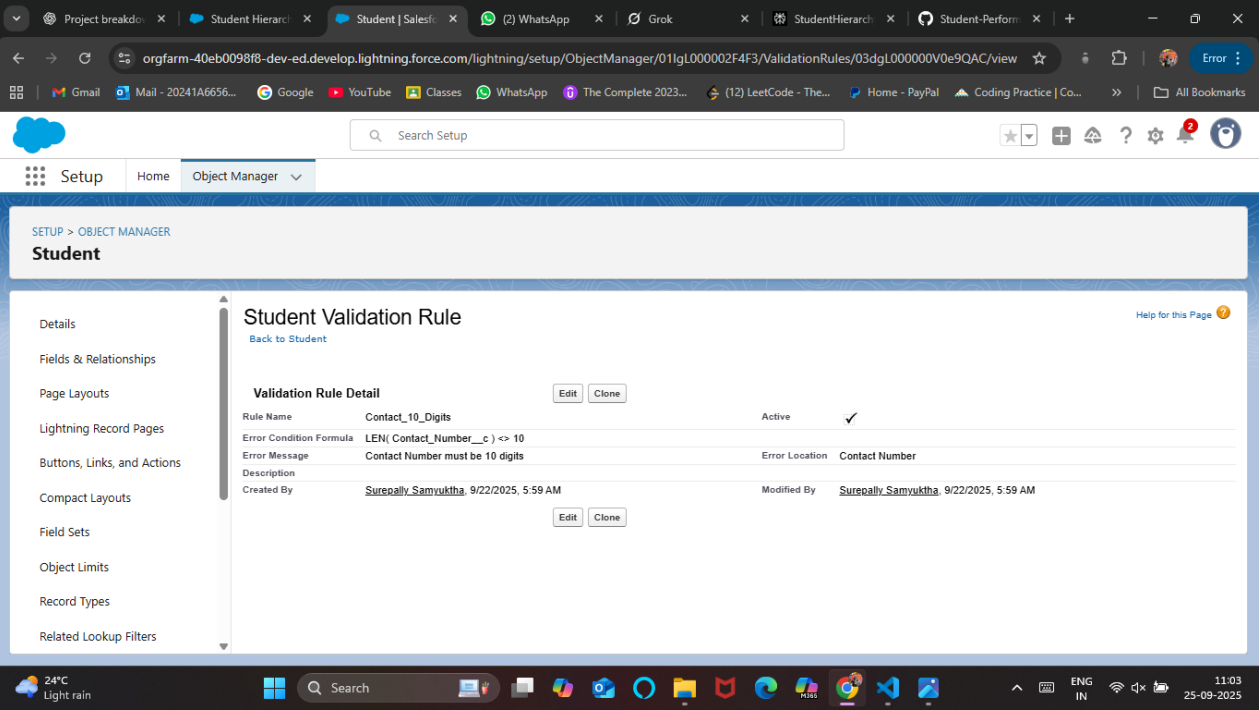
ENG IN

10:58 25-09-2025

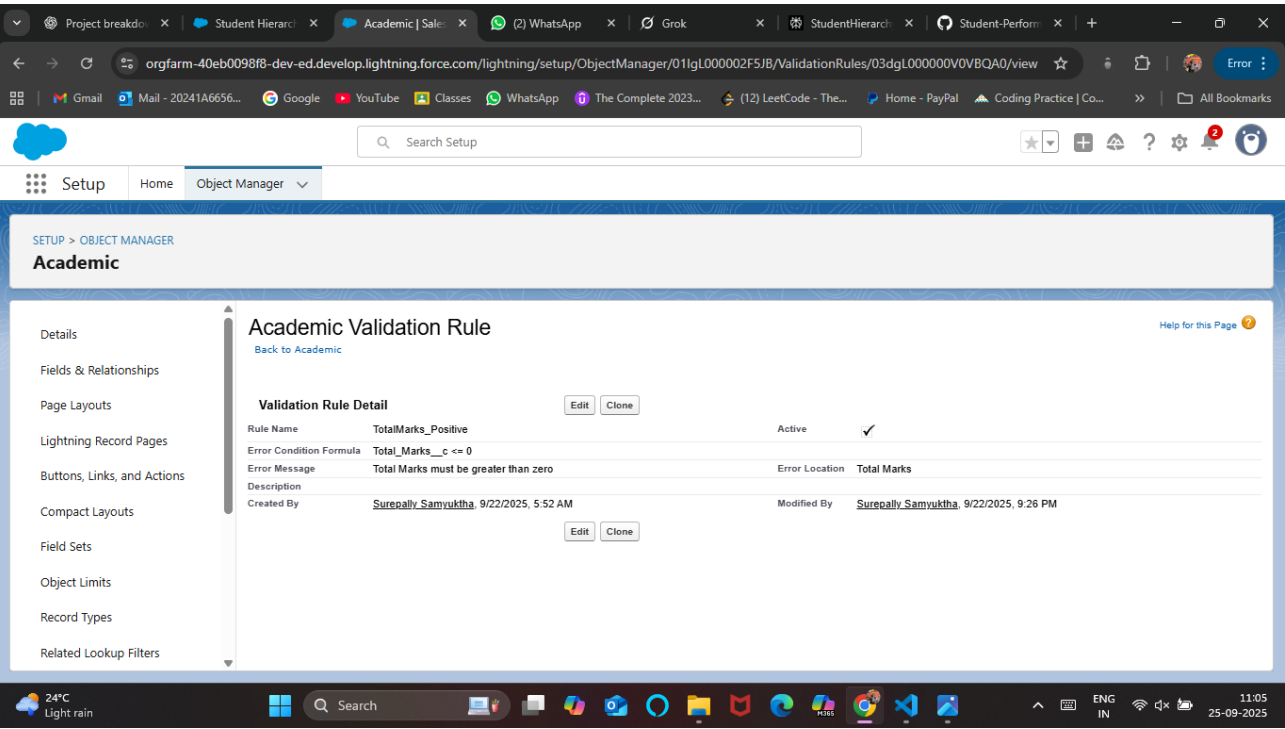
Schema Builder

Phase 4 – Process Automation (Admin)

- Implemented Validation Rules to ensure data integrity across all objects.
- Built Record-Triggered Flows to automate calculations and notifications (e.g., sending alerts when academic percentage < 35%).



Student Validation Rule



Academic Validation Rule - 1

Project breakdo x Student Hierarc x Academic | Sale x (2) WhatsApp x Grok x StudentHierarch x Student-Perform x +

orgfarm-40eb0098f8-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgL000002F5JB/ValidationRules/03dgL000000VCMfQAO/view

Search Setup

Setup Home Object Manager

SETUP > OBJECT MANAGER

Academic

Details Fields & Relationships Page Layouts Lightning Record Pages Buttons, Links, and Actions Compact Layouts Field Sets Object Limits Record Types Related Lookup Filters

Academic Validation Rule

[Back to Academic](#)

Validation Rule Detail Edit Clone

Rule Name	Marks_Less_Than_Total	Active	✓
Error Condition Formula	OR(Marks__c > Total_Marks__c, Marks__c < 0)		
Error Message	Marks cannot be greater than Total Marks Or Marks cannot be negative	Error Location	Marks
Description			
Created By	Surepally Samyuktha, 9/22/2025, 9:25 PM	Modified By	Surepally Samyuktha, 9/22/2025, 9:25 PM

Edit Clone

24°C Light rain Search ENG IN 11:05 25-09-2025

Academic Validation Rule - 2

Project breakdo x Student Hierarc x Extracurricular A x (2) WhatsApp x Grok x StudentHierarch x Student-Perform x +

orgfarm-40eb0098f8-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgL000002F5JB/ValidationRules/03dgL000000V133QAC/view

Search Setup

Setup Home Object Manager

SETUP > OBJECT MANAGER

Extracurricular Activity

Details Fields & Relationships Page Layouts Lightning Record Pages Buttons, Links, and Actions Compact Layouts Field Sets Object Limits Record Types Related Lookup Filters

Extracurricular Activity Validation Rule

[Back to Extracurricular Activity](#)

Validation Rule Detail Edit Clone

Rule Name	ParticipationDate_NotFuture	Active	✓
Error Condition Formula	Participation_Date__c > TODAY()		
Error Message	Participation Date cannot be in the future	Error Location	Participation Date
Description			
Created By	Surepally Samyuktha, 9/22/2025, 6:02 AM	Modified By	Surepally Samyuktha, 9/22/2025, 6:02 AM

Edit Clone

24°C Light rain Search ENG IN 11:06 25-09-2025

Extracurricular Activity Validation Rule

Project breakdo x Student Hierar x Attendance | Sai x (2) WhatsApp x Grok x StudentHierarch x Student-Perform x +

orgfarm-40eb0098f8-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgL000002bEz/ValidationRules/03dgL000000V0WnQAK/view

Search Setup

Setup Home Object Manager

SETUP > OBJECT MANAGER

Attendance

Details

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Attendance Validation Rule

[Back to Attendance](#)

Help for this Page

Validation Rule Detail [Edit](#) [Clone](#)

Rule Name	Reason_Required_If_Absent	Active	<input checked="" type="checkbox"/>
Error Condition Formula	AND(ISPICKVAL(Attendance_Status__c , "Absent"), ISBLANK(Reason_for_Absence__c))		
Error Message	Please enter the Reason for Absence	Error Location	Reason for Absence
Description	Require reason if student is absent		
Created By	Surepally Samyuktha, 9/22/2025, 5:56 AM	Modified By	Surepally Samyuktha, 9/22/2025, 5:56 AM

[Edit](#) [Clone](#)

24°C Light rain

Search

ENG IN

11:06 25-09-2025

Attendance Validation Rule

Project x Studen x Flows x Notify x (2) Wh x Grok x Studen x Studen x Inbox x You Ch x +

orgfarm-40eb0098f8-dev-ed.develop.lightning.force.com/builder_platform_interaction/flowBuilder.app?flowId=301gL00000OuVrQAV

Flow Builder Notify Parent On Fail - V11

Select Elements Auto-Layout Last saved on 9/23/2025, 02:26 PM Active Run Debug View Tests Save As New Version Save Deactivate

Record-Triggered Flow

Start

Object: Academic [Edit](#)

Trigger: A record is created or updated

Conditions: 1

Optimize for: Actions and Related Recor...

[+ Add Scheduled Paths \(Optional\)](#)

[Open Flow Trigger Explorer for Acade...](#)

Run Immediately

Send Notification To Parent Action

End

Send Email

* Label

Send Notification To Parent

* API Name

Send_Notification_To_Parent

Description

Use values from earlier in the flow to set the inputs for the "Send Email" core action. To use its outputs later in the flow, store them in variables.

Set Input Values

Configure Recipient Details

Recipient Address List

...g Academic_c > Student ID > Parent Email

24°C Light rain

Search

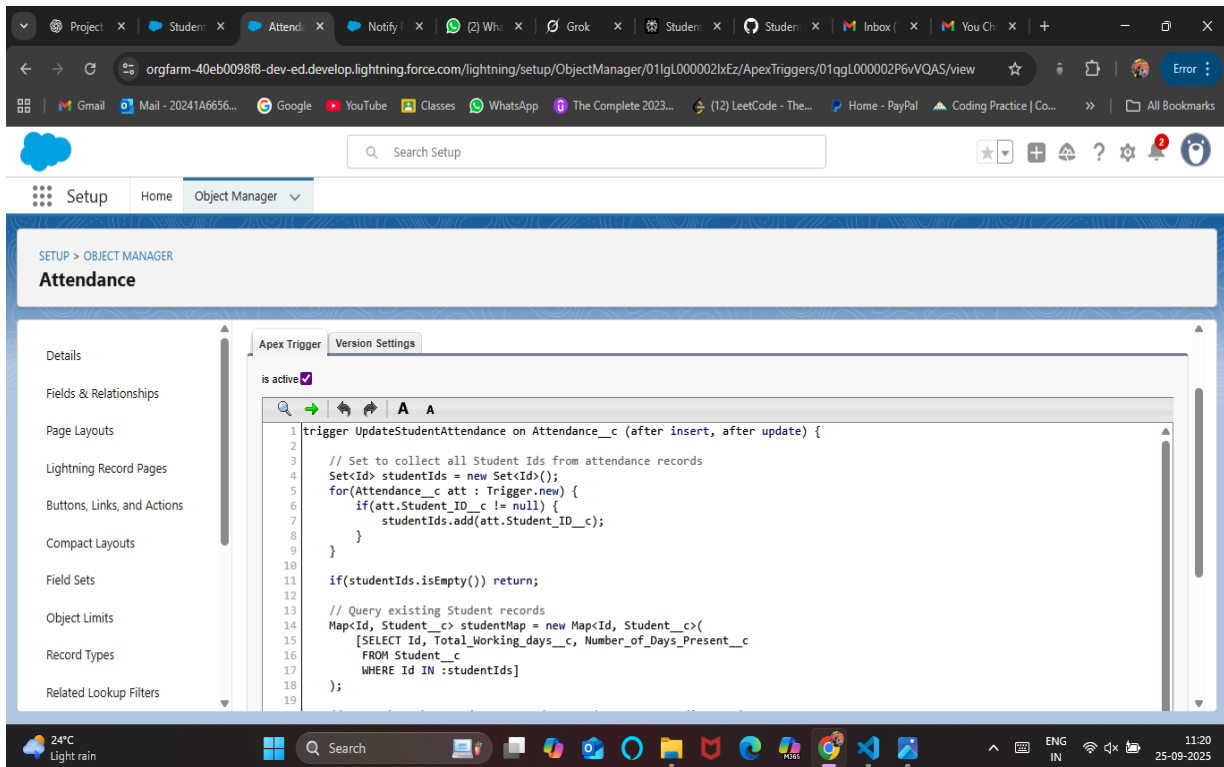
ENG IN

11:12 25-09-2025

Record Triggered Flow to Send Email

Phase 5 – Apex Programming (Developer)

- Developed an Apex Trigger on the Attendance object to update Student records automatically.
- Automated Total Working Days and Number of Days Present calculations.
- Improved data accuracy and reduced manual updates.



Apex Trigger Code :

```
trigger UpdateStudentAttendance on Attendance__c (after insert, after update) {
```

```
    Set<Id> studentIds = new Set<Id>();
```

```
    for(Attendance__c att : Trigger.new) {
```

```
        if(att.Student_ID__c != null) {
```

```
            studentIds.add(att.Student_ID__c);
```

```
        }
```

```
    }
```

```
    if(studentIds.isEmpty()) return;
```

```
    Map<Id, Student__c> studentMap = new Map<Id, Student__c>{
```

```
        [SELECT Id, Total_Working_days__c, Number_of_Days_Present__c
```

```
        FROM Student__c
```

```
        WHERE Id IN :studentIds]
```

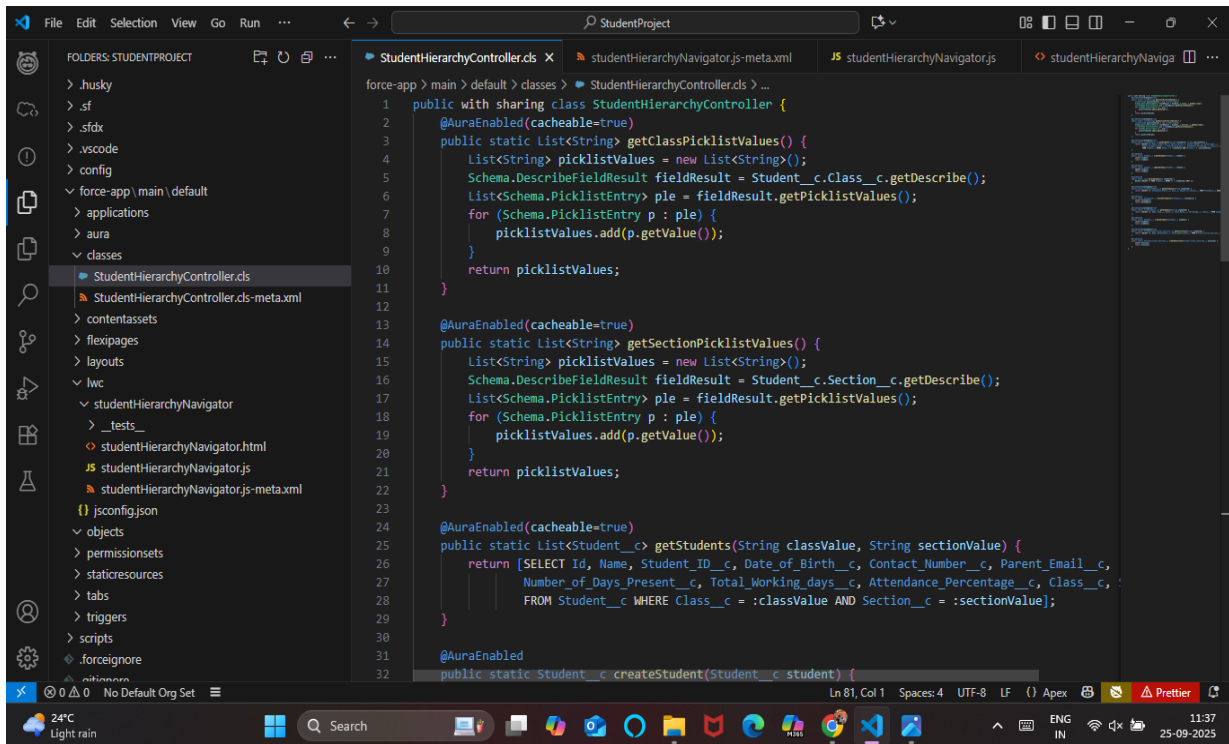
```

);
for(Attendance__c att : Trigger.new) {
    if(att.Student_ID__c != null && studentMap.containsKey(att.Student_ID__c)) {
        Student__c stud = studentMap.get(att.Student_ID__c);
        if(stud.Total_Working_days__c == null) stud.Total_Working_days__c = 0;
        if(stud.Number_of_Days_Present__c == null) stud.Number_of_Days_Present__c = 0
        stud.Total_Working_days__c += 1;
        if(att.Attendance_Status__c == 'Present') {
            stud.Number_of_Days_Present__c += 1;
        }
    }
}
if(!studentMap.isEmpty()) {
    update studentMap.values();
}
}

```

Phase 6 – User Interface Development

- Used Visual Studio Code (VS Code) for Salesforce development and deployment.
- Developed an Apex Class to fetch and process student-related data for the frontend.
- Built Lightning Web Components (LWC) with custom HTML and JavaScript to display student information interactively.
- Integrated backend logic with frontend to deliver real-time dashboards.



Lightning Web Component

StudentHierarchyController.cls

```

public with sharing class StudentHierarchyController {

    @AuraEnabled(cacheable=true)

    public static List<String> getClassPicklistValues() {

        List<String> picklistValues = new List<String>();

        Schema.DescribeFieldResult fieldResult = Student__c.Class__c.getDescribe();

        List<Schema.PicklistEntry> ple = fieldResult.getPicklistValues();

        for (Schema.PicklistEntry p : ple) {

            picklistValues.add(p.getValue());

        }

        return picklistValues;

    }

    @AuraEnabled(cacheable=true)

    public static List<String> getSectionPicklistValues() {

        List<String> picklistValues = new List<String>();

        Schema.DescribeFieldResult fieldResult = Student__c.Section__c.getDescribe();

        List<Schema.PicklistEntry> ple = fieldResult.getPicklistValues();

```

```

    for (Schema.PicklistEntry p : ple) {
        picklistValues.add(p.getValue());
    }
    return picklistValues;
}

@AuraEnabled(cacheable=true)
public static List<Student__c> getStudents(String classValue, String sectionValue) {
    return [SELECT Id, Name, Student_ID__c, Date_of_Birth__c, Contact_Number__c, Parent_Email__c,
        Number_of_Days_Present__c, Total_Working_days__c, Attendance_Percentage__c, Class__c,
        Section__c
        FROM Student__c WHERE Class__c = :classValue AND Section__c = :sectionValue];
}

@AuraEnabled
public static Student__c createStudent(Student__c student) {
    insert student;
    return student;
}

@AuraEnabled
public static Student__c updateStudent(Student__c student) {
    update student;
    return student;
}

@AuraEnabled
public static void deleteStudent(Id studentId) {
    delete [SELECT Id FROM Student__c WHERE Id = :studentId LIMIT 1];
}

@AuraEnabled(cacheable=true)
public static List<Attendance__c> getAttendance(String studentId) {
    return [SELECT Id, Attendance_Status__c, Date__c, Reason_for_Absence__c FROM Attendance__c
    WHERE Student_ID__c = :studentId];
}

@AuraEnabled

```

```

public static Attendance__c createAttendance(Attendance__c attendance) {
    insert attendance;
    return attendance;
}

@AuraEnabled(cacheable=true)
public static List<Academic__c> getAcademics(String studentId) {
    return [SELECT Id, Name, Exam__c, Marks__c, Total_Marks__c, Percentage__c, Status__c FROM
Academic__c WHERE Student_ID__c = :studentId];
}

@AuraEnabled
public static Academic__c createAcademic(Academic__c academic) {
    insert academic;
    return academic;
}

@AuraEnabled(cacheable=true)
public static List<ExtraCurricular_Activity__c> getActivities(String studentId) {
    return [SELECT Id, Name, Achievement__c, Participation_Date__c FROM ExtraCurricular_Activity__c
WHERE Student_ID__c = :studentId];
}

@AuraEnabled
public static ExtraCurricular_Activity__c createActivity(ExtraCurricular_Activity__c activity) {
    insert activity;
    return activity;
}
}

```

StudentHierarchyController.cls-meta.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<ApexClass xmlns="http://soap.sforce.com/2006/04/metadata">
    <apiVersion>60.0</apiVersion>
    <status>Active</status>
</ApexClass>

```

studentHierarchyNavigator.html

```
<template>

  <lightning-card title="Student Hierarchy Navigator">

    <template if:true={showClasses}>

      <lightning-combobox

        name="class"

        label="Select Class"

        value={selectedClass}

        placeholder="Select a Class"

        options={classOptions}

        onchange={handleClassChange}>

      </lightning-combobox>

    </template>

    <template if:true={showSections}>

      <lightning-combobox

        name="section"

        label="Select Section"

        value={selectedSection}

        placeholder="Select a Section"

        options={sectionOptions}

        onchange={handleSectionChange}>

      </lightning-combobox>

    </template>

    <template if:true={showStudents}>

      <lightning-button label="Add Student" onclick={handleAddStudent}></lightning-button>

      <lightning-button label="Refresh Students" onclick={fetchStudents} class="slds-m-
left_small"></lightning-button>

      <lightning-datatable key-field="Id" data={students} columns={studentColumns}
onrowaction={handleStudentSelect} hide-checkbox-column>

      </lightning-datatable>

    </template>

    <template if:true={showStudentDetails}>

      <lightning-tabset variant="scoped">

        <lightning-tab label="Attendance" value="attendance">
```

```

        <lightning-button label="Add Attendance" onclick={handleAddAttendance}></lightning-
button>

        <lightning-button label="Refresh Attendance" onclick={fetchAttendance} class="slds-m-
left_small"></lightning-button>

        <lightning-datatable key-field="Id" data={attendanceRecords}
columns={attendanceColumns} hide-checkbox-column></lightning-datatable>

    </lightning-tab>

    <lightning-tab label="Academics" value="academics">

        <lightning-button label="Add Academic" onclick={handleAddAcademic}></lightning-
button>

        <lightning-button label="Refresh Academics" onclick={fetchAcademics} class="slds-m-
left_small"></lightning-button>

        <lightning-datatable key-field="Id" data={academicRecords} columns={academicColumns}
hide-checkbox-column></lightning-datatable>

    </lightning-tab>

    <lightning-tab label="Activities" value="activities">

        <lightning-button label="Add Activity" onclick={handleAddActivity}></lightning-button>

        <lightning-button label="Refresh Activities" onclick={fetchActivities} class="slds-m-
left_small"></lightning-button>

        <lightning-datatable key-field="Id" data={activityRecords} columns={activityColumns}
hide-checkbox-column></lightning-datatable>

    </lightning-tab>

</lightning-tabset>

</template>

<!-- Modal for adding student -->

<template if:true={showAddStudentModal}>

    <section role="dialog" tabindex="-1" aria-modal="true" class="slds-modal slds-fade-in-open">

        <div class="slds-modal__container">

            <header class="slds-modal__header">

                <h2 class="slds-modal__title">Add Student</h2>

            </header>

            <div class="slds-modal__content slds-p-around_medium">

                <lightning-record-edit-form object-api-name="Student__c"
onsuccess={handleStudentSave}>

                    <lightning-messages></lightning-messages>

                    <lightning-input-field field-name="Name"></lightning-input-field>

```



```

        <lightning-input-field field-name="Student_ID__c"></lightning-input-field>
        <lightning-input-field field-name="Date_of_Birth__c"></lightning-input-field>
        <lightning-input-field field-name="Contact_Number__c"></lightning-input-field>
        <lightning-input-field field-name="Parent_Email__c"></lightning-input-field>
        <lightning-input-field field-name="Number_of_Days_Present__c"></lightning-input-
field>

        <lightning-input-field field-name="Total_Working_days__c"></lightning-input-field>
        <!-- Class and Section are pre-selected -->
        <lightning-input type="hidden" name="Class__c" value={selectedClass}></lightning-
input>

        <lightning-input type="hidden" name="Section__c"
value={selectedSection}></lightning-input>

        <lightning-button type="submit" label="Save"></lightning-button>

    </lightning-record-edit-form>

</div>

<footer class="slds-modal__footer">

    <lightning-button label="Cancel" onclick={closeModal}></lightning-button>

</footer>

</div>

</section>

<div class="slds-backdrop slds-backdrop_open"></div>

</template>

<!-- Modal for editing student -->

<template if:true={showEditStudentModal}>

    <section role="dialog" tabindex="-1" aria-modal="true" class="slds-modal slds-fade-in-open">

        <div class="slds-modal__container">

            <header class="slds-modal__header">

                <h2 class="slds-modal__title">Edit Student</h2>

            </header>

            <div class="slds-modal__content slds-p-around_medium">

                <lightning-record-edit-form object-api-name="Student__c" record-
id={editStudentData.Id} onSuccess={handleEditStudentSave}>

                    <lightning-messages></lightning-messages>

                    <lightning-input-field field-name="Name"></lightning-input-field>

```

```

        <lightning-input-field field-name="Student_ID__c"></lightning-input-field>
        <lightning-input-field field-name="Date_of_Birth__c"></lightning-input-field>
        <lightning-input-field field-name="Contact_Number__c"></lightning-input-field>
        <lightning-input-field field-name="Parent_Email__c"></lightning-input-field>
        <lightning-input-field field-name="Number_of_Days_Present__c"></lightning-input-
field>

        <lightning-input-field field-name="Total_Working_days__c"></lightning-input-field>
        <lightning-input-field field-name="Class__c"></lightning-input-field>
        <lightning-input-field field-name="Section__c"></lightning-input-field>
        <lightning-button type="submit" label="Save"></lightning-button>

    </lightning-record-edit-form>

</div>

<footer class="slds-modal__footer">
    <lightning-button label="Cancel" onclick={closeModal}></lightning-button>
</footer>

</div>

</section>

<div class="slds-backdrop slds-backdrop_open"></div>

</template>

<!-- Modal for adding attendance -->

<template if:true={showAddAttendanceModal}>
    <section role="dialog" tabindex="-1" aria-modal="true" class="slds-modal slds-fade-in-open">
        <div class="slds-modal__container">
            <header class="slds-modal__header">
                <h2 class="slds-modal__title">Add Attendance</h2>
            </header>
            <div class="slds-modal__content slds-p-around_medium">
                <lightning-record-edit-form object-api-name="Attendance__c"
onsuccess={handleAttendanceSave}>
                    <lightning-messages></lightning-messages>
                    <lightning-input-field field-name="Attendance_Status__c"></lightning-input-field>
                    <lightning-input-field field-name="Date__c"></lightning-input-field>
                    <lightning-input-field field-name="Reason_for_Absence__c"></lightning-input-field>

```

```

        <lightning-input-field field-name="Student_ID__c"
value={selectedStudentId}></lightning-input-field>

        <lightning-button type="submit" label="Save"></lightning-button>

    </lightning-record-edit-form>

</div>

<footer class="slds-modal__footer">

    <lightning-button label="Cancel" onclick={closeModal}></lightning-button>

</footer>

</div>

</section>

<div class="slds-backdrop slds-backdrop_open"></div>

</template>

<!-- Modal for adding academic -->

<template if:true={showAddAcademicModal}>

    <section role="dialog" tabindex="-1" aria-modal="true" class="slds-modal slds-fade-in-open">

        <div class="slds-modal__container">

            <header class="slds-modal__header">

                <h2 class="slds-modal__title">Add Academic</h2>

            </header>

            <div class="slds-modal__content slds-p-around_medium">

                <lightning-record-edit-form object-api-name="Academic__c"
onsuccess={handleAcademicSave}>

                    <lightning-messages></lightning-messages>

                    <lightning-input-field field-name="Name"></lightning-input-field>

                    <lightning-input-field field-name="Exam__c"></lightning-input-field>

                    <lightning-input-field field-name="Marks__c"></lightning-input-field>

                    <lightning-input-field field-name="Total_Marks__c"></lightning-input-field>

                    <lightning-input-field field-name="Student_ID__c"
value={selectedStudentId}></lightning-input-field>

                    <lightning-button type="submit" label="Save"></lightning-button>

                </lightning-record-edit-form>

            </div>

            <footer class="slds-modal__footer">

                <lightning-button label="Cancel" onclick={closeModal}></lightning-button>

```

```

        </footer>

    </div>

</section>

<div class="slds-backdrop slds-backdrop_open"></div>

</template>

<!-- Modal for adding activity -->

<template if:true={showAddActivityModal}>

    <section role="dialog" tabindex="-1" aria-modal="true" class="slds-modal slds-fade-in-open">

        <div class="slds-modal__container">

            <header class="slds-modal__header">

                <h2 class="slds-modal__title">Add Activity</h2>

            </header>

            <div class="slds-modal__content slds-p-around_medium">

                <lightning-record-edit-form object-api-name="ExtraCurricular_Activity__c"
onsuccess={handleActivitySave}>

                    <lightning-messages></lightning-messages>

                    <lightning-input-field field-name="Name"></lightning-input-field>

                    <lightning-input-field field-name="Achievement__c"></lightning-input-field>

                    <lightning-input-field field-name="Participation_Date__c"></lightning-input-field>

                    <lightning-input-field field-name="Student_ID__c"
value={selectedStudentId}></lightning-input-field>

                    <lightning-button type="submit" label="Save"></lightning-button>

                </lightning-record-edit-form>

            </div>

            <footer class="slds-modal__footer">

                <lightning-button label="Cancel" onclick={closeModal}></lightning-button>

            </footer>

        </div>

    </section>

    <div class="slds-backdrop slds-backdrop_open"></div>

</template>

</lightning-card>

</template>

```

studentHierarchyNavigator.js

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

import getClassPicklistValues from
'@salesforce/apex/StudentHierarchyController.getClassPicklistValues';

import getSectionPicklistValues from
'@salesforce/apex/StudentHierarchyController.getSectionPicklistValues';

import getStudents from '@salesforce/apex/StudentHierarchyController.getStudents';
import createStudent from '@salesforce/apex/StudentHierarchyController.createStudent';
import updateStudent from '@salesforce/apex/StudentHierarchyController.updateStudent';
import deleteStudent from '@salesforce/apex/StudentHierarchyController.deleteStudent';
import getAttendance from '@salesforce/apex/StudentHierarchyController.getAttendance';
import getAcademics from '@salesforce/apex/StudentHierarchyController.getAcademics';
import getActivities from '@salesforce/apex/StudentHierarchyController.getActivities';
import { ShowToastEvent } from 'lightning/platformShowToastEvent';

export default class StudentHierarchyNavigator extends LightningElement {

    @track showClasses = true;
    @track showSections = false;
    @track showStudents = false;
    @track showStudentDetails = false;
    @track selectedClass;
    @track selectedSection;
    @track selectedStudentId;
    @track classOptions = [];
    @track sectionOptions = [];
    @track students = [];
    @track attendanceRecords = [];
    @track academicRecords = [];
    @track activityRecords = [];
    @track showAddStudentModal = false;
    @track showEditStudentModal = false;
    @track editStudentData = {};
    @track showAddAttendanceModal = false;
    @track showAddAcademicModal = false;
    @track showAddActivityModal = false;
```

```

studentColumns = [
    { label: 'Student Name', fieldName: 'Name' },
    { label: 'Student ID', fieldName: 'Student_ID__c' },
    { label: 'Date of Birth', fieldName: 'Date_of_Birth__c', type: 'date' },
    { label: 'Contact Number', fieldName: 'Contact_Number__c', type: 'phone' },
    { label: 'Parent Email', fieldName: 'Parent_Email__c', type: 'email' },
    { label: 'Days Present', fieldName: 'Number_of_Days_Present__c', type: 'number' },
    { label: 'Total Working Days', fieldName: 'Total_Working_days__c', type: 'number' },
    { label: 'Attendance Percentage', fieldName: 'Attendance_Percentage__c', type: 'percent' },
    {
        type: 'action',
        typeAttributes: {
            rowActions: [
                { label: 'View Details', name: 'view' },
                { label: 'Edit', name: 'edit' },
                { label: 'Delete', name: 'delete' }
            ]
        }
    }
];

```

```

attendanceColumns = [
    { label: 'Attendance Status', fieldName: 'Attendance_Status__c' },
    { label: 'Date', fieldName: 'Date__c', type: 'date' },
    { label: 'Reason for Absence', fieldName: 'Reason_for_Absence__c' }
];

```

```

academicColumns = [
    { label: 'Subject Name', fieldName: 'Name' },
    { label: 'Exam', fieldName: 'Exam__c' },
    { label: 'Marks', fieldName: 'Marks__c', type: 'number' },
    { label: 'Total Marks', fieldName: 'Total_Marks__c', type: 'number' },
    { label: 'Percentage', fieldName: 'Percentage__c', type: 'number' },
    { label: 'Status', fieldName: 'Status__c' }
];

```

```

];

activityColumns = [
    { label: 'Activity Name', fieldName: 'Name' },
    { label: 'Achievement', fieldName: 'Achievement__c' },
    { label: 'Participation Date', fieldName: 'Participation_Date__c', type: 'date' }
];

@wire(getClassPicklistValues)
wiredClasses({ error, data }) {
    if (data) {
        this.classOptions = data.map(value => ({ label: value, value: value }));
    } else if (error) {
        this.showToast('Error', error.body.message, 'error');
    }
}

@wire(getSectionPicklistValues)
wiredSections({ error, data }) {
    if (data) {
        this.sectionOptions = data.map(value => ({ label: value, value: value }));
    } else if (error) {
        this.showToast('Error', error.body.message, 'error');
    }
}

handleClassChange(event) {
    this.selectedClass = event.detail.value;
    this.showSections = true;
    this.showStudents = false;
    this.showStudentDetails = false;
}

handleSectionChange(event) {
    this.selectedSection = event.detail.value;
    this.showStudents = true;
    this.showStudentDetails = false;
    this.fetchStudents();
}

```

```

}

fetchStudents() {
  getStudents({ classValue: this.selectedClass, sectionValue: this.selectedSection })
    .then(result => {
      this.students = result;
    })
    .catch(error => {
      this.showToast('Error', error.body.message, 'error');
    });
}

handleStudentSelect(event) {
  const actionName = event.detail.action.name;
  const row = event.detail.row;
  if (actionName === 'view') {
    this.selectedStudentId = row.Id;
    this.showStudentDetails = true;
    this.fetchStudentDetails();
  } else if (actionName === 'edit') {
    this.editStudentData = { ...row };
    this.showEditStudentModal = true;
  } else if (actionName === 'delete') {
    this.deleteStudent(row.Id);
  }
}

fetchStudentDetails() {
  this.fetchAttendance();
  this.fetchAcademics();
  this.fetchActivities();
}

fetchAttendance() {
  getAttendance({ studentId: this.selectedStudentId })
    .then(result => {
      this.attendanceRecords = result;
    });
}

```



```

    })
    .catch(error => {
        this.showToast('Error', error.body.message, 'error');
    });
}

fetchAcademics() {
    getAcademics({ studentId: this.selectedStudentId })
        .then(result => {
            this.academicRecords = result;
        })
        .catch(error => {
            this.showToast('Error', error.body.message, 'error');
        });
}

fetchActivities() {
    getActivities({ studentId: this.selectedStudentId })
        .then(result => {
            this.activityRecords = result;
        })
        .catch(error => {
            this.showToast('Error', error.body.message, 'error');
        });
}

handleAddStudent() {
    this.showAddStudentModal = true;
}

handleAddAttendance() {
    this.showAddAttendanceModal = true;
}

handleAddAcademic() {
    this.showAddAcademicModal = true;
}

handleAddActivity() {

```

```

    this.showAddActivityModal = true;
}
handleStudentSave(event) {
    this.showAddStudentModal = false;
    this.showToast('Success', 'Student added successfully', 'success');
    this.fetchStudents();
}
handleEditStudentSave(event) {
    const updatedStudent = { ...event.detail.fields, sojectType: 'Student__c' };
    updateStudent({ student: updatedStudent })
        .then(() => {
            this.showToast('Success', 'Student updated successfully', 'success');
            this.fetchStudents();
            this.showEditStudentModal = false;
        })
        .catch(error => {
            this.showToast('Error', error.body.message, 'error');
        });
}
deleteStudent(studentId) {
    deleteStudent({ studentId: studentId })
        .then(() => {
            this.showToast('Success', 'Student deleted successfully', 'success');
            this.fetchStudents();
            if (this.selectedStudentId === studentId) {
                this.showStudentDetails = false;
            }
        })
        .catch(error => {
            this.showToast('Error', error.body.message, 'error');
        });
}
handleAttendanceSave(event) {

```

```

    this.showAddAttendanceModal = false;

    this.showToast('Success', 'Attendance added successfully', 'success');

    this.fetchAttendance();
}

handleAcademicSave(event) {
    this.showAddAcademicModal = false;

    this.showToast('Success', 'Academic added successfully', 'success');

    this.fetchAcademics();
}

handleActivitySave(event) {
    this.showAddActivityModal = false;

    this.showToast('Success', 'Activity added successfully', 'success');

    this.fetchActivities();
}

closeModal() {
    this.showAddStudentModal = false;

    this.showEditStudentModal = false;

    this.showAddAttendanceModal = false;

    this.showAddAcademicModal = false;

    this.showAddActivityModal = false;
}

showToast(title, message, variant) {
    const event = new ShowToastEvent({
        title: title,
        message: message,
        variant: variant
    });

    this.dispatchEvent(event);
}
}

```

studentHierarchyNavigator.js-meta.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">

  <apiVersion>60.0</apiVersion>

  <isExposed>true</isExposed>

  <targets>

    <target>lightning__AppPage</target>

    <target>lightning__RecordPage</target>

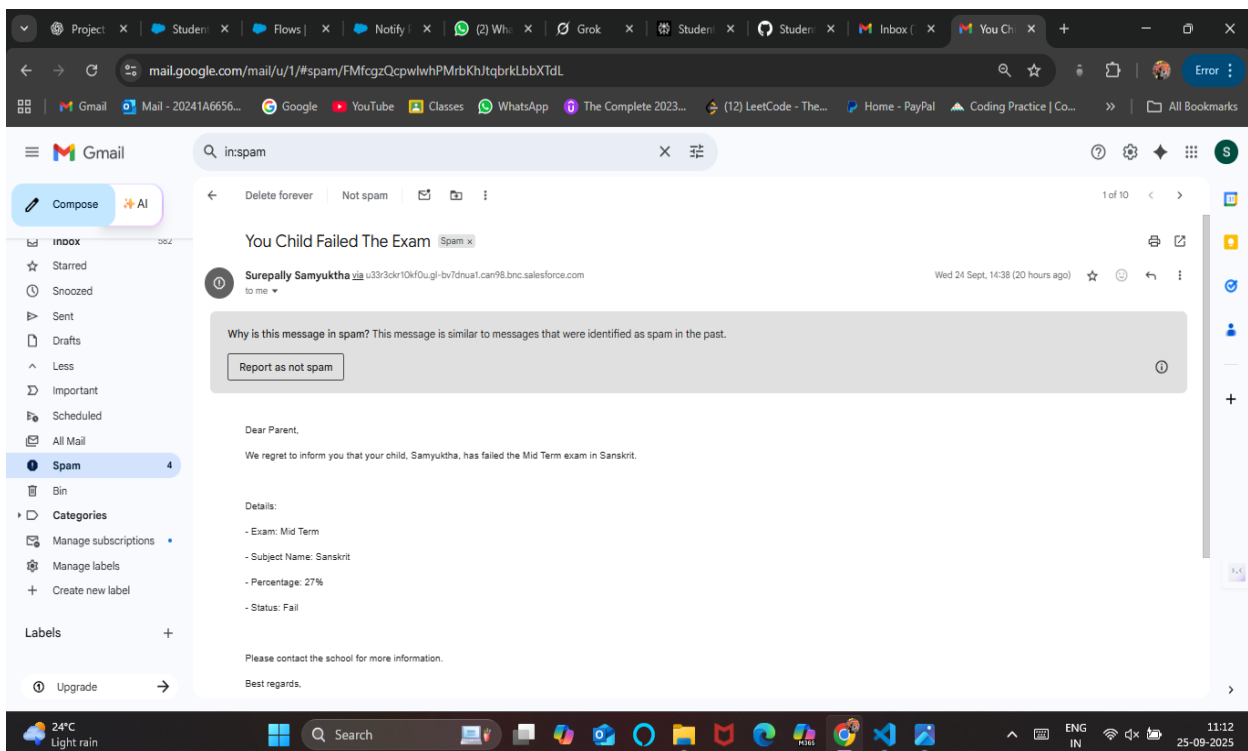
    <target>lightning__HomePage</target>

  </targets>

</LightningComponentBundle>
```

Phase 7 – Integration & External Access

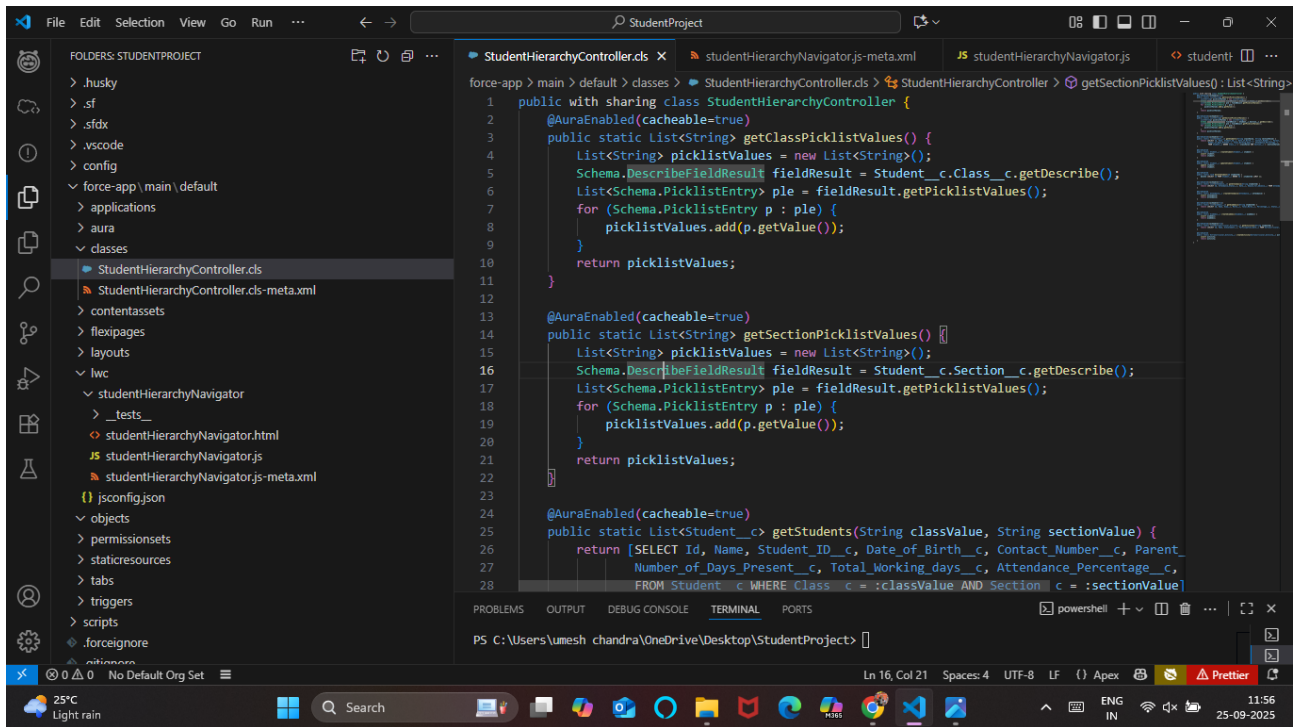
- Created a Record-Triggered Flow to send emails to parents when a student's exam percentage falls below 35%.
- Configured the flow to fetch Parent Email and include exam details in the email content.
- Verified functionality with screenshots of received emails.



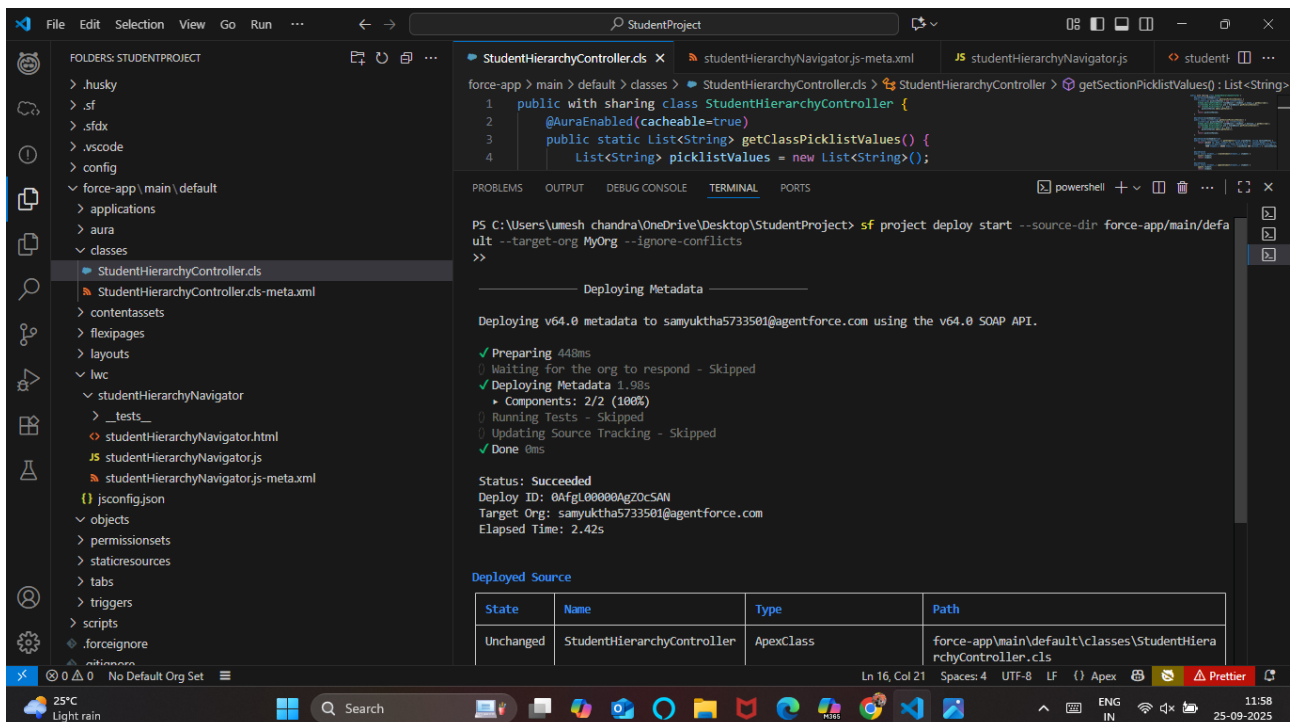
Email Received to Parent

Phase 8 – Data Management & Deployment

- Showcased the Salesforce DX (SFDX) project structure in VS Code.
- Deployed Apex classes and LWCs to the Salesforce Developer Org using SFDX commands.
- Ensured all custom logic and UI components were successfully integrated.



VSCode SFDX Project Structure



Apex Class & LWC Deployment

Phase 9 & 10 – Reporting, Dashboards & Security Review

- Displayed the final Lightning App interface with a complete student management dashboard.
- Showcased student academic, attendance, and activity details within the app.
- Provided forms for adding new records: Student, Attendance, Academic, and Activity.
- Highlighted the interactive, user-friendly interface and seamless navigation across sections.

The screenshot shows the Lightning App interface for Student Hierarchy. The top navigation bar includes a search bar and tabs for Student Performance, Student Hierarchy, and Reports. The Student Hierarchy section displays a table of students with columns for Student Name, Student ID, Date of Birth, Contact Number, Parent Email, Days Present, Total Working Days, and Attendance Percentage. Below this, there are tabs for Attendance, Academics, and Activities. The Academics tab is active, showing a table of subjects with columns for Subject Name, Exam, Marks, Total Marks, Percentage, and Status.

Student Name	Student ID	Date of Birth	Contact Number	Parent Email	Days Present	Total Working Da...	Attendance Perce...
Samyuktha	1	Feb 17, 2015	(912) 345-6789	samyuktha5733@...	1	2	50%
Umesh	2	Sep 15, 2015	(993) 245-6718	samyuktha5733@...	1	1	100%
Pranathi	3	Aug 31, 2015	(943) 218-7658	samyuktha5733@...	1	1	100%
Akshay	4	May 4, 2015	(911) 223-3445	samyuktha180205...	0	1	0%
Venu	5	Apr 15, 2015	(975) 312-4680	samyuktha5733@...	1	1	100%

Subject Name	Exam	Marks	Total Marks	Percentage	Status
English	Mid Term	34	100	34	Fail
Hindi	Mid Term	56	100	56	Pass
Telugu	Mid Term	72	100	72	Pass

App Outlook

The screenshot shows the Lightning App interface with the Add Student form open. The form is a modal dialog with fields for Student Name, Student ID, Date of Birth, Contact Number, Parent Email, and Number of Days Present. The background shows the Student Hierarchy section with a table of students.

Student Name	Student ID	Date of Birth	Contact Number	Parent Email	Days Present	Total Working Da...	Attendance Perce...
Samyuktha	1	Feb 17, 2015	(912) 345-6789	samyuktha5733@...	1	2	50%
Umesh	2	Sep 15, 2015	(993) 245-6718	samyuktha5733@...	1	1	100%
Pranathi	3	Aug 31, 2015	(943) 218-7658	samyuktha5733@...	1	1	100%
Akshay	4	May 4, 2015	(911) 223-3445	samyuktha180205...	0	1	0%
Venu	5	Apr 15, 2015	(975) 312-4680	samyuktha5733@...	1	1	100%

Add Student Form

Project x Student x Attend x Notify x (2) Wh x Grok x Student x Student x Inbox x You Ch x +

orgfarm-40eb0098f8-dev-ed.develop.lightning.force.com/lightning/n/Student_Hierarchy

Student Performance... Student Hierarchy

Add Student Refresh Students

Student Name	Student ID	Date
Samyuktha	1	Feb
Umesh	2	Se
Pranathi	3	Au
Akshay	4	Ma
Venu	5	Ap

Attendance Academics Activities

Add Academic Refresh Academics

Subject Name Exam

Marks

Total Marks

Student ID Umesh

Save Cancel

Subject Name	Exam	Total Marks
English	Mid Term	72
Hindi	Mid Term	100
Telugu	Mid Term	72

24°C Rain showers

Search

ENG IN 12:17 25-09-2025

Add Academic Form

Project x Student x Attend x Notify x (2) Wh x Grok x Student x Student x Inbox x You Ch x +

orgfarm-40eb0098f8-dev-ed.develop.lightning.force.com/lightning/n/Student_Hierarchy

Student Performance... Student Hierarchy

Student Hierarchy Navigator

Select Class 6th

Select Section A

Add Student Refresh Students

Student Name	Student ID	Date
Samyuktha	1	Feb
Umesh	2	Se
Pranathi	3	Au
Akshay	4	Ma
Venu	5	Ap

Attendance Academics Activities

Add Attendance Refresh Attendance

Attendance Status --None--

Date

Reason for Absence

Student ID Umesh

Save Cancel

Attendance Status	Date	Reason for Absence
2		50%
1		100%
1		100%
1		0%
1		100%

24°C Rain showers

Search

ENG IN 12:17 25-09-2025

Add Attendance Form

Project breal x (2) WhatsApp x Grok x StudentHie x Student-Per x Inbox (7,074 x You Child F x Student Hie x

orgfarm-40eb0098f8-dev-ed.develop.lightning.force.com/lightning/n/Student_Hierarchy

Search...

Student Performanc... Student Hiera

Select Class
6th

Select Section
A

Add Student Refresh Students

Student Name	Student ID	Da
Samyuktha	1	Fel
Umesh	2	Se
Pranathi	3	Au
Akshay	4	Ma
Venu	5	Ap

Attendance Academics **Activities**

Add Activity Refresh Activities

Activity Name Achievement Participation Date

Add Activity

Activity Name

Achievement

Participation Date

Student ID
Samyuktha

Save

Cancel

24°C Rain showers

Search

ENG IN

12:29 25-09-2025

Add Activity Form