

Phase 6

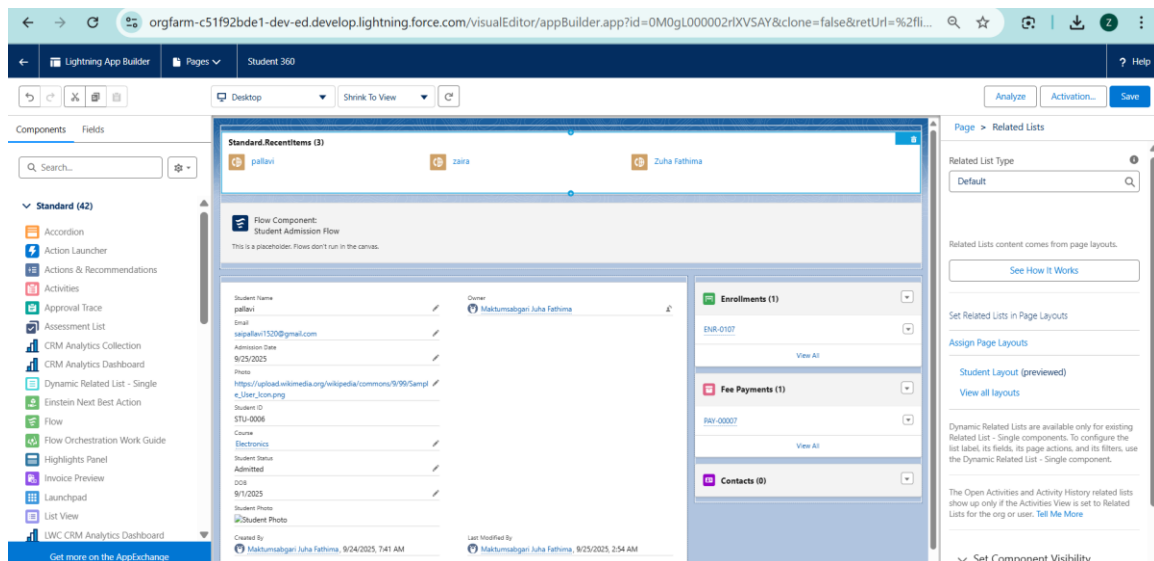
User Interface Development

Goal

Provide a user-friendly interface for managing students, enrollments, and fee payments in Salesforce.

Lightning Record Page for Student (Student 360 Page)

- Navigate to Setup → Object Manager → Student → Lightning Record Pages.
- Create a new App Page or Record Page.
- Add components:
 - Highlights Panel → Student's Name, Email, Status.
 - Related Lists: Enrollments, Fee Payments.
 - Tabs → Info | Related | Fee Calculator (if LWC is added).
- Deliverable → Screenshot of Student 360 Page showing student details + related lists.



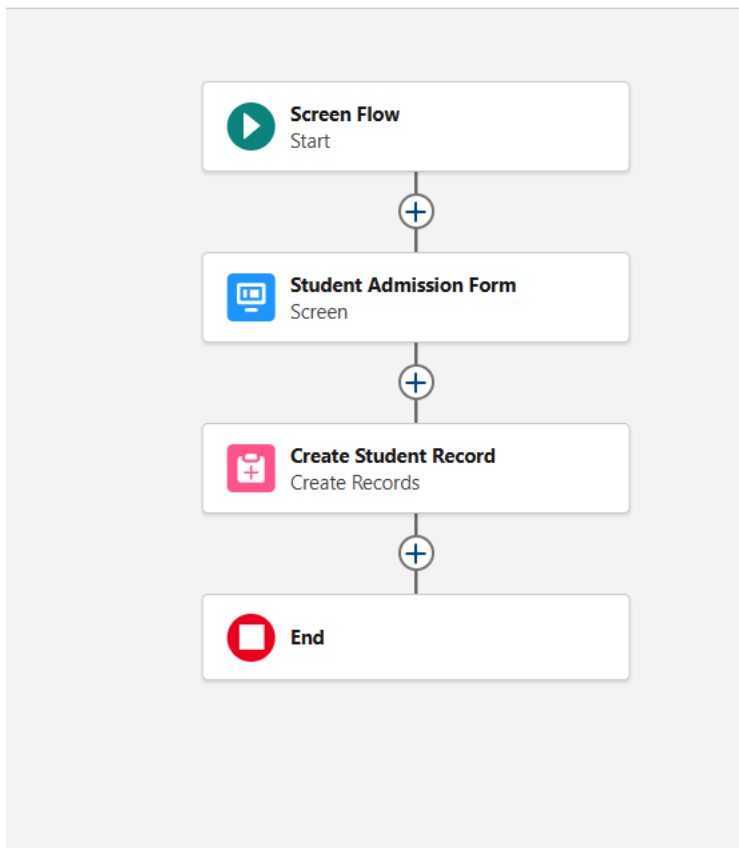
Screen Flow – Student Admission Form

Purpose: Let admins create a new student easily via a guided form.

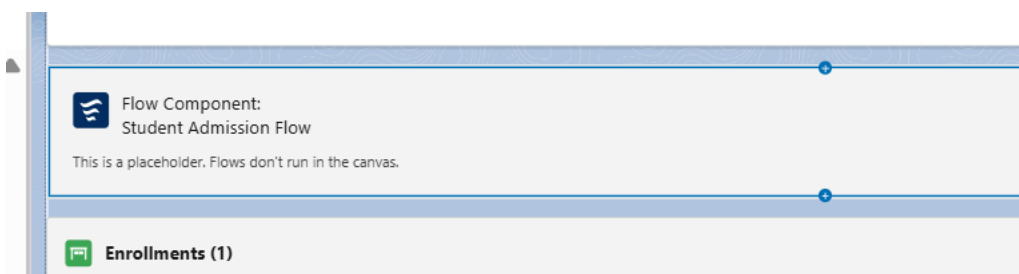
Steps:

1. Go to Setup → Flow → New Flow → Screen Flow.

2. Screen Element: Add fields for student details (Name, Email, Address, Course selection).
3. Create Records Element:
 - Object = Student
 - Map input fields to Student fields.
4. Connect elements → Save → Activate the Flow.



5. Add Flow to Student App Page using Flow Component.



Result: Admin can admit students using a simple UI form.

Lightning Web Component – Fee Calculator

Purpose: Allow quick calculation of fees (e.g., Course Fee – Paid Amount = Outstanding Fee).

LWC Development Steps:

1. Create Project (in VS Code Command Palette):
SFDX: Create Project with Manifest
2. Create LWC:
SFDX: Create Lightning Web Component
Name → feeCalculator
3. Edit Files:
- feeCalculator.html → Create form (inputs for Course Fee & Paid Fee).

Code:

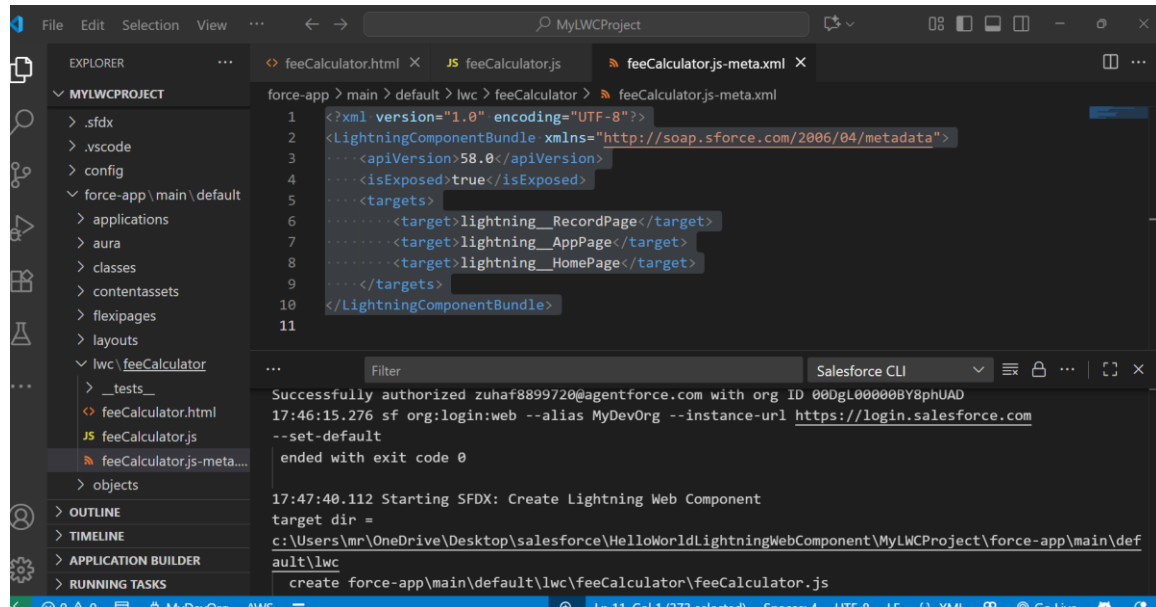
```
<template>
  <lightning-card title="Fee Calculator">
    <div class="slds-m-around_medium">
      <lightning-input
        label="Course Fee"
        type="number"
        value={courseFee}
        onchange={handleCourseFeeChange}>
      </lightning-input>

      <lightning-input
        label="Paid Amount"
        type="number"
        value={paidAmount}
        onchange={handlePaidAmountChange}>
      </lightning-input>

      <lightning-button
        label="Calculate"
        onclick={calculateOutstanding}
        class="slds-m-top_small">
      </lightning-button>

      <template if:true={outstandingFee}>
        <p class="slds-m-top_medium">
          <b>Outstanding Fee:</b> {outstandingFee}
        </p>
      </template>
    </div>
  </lightning-card>
</template>
```

```
</div>
</lightning-card>
</template>
```



- feeCalculator.js → Logic to calculate outstanding fee.

Code:

```
import { LightningElement } from 'lwc';

export default class FeeCalculator extends LightningElement {
  courseFee = 0;
  paidAmount = 0;
  outstandingFee;

  handleCourseFeeChange(event) {
    this.courseFee = parseFloat(event.target.value) || 0;
  }

  handlePaidAmountChange(event) {
    this.paidAmount = parseFloat(event.target.value) || 0;
  }

  calculateOutstanding() {
```

```
        this.outstandingFee = this.courseFee - this.paidAmount;
    }
}
```

- feeCalculator.js-meta.xml → Expose component to Record Page.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
  <apiVersion>58.0</apiVersion>
  <isExposed>true</isExposed>
  <targets>
    <target>lightning__RecordPage</target>
    <target>lightning__AppPage</target>
    <target>lightning__HomePage</target>
  </targets>
</LightningComponentBundle>
```

4. Deploy Component:

sf project deploy start --metadata LightningComponentBundle:feeCalculator -o

```
17:47:40.212 Finished SFDX: Create Lightning Web Component
17:50:27.412 Starting SFDX: Deploy This Source to Org

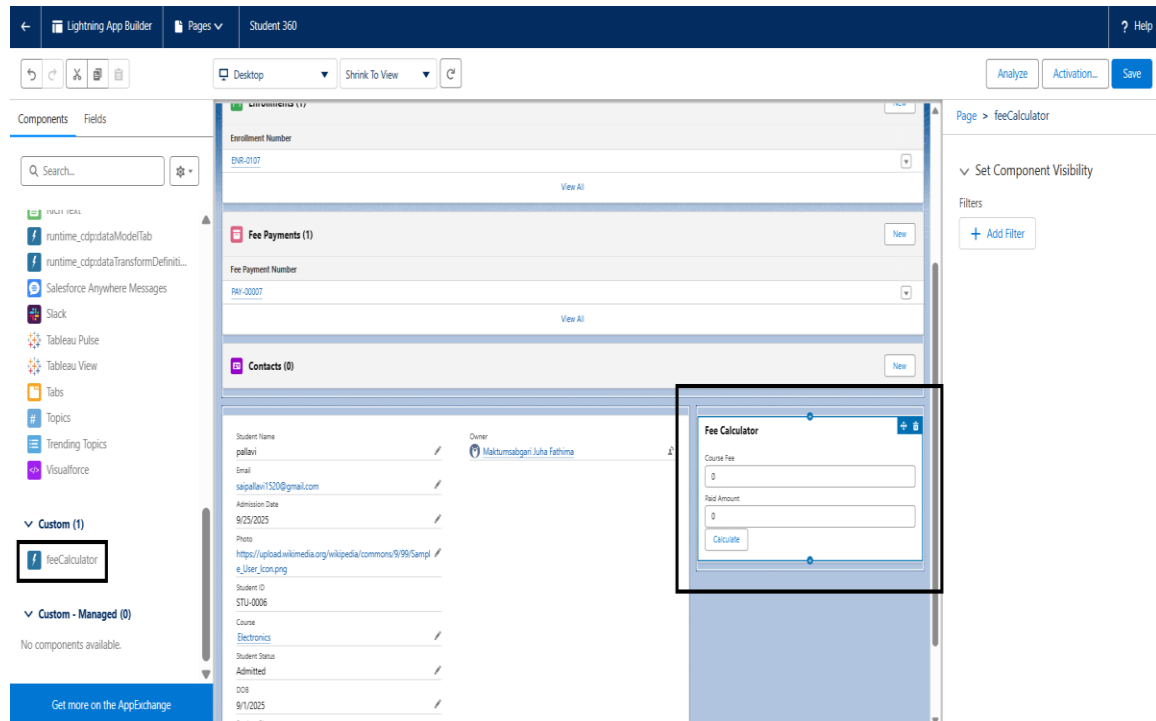
=== Deployed Source
STATE      FULL NAME                                TYPE                                PROJECT
PATH
-----
Created feeCalculator LightningComponentBundle
force-app\main\default\lwc\feeCalculator\feeCalculator.html
Created feeCalculator LightningComponentBundle
force-app\main\default\lwc\feeCalculator\feeCalculator.js
Created feeCalculator LightningComponentBundle
force-app\main\default\lwc\feeCalculator\feeCalculator.js-meta.xml

17:50:30.877 Ended SFDX: Deploy This Source to Org
```

MyDevOrg

5. Add to Lightning Page:

- Go to Lightning App Builder → Drag feeCalculator onto Student Page.



Result: Admins can instantly calculate and display outstanding fees inside the Student 360 page.

Results

- **Student 360 Page:** Displays student info and related records in a tabbed layout.
- **Screen Flow:** Enables guided student admission.
- **Fee Calculator LWC:** Interactive tool for fee management.
- **Overall Outcome:** Clean, intuitive, and functional UI for managing student-related data in Salesforce.